МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
И УПРАВЛЕНИЯ В ТЕХНИЧЕСКИХ СИСТЕМАХ

Кафедра информационных систем

В.Ю. Карлусов

ИССЛЕДОВАНИЕ ОПЕРАЦИЙ И МЕТОДЫ ОПТИМИЗАЦИИ

Учебное пособие

Севастополь
СевГУ
2019

УДК 303.725.35: 519.8
К23

Р е ц е н з е н т ы:
Н.А. Греков - доктор техн. наук, руководитель СНКТБ
института природно-технических систем, г. Севастополь
А.А. Ярошенко - доктор физ.-мат. наук, профессор кафедры
высшей математики СевГУ

Ответственный за выпуск:
И.П. Шумейко - заведующий кафедрой информационных систем СевГУ,
кандидат физ.-мат.наук, доцент

Карлусов В.Ю.
К23 Исследование операций и методы оптимизации: учеб. пособие / В.Ю. Карлусов. – Севастополь: СевГУ, 2019. – 315 с.

Цель учебного пособия: устранение дефицита теоретических положений дисциплины с целью наиболее полного овладения математическим и вычислительным аппаратами, применяемыми в организационно-техническом управлении. Предназначено для студентов и магистрантов направлений и профилей 09.03.02 (09.04.02) – “Информационные системы и технологии” и 09.03.03 (09.04.03) – “Прикладная информатика” всех форм обучения.

УДК 303.725.35: 519.8

Учебное пособие рассмотрено и утверждено на заседании кафедры Информационных систем, протокол № 07 от 28 января 2019 г.

Учебное пособие рассмотрено и рекомендовано к изданию на заседании Учёного Совета Института информационных технологий и управления в технических системах, протокол № 04 от 28 февраля 2019 г.

© ФГАОУ ВО «Севастопольский
государственный университет», 2019

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ 6
1. ПРЕДМЕТ И ТИПОВЫЕ ЗАДАЧИ ДИСЦИПЛИНЫ 7
1.1. Основные определения 7
1.2. Основная задача исследования операций 8
1.3. Математические модели в исследовании операций 9
1.4. Типовые задачи исследования операций 10
2. МАТЕМАТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ 19
2.1. Задачи математического программирования (ЗЛП) 21
2.2. Линейное программирование 21
2.2.1. Построение математической модели 22
2.2.2. Решение ЗЛП графическим методом 23
2.2.3. Расширенная (каноническая) форма записи ЗЛП 27
2.2.4. Определения и теоремы линейного программирования . . . 29
2.2.5. Решение ЗЛП прямым симплекс-методом 31
2.2.6. Решение ЗЛП методом искусственного базиса 41
2.2.7. Решение ЗЛП модифицированным симплекс-методом 46
2.2.8. Двойственность в ЗЛП 52
2.2.9. Формальная связь прямой и двойственной задач 53
2.2.10. Теоремы двойственности 54
2.2.11. Решение ЗЛП двойственным симплекс-методом 55
2.2.12. Вопросы для самоконтроля 59
2.3. Дискретное программирование 61
2.3.1. Решение задачи ЛЦП методом Гомори 62
2.3.2. Решение задачи ЛЦП методом ветвей и границ 66
2.3.3. Вопросы для самоконтроля 70
2.3.4. Транспортные задачи (ТЗ) 71
2.3.4.1. Постановка ТЗ и общий принцип её решения методом потенциалов 71
2.3.4.2. Нахождение начального опорного плана ТЗ методом северо-западного угла 75
2.3.4.3 Нахождение начального опорного плана ТЗ методом минимальной стоимости 76
2.3.4.4. Нахождение начального опорного плана ТЗ методом Фогеля (методом штрафов) 78
2.3.4.5. Алгоритм решения ТЗ методом потенциалов 81
2.3.4.6. Алгоритм решения ТЗ венгерским методом 88
2.3.4.7. Алгоритм решения ТЗ с ограниченной пропускной способностью коммуникаций 98
2.3.4.8. Решение задачи о назначениях 106
2.3.4.9. Вопросы для самоконтроля 113

2.4. Решение задач параметрического программирования. 115
2.4.1. Решение задачи линейного программирования при параметрических изменениях вектора ограничений 116
2.4.2. Решение задачи линейного программирования при вариации коэффициентов целевой функции 122
2.4.3. Вопросы для самоконтроля 130
2.5. Нелинейное программирование (НП – программирование) . . 131
2.5.1. Аналитические методы определения экстремумов 131
2.5.2. Методы поиска экстремумов в задачах без ограничений или в случае ограничений с разделяющимися переменными 135
2.5.2.1. Прямые методы поиска 136
2.5.2.2. Градиентные методы поиска 148
2.5.3. Поиск экстремумов в задачах нелинейного программирования при ограничениях типа “равенство” (метод Лагранжа) 156
2.5.4. Общий случай задачи нелинейного программирования . . . 159
2.5.4.1. Седловая точка в НП-задачах 161
2.5.4.2. Применение теоремы Куна-Таккера НП-задачам. . . . . .
2.5.5. Методы возможных направлений 165
2.5.5.1. Метод Зойтендейка 165
2.5.5.2. Метод проекции градиента Розена 170
2.5.6. Методы штрафных функций 174
2.5.6.1 Метод барьерных поверхностей (МБП) 176
2.5.6.2 Метод внешней точки 178
2.5.7. Задачи квадратичного программирования 179
2.5.8. Вопросы для самоконтроля 184
3. ЭЛЕМЕНТЫ ТЕОРИИ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ (СМО) 187
3.1. Марковские цепи и потоки событий 187
3.2. Простейший поток событий 189
3.3. Математические модели потоков 190
3.4. Модель Колмогорова для описания систем с вероятностными состояниями 192
3.5. Схема “гибели-размножения” и её модель 193
3.6. Понятие СМО. Формулы Литтла 195
3.7. Примеры СМО. Одноканальная СМО с отказами 197
3.8. Примеры СМО. Многоканальная СМО с отказами 199
3.9. Примеры СМО. Одноканальная СМО с неограниченной очередью 201
3.10. Пример решения практической задачи 204
3.11. Сводные показатели эффективности СМО 206

3.12. Вопросы для самоконтроля 207
4. ЭЛЕМЕНТЫ ТЕОРИИ ИГР 209
4.1. Основные определения 209
4.2. Формальное описание игры двух персон с нулевой суммой . . 210
4.3 Седловая точка и оптимальные стратегии 211
4.4. Понятие о смешанных стратегиях 213
4.5. Теорема об активных стратегиях 214
4.6. Поиск оптимальных стратегий 217
4.6.1. Графоаналитический метод решения игровых задач 218
4.6.2. Использование принципа доминирования для снижения размерности платёжной матрицы игровой задачи 223
4.6.3. Построение эквивалентной ЗЛП по платёжной матрице . . . 225
4.6.4. Итерационный метод решения матричной игры с нулевой суммой 227
4.7. Конечные позиционные игры двух персон 230
4.8. Многошаговые игры 241
4.8.1. Детерминированные игры 242
4.8.2. Стохастические игры 248
4.8.3 Рекурсивные игр 252
4.9. Бесконечные игры 254
4.9.1. Выпуклые и вогнутые игры 259
4.9.2 Игры с выбором момента времени действия в условиях полной информации (шумные дуэли) 266
4.9.3 Игры с выбором момента времени действия в условиях неполной информации (бесшумные и смешанные дуэли) . . 271
4.10. Игровые модели неантагонистических конфликтов (биматричные игры) 272
4.10.1. Некооперативные биматричные игры 273
4.10.2. Кооперативные игры 272
4.10.3. Содержательные примеры биматричных игр 289
4.11. Вопросы для самоконтроля 295
Заключение 296
БИБЛИОГРАФИЧЕСКИЙ СПИСОК 297
ПРИЛОЖЕНИЕ А. ПОСТОПТИМАЛЬНЫЙ АНАЛИЗ 303

ПРЕДИСЛОВИЕ

Аппарат математического программирования имеет широкий круг приложения: экономико-математические исследования, военное дело, проектирование автоматизированных систем, организационное управление et c. В его основу положены разнообразные оптимизационные алгоритмы, изучение которых, наряду с освоением вычислительных процедур, должно в немалой степени включать особенности применения и условия использования указанных алгоритмов.

В основу данного учебного пособия был положен опыт многолетнего преподавания дисциплин, связанных с методами исследования операций, на кафедре Информационных систем Севастопольского университета.

Анализ литературы в данной предметной области выявил тенденции, состоящие в том, что часть источников ориентирована на научных работников и специалистов в области прикладной математики, как это указывается в аннотации изданий, а другая часть — на экономистов. Поэтому, для “математической” группы книг характерна строгая доказательность предпосылок возникновения, шагов (этапов) и ожидаемых результатов применения изучаемых алгоритмов. “Экономисты” же довольствуются “пошаговыми инструкциями”, представляющими собой далеко не лучшие, из возможных реализаций, алгоритмы. Несколько компромиссными, по критерию симбиоза высшей математики и прикладной алгоритмизации, являются учебники Ю.П. Зайченко, адресованные, однако, прикладным математикам.

Поэтому в настоящем учебном пособии опущены доказательства теорем, и оставлены лишь те теоретические положения, которые отвечают за целостное восприятие излагаемого материала, его логическую связность и встроенность в общую картину мироздания. Отсутствие доказательств теорем компенсируется подробным (пошаговым) анализом вычислительного процесса.

Приводятся “сквозные” примеры, которые позволяют ярче выявить особенности осваиваемых методов.

По замыслу автора, после изучения настоящего учебного пособия студент должен

- **знать**: основные методы математического программирования и исследования операций;
- **уметь**: производить построение математических моделей и формулировать целевые функции в терминах исследования операций для решения экстремальных технических задач управления;
- **иметь**: навыки фактического применения оптимизационных методов.

1. ПРЕДМЕТ И ТИПОВЫЕ ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Основные определения [2]

Понятие **системы** в рамках курса заключается в следующем:

- имеется некоторый объект управления (или наблюдения);
- присутствует система сбора и обработки данных наблюдений над объектом;
- организованы каналы связи и воздействия на объект;
- существует управляющий (задающий) орган.

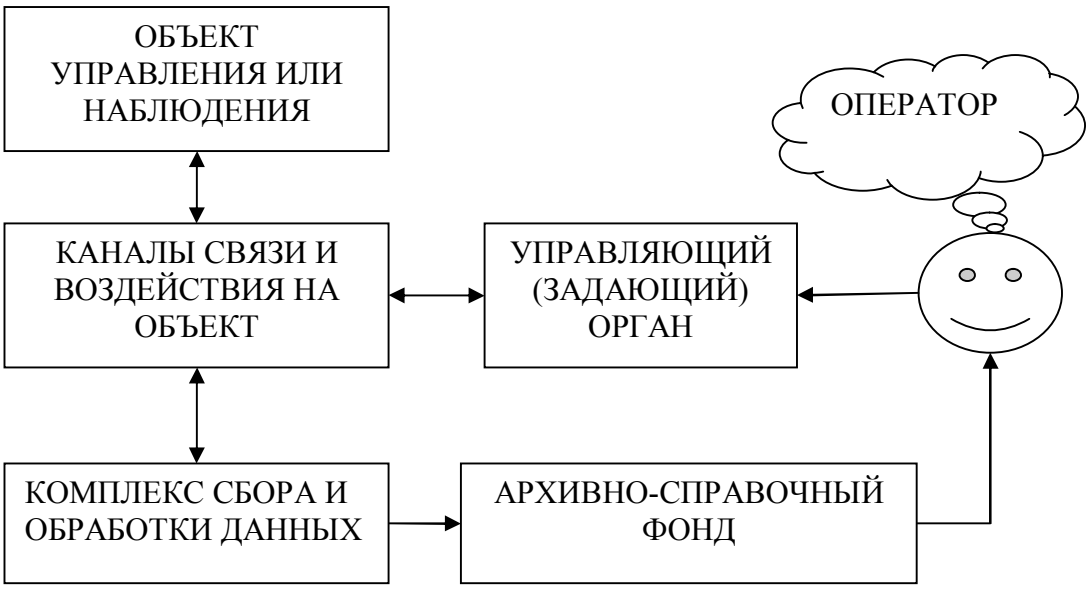


Рисунок 1.1 – Типовая схема автоматизированной системы

Как видно из набора компонентов, такая структура присуща, как правило, всем системам управления, в той или иной модификации [28]. Основное отличие систем, рассматриваемых в рамках курса в том, что они являются **автоматизированными**, а **не автоматическими**, т.е. в контур управления включен **оператор**.

Поэтому в общих чертах можно говорить об **организационно техническом управлении**, сочетающем в себе классические методы и способы обработки данных с необходимостью **принятия решений**, а сами системы получили название **автоматизированных систем управления**.

В данных системах оптимизации подвергаются:

- отдельные элементы систем;
- алгоритмы управления и обработки;
- структуры связей (соединения) между элементами.

”Исследование операций представляет собой искусство давать плохие ответы на практические вопросы, на которые даются еще худшие ответы другими методами”.

Т.Л. Саати

1.2. Основная задача исследования операций

Операция – последовательность действий, направленных на достижение какой либо цели.

Критерий эффективности – показатель совпадения (соответствия) цели операции и состояния системы.

Стратегия – способ расстановки сил и средств при проведении операции.

Математическая модель операции – это формальные соотношения, устанавливающие связь ***критерия*** и ***стратегии***.

Решение – множество параметров стратегии, полученных на основании математической модели.

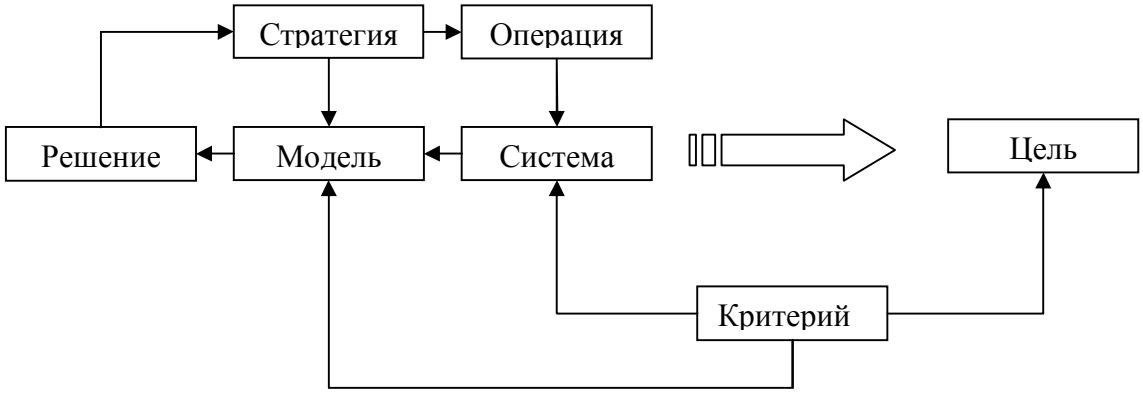


Рисунок 1.2 – Основные определения

Основная задача исследования операций: нахождение для выбранной ***математической модели*** решения, при котором критерий эффективности достигает экстремума (**min** или **max**).

Основные черты операционного подхода заключаются в следующем.

Чтобы подход можно было квалифицировать как операционный, он должен включать:

1. Ориентацию на принятие решения. (Расчеты должны иметь непосредственное и определенное отношение к выбору способа действия) (стратегии).

2. Оценку на основе критериев эффективности – сравнение на количественных оценках, позволяющих однозначно определить полезность ожидаемого подхода.

3. Доверие к математической модели. (Однозначность трактовки математической модели; опираясь на одни и те же данные, специалисты должны получать одинаковые результаты).

4*. Необходимость использования ЭВМ. (Это не благое пожелание, а обуславливается сложностью используемых математических моделей либо громоздкостью процедур, обеспечивающих те или иные системы наблюдения, управления или контроля).

1.3. Математические модели в исследовании операций [70]

1. ***Детерминированные модели*** операций. Их основные черты:

- полная определенность в настоящем, прошлом и будущем;
- повторяемость результатов при одинаковых исходных данных.

2. ***Вероятностные модели*** (статистические, стохастические):

- введены случайные факторы воздействия;
- возможны случайные реакции системы на детерминированные воздействия (Как в фантастическом рассказе, где некий правитель использовал следующий механизм наложения резолюций – “Утвердить”, “Отвергнуть”, “Вернуть на доработку”, данные резолюции налагались случайным порядком).

3. ***Игровые модели***:

- наличие конфликтной ситуации;
- моделирование условий конфликта;
- получение на основании модели оптимального способа поведения в конфликтной ситуации.

4. ***Неполные модели***. Строятся на основании ограниченного набора экспериментальных данных и экспертных оценок. Пополняются и корректируются по мере накопления данных

5. ***Эвристические модели***. В их основании лежат опыт или наблюдения, как правило, не поддающиеся математическому описанию. Как говорил К. Прутков [58]: “Щёлкни лошадь в нос, она махнет хвостом”.

6. ***Имитационные модели***. Это, обычно, программные, технические или технико-программные модули, ориентированные на проведение экспериментальных исследований.

1.4. Типовые задачи исследования операций [70]

1.4.1 Задачи поиска (поисковые задачи)

Постановка задачи.

В распоряжение пользователя предоставляется комплекс технических и программных средств при ограничениях на время или стоимость эксплуатационного оборудования.

Цель оптимизации.

Распределение ограниченных ресурсов таким образом, чтобы максимизировать вероятность нахождения объекта.

Таким образом, оптимизации подвергаются

- структуры хранения,
- алгоритма поиска,
- технические средства поиска как аргумент функции эффективности.

Классификация поисковых задач приводится на рисунке 1.3.

Рисунок 1.3 – Схема классификации задач поиска

1.4.2 Задачи распределения или распределительные задачи [69, 70]

Постановка задачи.

- У пользователя в наличии ограниченный объём или число ресурсов, которые отведены на осуществление комплекса операций для достижения поставленной цели.
- Предоставляется возможность выполнять операции можно выполнять различными (по эффективности) способами.
- Однако, не все операции можно выполнить с максимальной эффективностью вследствие ограниченности ресурсов, из-за чего и возникает данный тип оптимизационных задач.

Цель оптимизации.

Необходимо выбрать такое распределение ресурсов по операциям, при котором будет достигнута наибольшая суммарная эффективность.

Классификационная схема задач распределения приводится на рисунке 1.4. Частные задачи: выбор оптимального типажа, задача о раскрое, задача о рюкзаке.

Рисунок 1.4 – Классификация задач распределения

1.4.3 Задача управления запасами (ресурсами) [33, 57, 69]

Постановка задачи.

- Имеются некоторые запасы, содержание которых является функцией их величины.
- Расход запасов производится тем или иным образом (согласно заданной дисциплине)
- Отсутствие запаса штрафуется (или не допускается).

Цель оптимизации.

Определить размер запаса, оптимальный в смысле минимизации общих затрат.

Характеристики задач управления запасами показаны на рисунке 1.5.

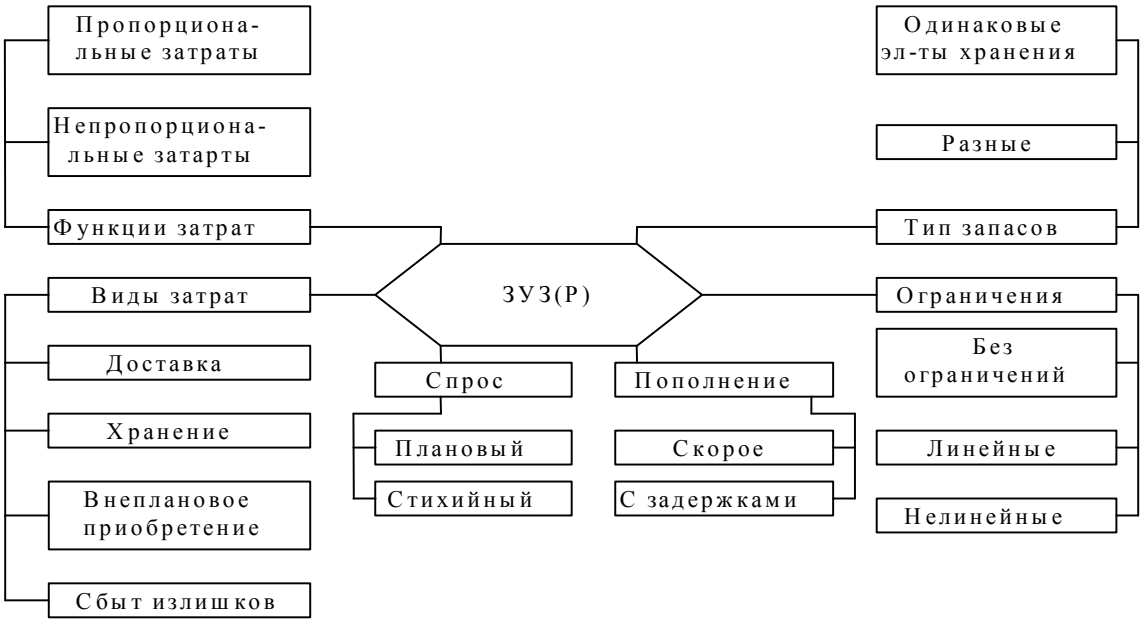


Рисунок 1.5 – Классификация задач управления запасами

1.4.4. Задачи теории систем массового обслуживания [4, 21, 29]

Постановка задачи.

Пусть на входе некоторого устройства имеется очередь из заявок, запросов или требований на использование (применение) этого устройства в определённом режиме.

Априори или гипотетически известны статистические характеристики указанных запросов: например, их среднее число в единицу времени, средняя продолжительность использования ими устройства.

Цель оптимизации.

Необходимо обеспечить обслуживание всех запросов или заданной производительности работы оборудования.

Для пояснения классификации используется схема, приведённая на рисунке 1.6.

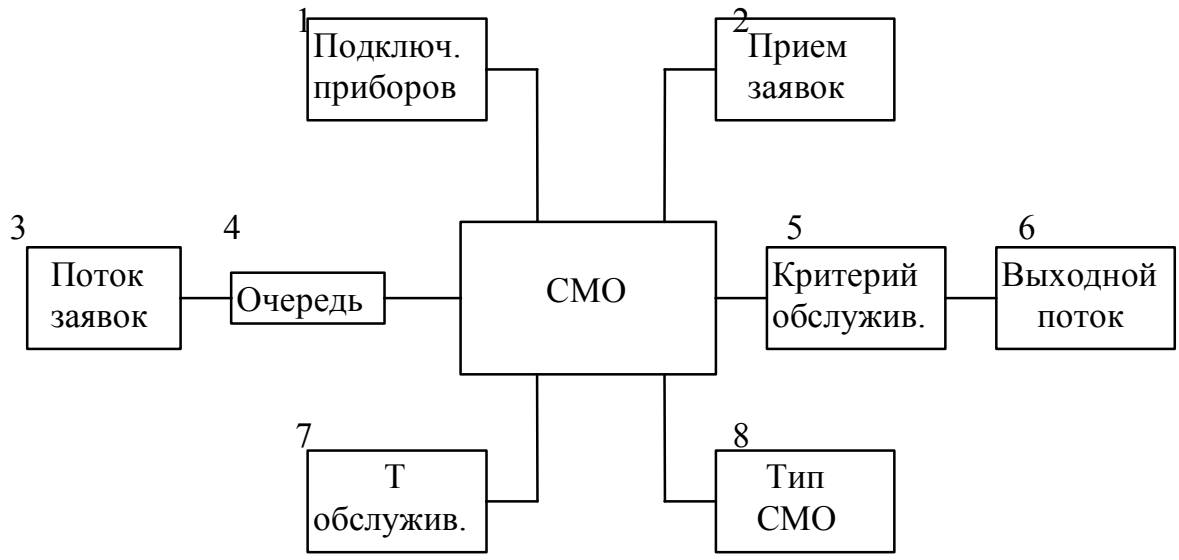


Рисунок 1.6 – Задачи теории массового обслуживания

1. Подключение приборов:
- в порядке номеров;
- по мере освобождения;
- случайным порядком.

2. Дисциплина приёма заявок [42, 43, 53]:
- по мере поступления (FIFO);
- случайным порядком;
- по установленным приоритетам;
- в порядке, обратном поступлению (LIFO);

- приём по определенным каналам.

3. Характеристики потока заявок:

- стационарный или не стационарный;
- с последействием и без оного;
- ординарный и не ординарный.

4. Характеристики очереди:

- с отказами и без отказов;
- с ограниченным временем ожидания и не ограниченным временем ожидания;
- с ограниченной длиной и не ограниченной длиной.

5. Критерии обслуживания заявок [60, 61]:

- вероятность пропуска (задержки) заявки;
- математическое ожидание числа задержанных заявок;
- математическое ожидание числа занятых каналов;
- математическое ожидание длины очереди.

6. Вид выходного потока заявок:

- установившийся процесс;
- не установившийся процесс.

7. Распределение времени обслуживания:

- случайное (экспоненциальный закон, закон Эрланга, прочие законы);
- неслучайное (квантованное).

8. Характеристик системы:

- однородная (неоднородная);
- однофазная;
- многофазная;
- одноканальная;
- многоканальная.

1.4.5 Теория игр [1, 11, 14 -16]

Теория игр – математическая теория конфликтных ситуаций, в которых сталкиваются интересы двух или более сторон преследующих различные цели (описываемых разными целевыми функциями).

Цель оптимизации.

Отыскание разумной стратегии поведения игроков в конфликтной ситуации.

Игры классифицируются, как это показано на рисунке 1.7, следующим образом.

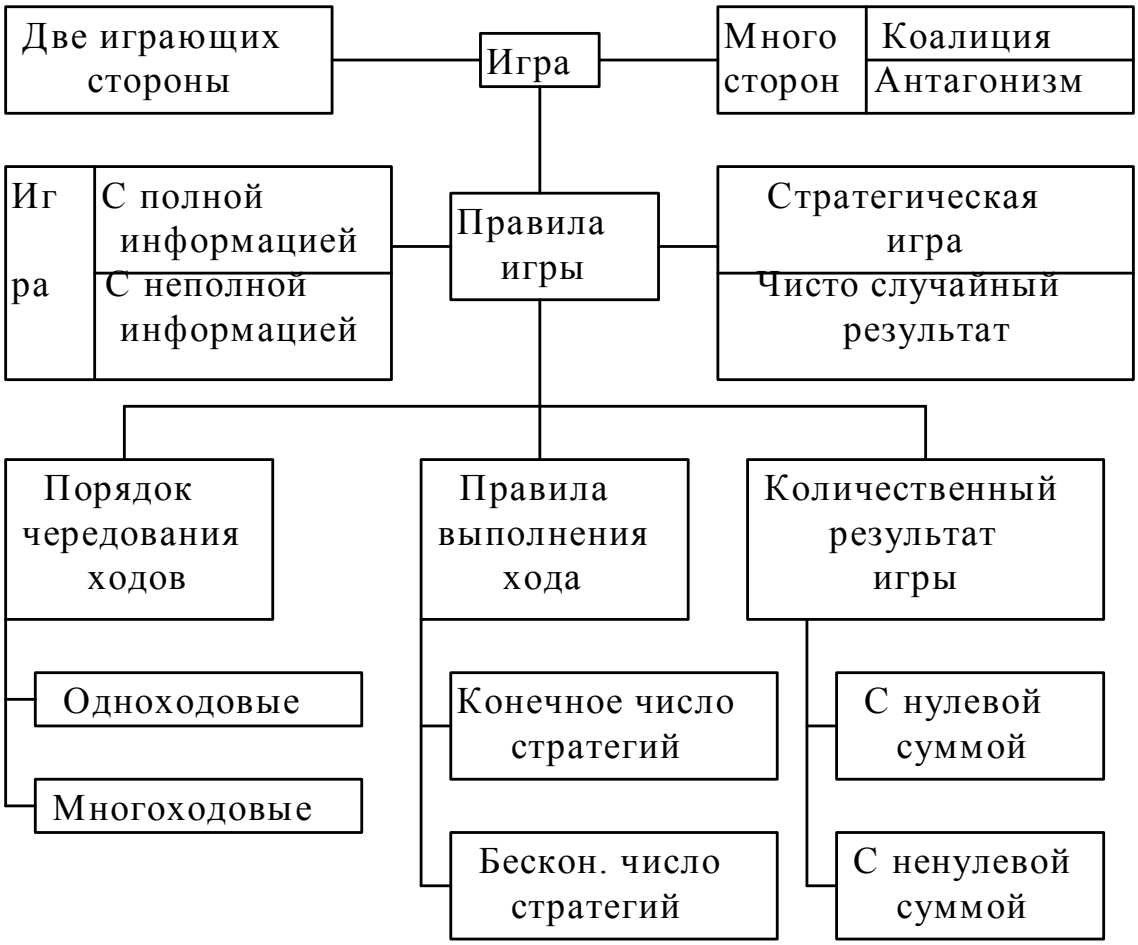


Рисунок 1.7 – Схема классификации задач теории игр

1.4.6 Задача замены оборудования [54]

Постановка задачи.

Имеется техническое устройство (или комплекс устройств), функциональные блоки которого подвержены отказам вследствие износа или перестают отвечать различным требованиям по причине материального старения и т.п.

Цель оптимизации.

Отыскание оптимальных сроков замены тех или иных технических устройств (их узлов, блоков) и определение потребностей в комплектующих для поддержания их в рабочем состоянии.

Пояснения дадим с использованием рисунка 1.8, по номерам блоков.

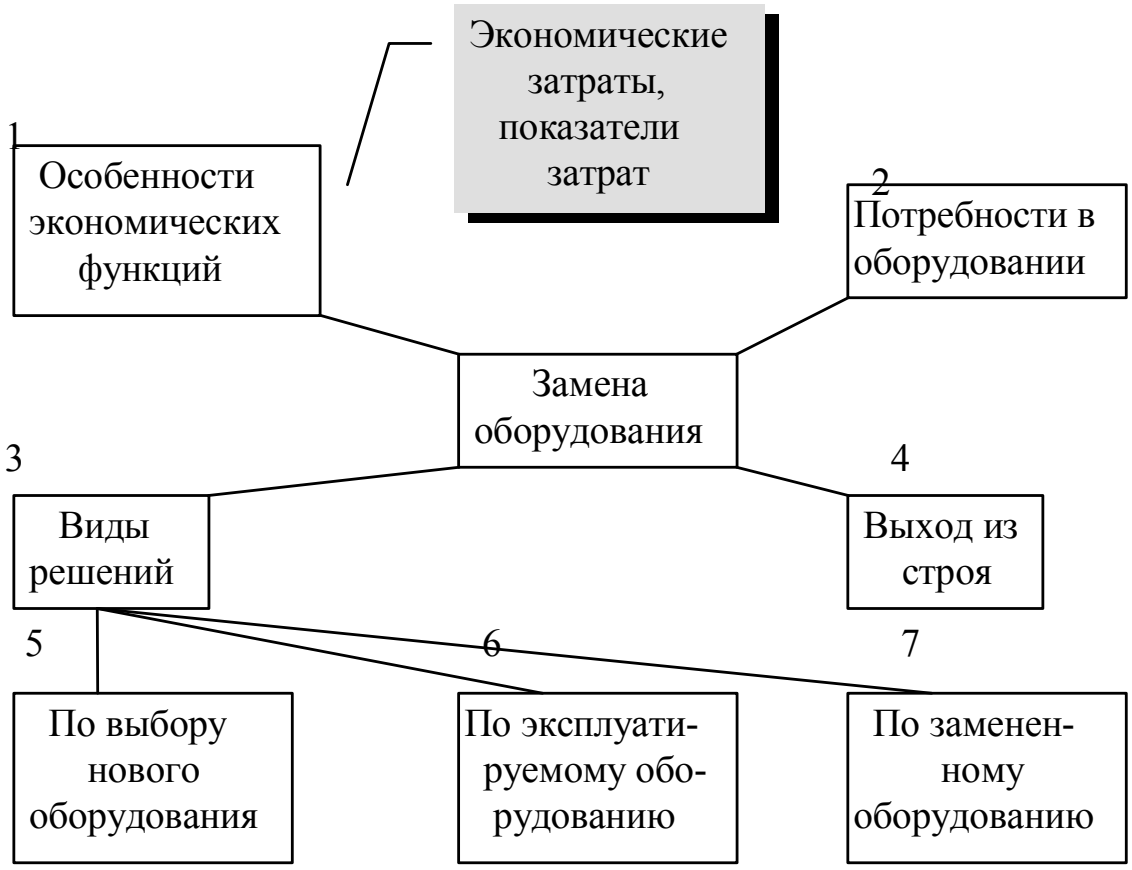


Рисунок 1.8 – Классификация задач о замене оборудования

1. Особенности экономических функций (экономические показатели):

- учет амортизации;
- учет роста эксплуатационных расходов;

- учет технического прогресса;
- учет эффективности капвложений;
- учет потерь от ремонта.

2. Потребности в оборудовании:

- переменные (спорадические);
- постоянные.

4. Выход из строя:

- через заданный срок.
- случайные сроки;
- смешанный процесс;
- не выходит из строя.

5. Выбор нового оборудования:

- старого типа;
- нового оборудования заданного типа;
- нового выбираемого типа;
- специально разрабатываемого типа.

6. Эксплуатация оборудования:

- производство осмотров разной сложности;
- производство регламентных работ;
- замена оборудования: а) плановым порядком, б) по выходу из строя, в) смешанным порядком.

7. Использование замененного оборудования:

- не реализуется (списывается);
- продается;
- используется в менее ответственных местах: а) одноэтапный перевод; б) многоэтапный перевод.

1.4.7 Задачи упорядочения (Задачи теории расписаний) [24, 38, 75]

Постановка задачи.

Имеется некоторый объем работ, подчиненный решению какой либо комплексной задачи. Причем отдельные работы могут быть связаны сроками выполнения, другие могут выполняться независимо.

Цель оптимизации.

Требуется выбрать оптимальный порядок действий и темпы выполнения отдельных работ для минимизации непроизводительных затрат.

Общая характеристика задач[76, 90]:

- все задачи связаны с необходимостью выбора оптимального порядка действий;
- оптимальность понимается в смысле минимума какого-либо критерия при заданных ограничениях;
- аналитические методы созданы лишь для простейших случаев;
- выбор оптимального маршрута не имеет общего решения (задача коммивояжера);
- применяются сетевые методы планирования и методы оптимизации на графах.

Сетевые методы планирования оперируют понятиями:

- работа;
- сеть;
- критический путь;
- события.

На их основании строится графическое изображение последовательности работ – сетевой график.

Сетевой график

- подвергается оптимизации после нахождения критического пути;
- используется не только при планировании, но и в ходе выполнения последовательности работ для контроля.

При этом обязательно существуют:

- минимальное время, уменьшить которое нельзя, как бы велики ни были затраты на выполнение работы;
- минимальная стоимость выполнения работы, уменьшить которую путем увеличения времени невозможно.

2. МАТЕМАТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

Математическое программирование – дисциплина, которая занимается изучением экстремальных задач и поисками методов их решения.

Указанная область исследования операций весьма схематично и приближённо представлена на рисунке 2.1, позаимствованном из [69]. Наименование блоков, представленных нумерацией, суть следующее.

1. Классические методы математического анализа.
2. Динамическое программирование.
3. Принцип Максимума Понтрягина.
4. Дискретный принцип максимума.
5. Ограничения отсутствуют либо в форме равенства.
6. Много ограничений
7. непрерывный процесс (8 – 11 задачи математического программирования)
8. Критерий $K = F(x_1, x_2, \ldots x_n)$ (минимизируется или максимизируется)
9. Ограничения
10. Число этапов
11. Тип процесса
12. Линейная функция
13. Нелинейная функция
14. В форме неравенств
15. Один
16. Много
17. Случайный
18. Выпуклая
19. Вогнутая
20. Требования целочисленности значений переменных
21. Линейное программирование
22. Нелинейное программирование
23. Целочисленное программирование
24. Стохастическое программирование
25. Блочное программирование
26. Параметрическое программирование

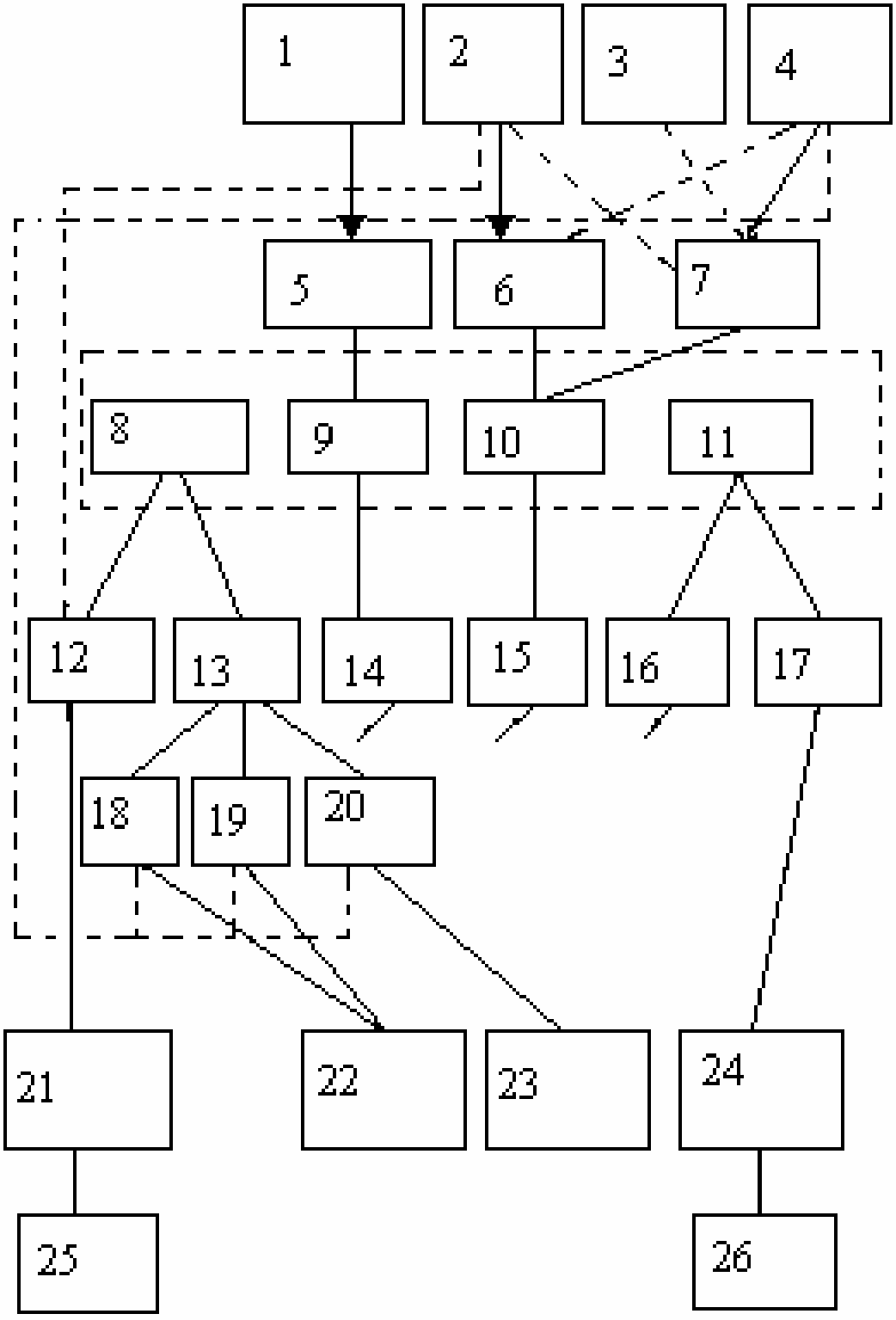


Рисунок 2.1 - Классификация задач математического программирования

2.1 Задачи математического программирования

Задачи математического программирования формируется следующим образом: [2, 17, 23, 26]

Пусть $f(x_1, x_2, \dots x_n)$ – функция нескольких переменных, а функции $g_i(x_1, x_2, \dots x_n) * b_i$, $i = 1, 2, \dots n$, – ограничения, связанные с переменными, где b_i – некоторое действительное число, а символ * – одно из ограничений вида $\geq$, $\leq$ или $=$.

Требуется найти минимум или максимум функции f при заданных ограничениях g_i . Функция f называется ***функцией цели*** или ***целевой функцией.***

В зависимости от вида функций f и g_i задача математического программирования бывает следующих видов:

1. Задача линейного программирования (ЗЛП), если f и g_i линейны.
2. Задача нелинейного программирования (ЗНП или НП-задача, НПЗ), если хотя бы одна из функций нелинейная.

Задачи делятся на классы:

1. Задача целочисленного программирования (ЦП), когда вектор переменных $X^T = \{x_1, x_2, \dots x_n\}$ целочисленный.
2. Задача параметрического программирования, если функции f и g_i зависят от параметров.
3. Если f – дробно-линейная функция, а ограничения g_i – линейны, то имеем задачу дробно-линейного программирования.
4. Стохастическое программирование, если среди компонентов функций f или g_i присутствуют случайные величины.
5. Динамическое программирование – многоэтапный процесс нахождения решения, при этом функции f и g_i зависит от времени или состояния (номера шага или этапа).

2.2. Линейное программирование [2, 3, 7, 24, 27]

Линейное программирование – решение экстремальных задач математического программирования с линейной зависимостью между переменными.

Важнейшие (основные) методы решения задач линейного программирования суть следующие:

1. Графический метод.
2. Прямой симплекс-метод (метод симплекс-таблиц).
3. Метод искусственного базиса.

4. Модифицированный симплекс-метод.

5. Двойственный симплекс-метод.

Далее нами будут подробно изложены указанные методы, их алгоритмические особенности, условия применения.

2.2.1. Построение математической модели [10, 70, 74 – 77]

Пусть имеем следующее описание задачи (содержательную постановку), позаимствованную в [10].

Фирма выпускает продукцию из картофеля трех видов: картофельные кубики, картофельные дольки жареные, картофельные чипсы. По технологии картофель сортируют по размерам и качеству и направляют на различные поточные линии. Исходный продукт покупается у двух поставщиков. Производство продукции и относительная прибыль (итоговая) на единицу закупки от продажи изделий сведены в таблицу. Каждый продукт не должен превышать в изготовлении соответствующий объем, в противном случае фирма не будет успевать осуществлять его доставку и продажу.

	Поставщики		
Продукция	I	II	Объём производства
Кубики	0,2	0,3	1,8
Дольки	0,2	0,1	1,2
Чипсы	0,2	0,3	2,4
Прибыль	5	6	

Требуется определить, какие объемы картофеля следует закупать у того или иного поставщика.

Чтобы построить математическую модель в терминах исследования операций, необходимо выполнить следующие шаги.

1. Определить переменные, для которых будет составлена целевая функция и ограничения на нее.

2. Сформулировать цель решение и составить функцию.

3. Составить ограничения задачи.

Продемонстрируем данную последовательность действий на изложенном выше содержательном примере.

Пусть k_1 - объем картофеля, закупаемого у первого поставщика, а k_2 - у второго. Тогда целевая функция (функция цели, или, сокращённо, ЦФ) будет выглядеть так:

$$f = 5k_1+6k_2 \rightarrow \max.$$

Поскольку речь идёт о прибыли, и, чем больше прибыль, тем лучше, то функция цели максимизируется.

Ограничения определяются строками таблицы, вид знака в ограничениях определяется словосочетанием “не должен превышать”:

$$\begin{cases} 0{,}2k_1 + 0{,}3k_2 \le 1{,}8; \\ 0{,}2k_1 + 0{,}1k_2 \le 1{,}2; \\ 0{,}3k_1 + 0{,}3k_2 \le 2{,}4; \\ k_1 \ge 0; \\ k_2 \ge 0. \end{cases}$$

Последние два ограничения называются ограничениями неотрицательности, они могут присутствовать в записи математической модели, а могут и опускаться, но всегда учитываются.

Важнейшими свойствами линейных моделей являются:

1) пропорциональность;

2) аддитивность.

То есть одна из эквивалентных моделей для данной задачи может иметь вид

$$f = 5x_1 + 6x_2 \to \max,$$

$$\begin{cases} 2x_1 + 3x_2 \le 18; \\ 2x_1 + 1x_2 \le 12; \\ 3x_1 + 3x_2 \le 24; \\ x_1 \ge 0; \\ x_2 \ge 0. \end{cases}$$

По сравнению с исходной, каждое из ограничений задачи пропорционально увеличено в десять раз. Свойство аддитивности означает возможность добавлять новые ограничения, паче такие возникнут. В данном примере эта возможность не понадобилась.

2.2.2. Решение ЗЛП графическим методом[3, 7, 33, 34]

Областью применения данного метода будут задачи с числом переменных равным двум.

Для демонстрации метода обратимся к задаче, сформулированной выше.

$$f = 5k_1 + 6k_2 \to \max,$$
$$\begin{cases} 0{,}2k_1 + 0{,}3k_2 \le 1{,}8; \\ 0{,}2k_1 + 0{,}1k_2 \le 1{,}2; \\ 0{,}3k_1 + 0{,}3k_2 \le 2{,}4; \\ k_1 \ge 0; \\ k_2 \ge 0. \end{cases}$$

Каждому из неравенств соответствует некая полуплоскость в координатах (k_1, k_2), ограниченная прямой, построенной для случая, когда в ограничениях знак неравенства заменён знаком равенства.

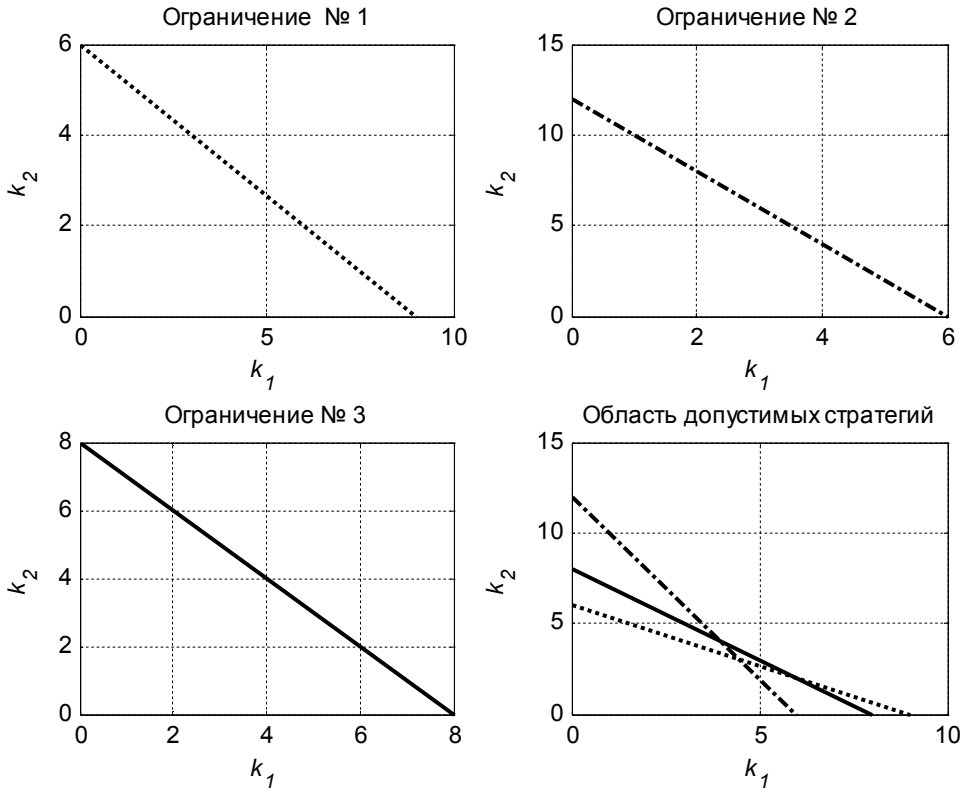


Рисунок 2.2 – Построение области ограничений

Для того, чтобы определить, по какую сторону от ограничивающей прямой находится область, необходимо подставить координаты точки начала (0, 0) в исходное ограничение и проследить, выполняется ли оно. Если неравенство выполняется, то точка (0, 0) принадлежит области. Следовательно, область распространяется, начиная от прямой, в направлении начала координат. Если неравенство не выполняется, начало координат (0, 0) не принадлежит области

Пересечение областей, соответствующих отдельным ограничениям, определяет область допустимых решений, называемой также областью допустимых стратегий. Ход построения областей иллюстрируется на рисунке 2.2.

Из приведённых рисунков видно, что неравенство № 3 не оказывает влияния на область допустимых стратегий закупок.

Рассмотрим целевую функцию $f(k_1, k_2)=5k_1+6k_2$. Выясним, существует ли экстремумы для данной функции?

Имеем $\frac{\partial f(k_1,k_2)}{\partial k_1} = c_1 = 5$ и $\frac{\partial f(k_1,k_2)}{\partial k_2} = c_2 = 6$.Это характеризует монотонно изменяющуюся функцию со скоростями изменения в указанных направлениях, определяемых этими частными производными.

Вектор, направленный от начала координат к точке с координатами (c_1, c_2), представляет собой нормаль к плоскости, определяемой целевой функцией *f*.

Нормаль ***перпендикулярна линии пересечения*** плоскости целевой функции с координатной плоскостью, а также ***проекциям линий равного уровня*** целевой функции на координатную плоскость.

Кроме того вектор нормали ***указывает направление возрастания*** целевой функции, а антинормаль, вектор, противоположный нормали –
– направление, в котором функция цели убывает.

Поэтому мы должны двигать перпендикуляр (который изображает местоположение равных значений функции) вдоль нормали, пока он не пересечёт границу области допустимых стратегий.

Направление движения определяется видом оптимизации:

- при решении задачи минимизации – от точки с координатами (c_1, c_2) к началу координат;
- при решении задачи максимизации – от начала координат к точке с координатами (c_1, c_2).

Обобщим наши рассуждения в виде алгоритма, приводимого в [3, 33, 34, 79, 83].

Алгоритм графического метода

1. Построить допустимое множество решений.

2. Построить нормаль к целевой функции и изобразить её проекцию на плоскости решений. Направление нормали указывают направление возрастания целевой функции.

3. Перемещать перпендикуляр к нормали до тех пор, пока он не достигнет крайней точки множества допустимых стратегий.

4. Определить значения координат крайней точки допустимого множества либо непосредственно по графику, либо по уравнениям ограничивающих прямых, пересекающихся в крайней точке.

5. Вычислить значение целевой функции, соответствующее оптимуму.

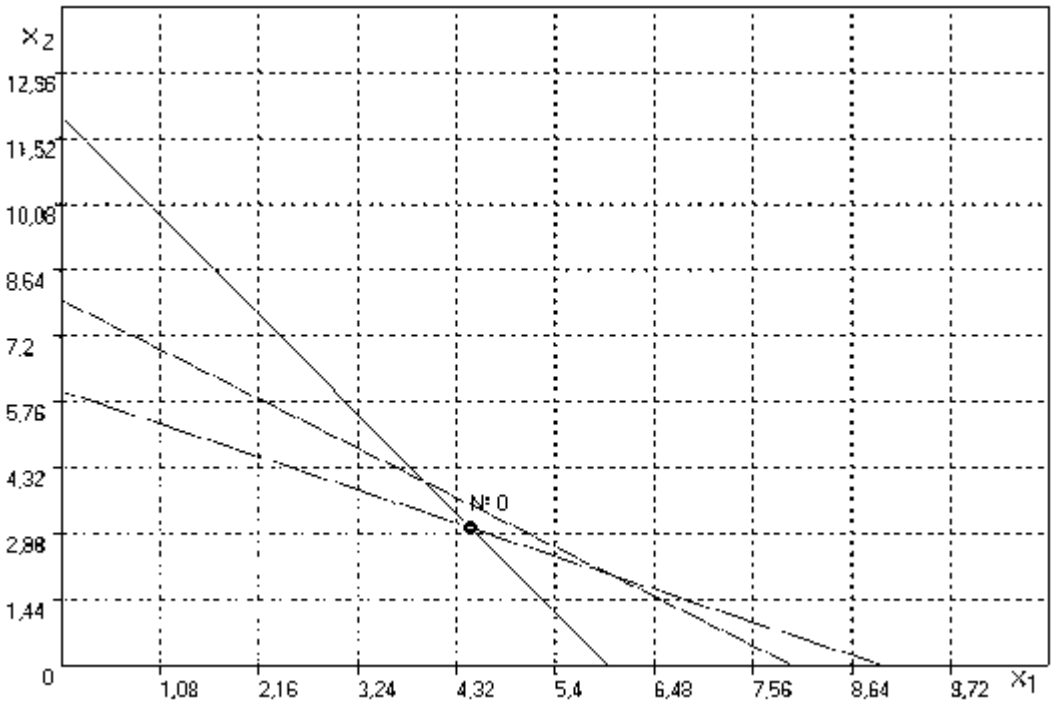


Рисунок 2.3 – Оптимум, найденный графическим методом

В ходе решения на различных наборах исходных данных иногда возникают ***частные случаи*** [33]:

1. Линия, ограничивающая область в направлении оптимизации, перпендикулярна нормали.
2. Область незамкнута в направлении оптимизации.
3. Несовместная система условий.
4. Невыпуклость области.

Алгоритм вполне распространим и на многомерный случай. В этом случае имеем:

- гиперплоскости;
- полупространства;
- выпуклое полиэдральное множество;
- выпуклый полиэдральный конус.

Проблемы заключаются в визуализации всего этого великолепия. Поэтому, как замечено выше, областью применения данного метода будут задачи с числом переменных равным двум.

Из решения, приведённого на рисунке 2.3, следует, что решение находится в крайней точке, определяемой пересечением прямых линий, соответствующих неравенствам 1 и 2. Поэтому для точного отыскания точки оптимума, достаточно решить систему уравнений вида

$$\begin{cases} 0{,}2k_1 + 0{,}3k_2 = 1{,}8; \\ 0{,}2k_1 + 0{,}1k_2 = 1{,}2. \end{cases}$$

Вычислим главный и вспомогательные определители этой системы:

$$\Delta = \begin{bmatrix} 0{,}2 & 0{,}3 \\ 0{,}2 & 0{,}1 \end{bmatrix} = -0{,}04\,,\ \Delta_1 = \begin{bmatrix} 1{,}8 & 0{,}3 \\ 1{,}2 & 0{,}1 \end{bmatrix} = -0{,}18\,,\ \Delta_2 = \begin{bmatrix} 0{,}2 & 1{,}8 \\ 0{,}2 & 1{,}2 \end{bmatrix} = -0{,}12\,,$$

откуда $k_1 = 4{,}5$, $k_2 = 3$, а значение функции цели после подстановки составляет $f_{opt} = 40{,}5$.

2.2.3. Расширенная (каноническая) форма записи ЗЛП [7, 8, 17, 18]

В общем виде ЗЛП, как мы знаем из предыдущего материала, формулируется следующим образом.

Найти максимум (минимум) целевой функции

$$f(x_1, x_2, ..., x_n) = c_1x_1 + c_2x_2 + ... + c_nx_n = \sum_{j=1}^{n} c_j x_j \qquad (2.1)$$

при заданных условиях (ограничениях)

$$\left.\begin{array}{l} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n \le b_1, \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n \le b_2, \\ ... \\ a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n \le b_{m1} \end{array}\right\} \text{ и} \qquad (2.2)$$

$$x_1 \ge 0, x_2 \ge 0, ..., x_n \ge 0\,. \qquad (2.3)$$

Ограничения (2.3) называются ограничениями ***неотрицательности***, а запись (2.1) – (2.3) называется ***развёрнутой***. Когда в ограничениях (3.2) присутствуют наряду с неравенствами равенства, говорят о ***смешанной***

форме записи, а когда только равенства, то такая форма называется ***канонической***.

Задачу записывают еще в матричной форме

$$\begin{aligned} &Z = C^T X \to \max, \\ &AX \le B = A_0. \end{aligned} \tag{2.4}$$

где обозначено:

- A – матрица ограничений размерностью $[m \times n]$;
- X – вектор переменных (неизвестных), которые называются основными, размерностью $[n \times 1]$;
- B – вектор свободных членов $[m \times 1]$;
- C – вектор коэффициентов линейной формы $[n \times 1]$.

Данная задача может быть записана и в векторной форме:

$$\left.\begin{aligned} &A_1 x_1 + A_2 x_2 + \ldots A_n x_n \le B, \ \textit{где} \\ &A_1 = \begin{bmatrix} a_{11} \\ a_{21} \\ \ldots \\ a_{m1} \end{bmatrix}, A_2 = \begin{bmatrix} a_{12} \\ a_{22} \\ \ldots \\ a_{m2} \end{bmatrix}, \ldots, A_n = \begin{bmatrix} a_{1n} \\ a_{2n} \\ \ldots \\ a_{mn} \end{bmatrix}, A_0 \equiv B = \begin{bmatrix} b_1 \\ b_2 \\ \ldots \\ b_m \end{bmatrix} \end{aligned}\right\}. \tag{2.5}$$

Для решения задачи необходимо привести ограничения задачи от ограничений типа “неравенство” к ограничениям типа “равенство”. С этой целью необходимо ввести ***дополнительные*** переменные (в отличие от основных, входящих изначально в условие) $x_{n+1} \ge 0$, $x_{n+2} \ge 0, \ldots, x_{n+m} \ge 0$, которые превращают неравенства (2.2) в равенства. Эта форма записи, по отношению к исходной математической модели, называется ***расширенной***.

Целевая функция при этом примет вид

$$f(x_1, x_2, \ldots, x_n, x_{n+1}, \ldots, x_{n+m}) = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n + 0 x_{n+1} + \ldots + 0 x_{n+m},$$

а система ограничений, путём введения дополнительных переменных в каждое ограничение, придёт в каноническую форму:

$$\left.\begin{aligned} &a_{11} x_1 + a_{12} x_2 + \ldots + a_{1n} x_n + 1 x_{n+1} + 0 x_{n+2} + \ldots + 0 x_{n+m} = b_1, \\ &a_{21} x_1 + a_{22} x_2 + \ldots + a_{2n} x_n + 0 x_{n+1} + 1 x_{n+2} + \ldots + 0 x_{n+m} = b_2, \\ &\ldots \\ &a_{m1} x_1 + a_{m2} x_2 + \ldots + a_{mn} x_n + 0 x_{n+1} + 0 x_{n+2} + \ldots + 1 x_{n+m} = b_{m1}. \end{aligned}\right\} \tag{2.6}$$

В матричной форме эта система выглядит следующим образом:

$$AX + EX_{ДОП} = B.$$

Решения расширенной и исходной задачи, в области основных переменных, совпадают, как это видно из нижерасположенного рисунка 2.4, приведённого в [33].

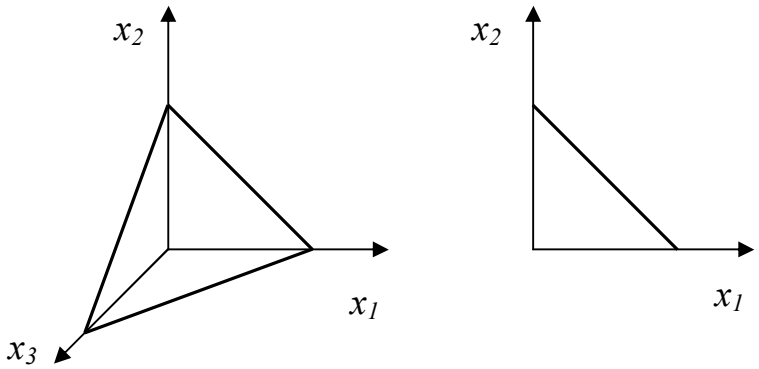


Рисунок 2.4 – Оптимальное решение в трёхмерном и двумерном пространствах

Поэтому, после получения оптимального решения дополнительные переменные можно отбросить. Содержательный смысл значения дополнительной переменной, входящей в оптимальное решение задачи, заключается в том, что это та величина, на которую отличаются левая и правая части соответствующего неравенства.

2.2.4. Определения и теоремы линейного программирования [33]

Допустимые решения – это совокупность чисел $\{x_1, x_2, \ldots x_n\}$, удовлетворяющих ограничениям ***исходной задачи***.

Оптимальное решение – допустимое решение, на котором достигается экстремум (оптимум, максимум или минимум) целевой функции.

Опорный (базисный) план – допустимое решение канонической задачи, в которое входит система линейно независимых векторов A_j, соответствующих переменным x_j.

В качестве напоминания [20]: вектора $\{y_1, y_2, \ldots y_n\}$ линейно независимы, если существуют такие $\alpha, \beta, \ldots, \gamma$, что $\alpha y_1 + \beta y_2 + \ldots + \gamma y_n = 0$ при $\alpha \neq \beta \neq \ldots \neq \gamma \neq 0$.

Каждый вектор содержит m компонент, а ЗЛП не более чем m решений.

Невырожденный опорный план содержит ровно m положительных компонент [80 – 82].

Теорема 1[33]. Если целевая функция принимает экстремальное (максимальное или минимальное) значение в некоторой точке допустимого множества решений $\boldsymbol{R}_1$, то она принимает это значение в крайней точке $\boldsymbol{R}_1$. Если целевая функция принимает экстремальное значение более чем в одной крайней точке, то она принимает это же значение в любой их выпуклой комбинации.

Теорема 2[33]. Если существует такое независимое множество m-мерных векторов $A_1, A_2, \ldots A_k$, $k \le m$, что $A_1 \cdot x_1 + A_2 \cdot x_2 + \ldots + A_k \cdot x_k = A_0$, то n-мерный вектор

$$X^T = [x_1, x_2, \ldots, x_k, \underbrace{0, 0, \ldots, 0}_{\leftarrow n-k \rightarrow}]$$

есть крайняя точка допустимого множества $\boldsymbol{R}_1$.

Теорема 3[33]. Если X_0^T – крайняя точка допустимых решений множества $\boldsymbol{R}_1$, то решение X_0^T – ***допустимое базисное решение*** (ДБР) системы ограничений.

Следствия:

1. При отыскании оптимума достаточно рассмотреть только крайние точки допустимого множества решений.
2. Для отыскания оптимума достаточно перебрать допустимые базисные решения.

На указанные следствия базируются практически все методы решения ЗЛП.

В качестве справочной информации [17, 18, 20, 35].

Линейной комбинацией множества векторов $A_1, A_2, \ldots, A_m$ называется вектор A^*, определяемый выражениями

$$A^* = \sum_{i=1}^{m} \beta_i \cdot A_i; \ \beta_i \ge 0, \ i = \overline{1,m}; \ \sum_{i=1}^{m} \beta_i = 1.$$

В частности, для $m = 2$ линейная комбинация такова: $A^* = \beta_1 A_1 + \beta_2 A_2 \Rightarrow \quad \beta_1 A_1 + (1 - \beta_1) A_2$, а для случая $m = 3$ – $A^* = \beta_1 A_1 + \beta_2 A_2 + \beta_3 A_3$. Ниже представлена графическая интерпретация этих двух случаев.

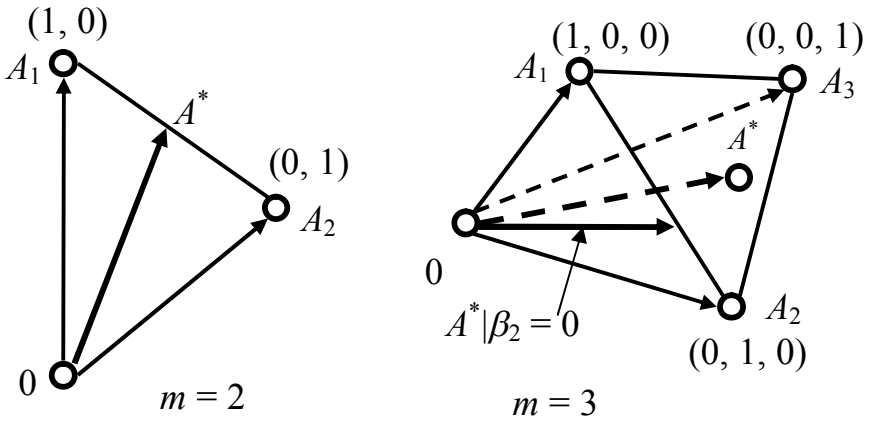


Рисунок 2.5 – Представление выпуклых комбинаций в в двумерном и трёхмерном пространствах

2.2.5 Решение ЗЛП прямым симплекс-методом [29 – 34]

Прямой симплекс-метод называется еще табличным, хотя использование таблиц присуще всем методам этой группы. Позволяет найти решение за конечное, хотя иногда и значительное, число шагов. Значение целевой функции при этом немонотонно возрастают (при решении задач на максимум) или немонотонно убывают (при решении задач на минимум).

Метод применяется, когда все ограничения системы имеют в записи знаки “≤”, то есть, математическая модель выглядит так.

$$
\begin{aligned}
& f(x_1, x_2, ..., x_n) \to opt \\
& \left.\begin{aligned}
& a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n \le b_1, \\
& a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n \le b_2, \\
& ... \\
& a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n \le b_{m1},
\end{aligned}\right\} \\
& x_1 \ge 0, x_2 \ge 0, ..., x_n \ge 0.
\end{aligned}
$$

Алгоритм прямого симплекс-метода

Алгоритм, структурная схема которого приведена на рисунке 2.6, включает следующие шаги: [3, 26, 27, 71, 72].

1. Приведение математической модели задачи к каноническому виду (2.6) и представление её в векторной форме (2.5).

Операция состоит во введении так называемых дополнительных переменных, преобразующих неравенства в равенства. При ограничениях "≤" указанные переменные введутся со знаком плюс. В результате имеем каноническую форму системы ограничений, и расширенную модель.

$$f(x_1, x_2, ..., x_n, x_{n+1}, ..., x_{n+m}) = c_1x_1 + c_2x_2 + ... + c_nx_n + 0x_{n+1} + ... + 0x_{n+m} \to opt,$$

$$\left.\begin{aligned} &a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n + 1x_{n+1} + 0x_{n+2} + ... + 0x_{n+m} = b_1, \\ &a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n + 0x_{n+1} + 1x_{n+2} + ... + 0x_{n+m} = b_2, \\ &... \\ &a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n + 0x_{n+1} + 0x_{n+2} + ... + 1x_{n+m} = b_{m1}. \end{aligned}\right\}$$

Векторы, соответствующие дополнительным переменным канонической системы ограничений $A_{n+1}^T = [1 \ \dots \ 0]$, …, $A_{n+m}^T = [0 \ \dots \ 1]$, образуют начальный базис n-мерного пространства, с их помощью можно разложить любой из векторов, не вошедших в базис.

2. В качестве начального (опорного) решения выбирается крайняя точка, имеющая координаты: $X_0 = [0, 0, ..., 0, b_1, b_2, ..., b_m]$, что означает следующее: $x_1 = 0, x_2 = 0, ..., x_n = 0, x_{n+1} = b_1, x_{n+2} = b_2, ..., x_{n+m} = b_m$, то есть $E \times X_{ДОП} = A_0$.

3. Каноническая форма ЗЛП (2.6) совместно с координатами крайней точки помещается в так называемую симплекс-таблицу, общий вид которой представлен ниже.

		c_j	c_1	...	c_n	0	...	0
Базис	$C_Б$	A_0	A_1	...	A_n	A_{n+1}	...	A_{n+m}
A_{n+1}	*0*	b_1	a_{11}	...	a_{1n}	*1*	...	*0*
A_{n+2}	*0*	b_2	a_{22}	...	a_{2n}	*0*	...	*0*
...	...	...	...	...	...	...	...	...
A_{n+m}	*0*	b_m	a_{m1}	...	a_{mn}	*0*	...	1
	δ	δ_0	δ_1	...	δ_n	δ_{n+1}	...	δ_{n+m}

В последнюю строчку таблицы записываются значения симплекс-разностей, пояснения к вычислениям которых даются в следующем пункте.

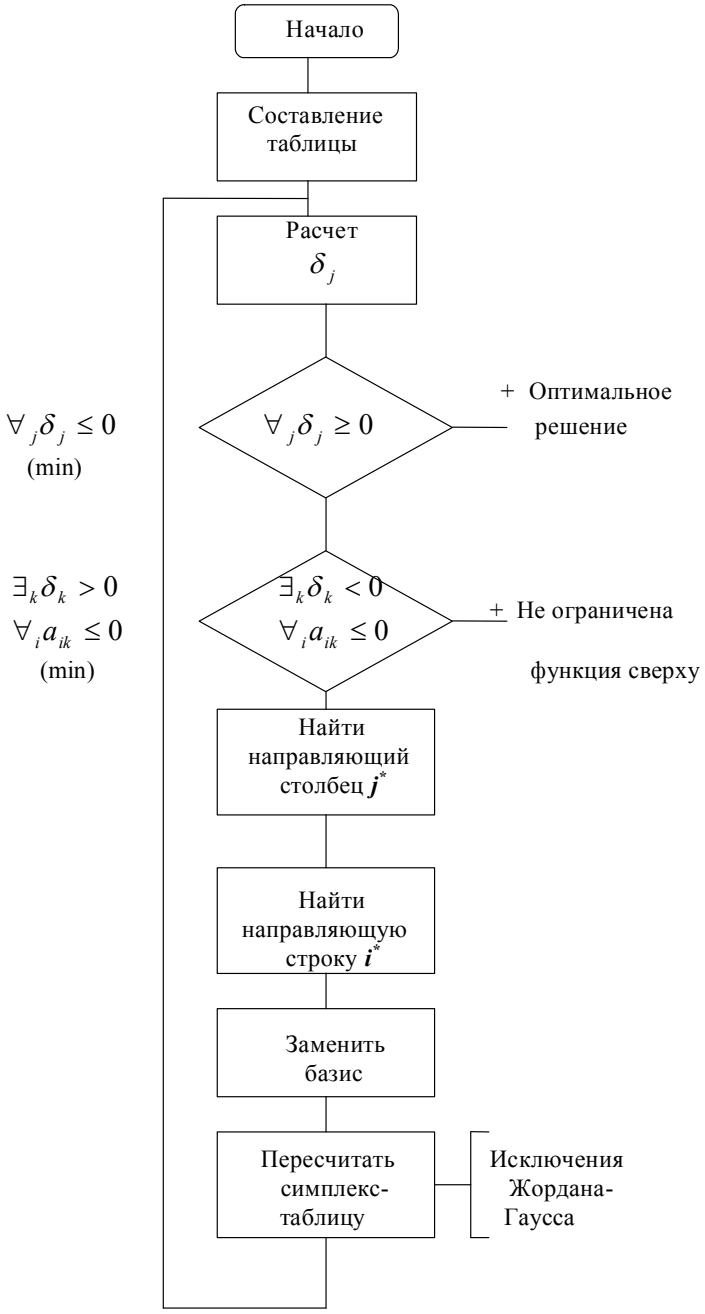


Рисунок 2.6 – Алгоритм симплекс-метода решения задачи максимизации

4. Расчёт симплекс-разностей. Эти величины характеризуют “удачность” текущего базисного плана и рассчитываются по формулам:

- текущее значение целевой функции $\delta_0 = \sum_{i=1}^{m} C_{i,Б} \times a_{i,o}$;
- симплекс-разности $\delta_j = \sum_{i=1}^{m} C_{i,Б} \times a_{i,j} - c_j$.

Последовательность текущих значения целевой функции позволяет контролировать ход расчётов: значения увеличиваются в ходе решения задачи максимизации и уменьшаются при поиске минимума целевой функции.

5. Если все симплекс-разности больше либо равны нулю (при решении задачи на максимум) или неположительные (при решении задачи на минимум), то достигнуто оптимальное решение. Признаки достижения оптимума можно сформулировать так: $\max : \forall_j \delta_j \geq 0$ или $\min : \forall_j \delta_j \leq 0$.

Почему это так, станет понятно при обосновании ввода и вывода базисных векторов ниже.

6. Если существуют столбцы с отрицательными симплекс-разностями, и в соответствующих столбцах ***все*** элементы неположительные то при решении задачи на максимум мы имеем дело с ***неограниченной системой неравенств***. Аналогичная ситуация при решении задачи на минимум, когда существуют столбцы с положительными симплекс-разностями, а в соответствующих столбцах все элементы неположительные. Формально условие выглядит так

$$\max : \exists_k \delta_k < 0 \;\&\; \forall a_{i,k} \leq 0,$$

$$\min : \exists_k \delta_k > 0 \;\&\; \forall a_{i,k} \leq 0.$$

Более точно будет сказать, что область ограничений ***не замкнута в направлении оптимизации***.

В противном случае, имеются отрицательные симплекс-разности, и в столбцах им соответствующих, есть положительные элементы (для решения задачи максимизации) или положительные симплекс-разности и положительные элементы в столбцах (в случае минимизации). В этой ситуации, может быть получено новое решение, лучше, нежели текущее.

Новое допустимое базисное решение буде связано с новым базисом.

7. При решении задачи максимизации выбирается столбец с минимальной симплекс-разностью (минимальной оценкой), который называется ***направляющим***, указывается в таблице вертикальной стрелкой "↑", а в формулах он и его компоненты обозначаются символом "*".

Если задача решается на поиск минимума, то в этом случае выбирается максимальная оценка.

Формально условие выбора записывается так:

$$\text{для задачи на } \max : \arg\min_j \delta_j < 0 \to j^*,$$

$$\text{для задачи на } \min : \arg\max_j \delta_j > 0 \to j^*.$$

Найденный вектор помещается на место вектора, выводимого из базиса. Соответствующая ему переменная включается в состав базисных переменных, эти изменения отображаются в содержимом столбца "Базис".

8. Вектор, выводимый из базиса, определяется путём нахождения ***направляющей*** строки. Независимо от направления проводимой оптимизации (минимизация или максимизация функции цели), направляющая строка определяется по правилу

$$\Theta = \min_i \left\{ \frac{a_{i,0} \geq 0}{a_{i,j^*} \geq 0} \right\} \rightarrow i^*$$

и обозначается символами "←" (в таблице) и "*" (в формулах).

9. После смены векторов строится новая симплекс-таблица, которая получаются модификацией текущей таблицы путем применения исключений Жордана-Гаусса [13, 33, 34, 36].

- Модифицируется направляющая строка путем деления на направляющий элемент a_{i^*,j^*}, стоящий на пересечении направляющей строки и направляющего столбца, а результат записывается на соответствующее место новой таблицы;
- Из всех остальных строк исходной таблицы вычитается поэлементно модифицированная направляющая строка, умноженная на элемент a_{i,j^*}, стоящий на пересечении "уменьшаемой" строки и направляющего столбца, результаты вычитаний записываются на соответствующие места в новой таблице.

10. Работа алгоритма повторяется циклически, начиная с пункта №4.

Замечание. Решать задачу минимизации можно точно так же, как и задачу максимизации, положив $F_{\max} = -1 \times F_{\min}$.

Математическое обоснование этапов ввода и вывода векторов в базис и из базиса

Как известно из теорем линейного программирования, если векторы $A_1, A_2, \ldots, A_m$ являются базисом m-мерного пространства, то

$$A_1 x_1 + A_2 x_2 + \ldots + A_m x_m = A_0, \qquad (2.7)$$

и любой вектор, входящий в каноническую систему ограничений, может быть разложен по векторам этого базиса [18, 20, 35]

$$A_j = \sum_{i=1}^{m} A_j x_{i,j}, \quad j = 1,2,\ldots,n+m \qquad (2.8)$$

или, в развёрнутом виде

$$A_1 x_{1,j} + A_2 x_{2,j} + \ldots + A_m x_{m,j} = A_j . \qquad (2.9)$$

Введём в рассмотрение некоторую величину $\theta > 0$, на которую умножим (2.9):

$$A_1 \theta x_{1,j} + A_2 \theta x_{2,j} + \ldots + A_m \theta x_{m,j} = A_j \theta ,$$

а результат умножения вычтем из (2.7). После приведения подобных членов выражения получим:

$$A_1 (x_1 - \theta x_{1,j}) + A_2 (x_2 - \theta x_{2,j}) + \ldots + A_m (x_m - \theta x_{m,j}) + A_j \theta = A_0 . \qquad (2.10)$$

Вектор X с координатами

$X^{\mathrm{T}} = \{ \quad x_1 - \theta x_{1,j}, \quad x_2 - \theta x_{2,j}, \quad \ldots, \quad x_m - \theta x_{m,j}, \quad \theta, \quad 0, \ldots 0 \quad \}$

$\leftarrow\!\!\!

В результате разложения любого A_j канонической формы представления ЗЛП в виде (2.8) и использования в качестве нового базисного (опорного) решения (2.10), значение целевой функции составит

$$F(A_j) = \sum_{i=1}^{m} c_i x_{i,j} = c_1(x_1 - \theta x_{1,j}) + c_2(x_2 - \theta x_{2,j}) + ... + c_j\theta = \\ = F(X) - \theta(c_1 x_{1,j} + c_2 x_{2,j} + ... - c_j).$$

Таким образом, изменение функции цели при переходе к новому базису есть

$$F(A_j) - F(X) = -\theta(c_1 x_{1,j} + c_2 x_{2,j} + ... - c_j) = -\theta \cdot \delta_j. \qquad (2.11)$$

Причём, величина θ всегда, по определению, положительна, δ_j – есть симплекс-разность базисного вектора A_j. Из (2.11) следует, что при поиске максимума значение δ_j должно быть самое отрицательное (новое значение больше старого), а неотрицательность (2.11) для всех векторов означает достижение оптимума и вызывает, как было отмечено выше, остановку алгоритма.

При решении задачи минимизации новое значение целевой функции должно быть меньше предыдущего, поэтому наблюдаем “зеркальную” ситуацию: δ_j должно быть максимально положительно, а отсутствие положительных симплекс-разностей означает достижения минимума функции цели при заданных ограничениях.

Продемонстрируем работу алгоритма на ранее рассмотренной модели.

$$f = 5k_1 + 6k_2 \to \max,$$
$$\begin{cases} 0{,}2k_1 + 0{,}3k_2 \le 1{,}8; \\ 0{,}2k_1 + 0{,}1k_2 \le 1{,}2; \\ 0{,}3k_1 + 0{,}3k_2 \le 2{,}4; \\ k_1 \ge 0;\ k_2 \ge 0. \end{cases}$$

Расширенная или каноническая форма для этого случая есть

$$f = 5k_1 + 6k_2 + 0k_3 + 0k_4 + 0k_5 \to \max,$$
$$\begin{cases} 0{,}2k_1 + 0{,}3k_2 + 1k_3 + 0k_4 + 0k_5 = 1{,}8; \\ 0{,}2k_1 + 0{,}1k_2 + 0k_3 + 1k_4 + 0k_5 = 1{,}2; \\ 0{,}3k_1 + 0{,}3k_2 + 0k_3 + 0k_4 + 1k_5 = 2{,}4; \\ k_1 \ge 0;\ k_2 \ge 0;\ k_3 \ge 0;\ k_4 \ge 0;\ k_5 \ge 0. \end{cases}$$

Построим таблицу

			c_j	5	6	0	0	0
	Базис	$C_Б$	A_0	A_1	A_2	A_3	A_4	A_5
←	A_3	0	1,8	0,2	0,3	1	0	0
	A_4	0	1,2	0,2	0,1	0	1	0
	A_5	0	2,4	0,3	0,3	0	0	1
		δ_j	0	-5	-6	0	0	0
					↑			

Точка текущего решения (0; 0; 1,8; 1,2; 2,4). Расчёт симплекс-разностей показан ниже.

$$\delta_0 = 0\cdot 1{,}8 + 0\cdot 1{,}2 + 0\cdot 2{,}4 = 0 \quad - \textit{значения целевой функции};$$

$$\delta_1 = 0\cdot 0{,}2 + 0\cdot 0{,}2 + 0\cdot 0{,}3 - 5 = -5;$$

$$\delta_2 = 0\cdot 0{,}3 + 0\cdot 0{,}1 + 0\cdot 0{,}3 - 6 = -6;$$

$$\delta_3 = 0;\ \delta_4 = 0;\ \ \delta_5 = 0.$$

Значения симплекс-разностей свидетельствуют о том, что текущий план необходимо улучшать.

1-я итерация.

Направляющий столбец – 2-й, т.к. самое отрицательное δ_2 = – 6, поэтому в базис вводится вектор A_2, а в решение – переменная k_2. направляющую строку определит минимальное из отношений

$$\Theta = \min_i \left\{ \frac{a_{i,0} \ge 0}{a_{i,j^*} \ge 0} : \frac{1{,}8}{0{,}3} = 6; \frac{1{,}2}{0{,}1} = 12; \frac{2{,}4}{0{,}3} = 8 \right\} \to i^* = 1.$$

Поэтому из базиса выводится вектор A_3 и, соответствующая ему, переменная k_3 из решения, направляющая строка – 1-я. K=(0, k_2, 0, k_4, k_5).

Выполним исключения Жордана-Гаусса, модифицируя таблицу.

Первая строка, согласно алгоритму, делится на направляющий элемент $a_{i^*,j^*} = 0{,}3$ (он выделен серым цветом в таблице), получаем

$$\frac{1{,}8}{0{,}3} = 6,\ \frac{0{,}2}{0{,}3} = \frac{2}{3},\ \frac{0{,}3}{0{,}3} = 1,\ \frac{1}{0{,}3} = \frac{10}{3},\ 0,\ 0.$$

Во второй строке, в месте пересечения с направляющим столбцом $a_{2,j^*} = a_{2,2} = \frac{1}{10}$.

$$a_{20} = \frac{12}{10} - 6\times\frac{1}{10} = \frac{6}{10} = \frac{3}{5};\quad a_{21} = \frac{2}{10} - \frac{2}{3}\times\frac{1}{10} = \frac{2}{15};$$

$$a_{22} = \frac{1}{10} - 1\times\frac{1}{10} = 0;\qquad a_{23} = 0 - \frac{10}{3}\times\frac{1}{10} = -\frac{1}{3};$$

$$a_{24} = 1 - 0 \times \frac{1}{10} = 1;\ a_{25} = 0 - 0 \times \frac{1}{10} = 0.$$

В третьей строке, на пересечении с направляющим столбцом $a_{3,j^*} = a_{3,2} = \frac{3}{10}$.

$$a_{30} = \frac{24}{10} - 6 \times \frac{3}{10} = \frac{6}{10} = \frac{3}{5};\ a_{31} = \frac{3}{10} - \frac{2}{3} \times \frac{3}{10} = \frac{1}{10};$$

$$a_{32} = \frac{3}{10} - 1 \times \frac{3}{10} = 0; \qquad a_{33} = 0 - \frac{10}{3} \times \frac{3}{10} = -1;$$

$$a_{34} = 0 - 0 \times \frac{3}{10} = 0; \qquad a_{35} = 1 - 0 \times \frac{3}{10} = 1.$$

Подставляем результаты расчётов в новую симплекс-таблицу и рассчитываем симплекс-разности.

			c_j	5	6	0	0	0
	Базис	$C_Б$	A_0	A_1	A_2	A_3	A_4	A_5
	A_2	6	6	$\frac{2}{3}$	1	$\frac{10}{3}$	0	0
←	A_4	0	$\frac{3}{5}$	$\frac{2}{15}$	0	$-\frac{1}{3}$	1	0
	A_5	0	$\frac{3}{5}$	$\frac{1}{10}$	0	-1	0	1
		δ_j	36	-1	0	20	0	0
				↑				

$\delta_0 = 6 \cdot 6 + 0 \cdot 0{,}6 + 0 \cdot 0{,}6 = 36,$ - целевая функция возрастает,

$$\delta_1 = \frac{2}{3} \cdot 6 + \frac{4}{30} \cdot 0 + 0 \cdot \frac{1}{10} - 5 = -1,$$

$$\delta_2 = 6 \cdot 1 + 0 + 0 - 6 = 0,$$

$$\delta_3 = 6 \cdot \frac{10}{3} = 20,\ \delta_4 = \delta_5 = 0.$$

Значения симплекс-разностей показывают, что оптимум не достигнут. Текущее решение имеет координаты $K^T = \left[0 \quad 6 \quad 0 \quad \frac{3}{5} \quad \frac{3}{5}\right]$.

2-я итерация.

Направляющий столбец – 1-й, ибо $\delta_2 = -1$, поэтому в базис вводится вектор A_1, в решение – переменную k_1. Определим направляющую строку

$$\Theta = \min_i \left\{ \begin{array}{l} a_{i,0} \ge 0 \\ a_{i,j^*} \ge 0 \end{array} : \frac{\frac{6}{1}}{\frac{2}{3}} = 9; \frac{\frac{6}{10}}{\frac{2}{15}} = \frac{9}{2} = 4{,}5; \frac{\frac{6}{10}}{\frac{1}{10}} = 6 \right\} \to i^* = 2..$$

Из базиса уходит вектор A_4, вводится A_1. Решение будет иметь структуру: K=(k_1, k_2, 0, 0, k_5).

Выполняем преобразование Жордана-Гаусса. Делим направляющую строку на элемент $a_{i^*,j^*} = a_{2,1} = \frac{2}{15}$.

$$a_{20} = \frac{3}{5} : \frac{2}{15} = \frac{9}{2} = 4{,}5; \quad a_{21} = \frac{2}{15} : \frac{2}{15} = 1;$$

$$a_{22} = 0 : \frac{2}{15} = 0; \qquad a_{23} = -\frac{1}{3} : \frac{2}{15} = -\frac{5}{2} = -2{,}5;$$

$$a_{24} = 1 : \frac{2}{15} = \frac{15}{2} = 7{,}5; \quad a_{25} = 0.$$

Преобразуем первую строку таблицы с элементом $a_{1,j^*} = a_{1,1} = \frac{2}{3}$.

$$a_{10} = 6 - \frac{9}{2} \times \frac{2}{3} = 3; \qquad a_{11} = \frac{2}{3} - 1 \times \frac{2}{3} = 0;$$

$$a_{12} = -1; \qquad a_{13} = \frac{10}{3} + \frac{25}{10} \times \frac{2}{3} = 5;$$

$$a_{14} = 0 - \frac{75}{10} \times \frac{2}{3} = -5; \qquad a_{15} = 0.$$

Для третьей строки множитель будет $a_{3,j^*} = a_{3,1} = \frac{1}{10}$.

$a_{30} = 0{,}6 - 4{,}5 \times 0{,}1 = 0{,}15$; $a_{31} = 0{,}1 - 1 \times 0{,}1 = 0$;
$a_{32} = 0 - 0 \times 0{,}1 = 0$; $a_{33} = -1 + 2{,}5 \times 0{,}1 = -0{,}75$;
$a_{34} = 0 - 7{,}5 \times 0{,}1 = -0{,}75$; $a_{35} = -1$.

Строим новую таблицу и рассчитываем значения симплекс-разностей.

		c_j	5	6	0	0	0
Базис	$C_Б$	A_0	A_1	A_2	A_3	A_4	A_5
A_2	6	3	0	1	5	–5	0
A_1	5	4,5	1	0	–2,5	7,5	0
A_5	0	0,15	0	0	–0,75	–0,75	1
	δ_j	40,5	0	0	17,5	7,5	0

$\delta_0 = 6 \times 3 + 5 \times 4{,}5 + 0 \times 0{,}15 = 40{,}5$; $\delta_1 = \delta_2 = \delta_5 = 0$;
$\delta_3 = 6 \times 5 - 5 \times 2{,}5 = 17{,}5$; $\delta_4 = -6 \times 5 + 5 \times 7{,}5 = 7{,}5$.

Все симплекс-разности неотрицательны, поэтому достигнут максимум целевой функции в точке с координатами (4,5; 3; 0; 0; 0,15). Оптимальное значение функции в этой точке равно F_{max} =40,5. Значения основных переменных получены: k_1 = 4,5 и k_2 = 3.

Результат решения совпадает с решением, рассчитанным при демонстрации графического метода.

Следовательно, для обеспечения максимальной прибыли следует закупать картофель в указанных или пропорциональных (3:2) количествах у первого и у второго поставщиков соответственно. Каков же содержательный смысл переменной k_5 = 0,15 в оптимальном решении?

Если подставить оптимальные значения в третье уравнение канонизированной системы, то мы получим равенство. Таким образом, 0,15 – величина, уравновешивающая третье ограничение в точке оптимума. Если бы правая часть указанного ограничения была бы меньше на эту величину, то в точке оптимума пересекались бы сразу три прямые.

2.2.6. Решение ЗЛП методом искусственного базиса [27, 50, 65]

Указанный метод называют ещё методом искусственных переменных. Он предназначен для решения ЗЛП с целевой функцией

$$Z = C^T X \to opt,$$

при наличии ограничениях на переменные вида

$$A \cdot X \otimes B, \qquad (2.12)$$

где обозначено:

- C – вектор коэффициентов целевой функции размерностью [n×1];
- T – символ транспонирования;
- X – вектор искомых параметров математической модели размерностью [n×1];
- opt - вид оптимизации (min или max);
- A - двумерная матрица [m×n] системы линейных ограничений;
- $\otimes$ – знак отношения ($\geq$, =, $\leq$);
- B – вектор правой части ограничений размерностью [m×1].

Достаточно хотя бы ***одного знака отношений*** “$\geq$” или “=” для необходимости использовать этот метод.

Система, в которой присутствуют различные знаки ограничений, называется ***смешанной***.

Пусть все знаки ограничений, для примера, имеют вид "≥". В этом случае, после введения дополнительных переменных и приведения задачи в каноническую форму имеем

$$A \cdot X - E \cdot X_{ДОП} = B \equiv A_0. \qquad (2.13)$$

Если воспользоваться прямым симплекс-методом, то координатами начальной точки решения являются значения дополнительных переменных, удовлетворяющие системе уравнений – $E \cdot X_{ДОП} = A_0$, но при этом ***будет нарушаться условие неотрицательности*** дополнительных переменных (в этом несложно убедиться, решив систему).

Поэтому появляется потребность во введении фиктивных ***искусственных переменных***, не имеющих содержательного смысла, но обеспечивающих существование корректного допустимого базисного решения (ДБР) на начальном шаге, благодаря чему метод и получил своё наименование.

После введения искусственных переменных x_{n+m+1}, x_{n+m+2}, …,$x_{n+2\times m}$ получаем

$$A \cdot X - E \cdot X_{ДОП} + E \cdot X_{ИСК} = A_0, \qquad (2.13)$$

полагая в котором равными нулю основные и дополнительные переменные, придём к ДБР, соответствующему уравнению

$$E \cdot X_{ИСК} = A_0.$$

В случае, когда исходная математическая модель представлена в канонической форме, также вводятся искусственные переменные для обеспечения начального решения.

Когда, наряду с ограничениями "≥" или "=", присутствуют ограничения "≤" на практике используют, для компактности расчётных таблиц, вместо целиком искусственного базиса, ***смешанный базис***, составленный из ортов, соответствующих как искусственным переменным, так и дополнительным переменным, введённым для канонизации неравенств со знаком "≤".

Подведя итоги, скажем, что метод искусственного базиса применяется в следующих случаях [45, 59].

- Все знаки отношения в системе ограничений имеют вид "≥". Имеем чисто искусственный базис.

- Все знаки имеют вид “=”. Также строится чисто искусственный базис.
- Имеется смесь знаков “≥” и “=”. Базис чисто искусственный.
- Имеется смесь знаков “≥”, “=” и “≤”. Базис смешанный.

Для того, чтобы, по мере потери надобности, избавляться от искусственных переменных, которые, как мы помним, не имеют содержательного смысла ни в постановке задачи, ни при её канонизации, в целевую функцию искусственные переменные вводятся с коэффициентами $-\mu$ для задач максимизации и $+\mu$ для решения задач минимизации, где μ – бесконечно большое число.

Обратите внимание, что знак выбран таким, чтобы помешать оптимизации.

Алгоритм метода дадим в отличиях от прямого симплекс-метода для избегания повторения, которое, вопреки расхожему мнению о “матери учения”, таковою быть не может, ибо – среднего рода.

Алгоритм метода искусственных переменных

1. Ограничения исходной математической модели подвергаются канонизации путём введения дополнительных переменных в ограничения со знаками “≥” и “≤”. Система ограничений при этом приобретёт вид (2.12).

2. В те ограничения, которые изначально имели знаки “≥” и “=”, вводятся искусственные переменные. Эти же одновременно вводятся в целевую функцию с бесконечно большими множителями $\pm\mu$, знаки которых определяются направлением оптимизации: $-\mu$ для задач максимизации и $+\mu$ для решения задач минимизации. Ограничения при этом трансформируются в форму (2.13). В развёрнутой форме записи, для случая всех знаков “≥”, получается

$$f(x_1,...,x_n,x_{n+1},...,x_{n+m}) = c_1x_1 + ... + c_nx_n + 0x_{n+1} + ... + 0x_{n+m} \pm \mu x_{n+m+1} ... \pm \mu x_{n+2m},$$

$$\left.\begin{aligned}
&a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n - 1x_{n+1} + 0x_{n+2} + ... + 0x_{n+m} + 1x_{n+m+1} + 0x_{n+m+2} ... + 0x_{n+2m} = b_1,\\
&a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n + 0x_{n+1} - 1x_{n+2} + ... + 0x_{n+m} + 0x_{n+m+1} + 1x_{n+m+2} ... + 0x_{n+2m} = b_2,\\
&...\\
&a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n + 0x_{n+1} + 0x_{n+2} + ... - 1x_{n+m} + 0x_{n+m+1} + 0x_{n+m+2} ... + 1x_{n+2m} = b_{m1}.
\end{aligned}\right\}$$

Векторы-орты A_{n+m+1}, A_{n+m+2}, ..., $A_{n+2\times m}$ – образуют начальный базис в крайней точке

$$X_0^T = \left[\underbrace{0,...,0}_{\leftarrow n\rightarrow},\quad \underbrace{0,...0}_{\leftarrow n\rightarrow},\quad b_1,b_2,...,b_m\right]$$

выпуклого полиэдрального множества.

3. По условиям расширенной задачи производится построение симплекс-таблицы следующего общего вида.

		c_j	c_1	...	c_n	0	...	0	±μ	±μ	...	±μ
Базис	$C_Б$	A_0	A_1	...	A_n	A_{n+1}	...	A_{n+m}	A_{n+m+1}	A_{n+m+2}	...	A_{n+2m}
A_{n+m+1}	±μ,	b_1	a_{11}	...	a_{1n}	-1	...	0	1		...	
A_{n+m+2}	±μ,	b_2	a_{22}	...	a_{2n}	0	...	0	0		...	
...	...	...	...	...	...	...	...	...	...		...	
A_{n+2m}	±μ,	b_m	a_{m1}	...	a_{mn}	0	...	-1	0		...	
	δ	δ_0	δ_1	...	δ_n	δ_{n+1}	...	δ_{n+m}	δ_{n+m+1}	δ_{n+m+2}	...	δ_{n+2m}

4. Задача решается изложенным в предыдущем разделе симплекс-методом, со всеми нюансами.

5. Если искусственная переменная выводится из базиса, то соответствующий ей столбец удаляется из таблицы.

6. В процессе решения необходимо вывести искусственные переменные из базиса. Если строка симплекс-разностей указывает на получение оптимума, а в базисе находятся искусственные переменные, то это означает несовместность системы ограничений. Это дополнительный признак неразрешимости по сравнению с прямым симплекс-методом.

Продемонстрируем работу алгоритма на ранее рассмотренном примере. Математическая модель модернизирована. Заменены: направление оптимизации, знаки в первом и втором неравенстве, а все неравенства умножены на 10 для удобства расчётов.

$$f = 5k_1 + 6k_2 \to \min,$$
$$\begin{cases} 2k_1 + 3k_2 \ge 18; \\ 2k_1 + 1k_2 \ge 12; \\ 3k_1 + 3k_2 \le 24; \\ k_1 \ge 0;\ k_2 \ge 0. \end{cases}$$

Каноническая форма с введёнными искусственными переменными для этого случая есть

$$f = 5k_1 + 6k_2 + 0k_3 + 0k_4 + 0k_5 + \mu k_6 + \mu k_7 + \mu k_8 \to \min,$$
$$\begin{cases} 2k_1 + 3k_2 + 1k_3 + 0k_4 + 0k_5 + 1k_6 + 0k_7 + 0k_8 = 18; \\ 2k_1 + 1k_2 + 0k_3 + 1k_4 + 0k_5 + 0k_6 + 1k_7 + 0k_8 = 12; \\ 3k_1 + 3k_2 + 0k_3 + 0k_4 + 1k_5 + 0k_6 + 0k_7 + 1k_8 = 24; \\ k_1 \ge 0;\ k_2 \ge 0;\ k_3 \ge 0;\ k_4 \ge 0;\ k_5 \ge 0;\ k_6 \ge 0;\ k_7 \ge 0;\ k_8 \ge 0. \end{cases}$$

В записи использован полностью искусственный базис A_6, A_7, A_8, однако, так как вектор A_8 дублирует вектор A_5, следует использовать смешанный базис A_6, A_7, A_5, составленный как из дополнительных, так и из искусственных переменных. Первоначально решение, в последнем случае, находится в точке с координатами (0; 0; 0; 0; 24; 18; 12).

Построим симплекс-таблицу, содержащую векторное представление канонической формы.

		c_j	5	6	0	0	0	μ	μ
Базис	$C_Б$	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7
A_6	μ	18	2	3	–1	0	0	1	0
← A_7	μ	12	2	1	0	–1	0	0	1
A_5	0	24	3	3	0	0	1	0	0
	δ_j	30μ	4μ–5	4μ–6	–μ	–μ	0	0	0
			↑						

Значения симплекс-разностей свидетельствуют о том, что текущий план необходимо улучшать. По поводу выбора направляющего столбца в шутку, хотя в ней немало логики, заметим, что “четыре мешка зерна без пяти зёрнышек” больше, чем “четыре мешка зерна без шести зёрнышек”

1-я итерация даёт:

			c_j	5	6	0	0	0	μ
	Базис	$C_Б$	A_0	A_1	A_2	A_3	A_4	A_5	A_6
←	A_6	μ	6	0	2	–1	1	0	1
	A_1	5	6	1	0,5	0	–0,5	0	0
	A_5	0	6	0	1,5	0	1,5	1	0
		δ_j	6μ+30	0	2μ–3,5	-μ	μ–2,5	0	0
					↑				

После 2-й итерации получим оптимальное решение:

		c_j	5	6	0	0	0
Базис	$C_Б$	A_0	A_1	A_2	A_3	A_4	A_5
A_2	6	3	0	1	-0,5	0,5	0
A_1	5	4,5	1	0	0,25	-0,75	0
A_5	0	1,5	0	0	0,75	2,25	1
	δ_j	40,5	0	0	-1,75	-0,75	0

Достигнут минимум целевой функции в точке с координатами (4,5; 3; 0; 0; 0,15). Это хорошо просматривается на рисунке 2.3, с учётом области

действия двух первых ограничений. Оптимальное значение при этом равно $F_{\max} = 40{,}5$ при основных переменных $k_1 = 4{,}5$ и $k_2 = 3$.

2.2.7. Решение ЗЛП модифицированным симплекс-методом [27, 26]

Указанный метод называется ещё ***методом обратной матрицы***.

Его особенностью [71, 33, 34] является работа только с базисными векторами, поэтому объём расчётом определяется числом базисных векторов, определяемым размером системы ограничений m. По этой причине наибольшая эффективность алгоритма, по сравнению с прямым симплекс-методом или методом искусственного базиса, проявляется, когда n значительно превосходит m. Экономия памяти под промежуточные результаты и сравнительно меньший объём вычислений обусловил преимущественную реализацию этого метода на ЭВМ. Впервые предложен Л.В. Канторовичем.

Для расчётов используются ***две таблицы***. Вспомогательная таблица, содержащая в постоянной части каноническую форму системы ограничений, а в переменной части – заранее не известное число строк симплекс-разностей, пополняемых по мере расчёта при завершении итерации. По необходимости, каноническая форма пополняется искусственными переменными, а таблица – соответствующими им столбцами векторов искусственного базиса.

		c_j	c_1	.....	c_n	0	.....	0
Базис	$C_Б$	A_0	A_1	.....	A_n	A_{n+1}	.....	A_{n+m}
A_{n+1}	0	b_1	a_{11}	...	a_{1n}	1	...	0
A_{n+2}	0	b_2	a_{22}	...	a_{2n}	0	...	0
...	...	...	...	...	...	...	...	...
A_{n+m}	0	b_m	a_{m1}	...	a_{mn}	0	...	1
	δ^0	δ_0^0	δ_1^0	...	δ_n^0	δ_{n+1}^0	...	δ_{n+m}^0
	...							
	δ^r	δ_0^r	δ_1^r	...	δ_n^r	δ_{n+1}^r	δ_0^r	δ_1^r

Основная таблица, в которой производятся расчёты и содержится матрица, обратная матрице, ***составленной из базисных векторов*** системы ограничений ***канонической задачи***, из-за чего метод и получил второе своё название.

			$\longleftarrow A_x^{-1} \longrightarrow$					
Базис	$C_Б$	e_0	e_1	e_2	...	e_m	A^*	Θ
A_{n+1}	*0*	b_1	1	0	…	0		
A_{n+2}	*0*	b_2	0	1	…	0		
...	...	...	...	…	…	...		
A_{n+m}	*0*	b_m	0	0	…	1		
	Λ	λ_0	λ_1	λ_2	...	λ_m		

Графы последней таблицы имеют следующее смысловое наполнение.

A_x^{-1} – обратная матрица, по отношению к базисному фрагменту исходной матрицы ограничений. Первоначально – это единичная матрица.

Столбец e_0 вычисляется по векторной формуле

$$e_0 = A_x^{-1} \times A_0, \tag{2.14}$$

что, на первых порах совпадает с A_0.

Строка оценок Λ определяется формулой

$$\Lambda^T = C_Б^T \times A_x^{-1}, \tag{2.15}$$

а произведение

$$\lambda_0 = C_Б^T \times e_0 \tag{2.16}$$

есть текущее значение целевой функции.

Столбец A^* рассчитывается после выбора направляющего столбца:

$$A^* = A_x^{-1} \times A_{j^*}. \tag{2.17}$$

Столбец оценок Θ служит для определения вектора, выводимого из базиса:

$$\Theta_i = \frac{e_{i,0} \geq 0}{a_i^* > 0}. \tag{2.18}$$

Алгоритм модифицированного симплекс-метода

1. Приведение задачи к канонической форме, введение, по необходимости, искусственных переменных, формирование на их базе основной и вспомогательных таблиц.

2. Расчёт текущего значения целевой функции (2.14), вектора оценок (2.15) и симплекс-разностей . Расчёт последних выполняют по формуле

$$\delta_j = \Lambda^T \times A_j - c_j, \tag{2.19}$$

где A_j – столбец вспомогательной таблицы.

3. Анализ симплекс-разностей на предмет получения оптимального решения. Осуществляется традиционным для прямого симплекс-метода способом. При наличии в базисе искусственных переменных здесь может быть определена несовместность системы ограничений.

4. Выбор направляющего столбца и вектора, вводимого в базис, выполняется на основании симплекс-разностей по известному правилу:

$$\max : \arg\min_j \delta_j < 0 \to j^*, \text{ или } \min : \arg\max_j \delta_j > 0 \to j^*.$$

5. Выполнение пересчёта вектора, вводимого в базис, (вектор A^*) и заполнение соответствующего столбца основной таблицы осуществляется по формуле (2.17).

6. Заполнение столбца Θ основной таблицы по выражению (2.18).

7. Выбор направляющей строки по минимальному значению компонентов столбца Θ. Если направляющую строку определить не удаётся, то необходимо выбрать другой столбец в качестве направляющего, повторив пп. 4 – 7. В случае невозможности выбора остаётся констатировать неразрешимость задачи по причине незамкнутости области ограничений в направлении оптимизации.

8. Преобразование основной таблицы по методу исключений Жордана – Гаусса.

9. После пересчёта таблицы алгоритм продолжает своё выполнение с п.2.

Пример. Используем математическую модель из демонстрационных материалов раздела 2.2.1 вида

$$f = 5x_1 + 6x_2 \to \max,$$
$$\begin{cases} 2x_1 + 3x_2 \le 18; \\ 2x_1 + 1x_2 \le 12; \\ 3x_1 + 3x_2 \le 24; \\ x_1 \ge 0; \\ x_2 \ge 0. \end{cases}$$

Построим каноническую форму и вспомогательную таблицу.

		c_j	5	6	0	0	0
Базис	$C_Б$	A_0	A_1	A_2	A_3	A_4	A_5
A_3	0	18	2	3	1	0	0
A_4	0	12	2	1	0	1	0
A_5	0	24	3	3	0	0	1
	δ^0	0	-5	-6↑	0	0	0
	δ^1	36	-1↑	0	2	0	0
	δ^2	40,5	1	0	2/3	2/3	0

Формируем основную таблицу.

	Базис	$C_Б$	e_0	e_1	e_2	e_3	A^*	Θ
←	A_3	*0*	18	1	0	0	3	6
	A_4	*0*	12	0	1	0	1	12
	A_5	*0*	24	0	0	1	3	8
		Λ	*0*	0	0	0		

$$A_x^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Рассчитываем оценки Λ по формулам (2.11) и (2.12) и заносим в основную таблицу.

$$\Lambda^T = C_Б^T \times A_x^{-1} = [0 \quad 0 \quad 0] \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = [0 \quad 0 \quad 0].$$

Рассчитываем симплекс-разности по формуле (2.15) и помещаем во вспомогательную таблицу.

$$\delta_1 = \Lambda^T \times A_1 - c_1 = [0 \quad 0 \quad 0] \times \begin{bmatrix} 2 \\ 2 \\ 3 \end{bmatrix} - 5 = -5,$$

$$\delta_2 = \Lambda^T \times A_2 - c_2 = [0 \quad 0 \quad 0] \times \begin{bmatrix} 3 \\ 1 \\ 3 \end{bmatrix} - 6 = -6,$$

$$\delta_3 = 0,\ \delta_4 = 0,\ \delta_5 = 0.$$

По значениям симплекс-разностей определяем: оптимум не достигнут, направляющий столбец – 2-й.

1-я итерация.

Пересчитываем столбец, используя (2.17),

$$A^* = A_x^{-1} \times A_{j^*} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 3 \\ 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 3 \end{bmatrix}$$ и помещаем в основную таблицу.

Вычисляем оценки (2.18), тако же помещаем в соответствующий столбец основной таблицы

$$\Theta^T = \left[\frac{e_{i,0} \geq 0}{a_i^* > 0} \right] = \left[\frac{18}{3} \quad \frac{12}{1} \quad \frac{24}{3} \right] = \begin{bmatrix} 6 & 12 & 8 \end{bmatrix}$$

и принимаем решение о выводе из базиса вектора A_3 (обозначен стрелкой), направляющий элемент выделен серым цветом.

Пересчитываем таблицу по методу исключений Жордана-Гаусса.

	Базис	$C_Б$	e_0	e_1	e_2	e_3	A^*	Θ
	A_2	*6*	6	$\frac{1}{3}$	0	0	$\frac{2}{3}$	9
←	A_4	*0*	6	$-\frac{1}{3}$	1	0	$\frac{4}{3}$	$\frac{9}{2}$
	A_5	*0*	6	-1	0	1	1	6
		Λ	*36*	2	0	0		

Рассчитываем оценки Λ по формулам (2.14) и (2.15) и заносим в основную таблицу.

$$\Lambda^T = C_Б^T \times A_x^{-1} = [6 \quad 0 \quad 0] \times \begin{bmatrix} \frac{1}{3} & 0 & 0 \\ -\frac{1}{3} & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} = [2 \quad 0 \quad 0].$$

$$\lambda_0 = C_Б^T \times e_0 = [6 \quad 0 \quad 0] \times \begin{bmatrix} 6 \\ 6 \\ 6 \end{bmatrix} = 36.$$

Рассчитываем симплекс-разности по формуле (2.19) и помещаем во вспомогательную таблицу.

$$\delta_1 = \Lambda^T \times A_1 - c_1 = [2 \quad 0 \quad 0] \times \begin{bmatrix} 2 \\ 2 \\ 3 \end{bmatrix} - 5 = -1,$$

$$\delta_2 = \Lambda^T \times A_2 - c_2 = [2 \quad 0 \quad 0] \times \begin{bmatrix} 3 \\ 1 \\ 3 \end{bmatrix} - 6 = 0,$$

$$\delta_3 = 2,\ \delta_4 = 0,\ \delta_5 = 0.$$

Оптимум не достигнут, направляющий столбец – 2-й.

Пересчитываем столбец, используя (2.17),

$$A^* = A_x^{-1} \times A_{j^*} = \begin{bmatrix} \frac{1}{3} & 0 & 0 \\ -\frac{1}{3} & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 2 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} \frac{2}{3} \\ \frac{4}{3} \\ 1 \end{bmatrix}$$

и помещаем в основную таблицу, затем вычисляем и заполняем столбец оценок Θ. На его основании выводим A_4. Преобразуем таблицу по методу Жордана-Гаусса.

Базис	$C_Б$	e_0	e_1	e_2	e_3	A^*	Θ
A_2	*6*	3	2/3	-1	0		
A_1	*5*	4,5	-0,5	1,5	0		
A_5	*0*	1,5	-0,5	-1,5	1		
	Λ	*40,5*	3/2	3/2	0		

Рассчитываем симплекс-разности и помещаем их во вспомогательную таблицу.

$$\delta_1 = \Lambda^T \times A_1 - c_1 = [\frac{3}{2} \quad \frac{3}{2} \quad 0] \times \begin{bmatrix} 2 \\ 2 \\ 3 \end{bmatrix} - 5 = 1,\ \delta_2 = \Lambda^T \times A_2 - c_2 = [\frac{3}{2} \quad \frac{3}{2} \quad 0] \times \begin{bmatrix} 3 \\ 1 \\ 3 \end{bmatrix} - 6 = 0,$$

$$\delta_3 = 2/3,\ \delta_4 = 2/3,\ \delta_5 = 0.$$

Решение можно считать законченным. Обратите внимание на значение оптимальной дополнительной переменной, соответствующей

вектору A_5. Её значение в десять раз больше значения соответствующей переменной, полученной для исходной модели.

2.2.8. Двойственность в ЗЛП [3, 26, 27, 28, 33, 37, 78]

Рассмотрим ранее упоминавшуюся содержательную задачу о картофельных чипсах, кубиках и дольках, для которой была построена модель

$$\begin{cases} f = 5k_1 + 6k_2 \to \max, \\ 0{,}2k_1 + 0{,}3k_2 \le 1{,}8; \\ 0{,}2k_1 + 0{,}1k_2 \le 1{,}2; \\ 0{,}3k_1 + 0{,}3k_2 \le 2{,}4; \\ k_1 \ge 0; \\ k_2 \ge 0. \end{cases} \qquad (2.20)$$

При построении модели было принято во внимание, что переменные k_1 и k_2 – суть количества картофеля, закупаемого у поставщиков, коэффициенты целевой функции показывают прибыль на единицу продукции из сырья соответствующего поставщика, элементы матрицы $a_{i,j}$ – выход i-ой продукции из сырья j-того поставщика.

Теперь построим наши рассуждения следующим образом [70, 75 – 77]. Пусть y_i, $i = 1, m$ – затраты на вид выпускаемой продукции, её себестоимость. Тогда себестоимость всей продукции есть

$$\sum_{i=1}^{m} b_i \times y_i ,$$

а величина

$$\sum_{i=1}^{m} a_{i,j} \times y_i$$

представляет суммарную стоимость продукции, получаемой из сырья j-того поставщика.

Поэтому мы вправе потребовать снижения себестоимости продукции (минимизацию её), а так как прибыль включаётся (закладывается) в суммарную стоимость продукции, то величина последней превышает прибыль (что определит знак ограничений как “больше или равно”).

Исходя из этих соображений, можно записать

$$g(y_1, y_2, y_3) = 1{,}8y_1 + 1{,}2y_2 + 2{,}4y_3 \to \min,$$
$$\begin{cases} 0{,}2y_1 + 0{,}2y_2 + 0{,}3y_3 \geq 5, \\ 0{,}3y_1 + 0{,}1y_2 + 0{,}3y_3 \geq 6, \\ y_1 \geq 0, y_2 \geq 0, y_3 \geq 0. \end{cases} \qquad (2.21)$$

Задачу (2.21) называют ***двойственной*** по отношению к (2.20), которую называют ***прямой***. Для каждой прямой задачи существует соответствующая двойственная задача. Двойственные переменные не имеют, в общем случае, физического (смыслового) содержания, исключая задачи производственные и экономические, где этим переменным приписывается смысл затрат разнообразной природы.

2.2.9. Формальная связь прямой и двойственной задач

Сопоставим математические модели прямой и двойственной задач.

Прямая задача	Двойственная задача

Развёрнутая форма представления

$\min \sum_{j=1}^{n} c_i x_i,$	$\min \sum_{i=1}^{m} b_i y_i,$
$\sum_{j=1}^{n} a_{i,j} x_i \leq b_i,\ i = \overline{1,m},$	$\sum_{i=1}^{m} a_{i,j} y_i \geq c_j,\ j = \overline{1,n},$
$x_j \geq 0,\ j = \overline{1,n}.$	$y_i \geq 0,\ i = \overline{1,m}.$

Матричная форма представления

$C^T X \to \max,$	$B^T Y \to \min,$
$AX \leq B.$	$A^T Y \geq C.$

Анализируя эти выражения, сформулируем **правила перехода между** этими **задачами**.

- Если прямая задача решается на максимум, то двойственная – на минимум.

- Коэффициенты целевой функции прямой задачи становятся элементами вектора ограничений двойственной задачи.
- Свободные члены в ограничениях прямой задачи становятся коэффициентами целевой функции в двойственной задаче.
- Матрица ограничений двойственной задачи получается транспонированием матрицы ограничений прямой задачи.
- Знаки ограничений в неравенствах заменяются противоположными знаками.
- Число ограничений прямой задачи равно числу переменных двойственной задачи, а число ограничений двойственной задачи равно числу переменных в прямой задаче.

Когда в ограничениях задачи присутствуют только неравенства, пара задач прямой и двойственной и сами задачи называется ***симметричными***. Если *i*-ая переменная не ***ограничена в знаке*** в прямой задаче, то *j*-ое ограничение в двойственной задаче будет ***равенством***.

В некоторых источниках, например в [3, 27, 25], правила перехода сформулированы в виде теорем.

2.2.10. Теоремы двойственности [33]

Теорема 1. Если X_0 и Y_0 суть допустимые решения прямой и двойственной задач соответственно (при этом выполняется неравенство $AX_0 \le B$ и $A^T Y_0 \ge$ C), то значение целевой функции прямой задачи не превышает значения целевой функции двойственной $C^{\mathrm{T}} X_0 \le B^{\mathrm{T}} Y_0$.

Теорема 2 (Основная). Если X_0 и Y_0 суть допустимые решения прямой и двойственной задач и если $C^{\mathrm{T}} X_0 = B^{\mathrm{T}} Y_0$, то X_0 и Y_0 – оптимальные решения пары двойственных задач.

Теорема 3. Если в оптимальном решении прямой задачи *i*-ое ограничение выполняется как строгое неравенство, то оптимальное значение соответствующей двойственной переменной равно нулю, то есть $A_i X^* < b_i \Rightarrow y_i^* = 0$, где A_i – *i*-я строка матрицы.

Теорема 4. Если в оптимальном решении двойственной задачи *j*-ое ограничение выполняется как строгое неравенство, то оптимальное значение соответствующей переменной прямой задачи равно нулю, то есть $A_j^T Y^* - c_j > 0 \Rightarrow x_j^* = 0$, где A_j – *j*-й столбец матрицы.

Из последних теорем просматривается связь оптимальных решений прямой и двойственной задач:

$$\delta_{n+i}^{ПрямойЗадачи} = y_i^*,\ i = \overline{1,m},$$
$$-\delta_{m+j}^{ДвойственнойЗадачи} = x_j^*,\ j = \overline{1,n},$$

где n и m – число переменных и ограничений прямой задачи.

Поэтому оптимальное решение одной из пары двойственных задач позволяет автоматически получить значение другой.

2.2.11. Решение ЗЛП двойственным симплекс-методом [11, 26, 33, 34]

Двойственный симплекс-метод предложен Дж. Данцигом [26] и называется ещё методом последовательного улучшения оценок. Метод базируется на ряде определений.

Сопряжённый базис (базис двойственной задачи) – система m независимых векторов, составленная из матрицы ограничений прямой задачи, базисное решение которой Y удовлетворяет ограничениям двойственной задачи: $A_j^T Y > c_j$.

Псевдоплан прямой задачи – допустимое базисное решение относительно сопряжённого базиса.

Иногда псевдоплан трактуется как разложение векторов прямой задачи, не вошедших в сопряжённый базис, по векторам сопряжённого базиса.

Если среди ***базисных*** компонентов псевдоплана нет отрицательных, то псевдоплан оказывается оптимальным решением прямой задачи, а опорный план – оптимальным решением двойственной задачи.

Алгоритм двойственного симплекс-метода

Подразумевается, что решается задача максимизации функции цели.

1. Необходимо привести систему ограничений в каноническую форму. Искусственные переменные при этом не вводятся.

Перед началом канонизации имеет смысл, путём умножения на –1, добиться одинаковых знаков в ограничениях, и, по необходимости, преобразовать задачу минимизации к задаче максимизации.

2. Выполнить построение двойственной задачи по отношению к канонической форме.

3. Осуществить отыскание базиса сопряжённой задачи (сопряжённый базис).

- Подбор сопряжённого базиса, осуществляется отчасти наугад.

- Нужно выбрать m векторов, руководствуясь определением, данным выше.
- Неравенства двойственной задачи, соответствующие включаемым в базис векторам, преобразуются в систему линейных алгебраических уравнений, результат решения которых подставляется в остальные неравенства, не вошедшие в сопряжённый базис.
- Если неравенства выполняются как истинные, базис подобран правильно, в противном случае, подбор базиса необходимо продолжить.
- Если сопряжённый базис подобрать не удалось, то система ограничений не совместна, и пара задач является неразрешимой.

4. Рассчитать псевдоплан, путём решения ряда систем уравнений вида

$$A_j = M \times \widetilde{A}_j,$$

где A_j – разлагаемый вектор, M – матрица составленная из векторов прямой задачи, образующих сопряжённый базис, $\widetilde{A}_j$ – искомое разложение вектора для всех векторов прямой задачи, не вошедших в базис.

Из практических соображений, в первую очередь рассчитывают A_0.

5. Если в полученном псевдоплане все элементы столбца A_0 неотрицательны, то указанный план является ***оптимальным***, алгоритм завершается нормально. В противном случае, когда $\exists_i a_{i,0} < 0 \ \& \ \forall_j a_{i,j} \geq 0$, имеем дело с неразрешимой задачей, целевая функция которой не ограничена в направлении оптимизации, и алгоритм завершает работу аварийно.

6. В иных случаях самый минимальный отрицательный элемент столбца A_0 определяет ***направляющую строку***: $\arg\min_i a_{i,0} < 0 \Rightarrow i^*$,

а ***направляющий столбец*** определится по правилу

$$\arg\min_j \left\{ \frac{-\delta_j \geq 0}{a_{i^*,j} < 0} \right\} \Rightarrow j^*.$$

7. Далее выполняются исключения Жордана-Гаусса, после чего переходим к п.5.

Замечания.

1. Симплекс-метод перемещается от одного опорного плана к другому, а двойственный симплекс-метод переходит от псевдоплана к псевдоплану.

2. К текущему псевдоплану допускается добавлять новые строки, соответствующие дополнительным ограничениям, усиливающим уже существующие ограничения задачи.

Продемонстрируем работу алгоритма на примере задачи о закупке картофеля.

Этап 1. Приведение математической модели в каноническую форму.

$$f = 5k_1 + 6k_2 + 0k_3 + 0k_4 + 0k_5 \to \max,$$
$$\begin{cases} 0{,}2k_1 + 0{,}3k_2 + 1k_3 + 0k_4 + 0k_5 = 1{,}8; \\ 0{,}2k_1 + 0{,}1k_2 + 0k_3 + 1k_4 + 0k_5 = 1{,}2; \\ 0{,}3k_1 + 0{,}3k_2 + 0k_3 + 0k_4 + 1k_5 = 2{,}4; \\ A_1 \quad A_2 \quad A_3 \quad A_4 \quad A_5 \quad A_0. \end{cases}$$

Этап 2. Построение двойственной задачи.

$$1{,}8y_1 + 1{,}2y_2 + 2{,}4y_3 \to \min,$$
$$\begin{cases} 0{,}2y_1 + 0{,}2y_2 + 0{,}3y_3 \geq 5, & A_1 \\ 0{,}3y_1 + 0{,}1y_2 + 0{,}3y_3 \geq 6, & A_2 \\ y_1 \geq 0, & A_3 \\ y_2 \geq 0, & A_4 \\ y_3 \geq 0. & A_5 \end{cases}.$$

Её целевая функция нам безразлична, в дальнейших расчётах она не используется.

Этап 3. Подбор сопряжённого базиса.

Постараемся включить в него один из векторов, соответствующий основной переменной. Ориентировочно выберем A_1, A_2 и A_3. Решим систему уравнений, составленную из соответствующих строк системы ограничений двойственной задачи:

$$\begin{cases} 0{,}2y_1 + 0{,}2y_2 + 0{,}3y_3 = 5, \\ y_1 = 0, \\ y_3 = 0. \end{cases}$$

В результате нами получено $y_2 = 25$.

Проверка на остальных ограничениях двойственной задачи показывает, что ограничение $A_4 : y_2 \geq 0$ – выполняется, а $A_2 : 0{,}3y_1 + 0{,}1y_2 + 0{,}3y_3 = 2{,}5 \geq 6$ – нет.

Поэтому базис A_1, A_2 и A_3 не подходит в качестве сопряжённого.

Попробуем базис A_2, A_3 и A_5. Соответствующая система уравнений выглядит следующим образом:

$$\begin{cases} 0{,}3y_1 + 0{,}1y_2 + 0{,}3y_3 = 6, & A_2 \\ y_1 = 0, & A_3 \\ y_3 = 0. & A_5 \end{cases}$$ её решение даёт $y_2 = 20$.

Легко показать, что выполняются неравенства A_1 и A_4. Поэтому A_2, A_3 и A_5 является сопряжённым базисом.

Этап 4. Найдём псевдоплан для этого базиса. Для этого нам необходимо решить несколько систем уравнений.

$$\left.\begin{aligned} A_0 &= A_2X_{20} + A_3X_{30} + A_5X_{50}, \\ A_1 &= A_2X_{21} + A_3X_{31} + A_5X_{51}, \\ A_4 &= A_2X_{24} + A_3X_{34} + A_5X_{54}.. \end{aligned}\right\}$$ – индексы суть координаты в таблице

псевдоплана, первый индекс указывает на привязку к базисному вектору, второй – к разлагаемому. Имеем:

$$\begin{bmatrix} 1{,}8 \\ 1{,}2 \\ 2{,}4 \end{bmatrix} = \begin{bmatrix} 0{,}3 & 1 & 0 \\ 0{,}1 & 0 & 0 \\ 0{,}3 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{20} \\ X_{30} \\ X_{50} \end{bmatrix}; \begin{bmatrix} 0{,}2 \\ 0{,}2 \\ 0{,}3 \end{bmatrix} = \begin{bmatrix} 0{,}3 & 1 & 0 \\ 0{,}1 & 0 & 0 \\ 0{,}3 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{21} \\ X_{31} \\ X_{51} \end{bmatrix};$$

$$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0{,}3 & 1 & 0 \\ 0{,}1 & 0 & 0 \\ 0{,}3 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{24} \\ X_{34} \\ X_{54} \end{bmatrix}.$$

Для первой системы $1{,}2 = 0{,}1 \times X_{20} \Rightarrow X_{20} = 12$, остальные переменные находятся путём последовательной подстановки в первое и третье уравнения. Окончательно рассчитаем:

$$\widetilde{A}_0 = \begin{bmatrix} 12 \\ -1{,}8 \\ -0{,}6 \end{bmatrix}; \qquad \widetilde{A}_1 = \begin{bmatrix} 2 \\ -0{,}4 \\ -0{,}3 \end{bmatrix}; \qquad \widetilde{A}_4 = \begin{bmatrix} 10 \\ -3 \\ -5 \end{bmatrix}.$$

Из базисных векторов и векторов разложения формируем симплекс-таблицу и подсчитываем симплекс-разности.

			c_j	5	6	0	0	0
	Базис	$C_Б$	A_0	A_1	A_2	A_3	A_4	A_5
	A_2	6	12	2	1	0	10	0
←	A_3	0	–1,8	–0,4	0	1	–3	0
	A_5	0	–0,6	–0,3	0	0	–5	1
		δ_j	72	7	0	0	60	0
				↑				

Направляющая строка определяется самым отрицательным элементом столбца A_0. направляющий столбец определится как $\min\left\{\frac{-7}{-0,4};\ \frac{-60}{-3}\right\}$.

Пересчитываем таблицу и проводим расчёт симплекс-разностей.

		c_j	5	6	0	0	0
Базис	$C_Б$	A_0	A_1	A_2	A_3	A_4	A_5
A_2	6	3	0	1	5	–5	0
A_1	5	4,5	1	0	–2,5	7.5	0
A_5	0	0,15	0	0	–0,75	–0,75	1
	δ_j	40,5	0	0	17.5	7.5	0

Получено оптимальное решение, которое совпадает с полученными нами ранее.

2.2.12. Вопросы для самоконтроля

1. Какие модели являются предметом исследования в линейном программировании?
2. Что утверждают основные теоремы линейного программирования?
3. Каковы условия применения графического метода?
4. Каковы особые случаи, возникающие при решении ЗЛП?
5. Почему при решении ЗЛП графическим методом используют именно перпендикуляр к нормали, а не линию под каким-либо другим углом?
6. Поясните, с чем связан выбор направления движения перпендикуляра к нормали?
7. Что такое каноническая форма ЗЛП?
8. Какое функциональное назначение отводится дополнительным переменным?
9. В чём состоят признаки (условия) неразрешимости задачи при решении её симплекс-методом?
10. Чем обосновано правило выбора вектора, вводимого в базис?
11. Каков смысл симплекс – разности?
12. Чем обоснован выбора выводимого из базиса вектора?
13. Какова последовательность работы алгоритма Жордана-Гаусса?
14. Как проконтролировать правильность хода решения задачи по значениям симплекс – разностей?

15. Чем обосновано требование положительности к вектору свободных членов системы ограничений?

16. В чём состоит связь обычной и канонической форм задач ЛП?

17. Что значит присутствие в столбце оптимального решения ненулевых значений дополнительных переменных?

18. Для чего требуются искусственные переменные?

19. В чём сходство и различие дополнительных и искусственных переменных?

20. От чего зависят знаки и множители при искусственных переменных?

21. Как по последовательности значений целевой функции определить правильность хода решения задачи?

22. Как определить, что задача имеет несовместные ограничения?

23. Какие случаи неразрешимости ЗЛП отображаются в симплекс-таблице?

24. Почему, при наличии ограничений больше или равно (“≥”), нельзя обойтись базисом соответствующим, дополнительным переменным?

25. В чем привлекательность машинной реализации модифицированного симплекс-метода?

26. Почему модифицированный симплекс-метод наиболее эффективен в случаях, когда число переменных n превышает число ограничений m?

27. Почему модифицированный симплекс-метод ещё называют методом обратной матрицы?

28. Где располагается обратная матрица в симплекс-таблице, и по отношению к какой матрице она является обратной?

29. Что утверждают теоремы двойственности.

30. Как связаны прямая и двойственная задачи?

31. Как по оптимальному решению прямой задачи получить оптимальное решение двойственной?

32. Что такое псевдоплан?

33. Что такое сопряженный базис?

34. Как узнать при решении двойственным симплекс-методом, что ограничения, заданные в математической модели, несовместны?

35. В чём проявляются особенности алгоритма двойственного симплекс-метода при определении вводимого и выводимого векторов?

36. Какой вид имеет симплекс-таблица двойственного метода в случае неразрешимости задачи?

37. Как соотносятся целевые функции прямой и двойственной задач в ходе решения и в оптимальном решении?

38. В каких случаях основные переменные двойственной задачи имеют содержательный смысл, и какой именно?

2.3. Дискретное программирование

В общем случае задача дискретного программирования имеет следующую постановку [2, 3, 10, 41, 63].

Найти максимум (минимум) целевой функции $f(x_1, x_2, ..., x_n)$ при заданных условиях

$$\left.\begin{aligned} & g_1(x_1, x_2, ..., x_n) \le b_1, \\ & g_2(x_1, x_2, ..., x_n) \le b_2, \\ & ... \\ & g_m(x_1, x_2, ..., x_n) \le b_m, \\ & X = \{x_1, x_2, ..., x_n\} \subseteq D, \end{aligned}\right\} \qquad (2.22)$$

где D – конечное или счётное множество.

В этом случае говорят о ***дискретном программировании***. Если X ограничено множеством целых чисел, то задачу назначают задачей ***линейного целочисленного программирования (ЛЦП)***.

Известны следующие классы задач дискретного программирования.

- Задачи с неделимостями (задачи о рюкзаке), обусловлены физическими свойствами объектов. Это задача о размещении массивов информации на внешних устройствах ЭВМ при ограничениях на объём, скорость вращения, стоимостные рамки и др.
- Экстремальные комбинаторные задачи (выбора, о назначениях, коммивояжёра, о покрытиях).
- Задачи на несвязных и невыпуклых областях.
- Задачи с разрывной целевой функцией.
- Транспортная задача. Когда задано ограничение целочисленности, то есть при целых значениях массивов поставок, потребления и стоимостей или ограничениях на пропускные способности коммуникаций, является задачей линейного целочисленного программирования.

Для решения задач дискретного программирования применяются как строго обоснованные, так и неформальные метода поиска решений.

1. Метод отсечений, отсекающих плоскостей, он же метод ГОмори (или ГомОри).
2. Метод ветвей и границ.
3. Методы, учитывающие особенности задачи.
4. Методы случайного поиска (эвристические).

2.3.1. Решение задачи ЛЦП методом Гомори

Данный метод носит название метода ***отсекающих плоскостей*** или метода ***целочисленных форм***, но чаще именуется по имени автора. В основании метода положены следующие теоретические положения [8, 10].

Любое уравнение или неравенство линейной системы ограничений представимо линейной комбинацией базисных векторов и в канонической форме записывается так:

$$\sum_{j=1}^{n} d_{ij} \cdot x_j = p_i,\ i = \overline{1,m}. \tag{2.23}$$

Обозначим заключением в квадратные скобки $[d]$ целую часть d:

$[2{,}5] = 2$; $[10/3] = 3$; $[4] = 4$; $[-2{,}5] = -3$; $[-10/3] = -4$; $[-4] = -4$.

По сути дела, операция […] представляет собой округление по недостатку (в меньшую сторону).

Если считать переменные $X = (x_1,\ x_2,\ \ldots\ ,\ x_n)$ целыми числами, то можно перейти к более слабому, по сравнению с (2.23), выражению

$$\sum_{j=1}^{n} \left[d_{ij}\right] \cdot x_j \le p_i,\ i = \overline{1,m}, \tag{2.24}$$

а учитывая, что сумма в (2.20) целочисленна, то справедливо и

$$\sum_{j=1}^{n} \left[d_{i,j}\right] \cdot x_j \le [p_i],\ i = \overline{1,m}. \tag{2.25}$$

Введя свободную целочисленную переменную x_{n+t}, канонизируем (2.25):

$$\sum_{j=1}^{n} \left[d_{ij}\right] \cdot x_j + x_{n+t} = [p],\ i = \overline{1,m}. \tag{2.26}$$

Очевидно, что добавление последнего равенства к исходной канонизированной системе ***не противоречит*** исходной системе ограничений.

Чтобы получить (2.26) из (2.23), необходимо «отсечь» от (2.26) дробную часть. С этой целью формируется ***отсечение*** для приведения произвольного уравнения в целочисленную форму. Указанное отсечение представляет собою уравнение, которое, будучи прибавленным к исходному уравнению, делает его целочисленным:

$$\sum_{j=1}^{n}\{-d_{ij}\}\cdot x_j + x_{n+t} = \{-p\},\ i=\overline{1,m}. \qquad (2.27)$$

В (2.27) обозначена символами {…} операция нахождения дробной части. Отсюда просматривается простой вычислительный алгоритм.

Алгоритм формирования отсечения Гомори

1. Для выбранного канонизированного уравнения (2.23) сформировать желаемую целочисленную форму вида (2.26).

2. Из целочисленной формы (2.26) вычесть исходное уравнение (2.23), получится уравнение отсекающей плоскости (2.27).

Дадим и нотацию этого алгоритма в виде формулы:

<Отсечение> = <Целочисленная_форма> – <Исходная_строка>. (2.28)

Алгоритм решения задачи ЛЦП методом Гомори

Процедура получения решения структурно состоит из

- предварительного этапа,
- проверки условия окончания и
- так называемой “большой итерации”, которая включает операцию формирования отсечения и несколько шагов итеративной части двойственного симплекс-метода.

1. ***Предварительный этап***. В ходе его получают оптимальное решение ЗЛП без учёта целочисленности. Решение выполняется любым удобным методом, кроме, разумеется, графического метода.

2. ***Условие окончания*** расчётов. Если в текущем решении в***се компоненты базисного столбца*** A_0, соответствующие основным переменным, ***являются целыми*** числами, то найдено оптимальное решение задачи ЛЦП. В противном случае, выполняется большая итерация.

3. ***Большая итерация***.

3.1. ***Отсечение Гомори*** формируется

- ***для тех строк*** симплекс-таблицы, ***в которых*** компоненты $a_{i,0}$, соответствующие основным переменным задачи, ***дробные числа***;
- ***на*** каждом ***шаге*** алгоритма отсечение Гомори формируется только ***по одной*** из строк;
- ***очерёдность*** формирования отсечений ***не регламентируется***.

3.2. В базисное решение вводится дополнительная переменная x_{r+t}, соответствующая канонизированному уравнению отсекающей

плоскости, одновременно симплекс-таблица пополняется строкой и столбцом-ортом A_{n+t}.

3.3. Выполняется итерационная часть двойственного симплекс-метода. Направляющая строка соответствует вектору A_{n+t}, а направляющий столбец определяется по обычному условию этого метода:

$$\arg\min_{j}\left\{\frac{-\delta_j \geq 0}{a_{i^*,j} < 0}\right\} \Rightarrow j^*.$$

3.4. Если вектор A_{n+t}, ранее выведенный из базиса, в ходе расчётом снова в него вводится в процессе итераций, то строку и столбец симплекс-таблицы, соответствующие переменной x_{n+t} после пересчёта по методу Жордана-Гаусса вычёркивают (удаляют) из неё.

На этом циклическая часть алгоритма завершена, а цикл возобновляется с п. 2.

Замечания к методу Гомори [10].

1. Сходимость вычислений обеспечивается за конечное число итераций, что и обусловливает применение данного метода на практике

2. Метод особенно эффективен, когда ***большинство*** переменных в оптимальном нецелочисленном решении имеют целочисленные значения.

4. После выполнения нескольких больших итераций на шаге отсечения Гомори появляются многочисленные альтернативы. Это ведёт к зацикливанию, именуемому Г. Вагнером [10] “сплошной вырожденностью”, когда решение возвращается к ранее бытовавшей позиции, вследствие неверного выбора строки для формирования отсечения. Вагнеру известны многочисленные примеры, когда при значениях n и m, не превышающих десяти, потребовались тысячи итераций, прежде чем оптимум был достигнут.

5. Затруднена сходимость при решении задач, в которых значения элементов $a_{i,j}$ и b_i велики.

6. Иногда, для достижения успеха, требуется видоизменить постановку задачи в сторону усиления, например, введя ограничения $x_1 \leq 6$ и $x_2 \leq 6$ в дополнение к уже существующему ограничению $x_1 + x_2 \leq 6$.

Рассмотрим пример применения метода Гомори.

Поскольку операции с симплекс-таблицами нами ранее вельми подробно рассматривались, ограничим содержание нашего примера процессом формирования отсечений Гомори по результатам решения демонстрационной задачи о производстве изделий из картошки.

Оптимальное решение этой задачи без учёта ограничения целочисленности имеет вид:

		c_j	5	6	0	0	0
Базис	$C_Б$	A_0	A_1	A_2	A_3	A_4	A_5
A_2	6	3	0	1	5	–5	0
A_1	5	4,5	1	0	–2,5	7,5	0
A_5	0	0,15	0	0	–0,75	–0,75	1
	δ_j	40,5	0	0	17,5	7,5	0

Видно, что основная переменная x_1 нецелая, поэтому необходимы отсечения.

Сформируем отсечение Гомори по второй строке, которая соответствует основной переменной x_1. Исходной строке соответствует уравнение

$$4{,}5 = 1x_1 + 0x_2 - 2{,}5x_3 + 7{,}5x_4 + 0x_5 .$$

Целочисленная форма для этой строки есть

$$4 = 1x_1 + 0x_2 - 3x_3 + 7x_4 + 0x_5 .$$

Поэтому отсечение будет

$$\begin{array}{r} 4 = 1x_1 + 0x_2 - 3x_3 + 7x_4 + 0x_5 \\ - \\ 4{,}5 = 1x_1 + 0x_2 - 2{,}5x_3 + 7{,}5x_4 + 0x_5 \\ \hline -0{,}5 = 0x_1 + 0x_2 - 0{,}5x_3 - 0{,}5x_4 + 0x_5 \end{array} .$$

Результат вычислений занесём в симплекс-таблицу в отдельную строку. Одновременно таблица пополнится дополнительным столбцом для вектора A_6 соответствующим переменной x_6.

			c_j	5	6	0	0	0	0
	Базис	$C_Б$	A_0	A_1	A_2	A_3	A_4	A_5	A_6
	A_2	6	3	0	1	5	–5	0	0
	A_1	5	4,5	1	0	–2,5	7,5	0	0
	A_5	0	0,15	0	0	–0,75	–0,75	1	0
←	A_6	0	–0,5	0	0	–0,5	–0,5	0	1
		δ_j	40,5	0	0	17,5	7,5	0	0
							↑		

Далее выполняются несколько итерационных шагов двойственного симплекс-метода:

- выводимая строка определяется отрицательной компонентой столбца A_0;
- вектор, вводимый в базис определяет условие $\min\left\{\frac{-17,5}{-0,5};\frac{-7,5}{-0,5}\right\}$, это A_4;
- осуществляется пересчёт симплек-таблицы по методу Жордана-Гаусса до получения условия окончания итераций – положительности компонентов A_0.

2.3.2. Решение задачи ЛЦП методом ветвей и границ [7, 41, 50, 63]

Этот метод применяется для решения как полностью целочисленных, так и частично целочисленных задач дискретного программирования.

Пусть математическая модель имеет вид

$$\begin{aligned} &C^T X \to \max, \\ &AX \le B. \end{aligned} \tag{2.29}$$

Компоненты вектора X положительны и целочисленны. Допустим, что исходная задача линейного программирования имеет решение. В этом случае область ограничений замкнута.

Тогда каждая переменная x_j и в допустимом решении, и оптимуме ограничена диапазоном

$$L_j \le x_j \le U_j, \tag{2.30}$$

где L_j – нижний предел, а U_j – верхний предел (граница), которые определяются границами области допустимых решений задачи. Это следует из самого факта наличия непротиворечивых ограничений, образующих замкнутую область

Пусть I есть целое число, такое, что $L_j \le I \le U_j - 1$. Тогда оптимальное ***целочисленное*** значение x_j, удовлетворяющее решению (2.29) и лежащее в пределах (2.30), будет находиться либо между L_j и I, либо между $I + 1$ и U_j. Это приводит к тому, что возникают дополнительные условия, по отношению к исходным условиям (2.29), не противоречащие им:

$$\left.\begin{array}{l} x_j \le I, \\ x_j \ge I+1 \end{array}\right\}. \qquad (2.31)$$

На базе ограничений (2.31) основана систематическая схема применения метода.

Ограничения, приписываемые к исходной задаче, есть ***дополнительные границы***, благодаря чему мы имеем, на каждом шаге постановку ***пары задач*** на базе одной нецелочисленной.

Интерпретация хода решения в виде дерева определило второе название метода – ***ветвей.***

Алгоритм метода ветвей и границ

Композиционно алгоритм состоит из предварительного этапа, проверки условия целочисленности текущего решения, построения задач G_{i1} и G_{i2}, большой итерации, которая представляет собой несколько итерационных шагов двойственного симплекс-метода, и заключительной части, на которой выбирается наилучшее из всех, ранее полученных, целочисленных решений. Цифровой код i в индексации задач соответствует положению текущей "родительской" задачи на дереве решений

1. Предварительный этап. Задача (2.29) решается любым удобным методом до отыскания нецелочисленного оптимального решения, соответствующего точке X_0.

2. Этой точке X_0 ставится в соответствие решение G_0 и его оценка – текущее значение целевой функции $\xi = C^T \times X_0$. Если X_0 – целочисленное решение для основных переменных математической модели, то задача считается решённой.

В противном случае, если X_0 – нецелочисленное решение, то, используя систему неравенств (2.31), получаем множество из двух задач G_{01} и G_{02} (ветвей). Особо подчеркнём, что пара задач ***возникает для одной нецелочисленной переменной одновременно***. Каждая задача решается в отдельности, при этом находят их оценки $\xi(G_{01})$ и $\xi(G_{02})$.

3. В ходе решения на k–й итерации, в зависимости от текущих оценок $\xi(G_{i1})$ и $\xi(G_{i2})$, может произойти дальнейшее ветвление задач.

4. Вычислительный процесс осуществляется до "перерешивания" всех возникающих задач или до получения признаков их неразрешимость. Из полученных решений выбирается то, которое является наилучшим (в смысле оптимума) решением исходной задачи (2.29).

5. Для решения возникающих задач (2.31) используют двойственный симплекс-метод, который, как нам известно, допускает ввод новых ограничений в псевдоплан по ходу решения.

6. Ограничения вводятся только для одной из основных базисных нецелочисленных переменых. Правила формирования ограничений по неравенствам (2.31) суть следующие. Пусть в базисе находится вектор A_j, соответствующая переменная которого x_j – дробное число.

- Задача G_{i1} формируется по ограничению $x_j \le I$, где I – целая часть $[x_j]$, округленная по недостатку. Первоначально формируется ограничение A_{n+t}, которое соответствует канонической форме неравенства и представляется в виде уравнения $I = x_j + x_{n+t}$. В симплекс-таблицу помещается строка, которая получается в результате операции вычитания $A_{n+t} — A_j$.
- Задача G_{i2} формируется по ограничению $- x_j \le - (I + 1)$, которой соответствует каноническая форма $- (I + 1) = - x_j + x_{n+t}$. В симплекс таблицу помещается строка, равная сумме $A_{n+t} + A_j$.

Каждая из исходных симплекс-таблиц задач G_{i1} и G_{i2}, дополняется строкой симплекс-разностей, взятой из таблицы, содержащей нецелочисленное решение G_i.

Продемонстрируем работу алгоритма на известном примере.

Оптимальное решение без учёта целочисленности есть

		c_j	5	6	0	0	0
Базис	$C_Б$	A_0	A_1	A_2	A_3	A_4	A_5
A_2	6	3	0	1	5	–5	0
A_1	5	4,5	1	0	–2,5	7,5	0
A_5	0	0,15	0	0	–0,75	–0,75	1
	δ_j	40,5	0	0	17,5	7,5	0

а его оценка $G_0[40,5] = 40$. Обе задачи формируются для переменной x_1 по 2-й строке таблицы A_2.

Задача G_{01}.

$x_1 \le 4 \Rightarrow A_6$: $4 = x_1 + x_6$, $\widetilde{A}_6 : A_6 - A_1$.

$$
\begin{array}{lrcrcrcrcrcr}
A_6 & 4 & = & 1x_1 & + & 0x_2 & - & 0x_3 & + & 0x_4 & + & 0x_5 & + & 1x_6 \\
- & & & & & & & & & & & & & \\
A_1 & 4{,}5 & = & 1x_1 & + & 0x_2 & - & 2{,}5x_3 & + & 7{,}5x_4 & + & 0x_5 & & 0x_6 \\
\hline
\widetilde{A}_6 & -0{,}5 & = & 0x_1 & + & 0x_2 & + & 2{,}5x_3 & - & 7{,}5x_4 & + & 0x_5 & + & 1x_6
\end{array}
$$

Задача G_{02}.

$-x_1 \le -5 \Rightarrow A_6' : -5 = -x_1 + x_6'$, $\widetilde{A}_6' : A_6' + A_1$

$$
\begin{array}{lrcrcrcrcrcr}
A_6' & -5 & = & -1x_1 & + & 0x_2 & - & 0x_3 & + & 0x_4 & + & 0x_5 & + & 1x_6' \\
- & & & & & & & & & & & & & \\
\hline
A_1 & 4{,}5 & = & 1x_1 & + & 0x_2 & - & 2{,}5x_3 & + & 7{,}5x_4 & + & 0x_5 & & 0x_6 \\
\hline
\widetilde{A}_6' & -0{,}5 & = & 0x_1 & + & 0x_2 & - & 2{,}5x_3 & + & 7{,}5x_4 & + & 0x_5 & + & 1x_6'
\end{array}
$$

Сформируем симплекс-таблицы для обеих задач.

Задача G_{01}.

			c_j	5	6	0	0	0	0
	Базис	$C_Б$	A_0	A_1	A_2	A_3	A_4	A_5	A_6
	A_2	6	3	0	1	5	–5	0	0
	A_1	5	4,5	1	0	–2,5	7,5	0	0
	A_5	0	0,15	0	0	–0,75	–0,75	1	0
←	$\widetilde{A}_6$	0	-0,5	0	0	2,5	–0,75	0	1
		δ_j	40,5	0	0	17,5	7,5	0	0
							↑		

Задача G_{02}.

			c_j	5	6	0	0	0	0
	Базис	$C_Б$	A_0	A_1	A_2	A_3	A_4	A_5	A_6'
	A_2	6	3	0	1	5	–5	0	0
	A_1	5	4,5	1	0	–2,5	7,5	0	0
	A_5	0	0,15	0	0	–0,75	–0,75	1	0
←	$\widetilde{A}_6'$	0	-0,5	0	0	–2,5	–0,75	0	1
		δ_j	40,5	0	0	17,5	7,5	0	0
						↑			

Вид дерева решений показан ниже, на рисунке 2.7.

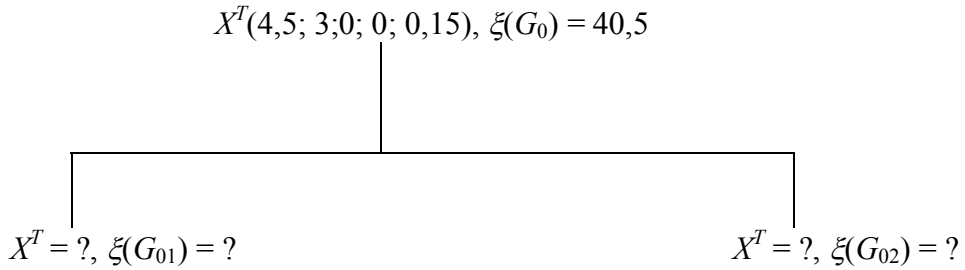


Рисунок 2.7 – Первоначальное дерево решений

Решение опускаем как освоенное нами в предыдущих разделах.

Метод ещё называют ***методом обрыва ветвей*** или ***методом возврата***: всё зависит от способа перемещения по дереву задач. Существует множество алгоритмов метода, адаптированных под разнообразные частные условия содержательной задачи.

2.3.3. Вопросы для самоконтроля

1. В чем сходства и различия терминов “дискретный” и “целочисленный”?
2. Как построить отсекающую плоскость Гомори?
3. Почему алгоритм ветвей и границ получил такое название, что является ветвями, а что — границами?
4. В чём идея сущность и неравенств (2.31)?
5. Почему в ходе решения ЛЦП используется двойственный симплекс-метод?
6. В каких случаях задача ЛЦП не будет иметь решения?
7. Как вы думаете, оптимальное решение ЛЦП будет единственным? Обоснуйте свои соображения по этому поводу.

2.3.4. Транспортные задачи (ТЗ) [7, 8, 22, 27, 33, 33, 65]

2.3.4.1. Постановка ТЗ и общий принцип её решения методом потенциалов

Содержательная постановка транспортной задачи заключается в следующем [22].

Пусть в пунктах $A_1, A_2, \ldots, A_m$, называемыми пунктами производства, в количествах $a_1, a_2, \ldots, a_m$ имеется некоторый однородный продукт.

Указанный продукт потребляется в пунктах $B_1, B_2, \ldots, B_n$, называемых пунктами потребления, в количествах $b_1, b_2, \ldots, b_n$. Стоимость перевозки единицы продукта из пункта A_i в пункт B_j составляет $c_{i,j}$, где $i = 1,m; j = 1,n$.

Решение задачи состоит в нахождении такого плана перевозок $\left\| x_{i,j} \right\|_{m,n} = X$, при котором:

- все запросы потребителя будут удовлетворены;
- весь произведённый товар или продукт будет вывезен из пунктов производства;
- стоимость ***всего объёма*** перевозок будет минимальна.

Условия задачи формулируют в одной из нижеследующих форм.

1. В виде совокупности массивов A, B и C, называемых, соответственно, вектором производства, вектором потребления и матрицей стоимостей.

2. В виде таблицы, комбинирующей указанные выше массивы.

	B_1	B_2	…	B_n	
A_1	$c_{1,1}$	$c_{1,2}$	…	$c_{1,n}$	a_1
A_2	$c_{2,1}$	$c_{2,2}$	…	$c_{i,j}$	a_2
…	...	...	$c_{i,j}$	...	…
A_m	$c_{m,1}$	$c_{m,2}$	…	$c_{m,n}$	a_m
	b_1	b_2	…	b_n	

Обозначив через $x_{i,j}$ количество продукта, перевозимого из i-ого пункта в j-й, можем, руководствуясь благоприобретёнными знаниями, составить математическую модель в терминах линейного программирования

Найти минимум целевой функции

$$L = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{i,j} \cdot x_{i,j}\,, \tag{2.32}$$

при условиях

$$\sum_{i=1}^{m} x_{i,j} \geq b_j\,,\ \ j = \overline{1,n}\ \text{ и} \tag{2.33}$$

$$\sum_{j=1}^{n} x_{i,j} \leq a_i\,,\ \ j = \overline{1,m}\,. \tag{2.34}$$

Ограничение (2.23) говорит о полном удовлетворении спроса потребителей, а (2.34) – о непревышении объёмом перевозок объёма произведенного продукта.

Матрица *X* называется матрицей перевозок, планом перевозок, а тако же планом коммуникаций или коммуникативным планом. Плану перевозок соответствует граф на плоскости, аналогичный представленному на рисунке 2.8 ниже.

Транспортная задача, как особая разновидность задачи линейного программирования, обладает рядом свойств:

- коэффициенты целевой функции неотрицательны (стоимости перевозок не могут быть отрицательными величинами);
- величины элементов вектора производственных запасов и вектора потребления неотрицательны;
- базисное решение закрытой модели транспортной задачи содержит *m+n-1* базисных компонент;
- допустимое решение является планом, а базисное допустимое решение является опорным планом, оптимальное решение – оптимальным планом перевозок.

Дуга графа *[A_i, B_j]* рисунка 2.8 называется коммуникацией, ей соответствует значение $x_{i,j} > 0$ – величина перевозки.

Решение задачи линейного программирования (2.32) – (2.34) известными методами весьма громоздко, причём объём вычислений растёт по мере увеличения *n* и *m*. Поэтому был предложен ряд специальных методов, рассчитанных непосредственно на решение транспортных задач.

Данные методы ориентированы на так называемую “замкнутую” или “закрытую” модель транспортной задачи.

Для такой модели и разрешимости транспортной задачи необходимо и достаточно, чтобы выполнялось ***условие баланса***:

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j\,. \tag{2.35}$$

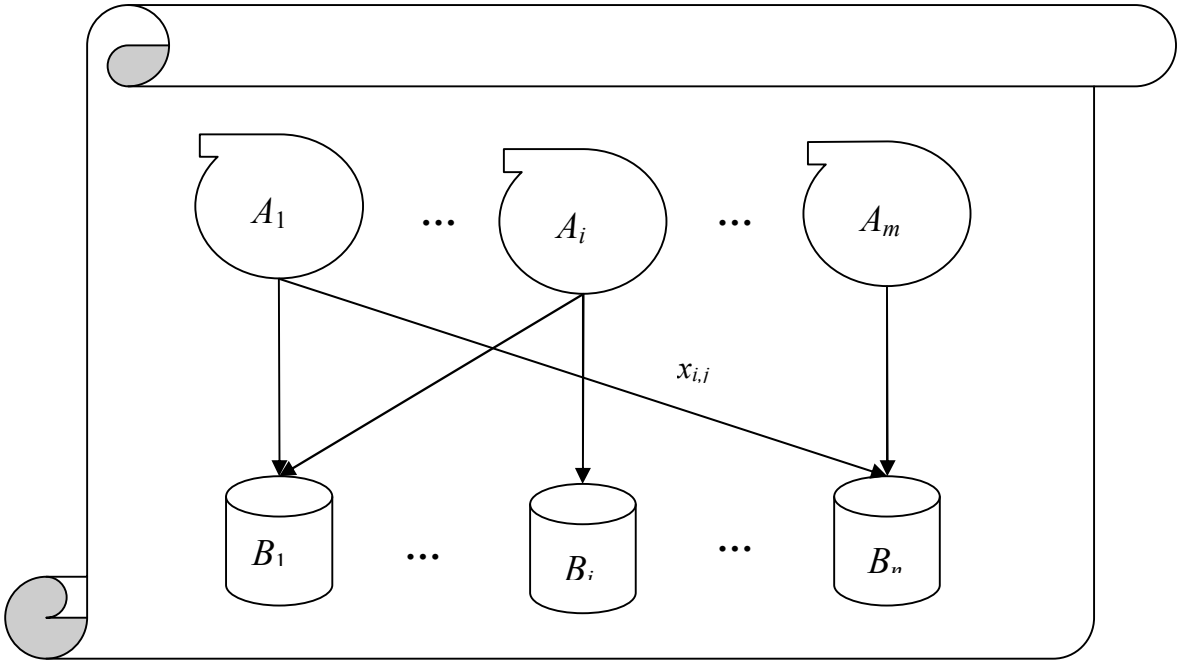


Рисунок 2.8 – Интерпретация плана перевозок

Для задачи с выполняющимся условием баланса в математической модели (2.33) и (2.34) приобретают вид равенства.

Если задача не является сбалансированной (не выполняется условие баланса), то для приведения её к сбалансированному виду необходимо ввести фиктивные пункты производства или пункты потребления. Пункт вводится с таким расчётом, чтобы обеспечить выполнения равенства (2.35), соответствующий компонент вектора A или B равен положительной разнице левой и правой частей (2.35) при отсутствии баланса. Одновременно в матрицу стоимостей C добавляется строка либо столбец с нулевым содержимым.

Иными словами, в случае, когда суммарные запасы превышают суммарные потребности, $\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j$, вводится фиктивный n+1-й потребитель, потребности которого определяются как $b_{n+1} = \sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j$. А в случае, когда суммарные потребности превышают суммарные запасы, $\sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j$, вводится фиктивный m+1-й поставщик, запасы которого есть $a_{m+1} = \sum_{j=1}^{n} b_j - \sum_{i=1}^{m} a_i$.

Стоимость перевозки единицы груза как до фиктивного потребителя, так и стоимость перевозки единицы груза от фиктивного поставщика полагают равными нулю, так как груз в обоих случаях не перевозится, и, следовательно, не может оказывать влияние на целевую функцию.

Примечание. Математическая модель обладает свойством пропорциональности: как матрицу стоимостей, так и вектора, входящие в условие задачи, можно умножать на коэффициент, отличный от нуля, либо увеличивать (уменьшать) на некоторой число. Это свойство может оказаться полезным для избавления от дробных чисел в условии задачи. Тако же необходимо помнить, что при отрицательных элементах компонентов задача потеряет содержательный смысл.

Исторически первым был подход к решению транспортных задач методом потенциалов. Считается, что данный метод ориентируется на ручное решение задачи, и, в дальнейшем, мы поймём, почему.

Алгоритм метода потенциалов структурно состоит из предварительного и итерационного этапов [29, 33, 34].

Предварительный этап состоит в ***нахождении начального опорного плана*** и расчёта ***потенциалов***.

На итерации

- проверяется условие окончания;
- определяется коммуникация, вводимая в базис;
- определяется коммуникация, выводимая из базиса;
- синхронно пересчитываются план коммуникаций, значения потенциалов и текущее значение функции цели.

Для нахождения начального опорного плана известно несколько методов. Мы рассмотрим ниже ***метод северо-западного угла***, ***метод минимальной стоимости*** и ***метод штрафов***, который часто нахывают по имени автора – методом ***Фогеля***.

Опорный план обладает рядом ***свойств***.

- Он ***невырожден***. Это означает, что число положительных элементов невырожденного плана должно быть ***не менее***, чем ранг матрицы ограничений эквивалентной ЗЛП, то есть

$$r = n + m - 1, \qquad (2.36)$$

где m и n – числа поставщиков и потребителей в ***замкнутой*** модели транспортной задачи.

- Опорный план представляет более или менее удачное решение ТЗ, в котором весь товар вывезён из пунктов производства, а потребности пунктов потребления полностью удовлетворены.
- Из элементов опорного плана нельзя составить замкнутую цепочку.

Для примера будем использовать, в дальнейшем, следующее условие ТЗ

	B_1	B_2	B_3	B_n	
A_1	7	8	1	2	160
A_2	4	5	9	8	140
A_3	9	2	3	6	170
	120	50	190	110,	

на примере которого будем демонстрировать работу алгоритмов.

Проверим условие баланса (2.35):

$$120 + 50 + 190 + 110 = 160 + 140 + 170 = 470,$$

следовательно, мы имеем дело с закрытой математической моделью ТЗ и можем приступать её к решению.

2.3.4.2. Нахождение начального опорного плана ТЗ методом северо-западного угла [33, 34]

Метод получил своё название за “географический” подход к построению плана. По аналогии с топографической картой, левый верхний угол текущего прямоугольного участка плана является северо-западным.

1. Первоначально строится пустая матрица размерами $m \times n$, снизу записываются компоненты вектора производства, а слева – потребления.

2. Для текущего северо-западного угла с координатами (i, j), первоначально (1, 1), выбирается минимальный элемент между i-ым значением вектора производства и j-ым значением вектора потребления

$$x_{i,j} = \min\{a_i, b_j\}, \qquad (2.37)$$

который помещается в план коммуникаций.

3. Элементы векторов производства и потребления, соответствующие заполненной позиции плана, уменьшаются на величину $x_{i,j}$, то есть

$$a_i = a_i - x_{i,j}, \;\; b_j = b_j - x_{i,j}. \qquad (2.38)$$

4. Строка или столбец плана, для которых компоненты векторов производства или потребления обнулились, из рассмотрения исключаются.

Таким образом, возникают координаты нового северо-западного угла: (*i+1, j*), (*i, j+1*) или (*i+1, j+1*).

5. Шаги настоящего алгоритма 2 – 4 продолжаются до тех пор, пока компоненты векторов производства и потребления не окажутся обнулены. Очевидно, что при закрытой модели ТЗ указанное событие будет иметь место.

6. Все элементы плана *X*, не заполненные по завершению работы алгоритма, полагаются равными нулю.

7. Подсчитав число ненулевых элементов, сравниваем его с оценкой (2.36), определяем невырожденность или вырожденность плана.

8. Вычисляем значение целевой функции по формуле (2.28)

$X_0 =$

$120^{1)}$	$40^{2)}$			160	40	0
	$10^{3)}$	$130^{4)}$		140	130	0
		$60^{5)}$	$110^{6)}$	170	110	0
120	50	190	110			
0	10	60	0			
	0	0				

Цифрами со скобками обозначен порядок заполнения плана. Слева и снизу от таблицы показана динамика изменения компонентов a_i и b_j по ходу построения X_0.

Исследуем полученный план на невырожденность: r =3 + 4 – 1 = 6. Число ненулевых элементов плана тако же равно 6-ти. Следовательно, план невырожден.

Рассчитаем значение целевой функции (2.32) для данного плана:

$$L = 7 \times 120 + 8 \times 40 + 5 \times 10 + 9 \times 130 + 3 \times 60 + 6 \times 110 = 3220.$$

Так как расчёты проводились вручную, то формуле (2.32) следовали не буквально, а опуская умножение на ноль.

Замечания.

- Построение опорного плана по методу северо-западного угла осуществляется алгоритмически однозначно.
- Встречается и чисто математическая формулировка алгоритма построения элементов плана

$$\begin{cases} a_\lambda^{(k)} = a_\lambda - \sum_{j=1}^{\mu-1} x_{\lambda,j}, \\ b_\mu^{(k)} = b_\mu - \sum_{i=1}^{\lambda-1} x_{i,\mu}, \\ x_{\lambda,\mu} = \min\left\{a_\lambda^{(k)}, b_\mu^{(k)}\right\}, \end{cases}$$

где литерами (k) обозначен номер итерация.

- Очевидно, что данный способ построения – один из наихудших, ибо ориентируется лишь на координаты заполняемой ячейки плана.

Последнее замечание стимулировало развитие методов, направленных на “удачное” составление начального плана, которые учитывают значения элементов матрицы стоимостей C.

2.3.4.3. Нахождение начального опорного плана ТЗ методом минимальной стоимости [33, 34]

Алгоритм метода опирается на очевидное житейское соображение о целесообразности дешёвой перевозки бОльшего количества товаров. Поэтому построение опорного плана начинается с минимальных элементов по мере возрастания.

1. Так же, как и в предыдущем методе, строится пустая матрица размерами $m \times n$, снизу записываются компоненты вектора производства, а слева – потребления.

2. Элементы матрицы C индексируются в порядке их немонотонного возрастания. Из-за того, что несколько элементов могут иметь одинаковые значения и, как следствие, порядок индексации, очередность заполнения плана, в этом случае, может разниться.

3. Заполнение плана X_0 выполняется в порядке проставленной нумерации, при этом руководствуются выражениями (2.36) и (2.37) для выбора величины перевозки и коррекции векторов A и B.

4. Строка или столбец матрицы стоимостей, для которых компоненты векторов производства или потребления обнулились, из рассмотрения исключаются. Следующая позиция плана определяется минимальным индексом из не исключённой для рассмотрения элементов матрицы C.

5. Шаги алгоритма выполняются до обнуления компонентов векторов A и B. Последнее для замкнутой модели ТЗ неизбежно. Не заполненные элементы плана X полагаются равными нулю. Рассчитывается значение целевой функции и проверяется невырожденность плана.

Замечания.

- Построение опорного плана по методу минимальной стоимости осуществляется алгоритмически неоднозначно. При наличии одинаковых элементов матрицы C, они могут получить разные индексы, в соответствии с волей индексирующего. Как следствие, для одних и тех же условий задачи теоретически могут быть получены различные начальные опорные планы.

- При очевидной целесообразности алгоритма, в процессе работы не учитывается топология распределения минимальных элементов в матрице стоимостей.

Распределения минимальных элементов в матрице *C* позволяет учесть алгоритм метода штрафов, изложение которого ниже.

При ручной реализации алгоритма часто используют таблицу, которая совмещает индексацию, начальный опорный план и матрицу стоимостей “в одном лице”. Пример работы алгоритма представим с использованием совмещённой таблицы.

В левом верхнем углу ячейки таблицы помещается индекс, а в правом верхнем – значение целевой функции, внизу, по центру, помещаются значение величины перевозки.

9) 7	9) 8	1) 1	2) 2	160	0		
		160[1)]					
5) 4	6) 5	11) 9	10) 8	140	20	0	
120[4)]			20[6)]				
12) 9	3) 2	4) 3	7) 6	170	120	90	0
	50[2)]	30[3)]	90[5)]				
120	50	190	110				
0	0	30	20				
		0					

Обратите внимание, что «2» и «8», встреченные в матрице *C*, могли бы иметь и отличную от полученной индексацию.

Мы видим, что сей план не вырожден, а его целевая функция равна

$$L = 160 \times 1 + 120 \times 4 + 20 \times 8 + 50 \times 2 + 30 \times 3 + 90 \times 6 = 1530,$$

что представляет результат, намного лучший, нежели тот, что получен по методу северо-западного угла.

2.3.4.3. Нахождение начального опорного плана ТЗ методом Фогеля [33, 34]

Метод ***Фогеля*** он же ***метод штрафов*** учитывает не только величины элементов матрицы *C*, но и их взаимоположение с другими элементами в столбце либо в строке.

Штрафом вектора, соответствующего строке либо столбцу матрицы C, называется неотрицательная разность между минимальным элементом вектора и следующим за ним ***по величине*** элементом этого же вектора.

Алгоритм метода Фогеля

1. Так же, как и в предыдущих алгоритмах, первоначально строится пустая матрица X_0 размерами $m \times n$, снизу записываются компоненты вектора производства, а слева – потребления.

2. Выполняется расчёт штрафов для столбцов и строк текущей подматрицы матрицы C.

3. Выбирается строка либо столбец с ***максимальным*** штрафом, при этом минимальный компонент для выбранного вектора определит позицию плана, подлежащую заполнению.

4. Заполнение позиции плана X_0 осуществляется согласно выражениям (2.33) и (2.34) для выбора величины перевозки и коррекции векторов A и B.

5. Строки и столбцы матрицы C, соответствующие обнулённым элементам векторов производства и потребления, при определении штрафов ***не рассматриваются***, формируя тем самым новую подматрицу матрицы C, используемую для дальнейших расчётов.

6.Функционирование алгоритма с п. 2 продолжается до тех пор, пока размеры подматрицы матрицы C не сократятся до строки (подстроки) или столбца (подстолбца).

7. Часть плана X_0, соответствующая незаполненной подматрице матрицы C заполняется по правилу минимальной стоимости, в порядке возрастания элементов.

8. Оставшиеся компоненты плана X_0 полагаются равными нулю, рассчитывается целевая функция, план проверяется на невырожденность.

Замечания.

- При равенстве штрафов отсутствует правило выбора для осуществления заполнения, поэтому построение опорного плана осуществляется однозначно.
- Метод Фогеля очень часто (подтверждено опытом) даёт решение ТЗ, совпадающее с оптимальным.

Используем известный численный пример для демонстрации хода выполнения алгоритма:

- литеры $Ш_i$ обозначают штрафы столбцов и строк на i–й итерации;
- серым цветом выделены значение штрафа, используемое на i–й итерации, и заполняемая ячейка плана;

- цифра со скобкой определяет порядок заполнения, который тождественен номеру итерации;
- ячейки (2, 2) и (2, 3) заполняются после четвёртого шага по методу минимальной стоимости.

| | | |

2.3.4.5. Алгоритм решения ТЗ методом потенциалов

Метод потенциалов является глубокой модификацией симплекс-метода решения ЗЛП применительно к транспортной задаче замкнутого типа. Он позволяет, приняв в качестве начального, некоторое допустимое решение, получить оптимальное решение за конечное число итераций.

Потенциал – это число, характеризующее удобство расположения пункта производства или потребления относительно текущего плана и заданной матрицы стоимостей.

Алгоритм решения транспортной задачи методом потенциалов структурно таков.

1. Проверка условия баланса (2.35) и, при необходимости, балансировка задачи.
2. Предварительный этап, который включает построение начального опорного невырожденного плана и расчёт предварительных потенциалов.
3. Итерация, которая включает в себя проверку на оптимальность решения, определение вводимой в базис коммуникации и коммуникации выводимой из него.
4. Корректировка значения целевой функции и матрицы потенциалов.
5. Структурная схема данного алгоритма представлена ниже, на рисунке 2.9.

Предварительный этап.

На этом этапе необходимо построить опорный план любым методом. План должен быть невырожденным. Для невырожденного опорного плана следует рассчитать потенциалы пунктов производства и пунктов потребления.

Потенциалы пунктов производства и потребления рассчитываются путём решения системы уравнений, которые решаются последовательно для базисных переменных плана.

$$\begin{cases} u_1 = 1, \\ c_{i,j} = v_j - u_i \mid x_{i,j} > 0. \end{cases} \qquad (2.38, \text{а})$$

$$\begin{cases} u_1 = 1, \\ c_{i,j} = v_j + u_i \mid x_{i,j} > 0. \end{cases} \qquad (2.38, \text{б})$$

В формулах (2.38) переменная u_i – потенциалы пунктов производства, а v_j – потенциалы пунктов потребления. Всего имеем $n + m$ неизвестных, а в опорном плане обычно $n + m - 1$ базисных переменных, соответствующих ненулевым коммуникациям.

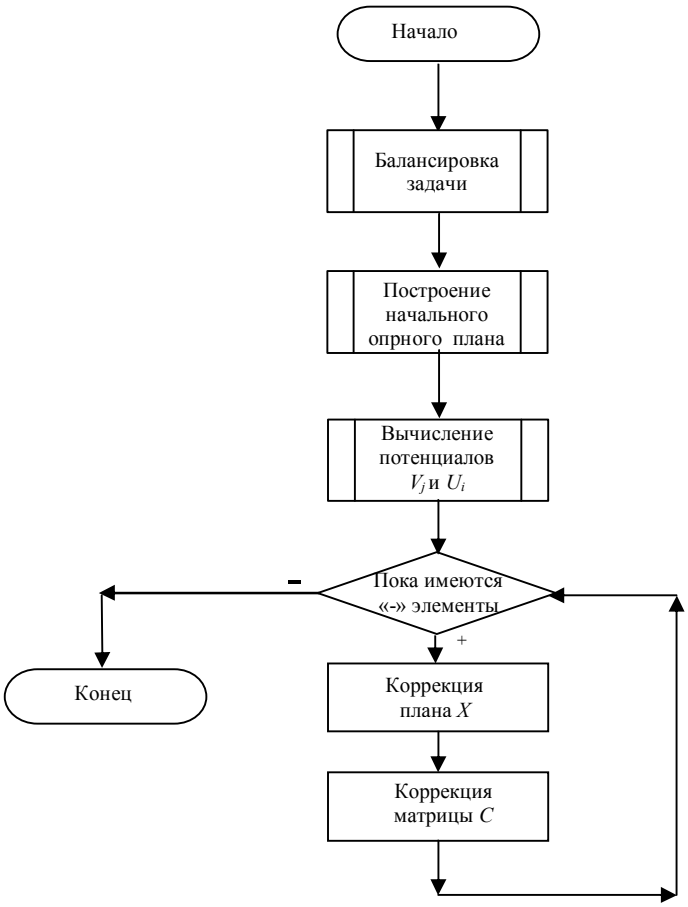


Рисунок 2.9 – Алгоритм метода потенциалов

Поэтому значение потенциала u_1 задаётся принудительно, величина не имеет значения, традиционно это единица или нуль.

Очевидно, что при вырожденном плане уравнений “не хватает”. Для приведения начального плана из вырожденного к невырожденному, в план коммуникаций X_0 добавляются так называемые ε-нули, обозначаемые в плане как 0^{ε} и считающиеся условно положительными. При этом, необходимо, чтобы

- из элементов плана нельзя было составлять замкнутые цепочки и
- обеспечивался в той или иной последовательности обход поворотами на 90° всех элементов плана.

Необходимо добавлять столько ε-нулей, сколько необходимо для получения невырожденного плана.

Постановка ε-нуля задача несколько нетривиальная, как известная головоломка о ферзях под боем. Вручную задача решается достаточно просто, алгоритмическая же её реализация относится к классу “жадных”

комбинаторных задач, требующая для своего решения временных ресурсов, что является минусом метода при реализации.

После расчёта потенциалов пунктов производства и пунктов потребления, матрица стоимости пересчитывается по одной из нижеследующих формул

$$C_{i,j}^{(0)} = c_{i,j} - (v_j - u_i), \qquad (2.39, \text{а})$$

$$C_{i,j}^{(0)} = v_j + u_i - c_{i,j}. \qquad (2.39, \text{б})$$

Причём, должно быть соответствие индексов *а*) и *б*) между формулами определения потенциалов (2.38).

При таком способе расчётов для базисных элементов в матрице C_0 будут получены нули, что может служить для проверки правильности хода вычислений. Матрицу C_0 и последующие C_i можно трактовать как “разность потенциалов” по аналогии с электрическим полем.

Проверка условия оптимальности.

Критерий достижения оптимума формулируется по-разному в зависимости от расчётных формул, применяемых на предварительном этапе:

- для случая а) компоненты текущей матрицы C стоимости должны быть неотрицательны;
- для случая б) компоненты матрицы C неположительные.

Поэтому итерационная часть выполняется до достижения неотрицательности, случай (2.39), а, либо до неположительности, случай (2.39), б, содержимого матрицы C_i.

Итерация.

Определяется коммуникация, вводимая в план X. На неё указывает:

- для случая а) – ***самый отрицательный*** компонент текущей матрицы C,
- для случая б) – ***самый положительный*** компонент текущей матрицы C, который обозначим как δ. Этот элемент называется ***направляющим***.

Соответствующая этой позиции коммуникация будет вводиться в план, обозначают её символом $\mathbf{0}^+$.

Построение кооректирующей цепочки.

Для определения коммуникации, выводимой из базиса, необходимо ***построить замкнутую цепочку***, поворачивающую под углами в 90°. Цепочка в данной задаче играет роль разгрузочного цикла: товар как бы

перемещается в равном количестве по цепочке, где-то прибывая, где-то убывая – закон Ломоносова-Лавуазье для замкнутой модели должен выполняться, ибо опорный план есть решение ТЗ.

Алгоритм построения замкнутой цепочки

Используется метод вычёркивания, суть которого состоит в следующем.

1. Вычеркнуть строки текущего плана X, содержащие менее двух ненулевых элементов. К таковым относятся: собственно коммуникации $x_{i,j}$, коммуникация, вводимая в базис 0^{+} и фиктивные коммуникации 0^{ε}, введённые ранее для обеспечения невырожденности плана.

2. Проделать такую же операцию по столбцам, не учитывая вычеркнутые ранее коммуникации

3. Выполнять пп. 1 и 2 до тех пор, пока удаётся выполнять вычёркивание.

Оставшиеся элементы образуют замкнутую цепочку.

Индексация цепочки.

Начиная с 0^{+}, вводимого в базис, нечётные элементы цепочки отмечаются знаком “–“ «минус», а чётные – знаком “+” «плюс». Направление индексации значения не имеет.

Выбор корректирующего элемента.

Из элементов цепочки выбирается минимальный элемент θ. В выборах участвуют и ε-нули 0^{ε}.

Пересчёт текущего значения целевой функции.

Можно, после коррекции плана коммуникаций, в следующем пункте настоящего алгоритма, воспользоваться формулой (2.32), но существует и более простая расчётная рекурсивная формула

$$L = L - |\delta| \cdot \theta . \qquad (2.40)$$

Коррекция плана коммуникаций.

Построение улучшенного плана коммуникаций осуществляется согласно индексации цепочки: элемент θ прибавляется к элементам, имеющим индекс «плюс» и вычитается из элементов, имеющих знак «минус». Элементы плана X, ***не попавшие*** в цепочку, ***не изменяются***.

Если несколько элементов цепочки, отмеченные знаком «минус» одинаковы и равны θ, то в ходе коррекции план может перестать быть

опорным (базисным). Чтобы этого не произошло ***всем обнулившимся элементам*** цепочки, ***кроме одного***, присваивается значение 0^{ε}.

Назначение ε-нулей 0^{ε} выполняется произвольно, без какого-нибудь критерия. Вовлечение ε-нулей в цепочке делает возможным их использование в качестве корректирующего элемента, таким образом, распределение ε-нулей в матрице X, по мере выполнения итераций, меняется, и становится возможным восстановление первоначальной конфигурации ε-нулей.

Обнаружение этого факта при машинной реализации проблематично – необходимо “помнить” всю последовательность планов коммуникации. Корректировка конфигурации ε-нулей – тако же нетривиальная задача. Поэтому, при машинной реализации алгоритма метода потенциалов поступают следующим образом: вводят параметр, назначение которого – максимальное число итераций, в течение которых целевая функция L неизменна. Превышение числа итераций вызывает остановку вычислительного процесса и позволяет избежать зацикливания [29].

Пересчёт потенциалов и матрицы стоимостей.

В методе потенциалов матрица стоимостей изменяется синхронно с планом коммуникаций. Пересчитать потенциалы и матрицу стоимостей C можно, в принципе, используя формулы (2.35) и (2.36), так сказать, « в лоб», но существует и следующий алгоритм.

Алгоритм преобразования матрицы *C*

1. Отмечаются нули текущей матрицы C, которые соответствуют базисным элементам нового пересчитанного плана коммуникаций X.
2. Вычёркивается направляющая строка.
3. Вычёркиваются столбцы, содержащие вычеркнутые базисные элементы.
4. Вычёркиваются строки, содержащие вычеркнутые при вычёркивании столбцов базисные элементы.
5. Пункты 3 и 4 повторяются циклически, пока элементы матрицы C, соответствующие базисным, окажутся либо не вычеркнуты ни разу, либо вычеркнуты двухкратно.
6. Ко всем элементам вычеркнутых строк необходимо прибавить $|\delta|$, а из всех столбцов – вычесть $|\delta|$ (модуль “дельта”).

Для контроля необходимо отметить, что элементы матрицы C, соответствующие базисным, должны остаться нулевыми.

На этом итерационную часть можно считать законченной.

Демонстрационный пример.
Используем уже известное нам условие.

	B_1	B_2	B_3	B_4	
A_1	*7*	*8*	*1*	*2*	*160*
A_2	*4*	*5*	*9*	*8*	*140*
A_3	*9*	*2*	*3*	*6*	*170*
	120	*50*	*190*	*110*	

Условие баланса для неё выполнено изначально, как мы установили выше, балансировка не потребовалась. Используем в качестве опорного плана невырожеднный план, построенный нами ранее по методу минимальной стоимости.

$$X_0 = \begin{array}{|c|c|c|c|} \hline 0 & 0 & 160^{1)} & 0 \\ \hline 120^{6)} & 0 & 0 & 20^{5)} \\ \hline 0 & 50^{3)} & 30^{2)} & 90^{4)} \\ \hline \end{array} \quad L = 1530$$

Выполним расчёт потенциалов для этого плана, руководствуясь формулой (2.38) и расположением элементов опорного плана.

$u_1 = 1$;
$c_{13} = 1 = v_3 - u_1 = v_3 - 1$; $v_3 = 2$;
$c_{33} = 3 = v_3 - u_3 = 2 - u_3$; $u_3 = -1$;
$c_{32} = 2 = v_2 - u_3 = v_3 + 1$; $v_2 = 1$;
$c_{34} = 6 = v_4 - u_3 = v_4 + 1$; $v_4 = 5$;
$c_{24} = 8 = v_4 - u_2 = 5 - u_2$; $u_2 = -3$;
$c_{21} = 4 = v_1 - u_2 = v_1 + 3$; $v_1 = 1$.

Порядок перемещения по опорному плану в ходе расчёта потенциалов следующий (1, 3) $\Rightarrow$ (3, 3) $\Rightarrow$ (3, 2) $\Rightarrow$ (3,4) $\Rightarrow$ (2, 4) $\Rightarrow$ (2,1), он показан индексами на опорном плане.
Это не единственно возможный порядок расчёта потенциалов. После (3, 3) он мог быть и таким: (3, 3) $\Rightarrow$ (3,4) $\Rightarrow$ (2, 4) $\Rightarrow$ (2, 1), $\Rightarrow$ (3, 2). Важно, чтобы на каждом этапе расчётов в уравнении (2.38) присутствовала только одна неизвестная.

Пересчитаем матрицу C в матрицу C_0, используя формулу (2.39), а.

$c_{11} = 7 - (1 - 1) = 7$; $c_{21} = 4 - (1 + 3) = 0$; $c_{31} = 9 - (1 + 1) = 7$;
$c_{12} = 8 - (1 - 1) = 8$; $c_{22} = 5 - (1 + 3) = 1$; $c_{32} = 2 - (1 + 1) = 0$;
$c_{13} = 3 - (2 - 1) = 0$; $c_{23} = 9 - (2 + 3) = 4$; $c_{33} = 3 - (2 + 1) = 0$;
$c_{14} = 2 - (5 - 1) = -2$; $c_{24} = 8 - (5 + 3) = 0$; $c_{34} = 6 - (5 + 1) = 0$.

Порядок пересчёта здесь не существенен. Видно, что среди элементов матрицы C_0 присутствуют отрицательные, поэтому необходимо улучшать текущий план.

Итерация 1.

Направляющий элемент наибольший отрицательный $c_{1,4} = -2$. Отмечаем в плане X_0 соответствующий элемент как $\mathbf{0}^+$, строим замкнутую цепочку и индексируем её.

$C_0 =$

7	8	$\bar{0}$	-2	x_1
$\bar{0}$	1	4	$\bar{0}$	
7	$\bar{0}$	$\bar{0}$	0	x_3
	x_4	x_2		

$X_0 =$

		160^-	$\mathbf{0}^+$	
120			20	x_3
	50	30^+	90^-	
	x_1	x_2		

Порядок вычёркивания показан крестиками с номерами, вычеркнутые строки и столбцы выделены фоном. Видно, что “белые” элементы образуют цепочку, и больше вычёркиваний произвести не удастся.

Выполняем индексацию от 0^+ хоть влево, хоть вниз, и выбираем элемент для коррекции $\theta = \min\{160,90) = 90$. Текущее значение функции цели при этом станет, согласно (2.40)

$$L = 1530 - |-2| \times 90 = 1350.$$

Соответствующий этой функции план X_1, полученный после коррекции, показан ниже. Пересчитаем матрицу C. Обозначим «крышками» сверху нули, ***соответствующие базисным элементам нового плана*** (важно!!!). Порядок вычёркивания будем обозначать крестиком с цифрой – номером вычёркивания, а вычеркнутые строки и столбцы обозначать фоном.

Прибавляем $|\delta| = 2$ к вычеркнутым строкам и вычитаем от вычеркнутых столбцов. Очевидно, что, в ходе этой операции, дважды вычеркнутые и ни разу не вычеркнутые элементы останутся неизменными.

$C_1 =$

9	8	0	0
0	-1	2	0
9	0	0	2

$X_1 =$

		70	90
120			20
	50	120	

Так как среди элементов матрицы C_1 имеются отрицательные числа, то решение не закончено.

Итерация 2.

$C_1 =$

9	8	$\bar{0}$	$\bar{0}$
$\bar{0}$	-1	2	0
9	$\bar{0}$	$\bar{0}$	2

x_2 x_1

$X_1 =$

		$70^{-}\rightarrow$	$90^{+}\downarrow$
120	$\mathbf{0}^{+}\downarrow$	$\leftarrow\uparrow$	$\leftarrow 20^{-}$
	$50^{-}\rightarrow$	$120^{+}\uparrow$	

x_1

Направляющий элемент в C_1 имеет координаты (2,2). Помещаем $\mathbf{0}^{+}$ в соответствующую позицию плана X_1 и приступаем к вычёркиванию. Получающаяся при этом цепочка имеет вид неправильной восьмёрки, проход по которой показан стрелками.

Индексация позволяет выбрать направляющий элемент $\theta = \min\{50,70,20\} = 20$. Целевая функция при этом составит

$$L = 1350 - |-1| \times 20 = 1330,$$

а соответствующий скорректированный план X_2 показан ниже. Пересчитаем теперь матрицу C_1, предварительно отметив базисные элементы плана X_2 в C_1.

Вычеркнем 2-ю строку и 1-й столбец, вычитая и прибавляя *1*, получим матрицу C_2, которая с соответствующим планом X_2 показана ниже.

$C_2 =$

8	8	0	0
0	0	3	1
8	0	0	2

$X_2 =$

		50	110
120	20		
	30	140	

Полученный план совместно со значением целевой функции, есть оптимальное решение транспортной задачи. Сравнив полученный результат с опорным планом, построенным по методу Фогеля, заметим, что оба решения полностью совпадают.

2.3.3.6. Алгоритм решения ТЗ венгерским методом [22, 29]

Данный метод был предложен в 1931 г. Р. Эгервари, венгром по национальности, и 1953 г. подвёргся модификации Г. Куном. В честь родины авторов получил своё наименование. Точнее, этим названием

обозначается группа методов. Задача представляется общепринятой замкнутой моделью (2.32) – (2.34).

Метод базируется на ряде определений.

1. ***Суммарной невязкой плана*** называют величину, определяемую выражением

$$\Delta = \sum_{i=1}^{m} a_i + \sum_{j=1}^{n} b_j - 2 \cdot \sum_{i=1}^{m} \sum_{j=1}^{n} x_{i,j} \,. \tag{2.40}$$

2. Величины

$$\delta_j = b_j - \sum_{i=1}^{m} x_{i,j} \tag{2.41}$$

и

$$\delta_i = a_i - \sum_{i=1}^{m} x_{i,j} \tag{2.42}$$

называют, соответственно, ***невязками по столбцам*** и ***невязками по строкам***. Поддаётся выводу и следующее выражение

$$\Delta = \sum_{i=1}^{m} \delta_i + \sum_{j=1}^{n} \delta_j \,. \tag{2.43}$$

3. Столбцы и строки матрицы C, отмеченные в ходе работы алгоритма символом «плюс» называются ***выделенными***.

4. Нулевой элемент матрицы C ($c_{i,j} = 0$), для которого в плане X существует коммуникация ($x_{i,j} \geq 0$), называется ***существенным нулём*** матрицы C.

Алгоритм венгерского метода

Граф-схема алгоритма представлена на рисунке 2.10. Структурно алгоритм состоит из следующих пунктов.

1. Предварительный этап.
2. Проверка оптимальности полученного плана.
3. Итерационная часть.

В свою очередь, итерационная часть состоит из трёх этапов, которым предшествует этап разметки:

- 1-й этап – поисковый;
- 2-й этап – построение цепочки и коррекция плана;

- 3-й этап – выполнение эквивалентных преобразований матрицы *C*.

Итерация, после разметки, начинается первым этапом, а оканчивается вторым, причём в процессе одной и той же итерации первый и третий этапы могут многократно повторяться.

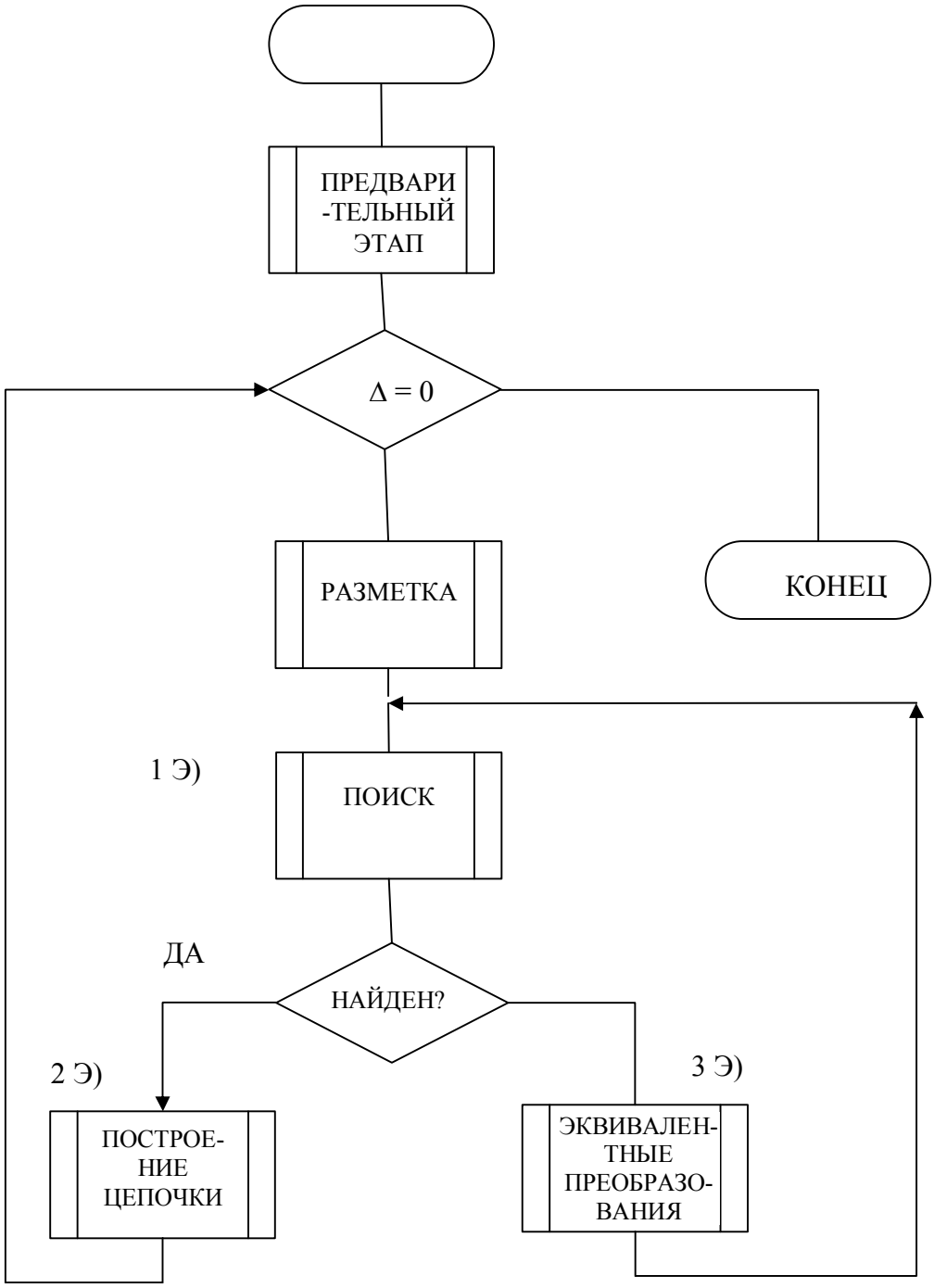


Рисунок 2.10 – Алгоритм венгерского метода

Предварительный этап.

Состоит в построении начального плана X_0 и определении его невязок.

- В каждом столбце матрицы C отыскивается минимальный элемент, который затем вычитается из всех элементов этого столбца. В результате, матрица C преобразуется в матрицу $C`$.
- Проделав аналогичную операцию над строками в матрице $C`$, получим матрицу C_0.
- Для нулевых элементов матрицы C_0, перемещаясь по столбцам сверху вниз и слева направо, заполним план X_0. Порядок определения коммуникаций, коррекции векторов производства и потребления рассмотрен нами ранее при построении начальных опорных планов.
- Вычисляется суммарная невязка (2.40) полученного плана.

Заметим, что невязки по столбцам (2.41) и строкам (2.42) получаются автоматически в ходе построения начального плана, а суммарную невязку проще получить, используя (2.43).

В отличие от опорных планов метода потенциалов, планы венгерского метода таковыми не являются (смотри свойства опорных планов).

Проверка условия окончания.

Если суммарная невязка Δ текущего плана X равна нулю, то полученный план является оптимальным. Необходимо рассчитать целевую функцию (2.32).

ИТЕРАЦИОННАЯ ЧАСТЬ АЛГОРИТМА.

Разметка.

Разметка текущей матрицы C выполняется в начале итерации и сохраняется до её конца с теми изменениями, которые вносятся в неё по мере выполнения алгоритма.

- Выделить знаком «плюс» j-е столбцы матрицы C, обладающие нулевой невязкой $\delta_j = 0$.
- Выделить чертой сверху существенные нули матрицы C.

Этап 1 – этап поиска.

Область поиска: невыделенная часть матрицы C – невыделенные столбцы и строки.

Цель поиска: найти в невыделенной части матрицы C нуль, стоящий в строке, которой в плане X соответствует положительная невязка $\delta_i \geq 0$.

Поисковый этап заканчивается одним из случаев:

- если все нули матрицы C находятся в выделенной части, то необходимо перейти к этапу 3 – эквивалентных преобразований матрицы C;
- поиск завершился успешно, найден ноль в строке с положительной невязкой. В этом случае далее выполняется этап 2 – построение цепочки и коррекция плана коммуникаций.

В ходе поиска невыделенная часть матрицы C просматривается по столбцам сверху вниз а столбцы – слева направо (“По-китайски”).

Пусть среди элементов найден ноль. Его отмечают апострофом (штрихом) и анализируют невязку по строке δ_i.

Если невязка δ_i положительна, то этот ноль со штрихом является искомым, а поиск заканчивается успешно.

Если невязка δ_i нулевая, то текущая стока выделяется знаком «плюс», и просматривается по местам её пересечения с выделенными столбцами. Если в месте пересечения стоит существенный нуль, то его обозначают звёздочкой (*), а знак выделения над столбцом уничтожают, обводя кружком или заключая в скобки. Столбец становится невыделенным и делается доступным для поиска. Поиск далее продолжают по этому столбцу со снятым выделением.

Этап 2 – этап построения цепочки и коррекции плана.

Цепочка ***не замкнута***, так как существуют невязки, которые, в замкнутой модели, фактически означают наличие неудовлетворённого спроса и невывезенного товара. За счёт последних и будет пополняться план коммуникаций.

1. Цепочка составляется из нулей со штрихом (0’) и нулей со звёздочками (0^*), содержит нечётное число элементов, и, в принципе, может состоять и из одного нуля со штрихом.

2. Цепочка начинается от последнего найденного нуля со штрихом 0’ к нулю со звездой 0^* по столбцу, далее, по направлению под 90° к предыдущему, по строке к нулю со штрихом и так далее. На нечётных местах цепочки будут стоять нули со штрихом, а на чётных – нули со звёздами. Цепочка начинается в строке с положительной невязкой и заканчивается в столбце с положительной невязкой нулём со штрихом.

3. Выбирается корректирующий элемент

$$\theta = \min\left\{\delta_i^{НАЧАЛА}, \delta_j^{КОНЦА}, x_{i,j}^*\right\}, \qquad (2.44)$$

где $\delta_i^{НАЧАЛА}$ – невязка строки начала цепочки; $\delta_j^{КОНЦА}$ – невязка столбца конца цепочки; $x_{i,j}^*$ – элементы, стоящие на чётных местах цепочки.

4. После выбора θ текущий план преобразуется по алгоритму:

$$x_{i,j} = \begin{cases} x_{i,j}, & \text{не входит в цепочку,} \\ x_{i,j}^{\nabla} + \theta, & \text{нечётный элемент цепочки,} \\ x_{i,j}^{*} - \theta, & \text{чётный элемент цепочки.} \end{cases} \qquad (2.45)$$

5. Рассчитываются Δ, δ_j и δ_j по соответствующим формулам, удобным рассчитывающему (2.40) – (2.43).

Этап 3 – этап эквивалентного преобразования матрицы C.

1. Среди невыделенных элементов матрицы C выбирается минимальный положительный (а другой и быть не может, но так – в первоисточнике) элемент $h > 0$. Этот элемент называется ***корректирующим***.

2. Корректирующий элемент вычитается от невыделенных строк матрицы C.

3. Корректирующий элемент прибавляется к выделенным столбцам матрицы C.

При этом, очевидно, выделенные однократно (сиречь, стоящие в выделенной строке либо в выделенном столбце) элементы не изменятся, двукратно выделенные – увеличатся на величину h, а в невыделенной части матрицы появится хотя бы один нуль, который, в дальнейшем, буде обработан алгоритмом поиска на 1-м этапе.

Замечания по методу Эгервари.

1. В ходе работы алгоритма не используются опорные планы, поэтому зацикливание при машинной реализации не возникает.

2. По величине суммарной невязки Δ можно грубо (наихудший из возможных ход решения) оценить число итераций до получения оптимального решения:

$$N_{ОСТ} \le \frac{\Delta}{2}.$$

При получения последней формулы учтено, что при записи в план коммуникаций некоторого числа $к$ суммарная невязка текущего плана Δ уменьшается на величину, равную $2 \times к$.

Предположив, что за каждую итерацию план коммуникаций помещается единица товара, придём к обсуждаемой формуле.

Демонстрационный пример.
Решить транспортную задачу:

	B_1	B_2	B_3	B_n	
A_1	7	8	1	2	160
A_2	4	5	9	8	140
A_3	9	2	3	6	170
	120	50	190	110	

Задача сбалансирована, модель замкнутая.

Предварительный этап.
Минимальные элементы в столбцах матрицы C показаны фоном. Их вычитание позволяет получить матрицу C'.

$C' =$

3	6	0	0
0	3	8	6
5	0	2	4

$C' = C_0 =$

3	6	0	0
0	3	8	6
5	0	2	4

Так как в каждой строчке есть по минимальному элементу – нулю (серый фон), то матрицы C' и C_0 совпадут.
Для данной матрицы C_0 по её нулям строим начальный план X_0, показанный ниже справа.

$C_0 =$

+	+		
3	6	$\bar{0}$	0
$\bar{0}$	3	8	6
5	$\bar{0}$	2	4

+

$X_0 =$

				δ_i
		160		0
120				20
	50			120
0	0	30	110	δ_j

Суммарная невязка этого плана, согласно (2.43) составляет

$$\Delta = 30 + 110 + 20 + 120 = 280,$$

поэтому итерационная часть алгоритма неизбежна.

1-я итерация.
Все почти нули матрицы существенные, обозначим их как $\bar{0}$. Выделим также 1-й и 2-й столбцы как имеющие нулевые невязки.

1-й этап. Поиск.

Отмечаем первый встреченный невыделенный нуль с координатами (1, 3) штрихом. Его невязка по строке – нулевая, поэтому строка отмечается плюсом. На этой строке нет существенных нулей, стоящих в выделенных столбцах.

В невыделенной части матрицы (серая) нет нулей, поэтому этап поиска закончился неудачей, необходимо осуществлять эквивалентные преобразования матрицы C_0.

3-й этап.

В невыделенной (серой) части матрицы минимальный элемент $h = 2$. Прибавим и отнимем его согласно разметке. Получим матрицу C_1, отметим, что вся индексация переносится. В её невыделенной части образовался нуль, снова переходим к этапу поиска.

Поиск.

Отмечаем штрихом нуль с координатами (3, 3). Его построчная невязка $\delta_j = 120$ – положительна, следовательно, этап поиска завершился успешно. Можно строить цепочку.

2-й этап.

Цепочка состоит из одного элемента. Выберем элемент коррекции:

$$\theta = \min \{30, 120\} = 30.$$

Скорректируем план и невязки:

$$\Delta = \Delta - 2 \times \theta = 280 - 60 = 220.$$

Невязка положительна, решение не закончено. План X_1 представлен ниже

$C_1 =$

+	+			
5	8	$\bar{0}$`	0	+
$\bar{0}$	3	6	4	
5	$\bar{0}$	0`	2	

$X_1 =$

				δ_i
		160		0
120				20
	50	30		90
0	0	0	110	

δ_j

2-я итерация.

Размечаем матрицу C_1 и приступаем к поиску.

1-й этап, поисковый.

Выделим штрихом нуль с координатами (1, 4). Он стоит в строке с нулевой невязкой, поэтому строку выделяем плюсом. Просматриваем её пересечения с выделенными столбцами.

$C_1 =$

+	+	(+)	
5	*8*	$\bar{0}^{*} \rightarrow$	$\rightarrow 0`$
$\bar{0}$	*3*	$\uparrow 6$	*4*
5	$\bar{0}$	$\uparrow \bar{0}`$	*2*

$+$

$X_1 =$

				δ_i
		160		*0*
120				*20*
	50	30		*90*
0	*0*	*0*	*110*	
δ_j				

На пересечении с выделенным столбцом – существенный нуль, отмечаем его звёздочкой, снимаем выделение столбца, заключая его в скобки. Продолжаем поиск в третьем столбце, отмечаем нуль (3, 3) штрихом. Его невязка положительна, этот ноль – искомый, этап поиска закончился удачно.

2-й этап.

Цепочка из элементов матрицы C_1 есть последовательность (3, 3) $\Rightarrow$ (1, 3) $\Rightarrow$ (1, 4) выделена в матрице серым. Выберем корректирующий элемент: невязка строки начала – 90, невязка столбца конца цепочки – 110, $X(\bar{0}^{*}) =$ 160.

$$\theta = \min \{90, 110, 160\} = 90.$$

Выполним изменение элементов, соответствующих цепочке в плане X_1, преобразуя его в X_2.

Суммарная невязка этого плана составит

$$\Delta = 220 - 2 \times 90 = 40.$$

Необходимо провести очередную итерацию.

$X_2 =$

				δ_i
		70	90	0
120				20
	50	120		0
0	0	0	20	
δ_j				

3-я итерация.

Размечаем матрицу C_1 и приступаем к поиску.

1-й этап, поисковый.

Выделим штрихом нуль с координатами (1, 4). Он стоит в строке с нулевой невязкой, поэтому строку выделяем плюсом. Просматриваем её пересечения с выделенными столбцами.

На пересечении с выделенным столбцом – существенный нуль, отмечаем его звёздочкой, снимаем выделение столбца, заключая его в скобки. Продолжаем поиск в третьем столбце, отмечаем нуль (3, 3) штрихом. Его невязка тако же нулевая, отмечаем строку плюсом, просматриваем пересечения с выделенными столбцами.

По результатам просмотра нуль (3, 2) получает звезду, а с третьего столбца – снимается выделение. Больше в невыделенной части нулей нет. На этом поиск заканчивается неудачей.

$C_1 =$

	(+)	(+)		
5	*8*	$\bar{0}^*$	$\bar{0}$`	+
$\bar{0}$	*3*	*6*	*4*	
5	$\bar{0}^*$	$\bar{0}$`	*2*	+

$X_2 =$

				δ_i
		70	90	*0*
120				*20*
	50	120		*0*
0	*0*	*0*	*20*	

δ_j

3-й этап, эквивалентные преобразования матрицы C_1.

В невыделенной части матрицы C_1 (выделена серым, извиняюсь за каламбур) отыскиваем положительный элемент для коррекции $h = 3$. В результате имеем матрицу C_2, показанную ниже.

$C_2 =$

8	*8*	$\bar{0}^* \rightarrow$	$\rightarrow \bar{0}$` +
$\bar{0}$	$\downarrow$*0*`	$\uparrow$*3*	*1*
8	$\bar{0}^* \rightarrow$	$\uparrow \bar{0}$`	*2* +

$X_2 =$

				δ_i
		50	110	*0*
120	20			*0*
	30	140		*0*
0	*0*	*0*	*0*	

δ_j

Приступаем к поиску, 1-й этап.

Отмечаем штрихом нуль (2, 2). Его невязка по строке показывает, что этот нуль является искомым.

Этап 2-й.

Строим цепочку (2, 2)` $\Rightarrow$ (3, 2)$^*$ $\Rightarrow$ (3, 3)` $\Rightarrow$ (1, 3)* $\Rightarrow$ (1, 4)`, она показана фоном и стрелками. Корректирующий элемент

$$\Delta = \min\{\delta_2 = 20,\ \delta_4 = 20,\ x_{3,2} = 50,\ x_{13} = 70\} = 20.$$

Полученный план перевозок X_2 показан рядом с матрицей C_2 его невязка нулевая.

Следовательно, достигнут оптимум, можно рассчитать целевую функцию $L = 1330$.

Это столько же, как и в методе потенциалов. Планы перевозок, полученных обоими методами, совпадают.

Замечания по процессу решения.

1. Текущие матрицы C и X изменяются несинхронно. В принципе, возможны решения, когда в ходе нескольких итераций матрица C остаётся одним и тем же, а план X – неоднократно меняется.

2. Известны случаи, когда при равенстве целевых функций, при одном и том же условии, решённом венгерским методом и методом потенциалов, получаются разные планы перевозок. Это есть своеобразная иллюстрация теорем линейного программирования (смотри раздел ранее).

2.3.4.7. Алгоритм решения ТЗ с ограниченной пропускной способностью коммуникаций [20, 24, 28 – 30]

Общая постановка транспортной задачи, ранее нами рассмотренная, характерна тем, что возможность перевозки $x_{i,j}$ единиц товара от i-го поставщика j-му потребителю физически ничем не ограничивается. Однако повседневная практика перевозок может повлечь за собой ограничения на количество перевозимого груза.

Указанные ограничения могут иметь различную физическую или иную природу:

- быть связанными с затратами на горючее (лимитироваться);
- определяться грузоподъёмностью транспортных средств, их проходимостью или предельным числом рейсов;
- диктоваться соотношением полезного груза в функции от дистанции перевозки, как это бывает, например, в дальнебомбардировочной авиации;
- и т.п.

  В таких случаях говорят о транспортной задаче с ограниченными пропускными способностями коммуникаций. Её математическая модель несколько отличается от общеизвестной (2.32) – (2.34), в части дополнительных ограничений:

$$\min \sum_{i=1}^{m}\sum_{j=1}^{n} c_{i,j} \cdot x_{i,j};$$

$$\sum_{j=1}^{m} x_{i,j} = a_i; \sum_{i=1}^{n} x_{i,j} = b_j;$$

$$0 \le x_{i,j} \le d_{i,j}; \qquad i = 1,\overline{m}; j = 1,\overline{n}.$$

Величина $d_{i,j}$ называется ограничением коммуникации, благодаря этому задачу в данной постановке называют ещё *Td*-задачей, т.к. подразумевается, что в дополнение к известным векторам и матрицам *A*, *B* и *C* добавляется еще и матрица ограничений *D*.

Td-задачи решаются с помощью модифицированного алгоритма венгерского метода, модификации касаются учёта ограничений в ходе решения.

При изложении метода будем использовать следующие определения.

1. Элемент $c_{i,j} = 0$ матрицы *C* называется ***X-неполным нулём*** (или просто ***неполным нулём***), если в плане *X* решаемой *Td*-задачи величина $x_{i,j} < d_{i,j}$ (меньше пропускной способности коммуникаций).

2. ***Полным нулём*** (***X-полным нулём***) называется элемент $c_{i,j} = 0$ матрицы *C*, для которого $x_{i,j} = d_{i,j}$ (равен пропускной способности коммуникации).

3. Элемент $c_{i,j} = 0$ матрицы *C* является ***существенным нулём***, если $x_{i,j} > 0$ в плане перевозок *X*.

4. Если $x_{i,j}$ в плане перевозок *X* нулевой, то соответствующий нуль матрицы $c_{i,j}$ называется ***несущественным нулём***.

5. Элемент, находящийся в матрице *C* на пересечении выделенной строки и выделенного столбца, называется ***дважды выделенным***.

Функционирование алгоритма

Венгерский алгоритм решения *Td*-задачи имеет такую же структуру, показанную на рисунке 2.9, как и при отсутствии ограничений с точностью до названия этапов. Поэтому алгоритм дадим в отличиях и дополнениях.

Предварительный этап.

Заполнение плана *X* осуществляется не на основании формулы (2.37), а с учётом ограничений:

$$X_{i,j} = \min\{a_i,\ b_j,\ d_{i,j}\}. \qquad (2.46)$$

ИТЕРАЦИЯ.

Разметка.

Дополнительно отмечают ***существенные нули*** матрицы *C*: точкой сверху – *x*-неполные нули $\dot{0}$, двумя точками – x-полные $\ddot{0}$.

1-й этап, поисковый.

Цель поиска: найти ***неполный нуль*** матрицы *C*, независимо от того существенный он или несущественный.

2-й этап, построение цепочки и коррекция плана X.

Корректирующий элемент выбирается по правилу

$$\theta = \min\left\{\delta_i^{НАЧАЛА}, \delta_j^{КОНЦА}, x^*_{i,j}, r'_{i,j}\right\}, \qquad (2.47)$$

где $\delta_i^{НАЧАЛА}$ – невязка строки начала цепочки; $\delta_j^{КОНЦА}$ – невязка столбца конца цепочки; $x^*_{i,j}$ – элементы, стоящие на чётных местах цепочки, $r'_{i,j} = d_{i,j} - x_{i,j}$ – величина насыщения, необходимая для приведения коммуникации, стоящей на нечётной позиции в цепочке, к *x*-полному нулю.

3-й этап, коррекция матрицы *C*.

1. Корректирующий элемент *h* определяется как минимальный среди ***невыделенных положительных*** элементов, и, взятых ***по модулю***, ***дважды выделенных отрицательных***.

$$h = \min\{c_{i,j} > 0, |c_{i,j}^{+} < 0|+\}. \qquad (2.48)$$

2. Если все *x*-неполные нули выделены при нулевых невязках, все невыделенные элементы *C* отрицательны, а дважды выделенные – положительны, то *Td*-***задача неразрешима***.

3. Если в процессе коррекции ***дважды выделенный отрицательный*** элемент матрицы *C* ***становится нулём***, то его ***помечают звёздочкой***, а ***знак выделения*** над столбцом ***уничтожают***, делая столбец доступным для выполнения поисковых операций.

Пример решения задачи

Используем условие уже известного примера, дополненное ограничениями.

	B_1	B_2	B_3	B_4	
A_1	7	8	1	2	160
A_2	4	5	9	8	140
A_3	9	2	3	6	170
	120	50	190	110	

$D =$

50	100	200	50
150	50	100	50
50	50	50	50

Условие сбалансировано, модель замкнутая.

Предварительный этап.

На предварительном этапе матрица C_0 будет такая же, как и в случае ТЗ без ограничений.

$C_0 =$

+	+		
3	6	$\dot{0}^{\nabla}$	0
$\dot{0}$	3	8	6
5	$\ddot{0}$	2	4

$+$ $X_0 =$

				δ_i
		160		0
120				20
	50			120
0	0	30	110	

δ_j

Начальный план X_0 получился тоже идентичный, поскольку он (2.46) соответствует. Заполнение X_0 читателю рекомендуется проделать самостоятельно. Суммарная невязка плана $\Delta = 280$.

1-я итерация.

Этап разметки.

Нули (1, 3), (2, 1) являются неполными, нуль (3, 2) – полным, выделяем 1-й и 2-й столбцы.

1-й этап.

Нуль (1, 3) выделяем штрихом, а первую строку – плюсом, так как её соответствует нулевая невязка в плане X_0. На этом поиск закончен, необходимы эквивалентные преобразования.

3-й этап.

Корректирующий элемент в невыделенной, серой части матрицы, согласно (2.46), равен 2. Применение алгоритма коррекции даёт C_1.

$C_1 =$

+	+			
5	8	$\dot{0}^{\nabla}$	0	+
$\dot{0}$	3	6	4	
5	$\ddot{0}$	0^{∇}	2	

$X_1 =$

				δ_i
		160		0
120				20
	50	30		90
0	0	0	110	

δ_j

1-й этап.

Выделяем элемент (3, 3) штрихом, на этом этап поиска заканчивается удачно.

2-й этап.

Цепочка состоит из одного элемента. Соответствующую позицию плана X заполняем элементом $\theta = \min\{120, 30, 50\} = 30$. $r'_{3,3} = d_{3,3} - x_{33} = d_{3,3} - 0 = 50$. Получаем план коммуникаций X_1.

Невязка полученного плана $\Delta = 220$, расчёты продолжаются.

2-я итерация.

Этап разметки.

$C_1 =$

+	+	(+)		
5	8	$\dot{0}^{*}$	$\rightarrow 0^{\nabla}$	+
$\dot{0}$	3	$\uparrow 6$	4	
5	$\ddot{0}$	$\uparrow \dot{0}^{\nabla}$	2	

$X_2 =$

				δ_i
		140	20	0
120				20
	50	50		70
0	0	0	90	

δ_j

Нуль, добавленный на предыдущей итерации, неполный, столбцы с первого по третий – обладают нулевыми невязками в плане X_1, выделяем их соответствующим образом.

1-й этап.

Нуль (1, 4) выделяем штрихом, а первую строку – плюсом, поскольку её невязка нулевая. Нуль (1,3) выделяем звёздочкой, снимаем выделение с третьего столбца, продолжаем поиск. Нуль (3, 3) выделяем штрихом, этот нуль – искомый, будем строить цепочку.

2-й этап.

Цепочка соединяет нули с координатами в матрице C_1 $(3, 3)^{\nabla} \Rightarrow (1, 3)^{*} \Rightarrow (1, 4)^{\nabla}$.

Выбираем корректирующий элемент по формуле (2.47)

$$\theta = \min\{\delta_3 = 90, \delta_4 = 110, x^*_{1,3} = 160, r'_{3,3} = 20, r'_{3,3} = 50\} = 20$$

и перестраиваем план. При этом невязка получится Δ = 180. Итерации продолжаются.

3-я итерация.

Этап разметки.

$C_1 =$

+	+	(+)		
5	*8*	0̇ *	0̇ ∇	+
0̇	*3*	*6*	*4*	
5	0̈	0̈	*2*	

$X_2 =$

				δ_i
		140	20	*0*
120				*20*
	50	50		*70*
0	*0*	*0*	*90*	

δ_j

Нули (1, 3), (1, 4) и (2, 1) являются неполными, нули (3, 2) и (3, 3) – полными, выделяем столбцы с 1-го по 3-й.

1-й этап.

Нуль (1, 4) выделяем штрихом, а первую строку – плюсом. Нуль (1,3) выделяем звёздочкой, снимаем выделение с третьего столбца, продолжаем поиск. Больше неполных нулей в невыделенной части матрицы нет.

3-й этап.

Корректирующий элемент матрицы h = 2. Преобразуем матрицу C_1 в C_2.

1-й этап.

Отмечаем нуль (3, 4) штрихом. Его невязка положительна, он несущественный и неполный. Поиск завершён.

3-й этап.

Цепочка состоит из одного нуля с координатами (3, 4), эта позиция в плане заполняется исходя из невязок строки, столбца и ограничения коммуникации

$$\theta = \min\{\delta_3 = 70, \delta_4 = 90, r'_{3,4} = 50\} = 50.$$

Получим план X_3. Его невязка Δ = *80*. Следовательно, оптимум не получен.

$C_2 =$

+	+			
7	10	$\dot{0}^{*}$	$\dot{0}^{\nabla}$	+
$\dot{0}$	3	4	2	
5	$\ddot{0}$	–2	0^{∇}	

$X_3 =$

				δ_i
		140	20	0
120				20
	50	50	50	20
0	0	0	40	

δ_j

4-я итерация.

Этап разметки.

$C_2 =$

+	+	(+)		
7	10	$\dot{0}^{*}$	$\dot{0}^{\nabla}$	+
$\dot{0}$	3	4	2	
5	$\ddot{0}$	–2	$\ddot{0}$	

$X_3 =$

				δ_i
		140	20	0
120				20
	50	50	50	20
0	0	0	40	

δ_j

Нули (1, 3), (1, 4) и (2, 1) являются неполными, нули (3, 2) и (3, 4) – полными, выделяем столбцы с 1-го по 3-й.

1-й этап.

Нуль (1, 4) выделяем штрихом, а первую строку – плюсом. Нуль (1,3) выделяем звёздочкой, снимаем выделение с третьего столбца, продолжаем поиск. Больше неполных нулей в невыделенной части матрицы нет.

3-й этап.

Корректирующий элемент матрицы $h = 2$ выбираем из невыделенной серой части. Преобразуем матрицу C_2 в C_3.

$C_3 =$

+	+			
9	*12*	$\dot{0}^{*}$	$\dot{0}^{\nabla}$	+
$\dot{0}$	*3*	*2*	0^{∇}	
5	$\ddot{0}$	*–4*	*–2*	

$X_4 =$

				δ_i
		140	20	*0*
120			20	*0*
	50	50	50	*20*
0	*0*	*0*	*20*	

δ_j

1-й этап.

Отмечаем штрихом нуль (2, 4), он стоит в строке с положительной невязкой, и неполный несущественный. Поиск закончен успешно.

2-й этап.

Имеем цепочку и одного нуля, помещаем в позицию (2, 4) значение

$$\theta = \min\{\delta_2 = 20, \delta_4 = 40, r'_{2,4} = 50\} = 20 .$$

Получаем план X_4. Его суммарная невязка $\Delta = 40$. Итерации продолжаются.

5-я итерация.

Этап разметки.

$C_3 =$

(+)	+	(+)		
9	12	$\dot{0}^{*}$	$\dot{0}^{\nabla}$	+
$\dot{0}^{*}$	3	2	$\dot{0}^{\nabla}$	+
5	$\ddot{0}$	–4	–2	

$X_4 =$

				δ_i
		140	20	0
120			20	0
	50	50	50	20
0	0	0	20	δ_j

Нули (1, 3), (1, 4), (2, 1) и (2, 4) являются неполными, нуль (3, 2) – полным, выделяем столбцы с 1-го по 3-й.

1-й этап.

Нуль (1, 4) выделяем штрихом, а первую строку – плюсом. Нуль (1,3) выделяем звёздочкой, снимаем выделение с третьего столбца, продолжаем поиск. В третьем столбце нулей нет, а в четвёртом выделяем нуль (2, 4).

Так как невязка в торой сторки – нулевая, выделяем её плюсом. На её пересечении с 1-м столбцом стоит существенный нуль (2, 1), обозначаем позицию звёздочкой и снимаем выделение с первого столбца. Больше в невыделенной части матрицы (серая) нулевых элементов нет. Поиск закончен неудачно.

3-й этап.

Отрицательные элементы выделены однократно, единственный положительный элемент равен 5. Коррекция матрицы с его использованием даёт C_4.

$C_4 =$

	+			
9	*17*	$\dot{0}^{*}$	$\dot{0}^{\nabla}$	+
$\dot{0}^{*}\rightarrow$	*8*	*2*	$\rightarrow\dot{0}^{\nabla}$	+
$\uparrow 0^{\nabla}$	$\ddot{0}$	*–9*	*–7*	

$X_5 =$

				δ_i
		140	20	*0*
100			40	*0*
20	50	50	50	*20*
0	*0*	*0*	*20*	δ_j

1-й этап.

Отмечаем нуль (3, 1) штрихом. Невязка соответствующей строки – положительна, поисковый этап закончился положительно.

2-й этап.

Цепочка соединяет нули с координатами в матрице $C_3\ (3,\ 1)^{\nabla} \Rightarrow (2,\ 1)^{*} \Rightarrow (2,\ 4)^{\nabla}$.

Элемент для её коррекции есть

$$\theta = \min\left\{\delta_3 = 20, \delta_4 = 20, x^{*}_{1,3} = 120, r'_{3,1} = 50, r'_{2,4} = 30\right\} = 20.$$

Получаем план X_5, невязка которого $\Delta = 0$. Следовательно, полученный план – оптимальный.

Целевая функция, соответствующая этому плану

$$L = 140 \times 20 \times 2 + 100 \times 4 + 40 \times 8 + 20 \times 9 + 50 \times (2 + 3 + 6) = 1630.$$

По сравнению с целевой функцией плана перевозок, полученного без учёта ограничений, целевая функция имеет большее значение. Это объясняется тем, что “благодаря” ограничениям не удаётся переместить потребное число единиц товара машрутом с минимальной стоимостью.

2.3.4.8. Решение задачи о назначениях [22, 33, 34]

Задача о *назначениях*, она же задача *распределения* или задача *выбора*, имеет следующую содержательную постановку.

Предположим, что имеется n различных работ: $B_1, B_2, \ldots, B_n$ и столько же исполнителей этих работ (механизмов, например): $A_1, A_2, \ldots, A_n$. Причём, каждый из исполнителей способен выполнять любую работу, но единовременно может быть задействован только на одной из них. Пусть работа механизма A_i при выполнении работы B_j характеризуется некоторой неотрицательной величиной $c_{i,j}$, $i = 1, n, j = 1,n$.

Требуется таким образом распределить работы среди исполнителей, чтобы был достигнут определённый эффект (оптимум), например, достичь максимальной производительности труда или минимального расхода ресурсов.

Формально задача ставится в виде: для заданной матрицы

$$C = \begin{bmatrix} c_{1,1} & c_{1,2} & \dots & c_{1,n} \\ c_{2,1} & c_{2,2} & \dots & c_{2,n} \\ \dots & \dots & \dots & \dots \\ c_{i,1} & c_{i,2} & \dots & c_{1,n} \\ \dots & \dots & \dots & \dots \\ c_{n,1} & c_{2,n} & \dots & c_{n,n} \end{bmatrix}$$

найти распределение (расстановку, выбор и т.д.) $X = [x_{i,j}]$, $i = 1, n, j = 1, n$, обеспечивающие оптимум целевой функции

$$L = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{i,j} \cdot x_{i,j} \to opt\,. \tag{2.49}$$

Если матрица C представляет собой ***производительность*** механизмов при выполнении отдельных работ, то задача решается на ***максимум***, а если смысл матрицы C – ***стоимостные или временные издержки***, то задача решается на ***минимум***.

Формально можно полагать, что задачи максимизации и минимизации связаны между собой выражением.

$$c_{i,j}^{Издержки} = 1 \Big/ c_{i,j}^{Производительнось}\,. \tag{2.50}$$

Житейски (2.50) можно прокомментировать так, что кто “с огоньком” работает, тот более выгоден, или, если так можно выразиться, лучше окупается.

Попутно заметим, что если, вследствие применения (2.50) числа окажутся дробными, то матрицу C путем умножения на соответствующую степень десятки и отбрасыванием оставшейся дробной части, легко привести к целому виду для удобства расчётов.

Так как при этом (2.49), в силу линейности, возрастёт, то необходимо будет её после расчётов откорректировать делением.

Матрица решения X состоит, в основном, из нулей и содержит всего n единиц, размещающихся в тех позициях, в которые соответствуют назначению i-го исполнителя на j-ю работу. Поэтому, при решении задачи вручную, определение значения (2.49) состоит в суммировании тех элементов исходной матрицы C, которым соответствуют единицы в матрице X.

Методы решения задачи о назначениях

Задача о назначениях представляет собой частный случай транспортной задачи без ограничения на пропускную способность коммуникаций.

Частности состоят в следующем:

- число поставщиков равно числу потребителей, отчего матрица C – квадратная;
- так как имеется условие о выполнении одним исполнителем в текущий момент времени только одной работы, то вектора с объёмами производства и объёмами потребления следует сделать единичными.

Исходя из этих допущений, для решения задачи о назначениях пригоден любой метод решения ТЗ, учитывающий матрицу стоимостей.

Понятно, что появились методы, учитывающие особенности задачи о назначениях. Наиболее популярна из них – модификация венгерского метода.

Ниже нами будет рассмотрен, для разнообразия, алгоритм решения задачи ***максимизации***. В этом случае полагается, что элементы матрицы C есть производительность i-го исполнителя при постановке его на выполнение j-й работы. Метод использует понятие “независимого нуля”.

Независимым нулём называется нуль матрицы C, не содержащий в строке и в столбце, на пересечении которых он находится, других независимых нулей.

Алгоритм решения задачи о назначениях с максимизацией целевой функции

Алгоритм имеет классическую структуру, характерную для всех венгерских методов.

Состоит из предварительного этапа и трёхэтапной итерации, в начале которой выполняется разметка.

Предварительный этап.

Состоит в построении матрицы C_0 и расстановке независимых нулей.

- В каждом столбце матрицы C отыскивается максимальный элемент, из которого затем вычитаются все элементы этого столбца. Результат записывается на место вычитаемого. В результате, матрица C преобразуется в матрицу $C`$.
- В каждой строке матрицы $C`$ отыскивается минимальный элемент, который затем вычитается из всех элементов этой строки. В результате, матрица $C`$ преобразуется в матрицу C_0.

- Произвольно отмечаем нуль в первом столбце матрицы C_0 звёздочкой, полагая его независимым. Просматриваем по порядку остальные столбцы матрицы, руководствуясь определением. Если в рассматриваемых столбцах есть нулевой элемент, стоящий на строке, где нет нуля со звёздочкой, то его полагаем независимым, отмечаем звёздочкой и переходим к рассмотрению следующего столбца.

Заметим, что если бы задача решалась на минимум, то, в этом случае, построение матрицы C_0 не отличалось бы от обычного венгерского метода.

Проверка условия окончания

Подсчитываем число нулей со звёздочками (независимых нулей). Если их число равно размерности матриц n, то достигнут оптимум.

Необходимо рассчитать целевую функцию (2.49). План X, содержащий оптимальную расстановку, полностью определяется текущей матрицей C: нули, отмеченные звёздочками, соответствуют, единичным позициям оптимального плана, все прочие позиции – нулевые.

ИТЕРАЦИОННАЯ ЧАСТЬ АЛГОРИТМА

По завершении итерации, число независимых нулей в текущей матрице C увеличивается на один.

Таким образом, точное число итераций есть

$$N_{\text{итераций}} = n - n^*, \tag{2.51}$$

где n – размерность матрицы, n^* - число независимых нулей.

Возможное число итераций, даже в начале счёта не более $n - 2$.

Разметка.

Разметка текущей матрицы C выполняется в начале итерации и сохраняется до её конца с теми изменениями, которые вносятся в неё по мере выполнения алгоритма.

Необходимо выделить знаком «плюс» j-е столбцы матрицы C, в которых присутствуют независимые нули.

Этап 1 – этап поиска.

Область поиска: невыделенная часть матрицы C – невыделенные столбцы и строки.

Цель поиска: найти в невыделенной части матрицы C нуль, стоящий в строке, в которой нет независимого нуля.

Поисковый этап заканчивается, как и в любом алгоритме венгерского метода, одним из случаев:

- если все нули матрицы C находятся в выделенной части, то необходимо перейти к этапу 3 – эквивалентных преобразований матрицы C;

- поиск завершился успешно, найден ноль в строке, где нет других независимых нулей. В этом случае далее выполняется этап 2 – построение цепочки и коррекция плана назначений.

В ходе поиска невыделенная часть матрицы *C* просматривается по столбцам сверху вниз а столбцы – слева направо.

Пусть среди элементов найден ноль. Его отмечают апострофом (штрихом) и анализируют строку, на которой он находится.

Если нет других независимых нулей (со звёздами), то этот ноль со штрихом является искомым, а поиск заканчивается успешно.

Если в строке уже имеется независимый нуль, то текущая стока выделяется знаком «плюс», и просматривается по местам её пересечения с выделенными столбцами. Если в месте пересечения стоит нуль со звёздочкой (*), то знак выделения над столбцом уничтожают, обводя кружком или заключая в скобки. Столбец становится невыделенным и делается доступным для поиска. Поиск далее продолжают по этому столбцу со снятым выделением.

Этап 2 – этап построения цепочки и коррекции плана назначений.

1. Цепочка ***не замкнута***, составляется из нулей со штрихом (0’) и нулей со звёздочками (0^*), содержит нечётное число элементов, и, в принципе, может состоять и из одного нуля со штрихом.

2. Цепочка начинается от последнего найденного нуля со штрихом 0’ к нулю со звездой 0^* по столбцу ($0' \to 0^*$), далее, по направлению под 90° к предыдущему, по строке от нуля со звездой к нулю со штрихом ($0^* \to 0'$) и так далее. На нечётных местах цепочки будут стоять нули со штрихом, а на чётных – нули со звёздами. Цепочка начинается в строке, в которой нет независимых нулей, и заканчивается в столбце, который в ходе разметки избежал выделения.

При переписи матрицы *C* апострофы (штрихи) возле нулей, составляющих цепочку, заменяются звёздами, а звёздочку – устраняют. В результате число независимых нулей увеличивается на единицу.

Этап 3 – этап эквивалентного преобразования матрицы C.

Этот этап ничем не отличается от изложенного ранее, приводится здесь для полноты восприятия.

1. Среди невыделенных элементов текущей матрицы *C* выбирается минимальный положительный элемент $h > 0$. Этот элемент называется ***корректирующим***.

2. Корректирующий элемент вычитается от невыделенных строк матрицы *C*.

3. Корректирующий элемент прибавляется к выделенным столбцам матрицы *C*.

Пример. Решить задачу о назначениях на максимум.

$C =$

9	7	8	6	3	9
2	8	6	8	5	4
4	2	7	7	7	5
3	5	5	2	6	4
6	4	4	5	6	8
5	6	6	3	4	7

Предварительный этап.

Максимальные элементы в столбцах C выделены серым. В результате преобразования исходной матрицы по столбцам имеем

$C' =$

0	1	0	2	4	0
7	0	2	0	2	5
5	6	1	1	0	4
6	3	5	6	1	5
3	4	4	3	1	1
4	2	2	5	3	2

Минимальные элементы в строках C' выделены серым. После вычитания получаем C_0 вида

$C_0 =$

(+)	(+)	(+)		+	+	
0^{*}	1	0^{∇}	2	4	0	+
7	0^{*}	2	0^{∇}	2	5	+
5	6	1	1	0^{*}	4	
5	2	4	5	0	4	
2	3	3	2	0	0^{*}	
2	0^{∇}	0^{*}	3	1	0	+

Расставляем звёзды при независимых нулях: (1, 1),. (2, 2), (6, 3), (3, 5) и (5, 6).

Проверка условия окончания.

Число независимых нулей равно пяти, следовательно, задача решится за одну итерацию.

Итерация.

Разметка.

Размечаем 1, 2, 3, 5, 6 столбцы плюсами.

1-й этап, поиск.

Отмечаем нуль (2, 4) штрихом. На этой строке – независимый ноль, выделяем её, а выделение со второго столбца снимаем.

Во втором столбце отмечаем нуль (6, 2) штрихом, и, так как в шестой строке стоит независимый нуль, выделяем и её. После чего, снимаем выделение с третьего столбца, и отмечаем нуль (1, 3).

Первая строка выделяется плюсом, а выделение с первого столбца снимается. Поиск завершился неудачно.

3-й этап.

В невыделенной части матрицы (серый фон) находим корректирующий элемент. Он равен единице (таких элементов в матрице целых два). Корректируем матрицу C_0, получаем C_1.

$C_1=$

				(+)	+	
0^*	1	0^∇	2	5	1	+
7	$0^*\rightarrow$	2	$\rightarrow 0^\nabla$	3	6	+
4	5	$\downarrow 0^\nabla$	0	$\leftarrow 0^*$	4	+
4	1	1	4	$\uparrow 0^\nabla$	4	
1	2	2	1	0	0^*	
2	$\uparrow 0^\nabla$	$\leftarrow 0^*$	3	2	1	+

1-й этап, снова поиск.

Нуль (3, 3) отмечаем штрихом, третья строка отмечается плюсом, и снимается выделение с пятого столбца.

В четвёртой строке отмечаем нуль (4, 5), который оказывается искомым. Этап удачно завершён.

2-й этап, коррекция плана.

Строим цепочку (4, 5) → (3, 5) → (3, 3) → (6, 3) → (6, 2) → (2, 2) → (2, 4). Проводимые согласно индексации замены даю следующую топологию размещения независимых нулей.

$C_1=$

0^*	1	0	2	5	1
7	0	2	0^*	3	6
4	5	0^*	0	0	4
4	1	1	4	0^*	4
1	2	2	1	0	0^*
2	0^*	0	3	2	1

Число независимых нулей равно шести, расчёты закончены. Имеем следующий план расстановки исполнителей по работам:

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Целевая функция полученного плана $L = 9 + 8 + 7 + 6 + 8 + 6 = 44$.

Решение предписывает, что 1-й исполнитель задействован на выполнении 1-й работы, 2-й – на 4-й, 3-й – на 3-й, 4-й – на 5-й, 5-й – на 6-й, а 6-й – на 2-й.

2.3.4.9. Вопросы для самоконтроля

1. Каковы недостатки алгоритма метода северо-западного угла?
2. В чём привлекательность метода минимальной стоимости?
3. Почему метод штрафов лучше метода минимальной стоимости?
4. Какой опорный план называется вырожденным и почему?
5. Как связаны между собой ЗЛП и ТЗ?
6. Что такое потенциал, и в чём его практический смысл?
7. Какова геометрическая интерпретация опорного плана *Т*-задачи?
8. В чём состоит понятие баланса в *Т*-задачах?
9. Для чего нужны цепочки при коррекции планов?
10. В чём заключается физический смысл построения цепочки?
11. Почему цепочки метода потенциалов замкнутые?
12. Каковы приёмы, применяемые для поддержания невырожденности плана?
13. В чём заключаются трудности, связанные с реализацией метода потенциалов на ЭВМ?
14. Какая транспортная модель называется замкнутой?
15. Для чего проверяется условие баланса и выполняется балансировка?
16. Что такое невязка?
17. Почему для венгерского метода нет необходимости в использовании опорного плана?
18. Что такое существенные и несущественные нули?
19. Как прогнозируется число итераций, оставшихся до получения оптимального решения?

20. Что является целью этапа поиска венгерского метода?

21. Где выполняется поиск?

22. В чём состоит назначение и сущность этапа эквивалентных преобразований матрицы *C*?

23. В чем сходство и различие при построении начальных планов для решения задачи венгерским методом и методом потенциалов?

24. Каковы, на Ваш взгляд, преимущества и недостатки обоих методов решения ТЗ? Обоснуйте высказываемые суждения.

25. Почему венгерский метод удобен при реализации на ЭВМ?

26. Поясните физический смысл, заключенный в процедуре построения цепочки.

27. Почему цепочка, которая строится в алгоритме метода потенциалов, замкнута, а в венгерском методе нет?

28. Каковы признаки неразрешимости *Td*-задачи?

29. Обоснуйте, почему *Td*-задачу трудно или невозможно решить методом потенциалов?

30. С чем связано увеличение целевой функции *Td*-задачи по сравнению с обычной *T*-задачей?

31. В чём состоят особенности применения алгоритма венгерского метода при наличии ограничений?

32. Приведите постановки *T*-задач применительно к технике передачи информации и компьютерным сетям.

34. Почему *Td*-задача не всегда может быть решена?

35. Как задается условие дискретности в задаче о назначениях?

36. Как связаны между собой задача о назначениях и Т-задача?

37. Всегда ли разрешима задача данного типа? Ответ обосновать.

38. Какой ноль называется независимым?

39. Какой алгоритм из известных алгоритмов решения транспортных задач кажется Вам наиболее эффективным? Дать обоснование.

40. Что будет, если наложить условие на допуск исполнителей к определенным работам?

41. Что, на Ваш взгляд, поменяется в постановке и алгоритме задачи, если каждый исполнитель будет в состоянии выполнять более одной работы, а каждая работа – быть исполнимой более чем одним исполнителем?

42. Как оценить число итераций, оставшихся до получения оптимального решения?

2.4. Решение задач параметрического программирования

В общем случае, к задачам параметрического программирования относятся задачи математического программирования, в которых компоненты математических моделей зависят от некого фактора (параметра).

Указанный фактор, в свою очередь, изменяется в соответствии с некоторым законом, однозначно определяемым математически. Наиболее целесообразно исследовать влияние на оптимальное решение задачи линейного программирования (ЗЛП) параметрических изменений компонентов векторов коэффициентов функции цели или свободных членов системы ограничений. Существуют задачи, в которых от параметра зависит как коэффициенты функции цели, так и свободные члены системы ограничений [3, 7].

Элементы матрицы системы ограничений в задачах экономико-производственной направленности, по сути, являются технологическими коэффициентами (нормами выхода), поэтому в наименьшей степени зависят от параметров.

Общая стратегия при решении задач параметрического программирования такова.

1. Полагают значение параметра равным некоторому неизвестному числу t_0, принадлежащему интервалу изменения параметра t, определяемого в виде границ $[\alpha, \beta]$, как правило, это α.
2. Выполняют решение ЗЛП в общем виде, с неизвестной, входящей в выражение для расчёта симплекс-разностей или (и) в компоненты столбца местоположения координат текущей точки решения A_0. Находят оптимальный план X^* или устанавливают неразрешимость полученной задачи линейного программирования в заданной постановке.
3. По строке симплекс-разностей или (и) элементам вектора A_0, соответствующим оптимальному решению, полученному в п.2 в ходе решения ЗЛП, составляют систему неравенств.
4. В результате решения системы неравенств определяют некое значение параметра t, удовлетворяющее этой системе. Это значение параметра t полагают равным верхней границей текущего интервала.
5. Если значение параметра, рассчитанное в ходе выполнения п. 4 настоящего алгоритма, превышает значение, заданное в условии, то задачу считают решённой.
6. В противном случае, необходимо определить границу очередного интервала, находящуюся правее значения параметра, определённого

в п.4. Производится “дорешивание” ЗЛП с новым значением параметра t в п.2.

7. Итоговое решение представляется совокупностью интервалов, на каждом из которых представлены свои координаты оптимальных точек совокупно с выражением для расчёта функции цели на заданном интервале

2.4.1. Решение задачи линейного программирования при параметрических изменениях вектора ограничений [3]

Пусть элементы вектора ограничений находятся в линейной зависимости от изменения параметра t, то есть

$$b_i^{\nabla} = b_i + tp_i,\ i = \overline{i,m}, \qquad t \in [\alpha, \beta], \tag{2.52}$$

где b_i – значение i-го элемента вектора ограничений в отсутствии параметра; p_i – коэффициент параметрического изменения по i-му ограничению; t – параметр.

В этом случае задача линейного программирования может быть сформулирована в виде

$$\begin{aligned} &\max F = c_1x_1 + c_2x_2 + \ldots + c_nx_n, \\ &\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \le b_1 + tp_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \le b_2 + tp_2, \\ \ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \le b_m + tp_m, \\ x_j \ge 0,\ j = \overline{1,n},\ t \in [\alpha, \beta]. \end{cases} \end{aligned} \tag{2.53}$$

Если зафиксировать произвольное значение параметра $t = t_0$, то задача (2.53) сведётся к обычной ЗЛП и может быть решена любым пригодным симплекс-методом. Как известно:

$$X^* = A^{-1} \times B, \tag{2.54}$$

где X^* – оптимальный план; A^{-1} – компоненты симплекс-таблицы для небазисных переменных; B – вектор свободных членов системы

ограничений. Причём равенство (2.54) является справедливым и на протяжении всего решения

Выражение (2.54) с учётом (2.52) примет вид

$$X^{\nabla *} = A^{-1} \cdot B^{\nabla} = A^{-1} \cdot B + t_0 A^{-1} \cdot P,$$

где $t_0 \in [\alpha, \beta]$ – фиксированное значение параметра t, $P^T = \{p_1, p_2, \ldots, p_m\}$ – вектор коэффициентов изменения параметра. Введя обозначение

$$\widetilde{B} = A^{-1} \cdot B \text{ и } \widetilde{P} = A^{-1} \cdot P, \qquad (2.55)$$

запишем

$$X^{\nabla *} = \widetilde{B} + t_0 \cdot \widetilde{P}. \qquad (2.56)$$

Таким образом, при фиксированном $t = t_0$, столбец A_0 может быть представлен тремя столбцами $\widetilde{B}$, $\widetilde{P}$ и $X^{\nabla *}(A_0^{\nabla})$, что позволит осуществить "надзор" за преобразованиями величин $\widetilde{b}_i$ и $\widetilde{p}_i$ в ходе итераций.

Компоненты вектора $X^{\nabla *}$ должны удовлетворять ограничению неотрицательности, то есть

$$\widetilde{b}_i + t \cdot \widetilde{p}_i \geq 0, \;\; i = \overline{i,m}, \qquad t \in [\alpha, \beta]. \qquad (2.57)$$

Безусловно, выполнение (2.57) означало бы, что план $X^{\nabla *}$ является оптимальным решением задачи параметрического программирования вида (2.53) для любых $t \in [\alpha, \beta]$.

Однако, в реальности встаёт задача последовательного деления заданного интервала изменения параметра t $[\alpha, \beta]$ на подынтервалы $[\alpha_r, \beta_r] \subset [\alpha, \beta]$, внутри каждого из которых неукоснительно выполняется условие неотрицательности $X^{\nabla *}$.

При этом, на каждом подынтервале $[\alpha_r, \beta_r] \subset [\alpha, \beta]$ для любого $t_0 \in [\alpha_r, \beta_r]$ оптимум целевой функции будет достигаться в вершинах многогранников областей допустимых стратегий (планов) задачи, и, каждому подынтервалу изменения параметра Δt_r, будет соответствует свой оптимальный план.

Поэтому представляет интерес определение границ изменения параметра t, для которого вектор $X^{\nabla *}$ будет являться оптимальным планом.

В оптимуме параметрической задачи обязательно должен будет присутствовать $\widetilde{p}_i \neq 0$, иначе параметрическая задача вырождается – пропадает зависимость от параметра.

Сформулируем правила для определения границ интервала изменения параметров, которые есть следствие решения системы неравенств (2.57):
для нижней границы

$$\alpha_{or} = \begin{cases} \max\limits_{\widetilde{p}_i > 0} \left\{ -\dfrac{\widetilde{b}_i}{\widetilde{p}_i} \right\}, & \exists i : \widetilde{p}_i > 0, \\ -\infty, & \forall i : \widetilde{p}_i \le 0 \end{cases} \qquad (2.58)$$

для верхней границы

$$\beta_{or} = \begin{cases} \min\limits_{\widetilde{p}_i < 0} \left\{ -\dfrac{\widetilde{b}_i}{\widetilde{p}_i} \right\}, & \exists i : \widetilde{p}_i < 0, \\ \infty, & \forall i : \widetilde{p}_i \ge 0. \end{cases} \qquad (2.59)$$

Индексы *or* в формулах (2.57) и (2.58) обозначают: о – оптимальное значение, *r* – номер интервала разбиения. В пределах этих границ решение устойчиво, то есть, сохраняется неизменным состав векторов базиса.

Алгоритм решения задачи параметрического программирования при параметрическом изменении элементов вектора свободных членов

1. Зафиксировав значение параметра $t = t_0$, решить ЗЛП симплекс-методом.

Замечание. Если имеется решение непараметрической ЗЛП, то это соответствует случаю, когда $t = 0$. В этом случае, решение ЗЛП для *t*, отличного от нуля, можно получить, используя матричные операции (2.54) и (2.55), избежав, тем самым, утомительных вычислений. **Важно**: так как обратная матрица A^{-1} зависит от выбранных базисных векторов, а состав базиса, в свою очередь – от значения параметра *t*, использование такого подхода требует осмысления и вдумчивости!

Задача может быть разрешима, а может быть, и нет [3]. В последнем случае неразрешимость может иметь двоякую природу.

Во-первых, целевая функция задачи не ограничена на области допустимых стратегий. В этой ситуации задача не решаема принципиально ни при каких значениях *t*.

Во-вторых, не обеспечивается положительность базисного решения (содержимого столбца A_0) при данном значении параметра *t*, что, как

правило, говорит о несовместности системы ограничений. Поэтому надо найти такой диапазон изменения параметра t, в котором система ограничений совместна.

В последнем случае отметим, что нарушения условия неотрицательности и есть причина несовместности ограничений.

2. Определить, используя формулы (2.58) и (2.59), границы изменения параметра t $[\alpha_{or}, \beta_{or}]$, r – номер интервала.

3. Если верхняя граница β_{or} интервала выйдет за пределы или окажется равной β, то есть, $\beta_{or} \geq \beta$, то задача решена.

4. В противном случае, необходимо отыскать направляющую строку по правилу

$$\min_{\widetilde{p}_i < 0} \left\{ -\frac{\widetilde{b}_i}{\widetilde{p}_i} \right\} \Rightarrow i^* . \qquad (2.60)$$

и выполнить несколько итераций двойственного симплекс-метода.

5. Вернуться к п.2 настоящего алгоритма.

В результате расчётов мы будем иметь ряд интервалов $[\alpha, \beta_1]$, $[\beta_1, \beta_2]$, … , $[\beta_{r-1}, \beta]$ на каждом из которых будем иметь свой оптимальный план решения. Поэтому целесообразно полагать в начале расчётов $t = t_0 = \alpha$.

Пример. Пусть модель параметрической ЗЛП имеет вид:

$$F(x_1, x_2) = 2x_1 + 3x_2 \to \max$$
$$\begin{cases} 2x_1 + 5x_2 \leq 10 + 8t, \\ 3x_1 + 2x_2 \leq 6 + 4t, \\ 1x_1 + 1x_2 \leq 4 + 2t, \\ x_1 \geq 0, x_2 \geq 0,\ t \in \left[-\frac{1}{2}; 2 \right]. \end{cases}$$

Из условия задачи следует, согласно (2.53), что

$$B = \begin{bmatrix} 10 \\ 6 \\ 4 \end{bmatrix}, \qquad P = \begin{bmatrix} 8 \\ 4 \\ 2 \end{bmatrix},\ \alpha = -\frac{1}{2},\ \beta = 2.$$

Пункт № 1 алгоритма.

По условиям задачи составим таблицу для значения параметра $t = -1/2$ и выполним её решение.

Таблица 2.1 – Начальное решение в канонической форме

				c_j	2	3	0	0	0
Базис	$C_Б$	$\widetilde{B}$	$\widetilde{P}$	$X^{\nabla *}$	A_1	A_2	A_3	A_4	A_5
$\leftarrow A_3$	0	10	8	6	2	5	1	0	0
A_4	0	6	4	4	3	2	0	1	0
A_5	0	4	2	3	1	1	0	0	1
			δ_j	0	-2	$-3\uparrow$	0	0	0

Таблица 2.2 – После 1-й итерации

				c_j	2	3	0	0	0
Базис	$C_Б$	$\widetilde{B}$	$\widetilde{P}$	$X^{\nabla *}$	A_1	A_2	A_3	A_4	A_5
A_2	3	2	8/5	6/5	2/5	1	1/5	0	0
$\leftarrow A_4$	0	2	4/5	8/5	11/5	0	$-2/5$	1	0
A_5	0	2	2/5	9/5	3/5	0	$-1/5$	0	1
			δ_j	18/5	$-4/5\uparrow$	0	3/5	0	0

Таблица 2.3 – Оптимальное значение при $t = -1/2$

				c_j	2	3	0	0	0
Базис	$C_Б$	$\widetilde{B}$	$\widetilde{P}$	$X^{\nabla *}$	A_1	A_2	A_3	A_4	A_5
A_2	3	18/11	16/11	10/11	0	1	3/11	$-2/11$	0
A_1	2	10/11	4/11	8/11	1	0	$-2/11$	5/11	0
A_5	0	16/11	2/11	15/11	0	0	$-1/11$	$-3/11$	1
			δ_j	46/11	0	0	5/11	4/11	0

В последней таблице представлено оптимальное решение задачи параметрического программирования $F\left(\frac{10}{11}, \frac{8}{11}\right) = \frac{46}{11}$ для $t = -1/2$.

В соответствии с замечанием к 1-му пункту алгоритма, этот же результат может быть получен по оптимальному решению задачи без параметров, что соответствует значению параметра $t = 0$.

Пункт № 2 алгоритма.

Нижняя граница изменения параметра t, согласно (2.58), есть

$$\alpha_{o1} = \max_{\widetilde{p}_i > 0}\left\{-\frac{\widetilde{b}_i}{\widetilde{p}_i}\right\} = \left\{\frac{16}{11},\frac{4}{11},\frac{2}{11}\right\}\left\{-\frac{\frac{18}{11}}{\frac{16}{11}},-\frac{\frac{10}{11}}{\frac{4}{11}},-\frac{\frac{16}{11}}{\frac{2}{11}}\right\} = -\frac{9}{8} = -1{,}125\,.$$

Так как $-\frac{9}{8} < -\frac{1}{2}$, то полученная нижняя граница устойчивости решения не хуже, чем задана в условии исходной задачи.

Верхняя же граница интервала существования неизменного базиса (A_1, A_2, A_5), определяемая (2.59), равна бесконечности, поскольку все компоненты вектора $\widetilde{P}$ положительны, это свидетельствует о том, что $\beta_{o1} \to \infty$.

Пункт алгоритма № 3.

Значение верхней границы интервала β_{o1} превышает заданную в условии задачи величину β = 2. Решение задачи параметрического программирования завершено, и нами получено окончательное решение. Ответ представляется одним интервалом изменения параметра *t*.

$$x_1 = \frac{10 + 4\cdot t}{11}, x_2 = \frac{18 + 16\cdot t}{11}; F(x_1, x_2) = \frac{74 + 56\cdot t}{11}, t \in \left[-\frac{9}{8}, \infty\right).$$

Замечание. Используем значение $t = \alpha_{o1} = -9/8$, полученное нами ранее, которое меньшее, нежели заданное в условии задачи $\alpha = -1/2$.

Согласно (2.56), в этом случае точка оптимального решения будет иметь координаты

$$X^{\nabla *} = \widetilde{B} + t_0 \cdot \widetilde{P} = \begin{bmatrix} 18/11 \\ 10/11 \\ 16/11 \end{bmatrix} - \frac{9}{8}\cdot \begin{bmatrix} 16/11 \\ 4/11 \\ 2/11 \end{bmatrix} = \begin{bmatrix} 0 \\ 1/2 \\ 5/4 \end{bmatrix}, \text{откуда}$$

$$X^T_{\text{opt}} = \{1/2 \quad 0 \quad 0 \quad 0 \quad 5/4\}.$$

Функция цели для этого решения составит $F(1/2, 0) = 1$. Мы можем убедиться, что структура решения сохранена, хотя план вырожден, так как компонента нулевая. Однако значение функции цели получилось меньше, нежели на интервале 1. Отсюда следует, что использовать значение параметра, ниже заданного в условии задачи, в нашем случае не целесообразно.

2.4.2. Решение задачи линейного программирования при вариации коэффициентов целевой функции [3, 7]

Пусть коэффициенты целевой функции зависят от параметра t следующим образом

$$c_i^{\nabla} = c_i + tp_i,\ i = \overline{i,n}, \qquad t \in [\alpha, \beta]. \tag{2.61}$$

В этом случае математическая модель задачи параметрического программирования представима так:

$$\max F = (c_1 + tp_1)x_1 + (c_2 + tp_2)x_2 + \ldots + (c_n + tp_n)x_n,$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \le b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \le b_2, \\ \ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \le b_m, \\ x_j \ge 0,\ j = \overline{1,n},\ t \in [\alpha, \beta]. \end{cases}$$

В ходе решения ЗЛП при фиксированном значении $t = t_0 \in [\alpha, \beta]$, будет получена таблица оптимального решения, значения строки симплекс-разностей для основных переменных, композиционно может быть представлено в виде

$$\delta_j^{\nabla} = \widetilde{\delta}_j + t_0 \cdot \widetilde{p}_j, \qquad j = \overline{1,n}, \tag{2.62}$$

где $\widetilde{\delta}_j$ – значение j-й симплекс-разности без учёта параметра; $\widetilde{p}_j$ – параметрическая составляющая симплекс-разности.

Для того, чтобы компоненты строки симплекс-разностей (2.62) указывали на достижение оптимального решения, необходимо выполнение условия неотрицательности

$$\delta_j^{\nabla} = \widetilde{\delta}_j + t_0 \cdot \widetilde{p}_j \ge 0, \qquad j = \overline{1,n}, \tag{2.63}$$

Выполнение системы (2.63) находятся в зависимости от знаков величин $\widetilde{\delta}_j$ и $\widetilde{p}_j$. Если имеются $\widetilde{\delta}_j \ge 0$, то нижняя граница параметра

$$\alpha_{or} \geq -\frac{\widetilde{\delta}_j}{\widetilde{p}_j} \Rightarrow \alpha_{or} = \max_{\widetilde{p}_j > 0} \left\{ -\frac{\widetilde{\delta}_j}{\widetilde{p}_j} \right\}, \tag{2.64}$$

$\alpha_{or} \leq \alpha$. Если отсутствуют $\widetilde{p}_j > 0$, то есть все $\widetilde{p}_j \leq 0$, **отсутствует и нижняя граница**: $\alpha_{or} \to -\infty$.

Когда имеются $\widetilde{p}_j < 0$, то можно определить верхнюю границу подынтервала

$$\beta_{or} \leq -\frac{\widetilde{\delta}_j}{\widetilde{p}_j} \Rightarrow \beta_{or} = \min_{\widetilde{p}_j < 0} \left\{ -\frac{\widetilde{\delta}_j}{\widetilde{p}_j} \right\}, \tag{2.65}$$

которая получается $\beta \leq \beta_{or}$. При отсутствии $\widetilde{p}_j < 0$, то есть, когда все $\widetilde{p}_j \geq 0$, **отсутствует верхняя граница интервала** $\beta_0 \to \infty$.

Алгоритм решения задачи параметрического программирования

1. Зафиксировать $t = t_0 \in [\alpha, \beta]$ и решить задачу параметрического программирования как обычную ЗЛП.

Дополнительные строки симплекс-таблицы ($m + 1$), ($m + 2$) и ($m + 3$), соответствующие $\widetilde{\delta}_j$, $\widetilde{p}_j$ и δ_j^{∇}, будем рассчитывать в ходе решения, согласуя с (2.62).

Пусть задача при неком значении параметра t не разрешима, то есть, существуют признаки неограниченности линейной формы $\delta_k^{\nabla} < 0$, $a_{i,k} \leq 0$. Если

- $\widetilde{p}_k = 0$, то задача неразрешима для любого t;
- $\widetilde{p}_k < 0$, то задача неразрешима для всех $t \leq \beta_{or}$.
- $\widetilde{p}_k > 0$, то задача неразрешима для всех $t > \alpha_{or}$.

2. Исследовать полученное решение с помощью системы неравенств допустимости (2.63), в ходе которого будут определены нижняя (2.64) и верхняя (2.65) границы параметра.

3. Если окажется, что $\beta_{or} \to \infty$, то найдено оптимальное решение, и задача параметрического программирования решена.

4. В противном случае, необходимо определить направляющий столбец j^* по минимальному значению отношения

$$\min_{\widetilde{p}_j < 0} \left\{ -\frac{\widetilde{\delta}_j}{\widetilde{p}_j} \right\} \Rightarrow j^*, \qquad (2.66)$$

и осуществить ряд итераций прямого симплекс-метода, после чего перейти к п. 2 настоящего алгоритма.

5. Решение представляется набором интервалов $[\alpha, \beta_1]$, $[\beta_1, \beta_2]$, … , $[\beta_{r-1}, \beta]$, на каждом из которых будет свой оптимальный план и своё выражение для расчёта целевой функции.

Пример.

$$F(x_1, x_2) = (2 + 2 \cdot t)x_1 + (3 - 1 \cdot t)x_2 \to \max$$
$$\begin{cases} 2x_1 + 5x_2 \le 10, \\ 3x_1 + 2x_2 \le 6, \\ 1x_1 + 1x_2 \le 4, \\ x_1 \ge 0, x_2 \ge 0, \ t \in \left[-\frac{1}{2}; 2 \right]. \end{cases}$$

Пункт № 1 алгоритма.

Положим $t = -1/2$, и решим ЗЛП прямым симплекс-методом.

Таблица 2.9 – Начальная таблица для $t = -½$

		c_j	$2 + 2t = 1$	$3 - 1t = \frac{7}{2}$	0	0	0
Базис	$C_Б$	A_0	A_1	A_2	A_3	A_4	A_5
←A_3	0	10	2	5	1	0	0
A_4	0	6	3	2	0	1	0
A_5	0	4	1	1	0	0	1
$m + 1$	$\widetilde{\delta}_j$	0	– 2	– 3	0	0	0
$m + 2$	$\widetilde{p}_j$	0	– 2	1	0	0	0
$m + 3$	δ_j^{∇}	0	$-2 - 2t = -1$	$-3 + 1t = -\frac{7}{2}$ ↑	0	0	0

Таблица 2.10 – Таблица оптимального решения для $t = -½$

		c_j	$2+2t=$ $=1$	$3-1t=$ $=\frac{7}{2}$	0	0	0
Базис	$C_Б$	A_0	A_1	A_2	A_3	A_4	A_5
A_2	$3-1t=$ $=\frac{7}{2}$	2	$2/5$	1	$1/5$	0	0
$\leftarrow A_4$	0	2	$11/5$	0	$-2/5$	1	0
A_5	0	2	$3/5$	0	$-1/5$	0	1
$m+1$	$\tilde{\delta}_j$	6	$-4/5$	0	$3/5$	0	0
$m+2$	$\tilde{p}_j$	–2	$-12/5$	0	$-1/5$		
$m+3$	δ_j^{∇}	$6-2\cdot t=$ $=7$	$-\frac{4}{5}-\frac{12}{5}t=$ $=\frac{2}{5}$ $\uparrow$	0	$\frac{3}{5}-\frac{1}{5}t=$ $=\frac{7}{10}$	0	0

Строка $m + 3$ указывает на достижение оптимального решения, для значения $t = – 1/2$ задача разрешима. Координаты оптимума суть $X^*(0, 2)$, а целевая функция имеет бесчисленное множество значений в функции от параметра:

$$F(0, 2, t) = (3 - 1\cdot t) \times 2.$$

Пункт № 2 алгоритма.

Поскольку в $m + 2$ отсутствуют положительные числа, то это говорит о том, что отсутствует нижняя граница параметра α. Верхняя граница интервала, по условию (2.65) составит

$$\beta_{o1} = \min_{\tilde{p}_j < 0}\left\{-\frac{\tilde{\delta}_j}{\tilde{p}_j}\right\} = \left\{-\frac{12}{5}, -\frac{1}{5}\right\}\left\{\frac{-\frac{4}{5}}{-\frac{12}{5}}, \frac{-\frac{3}{5}}{-\frac{1}{5}}\right\} = \left\{-\frac{1}{3}, 3\right\} = -\frac{1}{3}.$$

Пункт № 3 алгоритма.

Поскольку значение β_{o1} не превышает верхнюю границу, заданную в условии, то задача не решена, и вычисления продолжаются.

Пункт № 4 алгоритма.

Направляющий столбец, определяемый по (2.66), соответствует вектору A_1(отмечен стрелкой в последней таблице 2.10) и верхней границе первого подынтервала β_{o1}.

Поясним логику данного пункта алгоритма следующими рассуждениями. Оптимальное решение будет сохранять свою структуру в смысле набора базисных векторов, если выполняются (2.63). То есть, необходима реализация двух условий неотрицательности:

$$\begin{cases} -\frac{4}{5} - \frac{12}{5} \cdot t \geq 0, \\ \frac{3}{5} - \frac{1}{5} \cdot t \geq 0. \end{cases} \Rightarrow \begin{cases} t \leq -\frac{1}{3}, \\ t \leq 3. \end{cases}$$

Если окажется, что $t > -\frac{1}{3}$, то вектор A_1 будет, согласно алгоритму симплексного метода, вводиться в базис. В результате итерации симплекс-метода получим таблицу.

Таблица 1.11 – Таблица оптимального решения для $t = -1/3$

		c_j	$2+2t=$ $=\frac{4}{3}$	$3-1t=$ $=\frac{10}{3}$	0	0	0
Базис	$C_Б$	A_0	A_1	A_2	A_3	A_4	A_5
$\leftarrow A_2$	$3-1t=$ $=\frac{10}{3}$	18/11	0	1	3/11	$-2/11$	0
A_1	$2+2t=$ $=\frac{4}{3}$	10/11	1	0	$-2/11$	5/11	0
A_5	0	16/11	0	0	$-1/11$	$-3/11$	1
$m+1$	$\widetilde{\delta}_j$	74/11	0	0	5/11	4/11	0
$m+2$	$\widetilde{p}_j$	2/11	0	0	$-7/11$	12/11	
$m+3$	δ_j^{∇}	$\frac{74}{11}+\frac{2}{11}t=$ $=\frac{20}{3}$	0	0	$\frac{5}{11}-\frac{7}{11}t=$ $=\frac{2}{3}$ $\uparrow$	$\frac{4}{11}+\frac{12}{11}t=$ $=0$	0

Координаты точки оптимума $X^*\left(\frac{10}{11},\frac{18}{11}\right)$, а выражение для расчёта функции цели выглядит следующим образом $F\left(\frac{10}{11},\frac{18}{11},t\right)=\frac{74}{11}+\frac{2}{11}t$. Для проверки правильности расчетов необходимо убедиться, что значения целевых функций для предыдущей и последующей таблиц оптимального решения в точке со значением параметра $t = -1/3$, совпадают:

$$F\left(0,2,-\frac{1}{3}\right)=\left(3+1\cdot\frac{1}{3}\right)\cdot 2=\frac{20}{3},$$

$$F\left(\frac{10}{11},\frac{18}{11},-\frac{1}{3}\right)=\frac{74-2\cdot\frac{1}{3}}{11}=\frac{222-2}{11\cdot 3}=\frac{20}{3}.$$

Пункт № 2 алгоритма.

Система неравенств, определяющая неизменность состава базиса, полученная по (2.63) выглядит так:

$$\begin{cases}\frac{5}{11}-\frac{7}{11}\cdot t\ge 0,\\ \frac{4}{11}+\frac{12}{11}\cdot t\ge 0.\end{cases}\Rightarrow\begin{cases}t\le\frac{5}{7},\\ t\ge-\frac{1}{3}.\end{cases}$$

Эта система показывает верхнюю и нижнюю границу интервала изменения t, что полностью согласуется с условиями (2.64)

$$\alpha_{o2}=\max_{\widetilde{p}_j>0}\left\{-\frac{\widetilde{\delta}_j}{\widetilde{p}_j}\right\}=\max\left\{\frac{12}{11}\right\}\left\{-\frac{\frac{4}{11}}{\frac{12}{11}}\right\}=-\frac{1}{3}.$$

и (2.65)

$$\beta_{o2}=\min_{\widetilde{p}_j<0}\left\{-\frac{\widetilde{\delta}_j}{\widetilde{p}_j}\right\}=\min\left\{-\frac{7}{11}\right\}\left\{-\frac{\frac{5}{11}}{-\frac{7}{11}}\right\}=\frac{5}{7}.$$

Пункт № 3 алгоритма.

Поскольку значение β_{o2}, не превышает верхнюю границу равную двум, заданную в условии, то задача не решена, и вычисления продолжаются.

Пункт № 4 алгоритма.

Полагаем значение t = 5/7, направляющий столбец A_3, этот вектор будет в базис вводиться, а A_3 – выводиться из базиса. После расчётов получим таблицу.

Таблица 1.12 – Таблица оптимального решения для $t = 5/7$

		c_j	$2+2t=$ $=\frac{24}{7}$	$3-1t=$ $=\frac{16}{7}$	0	0	0
Базис	$C_Б$	A_0	A_1	A_2	A_3	A_4	A_5
A_3	0	6	0	$11/3$	1	$-2/3$	0
A_1	$2+2t=$ $=\frac{24}{7}$	2	1	$2/3$	0	$1/3$	0
A_5	0	2	0	$1/3$	0	$-1/3$	1
$m+1$	$\widetilde{\delta}_j$	4	0	$-5/3$	0	$2/3$	0
$m+2$	$\widetilde{p}_j$	4	0	$7/3$	0	$2/3$	
$m+3$	δ_j^{∇}	$4+4t=$ $=\frac{48}{7}$	0	$-\frac{5}{3}+\frac{7}{3}t=$ $=0$	0	$\frac{2}{3}+\frac{2}{3}t=$ $=\frac{16}{21}$	0

Оптимум $X^*(2, 0)$, при этом функция цели рассчитывается по формуле $F(2, 0, t) = 4 + 4 \cdot t$.

Выполним проверку совпадения значений целевых функций на границе интервала $\beta_{o2} = \alpha_{o3} = 5/7$.

$$F\left(\frac{10}{11},\frac{18}{11},\frac{5}{7}\right)=\frac{74+2\cdot\frac{5}{7}}{11}=\frac{518+10}{11\cdot 7}=\frac{528}{11\cdot 7}=\frac{48\cdot 11}{11\cdot 7}=\frac{48}{7},$$

$$F\left(2,0,\frac{5}{7}\right)=\left(4+4\cdot\frac{5}{7}\right)=\frac{48}{7}.$$

Значения функций целей одинаковы, как и ожидалось.

Пункт № 2 алгоритма.

Проанализируем границы интервалов

$$\alpha_{o3} = \max_{\widetilde{p}_j > 0}\left\{-\frac{\widetilde{\delta}_j}{\widetilde{p}_j}\right\} = \left\{\frac{2}{3},\frac{7}{3}\right\}\left\{-\frac{\frac{2}{3}}{\frac{2}{3}} = -1, -\frac{-\frac{5}{3}}{\frac{7}{3}} = \frac{5}{7}\right\} = \frac{5}{7}$$

Отмечаем совпадение границ интервалов $\beta_{o2} = \alpha_{o3}$. Так как в строке $\widetilde{p}_j$ отсутствуют отрицательные числа, это говорит о том, что $\beta_{o3} \to \infty$.

Пункт № 4 алгоритма.

Поскольку значение β выходит за границу изменения параметра, заданного в условии задачи, то задачу параметрического программирования можно считать решённой.

Ответ.

В результате решения имеем несколько интервалов изменения параметра *t*, на каждом из которых имеются своя точка оптимума и своё выражение для вычисления функции цели:

$$\left[-\frac{1}{2},-\frac{1}{3}\right]; X^*_{opt} = [0 \quad 2]; F\left(X^*_{opt},t\right) = (3-1\cdot t)\cdot 2 = 6-2t,$$

$$\left[-\frac{1}{3},\frac{5}{7}\right]; X^*_{opt} = \left[\frac{10}{11} \quad \frac{18}{11}\right]; F\left(X^*_{opt},t\right) = (2+2\cdot t)\cdot\frac{10}{11} + (3-1\cdot t)\cdot\frac{18}{11} = \frac{74+2\cdot t}{11},$$

$$\left[\frac{5}{7},\infty\right); X^*_{opt} = [2 \quad 0]; F\left(X^*_{opt},t\right) = (2+2\cdot t)\cdot 2 = 4+4\cdot t.$$

На рисунке 2.11 показаны местоположения нормалей при различных значениях параметра *t*.

$$f(x_1,x_2) = 1\cdot x_1 + \frac{7}{2}\cdot x_2, \quad t = -\frac{1}{2};$$
$$f(x_1,x_2) = 2\cdot x_1 + 3\cdot x_2, \quad t = 0;$$
$$f(x_1,x_2) = 6\cdot x_1 + 1\cdot x_2, \quad t = 2.$$

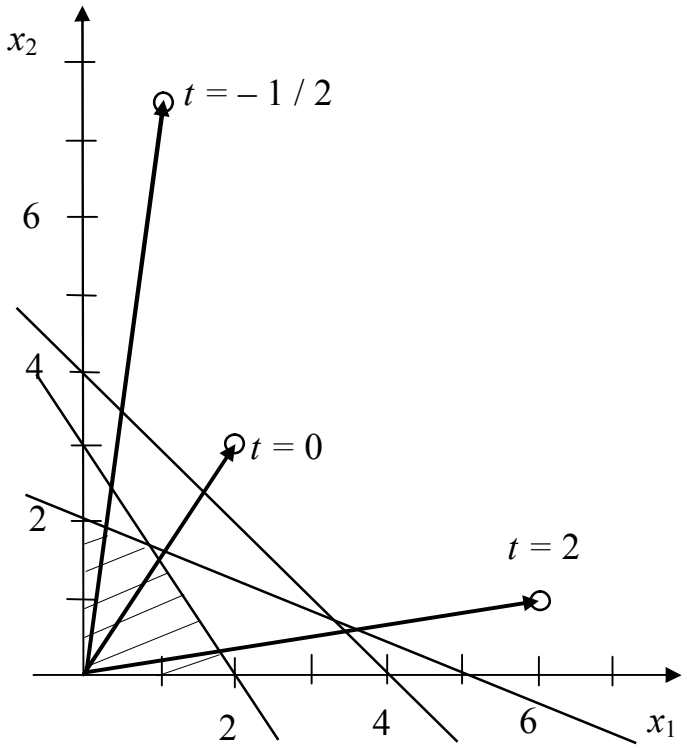


Рисунок 2.11 – Различные положения нормали

2.4.3 Вопросы для самоконтроля

1. Когда возникают задачи параметрического программирования?

2. Почему границей изменения параметра t при вариации вектора свободных членов служит отношение $\frac{\widetilde{b}_i}{\widetilde{p}_i}$?

3. Почему границей изменения параметра t при вариации коэффициентов функции цели служит отношение $\frac{\widetilde{\delta}_j}{\widetilde{p}_j}$?

4. Что означает физически или геометрически параметрические изменения элементов вектора свободных членов системы ограничений?

5. Что означает физически или геометрически параметрические изменения коэффициента целевой функции?

6. Всегда ли разрешима задача параметрического программирования?

7. Почему задача параметрического изменения элементов матрицы системы ограничений рассматривается и решается весьма редко?

8. В чем сущность и особенности методов решения задачи параметрического программирования по отношению к классической ЗЛП?

2.5. НЕЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ (НП – ПРОГРАММИРОВАНИЕ)

Постановка НП-задачи формулируется как нахождения оптимума целевой функции *f(X)* при ограничениях и задаётся моделью вида

$$\begin{cases} f(x_1,x_2,...,x_n) \to \max, \\ g_1(x_1,x_2,...,x_n) \ge 0, \\ g_2(x_1,x_2,...,x_n) \ge 0, \\ ... \\ g_m(x_1,x_2,...,x_n) \ge 0. \end{cases} \qquad (2.67)$$

где функции $f(X)$ и $g_i(X)$, $i = 1, m$, в общем случае, нелинейные.

НП-задачи существенно отличаются от ЗЛП неформализованностью методов их решения. Нелинейность приводит к тому, что:

- область принятия решения может быть невыпуклая;
- область может иметь бесконечное число крайних точек.

Поэтому для решения НП-задач разработаны методы, которые ориентированы на классы задач в их конкретной постановке.

Общего подхода, являющегося универсальным во всех случаях, ***создать не удалось*** [12, 46, 67, 68].

2.5.1. Аналитические методы определения экстремумов

Указанные методы основываются на известных вам методах классического математического анализа, базируясь на ряд теорем [33].

Теорема 1 (о существовании экстремума). Если функция многих переменных $f(x_1, x_2, \ldots, x_n)$ непрерывна и определена на замкнутом множестве $\mathfrak{R}$, то она достигает на этом множестве, ***по крайней мере, один раз*** своего минимального и максимального значений.

Теорема 2 (о местоположении экстремума). Если $f(x_1, x_2, \ldots, x_n)$ является функцией нескольких переменных, определённой на допустимой области $\mathfrak{R}$, то экстремальное значение f (если оно существует) достигается в одной или нескольких точках, принадлежащих:

- множеству стационарных точек $S(X)$;
- множеству точек границы $G(X)$;

- множеству точек, в которых (где) функция $f(x_1, x_2, \ldots, x_n)$ не дифференцируема.

Множество точек $S(X)$ функции $f(x_1, x_2, \ldots, x_n)$ называется множеством стационарных точек, если его элементы удовлетворяю условию

$$\nabla f(X) = \left\{ \frac{\partial f(X)}{\partial x_j}, j = \overline{1,n} \right\} = 0 . \qquad (2.68)$$

Вектор $\nabla f(X)$ – называют градиентом функции.

Находящий в стационарной точке ***минимум*** или ***максимум*** функции, может быть как ***абсолютным***, так и ***относительным***.

Относительный максимум функции $f(X)$ достигается в точке X^0 с координатами $(x_1^0, x_2^0, \ldots, x_n^0)$, если для всех точек, лежащих в малой окрестности точки X^0, имеет место неравенство $f(X^0) \geq f(X^0 + H)$, где $H = \{h_1, h_2, \ldots, h_n\}$.

Относительный максимум называется ещё ***локальным*** максимумом.

Абсолютный максимум функции $f(X)$ достигается в точке X^* с координатами $(x_1^*, x_2^*, \ldots, x_n^*)$, если для всех точек, принадлежащих множеству ограничений $\mathfrak{R}$ справедливо неравенство $f(X^*) \geq f(X)$, где $X = \{x_1, x_2, \ldots, x_n\} \in \mathfrak{R}$.

Абсолютный максимум называется ещё ***глобальным*** максимумом.

Аналогично, с точностью до знака неравенства, формулируются определения абсолютного и относительного минимумов.

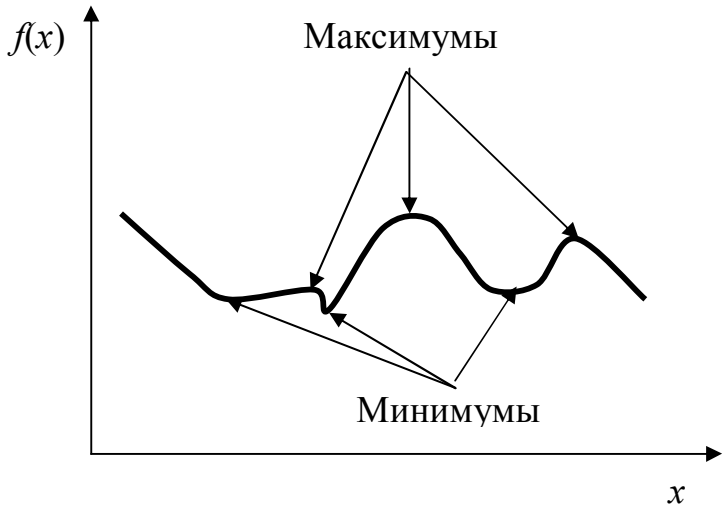


Рисунок 2.12 – Локальные и глобальные экстремумы

Характер экстремума, минимум или максимум, характеризуется выпуклостью или вогнутостью функции (рисунок 2.12).

Пусть $\mathfrak{R}$ – выпуклое множество точек n-мерного пространства.

Если для произвольного множителя $k \in [0, 1]$ и некоторого приращения ΔX выполняется неравенство

$$f(X + k\Delta X) \geq f(X) + k[f(X) - f(X + \Delta X)], \tag{2.69}$$

то функция называется ***вогнутой*** (обращена выпуклостью вверх, в отличие от математического анализа!!!), а если

$$f(X + k\Delta X) \leq f(X) + k[f(X) - f(X + \Delta X)], \tag{2.70}$$

то функция называется ***выпуклой***.

Если неравенства (2.69) и (2.70) строгие, говорят о **строгой вогнутости** и ***строгой выпуклости***. Для одномерного случая указанные неравенства интерпретируются графически, как это представлено на рисунке 2.13.

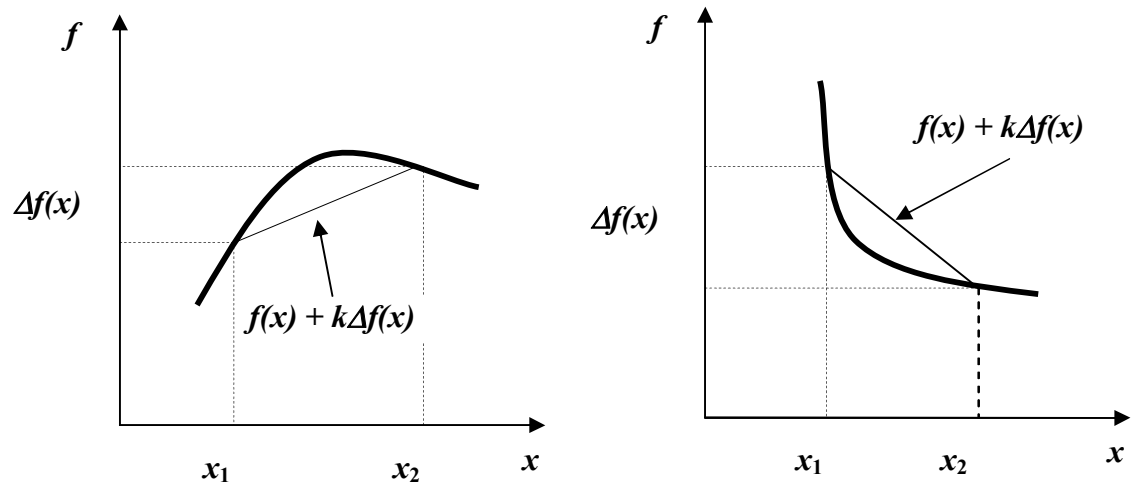


Рисунок 2.13 – Вогнутости и выпуклости функций

В многомерном случае непосредственное применение (2.69) или (2.70) является проблематичным. С этой целью применяется матрица Гёссе (или Гессе), элементы которой составляются из производных второго порядка и определяются так:

$$h_{i,j} = \left[\frac{\partial^2 f(X)}{\partial x_i \partial x_j} \right]_{x=x_0}, i = \overline{1,n}, j = \overline{1,m}. \tag{2.71}$$

Матрица Гёссе (обозначается как $\nabla^2 f(X)$ и $H(X)$) называется ***положительно определённой***, если её главные угловые миноры

положительны, и ***отрицательно определённой***, если её главные угловые миноры имеют знак $(-1)^k$, k – номер углового минора.

В качестве напоминания [18, 20, 35, 36]: *минором элемента $a_{i,j}$ матрицы A называется определитель, построенный из элементов этой матрицы, оставшихся после вычёркивания i-ой строки и j-ого столбца.*

Главные угловые миноры матрицы соответствуют элементам, расположенным вдоль главной (с Северо-запада на Юго-восток) её диагонали.

$$\mu_1 = \begin{pmatrix} h_{1,1} & h_{1,2} & h_{1,3} \\ h_{2,1} & h_{2,2} & h_{2,3} \\ h_{3,1} & h_{3,2} & h_{3,3} \end{pmatrix} = \begin{vmatrix} h_{2,2} & h_{23} \\ h_{3,2} & h_{3,3} \end{vmatrix} = h_{2,2} \cdot h_{3,3} - h_{3,2} \cdot h_{2,3}.$$

Положительная определённость матрицы Гёссе соответствует ***выпуклости*** функции, отрицательная определённость – ***вогнутости***. Случай, когда знаки чередуются не по порядку, соответствует ***перегибу***.

Теорема 3. Для того, чтобы в точке X_0 достигался внутренний относительный максимум, достаточно ***равенства нулю всех первых*** производных и ***строгой вогнутости*** функции в окрестностях X_0.

Теорема 4. Для того, чтобы в точке X_0 достигался внутренний относительный минимум, достаточно ***равенства нулю всех первых*** производных и ***строгой выпуклости*** функции в окрестностях X_0.

Пример. Исследовать на экстремум функцию без ограничений

$$f(x_1, x_2, x_3) = x_1 + 2x_3 + x_2 \cdot x_3 - x_1^2 - x_2^2 - x_3^2.$$

Градиент этой функции есть

$$\nabla f(x_1, x_2, x_3) = \{1 - 2x_1; x_3 - 2x_2; 2 + x_2 - 2x_3\}.$$

Используя условие стационарной точки (2.68), получим её координаты

$$X_0 = \left\{\frac{1}{2}; \frac{2}{3}; \frac{4}{3}\right\}.$$

Матрица Гёссе, построенная для рассматриваемого случая

$$\nabla^2 f(X) = \begin{pmatrix} -2 & 0 & 0 \\ 0 & -2 & 1 \\ 0 & 1 & -2 \end{pmatrix}.$$

Её угловые миноры суть

$$\mu_1 = \begin{vmatrix} -2 & 1 \\ 1 & -2 \end{vmatrix} = 3, \mu_2 = \begin{vmatrix} -2 & 0 \\ 0 & -2 \end{vmatrix} = 4, \mu_3 = \begin{vmatrix} -2 & 0 \\ 0 & -2 \end{vmatrix} = 4.$$

Они положительны, следовательно, матрица Гессе положительно определена, функция выпукла, и в точке X_0 с координатами $\left\{\frac{1}{2};\frac{2}{3};\frac{4}{3}\right\}$ достигается минимум.

2.5.2. Методы поиска экстремумов в задачах без ограничений или в случае ограничений с разделяющимися переменными

Ограничение $g_i(x_1, x_2, \ldots, x_n) = 0$ является функцией с ***разделяемыми переменными (сепарабельной)***, если его можно представить в виде

$$x_i = \varphi_i(\{x_j\}), \qquad i \neq j, \;\; j = \overline{1,n}.$$

Общий алгоритм решения

- Отыскивается множество всех стационарных точек S функции $f(X)$ внутри допустимого множества $\mathfrak{R}$ и выбираются координаты точки, в наибольшей степени отвечающие направлению оптимизации задачи.
- Рассматривается множество точек границы G. Для этого выполняют разделение переменных по каждому из ограничений, с последующей подстановкой выражений переменных в функцию цели $f(X)$. Полученные функции исследуются на экстремумы. Выбираются координаты интересующего нас оптимума.
- Определяется и подвергается исследованию множество точек, принадлежащих $\mathfrak{R}$, где функция не дифференцируема.
- Из результатов предыдущих шагов выбирается наилучшее решение.

Замечания.

1. Метод требует значительных вычислительных затрат и аналитических преобразований.

2. Не отвечает должной формализации для использования вычислительной техники.

3. Применение ограничивается задачами, область поиска решений которых описывается функциями с сепарабельными переменными.

Тем не менее, были разработаны многочисленные методы, направлены на решение задачи и ориентированные на применение ЭВМ [9, 29]. “Движение” текущей точки к оптимуму может происходить различными способами. В зависимости от этого, имеет место следующая классификация, методы подразделяются на:

- прямые методы или методы, использующие лишь значения функции;
- методы первого порядка, использующие, наряду со значениями функции значения первых частных производных;
- методы второго порядка, использующие дополнительно к значениям функции и первых частных производных и прочие частные вторые производные и выше.

Причём производные могут вычисляться как аналитически, так и численно.

2.5.2.1. Прямые методы поиска [9]

Одномерный поиск

При одномерном поиске функция *f(x)*, экстремум которой ищется, зависит от одной переменной. В дальнейшем будем предполагать, что ***решается задача минимизации***. Совершенно аналогичный алгоритм может быть использован и для случая поиска максимума. Отличия будут заключаться в знаках неравенств в проверках, производимых в процессе функционирования алгоритма.

Одномерный поиск является составной частью многих алгоритмов многомерного поиска в качестве вспомогательного. Алгоритмов одномерного поиска разработано изрядно, мы рассмотрим только самые интересные из них как с точки зрения скорости вычислений, так и их структуры.

Дихотомический поиск

Так же называется методом половинного деления. Это очень древний алгоритм, известный со времён раннего средневековья Ближнего Востока под названием “Поимка льва в пустыне”.

Есть пустыня, в ней лев. Делим пустыню пополам, лев окажется в одной из половинок. Часть пустыни, где находится лев, снова подвергаем делению… В конце концов, размеры пустыни окажутся чуть больше величины льва, и царя зверей остаётся только заключить в клетку.

Входные данные. Интервал поиска (сиречь, пустыня) [*a, b*]; допустимая конечная длина интервала, определяющая точность расчётов $l > 0$; константа различимости ε.

Условие окончания: $b - a \le l$.

Дадим пояснение алгоритма с использованием рисунков 2.14 и 2.15.

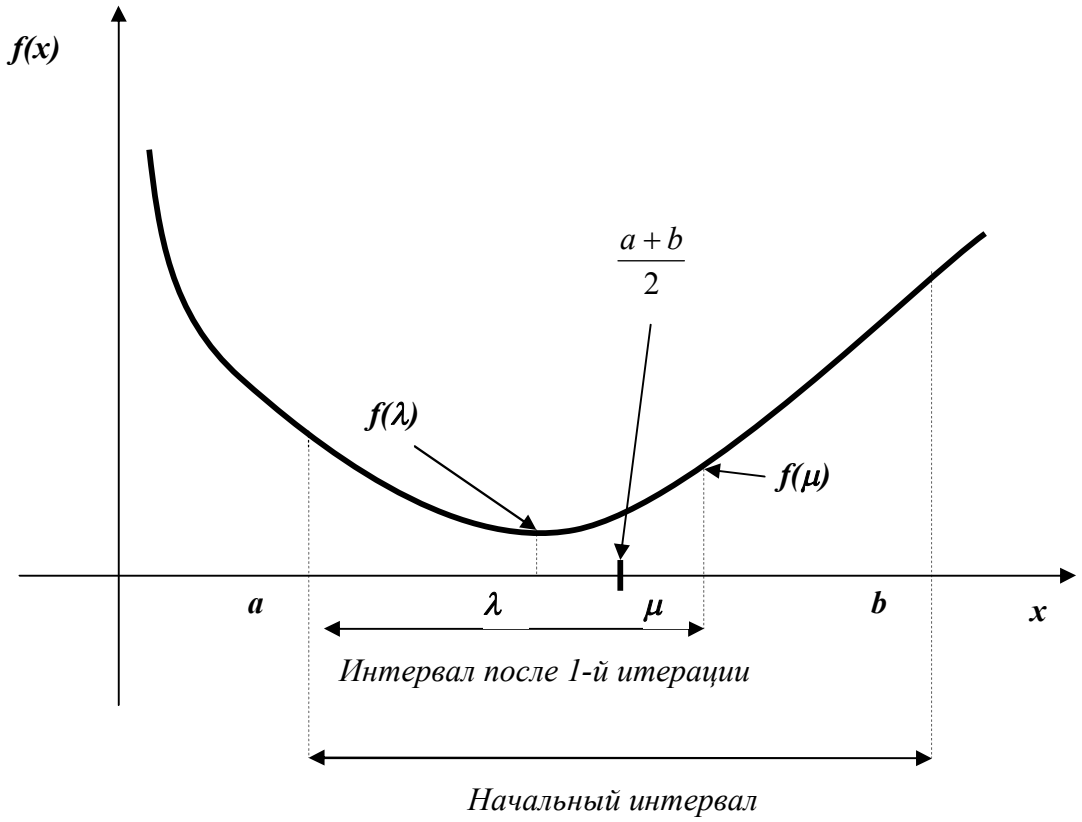


Рисунок 2.14 – Сужение интервала по мере дихотомии

Первоначально вычисляется пара величин: $\lambda = \frac{a+b}{2} - \varepsilon$ и $\mu = \frac{a+b}{2} + \varepsilon$, а тако же соответствующие им значения функций $f(\lambda)$ и $f(\mu)$. Полученные значения функций сравниваются между собой.

Пусть, например, $f(\lambda) \le f(\mu)$ (условие № 1 на схеме алгоритма, рисунок 2.14), выход «да». В этом случае сдвигается граница b, как это показано на рисунке 2.13. В противном случае, выход «нет», граница a сдвигается по направлению к середине интервала $\frac{a+b}{2}$.

Длительность вычислительного процесса контролируется условием № 2: вычисления заканчиваются при возникновении ситуации $b - a \le l$, которая означает, что границы интервала поиска сузились до заданного точностного размера l.

Имеется выражение, которое позволяет определить число итераций, необходимое для получения решения с заданной точностью:

$$b_k - a_k = \frac{(b-k)}{2^k} + 2 \cdot \varepsilon \left(1 - \frac{1}{2^k} \right),$$

где k – число итераций.

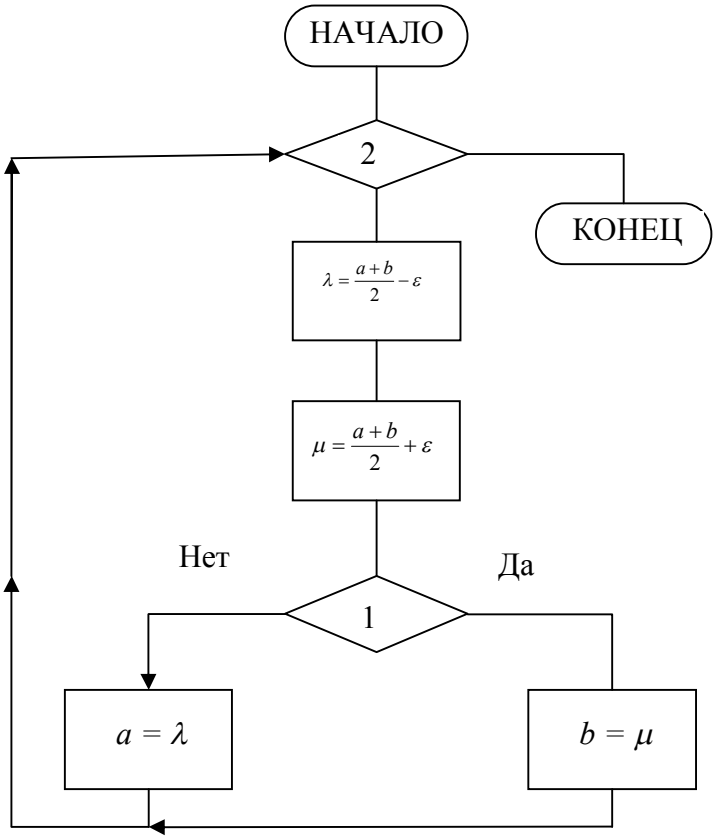


Рисунок 2.15 – Схема алгоритма дихотомического метода

Алгоритм, описанный выше, обладает самой меньшей скоростью сходимости, но, тем не менее, пользуется популярностью у программистов за простоту его реализации.

Метод золотого сечения

При функционировании метода выполняются условия:

- интервал поиска сужается равномерно;
- параметры точек сравнения и концов интервала соотносятся следующим образом

$$b - \lambda = \mu - a. \qquad (2.72)$$

Выполнение (2.72) достижимо, если значения λ и μ рассчитывать по формулам

$$\lambda = a + (1 - \alpha) \times (b - a) \text{ и} \quad (2.73)$$
$$\mu = a + \alpha \times (b - a), \quad (2.74)$$

где $|\alpha| < 1$.

Действительно, из (2.74) получается

$$\mu - a = \alpha \times (b - a). \quad (2.75)$$

Если λ, определяемое (2.73), подставить в левую часть (2.72), то

$$\boldsymbol{b - a} - (1 - \alpha) \times (\boldsymbol{b - a}) = \alpha \times (b - a). \quad (2.76)$$

Сопоставляя (2.75) и (2.76) видим, что условие (2.72) будет неизменно выполняться при определении λ и μ по формулам (2.73) и (2.74).

В ходе вычислений новые границы переставляются таким образом, чтобы либо $\lambda^{\nabla} = \mu$, либо $\mu^{\nabla} = \lambda$. Таким образом, пересчёт значений координат и функции в одной из этих точек не производится.

Коэффициент золотого сечения $\alpha = 0{,}618\ldots$ есть предел отношения соседних членов ряда Фибоначчи при их числе, стремящемся бесконечности.

Интервал поиска сужается равномерно, пропорционально α по закону

$$b_k - a_k = \alpha^k \ (b - a),$$

что позволяет оценить значение k – число итераций, необходимых до нахождения оптимума с заданной точностью.

Чертёж, поясняющий ход расчетов, и схема алгоритма представлены на рисунках 2.15 и 2.16.

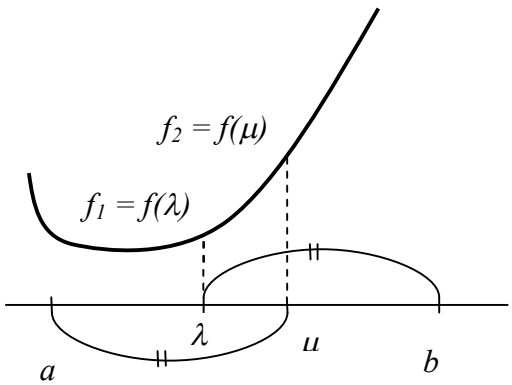


Рисунок 2.15 – Интервалы в методе золотого сечения

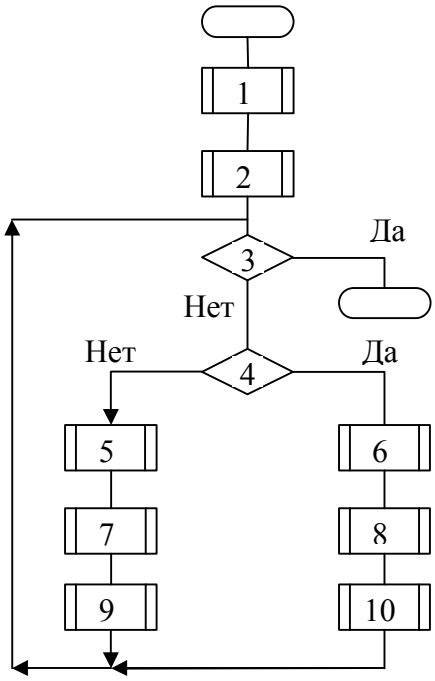


Рисунок 2.16 – Алгоритм метода золотого сечения

Входные данные. Интервал поиска [*a, b*]; точность расчётов $l > 0$.

Блоки № 1 рассчитывает параметры λ и μ по формулам (2.73) и (2.74), а блок № 2 определяет значения функции $f_1 = f(\lambda)$ и $f_2 = f(\mu)$, им соответствующие. Условие 3 $b - a \le l$ определяет остановку вычислительного процесса при достижении заданной погрешности. Условие 4 $f_1 \le f_2$ ответственно за сдвиги границ интервала поиска, аналогично дихотомическому методу. Блоки №№ 5, 7, 9 выполняют операции, связанные со сдвигом границы *a*.

- 5: $a = \lambda;\ \lambda = \mu$.
- 7: Расчёт μ по формуле (2.74)
- 9: Перестановка пересчёт значений функций $f_1 = f_2;\ f_2 = f(\mu)$

Блоки №№ 6, 8, 10 выполняют аналогичные операции, связанные со сдвигом границы *b*.

- 6: $b = \mu;\ \mu = \lambda$.
- 8: Расчёт λ по формуле (2.73)
- 10: Перестановка пересчёт значений функций $f_2 = f_1;\ f_1 = f(\lambda)$.

Метод Фибоначчи

В отличие от метода золотого сечения, указанный метод

- на каждой итерации изменяет коэффициент сжатия интервала и
- выполняется заранее известное числа шагов.

Основывается на использовании ряда Фибоначчи, элементы которого определяются рекуррентным соотношением

$$F_{n+1} = F_n + F_{n-1},\ F_0 = F_1 = 1, \tag{2.77}$$

определяющим, что каждый последующий член ряда образован суммой двух предыдущих:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 …

Входные данные. Интервал поиска [*a, b*], параметр, задающий точность расчётов $l > 0$, константа различимости $\varepsilon << l$.

Условие окончания: поиск заканчивается через определённое число шагов (итераций), определяемых по неравенству

$$\frac{b-a}{l} < F_n \Rightarrow n\,. \tag{2.78}$$

Координаты точек λ_k и μ_k, где k – номер итерации, рассчитываются по формулам

$$\lambda_k = a_k + \frac{F_{n-k-1}}{F_{n-k+1}}(b_k - a_k)\,, \tag{2.79}$$

$$\mu_k = a_k + \frac{F_{n-k}}{F_{n-k+1}}(b_k - a_k)\,. \tag{2.80}$$

Согласно формулам (2.79) и (2.80), от итерации к итерации, интервал поиска меняется по закону

$$b_{k+1} - a_{k+1} = \frac{F_{n-k}}{F_{n-k+1}}(b_k - a_k) = b_k - \lambda_k = \mu_k - a_k\,, \tag{2.61}$$

причём, перемещение границ интервала поиска и пересчёт $f(\lambda_k)$ и $f(\mu_k)$ происходит с соблюдением тех же принципов, что и в методе золотого сечения.

Константа различимости ε используется после окончания циклической части алгоритма для уточнения полученного результата. Схема алгоритма представлена рисунком 2.18.

Блок № 1 выполняет подготовительные операции: осуществляется расчёт ряда Фибоначчи (2.77), попутно определяется (2.78) – число членов ряда n, необходимых для расчёта; начальные значения

$$\lambda_0 = a + \frac{F_{n-2}}{F_n}(b-a), \quad \mu_0 = a + \frac{F_{n-1}}{F_n}(b-a), \quad f_1 = f(\lambda_0) \text{ и } f_2 = f(\mu_0).$$

Блок № 2 есть заголовок цикла изменения счётчика k от 1 до $n - 2$, тело цикла представляется блоками №№ 3 – 5.

Условие, управляющее сдвигом границ, $f_1 \le f_2$, представлено блоком № 3.

Сдвиг левой границы, выход «нет» блока № 3, выполняется в блоке № 4:

- $a = \lambda_k$; $\lambda_k = \mu_k$.
- Расчёт μ_k по формуле (2.80)
- Перестановка пересчёт значений функций $f_1 = f_2$; $f_2 = f(\mu_k)$.

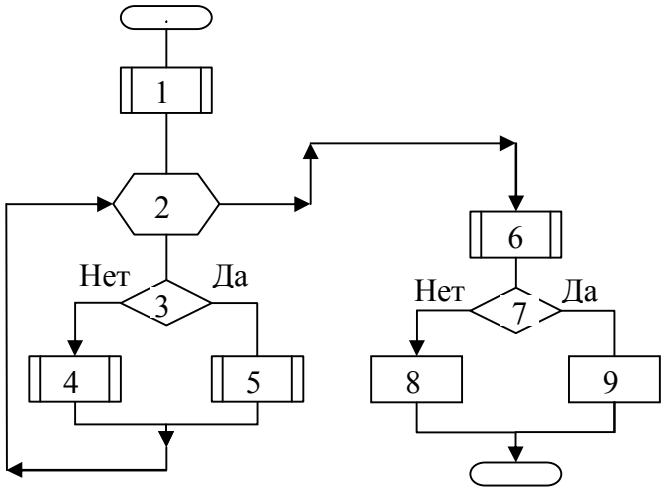


Рисунок 2.18– Алгоритм метода Фибоначчи

Сдвиг правой границы, выход «да» блока № 3, выполняется в блоке № 5:

- $b = \mu_k$; $\mu_k = \lambda_k$.
- Расчёт λ_k по формуле (2.79)
- Перестановка пересчёт значений функций $f_2 = f_1$; $f_1 = f(\lambda_k)$.

Блоки №№ 6, 7, 8 и 9 осуществляют уточнение результата методом половинного деления. Блок № 6 выполняет операции $\mu = \lambda + \varepsilon$ и $f_2 = f(\mu)$. Блок № 7 проверяет условие $f_1 \le f_2$ и, в зависимости от его выполнения, сдвигает левую границу, выход «нет» блок № 8, $a = \lambda$, либо правую, выход «да», блок № 9, $b = \mu$.

Многомерный поиск [8, 5]

При многомерном поиске функция *f(X)*, экстремум которой ищется, зависит от нескольких переменных. Алгоритмы многомерного поиска эвристические. Хотя некоторые из них носят имена своих первооткрывателей, не факт, что они не переизобретаются заново многочисленными последователями, хотя и более невежественными в плане оптимизации, но, тем не менее, более подкованными в плане алгоритмизации.

Метод конфигураций

Метод конфигураций называется тако же методом траекторий или, по имени авторов, методом Хука и Дживса.

Идея метода состоит в следующем.

- Направления поиска ориентированы, так сказать, по сторонам света или вдоль осей координат в обе стороны.
- Вокруг текущей точки производится поиск направления, в котором функция убывает.
- Если такое направление найдено, то шаг поиска увеличивается, а поиск продолжается в выбранном направлении – устанавливается так называемый тренд поиска, и осуществляется до тех пор, пока функция в этом направлении убывает.
- Если факта убывания функции не обнаружено, то шаг поиска уменьшается.

Таким образом, делается попытка найти «овраг» (при минимизации) функции и двигаться вдоль него к точке минимума.

В задачи максимизации ищется направление возрастания функции и «хребет».

Алгоритм метода показан на рисунке 2.19.

Входные данные. Начальное значение шага λ, координаты начальной точки X_0 ускоряющий множитель α, точность нахождения решения ε, список возможных направлений поиска $S = \{s_1, s_2, \ldots, s_n\}$. Для числа переменных равного двум, это орты: $s_1 = (0, 1)$ и $s_2 = (1,0)$.

Условие окончания: значение текущего шага укладывается в точность, $\varepsilon \geq \lambda$.

Блок № 1. Координаты начальной точки переприсваивается текущей переменной: $Y = X_0$.

Блок № 2. Организуется цикл перебора направлений поиска $j = 1, n$, тело цикла составляют блоки №№ 3 – 7.

Сравнение в блоке № 3 проверяется факт убывания функции в выбранном направлении: $f(Y + \lambda \times s_j) < f(Y)$. Если функция убывает, выход «да» блока № 3, то текущая точка перемещается в выбранном направлении $Y = Y + \lambda \times s_j$ (блок № 4).

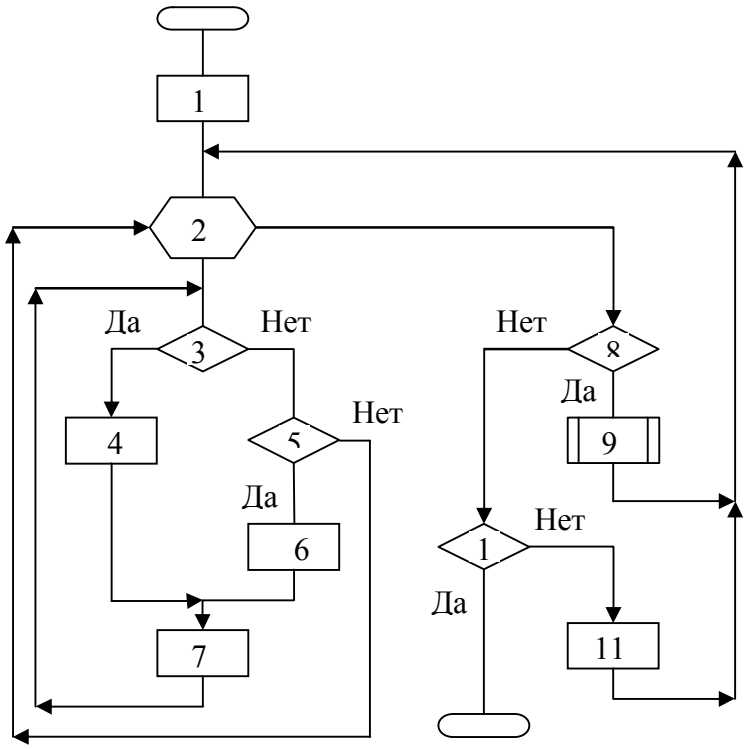


Рисунок 2.19 – Алгоритм метода Хука-Дживса

В противном случае, выход «нет» блока № 3, в блоке № 5 проверяется факт убывания функции в направлении, противоположном выбранному: $f(Y - \lambda \times s_j) < f(Y)$.

Если функция не убывает ни в выбранном, ни в противоположном направлениях (выход «нет», блок № 5), то выбирается очередное направление.

Если сравнение в блоке № 5 произошло удачно (выход «да»), то текущая точка решения перемещается в блоке № 6 по закону $Y = Y - \lambda \times s_j$.

Блок № 7 выполняет факультативное действие $\lambda = 2 \times \lambda$, которое увеличивает шаг, ускоряя тем самым поиск. В ряде реализаций алгоритма данное действие отсутствует.

Блок № 8 достигается в ходе работы алгоритма, после перебора всех направлений поиска и попадания в ситуацию, когда при заданном значении λ в окрестностях точки X_0 убывания функции не обнаружено.

Блок № 9 выполняет пересчёт координат текущей точки по формуле

$$Y = X_0 + \alpha \times (Y - X_0); \; X_0 = Y.$$

Блок № 10 осуществляет проверку окончания: $\varepsilon \geq \lambda$? Если «да» то алгоритм заканчивает работу, если «нет», то шаг поиска уменьшается

$$\lambda = \frac{\lambda}{2},$$

после чего начинается поиск вокруг текущей точки X_0 с уменьшенным шагом.

Пример траектории поиска представлен ниже, на рисунке 2.20.

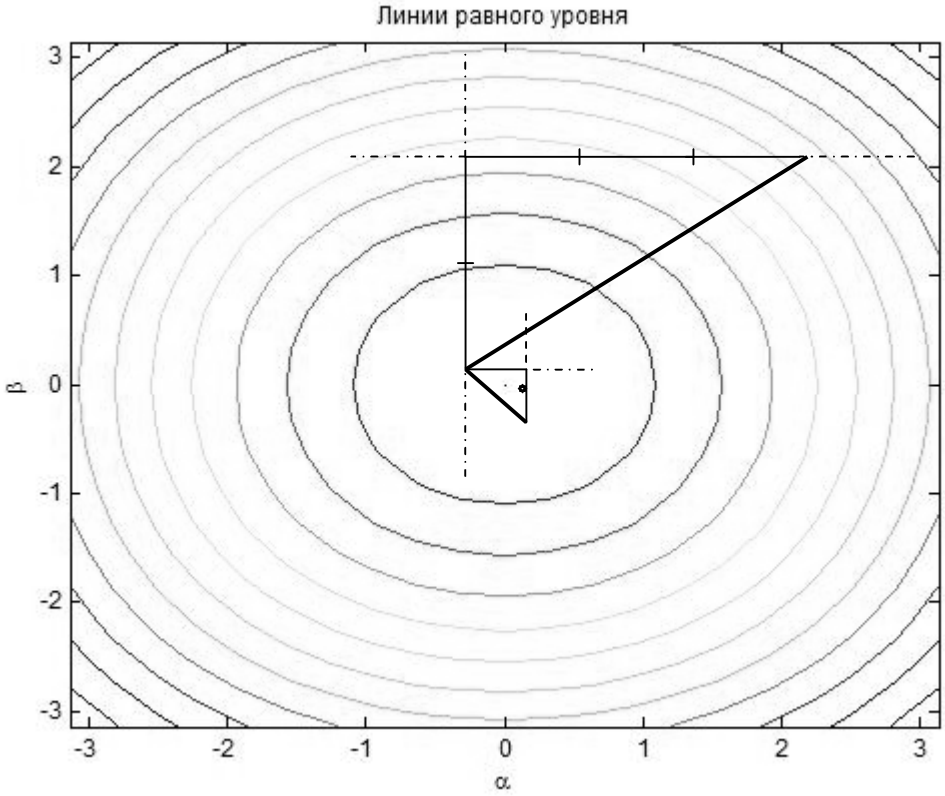


Рисунок 2.20– Поиск решения методом траекторий

Для демонстрации использована функция вида $f(\alpha,\beta) = -\frac{\sin(\alpha+\beta)}{\alpha+\beta}$, которая достигает своего «дна» в точке с координатами (0, 0), это положение отмечено точечкой в середине рисунка, пунктиром показаны неудачные пробы, жирным – траектория спуска.

Замечания.

К безусловным ***достоинствам*** метода следует отнести следующие:

- способность восстанавливать направление движения, когда при искривлении «оврага» тренд поиска теряется;
- явного задания функции не требуется;
- легко учитываются ограничения на переменные и область поиска.

В качестве недостатка отмечается, что при попадании в область размытого локального минимума есть риск остановиться, не достигнув глобального минимума.

Метод Розенброка [46, 47]

Идея метода состоит в поиске по взаимно-ортогональным направлениям на каждом шаге алгоритма, в отличие от алгоритма Хука-Дживса, где поиск осуществляется вдоль координатных осей.

Если поиск, в каком либо из направлений, оказывается удачным, то шаг поиска по этому направлению увеличивается, в противном случае шаг уменьшается.

Поисковая процедура заканчивается, когда на каждом из направлений поиска одна проба окажется успешной, а одна – неудачной.

Данный метод ***особенно хорош*** при работе с функциями, обладающими длинными, узкими и искривлёнными гребнями и оврагами.

Топология алгоритма представлена на рисунке 2.21.

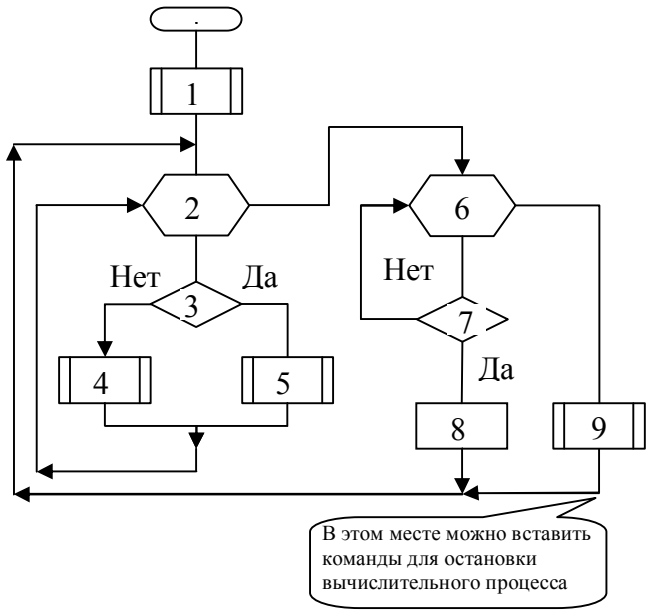


Рисунок 2.21 – Алгоритм метода Розенброка

Входные данные. Координаты начальной точки X_0; список возможных направлений поиска, первоначально набор ортогональных векторов $S = \{s_1, s_2, \ldots, s_n\}$; набор значений шагов поиска $\Lambda^T = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$; ускоряющий множитель α; замедляющий множитель β.

Условие окончания: автором не предусмотрено, ибо, скорее всего, считал вручную и вычислительный процесс контролировал визуально. При компьютерной реализации используют один из возможных подходов:

- выполняют несколько фиксированных шагов, после чего алгоритм прекращает свою работу;
- модуль вектора приращений (расстояние между смежными текущими точками на итерации) $|\Delta| \le \varepsilon$ несколько раз подряд.

Розенброком были предложены следующие значения констант: $\alpha = 3$, $\beta = -0{,}5$.

Блок № 1 осуществляет подготовительные операции, $Y = X_0$.

Блоки №№ 2 – 5 выполняют сканирование направлений: заголовок цикла перебора (блок № 2); вычисление точки в окрестностях текущей точки в выбранном направлении, значения функции и сравнение значений : $f(Y + \lambda_i s_i) < f(Y)$(блок № 3).

Если значение функции в выбранном направлении увеличивается или неизменно, выход «нет», то текущее направление поиска штрафуется, путём умножения $\lambda_i = \beta \times \lambda_i$ и запоминается соответствующие координаты в окрестностях текущей точки $P_i = Y + \lambda_i s_i$ – блок № 4.

Если функции в выбранном направлении убывает, выход «да», то направление поиска поощряется $\lambda_i = \alpha \times \lambda_i$, и также запоминаются координаты $P_i = Y + \lambda_i s_i$ – блок № 5.

Блоки №№ 6 и 7 осуществляют поиск удачного направления среди просканированных направлений. Блок № 6 – организует циклический перебор, блок № 7 проверяет условие $f(P_i) < f(Y)$.

Если первое же попавшееся направление удачное, текущая точка перемещается блоком № 8, $X_0 = P_i$, после чего вычислительный процесс возобновляется со сканирования направлений.

Если удачное направление поиска не найдено, то необходимо модифицировать систему направлений, что осуществляется блоком № 9:

- вычисляется приращение $\Delta = X_0 - Y$;
- переопределяется новая точка начала итерации $Y = X_0$;
- новое направление s_1 берется совпадающим с направлением вектора Δ (единичные значения по ненулевым координатам), а остальные направления пересчитываются с учётом условия ортогональности.

2.5.2.2. Градиентные методы поиска [40, 74, 83]

Эти методы основываются на использовании значений градиентов целевой функции, как следует из их названия.

В зависимости от порядка используемых градиентов бывают первого и второго порядков. Далее мы рассмотрим

- метод наискорейшего спуска (подъёма) – 1-го порядка;
- метод Ньютона (вторых производных) – 2-го порядка;
- метод сопряжённых направлений (сопряжённого градиента), предложенный Флетчером и Ривсом, 1-го порядка;
- методы переменной метрики, квазиньютоновские, 1-го порядка.

Общая идея, положенная в основу этих методов, для случая функции одной переменной, пояснена с помощью рисунка. 2.22.

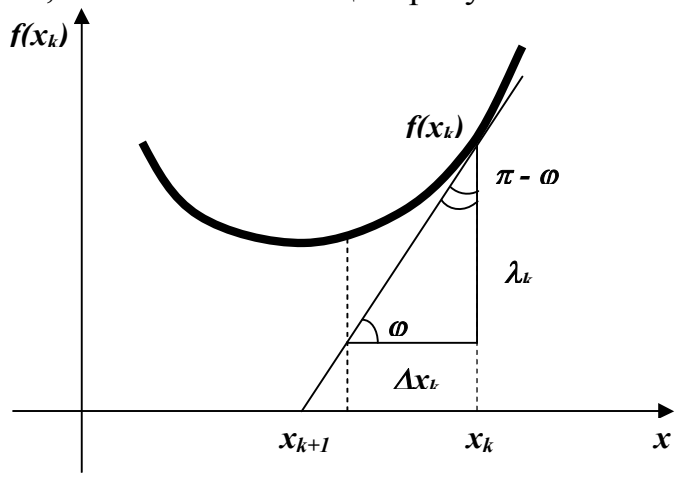


Рисунок 2.22 – Использование градиента

По рисунку видно, что $x_{k+1} = x_k - \Delta x_k$ и $\tan\varphi = f'(x_k)$ – геометрический смысл первой производной функции одной переменной.

Рассмотрение прямоугольного треугольника позволяет найти $\Delta x_k = \lambda_k \cdot f'(x_k)$ при заданном значении λ_k. В этом случае нами использованы формулы связи значений тригонометрических функций при изменении аргумента на величину $\pm\pi$.

Для многомерного случая справедливо аналогичное выражение

$$x_{k+1} = x_k - \lambda_k \times \nabla f(x_k), \qquad (2.81)$$

где $\lambda_k > 0$ – величина шага на k-ой итерации. Особенности отдельных градиентных методов заключаются в приёмах определения λ_k на каждой итерации.

Метод наискорейшего спуска (подъёма) [9, 85, 87]

При решении задачи на минимизацию речь идёт о спуске, а в случае максимизации – о подъёме. В алгоритмах указанное отличие отражается в знаках при выборе направления, которое совпадает с направлением градиента в задачах максимизации и противоположно ему (знак «минус») в задачах минимизации.

Луи Огюст Коши предложил выбирать значение λ_k путём решения задачи одномерной минимизации из условия

$$\lambda_k^* \Rightarrow \min_{\lambda>0} f(X_k + \lambda s_k), \tag{2.82}$$

где $s_k = -\nabla f(X_k)$ определяется значением градиента. При этом непосредственно значение λ_k может быть определено из уравнения

$$\frac{\partial f(X_k + \lambda s_k)}{\partial \lambda} = 0 \tag{2.83}$$

аналитически, либо путём численного решения задачи одномерной минимизации. Алгоритм метода тривиальный, изображён на рисунке 2.23.

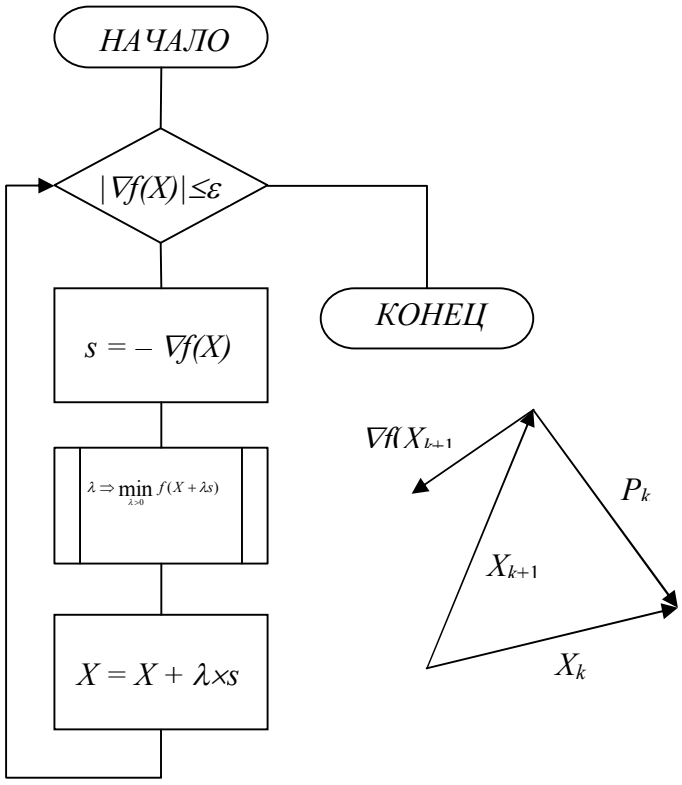


Рисунок 2.23 – Алгоритм метода наискорейшего спуска

Если ввести вектор, определяемый как $\vec{P}_k = \vec{X}_{k+1} - \vec{X}_k$, то вектор-градиент $\nabla\vec{f}(X_{k+1})$ перпендикулярен вектору приращения $\vec{P}_k$, то есть $\nabla\vec{f}(X_{k+1}) \perp \vec{P}_k$. Поэтому траектория спуска будет иметь, в общем случае, вид зигзага.

Тут уместно заметить, что метод наискорейшего спуска ***не обладает*** оптимальными свойствами, ***не обеспечивая минимум шагов*** при минимуме ***вычислений***, если только ***направление спуска не совпадает*** с направлением на минимум, локальный или глобальный.

Последнее справедливо только для ограниченного класса функций, поэтому метод считается ***полношаговым***. Наискорейшим спуск будет, например, для нелинейных функций чётных степеней.

Метод Ньютона (вторых производных) [9, 89 – 91]

Для итерационных методов, как мы знаем, характерна следующая связь между новыми и старыми координатами в пределах одной итерации

$$X_{k+1} = X_k + \Delta X_k. \tag{2.84}$$

Отыщем значение ΔX_k в ходе решения оптимизационной задачи с использованием ряда Тейлора 2-го порядка. Для функции одной переменной ряд, ограниченный до трёх членов, в окрестностях точки X_0 будет иметь вид

$$f(x) \cong f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2} f''(x_0)(x - x_0)^2,$$

а для многомерного случая

$$f(X) \cong f(X_0) + \nabla^T f(X_0)(X - X_0) + \frac{1}{2}(X - X_0)^T \nabla^2 f(X_0)(X - X_0),$$

где $\nabla^T f(X_0)$ – матрица Гёссе, составленная из частных производных второго порядка. Определяя ΔX_k ка (2.84), придём к формуле

$$f(X_{k+1}) = f(X_k) + \nabla^T f(X_k)\Delta X_k + \frac{1}{2}\Delta X_0^T \nabla^2 f(X_k)\Delta X_k.$$

Для отыскания оптимального значения ΔX_n последнее выражение подвергнем дифференцированию, а результат положим равным нулю. Отсюда

$$\Delta X_k = -\frac{\nabla f(X_k)}{\nabla^2 f(X_k)}.$$

Окончательно рекуррентное выражение для расчётов

$$X_{k+1} = X_k - \frac{\nabla f(X_k)}{\nabla^2 f(X_k)}. \qquad (2.85)$$

Если сравнить (2.85) и (2.81), видно, что для данного случая

$$\lambda_k = \left[\nabla^2 f(X_k)\right]^{-1},$$

то есть, λ_k на каждой итерации равно обращённой матрице Гёссе в текущей точке.

Выражение (2.85) есть задание алгоритма метода Ньютона в виде формулы, которая применяется циклически применяется циклически, до удовлетворения необходимой точности расчётов, то есть $|\Delta X_k| \le \varepsilon$.

Замечание. Обращение матрицы Гёссе на каждом шаге – основное неудобство метода, ведущее в временным затратам и росту погрешности вычисления. Последнее, отчасти, препятствует широкому использованию метода в практических приложениях.

Метод сопряжённого градиента (Флетчера – Ривса) [9, 34]

Понятие о сопряжённых направлениях

В общем случае, система линейно-назависимых направлений поиска S_1, S_2, $\ldots S_{n-1}$ называется ***сопряжённой*** по отношению к некоторой положительно определённой матрице Q если

$$S_i^T Q S_j = 0; \quad i \ne j; \quad i = 1,\overline{n-1}, \quad j = 1,\overline{n-1}.$$

Если $Q = I$ – единичная матрица, то вектора S_i и S_j, $i \ne j$ ортогональны.

Направление спуска S_1 будет сопряжено направлению S_0, если выполняется тождество:

$$S_0^T \cdot \nabla^2 f(X_0) \cdot S_1 = 0.$$

Как известно из математики [20, 21], для положительно определённой матрицы её собственные вектора образуют систему из $n - 1$ направлений, и такая система всегда существует, по крайней мере, одна.

Пусть исходное направление поиска в некоторой точке X_n есть

$$S_n = -\nabla f(x_n). \tag{2.86}$$

Сопряжённое направление будем строить по правилу

$$S_{n+1} = -\nabla f(x_{n+1}) + \omega_{n+1} S_n, \tag{2.87}$$

где ω_{n+1} – весовой коэффициент сопряжения. Этот коэффициент должен обеспечивать условие сопряжённости, то есть

$$S_n^T \nabla^2 f(x_n) S_{n+1} = 0. \tag{2.68}$$

Алгоритм движения точки в ходе поиска подчиняется известному выражению

$$x_{n+1} = x_n + \lambda S_n \Rightarrow \Delta x_n = \lambda S_n. \tag{2.89}$$

Необходимо найти нормирующий множитель ω_{n+1}.

Воспользуемся для начала аппроксимацией матрицы Гёссе по методу Л. Эйлера

$$\frac{\nabla f(x_{n+1}) - \nabla f(x_n)}{x_{n+1} - x_n} \cong H(x_n) = \nabla^2 f(x_n).$$

Из последнего выражения, определив знаменатель дроби через направление S_n по выражению (2.89), последует

$$\nabla f(x_{n+1}) - \nabla f(x_n) = H(x_n) \lambda S_n,$$

откуда получим

$$S_n^T = \frac{1}{\lambda} \left[\nabla f(x_{n+1}) - \nabla f(x_n)\right]^T H^{-1}(x_n). \tag{2.90}$$

Подстановка (2.86) в (2.87)

$$S_{n+1} = -\nabla f(x_{n+1}) - \omega_{n+1} \nabla f(x_n),$$

а, в дальнейшем, полученного результата совместно с (2.90) – в условие сопряжённости (2.88), даёт

$$\frac{1}{\lambda}\left[\nabla f(x_{n+1}) - \nabla f(x_n)\right]^T H^{-1}(x_n) H(x_n)\left[-\nabla f(x_{n+1}) - \omega_{n+1}\nabla f(x_n)\right] = 0.$$

Из последней записи, принимая во внимание, что $H^{-1}(x_n)H(x_n) = I$, выразим весовой коэффициент сопряжения:

$$\omega_{n+1} = \frac{\nabla^T f(x_{n+1}) \cdot \nabla f(x_{n+1})}{\nabla^T f(x_n) \cdot \nabla f(x_n)} = \frac{\left\|\nabla f(x_{n+1})\right\|^2}{\left\|\nabla f(x_n)\right\|^2}. \qquad (2.91)$$

Результат показывает, что коэффициент сопряжения (2.91) равен отношению квадратов норм (длин) векторов текущего и предшествующего градиентов.

Алгоритм расчётов представлен ниже, на рисунке 2.24. Шаг λ^* отыскивается по методу Коши, как и в методе наискорейшего спуска.

2.5.2.3. Методы поиска переменной метрики (квазиНьютоновские)

Обращение матрицы Гёссе на каждом шаге итерации есть основной недостаток метода второго порядка, предложенного Ньютоном и рассмотренного нами выше. Указанная матрица может быть плохо обусловленной, а численное обращение трудоёмким и приводить к погрешности – число операций умножения пропорционально n^3, где n – размерность матрицы. Методы переменной метрики или ***квазиньютоновские*** производят построение аппроксимирующей матрицы $\left[H(X_k)\right]^{-1}$ в процессе спуска [34, 67, 68].

Пусть алгоритм движения точки подчиняется закону

$$x_{n+1} = x_n - \lambda_n D_n \nabla f(x_n),$$

где D_n - аппроксимирующая матрица, λ_n - стабилизирующий множитель.

Аппроксимирующую матрицу D_n строят по правилу:

$$H^{-1}(x_n) \approx D_{n+1} = \omega(D_n + \Delta D_n), \qquad (2.92)$$

ω - нормирующий множитель.

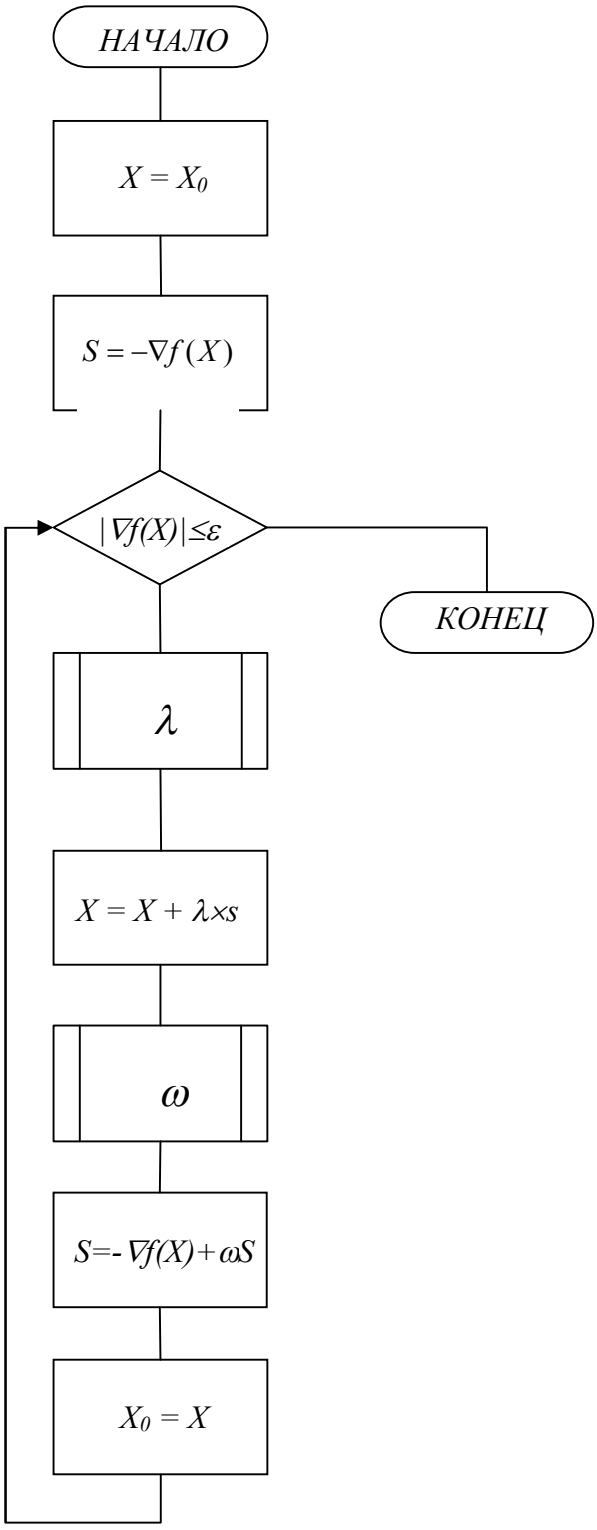


Рисунок 2.24 – Алгоритм метода Флетчера – Ривса

Определим приращение ΔD_n, обеспечивающее корректное построение матрицы $\left[H(X_k)\right]^{-1}$.

Воспользуемся аппроксимацией матрицы Гёссе

$$H(x_n) = \frac{\Delta g_n}{\Delta x_n}, \qquad (2.93)$$

где $\Delta g_n = \nabla f(x_{n+1}) - \nabla f(x_n)$, а $\Delta x_n = x_{n+1} - x_n$.

Приращение Δx_n определим из (2.93):

$$\Delta x_n = H^{-1}(x_n) \cdot \Delta g_n .$$

Подстановка сюда определения (2.92) даёт выражение

$$\Delta x_n = \omega(D_n + \Delta D_n)\Delta g_n \Rightarrow \Delta D_n \Delta g_n = \frac{1}{\omega}\Delta x_n - D_n \Delta g_n ,$$

откуда

$$\Delta D_n = \frac{1}{\omega} \cdot \frac{\Delta x_n}{\Delta g_n^T} \cdot \frac{Y^T}{Y} - \frac{D_n \Delta g_n}{\Delta g_n^T} \cdot \frac{Z^T}{Z}.$$

Фиктивные добавки – векторы Y и Z, размерности n, решают задачи определимости матричных операций, сходимости D_n к $H^{-1}(x_n)$ и её положительной определённости.

В алгоритме, предложенном ***Бройденом***, положено: $\omega = 1,\ Y = Z = \Delta x_n - D_n \Delta g_n$.

В алгоритме ***Давидона-Флетчера-Пауэлла*** аналогичные добавки определяются как: $\omega = 1,\ Y = \Delta x_n,\ Z = D_n \Delta g_n$.

В обоих случаях, в качестве начального приближения используется единичная матрица I.

Для последнего алгоритма

$$D_{n+1} = D_n + \frac{\Delta x_n}{\Delta g_n^T} \cdot \frac{\Delta x_n^T}{\Delta x} - \frac{D_n \Delta g_n}{\Delta g_n^T} \cdot \frac{(D_n \Delta g_n)^T}{D_n \Delta g_n} = D_n + A_n - B_n$$

или

$$D_{n+1} = I + \sum_{i=1}^{n} A_i - \sum_{i=1}^{n} B_i ,$$

в котором A_n обеспечивает сходимость D_n к $H^{-1}(x_n)$, а B_n положительную определённость.

Схема работы алгоритма – типовая для градиентных методов.

Параметр $\lambda_n^* \Rightarrow \min_{\lambda > 0} f(X_n + \lambda_n s_n)$, где $s_n = -\nabla f(X_n)$, вычисляется по методу Коши.

Условия окончания работы алгоритма задаются одним из нижеследующих способов.

1. Каждая из компонент векторов $D_n \nabla f(x_n)$ и $\lambda_n D_n \nabla f(x_n)$ меньше по модулю некоторой заданной величины ε.

2. Длина (норма) каждого из векторов $D_n \nabla f(x_n)$ и $\lambda_n D_n \nabla f(x_n)$ меньше величины ε.

2.5.3. Поиск экстремумов в задачах нелинейного программирования при ограничениях типа “равенство” (метод Лагранжа) [46, 69]

В основу метода положена идея сведения задачи поиска условного экстремума к поиску безусловного экстремума специальной функции. Пусть математическая модель представлена в виде

$$\begin{cases} f(x_1, x_2, ..., x_n) \to \max, \\ h_1(x_1, x_2, ..., x_n) \ge 0, \\ h_2(x_1, x_2, ..., x_n) \ge 0, \\ ... \\ h_m(x_1, x_2, ..., x_n) \ge 0. \end{cases} \qquad (2.94)$$

Допустим, что функции $f(X)$ и $h_i(X)$, $i = 1, m$, входящие в (2.94), нелинейные и имеют непрерывные частные производные.

Введём набор множителей $\lambda_1, \lambda_2, ..., \lambda_m$, по числу ограничений в модели (2.94), и составим функцию вида

$$L(x_1, x_2, ..., x_n, \lambda_1, \lambda_2, ..., \lambda_m) = f(x_1, x_2, ..., x_n) + \sum_{i=1}^{m} \lambda_i h_i(x_1, x_2, ..., x_n) \qquad (2.95)$$

Полученная ***функция*** называется функцией ***Лагранжа***, а множители λ, $i = 1, m$, – ***коэффициентами*** или ***множителями Лагранжа***.

Часть выражения, стоящего под знаком суммы в (2.95), можно интерпретировать как результат решения системы уравнений, в ходе которого происходит сложение или вычитание отдельных её уравнений, помноженных на коэффициенты.

Очевидно, что экстремум (2.95) будет при таких значениях $X^* = \{x_1^*, x_2^*, ..., x_n^*\}$ и $\Lambda^* = \{\lambda_1^*, \lambda_2^*, ..., \lambda_m^*\}$, которые будет удовлетворять системе уравнений

$$\left.\begin{aligned}\frac{\partial L(X^*,\Lambda^*)}{\partial x_j}=0, j=\overline{1,n},\\ \frac{\partial L(X^*,\Lambda^*)}{\partial \lambda_i}=0, i=\overline{1,m}.\end{aligned}\right\} \quad (2.96)$$

Эта система компактно записана в форме

$$\nabla f(X^*)+\sum_{i=1}^{m}\lambda_i^*\nabla h_i(X^*)=0. \quad (2.97)$$

Условия (2.96) суть ***необходимые*** условия существования экстремума, а запись (2.97) – отражает ***достаточность***.

Вектор $\nabla f(X^*)$ лежит в гиперплоскости, "натянутой на вектора $\nabla h_i(X^*)$" [33], эдакий зонтик. Известна следующая теорема, носящая имя Лагранжа.

Теорема [33]. Пусть существует точка X^*, в которой достигается экстремум функции *f(X)* при ограничениях $h_i(X) = 0,\ i = \overline{1,m}$. Если ранг матрицы $I=\left[\frac{\partial h_i}{\partial x_j}\right], i=\overline{1,m}; j=\overline{1,n}$ в точке X^* равен m, то существует m вещественных чисел $\lambda_1, \lambda_2, \ldots, \lambda_m$, ***не все из которых равны нулю одновременно***, при которых выполняется условие

$$\nabla f(X^*)+\sum_{i=1}^{m}\lambda_i^*\nabla h_i(X^*)=0.$$

Алгоритм метода Лагранжа

1. Составить функцию Лагранжа $L(X, \Lambda)$.
2. Найти частные производные функции Лагранжа (2.96).
3. Решить систему уравнений для определения X^* и Λ^*.
4. Исследовать полученные точки на максимум или минимум.

Пример практического применения метода Лагранжа

Содержательная постановка [6].

Цикл эксплуатации блока датчиков информационного зонда включает следующие фазы: $\tau_П$ – подготовки к применению; $\tau_Г$ – готовности к применению, $\tau_Р$ – работы (применения). Требование к надёжности формулируется в виде показателя безотказной работы R. Каковы должны

быть требования по надёжности к каждой фазе эксплуатации, если известны потери от отказа на каждой фазе, задаваемые неотрицательными величинами $C_i > 0$, $i = П, Г, Р$?

Время эксплуатации изделия в целом есть

$$\tau_\Sigma = \tau_П + \tau_Г + \tau_Р,$$

а надёжность характеризуется показателем

$$R = P_П \cdot P_Г \cdot P_Р,$$

где P_i, $i = П, Г, Р$ – вероятность безотказной работы на соответствующей фазе.

Если последствия отказов равнозначны, то задача имеет тривиальное решение:

$$P_i = \sqrt[3]{R}\ ,\ i = П, Г, Р.$$

В противном случае имеем оптимизационную задачу: необходимо рассчитать вероятностные характеристики таким образом, чтобы общие потери были минимальные. Функция цели или функция потерь имеет вид

$$C_\Sigma = C_П \cdot (1 - P_П) \cdot P_Г \cdot P_Р + C_Г \cdot P_П \cdot (1 - P_Г) \cdot P_Р + C_Р \cdot P_П\ P_Г \cdot (1 - P_р),$$

при ограничении

$$R = P_П\, P_Г\, P_Р\ .$$

Таким образом, имеем оптимизационную задачу, которую будем решать изложенным выше методом.

Функция Лагранжа получается такова:

$$L = C_\Sigma + \lambda[P_П \cdot P_Г \cdot P_Р - R].$$

В ходе применения метода, λ из первого, например, уравнения для частных производных $\dfrac{\partial L}{\partial P_i}$, подставляется в остальные.

$$\begin{cases} C_Г \cdot P_П - C_П \cdot P_Г = 0, \\ C_Г \cdot P_Р - C_Р \cdot C_Г = 0, \\ R = P_П \cdot P_Г \cdot P_Р. \end{cases} \Rightarrow \begin{cases} P_Р = \dfrac{C_П \cdot R}{C_Г \cdot P_П^2}, \\ P_П = \dfrac{C_Р \cdot R}{C_Г \cdot P_Р^2}. \end{cases}$$

Далее следуют алгебраические преобразования. Окончательно имеем оптимум при переменных

$$P_П^* = \sqrt[3]{\frac{C_П^2 R}{C_Г C_Р}}, \qquad P_Г^* = \sqrt[3]{\frac{C_Г^2 R}{C_П C_Р}}, \qquad P_Р^* = \sqrt[3]{\frac{C_Р^2 R}{C_П C_Г}}.$$

Иногда стоимость потерь при отказах удобно выражать в относительных единицах:

$$a_1 = \frac{C_Г}{C_П}, \quad a_2 = \frac{C_П}{C_Р}, \quad a_3 = \frac{C_Р}{C_Г}, \text{ причём } a_1 a_2 a_3 = 1.$$

Тогда решение в оптимуме примет вид

$$P_П^* = \sqrt[3]{\frac{Ra_2}{a_1}}, \; P_Г^* = \sqrt[3]{\frac{Ra_1}{a_3}}, \; P_Р^* = \sqrt[3]{\frac{Ra_3}{a_2}}.$$

2.5.4. Общий случай задачи нелинейного программирования [12, 68]

Задачи общего вида подразделяются на задачи ***выпуклого*** и ***вогнутого*** программирования, но, независимо от их типа, на переменные накладываются условия неотрицательности $x_j \geq 0$, $j = 1,n$.

Математическая модель для задачи выпуклого программирования имеет вид

$$\begin{aligned} & f(x_1, x_2, \ldots, x_n) \to \min, \\ & \begin{cases} g_1(x_1, x_2, \ldots, x_n) \leq 0, \\ g_2(x_1, x_2, \ldots, x_n) \leq 0, \\ \ldots \\ g_m(x_1, x_2, \ldots, x_n) \leq 0, \\ x_j \geq 0, \qquad j = \overline{1, n}. \end{cases} \end{aligned} \tag{2.98}$$

когда функции $f(X)$ и все $g_i(X)$, $i = 1, m$ выпуклы (в смысле НП-задач).

Для случая вогнутого программирования характерно

$$f(x_1,x_2,...,x_n) \to \max,$$
$$\begin{cases} g_1(x_1,x_2,...,x_n) \ge 0, \\ g_2(x_1,x_2,...,x_n) \ge 0, \\ ... \\ g_m(x_1,x_2,...,x_n) \ge 0, \\ x_j \ge 0, \qquad j = \overline{1,n}. \end{cases}$$

при функциях $f(X)$ и всех $g_i(X)$, $i = 1,m$ вогнутых (выпуклых вверх).

Связь между задачами выпуклого и вогнутого программирования в части целевой функции и системы ограничений определяется множителем (– 1). То есть

$$\max U(X) = \min [-U(X)].$$

Поэтому, обозначая

$$\hat{f}(X) = -f(X),$$
$$\hat{g}_i(X) = -g_i(X), \quad i = \overline{1,n}.$$

перейдём к следующей задаче выпуклого программирования:

$$\hat{f}(x_1,x_2,...,x_n) \to \min,$$
$$\begin{cases} \hat{g}_1(x_1,x_2,...,x_n) \le 0, \\ \hat{g}_2(x_1,x_2,...,x_n) \le 0, \\ ... \\ \hat{g}_m(x_1,x_2,...,x_n) \le 0, \\ x_j \ge 0, \qquad j = \overline{1,n}. \end{cases}$$

Рассмотрим общий случай задачи НП-программирования (2.98):

Если в точке минимума X^* неравенство $g_i(X)$ выполняется как ***равенство***, то оно называется ***активным***. В этом случае, согласно теореме Лагранжа, формула (2.97), для активных ограничений имеем

$$\nabla f(X^*) + \sum_I \lambda_i \nabla g_i(X^*) = 0, \qquad I = \{i : g_i(X^*) = 0\}. \tag{2.99}$$

Введём в рассмотрение вектор Λ с неотрицательными компонентами, обеспечивающими ***условие линейной независимости***

$$\sum_{i=1}^{m} \lambda_i g_i(X^*) = 0. \tag{2.100}$$

Условия (2.99) и (2.100) – суть условия ***регулярности*** ограничений.

Существует теорема Куна – Таккера.

Теорема[33]. Пусть функции $f(X)$ и $g_i(X)$, $i = 1, m$ обладают частными производными на некоторой области $\mathfrak{R}$, содержащей X^*. Точка X^* будет являться точкой минимума функции $f(X)$ при ограничениях $g_i(X) \le 0$, $i = 1, m$, удовлетворяющих условиям регулярности в виде линейной независимости, если существуют такие неотрицательные множители Лагранжа $\lambda_1, \lambda_2, \ldots, \lambda_m$, что

$$\begin{cases} \nabla f(X^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(X^*) = 0; \\ \sum_{i=1}^{m} \lambda_i g_i(X^*) = 0; \\ \lambda_i \ge 0, \ i = \overline{1, m}. \end{cases} \tag{2.101}$$

Ограничения (2.100) называются ***условиями дополняющей нежёсткости*** или условиями Слейтера, а точка X^* – точкой Куна-Таккера.

Смысл множителей Лагранжа в данной ситуации состоит в том, что они позволяют “нейтрализовать” (аннулировать) неактивные ограничения, которые и без того выполняются. Тем самым, задача нелинейного программирования общего вида сводится к задаче Лагранжа.

2.5.4.1. Седловая точка в НП-задачах

Определение. Пара векторов X^* и Λ^* называется ***седловой точкой*** функции $L(X, \Lambda)$ на области $\mathfrak{R}$, если для всех $\lambda \ge 0$ и $X \in \mathfrak{R}$ выполняется условие

$$L(X^*, \Lambda) \le L(X^*, \Lambda^*) \le L(X, \Lambda^*), \tag{2.102}$$

называемое ***неравенством седловой точки***.

В самой седловой точке выполняется равенство, называемое ещё неравенством(условием) максимина или мимнимакса.

$$\max_{\lambda_i \ge 0} \min_{X \in \Re} L(X, \Lambda) = \min_{X \in \Re} \max_{\lambda_i \ge 0} L(X, \Lambda), \quad i = \overline{1, n}.$$

Пример вида простейшей функции Лагранжа с седловой точкой показан на рисунке 2.25.

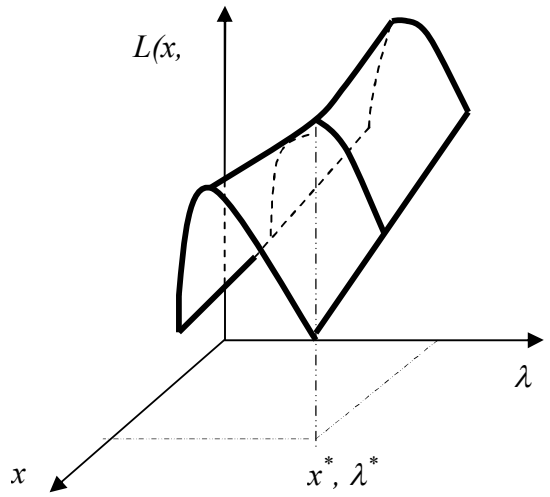


Рисунок 2.25 – Седло нелинейной функции

Существует нижеследующая теорема о седловой точке.

Теорема [33]. Пусть $f(X)$ и все $g_i(X)$, $i = 1, m$ выпуклы, а функции $g_i(X)$ удовлетворяют условиям ***регулярности Слейтера*** (вида $\exists_X \forall_i g_i(X) < 0$). Вектор X^* является решением задачи нелинейного программирования

$$\min f(X),$$
$$g_i(X) \le 0, \quad i = \overline{1, m}.$$

тогда и только тогда, когда существует такой вектор Λ с неотрицательными компонентами, что выполняются неравенство ***седловой точки*** (2.102) и ***условие линейной независимости*** (2.100).

Условие регулярности Слейтера означает наличие в области ограничений точек, в которых ограничения задачи выполняются строго.

Таким образом, НП-задача может быть, в принципе, сведена к задаче поиска седловой точки функции Лагранжа. При этом преференции будут отданы тому решению, ход решения которого более прост.

2.5.4.2. Применение теоремы Куна-Таккера к НП-задачам [68]

Имеем выпуклую задачу НП-программирования

$$\min f(X);$$
$$\begin{cases} g_i(X) \le 0, \quad i = \overline{1,m}; \\ x_j \ge 0, \qquad\qquad j = \overline{1,n}. \end{cases}$$

Введём дополнительные ограничения, учитывающие неотрицательность переменных x_j в явном виде: $x_j = -h_j(x_j)$, $j = 1, n$, откуда непосредственно следует, что $h_j(x_j) \le 0$, $j = 1, n$.

Для такой расширенной системы построим функцию Лагранжа

$$L(X,\Lambda,U) = f(X) + \sum_{i=1}^{m} \lambda_i g_i(X) + \sum_{j=1}^{n} u_j h_j(x_j),$$

к которой применим теорему Куна-Таккера, найдём $\nabla L(X, \Lambda, U)$. Это эквивалентно системе уравнений

$$\begin{cases} \dfrac{\partial L(X,\Lambda,U)}{\partial x_j} = \dfrac{\partial f(X)}{\partial x_j} + \sum_{i=1}^{m} \lambda_i \dfrac{\partial g_i(X)}{\partial x_j} - u_j = 0, \qquad j = \overline{1,n}; \\ u_j x_j = 0, \qquad j = \overline{1,n}; \\ \lambda_i g_i(X) = 0, \; i = \overline{1,m} \\ \lambda_i \ge 0, \qquad\qquad i = \overline{1,m}. \end{cases}$$

В точке Куна-Таккера будем иметь следующее:

$$\left\{ \begin{array}{l} \left\{ \begin{array}{l} \dfrac{\partial L(X^\circ,\Lambda^\circ)}{\partial x_j} \ge 0, \qquad j = \overline{1,n}; \\ \dfrac{\partial L(X^\circ,\Lambda^\circ)}{\partial x_j} \cdot x_j^\circ = 0, \; j = \overline{1,n}; \end{array} \right\} \\ \left\{ \begin{array}{l} \dfrac{\partial L(X^\circ,\Lambda^\circ)}{\partial \lambda_i} = g_i(X^\circ) \le 0, \; i = \overline{1,m}; \\ \lambda_i g_i(X^\circ) = 0, \qquad i = \overline{1,m}. \end{array} \right\} \end{array} \right.$$

Знак “≥” в первом неравенстве записан, потому, что не все значения u_j равны нулю, а первых два ограничения, выделенные “}”, есть условия дополняющей нежёсткости по u_j.

Для вогнутого программирования, благодаря симметрии задач получается

$$\left\{\begin{array}{l}\left.\begin{array}{ll}\dfrac{\partial L(X^{\circ},\Lambda^{\circ})}{\partial x_j}\le 0, & j=\overline{1,n};\\ \dfrac{\partial L(X^{\circ},\Lambda^{\circ})}{\partial x_j}\cdot x_j^{\circ}=0, & j=\overline{1,n};\end{array}\right\}\\ \left.\begin{array}{ll}\dfrac{\partial L(X^{\circ},\Lambda^{\circ})}{\partial \lambda_i}=g_i(X^{\circ})\ge 0, & i=\overline{1,m};\\ \dfrac{\partial L(X^{\circ},\Lambda^{\circ})}{\partial \lambda_i}\cdot \lambda_j^{\circ}=0,, & i=\overline{1,m}.\end{array}\right\}\end{array}\right.$$

Таким образом, теорема Куна-Таккера предоставляет исследователю удобный инструмент, позволяющий по условию задачи нелинейного программирования построить систему уравнений, решение которой даёт систему векторов X и Λ. Эти векторы дополнительно проверяются на выполнение условий линейной независимости и неотрицательности.

2.5.5. Методы возможных направлений

Указанные методы применяются для ***численного*** решения НП-задач.

Определение. Вектор с ненулевыми компонентами (ненулевой) S называется ***возможным направлением*** спуска в точке $X \in \mathfrak{R}$, если существует $\delta > 0$, такое, что $X + \lambda S \in \mathfrak{R}$ для всех $\lambda \in (0, \delta)$ и $f(X + \lambda S) < f(X)$.

Определение такого направления составляет краеугольный камень, лежащий в основании методов возможных направлений. Типичными представителями таковых являются алгоритмы, предложенные Зойтендейком (есть написание Заутендайка, в оригинале G. Zoutendijk) и Розеном.

2.5.5.1. Метод Зойтендейка [33, 34, 67, 68]

Существуют три разновидности указанного метода:

- при линейных ограничениях;
- при нелинейных ограничениях и
- улучшенной сходимости при нелинейных ограничениях.

Случай линейных ограничений

Постановка задачи, в этом случае, такова

$$\min f(X)$$
$$\begin{cases} AX \le b, \\ HX = h, \end{cases}$$

где $f(X)$ – нелинейна, $A[m \times n]$, $H[l \times n]$ – матрицы, а $b[m]$ и $h[l]$ – вектора системы линейных ограничений.

Пусть в текущей точке все ограничения со знаком “≤” представимы в виде

$$\begin{cases} A_1 X = b_1, \\ A_2 X < b_2. \end{cases}$$

При этом считаем, что исходные матрицы блочные (вернее подлежат разделению или перестановке) по знакам отношений

$$\begin{cases} A^T = \begin{bmatrix} A_1^T & A_2^T \end{bmatrix}, \\ b^T = \begin{bmatrix} b_1^T & b_2^T \end{bmatrix}. \end{cases}$$

Вектор S будет являться направлением спуска в точке X при выполнении условий:

$$\begin{cases} A_1 S \le 0, \\ HS = 0, \end{cases}$$

как это показано на рисунке 2.26 ниже.

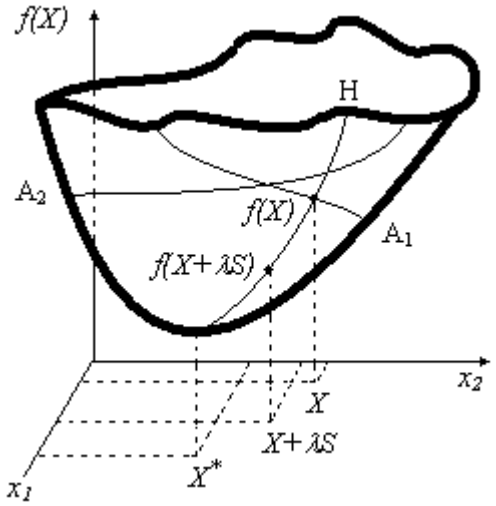


Рисунок 2.26 – К пояснению выбора направления

На изображении целевая функция показана жирной линией. Она напоминает кубок, линии на кубке есть проекции области ограничений на поверхность, описываемую целевой функцией. Из эскиза видно, что спуск должен выполняться «вдоль линии» (или не покидая гиперплоскости, в многомерном случае), по которой выполняются исходные ограничения вида «равенство», в сторону от активных ограничений (неравенства, выполняющиеся в точке в виде равенства, как нами было определено выше).

То есть, выбранное направление спуска S должно обеспечивать выполнение исходных ограничений модели $HX = h$ и устранять активность ограничений $A_1 X = b_1$.

На компоненты вектора S налагаются дополнительные ограничения (требования) нормировки:

- на величину элементов

$$-1 \le s_i \le 1,\ j = \overline{1,n}; \tag{2.103}$$

- на модуль вектора возможного направления S

$$S^T S \le 1, \|S\|^2 \le 1, \tag{2.104}$$

это ограничения объединяет условия $-1 \le s_j \le 1, j = \overline{1,n}$ и $\sum_{j=1}^{n} s_i \le 1, \sum_{j=1}^{n} s_i \ge -1;$

- на величину целевой функции ЗЛП

$$\nabla^T f(X) S \ge -1. \tag{2.105}$$

Поэтому возможна одна из трёх постановок задачи минимизации, которая может быть решена соответствующим методом линейного программирования:

$$\begin{array}{lll}
\min \nabla^T f(X) S & \min \nabla^T f(X) S & \min \nabla^T f(X) S \\
\begin{cases} A_1 S \le 0, \\ HS = 0, \\ s_j \le 1, \\ s_j \ge -1,\ j = 1, m. \end{cases} & \begin{cases} A_1 S \le 0, \\ HS = 0, \\ S^T S \le 1. \end{cases} & \begin{cases} A_1 S \le 0, \\ HS = 0, \\ \nabla^T f(X) S \ge -1. \end{cases} \\
1) & 2) & 3)
\end{array} \tag{2.106}$$

Алгоритм Зойтендейка для случая линейных ограничений

Предварительный этап.

Нахождение точки X, удовлетворяющей системе ограничений задачи.

Итерация.

1. Найти множество активных ограничений в текущей точке и композицию матрицы системы ограничений $A^T = \left[A_1^T \quad A_2^T \right]$ и решить ЗЛП (2.106) вида 1, 2 или 3, по желанию.

2. *Проверка условия окончания*.

Если оптимальное значение целевой функции ЗЛП равно нулю, ***то*** текущие координаты X определяют точку Куна-Таккера.

3. ***Иначе*** решить задачу одномерной минимизации Коши: $\min\limits_{0\le\lambda\le\lambda_{\max}} f(X+\lambda S)$, в которой граничное значение параметра λ определяется как

$$\lambda_{\max}=\begin{cases}\min\limits_k\left\{\dfrac{\widetilde{b}_k}{\widetilde{s}_k}\middle|\widetilde{s}_k>0\right\}, & \exists_k\widetilde{s}_k>0,\\ \infty, & \forall_k\widetilde{s}_k\le 0.\end{cases}\quad\text{где } \widetilde{b}=b_2-A_2X,\qquad \widetilde{S}=A_2S.$$

4. Перейти в новую текущую точку $X = X + \lambda \times S$.

Алгоритм не сложен, он представлен на рисунке 2.25.

Алгоритм Зойтендейка для случая нелинейных ограничений

Рассмотрим задачу НП-программирования вида

$$\min f(X);$$
$$\begin{cases} g_i(X)\le 0, & i=\overline{1,m};\\ x_j\ge 0, & j=\overline{1,n}.\end{cases}$$

Пусть текущая точка X является допустимой точкой, а $I=\{i: g_i(X)=0\}$ – множество ограничений, активных в этой точке. Если

$$\begin{cases}\nabla^T f(X)S<0,\\ \nabla^T g_i(X)S<0,\ i\in I,\end{cases}$$

то S – вектор возможного направления спуска в этой точке (смотри рисунок 2.26).

Алгоритм имеет топологию, аналогичную представленной рисунком 2.27.

Предварительный этап.

Найти начальную точку X, для которой выполняются ограничения задачи: $g_i(X)\le 0,\quad i=\overline{1,m}$.

Основной этап (итерация).

1. Найти множество активных ограничений задачи в текущей точке $I=\{i: g_i(X)=0\}$ и решить ЗЛП вида

$$\min z$$
$$\begin{cases} \nabla^T f(X)S - z \le 0, \\ \nabla^T g_i(X)S - z \le 0,\ i \in I = \{i : g_i(X) = 0\}. \end{cases}$$

при любом, оговоренном выше, условии нормировки компонентов вектора возможного направления S (2.103) – (2.105).

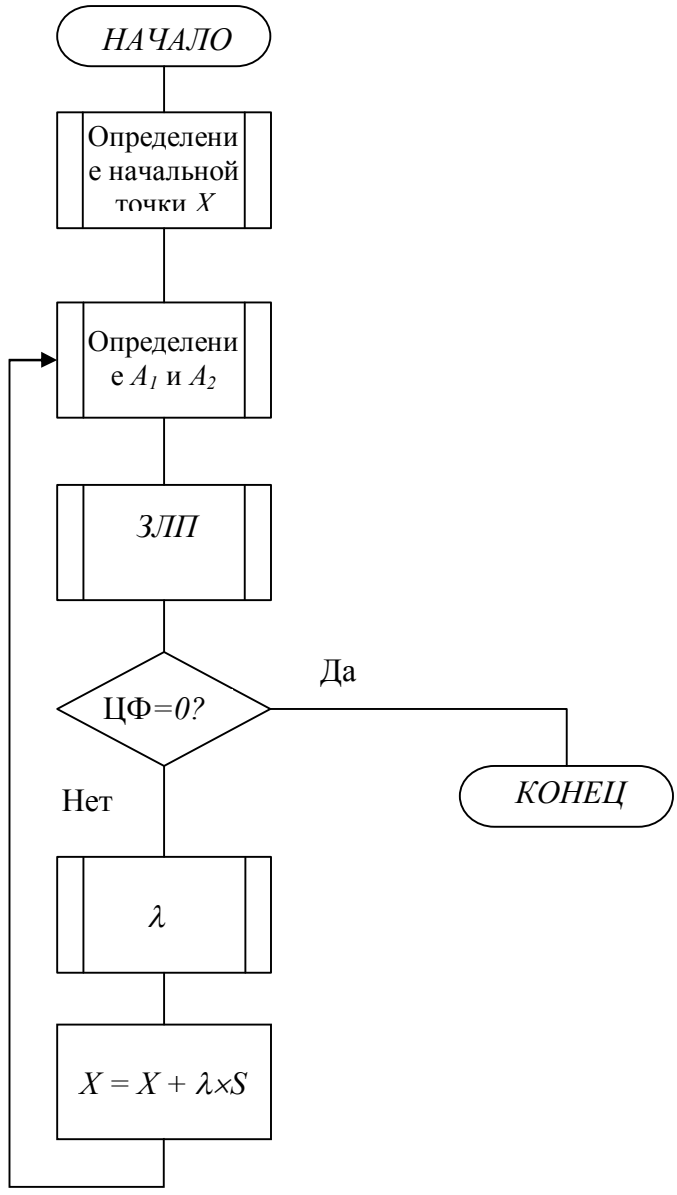


Рисунок 2.27– Граф-схема алгоритма Зойтендейка

2. *Проверка условия окончания*.

Если оптимальное значение целевой функции ЗЛП равно нулю, ***то*** X – оптимальное решение (точка Куна-Таккера) исходной НП-задачи.

3. *В противном случае* необходимо решить задачу одномерной минимизации Коши: $\min\limits_{0\le\lambda\le\lambda_{\max}} f(X+\lambda S)$, где параметр λ определяется из условия

$$\lambda_{\max} = \max\{\lambda : g_i(X+\lambda S) \le 0,\ i=1,m\}.$$

4. Переместится в следующую точку $X = X + \lambda \times S$ и продолжить решение.

Алгоритм Зойтендейка для случая нелинейных ограничений улучшенной сходимости

Шаги, генерируемые вдоль направления поиска, стремятся, по мере итераций, к нулю. Это может, в отдельных случаях, вызвать остановку вычислительного процесса без достижения оптимума. Чтобы избежать несанкционированной остановки, ограничения эквивалентной ЗЛП усиливают.

Модифицированная ЗЛП имеет вид

$$\min z$$
$$\begin{cases} \nabla^T f(X)S - z \le 0, \\ \nabla^T g_i(X)S - z \le -g_i(X),\ i=1,m. \end{cases}$$

При этом нововведения позволяют учитывать как активные, так и обычные ограничения в точке поиска. ЗЛП снабжается одним из условий нормировки (2.103) – (2.105).

2.5.5.2. Метод проекции градиента Розена [9, 68]

Метод предназначен для решения задач нелинейного программирования с линейной системой ограничений

$$\min f(X)$$
$$\begin{cases} AX \le b, \\ HX = h, \end{cases}$$

где $f(X)$ – нелинейная и ***дифференцируемая*** функция, A – матрица размером $[m \times n]$, H – $[l \times n]$-мерная матрица , а b – m-мерный вектор, а h – l-мерный вектор системы линейных ограничений.

Идея, лежащая в основе метода, состоит в следующем.

Пусть X – допустимая точка, удовлетворяющая системе ограничений, а направление спуска определяется градиентом $S = -\nabla f(X)$.

Движение вдоль направления спуска может, гипотетически, привести к ***нарушению допустимости***. Поэтому Розен предложил направление спуска строить по правилу

$$S = -P \times \nabla f(X),$$

где P – матрица проецирования или проецирующая (проектирующая) матрица, которая бы гарантировала сохранение допустимости текущей точки.

Пусть в допустимой точке часть неравенств активна, а другая часть – нет, то допускается представление системы ограничений по признаку активности в виде блочной матрицы:

$$\begin{aligned} &A_1 X = b_1, \quad A^T = \begin{bmatrix} A_1^T & A_2^T \end{bmatrix}, \\ &A_2 X < b_2, \quad b^T = \begin{bmatrix} b_1^T & b_2^T \end{bmatrix} \end{aligned} \tag{2.107}$$

Матрица проецирования P должна удовлетворять условию $P \times \nabla f(X) \neq 0$ и блокировать возможные направления в сторону активных ограничений. Это будет соблюдаться, когда проектирующая матрица определена как

$$P = I - M^T \times (M \times M^T)^{-1} \times M, \tag{2.108}$$

где I – единичная матрица, $M^T = [A_1{}^T H^T]$ – невырождена. Отметим, что выполняется тождество $M \times P = 0$, то есть $A_1 \times P = 0$ и $H \times P = 0$, таким образом, спуск в направлении ограничений, выполняемых как равенство, не производится.

Рассмотрим точку, в которой выполняется условие $P \times \nabla f(X) = 0$. Имеем

$$[I - M^T \times (M \times M^T)^{-1} \times M] \times \nabla f(X) = \nabla f(X) + M^T \times W = 0,$$

где $W^T = [U^T V^T]$, а

$$\nabla f(X) + A_1{}^T \times U + H^T \times V = 0.$$

Если компоненты вектора U неотрицательны, то X является точкой Куна-Таккера. В противном случае, можно построить новое направление спуска

$$S = -\widetilde{P} \times \nabla f(X),$$

где $\widetilde{P} = I - \widetilde{M}^T \times (\widetilde{M} \times \widetilde{M}^T)^{-1} \times \widetilde{M}$, в котором $\widetilde{M}^T = [\widetilde{A}_1^T H^T]$, а $\widetilde{A}_1$ получено из A_1 путём вычёркивания строк, соответствующих компонентам $u_j < 0$.

Схема алгоритма показана на рисунке 2.28.

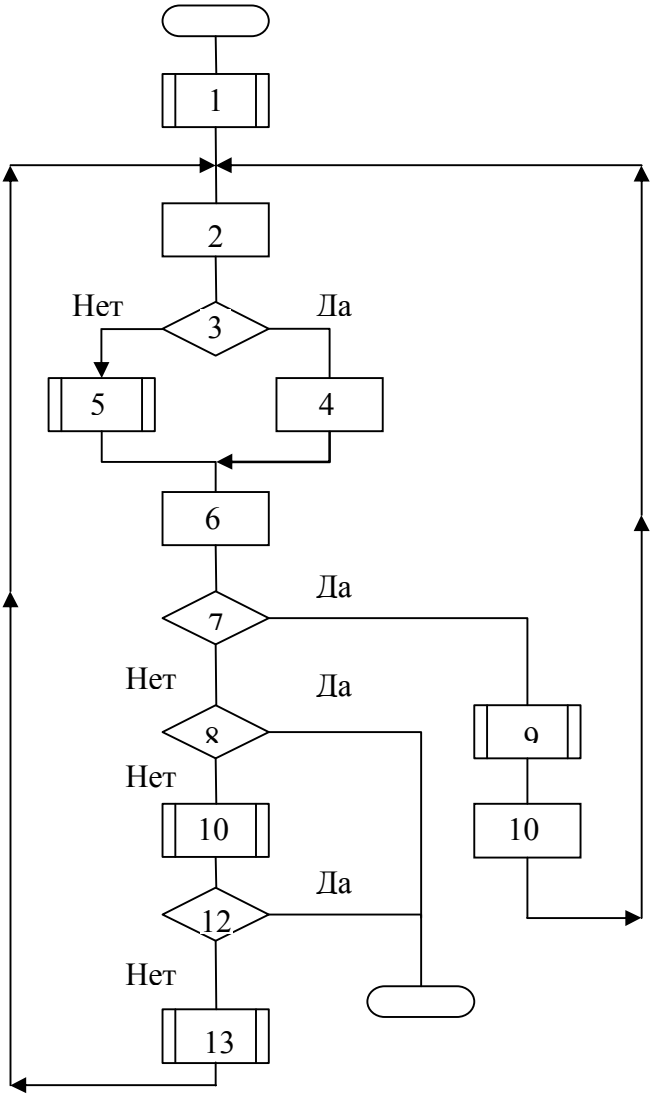


Рисунок 2.28 – Граф-схема алгоритма Розена

Блок № 1 выполняет выбор начальной точки X, удовлетворяющей системе ограничений задачи, отыскивает активные ограничения и разложение ограничений с неравенствами (2.107).

В блоке № 2 конструируется матрица $M^T = [A_1{}^T H^T]$, которая проверяется на равенство нулю всех её элементов условием блока № 3: $M = 0?$, если матрица вся нулевая (выход «да» блока № 3), то проецирующая матрица полагается равной единичной (блок № 4).

В противном случае (выход «нет» блока № 3), выполняется расчёт проецирующей матрицы в блоке № 5 по формуле (2.108).

После определения элементов проецирующей матрицы, рассчитывается возможное направление спуска $S = -P \times \nabla f(X)$ в блоке № 6.

Блок № 7 выполняет проверку: $S \neq 0?$, то есть, не является ли текущая точка точкой Куна-Таккера.

Если все компоненты вектора возможного направления нулевые (выход «нет» блока № 7), то это, возможно, искомое решение. Чтобы убедиться в этом, проверяется равенство нулю всех компонентов блочной матрицы M (блок № 8, $M = 0?$), и если это так (выход «да» блока № 8), то решение успешно заканчивается.

Если среди компонентов матрицы M есть ненулевые элементы (выход «нет» блока № 8), то ищется разложение $W^T = [U^T V^T]$, блок № 10.

Если компоненты подматрицы U неотрицательны, что проверяется блоком № 12 ($\forall_i u_i \geq 0?$), то X является точкой Куна-Таккера (выход «да»).

Если компоненты U отрицательны (выход «нет» блока № 12), то переопределяется матрицы $A_1 = \widetilde{A}_1$ и блочная матрица $\widetilde{M}^T = [\widetilde{A}_1^T H^T]$ (блок № 13).

Если не все элементы вектора возможного направления нулевые (выход «да» блока № 7), рассчитывается новая текущая точка. Сперва выбирается параметр λ решением задачи одномерной минимизации Коши (блок № 9): $\min\limits_{0 \leq \lambda \leq \lambda_{\max}} f(X + \lambda S)$, в которой граничное значение параметра λ определяется, как и в методе Зойтендейка:

$$\lambda_{\max} = \begin{cases} \min\limits_k \left\{ \dfrac{\widetilde{b}_k}{\widetilde{s}_k} \middle| \widetilde{s}_k > 0 \right\}, & \exists_k \widetilde{s}_k > 0, \\ \infty, & \forall_k \widetilde{s}_k \leq 0. \end{cases} \quad \text{где } \widetilde{b} = b_2 - A_2 X, \qquad \widetilde{S} = A_2 S.$$

Значение очередной текущей точки рассчитывается в блоке № 11 по рекуррентной формуле $X = X + \lambda \times S$.

2.5.6. Методы штрафных функций [46, 67, 68]

Методы штрафных функций основаны на практике перехода от задачи ***условной минимизации*** к задаче ***безусловной минимизации*** путём построения специальной штрафной функции.

Известны [3, 9, 24, 33, 40, 56] ***параметрические*** и ***непараметрические*** методы построения штрафных функций в виде полинома

При использовании ***параметрических*** методов, штрафной полином конструируется с использованием выражений, описывающих ограничения, в качестве параметров других функций (функционалов) и весовых коэффициентов.

В *непараметрическ*их методах функция рассматривается в качестве дополнительного ограничения, которое постоянно усиливается в процессе решения.

По характеру перемещения точки к оптимуму различают:

- методы внутренней точки,
- методы внешней точки и
- комбинированные методы.

При использовании методов ***внутренней точки***, последовательные приближения к оптимуму производятся ***внутри области***, определяемой ограничениями, благодаря специальной функции штрафа, называемой ***барьерной***.

Когда поиск оптимума осуществляется по методу ***внешней точки***, текущее решение находится ***за пределами области*** ограничений, попадая вовнутрь её на последнем шаге итераций.

Комбинированные методы используют, когда большинство ограничений задачи имеют вид равенства, и, в процессе решения, попеременно, одни ограничения выполняются, а другие нет.

Искомое решение получается, при удовлетворении заданных условий, в пределах отведённого допуска.

Пусть условия задачи имеют вид

$$\min f(X);$$
$$\begin{cases} h_i(X) = 0, & i = \overline{1, m}; \\ g_i(X) \geq 0, & i = \overline{m+1, l}. \end{cases}$$

Пользуясь параметрическим методом, по этим условиям можно построить следующую фунцию без ограничений:

$$P(X,\rho) = f(X) + \sum_{i=1}^{m} \rho_i H[h_i(X)] + \sum_{i=m+1}^{l} \rho_i G[g_i(X)],$$

где $P(X,\rho)$ – штрафная функция; ρ_i, $i = \overline{1,l}$ – весовые коэффициенты, отражающие значимость соблюдения того или иного ограничения; $H[h_i(X)]$ и $G[g_i(X)]$ – некоторые функционалы.

Рассмотрим, какими свойствами должны обладать эти функционалы.

Функционал $H[h_i(X)]$ должен сделать невыгодным любое отклонение аргумента X от поверхности $h_i(X) = 0$, то есть

$$\lim_{h_i(X)\to 0} H[h_i(X)] = 0, \qquad i = \overline{1,m}.$$

Поэтому в качестве функционала выбирают чётную степенную функцию

$$H[y] = y^p, \ p = 2,4,... \text{ либо } H[y] = |y|^p, \ p = 1,2,...$$

Функционал $G[g_i(X)]$ зависим от местоположения текущей точки в процессе решения.

- **Метод внутренней точки**:

$$\lim_{g_i(X)\to 0^+} G[g_i(X)] = \infty, \qquad i = \overline{m+1,l} \text{ для } g_i(X) > 0.$$

- **Метод внешней точки**:

$$\lim_{g_i(X)\to 0^-} G[g_i(X)] = 0, \qquad i = \overline{m+1,l} \text{ для } g_i(X) < 0.$$

- **Комбинированный метод**:

$$G[g_i(X)] > 0 \text{ для } g_i(X) < 0, \ i = \overline{m+1,l},$$
$$G[g_i(X)] = 0 \text{ для } g_i(X) = 0, \ i = \overline{m+1,l}.$$

При этом, не зависимо от траектории движения точки, в процессе решения задачи необходимо обеспечение выполнения условий:

$$\lim_{k\to\infty} \sum_{i=1}^{m} \rho_{i,k} H[h_i(X_k)] = 0,$$

$$\lim_{k\to\infty}\sum_{i=m+1}^{l}\rho_{i,k}G[g_i(X_k)]=0,$$
$$\lim_{k\to\infty}\left|P(X_k,\rho_k)-f(X_k)\right|=0,$$

где k – номер итерации.

Особо подчеркнём, что штрафная функция должна быть построена таким образом, что ***невыполнение*** какого либо их ограничений должно приводить ***к резкому возрастанию*** её в целом, и, по мере приближению к оптимуму, ***влияние*** штрафных добавок ***уменьшается***.

2.5.6.1 Метод барьерных поверхностей (МБП) [9, 34, 46, 68]

Данный алгоритм (рисунок 2.29) относится к группе методов внутренней точки, используется для решения задач вида

$$\min f(X);$$
$$\begin{cases} g_i(X)\ge 0, & i=\overline{1,m};\\ x_j\ge 0, & j=\overline{1,n}. \end{cases}$$

По условиям задачи строится штрафная функция вида

$$P(X,r,\Omega)=f(X)+r\cdot\sum_{i=1}^{m}\omega_i G[g_i(X)]. \qquad (2.109)$$

В формуле (2.109) использованы: параметр $r > 0$, убывающий по мере вычислений и Ω – вектор положительных весовых коэффициентов, учитывающих важность (значимость) ограничений модели.

Функции барьера $G[g_i(X)]$ выбирается из альтернатив

$$G[Y]=1/Y \text{ или } G[Y]=-\ln[Y].$$

Легко видеть, что $\lim_{Y\to 0^+}G[Y]=\infty$, то есть функция барьера, при приближении к нему изнутри области, неограниченно возрастает. Штрафная добавка к функции цели определяется формулой

$$\Delta = r \cdot \sum_{i=1}^{m} \omega_i G[g_i(X)]$$

Функционирование алгоритма представлено на рисунке 2.29, в его работе использована уменьшающая константа $0 < \beta < 1$, точность расчётов задаётся константой ε.

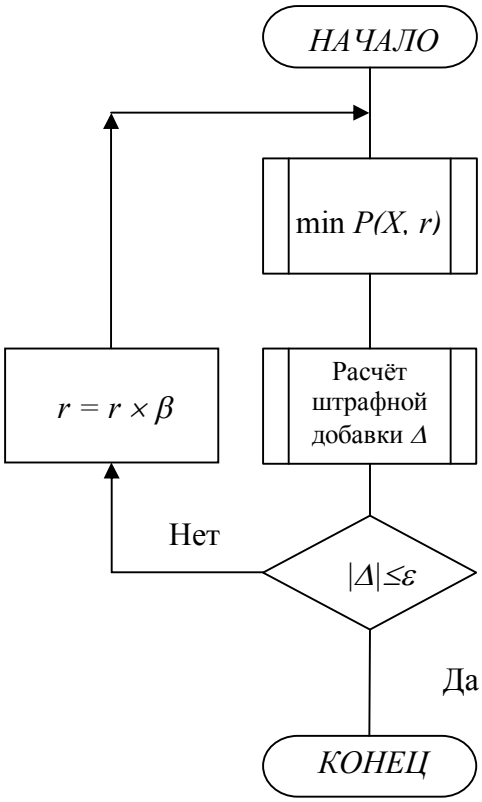


Рисунок 2.29 – Схема алгоритма МБП

Замечания:

- область ограничений должна быть непустой: $X_{opt} \in \{X: g_i(X) > 0, i = 1, m\}$;
- начальное значение X должно быть допустимой точкой;
- метод критичен по отношению к параметру r: если значение r мало, то будет отыскан оптимум функции, не совпадающий с истинным, а если значение r велико, то возрастёт число итераций;
- аналогично, близкие к единице значения β могут привести к преждевременному прекращению вычислений.

2.5.6.2 Метод внешней точки [3, 34, 46, 68]

Ориентируется на решение НП-задач в общей постановке

$$\min f(X);$$
$$\begin{cases} h_i(X) = 0, & i = 1, \overline{m}; \\ g_i(X) \geq 0, & i = m+1, \bar{l}. \end{cases}$$

Штрафная функция описывается выражением

$$P(X, r) = f(X) + rL(X),$$

где r – коэффициент штрафа, а величина штрафа в текущей точке X есть

$$L(X) = \sum_{i=1}^{m} H[h_i(X)] + \sum_{i=m+1}^{l} G[g_i(X)].$$

Используются функционалы:

$$G[Y] = [\max\{0; -Y\}]^P, \; P > 0, \text{ целое},$$
$$H[Y] = |Y|^P, \; P > 0, \text{ целое}.$$

Легко видеть, что функционал $G[Y]$ обращается в нуль при попадании текущей точки в область ограничений.

Алгоритм метода полностью дублирует алгоритм метода барьерных поверхностей, показанный на рисунке 2.29, но, в отличие от него, используется параметр увеличения штрафа β больше единицы.

Если взять параметры r и β достаточно большИми, то минимизация штрафной функции будет выполняться за счёт функции цели и величины штрафа. Поэтому, в ходе расчётов, коэффициент штрафа будет постоянно возрастать, а штраф – убывать. Следовательно, положив r и β порядка десятков, сотен, а то и тысяч, можно получить решение задачи с приемлемой точностью.

Алгоритм, сам по себе, обладает хорошей сходимостью и устойчивостью. Очевидно, что вельми проблемным местом его функционирования будет процедура минимизации штрафной функции, что является нетривиальной задачей.

2.5.7. Задачи квадратичного программирования

Квадратичное программирование – специальный класс НП-задач, в которых $f(X)$ – квадратичная (не выше второй степени переменных) вогнутая (выпуклая вверх) функция, а все ограничения $g_i(X)$, $i = 1,m$ – линейны [33].

Математическая модель такой задачи выглядит следующим образом [3, 23, 31, 33, 34, 36, 68]

$$\begin{aligned} &f(x) = b^T X + \frac{1}{2} X^T C X \to \max, \\ &AX \le A_0,\ X \ge 0, \end{aligned} \tag{2.110}$$

где C — симметричная отрицательно определённая матрица размерностью $[n \times n]$, b^T — вектор-строка размерностью $[1 \times n]$, A — матрица системы ограничений размерностью $[m \times n]$, A_0 — вектор свободных членов системы ограничений размерностью $[m \times 1]$, n — число переменных, m – число ограничений.

Задача решается путём применения теоремы Куна-Таккера, в результате чего получается система линейных ограничений, которую можно решить симплекс-методом.

Функция Лагранжа, построенная по условиям задачи, имеет вид

$$L(X,\Lambda) = b^T \cdot X + \frac{1}{2} X^T \cdot C \cdot X + \Lambda^T (A_0 - A \cdot X).$$

Воспользуемся ранее полученными результатами из пункта 2.5.4.2. Имеем систему уравнений:

$$\begin{cases} \dfrac{\partial L(X,\Lambda)}{\partial X} = b + C \cdot X - A^T \Lambda \le 0, \\ \dfrac{\partial L(X,\Lambda)}{\partial \Lambda} = A_0 - A \cdot X \ge 0. \end{cases}$$

Приведя её к каноническому виду и добавив условия дополняющей нежёсткости, получим констатирующую формулировку нижеследующей теоремы.

Теорема квадратичного программирования [33]. Вектор $X_0 \ge 0$ является оптимальным решением задачи квадратичного программирования

тогда и только тогда, когда существуют такие m-мерные вектора $\Lambda > 0$, $W \geq 0$ и n-мерный вектор $V \geq 0$, что выполняются следующие условия

$$\left.\begin{array}{ll} b + C \cdot X_0 - A^T \Lambda + V = 0, & \text{а)} \\ A_0 - AX_0 - W = 0, & \text{б)} \\ V^T X_0 = 0 & \\ W^T \Lambda = 0 & \text{в)} \end{array}\right\}. \qquad (2.111)$$

Компоненты всех векторов Λ, W и V — неотрицательны, а вектора W и V могут быть нулевыми. Условия (2.111, а) и (2.111, б) образуют систему из $n + m$ уравнений для $2 \times (n + m)$ неизвестных компонентов X, Λ, V и W. Ограничения (2.111, в) есть условия дополняющей нежёсткости.

Алгоритм решения задачи квадратичного программирования

1. Условия (2.111, а) и (2.111, б) необходимо представить в форме, обеспечивающей положительность элементов столбцов свободных членов

$$\begin{cases} A^T \Lambda - C \cdot X - V = b, \\ AX + W = A_0. \end{cases} \qquad (2.112)$$

2. Поскольку знак в ограничениях (2.112) – равенство, следует воспользоваться методом искусственного базиса. Необходимо добавить искусственные переменные $\{y_i\}$ в первую группу уравнений, и $\{z_i\}$, при этом система ограничений примет вид

$$\begin{cases} A^T \Lambda - C \cdot X - V + Z = b, \\ AX + W + Y = A_0. \end{cases} \qquad (2.113)$$

3. Конструируется псевдоцелевая функция [3] из искусственных переменных

$$\sum_{i=1}^{m} \mu \cdot y_i + \sum_{j=1}^{n} \mu \cdot z_j \to \min . \qquad (2.114)$$

4. Решается эквивалентная ЗЛП с функцией цели (2.114) при ограничениях (2.113).

Если в ходе решения ЗЛП векторы Y и Z будут выведены из базиса в полном составе (достигнут оптимум), а полученные значения X, Λ, V и W не отрицательны, а так же удовлетворяют (2.111, в), то компоненты вектора X представляют собой оптимальное решение исходной задачи квадратичного программирования (2.110).

5. Рассчитывается значение целевой функции.

Пример.

Пусть компоненты условия (2.110) выглядят так

$$A=\begin{pmatrix}2 & 5\\ 3 & 2\\ 1 & 1\end{pmatrix},\ A_0=\begin{pmatrix}10\\ 6\\ 4\end{pmatrix},\quad b^T=\begin{pmatrix}2 & 3\end{pmatrix},\ C=\begin{pmatrix}-\frac{1}{2} & \frac{1}{4}\\ \frac{1}{4} & -\frac{1}{2}\end{pmatrix}.$$

Матрица C симметричная, а об отрицательной определённости можно судить по критерию Сильвестра – знаки минорных определителей должны чередоваться [20, 35, 36].

$$C=\begin{pmatrix}-\frac{1}{2} & \frac{1}{4}\\ \frac{1}{4} & -\frac{1}{2}\end{pmatrix},\quad \Delta_1=c_{11}=-\frac{1}{2}<0,\quad \Delta_2=\begin{vmatrix}-\frac{1}{2} & \frac{1}{4}\\ \frac{1}{4} & -\frac{1}{2}\end{vmatrix}=\frac{1}{4}-\frac{1}{16}=\frac{3}{16}>0.$$

Следовательно, матрица C определена отрицательно.

Построение развёрнутой модели (2.110), соответствующей нашим условиям, даёт результат

$$f(x_1,x_2)=2\cdot x_1+3\cdot x_2+\frac{1}{4}\cdot x_1\cdot x_2-\frac{1}{4}\cdot x_1^2-\frac{1}{4}\cdot x_2^2\to\max$$

$$\begin{cases}2\cdot x_1+5\cdot x_2\le 10,\\ 3\cdot x_1+2\cdot x_2\le 6,\\ 1\cdot x_1+1\cdot x_2\le 4,\\ x_1\ge 0, x_2\ge 0,\end{cases}\quad\text{или}\quad\begin{cases}10-2\cdot x_1-5\cdot x_2\ge 0,\\ 6-3\cdot x_1-2\cdot x_2\ge 0,\\ 4-1\cdot x_1-1\cdot x_2\ge 0,\\ x_1\ge 0, x_2\ge 0.\end{cases}$$

Правая запись системы ограничений соответствует формальному виду, применяемому для представления ограничений в задачах вогнутого программирования.

Исследуем целевую функцию данной задачи на экстремум, найдём координаты стационарной точки.

$$\begin{cases} \dfrac{\partial f(x_1,x_2)}{\partial x_1} = 2 + \dfrac{1}{4}\cdot x_2 - \dfrac{1}{2}\cdot x_1 = 0, \\ \dfrac{\partial f(x_1,x_2)}{\partial x_2} = 3 + \dfrac{1}{4}\cdot x_1 - \dfrac{1}{2}\cdot x_2 = 0. \end{cases} \Rightarrow \begin{cases} x_1 = 2\left(2 + \dfrac{1}{4}\cdot x_2\right), \\ 3 + \dfrac{1}{2}\left(2 + \dfrac{1}{4}\cdot x_2\right) - \dfrac{1}{2}\cdot x_2 = 0. \end{cases} \Rightarrow X^* = \begin{pmatrix} \dfrac{28}{3} \\ \dfrac{32}{4} \end{pmatrix}$$

Полученная точка, в которой наблюдается безусловный глобальный максимум, лежит за пределами области ограничений, значение целевой функции в этой точке есть $f(X^*) = \dfrac{196}{9} \cong 21{,}7$.

По условиям задачи может быть составлена функция Лагранжа

$$L(X,\Lambda) = 2\cdot x_1 + 3\cdot x_2 + \frac{1}{4}\cdot x_1\cdot x_2 - \frac{1}{4}\cdot x_1^2 - \frac{1}{4}\cdot x_2^2 \quad +\lambda_1(10 - 2\cdot x_1 - 5\cdot x_2) + \\ +\lambda_2(6 - 3\cdot x_1 - 2\cdot x_2) + \\ +\lambda_3(4 - 1\cdot x_1 - 1\cdot x_2).$$

Применение подхода, связанного с поисками безусловного экстремума функции без ограничений (функции Лагранжа) в задачах вогнутого программирования, позволяет получить систему ограничений для поиска значений X и Λ, являющихся координатами экстремальной точки.

$$\begin{cases} \dfrac{\partial L(X,\Lambda)}{\partial x_1} = 2 + \dfrac{1}{4}\cdot x_2 - \dfrac{1}{2}\cdot x_1 - 2\cdot\lambda_1 - 3\cdot\lambda_2 - 1\cdot\lambda_3 \le 0, \\ \dfrac{\partial L(X,\Lambda)}{\partial x_1} = 3 + \dfrac{1}{4}\cdot x_1 - \dfrac{1}{2}\cdot x_2 - 5\cdot\lambda_1 - 2\cdot\lambda_2 - 1\cdot\lambda_3 \le 0, \\ \dfrac{\partial L(X,\Lambda)}{\partial \lambda_1} = 10 - 2\cdot x_1 - 5\cdot x_2 \ge 0, \\ \dfrac{\partial L(X,\Lambda)}{\partial \lambda_2} = 6 - 3\cdot x_1 - 2\cdot x_2 \ge 0, \\ \dfrac{\partial L(X,\Lambda)}{\partial \lambda_3} = 4 - 1\cdot x_1 - 1\cdot x_2 \ge 0. \end{cases}$$

Для канонизации системы ограничений добавим с соответствующими знаками компоненты векторов V и W. Получим следующую каноническую форму системы ограничений:

$$\begin{cases} 2+\frac{1}{4}\cdot x_2-\frac{1}{2}\cdot x_1-2\cdot\lambda_1-3\cdot\lambda_2-1\cdot\lambda_3+1\cdot v_1=0, \\ 3+\frac{1}{4}\cdot x_1-\frac{1}{2}\cdot x_2-5\cdot\lambda_1-2\cdot\lambda_2-1\cdot\lambda_3+1\cdot v_2=0, \\ 10-2\cdot x_1-5\cdot x_2-1\cdot w_1=0, \\ 6-3\cdot x_1-2\cdot x_2-1\cdot w_2=0, \\ 4-1\cdot x_1-1\cdot x_2-1\cdot w_3=0, \\ x_1\ge 0, x_2\ge 0,\ \Lambda>0, V\ge 0, W\ge 0. \end{cases}$$

Сопоставив полученные выражения с выражениями (2.111, а, б) теоремы квадратичного программирования, отметим их полное соответствие.

Пункт № 1 алгоритма.

Преобразуем данную каноническую форму системы ограничений, так, чтобы в столбце свободных членов находились бы положительные элементы.

$$\begin{cases} \frac{1}{2}\cdot x_1-\frac{1}{4}\cdot x_2+2\cdot\lambda_1+3\cdot\lambda_2+1\cdot\lambda_3-1\cdot v_1=2, \\ -\frac{1}{4}\cdot x_1+\frac{1}{2}\cdot x_2+5\cdot\lambda_1+2\cdot\lambda_2+1\cdot\lambda_3-1\cdot v_2=3, \\ 2\cdot x_1+5\cdot x_2+1\cdot w_1=10, \\ 3\cdot x_1-2\cdot x_2+1\cdot w_2=6, \\ 1\cdot x_1+1\cdot x_2+1\cdot w_3=4, \\ x_1\ge 0, x_2\ge 0,\ \Lambda>0, V\ge 0, W\ge 0. \end{cases}$$

Пункты № 2,3 алгоритма.

Построим эквивалентную ЗЛП, введя псевдоцелевую функцию и модифицировав систему ограничений. Функция

$$\mu\cdot y_1+\mu\cdot y_2+\mu\cdot z_1+\mu\cdot z_2+\mu\cdot z_3\to\min$$

Наименование “псевдоцелевая” функция получила, поскольку составляется только из искусственных переменных, а оптимальное решение, в случае его существования, приведёт к обращению её в нуль.

Система ограничений после введения искусственных переменных примет вид.

$$\begin{cases} \frac{1}{2}\cdot x_1 - \frac{1}{4}\cdot x_2 + 2\cdot\lambda_1 + 3\cdot\lambda_2 + 1\cdot\lambda_3 - 1\cdot v_1 + y_1 = 2, \\ -\frac{1}{4}\cdot x_1 + \frac{1}{2}\cdot x_2 + 5\cdot\lambda_1 + 2\cdot\lambda_2 + 1\cdot\lambda_3 - 1\cdot v_2 + y_2 = 3, \\ 2\cdot x_1 + 5\cdot x_2 + 1\cdot w_1 + z_1 = 10, \\ 3\cdot x_1 - 2\cdot x_2 + 1\cdot w_2 + z_2 = 6, \\ 1\cdot x_1 + 1\cdot x_2 + 1\cdot w_3 + z_3 = 4, \\ x_1 \ge 0, x_2 \ge 0,\ \Lambda > 0, V \ge 0, W \ge 0,\ Y \ge 0, Z \ge 0. \end{cases}$$

Поскольку переменных *X*, *Λ,* V, W, Z много больше числа ограничений, в качестве алгоритма решения ЗЛП следует модифицированный симплекс-метод.

Пункт № 4 алгоритма.

Подробный ход расчётов опускается, приводится конечный результат решения, опровергнуть или подтвердить который Читатель может самостоятельно, перерешав ЗЛП.

Координаты оптимума составляют:

$$x_1 = \frac{10}{11}, x_2 = \frac{18}{11}, \lambda_1 = \frac{103}{242}, \lambda_2 = \frac{109}{242}, \lambda_3 = 0, v_1 = 0, v_2 = 0, w_1 = 0, w_2 = 0, w_3 = \frac{16}{11}.$$

Элементы векторов *V* и *W* неотрицательны, среди элементов вектора *Λ* – есть положительные (не все равны нулю по Куну-Таккеру).

Так как $x_1 \cdot v_1 = 0,\ x_2 \cdot v_2 = 0,\ \lambda_1 \cdot w_1 = 0,\ \lambda_2 \cdot w_2 = 0,\ \lambda_3 \cdot w_3 = 0,$ то условие дополняющей нежёсткости выполнено, и решение задачи квадратичного программирования находится в крайней точке множества допустимых стратегий, значение функции равно $f\left(\frac{10}{11}, \frac{18}{11}\right) = \frac{753}{121} \cong 6{,}223.$

Вопросы для самоконтроля

1. Как выглядит в общем виде модель для задачи нелинейного программирования?
2. Какие существуют типы задачи НП, чем они различаются?
3. Как формулируются необходимые условия оптимальности в задачах безусловной оптимизации?

4. Как найти наибольшее и наименьшее значения функции нескольких переменных в замкнутой ограниченной области?
5. Что называется частной производной первого порядка функции нескольких переменных?
6. Как следует понимать термины «выпуклая» и «вогнутая» функции применительно к задачам нелинейного программирования?
7. Что называется линией уровня функции двух переменных?
8. Что называется поверхностью уровня функции трех переменных?
9. Что такое частная производная второго порядка функции нескольких пе-
10. ременных?
11. Что называется стационарной точкой?
12. Какие численные методы существуют для вычисления частных производных?
13. Что такое критическая точка функции нескольких переменных?
14. Обязана ли критическая точка быть точкой экстремума?
15. В чем состоит схема исследования функции нескольких переменных на экстремум?
16. Что такое условный экстремум функции нескольких переменных?
17. Какие задачи относят к выпуклому программированию, а какие – к вогнутому?
18. Где отмечаются минимальные и максимальные значения функций в НП-задачах?
19. Как используется матрица Гёссе при определении экстремумов целевой функции?
20. Для чего в задачах многомерной оптимизации применяются методы одномерной минимизации?
21. Какие методы поиска относятся к прямым, а какие – к градиентным?
22. Что такое градиент функции нескольких переменных?
23. Как определить составляющие вектора градиента?
24. Когда будет происходить наискорейший спуск, а когда – полношаговый?
25. Каковы основные свойства градиента функции в некой точке?
26. В чем заключается градиентные методы при поиске минимума функции?
27. Как определяется шаг поиска в градиентных методах?
28. Что такое сопряженные направления?
29. В чём состоят достоинства и недостатки метода Ньютона?
30. В чём предназначение квазиньютоновских методов?
31. Как сконструировать функцию Лагранжа?
32. Что представляют собой множители Лагранжа?
33. В чем заключается метод множителей Лагранжа?

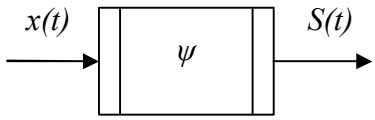
34. В каких случаях необходимые условия оптимальности в задаче НП также являются и достаточными?
35. Какое ограничение НП-задачи называется активным?
36. Как формулируется теорема Куна-Таккера?
37. В чём заключается условие дополняющей нежёсткости?
38. Каковы особенности седловой точки функции многих переменных?
39. Как формулируется теорема о седловой точке?
40. Что вкладывается в понятие “возможное направление”?
41. Для чего строится проецирующая матрица в методе Зойтендейка?
42. Каковы общие идеи, положенные в основу методов штрафных функций?
43. Какие требования предъявляют к штрафным функционалам методов барьерных поверхностей?
44. Какие требования предъявляют к штрафным функционалам методов внешней точки?
45. За счёт чего происходит решение НП-задачи в методах внешней точки?
46. Как формулируется теорема квадратичного программирования?
47. Как по «внешнему» виду математической модели определить: относится ли она к задачам квадратичного программирования?
48. Что означает термин “симметричная” матрица?
49. Что означают термины “отрицательно” и “положительно” определённые матрицы?
50. Когда задача квадратичного программирования неразрешима?
51. Можно ли решить задачу квадратичного программирования для случая с функцией цели вида $f(x) = b^T X + \frac{1}{2} X^T C X \to \max$, когда C — симметричная, положительно определённая матрица, и, если Вы полагаете, что можно, поясните, каким образом?
52. Какие шаги предпринять, чтобы пользуясь изложенным выше методом решить НП-задачу с целевой функцией вида $f(x) = b^T X + \frac{1}{2} X^T C X$ на минимум?

3. ЭЛЕМЕНТЫ ТЕОРИИ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ

3.1 Марковские цепи и потоки событий [4]

По определению, функция времени *x(t)*, которая принимает при каждом фиксированном значении аргумента случайное значение, называется ***случайным процессом***.

Пусть такой процесс воздействует на вход системы.



Наблюдаемое состояние системы при этом есть *S(t)*.

Определение. Если для любого момента времени t_0 вероятностные характеристики процесса *s(t)* в ***будущем*** зависят только от его состояния s_0 в данный момент и не зависят от ***предыстории процесса***, то процесс называется ***марковским***.

Примеры. 1) Поезд, идущий по маршруту, в график не укладывается. В какой-то момент опаздывает, а гдето должен наверстывать. 2) Интенсивность обстрела цели определяется боезапасом и количеством стволов и не зависит напрямую от предыдущих стрельб.

Существует парадокс, согласно которому любой процесс может стать марковским, если все параметры из “прошлого” от которых зависит “будущее”, включить в настоящее. Вот часы: будут ли идти через месяц - два? Если их состояние “идут”, то процесс не является марковским. Если учитывать время завода или вставки батарейки, то он становится таковым.

В технике и при исследовании операций используется класс марковских процессов с ***дискретными состояниями*** и ***непрерывным временем***.

Основные черты таких процессов:

- состояния (значения) образуют конечное перечислимое множество;
- переход из состояния в состояние осуществляется скачком (мгновенно);
- одновременно процесс может находиться только в одном из состояний.

Математической моделью марковского процесса является ***матрица переходных вероятностей***[65]

$$P(t) = \begin{pmatrix} p_{11}(t) & p_{12}(t) & \dots & p_{1k}(t) \\ p_{21}(t) & p_{22}(t) & \dots & p_{2k}(t) \\ \dots & \dots & p_{ij}(t) & \dots \\ p_{m1}(t) & p_{m2}(t) & \dots & p_{mk}(t) \end{pmatrix},$$

в которой элемент $P_{i,j}$ представляет вероятность перехода из i-ого состояния в j-ое.

Очевидно, что сумма по строке матрицы $P(t)$ при фиксированном значении t, равна единице, поскольку состояния процесса образуют полный ансамбль.

Процессы такого рода удобно представлять в виде графа, где вершины соответствуют состояниям процесса, а дуги изображают возможные переходы между состояниями.

Ситуация, когда нет перехода из i-ого состояния в j-ое, в матрице переходных вероятностей соответствующая координата будет нулевой.

Частный случай марковского процесса – процесс с ***дискретными состояниями*** и с ***дискретным временем***, называемый ***цепью Маркова***.

Условимся считать переход из одного состояния в другое ***событием***. Пусть события происходят в заранее неизвестные моменты времени. Процесс, связанный с наступлением событий будем называть ***потоком событий*** или просто – ***потоком***.

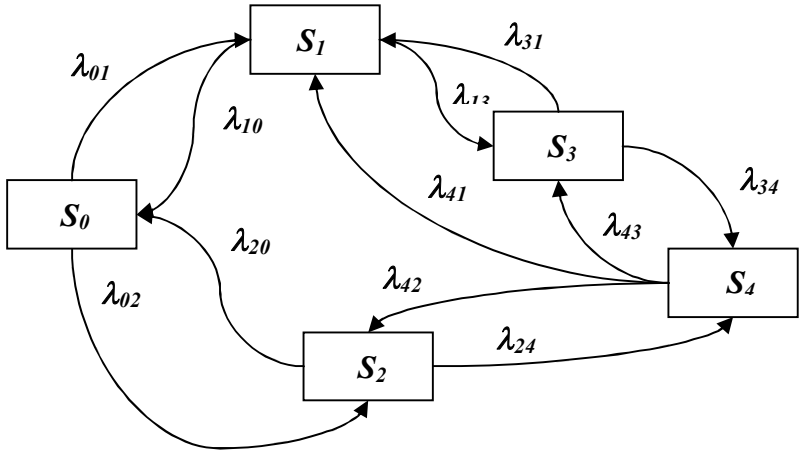
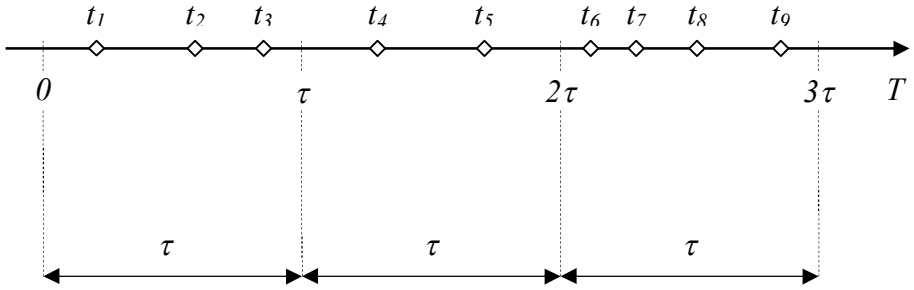


Рисунок 3.1 – Граф переходов между состояниями марковского процесса

Если события, происходящие в потоке, имеют одинаковую природу, поток называется ***однородным***. Для однородного потока матрица переходных вероятностей ***не зависит*** от времени

Однородные потоки событий являются одним из предметов изучения дисциплин исследования операций вообще и систем массового обслуживания, в частности.



Для оценки характеристик потоков оперируют понятиями теории вероятностей:

- среднее число событий, произошедших за единицу времени (интенсивность);
- среднее время между событиями.

3.2. Простейший поток событий [21, 27]

В качестве ***простейшего потока*** в теории систем массового обслуживания (СМО) выбран поток, обладающий следующими свойствами:

- стационарен, по крайней мере, в широком смысле;
- ординарен (ни какие два события в потоке не происходят одновременно, всегда существует такой минимальный квант времени τ, в течение которого происходит только одно событие);
- не имеет последействия (события потока не связаны между собой).

Причинами выбора послужило:

- к простейшему потоку СМО приспособится наиболее трудно, поэтому система, рассчитанная на обработку простейшего потока, будут работать надежно при обработке других потоков, если ***их интенсивности одинаковы***;
- если на вход системы поступает одновременно несколько потоков разной структуры, то механическое суммирование этих потоков даёт поток, близкий к простейшему;
- относительная простота и возможность получения решения в аналитичекой форме для большинства практических приложений.

Можно считать, что для СМО простейший поток играет роль аналогичную нормальному закону распределения в теории вероятностей.

3.3. Математические модели потоков [4, 21, 53, 60, 61]

В литературе (например, в [13]) простейший поток именуется пуассоновским, поскольку описывается моделью Пуассона:

$$P_k(\tau) = \frac{(\lambda\tau)^k}{k!} \cdot \mathrm{e}^{-\lambda\tau},$$

которая характеризует вероятность наступления k событий за отрезок времени, равный τ, λ – интенсивность потока, равная математическому ожиданию числа событий, происходящих в единицу времени.

Вероятность отсутствия заявок за время определится выражением

$$P_k(\tau) = \mathrm{e}^{-\lambda\tau},$$

а отсутствие оных есть

$$P_k(\tau) = 1 - \mathrm{e}^{-\lambda\tau}.$$

Отсюда плотность распределения вероятностей наступления событий

$$f(t) = \lambda\, \mathrm{e}^{-\lambda t}, \qquad t > 0, \tag{3.1}$$

которое называется показательным или экспоненциальным распределением.

Для этого распределения характерно равенство

$$m_\tau = \sigma_\tau = \frac{1}{\lambda} -$$

– средний интервал времени между соседними заявками, совпадает со среднеквадратическим его отклонением.

В этом случае известный в статистике коэффициент вариации случайной величины

$$V_\tau = \frac{\sigma_\tau}{m_\tau} = 1.$$

Рассмотрим некую периодическую или равномерно дискретизированную последовательность. В этом случае $m_\tau = T_0$ – шагу дискретизации или периоду, а $\sigma_\tau = 0$, откуда имеем нулевой коэффициент вариации.

Отсюда становится ясным, почему к обработке простейшего потока труднее всего приспособится – он обладает максимальным коэффициентом вариации.

Если простейший поток прореживается, то есть сохраняется каждое *k*-ое событие, а остальные не учитываются (отбрасываются, отбраковываются), то возникает ***поток Эрланга*** *k*-ого порядка, *k =1, 2,…*

Плотность распределения вероятности такого потока описывается формулой:

$$f(t) = \frac{\lambda(\lambda \cdot t)^{k-1}}{(k-1)!} \mathrm{e}^{-\lambda t}, \qquad t > 0.$$

Отметим, что простейший поток можно считать эрланговским при значении $k = 1$.

Для потока Эрланга характерно

- $m_k = \frac{k}{\lambda}$ – математическое ожидание, а
- $\sigma_k^2 = \frac{k}{\lambda^2}$ – дисперсия времени между наступлениями событий.

Иногда указанные формулы представляют в виде

$$m_k = \frac{1}{\lambda_k} \text{ и } m_k = \frac{1}{k\lambda_k^2}, \text{ где } \lambda_k = \frac{\lambda}{k},$$

но независимо от формы представления, коэффициент вариации определяется как

$$V_k = \frac{1}{\sqrt{k}}.$$

Поток Эрланга обладает следующими свойствами:

- стационарен;
- ординарен;
- обладает последействием;
- при возрастании *k* становится почти периодическим.

Когда модель потока априори не известна, её строят, по мере обретения информации о потоке в виде интегральной модели

$$f_k(\tilde{\tau}) = \tilde{\lambda} \cdot \int_0^{\tilde{\tau}} [\varphi_{k-1}(\tau) - \varphi_k(\tau)] d\tau \text{ , где}$$

$\varphi_0(\tau)$- функцией Пальма.

$\varphi_k(\tau)$ - функция Пальма-Хинчина, k=1,2,…

Потоки, описываемые таким образом, называются потоками Пальма или ***рекуррентными*** в силу ядра интеграла. Интегральная модель достаточно достоверно отображает потоки, встречающиеся в практической деятельности [27].

Полагая функцию распределения вероятностей равной

$$f(t) = 1 - \varphi_0(t),$$

$\varphi_0(t)$ – определяемой формулой (3.1), а $\tilde{\lambda} = \lambda$, можно перейти к простейшему потоку.

3.4. Модель Колмогорова для описания систем с вероятностными состояниями [13, 27, 53]

Выше было отмечено, что процессы, протекающие в системах, удобно представлять в виде графа состояний, как это показано на рисунке 3.1. На рисунке обозначено: S_i – состояния системы, λ_{ij} – потоки, переводящие процесс из i-го состояния, в j-ое состояние.

Колмогоров предложил описывать такие системы и процессы в виде системы дифференциальных уравнений, названных в его честь уравнениями Колмогорова.

Уравнения к конструируют по следующим правилам:

- число уравнений определяется количеством состояний системы (процесса);
- левая часть – производная по времени вероятности нахождение системы в i-ом состоянии;
- правая часть – сумма произведений вероятностей нахождения системы в j-ых состояниях, на интенсивности потоков, переводящих систему в i-е состояние из j-ых (так называемая, взвешенная сумма), за вычетом вероятности i-го состояния, умноженной на сумму интенсивностей потоков, выводящих систему из i-го состояния.

$$\frac{dP_i(\tau)}{d\tau} = \sum_{J+} \lambda_{ij} P_j(\tau) - P_i(\tau) \sum_{J-} \lambda_{ij} .$$

Для системы, изображённой на рисунке 3.1 можно построить следующую систему дифференциальных уравнений Колмогорова:

$$\left.\begin{aligned}
\frac{dP_0(\tau)}{d\tau} &= \lambda_{10}P_1(\tau) + \lambda_{20}P_2(\tau) - P_0(\tau)(\lambda_{01} + \lambda_{02}),\\
\frac{dP_1(\tau)}{d\tau} &= \lambda_{01}P_0(\tau) + \lambda_{31}P_3(\tau) + \lambda_{41}P_4(\tau) - P_1(\tau)(\lambda_{10} + \lambda_{13}),\\
\frac{dP_2(\tau)}{d\tau} &= \lambda_{02}P_0(\tau) + \lambda_{42}P_4(\tau) - P_2(\tau)(\lambda_{20} + \lambda_{24}),\\
\frac{dP_3(\tau)}{d\tau} &= \lambda_{13}P_1(\tau) + \lambda_{43}P_4(\tau) - P_3(\tau)(\lambda_{31} + \lambda_{43}),\\
\frac{dP_4(\tau)}{d\tau} &= \lambda_{24}P_2(\tau) + \lambda_{34}P_3(\tau) - P_4(\tau)(\lambda_{41} + \lambda_{42} + \lambda_{43}).
\end{aligned}\right\}$$

3.5. Схема “гибели – размножения” и её модель [13, 53]

Пусть на вход системы воздействует поток событий, которые переводят систему в ряд ***последовательных*** состояний. В свою очередь, система тем или иным способом реагирует на события и обладает свойством ***регенерируемости***, то есть обладает способностью “противиться” потоку и последовательно возвращаться к исходным состояниям.

Граф состояния такой системы, как это представлено на рисунке 3.2, будет вытянут, по аналогии с объектами биологии (где он впервые был использован), его называют схемой “гибели – размножения” (или “размножения – гибели”).

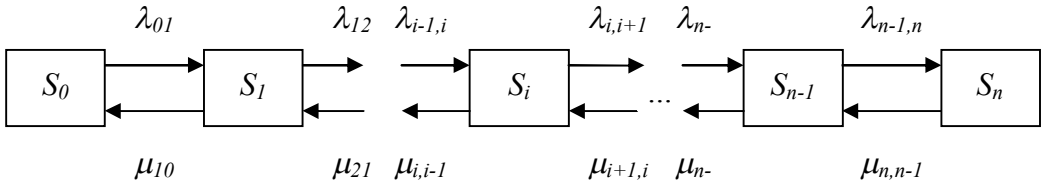


Рисунок 3.2 – Граф “гибели – размножения”

Для системы, описываемой с помощью такого графа можно составить уравнения Колмогорова в установившемся режиме[67]. При этом предполагается, что:

существуют “финальные” вероятности состояний $\lim\limits_{t\to\infty} P_i(t) = P_i$;

состояния совокупно с их вероятностями образуют полный ансамбль $\sum\limits_{i=0}^{n} P_i = 1$.

Очевидно, что дифференциальные уравнения Колмогорова, при этом, преобразуются в систему линейных уравнений, благодаря чему представляется возможным найти финальные вероятности нахождения системы в соответствующих состояниях. Составим указную систему и отыщем финальные вероятности.

Для понимания дальнейшего изложения рассмотрим первые два уравнения системы и сравним их между собой.

$$\left.\begin{aligned} 0 &= \mu_{10}P_1 - \lambda_{01}P_0 \\ 0 &= \lambda_{01}P_0 + \mu_{21}P_2 - \mu_{10}P_1 - \lambda_{12}P_1 \end{aligned}\right\} \Rightarrow 0 = \mu_{21}P_2 - \lambda_{12}P_1.$$

Сама же система имеет вид

$$\begin{cases} \mu_{10}P_1 = \lambda_{01}P_0, \\ \mu_{21}P_2 = \lambda_{12}P_1, \\ \ldots \\ \mu_{i,i-1}P_i = \lambda_{i-1,i}P_{i-1}, \\ \ldots \\ \mu_{n-1,n}P_n = \lambda_{n-1,n}P_{i-1}, \\ P_0 + P_1 + \ldots + P_i + \ldots + P_n = 1. \end{cases}$$

Эта система путем последовательных подстановок легко решается:

$$+\begin{cases} P_1 = \dfrac{\lambda_{01}}{\mu_{10}}P_0 \\ P_2 = \dfrac{\lambda_{12}}{\mu_{21}}P_1 = \dfrac{\lambda_{01}\cdot\lambda_{12}}{\mu_{10}\cdot\mu_{21}}P_0 \\ \ldots \\ P_i = \dfrac{\lambda_{01}\cdot\lambda_{12}\cdot\ldots\cdot\lambda_{i-1,i}}{\mu_{10}\cdot\mu_{21}\cdot\ldots\cdot\mu_{i,i-1}}P_0 \\ \ldots \end{cases}$$

$$\overline{P_0 + \sum_{i=1}^{n} P_i = 1} \qquad (3.2)$$

Отсюда находится вероятность нахождения системы в начальном состоянии S_0:

$$P_0 = \left[1 + \frac{\lambda_{01}}{\mu_{10}} + \frac{\lambda_{01} \cdot \lambda_{12}}{\mu_{10} \cdot \mu_{21}} + \ldots + \frac{\lambda_{01} \cdot \lambda_{12} \cdot \ldots \cdot \lambda_{i-1,i}}{\mu_{10} \cdot \mu_{21} \cdot \ldots \cdot \mu_{i,i-1}} + \ldots + \frac{\lambda_{01} \cdot \lambda_{12} \cdot \ldots \cdot \lambda_{n-1,n}}{\mu_{10} \cdot \mu_{21} \cdot \ldots \cdot \mu_{n,n-1}} \right]^{-1}, \quad (3.3)$$

а через нее выражаются все остальные вероятности, согласно (3.2).

3.6. Понятие СМО. Формулы Литтла [13, 21, 53]

Система массового обслуживания (СМО) – специфическая техническая система, предназначенная для обработки потока.

В её состав входят обрабатывающие (обслуживающие) единицы, которые называются ***каналами обслуживания***. Физически это линии телефонной связи, рабочие точки, душевые кабинки, выездные бригады скорой медицинской помощи, продавцы и тому подобное.

Подразумевается, что поток представлен ***требованиями (заявками) на обслуживание***, при этом, для краткости употребляют термин ***поток заявок***.

Считается, что процесс работы СМО является процессом с дискретными состояниями и непрерывным временем, марковский. Заявки однородны по своей природе, система описывается схемой “разложение-гибель”.

- Пусть процессы, протекающие в СМО, связаны со следующими потоками событий: поток заявок, поступающих на вход СМО;
- поток заявок, покидающих СМО.

При наступлении (установке) в системе стационарного режима, эти потоки стационарны, сколько заявок входит, в среднем, в систему, столько и выходит. Система переходит из состояния в состояние скачкообразно (смотри рисунок 3.3).

Пусть

- $x(t)$ – число заявок, поступивших в СМО;
- $y(t)$ – число заявок, оставивших СМО;
- $S(t)$ – текущее состояние СМО.

Состояние системы определяется числом заявок, в ней находящихся (смотри граф “размножение-гибель” на рисунке 3.1), как обслуживаемых в настоящий момент, так и находящихся в очереди запросов:

$$S(t) = x(t) - y(t).$$

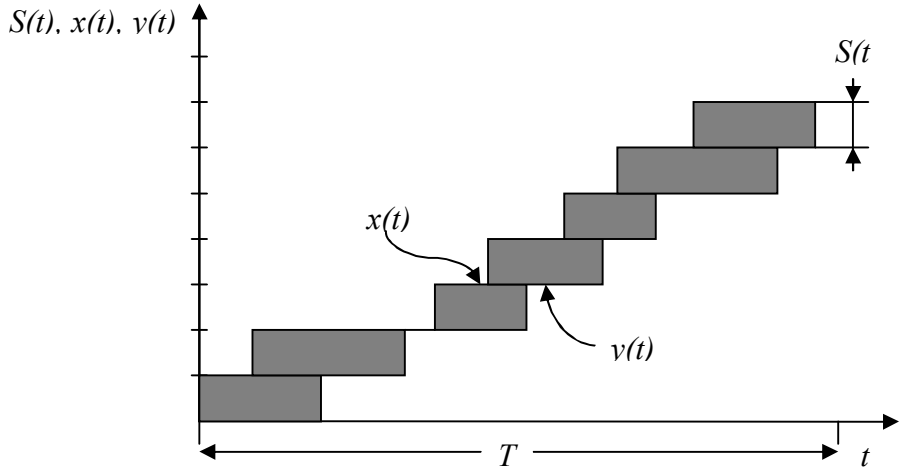


Рисунок 3.3 – Текущее состояние СМО в стационарном режиме

Среднее число заявок, находящихся в системе, есть

$$N_C = \lim_{T\to\infty} \frac{1}{T} \int_0^T S(t)dt\,,$$

где T – интервал наблюдения.

В силу дискретности $S(t)$, ординарности потоков (никакие два события несовместимы в пределах кванта времени) и установившегося режима (высота "ступеньки" единичная), как это показано на рисунке 3.3, интеграл можно заменить суммой

$$N_C = \lim_{T\to\infty} \frac{1}{T} \sum_i t_i\,,$$

где t_i – время наступления i-го события.

При этом, среднее число заявок, поступивших на вход системы за время наблюдения T, равно $m_з = \lambda T$. Учитывая это, избавимся от бесконечного предела, умножая и деля на λ.

$$N_C = \lim_{T\to\infty} \frac{1}{T} \cdot \frac{\lambda}{\lambda} \sum_i t_i = \lambda \cdot \frac{\sum\limits_i t_i}{m_3}\,.$$

Обозначим среднее время пребывания заявки в системе как

$$T_C = \frac{\sum\limits_i t_i}{m_з},$$

придём к выражению, называемому формулой Литтла [74]:

$$T_C = \frac{1}{\lambda} N_C . \qquad (3.4)$$

Справедливо утверждение, озвучивающее эту формулу.

При любом распределении времени обслуживания и любой дисциплине обслуживания, среднее время пребывания заявки в системе обратно пропорционально интенсивности входного потока и прямо пропорционально числу заявок в системе.

Подобные же рассуждения могут быть проделаны для получения связи среднего времени пребывания заявки в очереди и длины очереди.

$$T_{Оч} = \frac{1}{\lambda} N_{Оч} . \qquad (3.5)$$

При любом распределении времени обслуживания и любой дисциплине обслуживания, среднее время пребывания заявки в очереди обратно пропорционально интенсивности входного потока и прямо пропорционально числу заявок в очереди.

3.7. Примеры СМО. Одноканальная СМО с отказами [13, 29]

К исходным данным для расчёта СМО данного типа следует отнести следующие параметры и допущения:

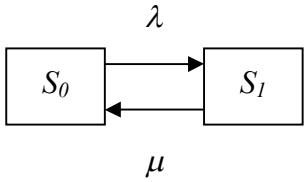
- система имеет один канал обслуживания;
- система имеет два состояния: «свободно» и «занято»;
- поток заявок, поступающих на вход системы, простейший, его интенсивность λ заявок в единицу времени задана;
- время обслуживания одной заявки определяется показательным законом $f(t) = \mu \mathrm{e}^{-\mu t}, t > 0$, где μ – интенсивность обслуживания, заявок в единицу времени;
- заявка пришедшая, когда СМО свободна, принимается на обслуживание;

- заявка пришедшая, когда СМО занята, отвергается и аннулируется (пропадает).

Требуется определить:

- вероятности отказа и обслуживания;
- абсолютную и относительную пропускные способности.

Граф «размножения – гибели», в этом случае, простой



Его расчёт даёт значения финальных вероятностей

$$\begin{cases} P_0\lambda = P_1\mu, \\ P_0 + P_1 = 1 \end{cases} \Rightarrow \begin{vmatrix} P_1 = \rho P_0, \\ P_0 = (1+\rho)^{-1}, \end{vmatrix}$$

где ρ – так называемая, ***приведённая интенсивность***. Последняя может трактоваться как

$$\rho = \frac{\lambda}{\mu} = \lambda \times \left(\frac{1}{T_{Обсуживания}} \right)^{-1} = \lambda \cdot T_{Обсуживания}$$

– среднее число заявок, ***поступающих на вход*** системы ***за время*** обслуживания ***одной*** заявки.

Окончательно имеем:

- вероятность отказа $P_1 = \dfrac{\lambda}{\mu + \lambda}$,
- вероятность обслуживания $P_0 = \dfrac{\mu}{\mu + \lambda}$.

Последняя формула может трактоваться как предельное значение частоты

$$\lim_{N_{Поступающих} \to \infty} \frac{N_{Обслуженных}}{N_{Поступающих}} = P_0 .$$

Таким образом, относительная пропускная способность (для данного типа СМО) совпадает с вероятностью нахождения системы в свободном состоянии и с вероятностью обслуживания.

Абсолютная же пропускная способность может быть определена двояко.

С одной стороны, уравнение $P_0\lambda = P_1\mu = \dfrac{\lambda\mu}{\lambda+\mu}$ описывает “баланс” обслуженных и поступающих заявок, а с другой стороны, можно воспользоваться понятийным аппаратом.

Абсолютная пропускная способность есть доля от числа заявок, поступающих на вход СМО в единицу времени, которые удаётся обслужить

$$A = q = \lambda P_0 = \frac{\lambda\mu}{\lambda+\mu}.$$

3.8. Примеры СМО. Многоканальная СМО с отказами [13, 29, 33]

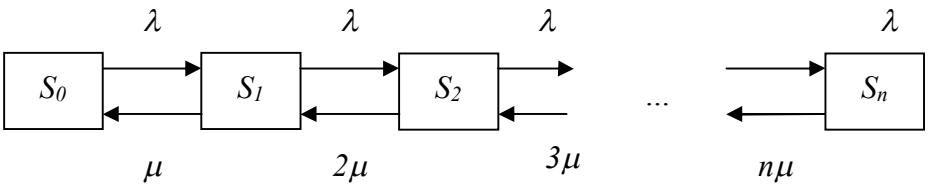
Расчёт многоканальной СМО с отказами получил наименования задача Эрланга. Исходными данными для проведения расчётов служат следующие:

- система имеет *n* однотипных каналов обслуживания:
- на вход поступает поток с интенсивностью λ заявок в единицу времени;
- интенсивность обслуживания одного канала составляет μ заявок в единицу времени;
- заявки принимаются на обслуживание, пока хотя бы один канал свободен;
- заявка, поступившая в момент времени, когда все каналы заняты, отвергается (пропускается, в терминах ПВО, когда эта задача впервые решалась).

Требуется рассчитать:

- вероятности обслуживания и отказа заявок;
- среднее число заявок, обслуживаемых в единицу времени (абсолютная пропускная способность);
- долю обслуживаемых заявок (относительная пропускная способность);
- среднее число занятых каналов.

Граф системы имеет вид (обратите внимание на интенсивности “возврата” в предыдущие состояния):



Вычисление финальных вероятностей для этого случая даёт следующее. Вероятность того, что система полностью свободна от заявок составляет

$$P_0 = \frac{1}{1 + \rho + \frac{\rho^2}{2} + ... + \frac{\rho^n}{n!}},$$

где ρ – приведённая интенсивность.

По условию, пришедшая заявка будет отвергнута, когда все каналы заняты и система находится в состоянии S_n:

$$P_{ОТК} = P_0 \frac{\rho^n}{n!}.$$

Очевидно, что в противном случае заявка будет обслужена, и вероятность такого события есть:

$$P_{ОБСЛ} = 1 - P_{ОТК}.$$

В данном случае, вероятность обслуживания, по понятным соображениям, совпадает по значению с относительной пропускной способностью. Отсюда находится абсолютная пропускная способность, как доля обслуженных заявок от заявок, поступающих в единицу времени:

$$A = q = \lambda P_{ОБСЛ}.$$

Среднее число занятых каналов

$$m_K = \frac{A}{\mu} = \rho P_{ОБСЛ}$$

есть отношение числа заявок, забираемых в единицу времени на обслуживание, к скорости обслуживания.

3.9 Примеры СМО. Одноканальная СМО с неограниченной очередью

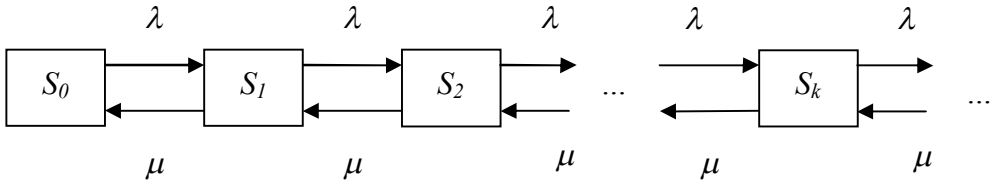
В модели СМО данного типа присутствуют следующие параметры и гипотезы [13, 29, 60, 61]:

- система имеет один канал обслуживания;
- поток заявок, поступающих на вход системы, простейший, его интенсивность λ заявок в единицу времени задана;
- время обслуживания одной заявки подчиняется показательному закону с интенсивностью обслуживания μ заявок в единицу времени;
- заявка пришедшая, когда СМО свободна, принимается на обслуживание;
- заявка пришедшая, когда СМО занята, помещается в очередь с дисциплиной обслуживания FIFO (первый пришедший обслуживается первым), длина которой теоретически ничем не ограничена.

Требуется определить:

- среднее число заявок в системе N_c;
- среднее время пребывания заявки в системе T_c;
- среднее число заявок в очереди $N_{оч}$.

Граф «размножения – гибели», в этом случае, имеет бесконечное число состояний и не ограничен



Используя ранее полученный результат решения модели Колмогорова (3.3), запишем

$$P_0 = \frac{1}{1 + \rho + \rho^2 + ... + \rho^k + ...}, \qquad (3.5)$$

где ρ – приведённая интенсивность.

В знаменателе дроби – геометрическая прогрессия с начальным членом 1 и знаменателем ρ. Сумма членов этой прогрессии будет существовать при $\rho < 1$ и равна $\frac{1}{1-\rho}$. В случае, когда $\rho \geq 1$, вероятность обслуживания (3.5) будет стремиться к нулю. Физически это означает, что очередь возрастает быстрее, нежели заявки попадают на обслуживание, и,

в перспективе, у «поздних» заявок неограниченно возрастает время ожидания в очереди и пребывания в системе.

Поэтому говорят, что если $\rho \geq 1$, то система с такими параметрами λ и μ нежизнеспособна. Совершенно очевидно, что для успешного функционирования одноканальной СМО с бесконечной очередью, необходимо обеспечить отношение $\lambda > \mu$.

Из (3.5), когда $\rho < 1$, следует, что вероятность отсутствия заявок в системе равна

$$P_0 = 1 - \rho .$$

Вероятность же наличия k заявок в системе есть

$$P_k = (1 - \rho) \cdot \rho^k .$$

Воспользуемся для нахождения среднего числа заявок в системе N_c известной формулой среднего дискретной случайной величины $m_x = \sum_i x_i P_i$. Имеем

$$N_C = \sum_{k=0}^{\infty} k(1 - \rho)\rho^k ,$$

применим элементарное алгебраическое преобразование:

$$N_C = (1 - \rho)\rho \sum_{k=1}^{\infty} k\rho^{k-1} .$$

Под знаком суммы в последнем выражении стоит производная, поэтому

$$N_C = (1 - \rho)\rho \sum_{k=1}^{\infty} \frac{d}{d\rho}\rho^k .$$

Так как операции дифференцирования и суммирования переставимы, а при перестановке под знаком суммы окажется геометрическая прогрессия, имеем ряд преобразований

$$N_C = (1 - \rho)\rho \frac{d}{d\rho} \sum_{k=1}^{\infty} \rho^k = (1 - \rho)\rho \frac{d}{d\rho}\left[\frac{\rho}{(1 - \rho)}\right].$$

Окончательно среднее число заявок в СМО равно

$$N_c = \frac{\rho}{1-\rho}.$$

Воспользовавшись формулой Литтла (3.4), получим среднее время пребывания запроса в системе (время отклика):

$$T_c = \frac{N_c}{\lambda} = \frac{\rho}{\lambda(1-\rho)}.$$

Определим среднее число заявок, находящихся непосредственно на обслуживании

$$m = 0 \cdot P_0 + 1 \cdot (1 - P_0) = \rho.$$

Поэтому в очереди находится (средняя длина очереди), в среднем

$$N_{оч} = N_C - m = \frac{\rho^2}{1-\rho}.$$

заявок. Воспользовавшись формулой Литтла (3.5) получим среднее время ожидания в очереди:

$$T_{оч} = \frac{N_{оч}}{\lambda} = \frac{\rho^2}{\lambda(1-\rho)}.$$

3.10. Пример решения практической задачи

Задача. Некая фирма работает 24 часа в сутки. В среднем, у 48 служащих возникает желание, чтобы размяться, по одному разу покрутить педали велотренажёра. Среднее время занятий на велотренажёре составляет 20 минут.

Служащие жалуются, что время ожидания у велотренажёра велико и просят у администрации установки ещё дополнительных велотренажёров.

Со своей стороны, дирекция считает, тренажёр занят только 2/3 всего времени, а установка дополнительного тренажёра – пустая трата денег.

Требуется вынести квалифицированное заключение по этому вопросу.

Воспользуемся для расчётов моделью СМО с бесконечной очередью.

Из условия задачи следует, что интенсивность обслуживания составит

$$1/\mu = 20 \text{ [мин]} \Rightarrow \mu = 3 \text{ [сотрудника/час]}.$$

Считая поток на входе тренажёрной комнаты пуассоновским, найдём его интенсивность:

$$\lambda = 48/24 = 2 \text{ [сотрудника/час]}.$$

Приведённая интенсивность, при этом, равна

$$\rho = 2/3.$$

Среднее время ожидания в очереди, составит:

$$T_{оч} = \frac{\rho^2}{\lambda(1-\rho)} = \frac{\left(\frac{2}{3}\right)^2}{2\times\left(1-\frac{2}{3}\right)} = \frac{2}{3}\text{[час]},$$

то есть сорок минут, а нахождение в спортивном зале, будет равно

$$T_c = \frac{\rho}{\lambda(1-\rho)} = \frac{\frac{2}{3}}{2\times\left(1-\frac{2}{3}\right)} = 1\text{[час]}.$$

Всего же в системе будут, в среднем находится

$$N_c = \frac{\rho}{1-\rho} = \frac{\frac{2}{3}}{\frac{1}{3}} = 2$$

сотрудника фирмы, один из которых оздоровляется, а другой – бездельничает, ожидая своей очереди к велотренажёру.

Максимальное время в очереди для 90% времени составит

$$\pi(90\%) = T_{оч} \times \ln [10\rho] = 113{,}8 \text{ [мин]},$$

то есть, установка дополнительных велотренажёров необходима.

3.11. Сводные показатели эффективности СМО [7, 29, 33]

1. Вероятность потери требования (отказа) СМО.

2. Вероятность P_k занятости k приборов из n, для многоканальной СМО частными случаями являются:

- P_0 – все приборы свободны;
- $P_n = P_{отк}$ – все приборы заняты, если СМО без очереди.

3. Среднее число занятых приборов

$$N_{Зан} = \sum_{k=1}^{n} kP_k$$

характеризует степень занятости.

4. Среднее число свободных приборов

$$N_{\text{Св}} = \sum_{k=1}^{n} (n-k)P_k \, .$$

5. Коэффициент простоя приборов

$$K_{Пр} = \frac{N_{\text{Св}}}{n} \, .$$

6. Аналогично определяется коэффициент загрузки приборов или коэффициент занятости

$$K_{Зан} = \frac{N_{\text{Зан}}}{n} \, .$$

Если используется система с ожиданием в очереди, то дополнительно определяются:

7. Среднее время пребывания в очереди (ожидания)

$$T_{оч} = T_{ож} = \int_0^{\infty} t dP(t_{ож} > t) \text{ где } P(t_{ож} > t) = \sum_{k=0}^{n} \rho_k P_k(t_{ож} > t)$$

где $P_k(t_{ож} > t)$ – условная вероятность того, что время ожидания $t_{ож} > t$ при условии, что в момент поступления требования в систему в ней уже находилось k заявок.

8. Вероятность того, что время пребывания в очереди не превысит определённого, есть

$$P(t_{ож} < t) = \sum_{k=n}^{\infty} \rho_k P_k(t_{ож} < t).$$

9. Средняя длина очереди, большей заданной длины n

$$N_{оч} = \sum_{k=n}^{\infty} (k-n) P_k .$$

10. Среднее число заявок в системе

$$N_C = N_{оч} + N_З = \sum_{k=1}^{\infty} k P_k .$$

11. Вероятность того, что число требований в очереди больше некоторого числа n

$$P_{k>n} = \sum_{k=n+1}^{\infty} P_k .$$

Существуют вполне очевидные связи показателей (характеристик)

- $n = N_З + N_{Св}$ по каналам обслуживания;
- $T_C = T_{обсл} + T_{оч}$ по времени.

В экономических расчётах применяется следующие показатели оценки СМО [11, 44, 76, 77]

- $q_{обсл}$ – стоимость обслуживания отдельного требования в СМО;
- $q_{ож}$ – стоимость потерь от ожидания в очереди;
- $q_{отк}$ – стоимость потерь от отказа заявке в обслуживании;
- $q_к$ – стоимость эксплуатации отдельного прибора в единицу времени;
- $q_{пк}$ – стоимость единицы времени простоя.

На базе этих показателей синтезируются следующие целевые функции в оптимизационных задачах:

- для систем с ожиданием $G = (q_{ож} \times T_{оч} + q_{пк} \times N_{Св} + q_к \times N_З) \times T;$
- для систем с отказами $G = (q_к \times N_З + q_{отк} \times P_{отк} \times \lambda) \times T;$
- для смешанных систем

$$G = (q_{пк} \times N_{Св} + q_{ож} \times T_{оч} + q_к \times N_З + q_{отк} \times P_{отк} \times \lambda) \times T.$$

3.12. Вопросы для самоконтроля

1. Что такое поток?
2. Существуют ли реальные системы, в которых протекающие в них случайные процессы являются марковскими?
3. Что такое стационарность в широком и узком смысле?
4. Когда случайный процесс с непрерывным временем не обладает эргодическим свойством?
5. Обладает ли эргодическим свойством случайный процесс с непрерывным временем, имеющий бесконечное число состояний?
6. Как вы понимаете термин “ординарен”?
7. Как вы объясните, что означает отсутствие последействия?
8. Как связаны интенсивность потока и среднее время между событиями потока?
9. Когда оправдано использование предположения о простейшем характере потока заявок?
10. Каковы единицы измерения интенсивности потока?
11. Почему в моделях потоков положено $t \geq 0$?
12. Поясните, почему при больших значениях k поток Эрланга вырождается в периодическую последовательность?
13. По какой причине для проведения расчётов очень часто используется простейший (пуассоновский) поток?
14. Как влияет изменение параметра k на вид (форму) распределения?
15. Как влияет изменение параметра λ на вид (форму) распределения?
16. В чём заключается физический смысл понятия интенсивности входного потока?
17. Пояснить физический смысл понятия интенсивности обслуживания.
18. Каковы размерность интенсивности потока и величины обратной ей?
19. Какие допущения положены в основание модели одноканальной СМО с отказами?
20. Какие допущения положены в основание модели многоканальной СМО с бесконечной очередью?
21. Какие допущения положены в основание модели многоканальной СМО с ограниченной очередью?

22. Почему в СМО с бесконечной очередью даже если интенсивность обслуживания выше интенсивности поступления заявок, возникают очереди? В каких случаях они не возникают?
23. Как изменятся характеристики СМО при введении ограничений на длину очереди?
24. Как изменятся характеристики СМО при введении ограничений на время пребывания в очереди?
25. Что такое "установившийся режим" в СМО?
26. Какой вид имеют уравнения Колмогорова в "установившемся режиме"?
27. Каков физический смысл понятия "приведенная интенсивность"?
28. Пояснить физический смысл понятия интенсивности обслуживания.
29. Какие допущения положены в основание многоканальной модели СМО с отказами?
30. Как, по Вашему мнению, изменятся характеристики СМО при введении дополнительного числа каналов
31. Приведите примеры *n*-канальных СМО, используемых как в быту, так и в вычислительной технике.
32. Что включает в себя понятие *n*-канальная СМО??
33. Как вид распределения времени обслуживания заявки влияет на зависимости коэффициента загрузки и времени ожидания заявки от интенсивности входного потока заявок?

4. ЭЛЕМЕНТЫ ТЕОРИИ ИГР

4.1 Основные определения

Теория игр – математическая теория конфликтов.

Конфликт – такая ситуация (стечение обстоятельств), в которой сталкиваются интересы сторон, и происходит борьба интересов.

Для возникновения игровой ситуации ещё необходимо желание сторон участвовать в конфликте, не всякий конфликт – игра. К примеру, продавец на рынке стремится поднять цену на товар, а покупатель – сбить её. Однако, если цена заломлена высоко, покупатель разворачивается и идёт дальше, в то же время, продавец может попытаться удержать покупателя, предлагая сниженную цену.… То есть, столкновение интересов сторон – налицо, но, тем не менее ситуация не переросла в игровую.

Правила игры – возможности, предоставляемые каждому игроку совместно с результатами, к которым приводит использование каждой возможности.

Далеко не каждый конфликт на практике протекает по строго определённым заранее правилам, часто один игрок, за неимением информации о действиях противоположной стороны и невозможности оценить игровую ситуацию в целом, не может быть со стопроцентной гарантией уверен в результатах своих действий. Поэтому в теории игр используют термин ***стратегия***.

Стратегия (образ действия) – порядок использования правил игры.

Тако же, в силу вероятностных причин говорят не о “результате” игры, а о “среднем результате”.

Предполагается, что ***средний результат игры***, будь то выигрыш или проигрыш, ***выражается числом***.

Основная задача теории игр формулируется так: “Как должен себя вести (какую избрать стратегию) разумный игрок в конфликте с разумным противником, чтобы обеспечить себе в ***среднем*** наибольший выигрыш (или наименьший проигрыш)?”.

Разумность игроков – категория философская, но ниже, на историческом примере, мы дадим её интерпретацию с позиции теории игр.

Пусть в каждой игре принимают участие два лица (или персоны) – Ян и ТатьЯна, и пусть каждая игра – конфликт между этими игроками. Такая игра называется ***парной***.

Игра называется ***игрой с нулевой суммой***, если одна из сторон выигрывает то, что проигрывает другая.

Пусть мы принимаем участие на стороне Яна (он же Красный или 1-й игрок). Ян заинтересован в том, чтобы сделать этот выигрыш наибольшим, а Татьяна (Синий или 2-й игрок) – сделать его наименьшим.

Если оба противника одинаково разумны, то, по-видимому, можно отыскать некоторое ***равновесное положение***, при котором каждый из игроков получит своё. Этот равновесный средний выигрыш, на который Ян вправе рассчитывать при своём оптимальном поведении, называется ***ценой игры***.

Таким образом, решит игру означает [30, 48, 79, 80]:

- найти пару стратегий для обоих игроков;
- цену игры Красного игрока.

А если один из игроков ведёт себя неразумно? Тем хуже для него, фактический выигрыш будет отклоняться от цены игры в невыгодную для неразумного игрока сторону.

4.2.Формальное описание игры двух персон с нулевой суммой

Как следует из определений предыдущего параграфа, для описания игры необходимо задать перечень возможностей, которыми обладают игроки, совместно с функцией, описывающей значения выигрыша при стечении обстоятельств, вызванными действиями игроков.

Рассмотрим те игры которые обладают конечным набором возможностей или стратегий, предоставляемым игрокам. В этом случае, функция выигрыша может быть задана таблично, в виде матрицы.

Пусть у Яна имеется m стратегий, а у Татьяны n. В этом случае игра носит название $m \times n$ игры. Игровую ситуацию при этом можно представить в виде таблицы. Строки соответствуют стратегиям Яна (красного), а столбцы – стратегиям Татьяны (синего). На пересечении столбцов и строк помещается значение выигрыша, получаемого красным игроком при выборе соответствующей пары стратегий обоими игроками.

Стратегии 1-го игрока	Стратегии 2-го игрока				
		C_1	C_2	…	C_n
	K_1	h_{11}	h_{12}	…	h_{1n}
	K_2	h_{21}	h_{22}	…	h_{2n}
	...				
	K_m	h_{m1}	h_{m2}	…	h_{mn}

Матрица H называется ***платёжной матрицей*** или ***матрицей игры***. О такой форме представления игры говорят, что игра приведена к ***нормальной форме***.

Если игра не является игрой с нулевой суммой, то функции выигрыша у игроков различны и представимы разными матрицами, поэтому эта игра называется ***биматричной***.

Если число стратегий, предоставляемых игрокам бесконечно, то при их нормировке возникают ***игры на квадратах***.

4.3. Седловая точка и оптимальные стратегии

Рассмотрим конфликтную ситуацию, которая сложилась в ходе 2-й мировой войны в результате боевых действий на Тихом океане между США и Японией [19, 64].

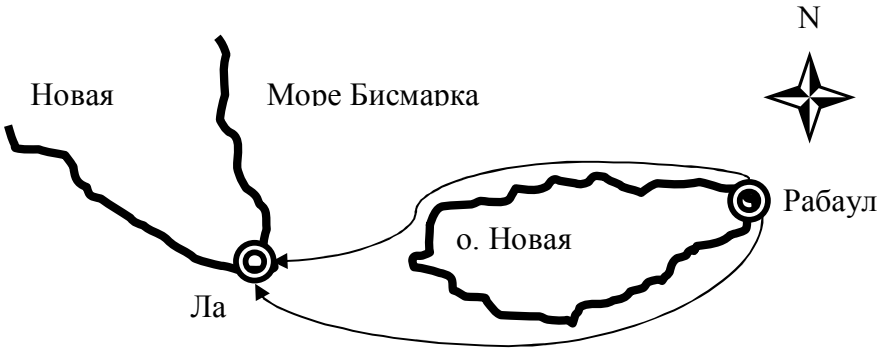


Рисунок 4.1 – Схема района перехода, февраль – март 1943 г., Тихий океан

Японская сторона в ходе эвакуации должна была выполнить переход между портами Рабаул и Лаэ в районе Новой Гвинеи. Остров Новая Британия можно было огибать как с северной стороны так и с южной. Не зависимо от маршрута время нахождения в пути занимало 3 дня. У командования США были аналогичные альтернативные варианты сосредоточения основных сил разведывательной авиации по направлениям: на севере были дождевые шквалы, на юге – относительно хорошая погода.

По замыслу американского командования, после обнаружения вышедшего из Рабаула каравана, и определения курсовых параметров, можно было ставить конкретные задачи на бомбометание. Своеобразие конфигурации мест базирования авиации состояла в том, что возможности нанесения ударов на пути следования конвоев с эвакуируемыми войсками значительно зависели от выбора маршрута движения судов и пунктов сосредоточения самолётов.

Работа штабных аналитиков вылилась в такое описание ситуации в форме матричной игры.

		Japan	
		Север	Юг
US	Север	2	2
	Юг	1	3

В качестве платежей выступает количество дней бомбардировки конвоя в пути. Очевидно, что ситуация складывается для японской стороны неудачно. Как мы видим, ни один из противников в этой игре не мог сделать выбор, который полностью отвечал бы его целям. Каждый из них, выбирая свою стратегию, должен был сознательно учитывать возможный выбор своего противника.

Аналитики США, анализируя таблицу, считают выгодным вариант (Юг, Юг): авиация сосредоточена на юге, и японскому командованию вздумалось огибать Новую Британию с юга. Однако, ***исходя из разумности противника***, в штабе генерала Кеннея решили избрать Северную стратегию, ***которая независимо от действий противника***, в любом случае ***гарантирует*** им два дня бомбардировки.

Японское руководство, разбираясь в ситуации, тоже выбрало Северную стратегию, которая, в ***случае неразумности генералитета*** США, обеспечило им всего один день перемещения под ударами авиации.

В результате – два дня под бомбами.

В данной жестокой игре целями игроков были:

- для США – увеличение (максимизация) дней бомбардировки, а
- для Японии – снижение (минимизация) их.

При этом в игре сложилось положение равновесия, так как в платёжной матрице присутствует седловая точка с координатами (Север, Север).

Определение [64]. Пусть дана игра Г. Ситуация (i_0, j_0) называется равновесной, если для любых i = 1, m и j = 1, n имеет место двойное неравенство

$$H(i, j_0) \le H(i_0, j_0) \le H(i_0, j),$$

где H – платёжная матрица (функция выигрыша) 1-го игрока (красного).

Для рассмотренной исторической ситуации это правило выполняется.

Выявление ситуаций равновесия имеет большое значение с точки зрения поиска благоприятных исходов.

Алгоритм нахождения равновесных ситуаций представлен ниже. Для удобства изложения и восприятия он представлен мнемонической схемой:

$$\left(\begin{array}{cccc} h_{11} & h_{12} & \dots & h_{1n} \\ h_{21} & h_{22} & \dots & h_{2n} \\ \dots & \dots & \dots & \dots \\ h_{m1} & h_{m2} & \dots & h_{mn} \end{array}\right) \left\{\begin{array}{c} \min\limits_j h_{1j} \\ \min\limits_j h_{2j} \\ \\ \min\limits_j h_{mj} \end{array}\right\} \quad v_1 = \max_i \min_j h_{ij}$$

$$\underbrace{\max_i h_{i1} \quad \max_i h_{i2} \quad \dots \quad \max_i h_{in}}$$

$$v_2 = \min_j \max_i h_{ij}$$

При этом неизменно $v_1 \le v_2$, то есть, ***нижний выигрыш*** первого игрока ***не превышает верхнего проигрыша*** второго игрока. Там, где $v_1 = v_2$ – есть ***седловая точка***. Элемент h_{i0j0} ***одновременно минимален*** в строке i_0 и ***максимален*** в столбце j_0.

Говорят, что первый игрок (красный) придерживается ***максиминно***й стратегии (увеличивает свой минимальный выигрыш), а второй игрок (синий) – ***минимаксной*** (минимизирует свой максимальный проигрыш) стратегии.

Когда оба игрока ведут себя разумно, то выигрыш красного игрока ***v*** должен быть ***не меньше***, чем максимин, но и ***не больше***, чем минимакс, то есть

$$v_1 \le v \le v_2.$$

Отклоняющийся от своей оптимальной стратегии будет нести потери, при которых значение выигрыша или проигрыш может выйти за пределы интервала.

В ходе решения игры может быть несколько точек равновесия. В этом случае, не играет роли, координаты какой из полученных равновесных точек использовать в качестве номеров стратегий в игре.

4.4. Понятие о смешанных стратегиях [51, 86, 88]

Рассмотрим ситуацию, известную нам по программе капитал-шоу “Поле чудес”, наступающую после угадывания трёх букв кряду: Л.А. Якубович предлагает на выбор две шкатулочки, в одной из которых лежат деньги, а другая пуста. Считая Якубовича игроком номер 1, представим игру в нормальной форме:

		Игрок	
		Лево	Право
Л.А.	Лево	– 1	0
	Право	0	– 1

Из приведённой таблицы следует, что Якубович в ходе игры ни не имеет шансов на выигрыш, но может остаться при своих деньгах.

Имеем $v_1 = \max\limits_i \min\limits_j h_{ij} = -1$, а $v_2 = \min\limits_j \max\limits_i h_{ij} = 0$. Таким образом, в данной игре ***отсутствует седловая точка***.

Ситуации, в которых $v_1 < v_2$ нередки в реальных играх. Может показаться даже, что игра не антагонистическая и не с нулевой суммой. Однако, в каждой конкретной игре выигрыш неизменно равен проигрышу.

Выход в создавшейся ситуации заключается в том, чтобы ***выбирать*** свои стратегии ***случайным образом***. Для этого необходимо

- задать вероятности использования каждой из стратегий и
- задействовать механизм случайного выбора.

Исходные стратегии, заданные в условии задачи (помещённые в таблицу) называют ***чистыми стратегиями***, а полученные с использованием вероятностного механизма – ***смешанными стратегиями***.

Определение. Смешанная стратегия игрока есть распределение вероятностей на множестве его чистых стратегий.

Смешанные стратегии указывают в виде набора вероятностей, с которыми игрок выбирает свои чистые стратегии, то есть

$$P = \{p_1, p_2, \ldots, p_m\},$$
$$Q = \{q_1, q_2, \ldots, q_n\},$$

где p_i — вероятность выбора Яном i-й чистой стратегии, а q_j — вероятность выбора Татьяной своей j-й чистой стратегии.

Очевидно, что $\sum\limits_{i=1}^{m} p_i = 1$ и $\sum\limits_{j=1}^{n} q_i = 1$. Это имеет место и для игр с седловой точкой: оптимальные стратегии задаются с единичной вероятностью, а остальные – с нулевыми.

4.5. Теорема об активных стратегиях

Предположим, что Якубович, чтобы не оставаться внакладе, слегка модифицировал правила игры.

В одну из коробочек прячется фишка. Отныне, игрок, нашедший фишку в левой шкатулке, получает 2 млн. рублей, а если фишка была спрятана слева, а он искал справа, то платит Л.А. Якубовичу штраф в размере 2 млн. рублей. Аналогично, за найденную правую фишку, выигрыш игрока составит 1 млн. рублей, а за ненайденную последует такой же штраф. Эта ситуация отображается такой платёжной матрицей.

Л.А. \ Игрок	Лево	Право
Лево	– 2	2
Право	1	– 1

Имеем: $v_1 = \max\limits_i \min\limits_j h_{ij} = -1$, а $v_2 = \min\limits_j \max\limits_i h_{ij} = 1$, седловая точка отсутствует. Попробуем рассчитать вероятности использования игроками своих чистых стратегий.

Игроки информацией не обмениваются, поэтому каждое игровое сочетания стратегий противников реализуется с вероятностью $p_i \cdot q_j$.

Поэтому, математическое ожидание выигрыша красного игрока (Якубовича) составит

$$H(P,Q) = \sum_{i=1}^{m} \sum_{j=1}^{n} h_{ij} p_i q_j \,, \tag{4.1}$$

при ожидаемом нижнем выигрыше не менее $v_1 = \max\limits_P \min\limits_Q H(P,Q)$,

и при ожидаемом верхнем проигрыше второго игрока не более, чем $v_2 = \min\limits_P \max\limits_Q H(P,Q)$.

Для игр с использованием оптимальных смешанных стратегий всегда выполняется равенство

$$v = v_1 = v_2 \text{ или}$$
$$\max_P \min_Q H(P,Q) = \min_Q \max_P H(P,Q). \tag{4.2}$$

Выражение (4.2) составляет сущность ***основной*** теоремы теории игр, называемой тако же теоремой о ***минимаксе*** или теоремой о ***максимине***.

Теорема[64]. Всякая матричная игра имеет решение, а каждый игрок всегда имеет оптимальную стратегию.

Найдём оценку (4.1) для данной игры.

$$H(P, Q) = -2 \cdot p_1 \cdot q_1 + 2 \cdot p_1 \cdot q_2 + 1 \cdot p_2 \cdot q_1 - 1 \cdot p_2 \cdot q_2 = 2 \cdot p_1 \cdot (q_2 - q_1) - p_2 \cdot (q_2 - q_1) =$$

$= (q_2 - q_1)\cdot(2p_1 - p_2)$

Воспользовавшись условиями

$$p_1 + p_2 = 1 \text{ и } q_1 + q_2 = 1,$$

придем к выражению

$$H(P, Q) = (3p_1 - 1)(1 - 2q_1).$$

Из полученной формулы видно, что при $q_1 = 0{,}5$ значение выигрыша $H(P, Q) = 0$, не зависимо от величины p_1. То же наблюдается когда $p_1 = \frac{1}{3}$.

Получившийся результат не случаен, он отражает сущность теоремы об ***активных стратегиях***.

Теорема[64]. Если один из игроков придерживается своей оптимальной стратегии, ожидаемый выигрыш ***останется неизменным*** и равным цене игры, ***независимо от характера действий второго игрока*** в пределах его активных стратегий.

Подтверждающие теорему результаты имитационного моделирования [39] приводятся на рисунке 4.2.

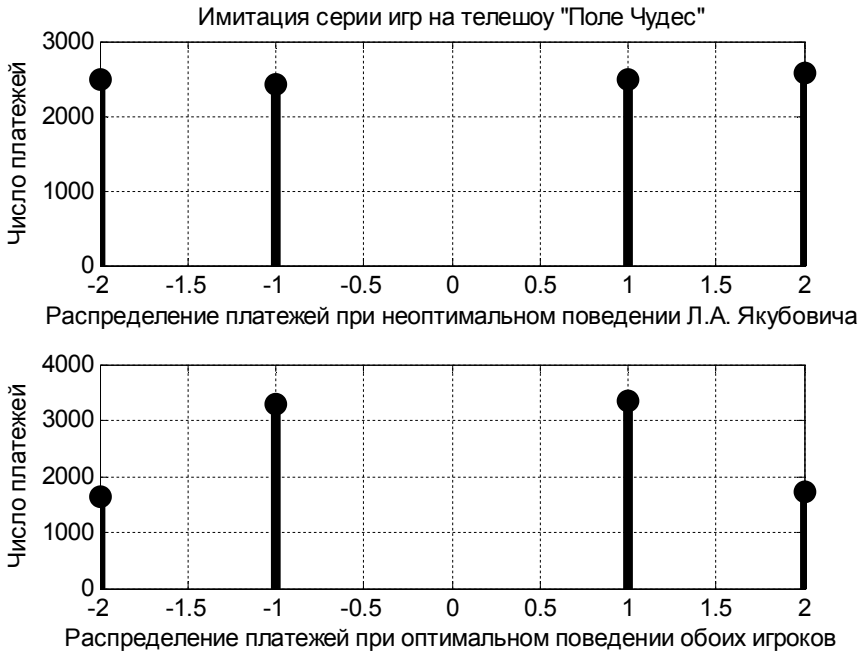


Рисунок 4.2 – Гистограммы платежей в серии из 10000 игр

Если мы сравним гистограммы, то увидим, что число выигрышей и число проигрышей одинаково для обеих сумм, поставленных на кон.

В рассмотренной игре оптимальные стратегии игроков суть: красного — $P^* = \left\{\frac{1}{3};\frac{2}{3}\right\}$, синего — $Q^* = \left\{\frac{1}{2};\frac{1}{2}\right\}$, цена игры — $H(P, Q) = 0$.

Таким образом, игра справедлива. Игрок, угадывающий шкатулочки, должен чередовать свои стратегии “фифти-фифти”, например, подбрасывая монетку. Якубович же должен применять свои стратегии в пропорции “один к двум”, воспользовавшись правильной игральной костью и положив выпадение тройки либо шестёрки за указание применять стратегию № 1 (то есть прятать фишку в левой шкатулочке), а в остальных случаях – пользоваться стратегией № 2 (правая шкатулка).

4.6. Поиск оптимальных стратегий в матричных играх

Совершенно очевидно, что непосредственное применение (4.1) для поиска оптимальных стратегий вельми неудобно, поэтому целесообразно было ожидать возникновения разнообразных методов решения, которые мы ниже и рассмотрим, это

- графо-аналитический метод;
- метод, основанный на построении эквивалентной ЗЛП;
- итерационный метод.

Будем подразумевать, что первым этапом поиска оптимальных стратегий, является нахождение решения в чистых стратегиях, и, если он закончился неудачей, придётся прибегнуть к вышепоименованным алгоритмам.

Перед началом изложения отметим (и напомним) особенности игровых задач, описываемых платёжными матрицами.

1. Матричная игра ***всегда*** имеет решение, согласно основной теореме.

2. ***Прибавление или вычитани***е действительного ***числа*** ко всем элементам платёжной матрицы ***не меняет*** пространства оптимальных стратегий.

3. Элементы платёжной матрицы ***могут быть преобразованы*** по формуле

$$a_{i,j} = k \cdot h_{i,j} + d\,, \qquad (4.3)$$

где d — любое число, k — любое положительное число. При этом оптимальные стратегии обоих игроков не изменятся, а цена игры, соответствующая исходной платёжной матрице составит

$$v_h = \frac{v_a - d}{k}. \qquad (4.4)$$

4. Число активных стратегий в $m \times n$ игре не бывает выше наименьшего из чисел n и m.

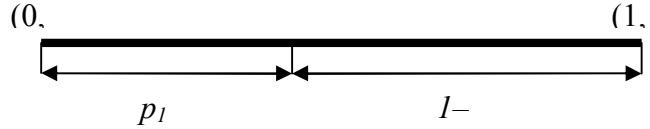
4.6.1. Графоаналитический метод решения игровых задач [27, 64]

Данный метод применяется в том случае, когда одна из размерностей платёжной матрицы равна двум. Первоначально, в качестве основания для дальнейших рассуждений, рассмотрим игру размерностью 2×2.

Матрица выигрышей в этом случае имеет вид

	q_1	q_2
p_1	h_{11}	h_{12}
p_2	h_{21}	h_{22}

Очевидно, что любую смешанную стратегию первого игрока можно представить в виде $P = \{p_1, p_2\} = \{p_1, 1 - p_1\}$, и отобразить единичным отрезком прямой.



Предположим, что Ян применяет смешанную стратегию, а Татьяна – только первую. Ожидаемый выигрыш Яна при этом составит

$$H(P, 1) = h_{12} \times p_1 + h_{21} \times (1 - p_1) = h_{21} - (h_{21} - h_{12}) \times p_1,$$

что в графической интерпретации выглядит как это показано на рисунке 4.2.

При фиксированной стратегии Татьяны выигрыш Яна перемещается вдоль прямой по мере использования различных вероятностей чистых стратегий в смешанной стратегии.

Указанное справедливо и для случая, когда Татьяна использует свою вторую чистую стратегию, а Ян – смешанную. В этом случае

$$H(P, 2) = h_{12} \times p_1 + h_{22} \times (1 - p_1) = h_{22} - (h_{22} - h_{12}) \times p_1,$$

что тако же представимо графически.

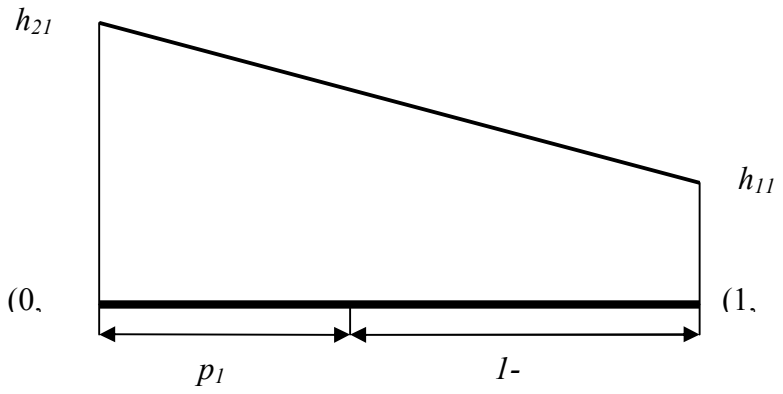


Рисунок 4.3 – Значение выигрыша при оптимальной стратегии 1-го игрока

Пересечение двух линий *H(P, 1)* и *H(P, 2)* на рисунке 4.4 даст точку, в которой находится оптимальное решение. Эта точка описывается системой уравнений:

$$\left.\begin{aligned} v_1 = H(P,1) = h_{21} - (h_{21} - h_{11})\,p_1^* \\ v_1 = H(P,2) = h_{22} - (h_{22} - h_{12})\,p_1^* \end{aligned}\right\}, \tag{4.5}$$

Вероятность стратегии, соответствующей максимальному выигрышу первого игрока определится при решении системы уравнений (4.5), откуда оптимальное значение вероятности применения 1-й чистой стратегии красного игрока составит

$$p_1^* = \frac{h_{22} - h_{21}}{h_{22} + h_{11} - h_{12} - h_{21}}. \tag{4.6}$$

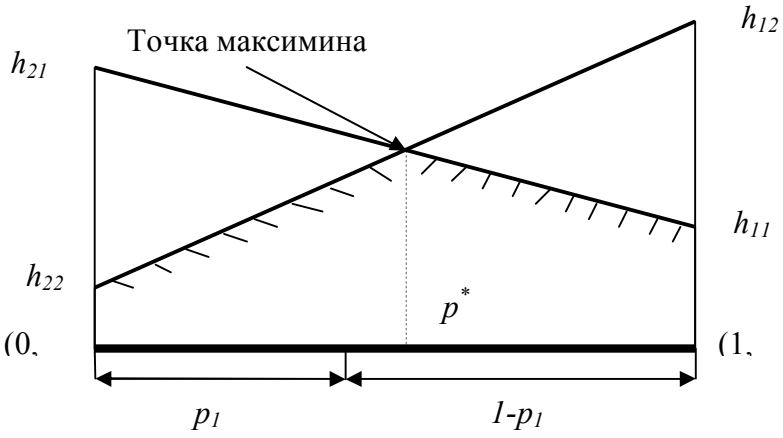


Рисунок 4.4 – Графическое решение для красного игрока

Аналогичные рассуждения можно проделать и для второго, синего игрока (см. рисунок 4.5), получим систему

$$\left.\begin{aligned} v_2 = H(1,Q) = h_{11}q_1^* + h_{12}(1-q_1^*) = h_{12} - (h_{12}-h_{11})q_1^* \\ v_2 = H(2,Q) = h_{21}q_1^* + h_{22}(1-q_1^*) = h_{22} - (h_{22}-h_{21})q_1^* \end{aligned}\right\}. \qquad (4.7)$$

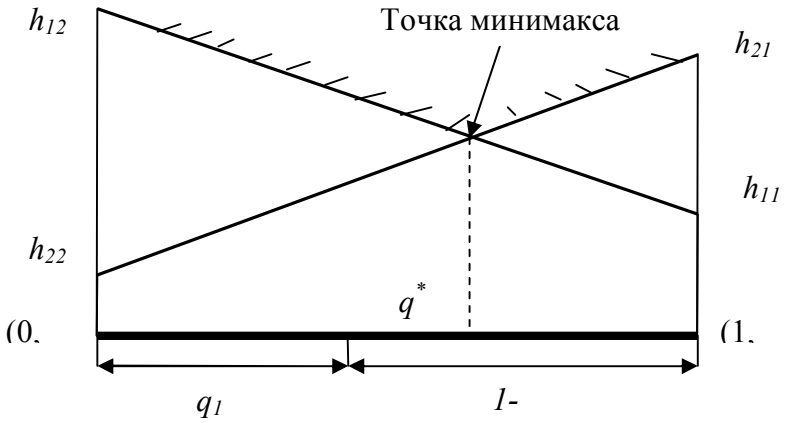


Рисунок 4.5 – Оптимальная стратегия синего игрока

Решение системы (4.7) даёт значение оптимальной вероятности использования синим игроком 1-й своей стратегии:

$$q_1^* = \frac{h_{22} - h_{12}}{h_{22} + h_{11} - h_{12} - h_{21}}. \qquad (4.8)$$

Цена игры, определяемое как математическое ожидание выигрыша первого игрока, можно найти, используя (4.1), либо одно из четырёх выражений, представленных системами (4.5) или (4.7), подставляя в них (4.6) или (4.8):

$$v^* = \frac{h_{11} \cdot h_{22} - h_{12} \cdot h_{21}}{h_{22} + h_{11} - h_{12} - h_{21}}. \qquad (4.9)$$

Для рассмотренной ранее задачи с Л.А. Якубовичем, платёжная матрица есть $H = \begin{pmatrix} -2 & 2 \\ 1 & -1 \end{pmatrix}$. Использование формул (4.6), (4.8) и (4.9) даёт

$$p_1^* = \frac{-1-1}{-2-2-1-1} = \frac{-2}{-6} = \frac{1}{3}; q_1^* = \frac{-1-2}{-6} = \frac{1}{2}; v = 0.$$

что совпадает с ответом, выведенным аналитически.

Однако, если использовать полученный результат для расчёта оптимальных стратегий при анализе конфликтной ситуации между Японией и США, то получим оптимальное решение, не соответствующее правильному:

$$p_1^* = \frac{3-1}{2+3-1-2} = \frac{2}{2} = 1; q_1^* = \frac{3-2}{2} = \frac{1}{2}; v = \frac{2\cdot 3 - 1\cdot 2}{2} = 2.$$

Из этого решения, видно, что у красного игрока стратегия чистая, а у синего игрока – как бы смешанная. Хотя в соответствии с теоремой об активных стратегиях это не влияет на цену игры. Отсюда можно сделать вывод, что применять метод к решению игр с седловой точкой следует осторожно.

Перейдём далее к рассмотрению матричных игр, у которых одно из измерений равно двум. Графическая часть алгоритма применяется для определения активных стратегий игроков и сведения платёжной матрицы, таким образом, к размерности 2×2, после чего выполняются численные расчёты.

Рассмотрим игру *2×n*. Платёжная матрица такой игры имеет вид:

$$H = \begin{bmatrix} h_{11} & h_{12} & \dots & h_{1n} \\ h_{21} & h_{22} & \dots & h_{2n} \end{bmatrix}.$$

Будем рассматривать игровую ситуацию с позиции красного игрока. Изобразим функции выигрыша 1-го игрока прямыми линиями, соединяющими точки (h_{2j} h_{1j}) на отрезке единичной длины, для смешанных стратегий 1-го игрока: $H(P, \text{j}) = h_{1j} \times p_1 + h_{2j} \times (1 - p_1)$.

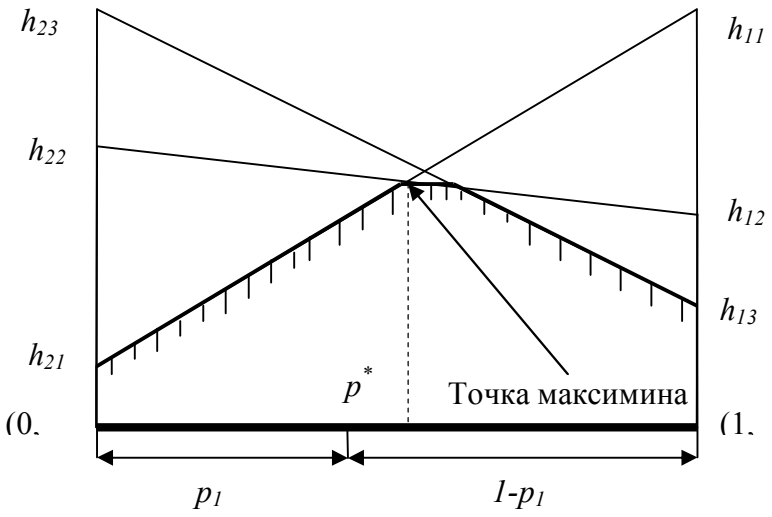


Рисунок 4.6 – Пояснение к решению игры 2 × *n*

На области, ограниченной отрезками прямых (h_{2j}, h_{1j}) и осью абсцисс, найдём пару стратегий, формирующих максимин. Линии, которые пересекаются в точке максимина, соответствуют активным стратегиям первого игрока. Для ситуации, отражённой на рисунке, стратегия № 3 синего игрока не является активной, поэтому может быть исключена из рассмотрения, а соответствующий столбец платёжной матрицы должен быть удалён.

В результате размерность платёжной матрицы сократилась до двух, и задача может быть решена применением формул (4.6), (4.8) и (4.9).

Если в точке максимина ***пересекаются более двух прямых***, то в качестве активных ***можно взять любую пару*** из них без потери точности решения.

Если возникает две точки максимина, то для определения пары активных стратегий ***можно взять любую точку***.

Рассмотрим игру $m\times2$. Платёжная матрица такой игры имеет вид:

$$H = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \\ \dots & \dots \\ h_{m1} & h_{m2} \end{bmatrix}.$$

Будем рассматривать игровую ситуацию с позиции синего игрока. Изобразим функции выигрыша 2-го игрока отрезками прямых вида (h_{i2}, h_{i1}) на отрезке единичной длины, что соответствует смешанным стратегиям 2-го игрока:

$$H(i, Q) = h_{i1} \times q_1 + h_{i2} \times (1 - q_1)$$

и отыщем точку минимакса, руководствуясь теми же соображениями, что и для решения матричной игры $2\times n$. После определения пары активных стратегий игроков остаётся лишь выполнить расчеты по формулам (4.6), (4.8) и (4.9). Точка минимакса показана на чертеже, показанном на рисунке4. 7.

Очевидно, что как и ранее, если в точке минимакса ***пересекаются более двух прямых***, то в качестве активных ***можно взять любую пару*** из них без потери общности решения.

Если возникает две точки минимакса (одна из прямых, участвующих в его формировании, параллельна оси абсцисс), то для определения пары активных стратегий ***можно взять любую точку***.

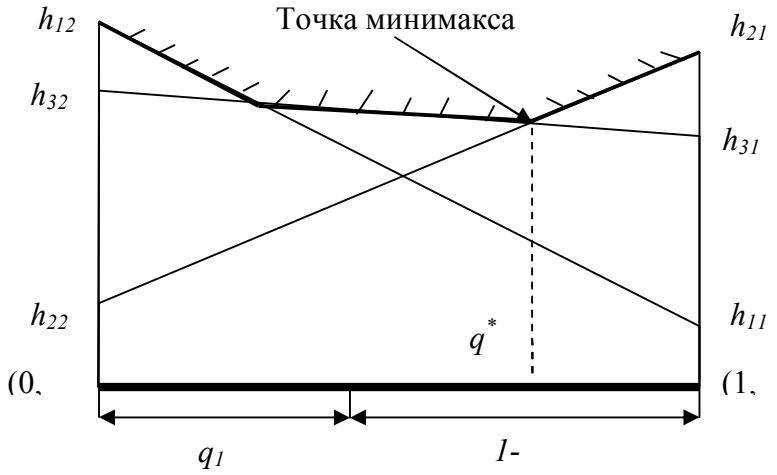


Рисунок 4.7 – Пояснение к решению игры $m \times 2$

Замечание. При значительном числе n или m поиск оптимальных стратегий становится визуально затруднительным, что должно компенсироваться выбором правильного размера и масштаба изображения.

4.6.2. Использование принципа доминирования для снижения размерности платёжной матрицы игровой задачи [64]

Определение. Вектор α размерности n строго доминирует вектор β размерности n, если каждая координата вектора α строго больше соответствующей координаты вектора β.

Принимая во внимание алгоритм нахождения равновесных ситуаций максимина и минимакса, изложенный нами ранее, можно сформулировать принцип доминирования следующим образом.

1. Если i-я строка платёжной матрицы строго ***доминируется*** некоторой выпуклой комбинацией других строк (в частности, если элементы некоторой строки больше соответствующих элементов строки i), то i-я строка может быть вычеркнута из матрицы без изменения множества оптимальных стратегий первого игрока.

2. Если j-й столбец платёжной матрицы строго ***доминирует*** некоторую выпуклую комбинацию других столбцов (в частности, если элементы столбца j больше соответствующих элементов некоторого другого столбца), то j-й столбец может быть вычеркнут из матрицы без изменения множества оптимальных стратегий второго игрока.

То есть, из множества стратегий игрока исключается та, которая принесёт заведомо худший, по сравнению с другими, результат: если одна стратегия игрока лучше, нежели другая, то худшая должна быть исключена. Для красного игрока – это строка, по стратегии,

соответствующей которой, получается заведомо меньший выигрыш. А для синего игрока стратегия, соответствующая вычеркнутому столбцу, ведёт к большему проигрышу.

Поочерёдное применение принципов доминирования позволяет, в ряде случаев, существенно снизить размерность платёжной матрицы и уменьшить, тем самым, объём расчётов.

Пример [64]. Пусть задана платёжная матрица вида.

24	2	10	30
3	8	7	9
4	4	7	15
2	3	3	1

Оценки цены игры суть $v_1 = \max_i \min_j h_{ij} = 4$, а $v_2 = \min_j \max_i h_{ij} = 8$. Элементы, сформировавшие максимин и минимакс выделены серым цветом. Следовательно, седловая точка в платёжной матрице отсутствует. Видно, что 4-я строка строго доминируется одновременно 2-й и 3-й строками. Поэтому соответствующая стратегия красного игрока активной не является, и может быть удалена. Получим

24	2	10	30
3	8	7	9
4	4	7	15

В этой матрице 4-й столбец строго доминирует любой другой, поэтому соответствующая стратегия синего игрока исключается из списка активных. Новая платёжная матрица:

24	2	10
3	8	7
4	4	7

Теперь 3-я строка строго доминируется выпуклой комбинацией 1-й и 2-й строк: $24^{(1)} > 4$, $8^{(2)} > 4$, $10^{(1)} > 7$, и исключается вместе со стратегией № 3 красного игрока. Получается

24	2	10
3	8	7

Теперь можно применять графоаналитический метод. Отметим, что в новой платёжной матрице понизилось значение максимина $v_1 = \max\limits_i \min\limits_j h_{ij} = 3$.

Однако, в полученной матрице 3-й столбец доминирует выпуклую комбинацию двух других: $10^{(3)} > 2^{(2)}$, $7^{(3)} > 3^{(1)}$. Справедливость этой операции может быть проверена читателем графически.

24	2
3	8

Очевидно, что размерность не всякой платёжной матрицы может быть уменьшено до 2-х.

4.6.3. Построение эквивалентной ЗЛП по платёжной матрице [27, 64]

Пусть имеется игра, заданная в нормальной форме, а элементы платёжной матрицы либо положительны, либо приведены к таковому виду на основании использования выражения (4.3).

Из теоремы об активных стратегиях следует, что для любой чистой *j-й* стратегии 2-го игрока при использовании первым игроком своей оптимальной стратегии выполняется неравенство.

$$a_{1j} p_1^* + a_{2j} p_2^* + \ldots + a_{mj} p_m^* \geq v. \qquad (4.10)$$

Знак “≥” возникает в неравенстве за счёт того, что *j*-я стратегия может не являться активной, а результат игры получится хуже. Неравенство (4.10) может послужить основой для построения системы ограничений

$$\begin{cases} a_{11} p_1^* + a_{21} p_2^* + \ldots + a_{m1} p_m^* \geq v, \\ a_{12} p_1^* + a_{22} p_2^* + \ldots + a_{m2} p_m^* \geq v, \\ \ldots \\ a_{1n} p_1^* + a_{2n} p_2^* + \ldots + a_{mn} p_m^* \geq v. \end{cases}$$

Прибегнув к нормировке, обозначим $x_i = \dfrac{p_i^*}{v}$, введём функцию цели вида

$$f(X) = \sum_{i=1}^{m} x_i = \frac{1}{v} \Rightarrow \min \tag{4.11}$$

и получим систему ограничений, пригодную для расчётов:

$$\begin{cases} a_{11}x_1 + a_{21}x_2 + \ldots + a_{m1}x_m \geq 1, \\ a_{12}x_1 + a_{22}x_2 + \ldots + a_{m2}x_m \geq 1, \\ \ldots \\ a_{1n}x_1 + a_{2n}x_2 + \ldots + a_{mn}x_m \geq 1. \end{cases} \tag{4.12}$$

Выражения (4.11) и (4.12) представляют собой формулировку ЗЛП, построенной по платёжной матрице.

Для синего игрока, который является антагонистом красного, может быть сформулирована двойственная задача линейного программирования.

$$\begin{gathered} f(X) = \sum_{i=1}^{n} y_i \Rightarrow \max, \\ \begin{cases} a_{11}y_1 + a_{12}y_2 + \ldots + a_{1n}y_n \leq 1, \\ a_{21}y_1 + a_{22}y_2 + \ldots + a_{2n}y_n \leq 1, \\ \ldots \\ a_{m1}y_1 + a_{m2}y_2 + \ldots + a_{mn}y_n \leq 1. \end{cases} \end{gathered} \tag{4.13}$$

В (4.13) обозначено $y_j = \frac{q_j^*}{v}$. Из теории решения двойственных задач известно, что вектор симплекс-разностей для дополнительных переменных соответствует вектору оптимальных значений переменных сопряжённой задачи. Пусть нами найдено оптимальное решение задачи (4.13). Тогда оптимальное решение матричной игры есть:

$P^* = \left\{ \frac{\delta_{n+1}^*}{f_{opt}}; \frac{\delta_{n+2}^*}{f_{opt}}; \ldots; \frac{\delta_{n+m}^*}{f_{opt}} \right\}$, $Q^* = \left\{ \frac{y_1^*}{f_{opt}}; \frac{y_2^*}{f_{opt}}; \ldots; \frac{y_n^*}{f_{opt}} \right\}$ где $f_{opt} = \frac{1}{v^*}$, v^* — цена игры.

Пример. Пусть платёжная матрица задачи имеет вид

5	1	1
2	3	1
3	2	4

Оценки игровой ситуации: $v_1 = \max\limits_i \min\limits_j h_{ij} = 2$, $v_2 = \min\limits_j \max\limits_i h_{ij} = 3$. По данному условию может быть составлена ЗЛП вида

$$f_{\max} = y_1 + y_2 + y_3,$$
$$\begin{cases} 5y_1 + 1y_2 + 1y_3 \le 1, \\ 2y_1 + 3y_2 + 1y_3 \le 1, \\ 3y_1 + 2y_2 + 4y_3 \le 1. \end{cases}$$

В ходе её решения получилась оптимальная симплекс-таблица

		c_j	1	1	1	0	0	0
Базис	$C_Б$	A_0	A_1	A_2	A_3	A_4	A_5	A_6
A_1	1	3/20	1	0	0	1/4	-1/20	-1/20
A_2	1	9/40	0	1	0	-1/8	17/40	-3/40
A_3	1	1/40	0	0	1	-1/8	-7/40	13/40
	δ_j	2/5	0	0	0	0	1/5	1/5

Из последней получаем цену игры $v = \frac{5}{2} = 2{,}5$ и оптимальные смешанные стратегии игроков: красного $P^* = \left\{0; \frac{1}{2}; \frac{1}{2}\right\}$ и синего $Q^* = \left\{\frac{3}{8}; \frac{9}{16}; \frac{1}{16}\right\}$. У красного игрока, таким образом, две активных стратегии.

4.6.4. Итерационный метод решения матричной игры с нулевой суммой

При большом числе стратегий игроков решение эквивалентной ЗЛП представляется вычислительно трудоёмкой, требующей применения вычислительной техники.

Итерационный метод основывается на имитации серии игр, требует использования арифметики и наличия времени, необходимого для выполнения расчётов. Внешне ситуация напоминает игру крокодила Гены в шахматы с самим собой, сюжет визуализирован в 1-й части Советского мультблокбастера о похождениях Чебурашки со товарищи.

Применение метода состоит в попеременном выполнении за играющие стороны ходов. При этом синий игрок стремится уменьшить выигрыш красного в среднем (принцип минимакса), а красный – наоборот, увеличить его (принцип максимина).

1. Первый ход красного игрока может быть сделать случайно, либо по строке с максимальным средним значением выигрыша. Результаты хода, с разбивкой по стратегиям синего игрока, фиксируется в строке итога.

2. Ответный ход синих выбирается по элементу с наименьшим значением выигрыша, результат запоминается в столбце итога, разнесённый по стратегиям красного игрока.

3. Последующие ходы выполняются игроками поочерёдно, руководствуясь правилами максимина и минимакса. Результаты ходов суммируются накопительно, а выбираемые стратегии запоминаются.

4. По прошествии заданного числа ходов, игра прекращается, и подсчитываются относительные частоты использования тех или иных стратегий конфликтующими сторонами и цена игры.

Пусть сторонами проведено n игр. Тогда оптимальная смешанная стратегия 1-го игрока есть

$$\widetilde{p}_i = \frac{m_i^{red}}{n}, \tag{4.14}$$

где m_i^{red} — число использований красным игроком своей i-ой чистой стратегии. Для второго игрока аналогично

$$\widetilde{q}_j = \frac{m_j^{blue}}{n}, \tag{4.15}$$

где m_j^{blue} — число использований синим игроком своей j-ой чистой стратегии.

Оценка цены игры находится как

$$\widetilde{v} = \frac{\widetilde{v}_1 + \widetilde{v}_2}{2 \times n}. \tag{4.16}$$

В выражении (4.16) $\widetilde{v}_1$ и $\widetilde{v}_2$ оценки суммарного максимина и минимакса, получающиеся в ходе накопления результатов ходов.

Результаты игры представляют совокупностью таблиц, заголовки которых определяются чистыми стратегиями игроков, а строки

соответствуют ходам игроков и содержат суммарные текущие значения выигрыша красного игрока.

Иногда такие таблицы представляют в виде комплекса, Г-образно, как это будет показано ниже. Рассмотрим пример из предыдущего раздела с платёжной матрицей вида:

5	1	1
2	3	1
3	2	4

Итерационное решение игры, в ходе которой было проведено 20 розыгрышей, показано ниже.

5	1	1	1	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	*0*
2	3	1	1	3	6	9	12	**15**	**18**	**21**	22	**25**	**28**	29	32	**35**	**38**	39	42	**45**	**48**	49	*9*
3	2	4	**4**	**7**	**9**	**11**	**13**	15	17	19	**23**	25	27	**31**	**33**	35	37	**41**	**43**	45	47	**51**	*11*
2	3	**1**																					
5	5	6																					
8	**7**	7																					
11	**9**	11																					
14	**11**	15																					
17	**13**	19																					
19	**16**	20																					
21	**19**	21																					
23	22	**22**																					
26	**24**	26																					
28	**27**	27																					
30	30	**28**																					
33	**32**	32																					
36	**34**	36																					
38	**37**	37																					
40	40	**39**																					
43	**42**	43																					
46	**44**	47																					
48	**47**	48																					
50	50	**49**																					
1	*14*	*5*																					

В ходе игры, стратегии, используемые игроками в текущем розыгрыше, выделены жирным шрифтом и серым фоном.

Курсивом справа от боковика таблицы показано количество использования красным игроком каждой из своей стратегий, а внизу хвостовика таблицы такая же информация представлена по синему игроку. Суммарные оценки игры по максимину и минимаксу суть $\widetilde{v}_1 = 49$ и $\widetilde{v}_2 = 51$.

Имеем, по (4.16), оценку цены игры $\widetilde{v} = \dfrac{49+51}{2\times 20} = 2{,}5$.

Оценки оптимальных стратегий игроков, используя (4.14) и (4.15), составят:

для красного — $\widetilde{P}^* = \left\{0; \frac{9}{20}; \frac{11}{20}\right\}$, а для синего — $\widetilde{Q}^* = \left\{\frac{1}{20}; \frac{7}{10}; \frac{1}{4}\right\}$.

Сравнивая результаты расчётов – точного по ЗЛП и итерационного метода, можно отметить: цена игры совпала, значения вероятностей применения стратегий красного игрока практически совпали, для синего результаты несколько хуже.

Примечание. Для повышения точности результатов, надо сыграть изрядное число игр [66]. Математическая статистика гласит, что устойчивость среднего наступает после серии не менее 50 экспериментов, а устойчивость по дисперсии – после 400.

Очевидно, что бОльший размер платёжной матрицы потребует бОльшего объёма розыгрышей.

4.7. Конечные позиционные игры двух персон [64, 89]

К понятию позиционных игр можно прийти следующим образом. Необходимо отображать динамику действий, связанную с дополнительным приобретением или потерей информации, изменением игровой ситуации, ставок, возможных расценок и д.п.

Подобные ситуации моделируются теоретико-игровыми моделями, которые называются ***антагонистические позиционные игры***. В ходе игры (процессе развития игры) стороны проходят ***последовательно*** конечное число этапов, на каждом из которых необходимо принимать некоторое частное решение.

Ход – выбор игроком одной из его альтернатив.

На каждом этапе ход выполняется только одним игроком. Сами ходы ***бывают личными*** и случайными.

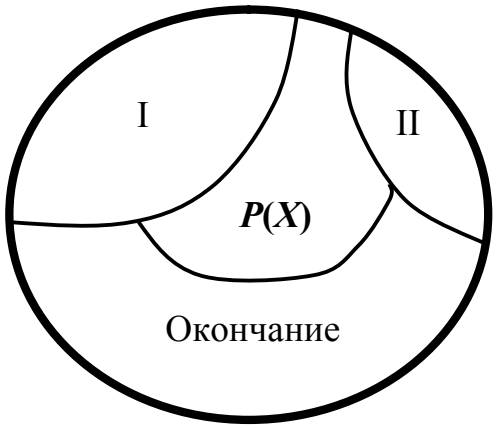
Личный ход – ***сознательный выбор*** игроком одной из имеющихся в его распоряжении альтернатив.

Случайный ход – отражение закономерностей случайных событий или величин.

Предполагается, что случайный ход выполняет природа, фактически не заинтересованная в чьей либо победе. Для этого случая задаётся распределение вероятностей на множестве всех альтернатив природы. Указанное множество априори известно.

Позиция – игровая ситуация, в которой игроки оказываются в результате совершения своих ходов.

Множество всех позиций разбивается на подмножества:
смешанных стратегий.



- а) позиции, принадлежащие 1-му игроку (Яну);
- б) позиции, принадлежащие 2-му игроку (ТатьЯне);
- в) позиции со случайными ходами, принадлежащие природе;
- г) окончательные позиции, в которых игра завершена и определяется выигрыш (проигрыш) игроков.

Позиционная игра будет являться ***конечной*** при ***конечном числе*** позиций и альтернатив. Такая игра представима в виде ***дерева***, называемого ***деревом игры***, вариант какового представлен на рисунке 4.8.

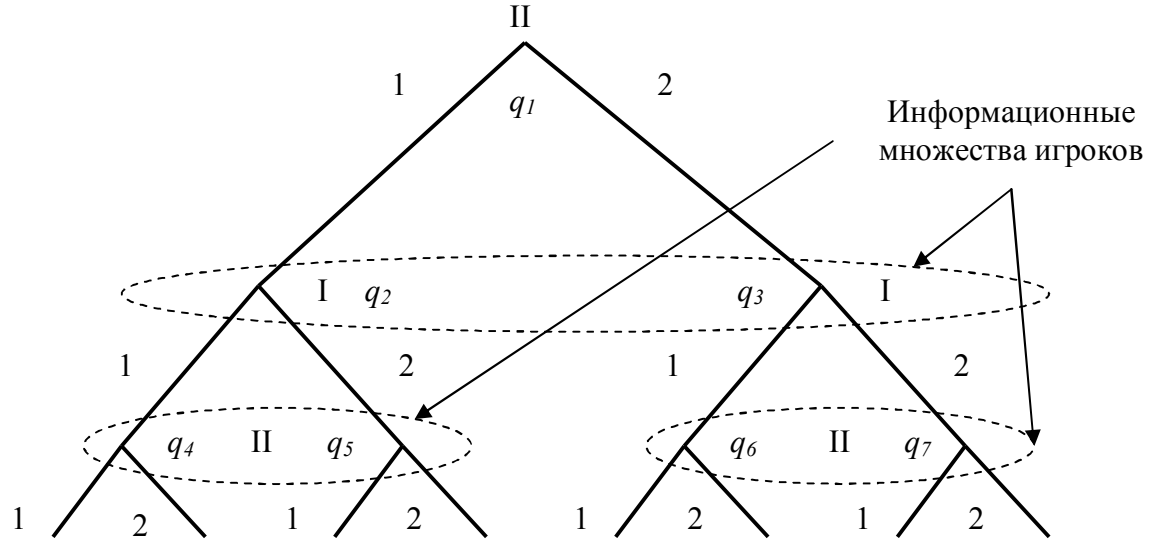


Рисунок 4.8 – Дерево конечной позиционной игры

Партия – путь на дереве игры от начальной вершины к заключительной вершине.

Если игрок при выборе своего очередного хода знает результаты предыдущих ходов, как своих, так и противника, то игра является игрой с ***полной информацией***. В противном случае игра называется игрой с

неполной информацией, и игрок может определить своё местоположение с точностью до некоторого множества вершин игрового дерева.

Такое множество называется ***информационным множеством*** игрока.

Свойства информационного множества [64]

- все позиции одного и того же информационного множества принадлежат одному и тому же игроку;
- все позиции одного и того же информационного множества должны иметь одинаковое число альтернатив (в противном случае игрок определится со своим местоположением бале или менее точно);
- одно информационное множество не должно содержать позиций различных этапов одной и той же партии игры.

Эти множества показаны пунктиром на рисунке 4.8.

Стратегии игроков в позиционных конечных играх суть следующие.

1. **Чистая стратегия**. Выбирается до начала игры в форме “если — то” перебором путей по дереву игры. В ходе перебора игра сводится к матричной нормальной форме. Если ***игра*** является игрой с ***полной информацией***, то имеется седловая точка платёжной матрицы и решение в чистых стратегиях.

2. **Стратегия поведения**. Представляет собой набор из r вероятностных распределений, каждое из которых задано на множестве возможных альтернатив в каждом информационном множестве. В общем случае, при наличии большого числа информационных множеств и альтернатив в них, решение задачи отыскания стратегий поведения нетривиально.

3. **Смешанная стратегия**. В соответствии с ранее данным определением, позволяет реализовывать случайный выбор на множестве чистых стратегий. Если ***игра*** является игрой ***с полной памятью***, то смешанная стратегия эквивалентна стратегии поведения. В этой ситуации необходимо решать ЗЛП, построенную по платёжной матрице. Реализация стратегии поведения предпочтительнее, чем смешанной стратегии.

Уместна следующая аналогическая связь этих стратегий.

Чистая стратегия представляет собой книжку инструкций, каждая страница которой относится к одному из информационных множеств, и где чётко и точно прописано, что делать игроку в том или ином информационном множестве.

Множество чистых стратегий, таким образом, представляет собой библиотеку, из которой, посредством смешанной стратегии, выбирается книжка строгих инструкций.

Стратегия поведения тако же представляет собой книжку, на каждой странице которой содержится распределение вероятностей по

альтернативам соответствующего информационного множества, а не жёсткую инструкцию.

Формальное описание антагонистической позиционной игры[64]

Описание игры предусматривает задание системы компонент, в которую входят.

1. Конечное дерево с выделенной вершиной, называемой начальной позицией игры.
2. Функция выигрыша красных, которая ставит в соответствие с каждой окончательной позицией (окончательной вершиной дерева) некоторый выигрыш красного игрока I.
3. Разбиение всех позиций по принадлежности к игрокам.
4. перечисление альтернатив в каждой позиции в каждом информационном множестве.
5. Вероятностные распределения на множестве альтернатив по каждому случайному ходу (выполняемому природой).
6. Подразбиение позиций на информационные множества, при этом
    - позиции множества имеют одинаковое число следующих за ними альтернатив;
    - никакая другая позиция не может следовать за другой позицией из этого информационного множества.

Пример, навеянный произведением А.Н. Толстого “Золотой ключик”.

Папа Карло с Мальвиной, Буратино и прочей кукольной братией укрываются от Карабаса-Барабаса с Дуремаром и их приспешниками.

У беглецов имеются две альтернативы: поехать по правой дороге (П) или левой дороге (Л). В свою очередь, Карабас тоже имеет два варианта противодействия беглецам – выставить засады – блокпосты на путях следования кукол, это

- КДБА – Карабас и Дуремар располагаются на левой дороге, а кот Базилио с лисой Алисой на правой; и
- БАКД – кот Базилио с лисой Алисой располагаются на левой дороге, а Карабас и Дуремар на правой.

При столкновении с засадой, когда будут ясен состав блокпоста, папа Карло со товарищи, может прорываться через заслон (атаковать, А) или спасаться бегством (Б).

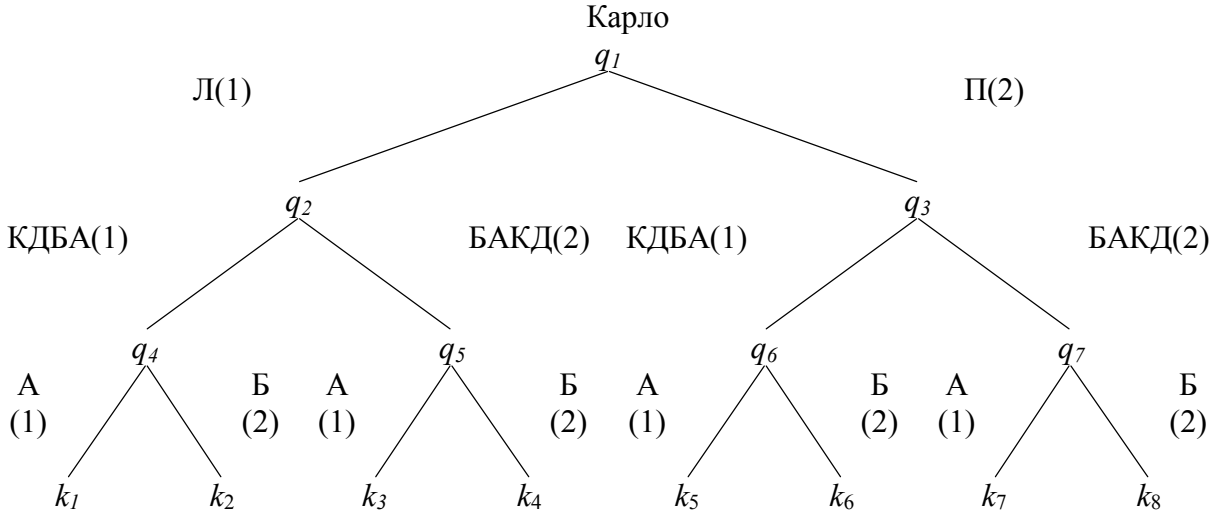
Таким образом, формально игра состоит из этапов:

1. Выбор пути следования папы Карло.

2. Расстановка Карабасом групп захвата.

3. Выбор способа действий папы Карло и его спутниками при попадании в засаду.

По условию задачи строим следующее игровое дерево, показанное ниже.



В качестве функции выигрыша Карабаса-Барабаса примем вероятности пленения кукольной труппы.

Конечная позиция	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8
Выигрыш	0,1	0,8	0,4	0,6	0,5	0,2	0,3	0,7

Обычно (в жизни) участники конфликта принимают решения по мере развития конфликтной ситуации во времени.

Однако, абстрагируясь от этого практического соображения, будем считать, что игроки учли все возможные обстоятельства до начала игры и подготовили (запаслись домашними заготовками) набор алгоритмов действия (чистые стратегии) в форме “если — то”.

Формально, если стратегии пронумерованы, чистая стратегия определяется на совокупности информационных множеств и является функцией, приписывающей множеству число из интервала [1, k_i], где k_i — число альтернатив в i-м информационном множестве игрока. Таким образом, общее число чистых стратегий у игрока равно

$$k_1 \cdot k_2 \cdot \ldots \cdot k_r = \prod_{i=1}^{r} k_i \, .$$

Если стратегия игры приводит к окончательной вершине t, то выигрыш игрока составит величину $M(t)$ при отсутствии случайных ходов. Если в игре присутствуют случайные ходы, то математическое ожидание выигрыша при этом составит

$$\sum_t M(t)P(t), \tag{4.17}$$

где $P(t)$ - вероятность того, что игра окончится в позиции t.

Так как число позиций на дереве игры конечно, мы можем, в любом случае, свести игру к матричной, перечислив все стратегии игроков в виде иерархической системы чисел

$$[i\ (i_1,\ i_2,\ ,\ \dots\ ,\ i_k,,\dots\ ,\ i_m)],$$

что означает выполнить ход с номером i на первом этапе, а на втором использовать ход i_k, если противник использовал k-ю альтернативу.

Например, запись [1 (1, 1)] означает: использовать на 1-м шаге 1-ю стратегию, а на 2-м использовать 1-ю, не зависимо от действий противника. Или [2 (2, 1)] означает: использовать на 1-м шаге 2-ю стратегию, а на 2-м использовать 2-ю, если противник использовал накануне 1-ю, и 1-ю, если противником была использована 2-я.

Отсюда перечень стратегий папы Карло есть [1 (1, 1)], [1 (1, 2)], [1 (2, 1)], [1 (2, 2)], [2 (1, 1)], [2 (1, 2)], [2 (2, 1)], [2 (2, 2)]. Остаётся лишь проследить по дереву игры, в какие конечные вершины нас приведут указанные стратегии. Таким образом, имеем следующую матричную игру, представленную в нормальной форме

	1(1,1)	1(1,2)	1(2,1)	1(2,2)	2(1,1)	2(1,2)	2(2,1)	2(2,2)
	Л(А,А)	Л(А,Б)	Л(Б,А)	Л(Б,Б)	П(А,А)	П(А,Б)	П(Б,А)	П(Б,Б)
	1	2	3	4	5	6	7	8
КДБА	k_1	k_1	k_2	k_2	k_5	k_5	k_6	k_6
(1)	0,1	0,1	0,8	0,8	0,5	0,5	0,2	0,2
БАКД	k_4	k_4	k_3	k_4	k_7	k_8	k_7	k_8
(2)	0,6	0,6	0,4	0,6	0,3	0,7	0,3	0,7

Платёжная матрица имеет седловую точку, показанную серым цветом. Цена игры – вероятность захвата кукол-беглецов Карабасом-Барабасом составляет 0,3.

Противники должны использовать свои стратегии следующего содержания.

- В засаде на левом пути должны располагаться лиса Алиса и кот Базилио, а на правом пути – Карабас-Барабас с Дуремаром.
- Папа Карло должен для следования выбрать правый путь, при встречи с котом и лисой – спасаться бегством, а при виде Дуремара и Карабаса – пытаться прорваться напролом.

  Приведённый содержательный пример служит иллюстрацией теоремы.

Теорема [64]. Всякая игра с полной информацией, представленная в нормальной форме, имеет седловую точку и решение в чистых стратегиях.

4.7.1. Игры с полной информацией и игры с полной памятью

Игрок I (или игрок II), если для него игра является игрой с полной информацией, имеет дело с ситуацией, когда ***каждое*** его информационное множество состоит лишь из 1-го элемента.

Игра является игрой ***с полной информацией***, если в ней ***каждый игрок*** имеет ***полную*** информацию.

Ранее нами отмечено, что всякая такая игра имеет в нормальной форме седловую точку в платёжной матрице.

Игрой с полной памятью называется игра, в которой ***каждый*** из ***игроков*** помнит всё, что оно делал или знал на каждом этапе.

Однако, в отличие от игр с полной информацией, он ***может не знать***, какой выбор сделал его противник.

Когда противники ординарные (единичные, не делимые) это, как правило, соблюдается всегда, хотя и наблюдаются примеры иного свойства, например потеря кораблём своего места из-за ошибок в счислении или отказе навигационного оборудования. В этом случае, имеем игру с неполной памятью.

Если одна из противоборствующих сторон представляет собой группу из 2-х и более участников, то есть, является составной, а взаимодействие между членами группы, по объективным причинами, отсутствует, то такой игрок может попеременно вспоминать и забывать свои действия и их результаты на предыдущих этапах. В этом случае, тако же имеем игру с неполной памятью.

Формально игрок ***имеет полную память***, если для двух его любых информационных множеств U и V, одно из которых расположено на предыдущем этапе игры (выше на дереве игры) в другое множество ***можно попасть по единственной альтернативе***.

Для случая игры с полной памятью, любой вершины множества $\{q_4, q_5\}$ можно достичь, двигаясь по 1-му (левому) ребру дерева, представленному

на рисунке 4.8. Совершенно аналогично, двигаясь по 2-му правому ребру этого дерева, можно достичь вершин $\{q_6, q_7\}$.

Для случая игры с неполной памятью, вершины q_4 из информационного множества $\{q_4, q_5, q_6, q_7\}$ нельзя достичь только по одному ребру.

Поэтому, 1-й игрок в одном случае может иметь полную память, а в другом – не иметь её.

Игры с полной памятью удобны тем, что каждый игрок может обойтись стратегиями поведения, описание которых и реализация проще, чем в случае

Сущность упрощения заключается в том, что ***один выбор*** из $k_1 \times k_2 \times \ldots \times k_i \times \ldots \times k_r$ возможных чистых стратегий заменяется r выборами из k_i возможных альтернатив в каждом информационном множестве.

По существу, ***стратегия поведения*** игрока есть функция, ***определённая на классе его информационных множеств***, которая назначает для каждого информационного множества распределение вероятностей альтернатив этого множества.

Поэтому, ***если*** у игрока только ***одно информационное множество***, то его ***стратегия поведения эквивалентна смешанной стратегии***.

Пример, навеянный книгой замечательного детского писателя А. Гайдара “Тимур и его команда”.

Пусть вожаки малолетних хулиганов Мишка Квакин и Пётр Пятаков (Фигура) организуют, действуя последовательно, налёт на сад, обороняемый Тимуром и С°. Пусть каждая из разбойных шаек имеет по 2 способа налёта (проникновения в сад), допустим, 1 – через дыру в заборе, 2 – ползком по канаве. Тимур, в свою очередь, из-за малочисленности своих сторонников, имеет возможность блокировать только один из путей.

Игра, по существу, состоит из трёх этапов:

- I: нападает группа Мишки Квакина;
- II: Тимур и С° противодействуют налётчикам;
- III: нападает группа, ведомая Фигурой.

В зависимости от степени доверительного отношения нападающих, возможны две теоретико-игровые модели ситуации.

1. Пусть Квакин и Фигура взаимодействуют и договариваются о совместных действиях при налёте на сад. В этом случае, имеем игру с ***неполной информацией,*** но с ***полной памятью***. Соответствующее игровое дерево показано на рисунке 4.9. Серым цветом выделены информационные множества игроков.

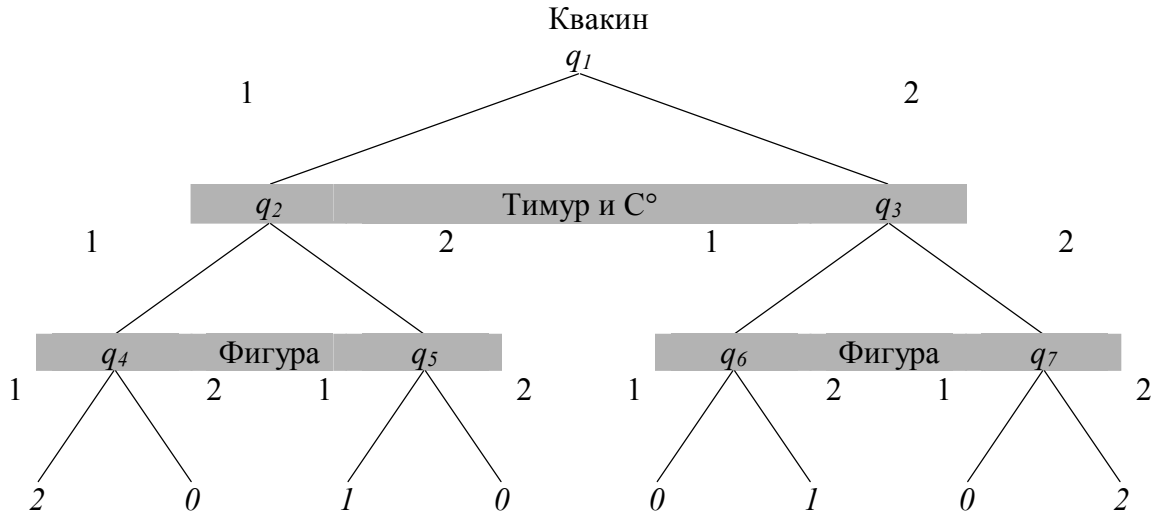


Рисунок 4.9 – Игра с неполной информацией и полной памятью

2. Пусть Квакин и Фигура не взаимодействуют, и выбор способа проникновения в сад ими не согласовывается. Тогда на III-м этапе составной игрок Квакин+Фигура “не помнит” своего хода, сделанного на I-м этапе. Следовательно, ***полной памяти он не имеет***, а игра может быть отнесена к играм ***с неполной информацией и неполной памятью***. Дерево показано на рисунке 4.10, серым цветом выделены информационные множества игроков.

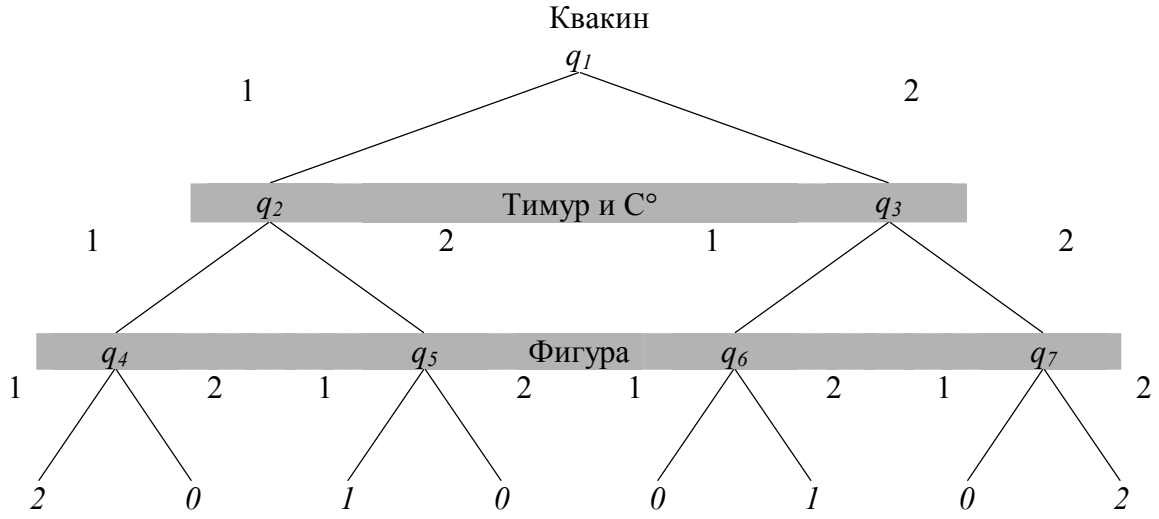


Рисунок 4.10 – Игра с неполной информацией и неполной памятью

На деревьях игры 4.9 и 4.10 в качестве платежей показано количество групп Квакин+Фигура, которым удаётся прорваться в сад.

Для игры с полной памятью, стратегия поведения 1-го игрока (Квакин+Фигура) есть система функций распределения по альтернативам информационных множеств

$$\left.\begin{aligned} f(q_1) &= (\alpha_1, 1-\alpha_1), \\ f(q_4, q_5) &= (\alpha_2, 1-\alpha_2), \\ f(q_6, q_7) &= (\alpha_3, 1-\alpha_3), \end{aligned}\right\} \tag{4.18}$$

где α_i — вероятность выбора 1-й стратегии в соответствующем информационном множестве.

У 2-го игрока стратегия поведения и смешанная стратегия совпадают:

$$g(q_2, q_3) = (\beta, 1-\beta). \tag{4.19}$$

Математическое ожидание 1-го игрока, при известных распределениях вероятностей, можно вычислить по дискретному аналогу формулы (4.17)

$$\begin{aligned} H(f,g) = 2\cdot\alpha_1\alpha_2\beta + 1\cdot\alpha_1\alpha_2(1-\beta) + 1\cdot(1-\alpha_1)(1-\alpha_3)\beta + \\ + 2\cdot(1-\alpha_1)(1-\alpha_3)(1-\beta), \end{aligned} \tag{4.20}$$

получаемой при обходе дерева.

И.В. Романовский, в своей работе [62], показал, что для игр с полной памятью всякая смешанная стратегия эквивалентна стратегии поведения. Таким образом, игра сводится к нормальной форме матричной игры путём перечисления альтернатив.

		Тимур и С°		
Квакин+Фигура	Стратегии	1	2	
	1=[1(1, 1)]	2	1	
	2=[1(1, 2)]	2	1	
	3=[1(2, 1)]	0	0	← Не активная стратегия
	4=[1(2, 2)]	0	0	← Не активная стратегия
	5=[2(1, 1)]	0	0	← Не активная стратегия
	6=[2(1, 2)]	1	2	
	7=[2(2, 1)]	0	0	← Не активная стратегия
	8=[2(2, 2)]	1	2	

Решение эквивалентной ЗЛП для данного случая даёт значение оптимальных смешанных стратегий игроков:

$$P^* = \left\{\frac{1}{2},0,0,0,0,\frac{1}{2},0,0\right\},\quad Q^* = \left\{\frac{1}{2},\frac{1}{2}\right\},$$
$$v = \frac{3}{2}.$$

Раздолбаи, однако, эти Квакин и Фигура, в заведомо выгодной для них стратегической конфигурации, проиграли слабому числом, но сильному духом Тимуру. Воистину, знания – сила.

Таким образом, 1-я и 6-я стратегии Квакин+Фигура должны применяться равновероятно на I-м этапе, далее, в информационном множестве $\{q_4, q_5\}$ – выбирается только 1-я, а в информационном множестве $\{q_6, q_7\}$ – выбирать только 2-ю. Последнее следует из анализа ситуаций на дереве игры. Поэтому стратегия поведения игрока Квакин+Фигура, эквивалентная его смешанной стратегии, то есть

$$\alpha_1 = \frac{1}{2}; \alpha_2 = 1; \alpha_3 = 0.$$

Для модели игры неполной информацией и неполной памятью, соответствующей рисунку 4.10, имеем функции распределения вероятностей в информационных множествах

$$\left.\begin{aligned} &f(q_1) = (\alpha_1, 1-\alpha_1), \\ &f(q_4,q_5,q_6,q_7) = (\alpha_2, 1-\alpha_2), \\ &g(q_2,q_3) = (\beta, 1-\beta), \end{aligned}\right\} \qquad (4.21)$$

совместно с функцией выигрыша, аналогичной (4.20).

$$\begin{aligned} H(f,g) = 2\cdot\alpha_1\alpha_2\beta + 1\cdot\alpha_1\alpha_2(1-\beta) + 1\cdot(1-\alpha_1)(1-\alpha_2)\beta \\ + 2\cdot(1-\alpha_1)(1-\alpha_2)(1-\beta). \end{aligned} \qquad (4.22)$$

В этом случае придётся использовать методы НП-программирования для отыскания оптимума функции (4.22).

Имеем градиент функции выигрыша

$$\nabla H(f,g) = \{3\alpha_2 - 2 + \beta; 3\alpha_1 - 2 + \beta; \alpha_1 + \alpha_2 - 1\} = 0,$$

откуда следует, что $\alpha_1 = \alpha_2 = \beta = 0{,}5$. Матрица Гёссе для рассматриваемого случая есть

$$\nabla^2 H(f,g) = \begin{bmatrix} 0 & 3 & 1 \\ 3 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix},$$

угловые миноры суть (– 1, – 1, – 9), что говорит о том, что исследуемая точка является точкой перегиба, что характерно для седла функций.

Цена игры составит 0,125 × (2 + 1 + 1 + 2) = 0,75. Таким образом, отсутствие полной памяти стоит, в нашем случае, половины выигрыша.

4.8. Многошаговые игры

Действия сторон иногда принимают форму чередующихся циклов. В общем случае, ***схема цикла*** такова:

- выполняется ход I-й играющей стороной из конечного множества альтернатив;
- не зная выбора противника, выполняется ход II-й играющей стороной;
- следует случайное событие, после чего каждый игрок получает информацию о действиях, предпринятых другой стороной и об их результатах;
- после оценки результатов принимается решение на возобновление игрового цикла (продолжить игру) или прекращение игры.

Пример [64] из боевых действий, преследование подводной лодки (ПЛ) морским (малым) охотником (МО): уклонение ПЛ, заход МО на бомбометание и сброс бомб. После этого оба игрока узнают о действиях друг друга и о результатах.

Таким образом, на каждом шаге игры разыгрывается игра с нулевой суммой, которая называется "***игрой компонентой***" (синонимы: ***игровой элемент***, ***игровая позиция***).

Число таких ***шагов*** может быть различным: конечным или бесконечным, фиксированным или нефиксированным заранее, но ***длительность партии*** обычно ***определяется реализацией*** некоторого ***случайного события***.

Если в партии повторяется ***одна*** и та же игра-компонента, то игра называется ***однокомпонентной***, в ***противном*** случае – ***многокомпонентной***.

Известны классы многошаговых игр:

- детерминированные;
- рекурсивные;
- стохастические.

4.8.1. Детерминированные игры

В данном классе игровых моделей задаётся, заранее, число шагов, не превышающее некоторое фиксированное значение, N и, как правило, одна игра-компонента.

Игра-компонента детерминированной игры задаётся $m \times n$ платёжной матрицей, элементы которой представляют собой либо обычные выигрыши, либо являются играми, разыгрываемыми при использовании игроками соответствующей пары стратегий.

Пусть, например, на первом шаге игры игра-компонента выглядит

$$\Gamma_N \sim \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & \Gamma_{N-1} \end{bmatrix},$$

где Γ_{N-1} — такой исход игры-компоненты Γ_N, при которым игроки должны разыграть игру-компоненту

$$\Gamma_{N-1} \sim \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & \Gamma_{N-2} \end{bmatrix}$$

и так далее.

Если в такой игре пара чистых стратегий приводит к элементу Γ_{N-k-1} на шаге $N - k$, $(k = 0, 1, \ldots, N-2)$, то игра закончится через N шагов, в противном случае, на $N - k$-ом шаге, получаем обычный выигрыш. Поэтому результат игры зависит от действий игроков на каждом шаге, а не определяются случаем.

Одним из интересных представителей этого класса игр выступает игра на разорение.

В ***общем случае***, постановка задачи такова. Играют двое игроков, в распоряжении которых имеется r и R-r ресурсов соответственно, и конечное число стратегий. В результате розыгрыша, на каждом шаге игры-компоненты, ресурсы одного из игроков увеличиваются на 1-цу, а другого – на 1-цу уменьшаются. При этом общий объём ресурсов тако же

изменяется в сторону увеличения или уменьшение. Выигрывает тот, кто первым достигнет заданного уровня, либо разорит своего противника.

В [66], показано, что если вероятности успеха и неуспеха в отдельном розыгрыше игроков равны p и q соответственно, то вероятность разорения определяется выражением

$$q_r = \begin{cases} \dfrac{\left(\dfrac{q}{p}\right)^R - \left(\dfrac{q}{p}\right)^r}{\left(\dfrac{q}{p}\right)^R - 1}, & p \neq \dfrac{1}{2}, \\ \dfrac{R-r}{R}, & p = \dfrac{1}{2}. \end{cases}$$

Пример. Пусть игроки обладают следующими объёмами ресурсов:

- ресурс “красного” r,
- ресурс “синего” $R – r$.

Правила игры на разорение таковы:

- если игроки одновременно выбирают i-ю чистую стратегию, то на единицу уменьшаются ресурсы II-го игрока;
- если игроки выбирают разные чистые стратегии, то на единицу уменьшаются ресурсы I-го игрока;
- игра завершается, когда будут исчерпаны ресурсы одного из играющих;
- выигрыш красного игрока, по завершению партии, составит + 1, если он разорит противника, и – 1, если разорится сам.

Общий подход к решению многошаговых игр состоит в том, что они решаются “задом наперёд”, от заключительного шага игры к начальному. В ходе рассмотрения этих шагов крайне желательно вывести (найти) рекуррентное выражение. Если рекуррентное выражение построить не удаётся, то оптимальные стратегии и цену игры приходится определять, перерешав все игровые ситуации. Последнее легко осуществимо лишь для игр с небольшой размерностью. Очевидно, что с ростом размерности игровых матриц и числа шагов, будут расти вычислительные расходы.

Обозначим как $\Gamma_{k,l}$, $1 \le k \le r$, $1 \le l \le R - r$ игру-компоненту, которой I-й игрок имеет k единиц ресурсов, а II-й игрок — l единиц ресурсов. Матрица такой игры есть

$$\Gamma_{k,l} \sim \begin{bmatrix} \Gamma_{k,l-1} & \Gamma_{k-1,l} \\ \Gamma_{k-1,l} & \Gamma_{k,l-1} \end{bmatrix},$$

причём $\Gamma_{k,0} \sim 1,\ \Gamma_{0,l} \sim -1.$

За шаг до окончания партии (в случае её максимальной продолжительности), будем иметь такую игру-компоненту

$$\Gamma_{11} \sim \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

Её значение, то есть цена игры-компоненты, есть $val\,\Gamma_{11} = v_{11} = 0.$

За два шага до окончания игры получим одну из игр

$$\Gamma_{12} \sim \begin{bmatrix} \Gamma_{11} & -1 \\ -1 & \Gamma_{11} \end{bmatrix} \text{ или } \Gamma_{2,1} \sim \begin{bmatrix} 1 & \Gamma_{11} \\ \Gamma_{11} & 1 \end{bmatrix},$$

где Γ_{11} — такая ситуация (исход) игр Γ_{12} и Γ_{21}, при которой разыгрывается игра-компонента Γ_{11}.

Для определения цены игры возможна замена вида

$$\Gamma_{12}(v_{11}) \sim \begin{bmatrix} v_{11} & -1 \\ -1 & v_{11} \end{bmatrix} \text{ и } \Gamma_{21}(v_{11}) \sim \begin{bmatrix} 1 & v_{11} \\ v_{11} & 1 \end{bmatrix}.$$

И, таким образом, игра-компонента заменяется матричной игрой, для которой находятся

$$v_{12} = val\,\Gamma_{12}(v_{11}) \text{ и } v_{21} = val\,\Gamma_{21}(v_{11}).$$

На третьем шаге от конца игры, когда у игроков, в общей сложности, 4 единицы ресурсов, будет разыгрываться одна из следующих компонент

$$\Gamma_{13} \sim \begin{bmatrix} \Gamma_{12} & -1 \\ -1 & \Gamma_{12} \end{bmatrix},\ \Gamma_{22} \sim \begin{bmatrix} \Gamma_{21} & \Gamma_{12} \\ \Gamma_{12} & \Gamma_{21} \end{bmatrix} \text{ и } \Gamma_{31} \sim \begin{bmatrix} 1 & \Gamma_{21} \\ \Gamma_{21} & 1 \end{bmatrix}.$$

В этих играх также производится приведение их к матричным играм

$$\Gamma_{13}(v_{12}) \sim \begin{bmatrix} v_{12} & -1 \\ -1 & v_{12} \end{bmatrix},\ \Gamma_{22}(v_{12}, v_{21}) \sim \begin{bmatrix} v_{21} & v_{12} \\ v_{12} & v_{21} \end{bmatrix} \text{ и } \Gamma_{31}(v_{21}) \sim \begin{bmatrix} 1 & v_{21} \\ v_{21} & 1 \end{bmatrix}.$$

После чего находятся

$$v_{13} = val\,\Gamma_{13}(v_{12}),\ v_{22} = val\,\Gamma_{22}(v_{12}, v_{21})\ \text{и}\ v_{31} = val\,\Gamma_{31}(v_{21}).$$

Далее, можно решать игры-компоненты с объёмом ресурсов 5, 6, …, *N*. При этом будет возникать игра-компонента вида

$$\Gamma_{r,R-r} \sim \begin{bmatrix} \Gamma_{r,R-r-1} & \Gamma_{r-1,R-r} \\ \Gamma_{r-1,R-r} & \Gamma_{r,R-r-1} \end{bmatrix},,$$

которая, в общем случае, заменяется матричной игрой

$$\Gamma_{r,R-r}(v_{r-1,R-r}, v_{r,R-r-1}) \sim \begin{bmatrix} v_{r,R-r-1} & v_{r-1,R-r} \\ v_{r-1,R-r} & v_{r,R-r-1} \end{bmatrix},\text{где}$$

$v_{r-1,R-r} = val\,\Gamma_{r-1,R-r}(v_{r-1,R-r-1}, v_{r-2,R-r})$, $v_{r,R-r-1} = val\,\Gamma_{r,R-r-1}(v_{r,R-r-2}, v_{r,R-r-1})$, и так далее.

Ещё раз обратим внимание на необходимость вывода рекуррентного соотношения для расчёта, дабы не погрязнуть в рассмотрение многочисленных игр-компонент, число которых может быть весьма велико.

Пример, опять же навеянный детским триллером “Золотой Ключик”.

Буратино доложен отпереть дверь золотым ключиком. Манипуляции с холстом скважиной и ключом занимают время ΔT. Патрульный Дуремар, за время, равное *T*, обходит город в случайном порядке, так как не знает, где появится Буратино. Внезапно появившись во время манипуляций возле двери, он помешает открыть проход в кукольную страну и поймает Буратино.

Задача Буратино – открыть портал в кукольную страну, а Дуремара – изловить строптивца и возмутителя спокойствия кукольного мира.

Пусть отношение $\frac{T}{\Delta T} = N$ — целочисленно, и Буратино может отпереть дверь в моменты времени $t = \Delta T,\ 2\Delta T, ..., k\Delta T, ..., N\Delta T$. Стратегии игроков в игре-компоненте, разыгрываемой в текущий момент времени, суть следующие: для Буратино – отпирать (1) или не отпирать (2), а для Дуремара – появиться (1) или не появиться (2) в районе каморки папы Карло.

Если оба действуют, то Дуремар поймает Буратино

На первом от конца шаге (за шаг до окончания) игры имеем для красного игрока Буратино:

$$\Gamma_1 \sim \begin{bmatrix} -1 & 1 \\ 0 & 0 \end{bmatrix}.$$

Имеем седловую точку в платёжной матрице с координатами (2, 1). То есть, Буратино доложен выжидать, а Дуремар – действовать. В этом случае, Буратино не решит своей задачи, но останется на свободе и может попытать счастье в следующей игре. Цена игры-компоненты составит $val\,\Gamma_1 = v_1 = 0$.

Для игры-компоненты за два шага до окончания имеем

$$\Gamma_2(v_1) \sim \begin{bmatrix} -1 & 1 \\ 1 & v_1 \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix},$$

То есть, действующий, в отсутствие Дуремара, Буратино победит, победу обеспечит и его бездействие при наличии Дуремара: дождавшись, когда Дуремар уйдёт, Буратино беспрепятственно откроет дверцу. Цена игры составит $v_2 = val\,\Gamma_2(v_1) = \frac{1}{3}$, можно определить и оптимальные стратегии.

В общем случае, игра-компонента будет иметь вид

$$\Gamma_N \sim \begin{bmatrix} -1 & 1 \\ 1 & \Gamma_{N-1} \end{bmatrix}.$$

Для платёжной матрицы вида

$$\begin{bmatrix} -1 & 1 \\ 1 & a \end{bmatrix},$$

при $a < 1$, седловой точки в платёжной матрице не имеется, $v(a) < 1$ и составит величину $v(a) = \frac{1+a}{3-a}$, откуда непосредственно последует рекуррентное соотношение $v_N = \frac{v_{N-1}+1}{3-v_{N-1}}$.

В [55], для игр такого типа показано $v_N = \frac{N-1}{N+1}$.

То есть, для второго шага игры цена составит

$$v_{N-1} = \frac{N-2}{N}.$$

Применение последней формулы для начального шага игры даёт

$$\Gamma_N(v_{N-1}) \sim \begin{bmatrix} -1 & 1 \\ 1 & \dfrac{N-2}{N} \end{bmatrix},$$

откуда легко находятся оптимальные смешанные стратегии игроков:

$$P_N^* = \left(\frac{1}{N+1}, \frac{N}{N+1} \right), \qquad Q_N^* = \left(\frac{1}{N+1}, \frac{N}{N+1} \right), \text{ при } N \geq 2.$$

Пусть T = 1 час, ΔT = 12 мин, тогда $N = 5$.

В начале 1-го кванта времени, в момент ΔT:

$$N=5;\ P_5^* = \left(\frac{1}{6}, \frac{5}{6} \right), Q_5^* = \left(\frac{1}{6}, \frac{5}{6} \right), v_5 = \frac{2}{3}.$$

В последующий, 2-й квант, в момент $2\Delta T$:

$$N=4;\ P_4^* = \left(\frac{1}{5}, \frac{4}{5} \right), Q_4^* = \left(\frac{1}{5}, \frac{4}{5} \right), v_4 = \frac{3}{4}.$$

И так далее…

$$N=2;\ P_2^* = \left(\frac{1}{3}, \frac{2}{3} \right), Q_2^* = \left(\frac{1}{3}, \frac{2}{3} \right), v_2 = \frac{1}{3}.$$

В самом конце интервала: $N=1;\ P_1^* = (0,1), Q_1^* = (1,0), v_1 = 0.$

Мы видим, что в самом начале игры Буратино должен пережидать, надеясь, что Дуремар покинет район каморки, и можно будет без помех отпереть дверь.

4.8.2. Стохастические игры

Стохастическая игра является ***многокомпонентной многошаговой*** игрой. В отличие от детерминированных игр, на каждом шаге стохастической игры разыгрывается игра компонента, которая будет использоваться на следующем шаге и определяются выигрыши игроков.

Принципиально, стохастическая игра может возвращаться в предыдущую позицию, и, теоретически, партия будет тянуться до бесконечности. Но, так как заданные априори ***вероятности*** окончания игры ***не равны*** нулю (***положительны***), и ***число позиций*** игры ***конечно***, то вероятность бесконечной продолжительности партии равна нулю. Таким образом, стохастическая игра завершится после неопределенного (стохастического, случайного) числа шагов.

Формально, стохастическая игра есть набор из P позиций, каждая игровая позиция Γ_k задаётся матрицей $m_k \times n_k$, $k = 1, P$ элементы которой имеют вид

$$h_{ij}^k = a_{ij}^k + \sum_{\mu=1}^{P} q_{ij}^{k\mu}\,\Gamma_\mu + q_{ij}^{ko}\,\Gamma_o, i = 1,\overline{m_k}, j = 1,\overline{n_k}, \qquad (4.23)$$

где a_{ij}^k — выигрыш игрока I, если он выберет стратегию i, а II-й игрок – стратегию j; $q_{ij}^{k\mu} \geq 0$ — вероятность перехода из позиции k в игровую позицию μ при выборе игроками стратегий i и j соответственно; $q_{ij}^{ko} = 1 - \sum_{\mu=1}^{P} q_{ij}^{k\mu} > 0$ — вероятность окончания партии в k-й игровой позиции. Иногда, запись компоненты $q_{ij}^{k_o}\,\Gamma_o$ в элементах платёжной матрицы опускают из очевидных соображений.

Пример описания двухкомпонентной стохастической игры приводится ниже. Партия между этими игроками закончится, когда испытание даст исход Γ_o.

Игра-компонента Γ_1.

	1	2
1	$6 + (0{,}3\,\Gamma_0 + 0{,}5\,\Gamma_1 + 0{,}2\,\Gamma_2)$	$0 + (0{,}1\,\Gamma_0 + 0{,}2\,\Gamma_1 + 0{,}7\,\Gamma_2)$
2	$-3 + (0{,}2\,\Gamma_0 + 0{,}3\,\Gamma_1 + 0{,}5\,\Gamma_2)$	$3 + (0{,}4\,\Gamma_0 + 0{,}4\,\Gamma_1 + 0{,}2\,\Gamma_2)$

Игра-компонента Γ_2.

	1	2
1	$2 + (0{,}1\ \Gamma_0 + 0{,}4\ \Gamma_1 + 0{,}5\ \Gamma_2)$	$0 + (0{,}5\ \Gamma_0 + 0{,}3\ \Gamma_1 + 0{,}2\ \Gamma_2)$
2	$0 + (0{,}8\ \Gamma_0 + 0{,}1\ \Gamma_1 + 0{,}1\ \Gamma_2)$	$3 + (0{,}3\ \Gamma_0 + 0{,}4\ \Gamma_1 + 0{,}3\ \Gamma_2)$

Таким образом, партия стохастической игры будет переходить от одной позиции к другой, согласно вероятностям перехода. При этом выбор стратегии оказывает влияние не только на выигрыш при текущем шаге, но и на выигрыши всех последующих шагов.

В стохастической игре стратегию игрока I ***определяют для каждого игрового элемента*** (игры-компоненты) $k = 1, 2, \ldots, P$ и ***для всех шагов*** t как набор m_k-мерных векторов $X^k(t)$, удовлетворяющих условию

$$\sum_{i=1}^{m_k} x_i^k(t) = 1.$$

Стратегии же игрока II определяются аналогичным набором векторов $Y^k(t)$ размерностью n_k.

Для простоты предполагается, что используется одна и та же схема рандомизации при разыгрывании одного и того же элемента. Это означает, что для каждой игры-компоненты Γ_k используется один и тот же набор вероятностей применения чистых стратегий, сколько бы раз этот элемент не разыгрывался.

Такая стратегия называется ***стационарной***.

Если стохастическая игра начинается с компоненты Γ_k, то пара стратегий игроков определяется как в обычной матричной игре математическим ожиданием выигрыша I-го игрока v_k. Так как стохастическая игра может начаться с любой игры-компоненты, то имеем вектор оценок выигрышей по играм $V = \{v_1, v_2, \ldots, v_P\}$.

Таким образом, стохастическая игра ***должна решаться для каждого*** начального условия (т.е. для каждого ***игрового элемента, с которого начнётся игра***).

Метод решения основан на “усечении” игры. Предполагается, что игра продолжается ровно r шагов, а затем заканчивается.

Это допущение равносильно замене разыгрывания на r+1-м шаге игр-компонент $\left\{\Gamma_\mu^{r+1}\right\}$ их значениями $\left\{v_\mu^O\right\}$, что называется усечением игры на r-м шаге посредством выигрышей $\left\{v_\mu^O\right\}$. Когда r достаточно велико, то игра не отличается от первоначальной, а величины $\left\{v_\mu^O\right\}$ не должны оказывать сильное влияние на значение усечённой игры.

При этом последовательность $\left\{v_\mu^O\right\}$, $\left\{v_\mu^1\right\}$, …, $\left\{v_\mu^r\right\}$ сойдётся к пределу $V = \{v_1, v_2, \ldots, v_P\}$.

В общем случае, применяется следующий алгоритм пересчёта стохастической игры в матричную игру

$$\begin{cases} v_0 = (0,0,...,0); & (4.24) \\ b_{ij}^{kr} = a_{ij}^{k} + \sum_{\mu=1}^{P} q_{ij}^{k\mu} v_{\mu}^{r}, r = 0,1,..., i = 1,\overline{m_k}, j = 1,\overline{n_k}; & (4.25) \\ v_k^{r+1} = \mathrm{val}\, B_k^r = \mathrm{val}\left\| b_{ij}^{kr} \right\|, i = 1,\overline{m_k}, j = 1,\overline{n_k}. & (4.26) \end{cases}$$

Оптимальными стационарными стратегиями игроков в стохастической игре являются стратегии серии матричных игр $\{B_k(v)\}$, в которых элементы матриц $m_k \times n_k$ игр-компонент определяются формулой

$$b_{ij}^{k} = a_{ij} + \sum_{\mu=1}^{P} q_{ij}^{k\mu} v_{\mu}, i = 1,\overline{m_k}, j = 1,\overline{n_k},$$

как это следует из (4.25). Г. Оуэн показал [55], что вероятность продолжения игры, более чем r шагов, не превосходит величины S^r, какие бы стратегии игроками не применялись, где

$$S = \max_{\{i,j,k\}} \left\{ \sum_{\mu=1}^{P} q_{ij}^{k\mu} \right\}. \quad (4.27)$$

Руководствуясь этой оценкой можно определить компромиссное значение r, при котором усечённая игра сходится, в процессе решения, к оптимальным стратегиям и цене стохастической игры. Выражение (4.27) означает выбор максимальной возможной вероятности продолжения игры по всем играм-компонентам.

Пример [55, 64]. Пусть стохастическая игра задана своими играми-компонентами вида

$$\Gamma_1 \sim \begin{bmatrix} 2+0{,}5\Gamma_3 & -1 \\ -1 & 2+0{,}5\Gamma_2 \end{bmatrix}, \quad \Gamma_2 \sim \begin{bmatrix} 1 & -1+0{,}5\Gamma_1 \\ -1+0{,}5\Gamma_1 & 1 \end{bmatrix} \text{ и}$$

$$\Gamma_3 \sim \begin{bmatrix} 1 & -1+0{,}5\Gamma_2 \\ -1+0{,}5\Gamma_2 & 1 \end{bmatrix}.$$

В ходе анализа игр компонент выяснено, что S = 0,5. Пусть r = 5, тогда

$$S^r = \frac{1}{32} = 0{,}03125,$$

то есть, порядка трёх процентов, так называемой статистической погрешности. Поэтому мы вполне можем воспользоваться усечённой до 5-и шагов игрой.

Согласно (4.24), $v = (0, 0, 0)$, а согласно (4.25) получаем игры

$$B_1^0 \sim \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix},\ B_2^0 \sim \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \text{ и } B_3^0 \sim \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

В соответствии с (4.26), определяем цены игр:

$$v_1^1 = \operatorname{val} B_1^0 = \frac{1}{5},\ v_2^1 = \operatorname{val} B_2^0 = 0 \text{ и } v_3^1 = \operatorname{val} B_3^0 = 0.$$

Следовательно, для данного шага усечения имеем $v^1 = (0{,}2;\ 0;\ 0)$.
Затем, используя компоненты вектора v^1, находим v^2 и так далее…
Имеем последовательность векторов

$$\begin{cases} v^1 = (0{,}2 \quad 0 \quad 0), \\ v^2 = (0{,}2 \quad 0{,}05 \quad 0), \\ v^3 = (0{,}21 \quad 0{,}05 \quad 0), \\ v^4 = (0.21 \quad 0{,}05 \quad 0), \\ v^5 = (0{,}21 \quad 0{,}05 \quad 0). \end{cases}$$

Полагая последний вектор v^5 за вектор значения неусечённой стохастической игры, имеем

$$B_1^5 \sim \begin{bmatrix} 2 & -1 \\ -1 & 1{,}25 \end{bmatrix},\ B_2^5 \sim \begin{bmatrix} 1 & -0{,}88 \\ -0{,}88 & 1 \end{bmatrix} \text{ и } B_3^5 \sim \begin{bmatrix} 1 & -0{,}975 \\ -0{,}975 & 1 \end{bmatrix}.$$

По последним полученным игровым матрицам рассчитываем оптимальные стационарные стратегии игроков по каждой из игр-компонент:

$x_1 = (0{,}43;\ 0{,}57)$; $y_1 = (0{,}43;\ 0{,}57)$;
$x_2 = (0{,}5;\ 0{,}5)$; $y_2 = (0{,}5;\ 0{,}5)$;
$x_3 = (0{,}5;\ 0{,}5)$; $y_3 = (0{,}5;\ 0{,}5)$.

4.8.3. Рекурсивные игры

Рекурсивная игра отличается от стохастической тем, что вероятность её бесконечного продолжения отлична от нуля, следовательно, партия такой игры может продолжаться бесконечно долго.

В этом случае выигрыш обозначается как h_∞ и полагается равным нулю.

Формально, рекурсивная игра есть набор из P позиций, каждая игровая позиция Γ_k задаётся матрицей $m_k \times n_k$, элементы которой записываются так

$$h_{ij}^k = a_{ij}^{ko} \cdot q_{ij}^{ko} + \sum_{\mu=1}^{P} q_{ij}^{k\mu} \Gamma_\mu, i = \overline{1, m_k}, j = \overline{1, n_k}, \qquad (4.28)$$

где $\sum_{\mu=0}^{P} q_{ij}^{k\mu} = 1, q_{ij}^{k\mu} \geq 0, \mu = \overline{0, P}$. В отличие от детерминированной игры, здесь допускаются ситуации, в которых $\exists(i, j, k) q_{ij}^{ko} = 0$, то есть игра завершиться не может и продолжается. Элемент h_{ij}^k означает, что при выборе пары чистых стратегий (i, j) в игровой позиции k вероятность окончания игры составит q_{ij}^{ko}, при этом игрок I получит выигрыш a_{ij}^{ko}, а вероятность того, что в следующей позиции будет разыграна игра-компонента Γ_μ, равна $q_{ij}^{ko} \geq 0$.

Пример рекурсивной однокомпонентной игры.

$\Gamma \sim$		1	2
	1	$0{,}7\,\Gamma + 3 \cdot 0{,}3$	2
	2	3	Γ

Так как игра протекает теоретически бесконечно, то применение алгоритма, соответствующего формулам (4.24) – (4.26) не возможно, ибо не существует такого числа партий r, начиная с которого было бы проделано корректное усечение игры. В рекурсивных играх последовательность оценок v^r не обязательно сходится к истинному значению вектора цен игр-компонент v.

Проиллюстрируем последнее суждение примером. Дана однокомпонентная рекурсивная игра

$$\Gamma \sim \begin{pmatrix} \Gamma & 1 \\ 1 & 0 \end{pmatrix},$$

для которой, в общем случае, можно записать

$$v^{r+1} \sim \mathrm{val}\begin{pmatrix} v^r & 1 \\ 1 & 0 \end{pmatrix}.$$

Как следствие решения игры 2 × 2 имеем

$$v^{r+1} = \frac{1}{2 - v^r}.$$

Полагая, что игроки играют долго и безрезультатно ($v_0 = 0$), получим формулу $v = \frac{1}{1+r}$, откуда последует, что $\lim\limits_{r\to\infty} \frac{1}{1+r} = 1$.

Окончательно получим платёжную матрицу для расчётов $B \sim \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$, которой имеется пара седловых точек (1,1) и (1,2).

Обратимся теперь к исходной игре. Если в ней синий игрок будет выбирать свою первую стратегию в качестве оптимальной, то ***игра не завершится*** никогда, то есть выигрыш красного игрока, в этом случае, составит $h_\infty = 0$.

Поэтому, ***оптимальных стратегий может не существовать***, а существуют ***ε-оптимальные***.

Если, в рассматриваемой игре, игрок I будет осуществлять смешанную стратегию вида $(1 - \varepsilon, \varepsilon)$, где $0 < \varepsilon < 1$, то партия закончится гарантированно. Математическое ожидание выигрыша составит $v = 1 - \varepsilon$, не зависимо от действий игрока II.

Если же II-й игрок применит 2-ю чистую стратегию, то игра заканчивается, а математическое ожидание выигрыша I-го игрока всё равно составит $v = (1 - \varepsilon) \cdot 1 + \varepsilon \cdot 0$.

Поэтому игрок I может гарантировать математическое ожидание выигрыша, близкое к единице, но не достигающее его. При значениях ε, стремящихся к нулю, математическое ожидание продолжительности игры возрастет, а математическое ожидание выигрыша претерпевает разрыв при переходе к предельным стратегиям с $v = (1 - \varepsilon)$ на $v = \varepsilon$.

Пример. По колумбийским чащам следуют два наркобарона: дон Педро и дон Базилио со своими телохранителями (три телохранителя у Педро, два – у его оппонента). Оба ненавидят друг друга, стремятся уничтожить и знают, что находятся поблизости один от одного, и, поэтому, решаются на нападение.

При этом существует дилемма: сколько человек послать в нападение, а сколько оставить при теле хозяина. Обозначим стратегии игроков как

систему чисел (i, j), где i – число телохранителей, участвующих в нападении, а j – число телохранителей, охраняющих хозяина.

Установим следующие правила игры. Если число нападающих больше, чем, охраняющих, то босс уничтожается, если наоборот, то нападение прекращается и организовывается снова (возобновляется игровой цикл).

Выигрыш дона Педро составит + 1, если он уничтожит дона Базилио, и – 1 в противном случае. В нормальной форме игра представима игрой-компонентой

$\Gamma \sim$

	{0, 2}	{1, 1}	{2, 0}
{0, 3}	Г	Г	Г
{1, 2)	Г	Г	1
{2, 1}	Г	1	— 1
{3, 0}	1	— 1	1

В данной случае имеем δ-оптимальную стратегию игрока I (дона Педро): $(0, 1 - \delta - \delta^2, \delta, \delta^2)$, получающуюся при рассмотрении предельной игры вида $\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & -1 & -1 \end{pmatrix}$.

Чем меньше значение δ, тем больше вероятность победы и тем больше математическое ожидание продолжительности игры. При значении δ, равном нулю, игра будет бесконечной продолжительности.

4.9. Бесконечные игры

На практике возникают ситуации, в которых каждая из сторон выбирает некоторый ***непрерывный параметр***. Это может быть дистанция до цели или момент открытия огня, соотношение сил нападения и поиска, отношение сигнал-шум и т.д.

Как правило, такой параметр имеет бесчисленное число значений (в силу его непрерывности) или такое большое конечное число значений, что его удобно рассматривать как бесконечное.

Теоретическими моделями таких игровых ситуаций являются ***бесконечные игры***, то есть такие игры, в которых чистые стратегии представляют собой выборы тех или иных чисел из бесконечных множеств A и B.

Практически данные множества ограничены, поэтому представляют собой некоторые ***замкнутые интервалы*** либо ***замкнутые подмножества*** конечномерных эвклидовых пространств.

При этом стратегии игроков удобно отождествлять с отрезками единичной длины к которым, в принципе, может быть приведён интервал между максимальным и минимальным числами. Таким образом, пара чистых стратегий соответствует точке единичного квадрата.

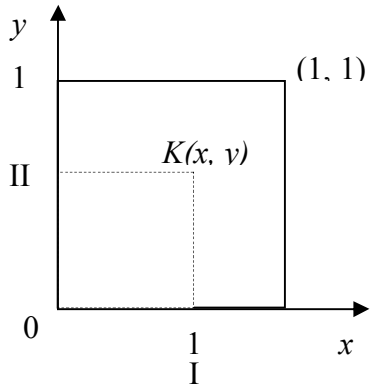


Рисунок 4.11 – Игровой квадрат

Термин "***бесконечная игра***" иногда заменяется термином "***непрерывная игра***", калькой от латинского слова ***continuum***. В качестве стратегий красного игрока принимается число $0 \le x \le 1$, а в качестве стратегий синего – $0 \le y \le 1$. Пара выбранных стратегий *(x, y)* определяет ситуацию, в которой игрок получит выигрыш равный *K(x, y)*, а игрок – выигрыш – *K (x, y)* (игра же с нулевой суммой!).

Так как множество пар точек *(x, y)* заполняет квадрат, то игра получила название "***игра на квадрате***" (смотри рисунок 4.11).

Функция *K (x, y)*, определяемая на единичном квадрате $[0, 1] \times [0, 1]$ и ставящая в соответствие каждой игровой ситуации выигрыш, который получает I-й игрок, называется ***функцией выигрыша*** или ***ядром*** (***игры***).

Для игр на квадрате ***остаются справедливыми*** все те зависимости, которые нами были рассмотрены ранее, для парной игры с нулевой суммой, касательно цены игры и седловой точки с точностью до обозначений.

$$v_1 = \max_x \min_y K(x, y); \qquad v_2 = \min_y \max_x K(x, y); \qquad v_1 \le v_2 .$$

$$K(x, y_0) \le K(x_0, y_0) \le K(x_0, y) .$$

Если находятся пары чисел, удовлетворяющих двойному неравенству седловой точки, то говорят, что игра имеет решение в ***чистых стратегиях***, а ядро имеет седловую точку, в общем же случае $v_1 \le v \le v_2$.

Смешанная стратегия, в соответствии с определением, представляет собой функцию распределения вероятностей, определяемую на интервале [0, 1] и обладающую известными свойствами:

1. $F(0) = 0$;
2. $F(1) = 1$;
3. $x > x_1 \Rightarrow F(x) \geq F(x_1)$;
4. $x_n \to x,\ x < 1,\ x_n < x \Rightarrow F(x_n) \to F(x)$.

Вероятность выбора числа из интервала $[x_1, x_2]$ есть $P(x_1, x_2) = F(x_2) - F(x_1)$, а функция распределения определяется интегралом

$$F(x) = \int_0^x f(x)dx,$$

в котором $f(x)$ есть плотность распределения вероятностей, называемая тако же дифференциальной функцией распределения или дифференциальным законом распределения. В теории игр часто используют специальную функцию распределения вида

$$F(x) = \alpha_1 I_{x_1}(x) + \alpha_2 I_{x_2}(x) + \ldots + \alpha_n I_{x_n}(x), \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i = 1, 0 \leq x_i \leq x_{i+1} \leq 1, \quad (4.29)$$

а $I_a(x)$ одноступенчатая функция

$$I_a(x) = \begin{cases} 1, & x \geq a, \\ 0, & x < a. \end{cases} \quad (4.30)$$

Функция (4.29) называется ступенчатой функцией с n степенями.

Для синего игрока определятся функция $G(y)$, аналогичная (4.29) для красного игрока.

Пусть красный игрок использует смешанную стратегию F, а синий – чистую стратегию y. Тогда математическое ожидание выигрыша красного игрока (если существует)

$$E(F, y) = \int_0^1 K(x, y)dF(x).$$

Аналогично для синего игрока

$$E(x, G) = \int_0^1 K(x, y)dG(y).$$

При использовании игроками своих оптимальных смешанных стратегий, математическое ожидание выигрыша, если оно существует, равно

$$E(F,G)=\int_0^1\int_0^1 K(x,y)dF(x)dG(y).$$

Поэтому, как и в матричных играх, могут существовать максимин, минимакс

$$\max_F \min_G E(F,G) \le \min_G \max_F E(F,G)$$

и оптимальные стратегии игроков F^* и G^*, такие, что $E(F^*, G) \ge v$, $E(F, G^*) \le v$.

Воспользуемся примером по открыванию двери Буратино, изложенному в разделе 4.8.1 в качестве детерминированной игры, при гипотезе о дискретности времени наступления события. На этот раз будем исходить из того, что время непрерывно:

- T – время, в течение которого Буратино должен открыть дверь,
- T – время, в течение которого может появиться Дуремар,
- ΔT – время ($\Delta T < T$), за которое Буратино должен открыть дверь.

Пусть $0 \le t_{\text{I}} \le T - \Delta T$ есть время начало действия Буратино, а $0 \le t_{\text{II}} \le T$ момент появления Дуремара.

Введём обозначения: $t=\frac{\Delta T}{T}; x=\frac{t_I}{T}; y=\frac{t_{II}}{T}$, после чего будем иметь игру на квадрате, в которой стратегии игроков будут представимы числами из интервала: [0, 1 – t] для первого игрока и [0, 1] для второго, а функция выигрыша представит собой разрывную функцию

$$K(x,y)=\begin{cases}1, & y \le x \le 1-t \equiv y \le x, x \le 1-t;\\ 1, & x+t \le y \le 1 \equiv x+t \le y, y \le 1;\\ 0, & \text{в остальных случаях,}\end{cases}$$

область единичных значений, которые показаны на рисунке 4.12.

Нижняя область рисунка соответствует случаю, когда Дуремар не появился до момента времени , а нижняя – когда он появился и ушёл.

Для такой игры Дрешером [30] найдено следующее оптимальное решение.

Оптимальная стратегия красного игрока есть

$$F^*(x) = \frac{1}{[t^{-1}]} \cdot \sum_{j=0}^{[t^{-1}]-1} I_{j\times t}(x).$$

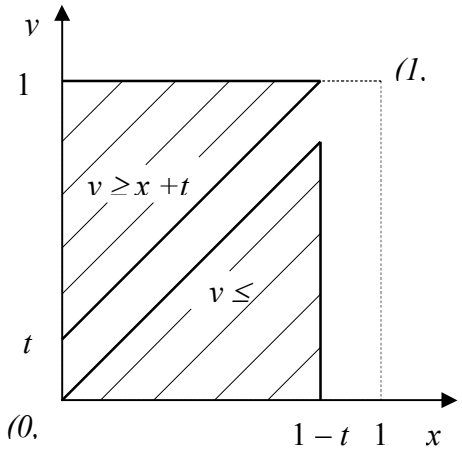


Рисунок 4.12 – Области ядра игры

Для синего игрока имеем

$$G^*(y) = \begin{cases} y, & t^{-1} - \text{целое число}, \\ \dfrac{1}{[t^{-1}]} \cdot \sum\limits_{j=0}^{[t^{-1}]} I_{\frac{j}{[t^{-1}]+1}}(y), & t^{-1} - \text{не целое число}. \end{cases}$$

При этом цена игры составит

$$v = 1 - \frac{1}{[t^{-1}]},$$

где запись […] обозначает не превосходящее целое, то есть, округление по недостатку.

Пусть, например, T = 6 часов, а ΔT = 2,5 часа. Находим:

$$t^{-1} = \frac{6}{2{,}5} = \frac{12}{5} \Rightarrow [t^{-1}] = 2.$$

Имеем: оптимальную стратегию красного игрока

$$F^*(x) = 0{,}5I_0(x) + 0{,}5I_{\frac{5}{12}}(x),$$

синего игрока

$$G^*(y) = 0{,}5I_{\frac{1}{3}}(y) + 0{,}5I_{\frac{2}{3}}(y)$$

и цену игры $v = 0{,}5$. Получается, что Буратино должен начать манипуляции по отпиранию двери в начальный момент времени и через 2,5 часа равновероятно, а Дуремар – появиться в районе каморки в моменты 2 и 4 часа.

Пусть, как в примере раздела 4.8.1, теперь $T = 1$ час, $\Delta T = 12$ мин, тогда $t^{-1} = 5$ – целое число. Цена игры составит $v = 0{,}8$.

Стратегия Буратино есть $F^*(x) = \frac{1}{5}\left[I_0(x) + I_{\frac{1}{5}}(x) + I_{\frac{2}{5}}(x) + I_{\frac{3}{5}}(x) + I_{\frac{4}{5}}(x) \right]$,

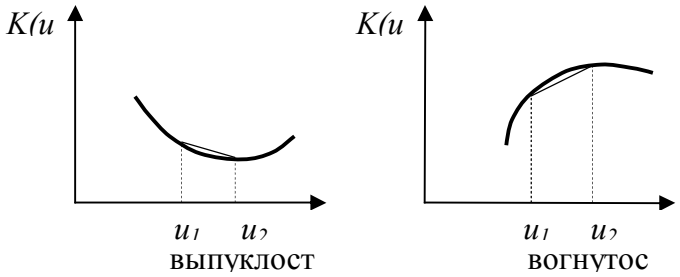
Дуремар же имеет равномерное распределение по всему временному интервалу $G^*(y) = y$, $0 \le y \le 1$.

Эта игра имеет ядро с разрывом. Для разрывного ядра игра, в ряде случаев, может не иметь решения, а игроки – своих оптимальных стратегий. Игры с ***разрывным ядром***, в которых выбирается ***момент времени***, называются ***дуэлями***.

Если ядро непрерывно, то решение может быть найдено всегда, но универсальных методов решения не существует.

4.9.1. Выпуклые и вогнутые игры [56, 64]

В случае, когда ядро игры (функция выигрыша) *K(x, y)* является нелинейным, то могут иметь место ***выпуклость*** и ***вогнутость*** по переменным *x* и *y*. Определение этих понятий аналогично определению, данному нами при рассмотрении НП-задач.



Функция *K(u)* называется ***выпуклой*** по *u* на интервале [0, 1] если

$$K[\lambda u_1 + (1 - \lambda)u_2] \le \lambda K(u_1) + (1 - \lambda)K(u_2),\ u_1,\ u_2 \in [0, 1],\ 0 \le \lambda \le 1,$$

и ***вогнуто***й по *u* если

$$K[\lambda u_1 + (1-\lambda)u_2] \geq \lambda K(u_1) + (1-\lambda)K(u_2),\ u_1,\ u_2 \in [0,\ 1],\ 0 \leq \lambda \leq 1.$$

Дважды дифференцируемая функция выигрыша $K(x,\ y)$ является выпуклой по y для любого x, если

$$\frac{\partial^2 K(x,y)}{\partial y^2} \geq 0.$$

Игра называется ***выпуклой***. В этом случае, игрок II имеет ***чистую*** оптимальную стратегию в случае ***строгой выпуклости***, и целое ***множество стратегий***, если имеет место ***нестрогая*** выпуклость.

Если

$$\frac{\partial^2 K(x,y)}{\partial x^2} \leq 0,$$

то игра вогнута по x для любого y. В этом случае, игрок I имеет ***чистую*** оптимальную стратегию в случае ***строгой вогнутости***, и целое ***множество стратегий***, если имеет место ***нестрогая*** вогнутость.

Чистая стратегия игрока задаются в виде одноступенчатых функций вида (4.30)

$$I_{y_1}(y) = \begin{cases} 1, & y \geq y_1, \\ 0, & y < y_1 \end{cases} \text{ и } I_{x_1}(x) = \begin{cases} 1, & x \geq x_1, \\ 0, & x < x_1 \end{cases},$$

в которых y_1 и x_1 определяются из уравнений:

$$v = \max_x K(x, y_1) \text{ и } v = \min_y K(x_1, y),$$

соответственно, а цена игры есть

$$v = \min_y \max_x K(x,y) = \max_x K(x, y^*) \text{ и } v = \max_x \min_y K(x,y) = \min_y K(x^*, y).$$

Если игра определяется функцией $K(x,\ y)$, которая вогнута по x при любом y и выпукла по y при любом x, то игра называется ***вогнуто-выпуклой***, имеет седловую точку, а игроки – по паре чистых оптимальных стратегий $(x_0,\ y_0)$.

Существует ряд определений и теорем.

Определение [64]. Чистая стратегия x игрока I называется существенной, если для некоторой оптимальной стратегии y^* игрока II выполняется условие $K(x, y^*) = v$.

Теорема 1. Пусть Г – выпуклая игра с функцией выигрыша K, дифференцируемая по y при любом x; y^* – чистая оптимальная стратегия игрока II в ней, а v – её значение.

Тогда

1) если $y^* = 1$, то среди оптимальных стратегий игрока I имеется чистая стратегия x', которая является существенной и для которой $K'_y(x',1) \le 0$, где $K'_y(x,y) = \dfrac{\partial K(x,y)}{\partial y}$ (предполагается, что производная $K'_y(x,y)$ существует для всех значений x);

2) если $y^* = 0$, то среди оптимальных стратегий игрока I имеется чистая стратегия x'', которая является существенной и для которой $K'_y(x'',0) \ge 0$.

3) если $0 < y^* < 1$, то среди оптимальных стратегий игрока I найдётся такая, которая является смесью других существенных стратегий x' и x''. Для этих стратегий $K'_y(x', y^*) \le 0$ и $K'_y(x'', y^*) \ge 0$.

При этом стратегии x' и x'' выбираются с вероятностями α и $1 - \alpha$, где α – есть решение уравнения

$$\alpha K'_y(x', y^*) + (1-\alpha) K'_y(x'', y^*) = 0,$$

то есть, стратегии x' и x'' определяются функцией распределения

$$F^*(x) = \alpha \cdot I_{x'}(x) + (1-\alpha) \cdot I_{x''}(x), \qquad (4.31)$$

где $I_{x'}(x)$ и $I_{x''}(x)$ одноступенчатые функции распределения вида (4.30).

Теорема 2. Пусть Г – вогнутая игра с функцией выигрыша K, дифференцируемая по x при любом y; x^* – чистая оптимальная стратегия игрока I в ней, а v – её значение.

Тогда

1) если $x^* = 1$, то среди оптимальных стратегий игрока II имеется чистая стратегия y', которая является существенной и для которой $K'_x(1, y') \ge 0$, где $K'_x(x,y) = \dfrac{\partial K(x,y)}{\partial x}$ (предполагается, что производная $K'_x(x,y)$ существует для всех значений y);

2) если $x^* = 0$, то среди оптимальных стратегий игрока II имеется чистая стратегия y'', которая является существенной и для которой $K'_x(0, y'') \le 0$;

3) если $0 < x^* < 1$, то среди оптимальных стратегий игрока II найдётся такая, которая является смесью других существенных стратегий y' и y''. Для этих стратегий $K'_x(x^*, y') \ge 0$ и $K'_x(x^*, y'') \le 0$.

При этом стратегии y' и y'' выбираются с вероятностями β и $1 - \beta$, где β – есть решение уравнения

$$\beta K'_x(x^*, y') + (1 - \beta) K'_x(x^*, y'') = 0,$$

то есть, стратегии y' и y'' определяются функцией распределения

$$G^*(y) = \beta \cdot I_{y'}(y) + (1 - \beta) \cdot I_{y''}(y), \qquad (4.32)$$

где $I_{y'}(y)$ и $I_{y''}(y)$ одноступенчатые функции распределения вида (4.30).

Реализация стратегий (4.31) и (4.32) заключается в генерации случайного числа ξ из интервала (0, 1) и сравнение его с пороговым значением α или β (смотри рисунок 4.13).

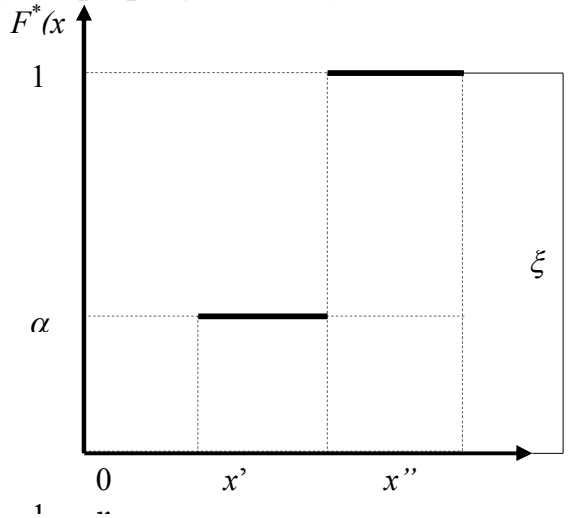


Рисунок 4.13 – Реализация стратегии $F^*(x)$

Если $\xi < \alpha$, то $x = x'$, в противном случае, если $\xi \ge \alpha$, то $x = x''$. Аналогичная схема применяется для смеси стратегий синего игрока $G^*(y)$ (4.32).

Пример 1. Пусть ядро игры имеет вид

$$K(x, y) = -x^2 + 2y^2 + 5xy - x - 2y.$$

Вторые производные ядра

$$\frac{\partial^2 K(x,y)}{\partial x^2} = -2 < 0 \text{ и } \frac{\partial^2 K(x,y)}{\partial y^2} = 4 > 0.$$

Следовательно, данная игра вогнуто-выпуклая, при такой игре существует седловая точка (x_0, y_0). Её будем искать по алгоритму, предложенному в [55]. Необходимо решить систему вида

$$\left.\begin{aligned} x_0 &= \psi(y_0), \\ y_0 &= \varphi(x_0) \end{aligned}\right\},$$

где $\psi(y)$ – такое значение x, которое максимизирует $K(x, y)$, а $\varphi(x)$ – такое значение y, которое минимизирует $K(x, y)$, то есть

$$K[\psi(y), y] = \max_x K(x,y),$$
$$K[x, \varphi(x)] = \min_y K(x,y).$$

Найдём $\frac{\partial K(x,y)}{\partial x} = -2x + 5y - 1 \Rightarrow x = \frac{5y-1}{2}$. Так как значение $y < \frac{1}{5}$ делает $x < 0$, что противоречит исходным посылкам, то функция будет ступенчатой:

$$\psi(y) = \begin{cases} 0, & y \le \frac{1}{5}, \\ \frac{5y-1}{2}, & y > \frac{1}{5}. \end{cases}$$

Аналогично имеем $\frac{\partial K(x,y)}{\partial y} = 4y + 5x - 2 \Rightarrow y = \frac{2-5x}{2}$ и функцию $\varphi(x)$:

$$\varphi(x) = \begin{cases} \frac{2-5x}{2}, & x < \frac{2}{5}, \\ 0, & x \ge \frac{2}{5}. \end{cases}$$

Получаем систему уравнений для определения координат седловой точки и оптимальных стратегий:

$$\begin{cases} x_0 = \frac{5y-1}{2}, \\ y_0 = \frac{2-5x}{2}, \end{cases} \Rightarrow x_0 = \frac{2}{11};\ y_0 = \frac{3}{11}.$$

Пример 2.[64] Пусть игрок II выполняет стрельбу торпедой по игроку I. При обнаружении торпеды, игрок I уклоняется ходом: от неподвижного (Stop) до полного (Fool Speed). В свою очередь, игрок II выбирает упреждение, исходя из предположительной гипотезы о характере манёвра уклонения цели. Пусть ядро функции игры есть $K(x, y) = (x - y)^2$.

Полагая $x = 0$ для Stop, и $x = 1$ для Fool Speed, а в качестве y нормированную поправку на движение цели, придём к игре на квадрате.

Анализ свойств ядра показывает, что игра выпукла для обоих игроков

$$\frac{\partial^2 K(x,y)}{\partial x^2} = \frac{\partial^2 K(x,y)}{\partial y^2} = 2 > 0\,.$$

Поэтому игрок II имеет чистую оптимальную стратегию, а I – смешанную стратегию $F^*(x)$ при $x' = 0$ и $x'' = 1$, такую что

$$v = \min_y \max_x (x - y)^2 = \max_x (x - y^*)^2\,.$$

Для фиксированного значения y максимум v будет достигаться на концах интервала при x =0 или $x = 1$ и будет равен

$$\max\left\{y^2, (1 - y^2)\right\}.$$

Указанное выше соображение иллюстрируется на рисунке 4.13, а. Решим уравнение

$$y^2 = (1 - y)^2 \Rightarrow y^* = 0{,}5;\ v^* = 0{,}25.$$

Вычислим, в соответствии с теоремой,

$$K'_y(x', y^*) = \left.\frac{\partial K(x,y)}{\partial y}\right|_{y^* = \frac{1}{2}; x' = 0} = 1, K'_y(x'', y^*) = \left.\frac{\partial K(x,y)}{\partial y}\right|_{y^* = \frac{1}{2}; x'' = 1} = -1.$$

Согласно третьему пункту теоремы 1, имеем уравнение

$$\alpha - (1 - \alpha) = 0 \Rightarrow \alpha = 0{,}5\,,$$

поэтому оптимальная стратегия первого игрока (4.31)

$$F^*(x) = 0{,}5 \cdot I_0(x) + 0{,}5 \cdot I_1(x)\,,$$

а ожидаемый выигрыш первого игрока составит

$$E(F, y) = 0{,}5y^2 + 0{,}5(1 - y)^2 \geq 0{,}25.$$

Это показано на рисунке 4.14, б.

Таким образом, согласно оптимальным стратегиям, атакующий выбирает точку прицеливания посередине от возможного нахождения корабля от неподвижного до максимально уходящего. При этом обеспечивается вероятность уклонения не менее 0,25.

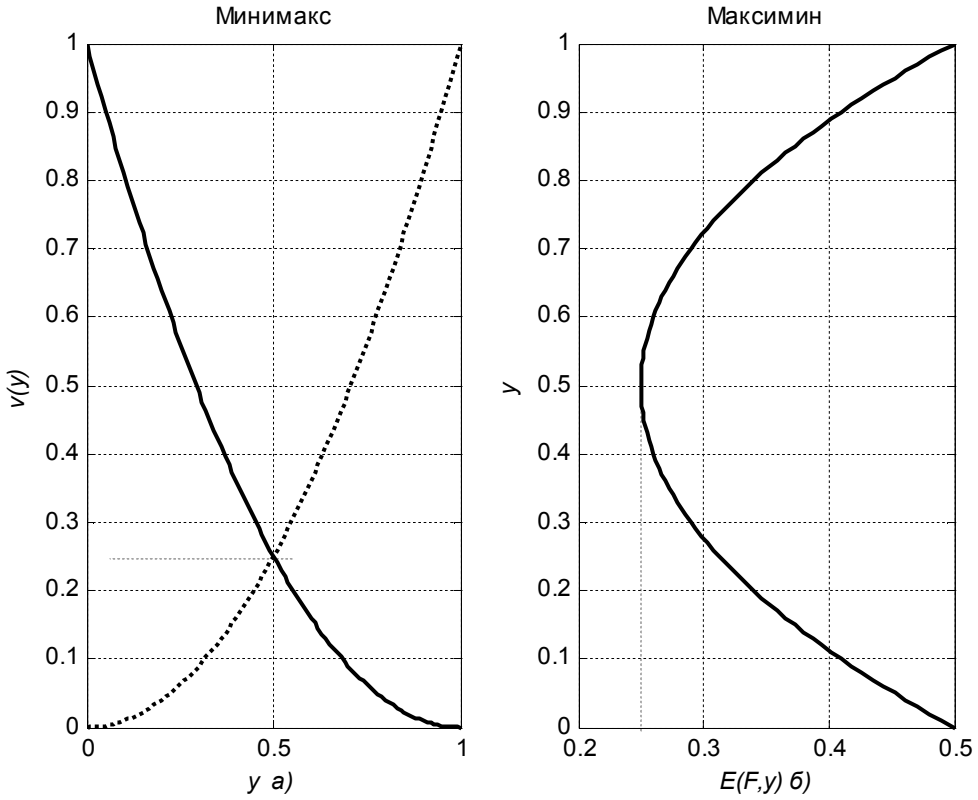


Рисунок 4.14 – Пояснения к исследованию функции ядра

Значения вероятности, равное 0,25, ни какая из противоборствующих сторон односторонними действиями изменить не может: для красного игрока это иллюстрируется рисунком 4.13, б. Если же синий игрок выбирает оптимальную стратегию $y^* = 0{,}5$, то, что бы ни делал красный, его выигрыш не превысит 0,25. Это следует из неравенства

$$(x - 0{,}5)^2 = x \cdot (x - 1) + 0{,}25 \leq 0{,}25.$$

4.9.2. Игры с выбором момента времени действия в условиях полной информации (шумные дуэли)

Существуют ситуации, столкновения интересов, в которых определяется стремление каждой из сторон, участвующих в конфликте, выбрать момент действия, обеспечивающий максимальную вероятность решения своей задачи, при условии упреждения противника.

Областью деятельности, описываемой подобными игровыми моделями, являются: военное дело [69], маркетинг, навигация [16, 19, 38], задачи управления в реальном масштабе времени быстропротекающими процессами и тому подобное.

По очевидному сходству с реальными поединками чести, данные игры называются дуэльными ситуациями или просто – дуэлями.

Характерной особенностью дуэлей является то, что противник стремится, по возможности, задержать свой выстрел (момент действия), так как с течением времени увеличивается вероятность поражения визави. Однако, задержка с применением своего оружия имеет разумные пределы, так как противник может выстрелить первым.

Дуэльная ситуация адекватно моделируется игрой на квадрате, если считать, то стратегии игроков являются числами x и $y \in [0, 1]$, интерпретируемыми как нормированные моменты времени применения оружия каждым игроком.

Функция выигрыша в такой игре есть

$$K(x,y)=\begin{cases} L(x,y), & x<y, \\ \Phi(x,y), & x=y,, \\ N(x,y), & x>y, \end{cases} \qquad (4.33)$$

где

- *L(x, y)* – вероятность поражения I-м игроком II-го игрока, если I-й игрок произвёл выстрел первым;
- *Ф(x)* – вероятность поражения I-м игроком II-го игрока, при условии одновременного применения оружия;
- *N(x, y)* – вероятность поражения I-м игроком II-го игрока, если II-й игрок произвёл выстрел первым.

Важно (!): При заданной функции выигрыша считается, что если II-й игрок предполагает применить оружие в момент времени *y*, то I-й игрок увеличивает свой выигрыш, ожидая сколь возможно, но ***действуя раньше*** II-го (читай красочное описание дуэли в романе в стихах А.С. Пушкина

“Евгений Онегин”). То есть, $L(x, y)$ монотонно возрастает по переменной x для каждого y.

При $x = y$ ядро $K(x, y)$ претерпевает разрыв.

Далее, в случае промаха игрока II, шансы на успех I-го игрока возрастают, таким образом, $N(x, y)$ монотонно возрастает по x для каждого y.

Рассуждая аналогичным образом, относительно II-го игрока, придём к выводу что $L(x, y)$ и $N(x, y)$ монотонно убывают по для каждого , а при $x = y$ ядро имеет разрыв.

Выигрыши по обе стороны разрыва могут существенно отличаться. При игре с полной информацией противник знает о применении другим противником оружия (шум выстрела), откуда следует, что $L(x, y) = l(x)$, а $N(x, y) = n(y)$. Поэтому ядро (4.33) трансформируется и выглядит так

$$K(x,y)=\begin{cases} l(x), & x<y, \\ \varphi(x), & x=y, \\ n(y), & x>y. \end{cases} \qquad (4.34)$$

Если предположить, что в ядре вида (4.34), функции $l(x)$ и $n(y)$ имеют непрерывные производные второго порядка, $\varphi(x)$ произвольная ограниченная функция, а $l(1) > n(1)$ и $l(0) < n(0)$, существует теорема [28].

Теорема [64]. Игра с ядром (4.34) имеет значение $v = l(a_0) = n(a_0)$, а оптимальные стратегии x^* и y^* описываются следующим образом. Если $\varphi(a_0) > v$, то $x^* = x_{a_0}$, а игрок II не имеет достижимой оптимальной стратегии; ε-оптимальная стратегия для игрока II имеет плотность в достаточно малом интервале справа от a_0. Если $\varphi(a_0) = v$, то $x^* = y^* = I_{a_0}$ – вырожденное распределение, сконцентрированное в точке a_0. Если $\varphi(a_0) = v$, то $y^* = y_{a_0}$, а игрок I не имеет достижимой оптимальной стратегии; ε-оптимальная стратегия для игрока I имеет плотность в достаточно малом интервале справа от a_0.

Стратегии x_0 и y_0 будут являться ε-оптимальными в случае выполнения условий

$$\begin{cases} K(x_0, y) \ge v - \varepsilon, & \forall y \in Y, \\ K(x, y_0) \le v + \varepsilon, & \forall x \in X, \end{cases}$$

v – значение игры.

ε-оптимальность означает, что игрок II применяя ε-оптимальную стратегию, может помешать игроку I получить выигрыш больше чем $v + \varepsilon$.

Аналогично игрок I, применяя ε-оптимальную стратегию, может гарантировать себе, по крайней мере $v - \varepsilon$ независимо от стратегии игрока II для любого $\varepsilon > 0$.

Пример. [64] Пусть $P_1(x)$ – вероятность поражения игрока II игроком I является непрерывной функцией, причём $P_1(0) = 0$ и $P_1(1) = 1$. А $P_2(y)$ – вероятность поражения игрока I игроком II возрастает по y и $P_1(0) = 0$ и $P_1(1) = 1$.

Пусть, если I поражает II, то его выигрыш составляет + 1, а если наоборот, то красный выигрывает – 1, если оба игрока промахнулись или уничтожены, то выигрыш I составил 0.

Если попадание гарантируется (самонаводящиеся торпеды, ракеты либо снаряды), то имеем матричную игру в нормальной форме:

		II	
		Пли	Дробь
I	Пли	0	1
	Дробь	– 1	0

Для этой игры мы уверенно можем отыскивать оптимальные смешанные стратегии (стрелять или не стрелять).

Определим ядро данной игры. Функция цены игры будет иметь вид

$$v = (-1) \times [1 - P_1(x)].$$

Поэтому, когда I-й игрок действует, а II-й – выжидает, то

$$l(x) = P_1(x) - [1 - P_1(x)] = 2\,P_1(x) - 1.$$

По аналогии, если как I-й так и II-й игрок ведут огонь, то

$$\varphi(x) = P_1(x)\,[1 - P_2(x)] - P_2(x)\,[1 - P_1(x)] = P_1(x) - P_2(x)].$$

И, наконец, если II-й игрок действует, а I-й выжидает, имеем

$$n(y) = -P_2(y) + [1 - P_2(y)] = 1 - 2\,P_2(y).$$

Таким образом, ядро игры есть

$$K(x,y) = \begin{cases} 2P_1(x) - 1, & x < y, \\ P_1(x) - P_2(x), & x = y, \\ 1 - 2P_2(y), & x > y. \end{cases} \qquad (4.35)$$

Для данного ядра справедливо утверждение теоремы, поэтому могут быть найдены оптимальные чистые стратегии. Параметр a_0 может быть найдено при решении уравнения

$$2P_1(x)-1=1-2P_2(x)=v,$$

а сама цена

$$v=2P_1(a_0)-1=1-2P_2(a_0)\Rightarrow 2P_1(a_0)-1\cdot[P_1(a_0)+P_2(a_0)]=P_1(a_0)-P_2(a_0).$$

Оптимальная стратегия первого игрока определяется решением

$$P_1(x^*)+P_2(x^*)=1.$$

Оптимальная стратегия второго игрока определяется решением

$$P_1(y^*)+P_2(y^*)=1$$

Заметим, что равенство

$$P_1(y^*)-P_2(y^*)=P_1(x^*)-P_2(x^*)$$

есть отражение равновесия для пары (x^*, y^*) между желанием задержать выстрел и опасностью промедления.

Ядро (4.35) справедливо для одинаковой результативности выстрелов обоих противников (враг убит или остался в живых). Если же, при завершении игры, результат будет отличен от единичного, то, полагая вероятности поражения равными $P_1(x)$ и $P_2(y)$, а система выигрышей красного игрока такова, что:

α – если поражён синий игрок;

β – если поражён красный игрок;

γ – поражены оба игрока;

0 – если оба игрока не поражены, промахнылись.

Если определить стратегию x игрока I, а стратегию y игрока II как моменты упреждающего выстрела или выстрела после промаха противника, при котором вероятность поражения будет равна единице, можно отыскать математическое ожидание выигрыша игрока I.

$$K(x,y)=\begin{cases}(\alpha-\beta)P_1(x)+\beta, & x<y,\\ \alpha P_1(x)+\beta P_2(x)+(\gamma-\alpha-\beta)\cdot P_1(x)\cdot P_2(x), & x=y,\\ \alpha-(\alpha-\beta)\cdot P_2(y), & x>y.\end{cases}$$

При $\alpha > \beta$ определятся значение a_0

$$(\alpha-\beta)\cdot P_1(x)+\beta=\alpha-(\alpha-\beta)\cdot P_2(x)=v.$$

Когда $\varphi(a_0)\geq v$, красный игрок будет иметь оптимальную стратегию

$$\alpha P_1(a_0)+\beta P_2(a_0)+(\gamma-\alpha-\beta)\cdot P_1(a_0)\cdot P_2(a_0)\geq \alpha P_1(a_0)+\beta P_2(a_0),$$

если $\gamma-\alpha-\beta\geq 0$, то игрок I, в соответствии с теоремой имеет стратегию $x^*=x_{a_0}$. А у игрока II, в этом случае, нет оптимальной стратегии, и он должен придерживаться стратегии, сколь угодно близко совпадающей с x^*.

Если же $\gamma-\alpha-\beta<0$, то у игрока II имеется чистая стратегия $y^*=y_{a_0}$, а красный игрок таковой не имеет.

Пример. Рассмотрим, дуэль Е. Онегина и В. Ленского. По её условиям, противники начинают сходиться на $D_{\max}$ шагах от барьеров, которые расположены на расстоянии $D_{\min}$ от центра позиции, и за время сближения, включая начальную и конечную точки, могут произвести выстрел.

За игрока I примем Онегина, а за игрока II – Ленского, произведя нормировку дистанций вида

$$x=y=\frac{D_{\max}-d}{D_{\max}-D_{\min}},$$

где $D_{\min}\leq d\leq D_{\max}$ – дистанция выстрела, придём к игре на квадрате. Условимся, что "подстреленный" Ленский оценивается Онегиным в + 1, а собственные ранение или гибель – в – 1, взаимные повреждения или промахи – в 0.

Для данного случая будет справедливо ядро (4.35) со всеми вытекающими последствиями.

Пусть [55] вероятности в (4.35) описываются функциями вида

$$P_1(x) = 1 — x,\ P_2(y) = 1 — y^2,$$

являющимися монотонно убывающими, тогда

$$1 = P_1(x^*) + P_2(x^*) = \Rightarrow x + x^2 = 1,$$

откуда $x^* = 0{,}62$, аналогично и $y^* = 0{,}62$. Цена игры составит 0,24. Расчёт дистанции выстрела, из нормированного пространства в реальное, будет определяться выражением

$$d = D_{\max} - x^*\cdot(D_{\max} - D_{\min}) = D_{\max} - y^*\cdot(D_{\max} - D_{\min}).$$

Пусть барьеры положены в шести шагах друг от друга (жёсткая дуэль по условиям того времени) $D_{\min} = 6$ "…Зарецкий 32 шага отмерил с точностью отменной, развёл друзей на крайний след и каждый взял свой пистолет…", то есть $D_{\max} = 32$. Отсюда мы можем рассчитать дистанцию выстрела как $32 — 0{,}62 \times 26 = 15{,}88$, то есть примерно за восемь шагов друг от друга или за два шага до барьера.

4.9.3. Игры с выбором момента времени действия в условиях неполной информации (бесшумные и смешанные дуэли)

В ряде дуэльных ситуаций каждая сторона имеет лишь частичную информацию (а точнее, хотя бы одна из сторон не имеет информации) о действиях противоположной стороны. И необходимо выбирать момент времени в условиях неполной информации.

В этих дуэлях функции L и N зависят как от x так и от y, а оптимальные стратегии будут смесями чистых стратегий типа I_x.

Пусть игроки производят только по одному выстрелу, но не знают, выстрелил ли его противник или нет.

Рассчитаем ядро при условиях, что $P_1(x) = P_2(x) = x$, $P_1(y) = P_2(y) = y$, а так же если игрок I поражает игрока II, то его выигрыш составляет + 1, а если наоборот, то красный выигрывает – 1, если оба игрока промахнулись или уничтожены, то выигрыш I составил 0.

Считая величины x и y независимыми, получим

$$K(x, y) = \begin{cases} x - y + xy, & x < y, \\ 0, & x = y, \\ x - y + xy, & x > y, \end{cases}$$

которое является симметричным, так как $K(x, y) = -K(x, y)$, такая игра называется симметричной и стратегии, которые оптимальны для одного игрока, являются оптимальными и для другого. В [30] получено, что, в этом случае, оптимальные стратегии игроков имеют плотность вероятностей вида

$$f^*(x) = \begin{cases} 0, & 0 \le x \le \dfrac{1}{3}; \\ \dfrac{1}{4x^2}, & \dfrac{1}{3} \le x \le 1. \end{cases}$$

Отсюда интегрированием можно найти

$$F^*(x)=\begin{cases}0, & 0\le x\le \dfrac{1}{3};\\ \displaystyle\int_{\frac{1}{3}}^{x} f^*(x)dx=\frac{1}{8}\left(9-\frac{1}{x^2}\right), & \dfrac{1}{3}\le x\le 1.\end{cases}$$

Пусть красный игрок знает о производстве выстрела синим игроком, а синий игрок такого рода информацией о действиях красного игрока не располагает. Вероятности поражения примем $P_1(x) = P_2(x) = x$, $P_1(y) = P_2(y) = y$. Тогда функция выигрыша примет следующий вид

$$K(x,y)=\begin{cases}x-y+xy, & x<y,\\ 0, & x=y,\\ x-2y, & x>y.\end{cases}$$

4.10. Игровые модели неантагонистических конфликтов (биматричные игры)

Антагонизм матричных игр двух персон заключался в том, что одна из сторон выигрывала то, что проигрывала другая.

Однако часто имеет место ситуация, когда игроки, чьи интересы сталкиваются, получают (или выплачивают) разные суммы при одних и тех же обстоятельствах. Поэтому суммарный выигрыш оказывается, в этом случае, не равен нулю, а игры называются играми с ***ненулевой суммой***.

Всякая конечная игра с ненулевой суммой формально описывается парой ($m \times n$) матриц $A = [a_{i,j}]$ и $B = [b_{i,j}]$, или биматрицей $(A, B) = [a_{i,j}, b_{i,j}]$, $i = 1, \ldots, m$; $j = 1, \ldots, n$. Благодаря особенности описания, такие игры ***называются биматричными***.

Пары (i, j) — ситуации в чистых стратегиях, числа $a_{i,j}$ и $b_{i,j}$ составляют выигрыши игроков I и II соответственно в указанных ситуациях.

По аналогии с играми с нулевой суммой, распределение вероятностей на множестве чистых стратегий будем определять как смешанные стратегии, и математические ожидания выигрышей игроков будут определяться выражениями

$$\sum_{i=1}^{m}\sum_{j=1}^{n} a_{i,j}p_iq_j \text{ и } \sum_{i=1}^{m}\sum_{j=1}^{n} b_{i,j}p_iq_j .$$

Игры с ненулевой суммой условно подразделяются на два класса:

1) бескоалиционные (некооперативные) биматричные игры, в которых напрочь исключаются совместные договорённости при выборе стратегии игроками;
2) кооперативные биматричные игры, в которых допускаются любые коалиции или кооперации.

4.10.1. Некооперативные биматричные игры

В основу построения оптимальных стратегий в биматричных играх положен следующий ***принцип***: отклонение от ситуации, сложившейся в результате оптимального поведения, либо невыгодно само по себе, либо, в результате отклонения появляется возможность перейти с выгодой к новой ситуации, достижимой при помощи оптимального поведения.

Ситуация ***является равновесной***, если ни один из игроков не имеет основания для изменения своей стратегии, если другой игрок будет придерживаться своей.

Математическое описание ситуации равновесия определяется через средние выигрыши обоих игроков:

$$\left.\begin{aligned} v_I(P^*,Q^*) &= \sum_{i=1}^{m}\sum_{j=1}^{n} a_{i,j} p^*_i q^*_j \geq \sum_{i=1}^{m}\sum_{j=1}^{n} a_{i,j} p_i q^*_j, \\ v_{II}(P^*,Q^*) &= \sum_{i=1}^{m}\sum_{j=1}^{n} b_{i,j} p^*_i q^*_j \geq \sum_{i=1}^{m}\sum_{j=1}^{n} b_{i,j} p^*_i q_j, \end{aligned}\right\} \qquad (4.36)$$

где обозначено $v_I(P^*, Q^*)$ – математическое ожидание выигрыша I-го игрока, а $v_{II}(P^*, Q^*)$ – математическое ожидание выигрыша II-го игрока.

Ситуация равновесия в биматричных играх аналогична понятию оптимальная стратегия в матричных играх с нулевой суммой. Строго говоря, в последних ситуация равновесия образуется оптимальными стратегиями: если (P^*, Q^*) и $(P^\blacklozenge, Q^\blacklozenge)$ равновесные ситуации, то $(P^*, Q^\blacklozenge)$ и $(P^\blacklozenge, Q^*)$ тако же являются ситуациями равновесия,

$$v_I = \sum_{i=1}^{m}\sum_{j=1}^{n} h_{i,j} p^*_i q^*_j = \sum_{i=1}^{m}\sum_{j=1}^{n} h_{i,j} p^\diamondsuit_i q^\diamondsuit_j = \sum_{i=1}^{m}\sum_{j=1}^{n} h_{i,j} p^\diamondsuit_i q^*_j = \sum_{i=1}^{m}\sum_{j=1}^{n} h_{i,j} p^*_i q^\diamondsuit_j ,$$

где P^*, $P^\blacklozenge$, Q^*, $Q^\blacklozenge$ – оптимальные смешанные стратегии игроков. То есть, в матричных играх, один из векторов, P^* либо Q^* определяет равновесную ситуацию независимо от другого.

Определение [8, 64]. Ситуации равновесия $(P,\ Q)$ и $(P^\blacklozenge,\ Q^\blacklozenge)$ называются:

а) эквивалентными, если $v_I(P,\ Q) = v_I(P^\blacklozenge,\ Q^\blacklozenge)$ и $v_{II}(P,\ Q) = v_{II}(P^\blacklozenge,\ Q^\blacklozenge)$;

б) взаимозаменяемыми, если $(P,\ Q^\blacklozenge)$ и $(P^\blacklozenge,\ Q)$ также являются ситуациями равновесия.

В играх с нулевой суммой ситуации равновесны, эквивалентны и взаимозаменяемы.

Определение [8, 64]. Пара $(P,\ Q)$ называется совместно недопустимой (совместно подчинённой) если существует пара $(P^\blacklozenge,\ Q^\blacklozenge)$, такая что

$$v_I(P^\blacklozenge,\ Q^\blacklozenge) > v_I(P,\ Q) \text{ и } v_{II}(P^\blacklozenge,\ Q^\blacklozenge) > v_{II}(P,\ Q). \qquad (4.37)$$

В этом случае говорят, что пара $(P^\blacklozenge,\ Q^\blacklozenge)$ совместно доминирует над парой $(P,\ Q)$. Пара $(P^\blacklozenge,\ Q^\blacklozenge)$ является совместно допустимой тогда и только тогда, когда она не является совместно подчинённой другой паре.

В [8, 64] показано, что каждая биматричная игра имеет, по крайней мере, одну ситуацию равновесия. Стратегии игроков, её образующие, называются равновесными, а пара чистых равновесных стратегий – ***седловой точкой Нэша***.

Как просматривается из сказанного выше, седловая точка Нэша является аналогом седловой точки матричной игры, при условии, что ни одному игроку не выгодно менять свою чистую стратегию в ситуации, когда противник придерживается своей чистой стратегии.

Пример 1. Пусть имеем биматричную игру в форме

$$A = \begin{pmatrix} 3 & 2 \\ 1 & 0 \end{pmatrix},\ B = \begin{pmatrix} 4 & 2 \\ 3 & 1 \end{pmatrix}.$$

В этом случае существует одна точка Нэша $P^* = (1,\ 0)$ и $Q^* = (1,\ 0)$. Эта же пара является совместно допустимой.

Определение [8, 48, 64]. Биматричная игра имеет решение в строгом смысле, когда среди совместно допустимых пар стратегий существует пара равновесных стратегий, а все совместно допустимые равновесные пары взаимозаменяемы и эквивалентны.

Пример 2. Биматричная игра

$$A = \begin{pmatrix} 30 & 10 \\ 15 & 20 \end{pmatrix}, \qquad B = \begin{pmatrix} 40 & 20 \\ 17 & 30 \end{pmatrix}.$$

В этом случае, согласно (4.36), существует сразу две точка Нэша:

$$[P^* = (1,\ 0);\ Q^* = (1,\ 0)] \text{ и } [P^\blacklozenge = (0,\ 1);\ Q^\blacklozenge = (0,\ 1)].$$

Однако пара (P, Q) совместно доминирует пару ($P^{\blacklozenge}$, $Q^{\blacklozenge}$), потому что $v_I(P, Q) = 30 > v_I(P^{\blacklozenge}, Q^{\blacklozenge}) = 20$, поэтому данная игра имеет решение в строгом смысле.

Решение в строгом смысле может существовать не всегда. Полагается, что если игра не имеет ни одной совместно допустимой пары, и если две любые равновесные пары взаимно заменяются, то игра называется ***разрешимой в смысле Нэша***[51, 52].

Пример 3. Задана биматричная игра

$$A = \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix}, \quad B = \begin{pmatrix} 3 & 3 \\ 1 & 1 \end{pmatrix}.$$

В ней всякая пара чистых или смешанных стратегий определяет ситуацию равновесия, следовательно, игра разрешимо в смысле Нэша.

Пример 4. А вот в случае биматричной игры

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$$

имеется две ситуации равновесия [$P^* = (1, 0)$; $Q^* = (1, 0)$] и [$P^{\blacklozenge} = (0, 1)$; $Q^{\blacklozenge} = (0, 1)$], но они не взаимозаменяемы, поэтому игра не разрешима в смысле Нэша, но разрешима в строгом смысле.

Пример 5. Для биматричной игры

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 3 \\ 4 & 1 \end{pmatrix}$$

нет точек Нэша и решений в чистых стратегиях. Необходимо искать смешанные стратегии.

Теорема[64]. Если в некооперативной биматричной игре (A, B) в стратегиях (P^*, Q^*) ситуации равновесия (P^*, Q^*) активны все чистые стратегии ($p_i^* > 0$ и $q_i^* > 0$) и если один из игроков придерживается равновесной смешанной стратегии, то математическое ожидание выигрыша другого игрока остаётся постоянным.

На основании этой теоремы выполняется построение систем уравнений для поиска оптимальных смешанных стратегий.

Для вычисления равновесных смешанных стратегий в ($m \times n$) биматричных играх, необходимо составить $n_a + 1$ уравнение для синего игрока, и $m_a + 1$ уравнение для игрока красного, где m_a и n_a – число активных стратегий для красного и синего игрока соответственно, вида

$$v_I(P^*, Q^*) = \sum_{j=1}^{n_a} a_{i,j} q_j; \qquad \sum_{j=1}^{n_a} q_j = 1; \qquad v_{II}(P^*, Q^*) = \sum_{i=1}^{m_a} b_{i,j} p_i; \qquad \sum_{i=1}^{m_a} q_i = 1;$$

и решить эти две системы, после чего проверить выполнение неравенств (4.36) ситуации равновесия для полученных решений P^*, Q^*, $v_I(P^*, Q^*)$ и

$v_{II}(P^*, Q^*)$. Если неравенства выполняются, то смешанные стратегии P^* и Q^* являются равновесными.

Рассмотрим систему уравнений, для игры 2 × 2 (как тривиальный вариант)

$$\left.\begin{aligned} v_I &= a_{11}q_1^* + a_{12}q_2^* \\ v_I &= a_{21}q_1^* + a_{22}q_2^* \\ q_1^* + q_2^* &= 1 \\ v_{II} &= b_{11}p_1^* + b_{21}p_2^* \\ v_{II} &= a_{12}p_1^* + a_{22}p_2^* \\ p_1^* + p_2^* &= 1 \end{aligned}\right\} \Rightarrow \begin{cases} p_1^* = \dfrac{b_{22} - b_{21}}{b_{11} - b_{12} - b_{21} + b_{22}} \\ q_1^* = \dfrac{a_{22} - a_{12}}{a_{11} - a_{12} - a_{21} + a_{22}} \end{cases}. \qquad (4.38)$$

Применяя эти формулы к расчёту биматричной игры последнего примера, получим

$$P^* = \left\{\frac{3}{5}; \frac{2}{5}\right\}, Q^* = \left\{\frac{2}{3}; \frac{1}{3}\right\}; \qquad v_I(P^*, Q^*) = \frac{11}{5}; \quad v_{II}(P^*, Q^*) = \frac{11}{5}.$$

Если биматричная игра не разрешима ни в строгом в смысле, ни в смысле Нэша, то вводится в оборот понятие “приведённая игра”, получающаяся при исключении доминируемых стратегий в исходной игре.

При этом говорят, что биматричная игра разрешима в слабом смысле, если соответствующая ей приведённая игра разрешима в строгом смысле.

Принцип доминирования, несколько более видоизменённый, выглядит следующим образом.

1. Если *i*-я строка матрицы *A* строго доминируется некоторой выпуклой комбинацией других строк (в частности, если элементы некоторой строки матрицы *A* больше соответствующих элементов строки *i*), то, без изменения множества равновесных стратегий обоих игроков, *i*-я строка матриц из матриц *A* и B может быть вычеркнута.

2. Если *j*-й столбец матрицы *B* строго доминируется некоторой выпуклой комбинацией других столбцов (в частности, если элементы некоторого столбца матрицы *B* больше соответствующих элементов столбца *j*), то, без изменения множества равновесных стратегий обоих игроков, *j*-й столбец из матриц *A* и B может быть вычеркнут.

Таким образом, стратегии исключаются из списка активных при выполнении условий:

$$\left.\begin{array}{l} p_k^* = 0 \text{ если } a_{i,j} > a_{k,j}, \text{ для } j = \overline{1,n} \text{ либо } a_{k,j} < \sum_{i=1}^{m} \lambda_i a_{k,j}, \text{ для } i = \overline{1,m}; \\ q_r^* = 0 \text{ если } b_{i,j} > b_{k,r}, \text{ для } i = \overline{1,m} \text{ либо } b_{i,r} < \sum_{j=1}^{n} \lambda_j b_{i,r}, \text{ для } j = \overline{1,n}; \end{array}\right\}$$

где $\begin{cases} \sum_{i=1}^{m} \lambda_i = \sum_{j=1}^{n} \lambda_j = 1, \\ \lambda_i \geq 0, \lambda_j \geq 0. \end{cases}$.

Пример 6. Пусть биматричная игра (*A, B*) задаётся следующими матрицами [64]:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 0 & 1 \\ 2 & 3 & 0 \end{pmatrix}, \qquad B = \begin{pmatrix} 2 & 4 & 0 \\ 4 & 3 & 6 \\ 4 & 0 & 8 \end{pmatrix}.$$

Видно, что игра седловых точек Нэша не имеет, посему применим принцип доминирования. В матрице *B* для первого столбца выполняется неравенство

$$b_{i,r} < \sum_{j=1}^{n} \lambda_j b_{i,r}, \text{ так как } 2 = \frac{1}{2} \cdot 4 + \frac{1}{2} \cdot 0; 4 < \frac{1}{2} \cdot 3 + \frac{1}{2} \cdot 6; 4 = \frac{1}{2} \cdot 0 + \frac{1}{2} \cdot 8,$$

по каковой причине матрицы сокращаются

$$A' = \begin{pmatrix} 2 & 3 \\ 0 & 1 \\ 3 & 0 \end{pmatrix}, B' = \begin{pmatrix} 4 & 0 \\ 3 & 6 \\ 0 & 8 \end{pmatrix}.$$

В уменьшенной матрице A' вторая стратегия доминируется первой, ибо $a_{2j} < a_{1j}$, $j = 2, 3$, после исключения которой матрицы сократятся до размерностей 2×2.

$$A'' = \begin{pmatrix} 2 & 3 \\ 3 & 0 \end{pmatrix}, B'' = \begin{pmatrix} 4 & 0 \\ 0 & 8 \end{pmatrix}.$$

Формальный расчёт по (4.38) и учёт стратегий, не признанных активными, позволяет найти решение исходной биматричной игры

$$P^* = \left\{\frac{2}{3}; 0; \frac{1}{3}\right\}, \qquad Q^* = \left\{0; \frac{3}{4}; \frac{1}{4}\right\}; \qquad v_I(P^*, Q^*) = \frac{3}{2}; \qquad v_I(P^*, Q^*) = \frac{8}{3}.$$

Результат проверки неравенства (4.36), показывает, что нами найдены равновесные стратегии игроков I и II в биматричной игре (*A, B*). Полученная пара стратегий не является совместно допустимой. Об этом свидетельствует неравенство (4.37). Поэтому, данная игра (*A, B*) неразрешима в строгом смысле.

Тем не менее, найденная пара (P^*, Q^*) – единственная ситуация равновесия, которая и будет решением в смысле Нэша.

Поскольку под теоретическими игроками, как правило, понимаются люди, то им не чуждо всё человеческое. При этом возникает феномен "психологического" доминирования.

Пример 7. Пусть биматричная игра описывается матрицами

$$A = \begin{pmatrix} 4 & 10 \\ 8 & 5 \end{pmatrix}, \quad B = \begin{pmatrix} -30 & 6 \\ 8 & 4 \end{pmatrix}.$$

Пары равновесных стратегий, [P = (0, 1); Q = (1, 0)] и [$P^{\blacklozenge}$ = (1, 0); $Q^{\blacklozenge}$ = (0, 1)] являются совместно допустимыми, но невзаимозаменяемыми и неэквивалентными. Смешанные стратегии образуют ещё одну ситуацию равновесия

$$P^{\lozenge\lozenge} = \left\{\frac{1}{10}; \frac{9}{10}\right\}, \qquad Q^{\lozenge\lozenge} = \left\{\frac{5}{9}; \frac{4}{9}\right\}; \; v_I(P^{\lozenge}, Q^{\lozenge}) = \frac{20}{3}; \qquad v_{II}(P^{\lozenge}, Q^{\lozenge}) = \frac{21}{5}.$$

Пары [P, Q] и [$P^{\blacklozenge}$, $Q^{\blacklozenge}$] или [$P^{\blacklozenge}$, $Q^{\blacklozenge}$] и [$P^{\blacklozenge\blacklozenge}$, $Q^{\blacklozenge\blacklozenge}$] невзаимозаменяемые. Поэтому данная игра (A, B) неразрешима ни в строгом смысле, ни в смысле Нэша. Поскольку стратегии игроков активны, то приведение игры неосуществимо, поэтому игра неразрешима и в слабом смысле.

Если же рассматривать пару равновесных стратегий [$P^{\blacklozenge}$, $Q^{\blacklozenge}$], то второй игрок психологически будет избегать применения стратегии № 1, поскольку при выборе 1-м игроком стратегии № 1 (он будет на ней "настаивать", поскольку выигрыш по этой строке максимален) его проигрыш составит 30 единиц. Поэтому получается, что пара [$P^{\blacklozenge}$, $Q^{\blacklozenge}$] "психологически" доминирует над [P, Q].

Учет психологических особенностей просматривается и в игре [38].

$$A = \begin{pmatrix} 4 & 10 \\ 12 & 5 \end{pmatrix}, \qquad B = \begin{pmatrix} -3000 & 6 \\ 8 & 4 \end{pmatrix},$$

в которой пара равновесных стратегий [P = (1, 0); Q = (0, 1)] с вектором выигрышей (10, 6) совместно подчинена паре стратегий [$P^{\blacklozenge}$ = (0, 1); $Q^{\blacklozenge}$ = (1, 0)] при векторе (12, 8). Здесь психологически предпочтительна пара [P, Q], хотя пара [$P^{\blacklozenge}$, $Q^{\blacklozenge}$] есть решение игры (A, B) в строгом смысле.

4.10.2. Кооперативные игры

Данные игровые модели отражают ситуацию, когда участники игры, преследуя разные цели, которые частично совпадают, могут прийти к взаимному соглашению. В этом случае между игроками допускается кооперирование:

- заключаются совместные соглашения, совместно выбирается смешанная стратегия;
- возможна передача части выигрыша одного игрока другому.

При этом соблюдается следующее:

- игроки могут передавать друг другу необходимую информацию;
- все соглашения являются обязательными для обеих сторон и включаются в правила игры;
- переговоры между игроками не изменяют оценок игры.

Рассмотрим, первоначально, кооперативную игру двух персон.

Смешанную стратегию, на которую согласны оба игрока называют ***совместной смешанной стратегией***, обозначают как Z, а множество смешанных стратегий – **Z**.

Математическое ожидание выигрышей игроков составит

$$v_I(Z) = \sum_{i,j} a_{ij} z_{ij} ,$$
$$v_{II}(Z) = \sum_{i,j} b_{ij} z_{ij} .$$

где z_{ij} — вероятность, с которой выбирается пара чистых стратегий согласно смешанной стратегии Z.

Точки [$u = v_I(Z)$; $v = v_{II}(Z)$] образуют множество S, которое называется допустимым. Под допустимостью понимается, что для любой точки (u, v) $\in S$, игроки I и II могут получить выигрыши соответственно u и v.

Пусть для пары точек (u, v) и (u', v') выполняется условия $u' > u$ и $v' > v$, тогда точка (u, v) называется совместно подчинённой точке (u', v').

Очевидно, что совместно подчинённые точки рассматривать нецелесообразно, предпочтение следует отдавать совместно неподчинённым точкам. Указанные точки образуют граничную линию (a, b, c, d), показанную на рисунке 4.15 для абстрактной выпуклой области.

Эта линия удовлетворяет принципу оптимальности Парето, и образуют множество Парето. В основе оптимума по Парето лежит идея взаимной выгодности, постулат которой заключается в том, что ни один из выигрышей u или v не может быть увеличен без уменьшения другого. Формально множество Парето описывается следующим образом:

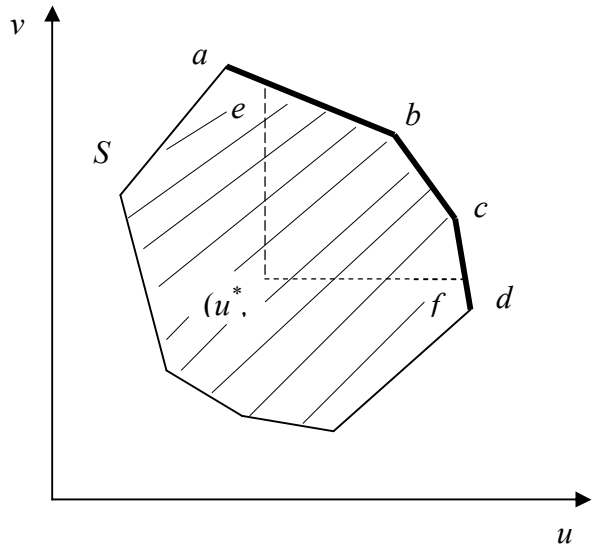


Рисунок 4.15 – Допустимое множество кооперативных стратегий

$$S^0 = \{(\tilde{u}, \tilde{v}) \in S : \neg\exists (u, v) \geq (\tilde{u}, \tilde{v})\}.$$

В любой точке множества S^0 один из игроков получает тем больше, чем меньше получает другой.

Глядя на рисунок 4.14, легко предположить, что игрок I выберет точку *d*, а игрок II выберет точку *a*. При этом не обязательно, что эти чаяния могут осуществиться, ибо возникает резонный вопрос, а сколько позволит выиграть противнику в ущерб себе? Практически в ходе решения игры, эти пожелания нужно как-то совместить.

Житейски разумно предположить, что выигрыш в кооперативной игре должен быть не меньше, чем игрок мог бы получить индивидуально, не связываясь с кооперацией, то есть, руководствуясь своей максиминной стратегией на соответствующей матрице:

$$u^* = \max_P \min_Q \sum_{i=1}^{m} \sum_{j=1}^{n} a_{i,j} p_i q_j ,$$

$$v^* = \max_P \min_Q \sum_{i=1}^{m} \sum_{j=1}^{n} b_{i,j} p_i q_j .$$

Точка с координатами (u^*, v^*) даёт каждому игроку гарантируемый выигрыш в индивидуальной (некооперативной) игре и образует (точнее, способствует образованию) так называемое ***переговорное множество*** (которое обозначается как $\overline{S}^0$) *ebcf*, которое представляет собой решение кооперативной биматричной игры [38].

Выбор точки переговорного множества зависит от индивидуальных различий и психологических свойств игроков, проявляющихся на

переговорах. В плане формальных методов известен подход [38, 55], называемый “схемами арбитража” или “задачей о сделках”, базирующийся на ***теории полезности***.

Изложение теории полезности не входит в задачи дисциплины, однако, некоторые, необходимые для дальнейшего изложения сведения, мы всё же рассмотрим.

Пусть $A, B, C \ldots Z$ – события или явления, одни из которых, для принимающего решение, предпочтителнее, чем другие, либо безразличны.

Между событиями существуют отношение предпочтения $\succ$ и отношение равноценности $\sim$, то есть

- $A \succ B$ – A предпочтительнее B;
- $B \succ A$ – B предпочтительнее A;
- $A \sim B$ – A и B равноценны, то есть ни $A \succ B$, ни $B \succ A$.

Отношения предпочтения обладают свойством транзитивности (переносимости).

Полезность – порядковая или количественная величина, выражающая отношение предпочтения и являющаяся вторичным понятием по сравнению с ним.

То есть, $A \succ B$ не потому, что а имеет б***о***льшую полезность, а наоборот, из-за того, что $A \succ B$, событию A приписывается б***о***льшая полезность.

Пусть, например, $A \succ B \succ C$, где B – достоверное событие, а A и C события, возникающие в лотерее (случайном испытании) с вероятностями p и $(1 - p)$. Очевидно, что наш выбор будет определяться распределением этих вероятностей.

При значении p, близком к 1, предпочтительнее лотерея, а при p около 0, следует выбрать достоверное событие B, и, таким образом происходит переход от выбора случайного испытания до выбора достоверного.

Аналогичная лотерея может быть распространена и на m событий. Главное при этом есть следующее:

- лотерея представляет собой испытание, которое проводится только 1 раз (единичное испытание);
- лотерея может сама по себе является событием;
- возможна лотерея, одним из исходов которой является другая лотерея, и, таким образом, возможна последовательность лотерей.

Лотереи организуются следующим образом:

1. важны лишь окончательные вероятности возможных исходов, а порядок, в котором разыгрывается последовательность лотерей, роли не играет;
2. лотереи, содержащие равноценные события являются равноценными;

3. когда одно из событий предпочтительнее другого, то лотерея, содержащее первое событие, предпочтительнее лотереи, содержащей второе событие;
4. если $A \succ C$, а $C \succ B$, то существует лотерея, включающая A и B с соответствующими вероятностями, которая является равноценной C.

С учётом допущений, изложенных выше, представляется возможным построить функцию полезности u, которая для любых двух событий $A \succ B$

$$u(A) > u(B),\ u[pA + (1 - p)B] = p \cdot u(A) + (1 - p) \cdot u(B)$$

В частности, когда $A_1 \succ\sim A_2 \succ\sim \ldots \succ\sim A_m$ можно построить функцию полезности со следующими параметрами:

$$u(A_1) = 1;$$
$$u(A_i) = u_i, \text{ для } 1 < i < m;$$
$$u(A_m) = 0;$$
$$u(p_1 \cdot A_1, \ldots, p_m \cdot A_m) = p_1 \cdot A_1 + p_2 \cdot A_2 + \ldots + p_m \cdot A_m.$$

Важнейшее свойство функции полезности – линейность относительно лотереи. Это означает, что средняя полезность исходов, входящих в лотерею, равна полезности лотереи.

Пусть на выпуклом допустимом множестве S игроки выбирают точки $T_1(u_1, v_1)$, $T_2(u_1, v_1)$ и т.д, а точка $T^*(u^*, v^*)$ означает случай когда выбора не происходит, а её координаты (u^*, v^*) – её полезности для игрков.

Каждый отрезок, соединяющий точки, даёт пару полезностей, которые получат игроки при розыгрыше лотереи вида $p \cdot T_r + (1 - p) \cdot T_k$, где p – вероятность выбора T_r, а T_r и T_k – крайние точки отрезка.

Как видно из рисунка 4.14, игрок II будет стремиться выбирать точки выше точки $T^*(u^*, v^*)$, а игрок I – правее. Поскольку эти чаяния не совместимы, в качестве компромисса выступает арбитражная схема.

Математически арбитражная схема определяется как

$$\varphi(S, u^*, v^*) = (\tilde{u}, \tilde{v}), \tag{4.39}$$

функция, которая по заданной тройке (S, u^*, v^*) даёт единственную точку $(\tilde{u}, \tilde{v})$, которая называется арбитражным решением игры.

Дж. Нэшем доказано [52], что “справедливое” арбитражное решение игры состоит в том, что точка $(\tilde{u}, \tilde{v})$ есть решение задачи максимизации вида

$$g(u, v) = (u - u^*)(v - v^*), \tag{4.40}$$

при условиях $(u,v) \in S,\ u > u^*,\ v > v^*$.

Схема Нэша означает, что дополнительная полезность должна делиться между игроками в таком же отношении, в котором она может передаваться.

В частности, в случае линейной зависимости, один игрок может передавать другому игроку 1-цу полезности, сам при этом лишаясь только одной единицы полезности, то есть, когда точки арбитражного множества лежат вдоль прямой $u + v = k$, то решением будет точка $(\tilde{u},\tilde{v})$ рассчитываемая по выражениям

$$\left.\begin{aligned} \tilde{u} &= \frac{1}{2}\left(u^* - v^* + k\right), \\ \tilde{v} &= \frac{1}{2}\left(v^* - u^* + k\right), \end{aligned}\right\} \qquad (4.41)$$

где k – максимально возможная полезность, которую игроки могут получить совместно.

Пример [64]. Игроки, действуя совместно, добиваются передачи одной из сторон объекта номинальной стоимостью 100 тыс. у.е. Сторона, получившая объект, выпланчивает партнёру часть его стоимости. Пусть сторона I получает объект и выплачивает компенсацию, а сторона II – получает компенсацию.

Даже не прибегая к мудрёным построениям, понятно, что приемлемое решение представляется парой (0,5; 0,5), что математически подтверждается значениями функций

$$u = 1 - \frac{x}{100}, \text{ и } v = \frac{x}{100},\ 0 \le x \le 100,$$

в предположении, что если игрок I не выплачивает компенсации, то его полезность единица, а если отдаёт полную стоимость объекта – то нуль.

Пусть функции полезности у игроков различны, например

$$u = 1 - \frac{x}{100}, \text{ а } v = \frac{\sqrt{x}}{100},\ 0 \le x \le 100.$$

Выразим функцию u через v как $u = 1 - 100 \cdot v^2$ (Рисунок 4.16).

Выполним максимизацию функции по (4.40)

$$g(u, v) = u \cdot v = (1 - 100 \cdot v^2) \cdot v$$

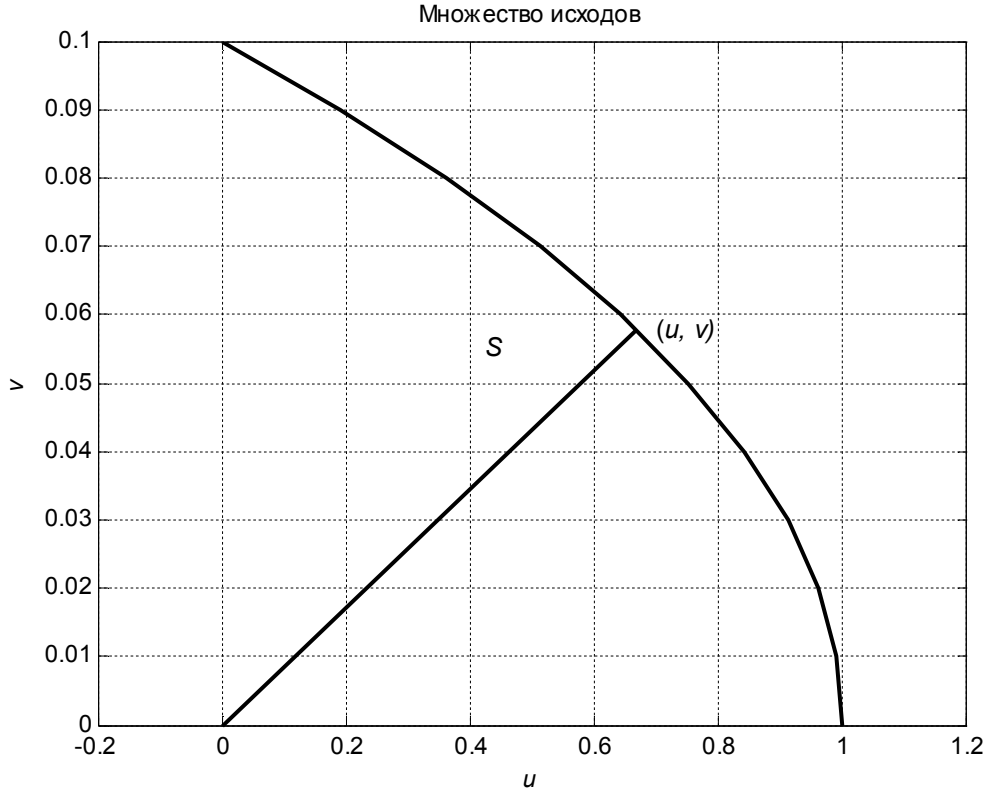


Рисунок 4.16 – Множество S для арбитражного решения

Выполним максимизацию функции по (4.40)

$$g(u, v) = u \cdot v = (1 - 100 \cdot v^2) \cdot v$$

на отрезке [0,0; 0,1]. Используя аппарат математического анализа, после дифференцирования и приравнивания к нулю производной, получим

$$\tilde{v} = \frac{1}{10 \cdot \sqrt{3}}; \tilde{u} = \frac{2}{3}, \text{ откуда } x = \frac{100}{3}.$$

В данном случае, фактическая полезность у игрока II убывает быстрее, чем у I, поэтому он вынужден уступать.

4.10.2.1 Кооперативные игры *n* лиц

Если в конфликт вовлечено более двух персон, то, по сравнению с игрой двух участников, возникают следующие особенности:

- появляется возможность сотрудничества с несколькими участниками;

- число возможных коалиций возрастает.

При этом различаю бескоалиционную и коалиционную игры.

Бескоалиционные игры представляют собой род некооперативной игры, распространённой на n участников [38]. В правилах таких игр особо оговорен запрет на кооперацию.

Коалиционные игры призваны моделировать ситуацию, в которых участников с разными, но не противоречивыми интересами, более двух, и они способны кооперироваться или вступать в коалицию.

Обычно обозначают $N = 1, 2, \dots, n$ – множество игроков, вовлечённых в конфликт.

Определение [64]. Любое непустое подмножество N, включая само N и все его одноэлементные подмножества, называется коалицией.

Для простоты считается, что интересы игроков внутри коалиции совпадают, а межкоалиционные – не пересекаются. Например, в игре могут образоваться две коалиции $\{1, 2, \dots, n\text{-}1\}$ и $\{n\}$, в этом случае n-1 игроков объединяют усилия в одной игре против одного из игроков.

Вместо термина "**выигрыш**" в кооперативной игре применяют термин "**платёж**", под которым понимается передача своего выигрыша или части его.

По правилам игры, для каждого игрока устанавливается платёж, обусловленный правилами игры, а иногда и побочный платёж, который он может заплатить или получить за сотрудничество с другими игроками.

Механизм платежей позволяет образовывать коалиции и определяет их устойчивость, позволяя избежать операций комбинаторики при решении игры.

Пример [64]. Пусть любые два игрока из трёх вступают в коалицию с целью взять с игрока, не вошедшего в коалицию, "дань" в размере 2-х единиц каждому коалиционеру. В данном случае возможны коалиции {1, 2}, {1, 3} и {2, 3}, в которых будут платежи игрокам, соответственно: (2, 2, – 4), (2, – 4, 2) и (– 4, 2, 2). Указанные платежи указывают, что все коалиции равноценны, поэтому любая из них может возникнуть.

Поэтому указанные платежи можно принять за результат игры и считать её решением.

Если правила игры скорректировать, как указано ниже

	Коалиция	Платежи
1	{1, 2}	(2,2; 1,8; – 4)
2	{1, 3}	(2, – 4, 2)
3	{2, 3}	(– 4, 2, 2)

то существование первой коалиции проблематично, игроки 2 и 3 могут получить большее, образовав коалицию {2, 3}, игроки 1 и 3, образовав

коалицию {1, 3}, так же получат по две единицы. Неустойчивость 1-й коалиции делает неясным результат игры, который можно бы было принять за решение.

Если игрок 1 будет согласен заплатить игроку 2, за сотрудничество с ним 0,2 единицы, то игра свелась к предыдущей.

Любую коалицию S можно рассматривать как игрока антагонистической игры, вторым игроком которой является коалиция $N \setminus S$.

Значение такой игры $v(S)$ называется характеристической функцией игры n лиц и определяется на всех подмножествах множества $N = 1, 2, \ldots, n$.

По смыслу характеристическая функция есть полезность, которую коалиция S может извлечь из игры, не зависимо от действий прочих игроков. Характеристическая функция обладает следующими свойствами

$$\begin{cases} v(\varnothing) = 0, \\ v(S \cup T) \geq v(S) + v(T), \end{cases}$$

где S и T – непересекающиеся коалиции.

После розыгрыша игры каждый из игроков получит некоторый размер прямого платежа, обусловленного правилами игры, и "побочного", определяемого соглашениями, заключаемыми при заключении коалиции.

Пусть суммарный платёж игрока i составит x_i. Тогда вся совокупность игроков получит вектор платежей $x = (x_1, x_2, \ldots, x_n)$. Элементы этого вектора (платёж конкретному игроку) удовлетворяют нижеследующим условиям.

1. Окончательный платёж не может быть меньше, чем тот, который бы он мог получить, действуя одиночно, сыграв один против всех

$$x_i \geq v(\{i\}), \quad \forall i \in N. \tag{4.42}$$

2. Какие бы коалиции не образовывались, суммарный платёж не превышает характеристической функции всего множества игроков

$$\sum_{i \in N} x_i \geq v(N). \tag{4.43}$$

Если допустить, что (4.43) нарушается, то каждый игрок был бы в состоянии увеличить свой выигрыш на величину

$$\frac{1}{n}\left(v(N) - \sum_{i \in N} x_i \right).$$

Так как $v(N)$ – наибольший доход, который игроки могут получить, образовав коалицию $N \setminus N = \varnothing$ против коалиции N, поэтому (4.43) равносильно равенству.

Вектор x, удовлетворяющий условиям (4.42) и (4.43), называется ***дележом***.

Множество всех дележей игры v обозначается как $E(v)$ и включает более одного вектора x. Таким образом, на множестве всех дележей необходимо найти такой делёж, который является “решением”.

Считается, что для решения кооперативной игры необходимо постоить множество дележей и характеристическую функцию.

В самом общем виде решение трактовать как набор дележей, любые из которых могут быть реализованы, если игроки выберут стратегии и составят соглашение наилучшим образом.

Для поиска решения вводится понятие C-ядра, обозначаемое как $C(v)$, определяемое множеством дележей, которые удовлетворяют условиям:

$$\left.\begin{aligned} &\sum_{i\in S} x_i \geq v(S),\ \forall S \subset N, \\ &\sum_{i\in N} x_i = v(N). \end{aligned}\right\} \qquad (4.44)$$

Любой делёж из ядра считается устойчивым, потому, что ни одна из коалиций не может обеспечить большего ыигрыша.

Если ядро отсутствует, то принимается концепция фон Неймана и Моргенштерна, называемая НМ-концепцией [52].

Для игры, изложенной выше, характеристическая функция запишется так

$$v(S) = \begin{cases} -4, & \text{когда множество } S \text{ состоит из одного игрока,} \\ 4, & \text{когда множество } S \text{ состоит из двух игроков,} \\ 0, & \text{когда множество } S \text{ состоит из трёх игроков.} \end{cases}$$

Исходом игры будет следующее множество $V = \{(2, 2, -4), (2, -4, 2), (-4, 2, 2)\}$. Оно характеризуется тем, что ни один из его дележей ни для какой коалиции не хуже другого, а любой делёж из $E(v)$, не входящий в V, для какой-нибудь коалицци хуже, чем некоторый делёж V.

То есть, множество V обладает внутренней устойчивостью (ни один делёж из V не хуже другого) и внешней устойчивостью (любой делёж, не входящий в V хуже, чем некоторый делёж из V).

К, сожалению, НМ-решений много и отсутствуют формальные методы и процедуры, поэтому на практике применяется подход Шепли.

Подход, предложенный Шепли, базируется на аксиомах, носящих его имя, и векторе значений игры (вектор Шепли, значение по Шепли)

Указанный вектор, в отличие от НМ-решений, представляющихся уравновешенными платежами, даёт набор априорных оценок всей игры каждым из участником

$$\varphi[v] = \{\varphi_1[v], \varphi_2[v], \dots, \varphi_n[v]\}$$

где i-ая компонента указывает величину, получаемую i-м игроком. Этот вектор удовлетворяет ряду аксиом.

Аксиома 1 [52]. Сумма выигрышей всех игроков равна выигрышу коалиции, состоящей из всех игроков:

$$\sum_{i=1}^{n} \varphi_i[v] = v(N).$$

Смысл аксиомы в том, что если игроки вносят в коалиции лишь ту величину, которую могли бы получить самостоятельно, то они её и получают.

Аксиома 2 [52]. Если для игрока i и любой коалиции S, в которую i не входит, выполняется равенство

$$v[S \cup i] = v(S) + v(i),$$

то

$$\varphi_i[v] = v(i).$$

Аксиома утверждает, что между игроками будет распределён выигрыш, который эти игроки могут получить вследствие объединения.

Аксиома 3 [52]. (Аксиома симметрии). Для любой перестановке игроков π и любого i выполняется равенство

$$\varphi_{\pi(i)}[\pi v] = \varphi_i[\pi],$$

где $\pi(i)$ – игрок, соответствующий перестановке π, а $\pi v(S)$ – такая характеристическая функция, что

$$\pi v(S) = v(\pi^{-1}S),$$

где π^{-1} – обратная перестановка игроков.

То есть, если игроки симметричны относительно характеристической функции, то они получат поровну.

Аксиома 4. Если u и v — две любые игры, то $\varphi_i[u + v] = \varphi_i[u] + \varphi_i[v]$.

Эта аксиома утверждает, что при сложении характеристических функций, выигрыши игроков складываются.

Шепли показал [55], что существует лишь один вектор, который удовлетворяет этим аксиомам, и который определяется выражением:

$$\varphi_i[v] = \sum_{\substack{T \subset N \\ i \in T}} \gamma_i(T)[v(T) - v(T \setminus \{i\})], \qquad (4.45)$$

где $\gamma_i = \dfrac{1}{n!}(t-1)!(n-t)!$, t – число элементов в T, а само выражение (4.44) есть суммы выигрышей по всем коалициям от $T = \{i\}$ до $T = \{1, \ldots, N\}$.

Коэффициент γ_i есть вероятность случайного образования коалиции T при последовательном объединении игроков, при условии, что игрок i присоединяется на шаге t.

Величина под знаком суммы […] теряется коалицией при выходе из неё игрока i.

Вектор (4.45) называется векторм Шепли.

10.4.3. Содержательные примеры биматричных игр

Пример 1 [64].

Противолодочная подводная лодка USA (игрок II) осуществляет поиск противолодочной лодки RU (игрок I), которая, в свою очередь, пытается обнаружить атомный ракетный подводный крейсер USA. В распоряжении сторон имеется пара стратегий поиска: работа ГАС в активном (№ 1) и пассивном (№ 2) режимах [53].

Цели игроков разные, но не прямо противоположные. Пусть a_{ij} – вероятность обнаружения подводной лодки USA, а b_{ij} – вероятность обнаружения нашей подводной лодки. Имеем классическую биматричную игру. Пусть, например,

$$A = \begin{pmatrix} 0{,}30 & 0{,}10 \\ 0{,}15 & 0{,}20 \end{pmatrix}, \quad B = \begin{pmatrix} 0{,}4 & 0{,}20 \\ 0{,}17 & 0{,}30 \end{pmatrix}.$$

В игре существует точка Нэша [$P^* = (1, 0)$; $Q^* = (0, 1)$] с парой стратегий, которые является совместно допустимыми. Следовательно, игра разрешима в строгом смысле. При заданных значениях вероятностей целесообразно использовать активный режим ГАС.

Пример 2 [64].

Некое КБ разработало устройство, которое можно внедрить на одном из двух объектов, при условии согласия всех трёх сторон [33]. Само КБ (игрок № 1) оценивает разработку в a единиц, в бухгалтерии первого предприятия (игрок № 2) считают возможным внедрить разработку за b единиц, а во втором (игрок № 3) за c единиц.

Очевидно, $a < b \le c$ что , так как в противном случае внедрение не состоится (оценки предприятий ниже, чем просит КБ). Фактическая стоимость внедрения x необходимо согласовать с бухгалтерией места внедрения. Понятно, что КБ будет максимизировать стоимость работ в одном из интервалов $[a, b]$ или $[a, c]$, а предприятия – в этих же интервалах минимизировать.

Каждая из сторон стремится к внедрению объекта, но даёт разные оценки стоимости внедрения, цели разные, но не прямо противоположные, поэтому данная игра будет игрой кооперативной.

Определим характеристическую функцию для коалиций {1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}:

$$v(S) = \begin{cases} 0, & \text{для коалиций } \{2\},\{3\},\{2,3\}; \\ a, & \text{для коалиций } \{1\}; \\ b, & \text{для коалиций } \{1,2\}; \\ c, & \text{для коалиций } \{1,3\},\{1,2,3\}. \end{cases}$$

Согласно (4.44), дележом будет являться любая тройка (x_1, x_2, x_3) при условии

$$\sum_{i \in N} x_i = c,\ x_1 \ge a,\ x_2 \ge 0,\ x_3 \ge 0,$$

а множество

$$C(v) = \{(b \le x_1 \le c),\ 0,\ (c - x_1)\}$$

– ядром.

То есть, внедрение состоится на втором предприятии, а его с тоимость будет между b и c.

Полученное “решение” согласуется с экономическим смыслом, поскольку на предприятии № 1 готовы заплатить не больше, чем b, а предприятие № 2 может больше.

Поэтому коалиций {1, 2}, {1, 3} при разумных действиях образоваться не должна, поэтому целесообразна коалиция {2, 3}, которая не позволит предприятию № 1 “остаться с носом”, а предприятию № 2 – переплатить КБ.

Обозначим прибыль коалиции {2, 3} как k и найдём максиминные бескоалиционные значения (u^*, v^*):

- по характеристической функции $u^* = v(\{2, 3\}) = 0$;
- по соображению, что предприятие № 1 реально воспрепятствует получению выигрыша больше, чем $v^* = c - b$, перебивая цену.

Воспользовавшись формулами (4.41) получим:

$$\widetilde{u} = \frac{1}{2}(k - c + b), \widetilde{v} = \frac{1}{2}(k + c - b).$$

Теперь необходимо рассмотреть игру между КБ и коалицией $\{2, 3\}$. На основании характеристической функции имеем $(u^*, v^*) = (a, 0)$. По схеме Нэша КБ получит: $\widetilde{u} = \frac{1}{2}(c + a)$, а коалиция предприятий – $\widetilde{v} = \frac{1}{2}(c - a)$, откуда прибыль коалиции составит $k = \frac{1}{2}(c - a)$, поэтому, в конце концов, получается такой вектор платежей

$$V = \left\{\frac{1}{2}(c + a); \frac{1}{4}(2b - a - c); \frac{1}{2}(3c - a - 2b)\right\}.$$

Указанное решение будет иметь смысл, когда $2b - a - c \geq 0 \Rightarrow c + a \leq 2b$.

Окончательно имеем значение игр по Шепли:

$$\begin{cases} \varphi_1[v] = \frac{1}{3}a + \frac{1}{6}b + \frac{1}{2}c; \\ \varphi_2[v] = -\frac{1}{6}a + \frac{1}{6}b; \\ \varphi_3[v] = -\frac{1}{6}a - \frac{1}{6}b + \frac{1}{2}c. \end{cases}$$

Пример 3[64].

Для заправки кораблей топливом n сторон должны строить хранилища. Пусть стоимость хранилища задаётся возрастающей функцией от его объёма. Потребности каждой стороны в топливе определяются с помощью функции $f_i(t)$, $i = \overline{1, n}$, где t – время наступления потребности, откуда суммарный объём хранилищ $\max\limits_t \sum\limits_{i=1}^{n} f_i(t)$. Когда образуется

коалиция S, то объём такого хранилища составит $\max\limits_{t}\sum\limits_{i\in S} f_i(t)$, а его стоимость – $F\left(\max\limits_{t}\sum\limits_{i\in S} f_i(t)\right)$.

Необходимо определить коалиции, которые займутся строительством и число таких хранилищ.

Будем считать, что это игра n лиц с характеристической функцией

$$v(s) = F\left(\max_{t}\sum_{i\in S} f_i(t)\right).$$

Применение подхода Шепли, после вычисления компонент вектора $\varphi_i[v]$ для каждой коалиции необходимо проверить выполнение неравенства

$$\varphi_{S_i}[v] < \varphi_i[v],$$

где $\varphi_{S_i}[v]$ – вектор Шепли, заданный функцией на подмножествах коалиций. При удовлетворении неравенства, коалиция S будет строить своё хранилище, игроки, не вошедшие в неё – своё. Если все игроки вошли в одну коалицию, то они сообща строят общее хранилище, а бремя расходов распределяется согласно вектору Шепли.

Пусть, в конкретных цифрах, имеем [64] характеристическую функцию вида

Коалиция $\{S\}$	{1}	{2}	{3}	{1, 2}	{1, 3}	{2, 3}	{1, 2, 3}
Характеристическая функция $v\{S\}$	2	3	2,5	4	3,9	5	6

Для коалиций {1, 2, 3}, {1, 2}, {2, 3} и {1, 3} по формуле (4.45) найдём, сперва

$$\varphi_1[v] = \frac{2!0!}{3!}(6-5) + \frac{1!1!}{3!}(4-3) + \frac{1!1!}{3!}(3{,}9-2{,}5) + \frac{0!2!}{3!}(2-0) = \frac{8{,}4}{6};$$

$$\varphi_2[v] = \frac{2!0!}{3!}(6-3{,}9) + \frac{1!1!}{3!}(4-2) + \frac{1!1!}{3!}(5-2{,}5) + \frac{0!2!}{3!}(3-0) = \frac{14{,}7}{6};$$

$$\varphi_3[v] = \frac{2!0!}{3!}(6-4) + \frac{1!1!}{3!}(3{,}9-2) + \frac{1!1!}{3!}(5-3) + \frac{0!2!}{3!}(2{,}5-0) = \frac{12{,}9}{6}.$$

Затем, для коалиций {1, 2}, {2, 3} и {1, 3} получается

$$\varphi_1'[v] = \frac{3}{2}; \varphi_1''[v] = \frac{3{,}5}{2}; \varphi_2'[v] = \frac{5}{2}; \varphi_2''[v] = \frac{5{,}5}{2}; \varphi_3'[v] = \frac{4{,}5}{2}; \varphi_3''[v] = \frac{4{,}4}{2}.$$

Сравнив $\varphi_i[v], \varphi_i'[v]$ и $\varphi_i''[v]$, заметим, что неравенство $\varphi_{S_i}[v] < \varphi_i[v]$ не выполняется, поэтому игрокам целесообразно построить одно мегахранилище, при этом расходы определяться таким образом:

- первый компаньон – $\frac{8{,}4}{6}$;

- второй компаньон – $\frac{14,7}{6}$;
- третий компаньон – $\frac{12,9}{6}$

условных единиц затрат.

Иногда распределение ресурсов n игроков задаётся при помощи функций вида $f(x_1, x_2, \ldots, x_n)$, которые обладают следующими свойствами

- f линейный по каждой переменной полином;
- $f(x) = 0$, если $x = 0$;
- $f(x) = f(\pi x)$, где π – произвольная перестановка.

В [55] доказана следующая **теорема**. Существует лишь один набор линейных по каждой переменной полиномов $\varphi_1, \varphi_2, \ldots, \varphi_n$ который удовлетворяет аксиомам:

а) если $x_i = 0$, то $\varphi_i(x) \equiv 0$;

б) если π_{ij} – произвольная перестановка, которая переводит игрока i в j, то $\varphi_i(x) \equiv \varphi_i(\pi_{ij} x)$;

в) если π_{ij} – произвольная перестановка, которая переводит игрока i в i, то $\varphi_i(x) \equiv \varphi_i(\pi_i x)$;

г) $\sum_{i=1}^{n} \varphi_i(x) \equiv f(x)$.

Этот набор является вектором Шепли для игры с характеристической функцией $v(S) = f(x^S)$, где $x_i^S = 0$ при $x \notin S$, $x_i^S = x_i$ при $x \in S$.

Действительно, по условиям полином имеет вид

$$\alpha_1 \sum_{i=1}^{n} x_i + \alpha_2 \sum_{i<j} x_i x_j + \alpha_3 \sum_{i<j<k} x_i x_j x_k + \ldots + \alpha_n x_1 x_2 \ldots x_n,$$

где α_i – некоторый коэффициент.

По аксиомам а) и г) получается, что

$$\varphi_1(x) = \beta_1 x_1 + \beta_2 x_1 \sum_{i \neq 1} x_j + \beta_3 x_1 \sum_{1 \neq j<k} x_j x_k + \ldots + \beta_n x_1 x_2 \ldots x_n,$$

аксиома б) обеспечивает

$$\varphi_i(x) = \beta_1 x_i + \beta_2 x_i \sum_{i \neq j} x_j + \beta_3 x_i \sum_{\substack{j<k \\ j \neq i \\ k \neq i}} x_j x_k + \ldots + \beta_n x_i \prod_{k \neq i}^{n} x_k, \qquad (4.46)$$

а применение аксиомы г)

$$f \equiv \sum_{i=1}^{n} \varphi_i(x) = \beta_1 \sum_{i=1}^{n} x_i + \beta_2 \sum_{i=1}^{n} x_i \sum_{i \neq j} x_j + \ldots + \beta_n \sum_{i=1}^{n} x_i \prod_{k \neq i}^{n} x_k,$$

откуда $\beta_1 = \alpha_1; \beta_2 = \frac{\alpha_2}{2}; \ldots; \beta_n = \frac{\alpha_n}{n}$.

Компоненты вектора Шепли для игры с характеристической функцией $v(S) = f(x^S)$ суть

$$\varphi_i[x] = \sum_{\substack{T \subset N \\ i \in T}} \frac{(i-1)!(n-1)!}{n!} \left[f(x^T) - f(x^{T \setminus \{i\}}) \right].$$

Пример 4 [64]. Пусть цепочка из n предприятий производит продукцию в таком порядке:

- 1-е предприятие добывает сырьё;
- 2-е предприятие делает полуфабрикат из сырья;
- 3-е предприятие из полуфабриката производит промежуточный продукт;

и так далее, по технологической цепочке, причём в каждом её звене единица входного продукта преобразуется в единицу выходного, а стоимость сырья равна 1.

Пусть стоимость возросла, а i-му предприятию требуется уже не единица входного продукта, а $1 + x_i$, $i > 1$.

В новых условиях стоимость полуфабриката n-ого предприятия будет равна

$\prod_{i=1}^{n} (1 + x_i)$, а убыток составит $f(x_1, ..., x_n) = \prod_{i=1}^{n} (1 + x_i) - 1$.

Требуется определить распределение убытков между предприятиями.

По формуле (4.46) получается

$$\varphi_i^*(x) = x_i + \frac{1}{2} x_i \sum_{i \neq j} x_j + \frac{1}{3} x_i \sum_{\substack{j<k \\ j \neq i \\ k \neq i}} x_j x_k + ... + \frac{1}{n} x_i \prod_{k \neq i}^{n} x_k .$$

Пусть $n = 3$. Тогда

$$f(x_1, x_2, x_3) = x_1 + x_2 + x_3 + x_1 x_2 + x_1 x_3 + x_2 x_3 + x_1 x_2 x_3,$$

$$\varphi_i^*(x) = x_i + \frac{1}{2} x_i \sum_{i \neq j} x_j + \frac{1}{3} x_1 x_2 x_3 .$$

Положим $x_1 = 1$. $x_2 = 2$, $x_3 = 3$ и получим:

$$\varphi_1^*(x) = 1 + \frac{1}{2} 1 \cdot (2 + 3) + \frac{1}{3} \cdot 1 \cdot 2 \cdot 3 = 5{,}5 ;$$

$$\varphi_2^*(x) = 2 + \frac{1}{2} \cdot 2 \cdot (1 + 3) + \frac{1}{3} \cdot 1 \cdot 2 \cdot 3 = 8{,}0 ;$$

$$\varphi_3^*(x) = 3 + \frac{1}{2} \cdot 3 \cdot (1 + 2) + \frac{1}{3} \cdot 1 \cdot 2 \cdot 3 = 9{,}5 ;$$

$f(x) = 23$.

В результате общий убыток составит 23 единицы, 1-е предприятие потеряет 5,5 единиц, 2-е потеряет 8, а на долю третьего придётся 9, 5 единиц потерь.

4.11. Вопросы для самоконтроля

1. Что называется “конфликтом”?
2. Что такое “цена игры”?
3. Что называется стратегией?
4. В чём состоит основная задача теории игр?
5. Что означает термин “игра с нулевой суммой”?
6. Что называется правилами игры?
7. Как формально описать игры двух персон?
8. Как Вы понимаете термины “чистые” и “смешанные” стратегии?
9. Что такое “платёжная матрица”?
10. Что утверждает основная теорема теории игр?
11. Что утверждает теорема об активных стратегиях?
12. Как скажется на стратегиях игроков умножение платёжной матрицы на положительное число?
13. Как скажется на стратегиях игроков добавление ко всем элементам платёжной матрицы произвольного числа?
14. Что вкладывается в понятие “оптимальная стратегия”?
15. Как определить седловые точки платёжной матрицы?
16. Когда представляется возможным применение графоаналитический метод для расчёта оптимальных стратегий?
17. На чём обоснован метод графоаналитического решения матричных игр?
18. Как, используя метод доминирования, добиться понижения размерности платёжной матрицы?
19. На чём основано применение симплекс-метода для решения задач теории игр?
20. Каковы достоинства и недостатки итерационного метода решения игровых задач?
21. Какие ситуации описываются моделями позиционных игр?
22. Какие игры называются играми с полной информацией, а какие – с полной памятью?
23. Что такое “стратегия поведения”?
24. Что вкладывается в понятие “информационное множество”?
25. Какая игра называется “игра-компонента”?
26. В чём заключается общий подход к решению многошаговых игр?

27. Когда конфликтная ситуация моделируется стохастической игрой?
28. В какой ситуации необходимо применять усечение игры?
29. В чём отличия рекурсивной игры от стохастической?
30. В каком случае игровая модель представляется игрой на квадрате?
31. Что такое “ядро игры”?
32. Как представляются оптимальные стратегии в играх на квадратах?
33. Что вкладывается понятие игр дуэльного типа?
34. В каком случае будет использована дуэль шумного типа, а в каком – бесшумного?
35. Что является побудительным мотивом возникновения кооперации игроков?
36. Когда игра разрешима в смысле Нэша?
37. Что представляет собой совместная смешанная стратегия?
38. Для чего может быть использован вектор Шепли?

ЗАКЛЮЧЕНИЕ

Математизация естественных наук означает формализацию содержательных моделей объективной реальности, которые используются этими науками для описания коллизий, возникающих в практической деятельности, для научного обоснования и принятия наилучших разумных и целесообразных решений. Формализация содержательных моделей является, как принято считать, прикладной функцией различных отраслей математики, которые и объединены в тот неудобопереваримый винегрет, который называется “исследованием операций” и изложению которого посвящён настоящий опус.

В заключение уместно процитировать Устав корпуса морских инженеров России 1910 г. “… Никакая самая совершенная инструкция не в состоянии переписать все возможные обязанности должностного лица, предусмотреть все отдельные случаи, и дать наперёд соответствующие указания. А по сему, господа инженеры должны проявить инициативу и употребить все свои знания и опыт для оправдания своего предназначения…”.

Тем и руководствоваться…

Автор

Библиографический список

1. Айзекс Р. Дифференциальные игры [Текст]: / Р. Айзекс. – М.: Мир, 1967.
2. Акоф Р. Исследование операций [Текст]: / Р. Акоф, П. Райветт. - М.: Мир.-1986. – 230 с.
3. Акулич И.Л. Математическое программирование в примерах и задачах [Текст]: / И.Л. Акулич. - М.: Высшая школа, 1986. – 317 с.
4. Анисимов В.В. Элементы теории массового обслуживания и асимптотический подход [Текст] / В.В. Анисимов, О.К. Закусило, В.С. Донченко. – К. : Вища школа, 1987. – 248 с.
5. Ашманов С.А. Теория оптимизации в задачах и упражнениях [Текст] / С.А. Ашманов, А.В. Тимохов. – СПб.; М.; Краснодар: Лань, 2012. – 448 с.
6. Байхельт Ф. Надёжность и техническое обслуживание. Математический подход [Текст] / Ф. Байхельт, П. Франкен. – М.: Радио и связь, 1988. – 392 с.
7. Балашевич В.А. Основы математического программирования [Текст]: / В.А. Балашевич. - Минск: Высшая школа, 1985. – 174 с.
8. Банди Б. Методы оптимизации: Вводный курс [Текст]: / Б. Банди. - М.: Радио и Связь, 1988. – 128 с.
9. Бейко И.В. Методы и алгоритмы решения задач оптимизации. [Текст]: / И.В. Бейко, Б.Н. Бублик, П.Н. Зинько. - Киев: Вища школа, 1983. - 512 с.
10. Вагнер Г. Основы исследования операций [Текст]: в 3-х томах. / Г. Вагнер. - М.: Мир Т.1: 1972. - 335 с. Т.2: 1973. - 488 с. Т.3: 1973. – 501 с.
11. Васин А.А. Теория игр и модели математической экономики [Текст] / А.А. Васин, В.В. Морозов. – М.: МАКС Пресс, 2005. – 272 с.
12. Введение в нелинейное программирование [Текст]: / Под. ред. К.-Х. Эльстера. – М.: Наука, 1985. – 263 с.
13. Вентцель Е.С. Исследование операций [Текст]: / Е.С. Вентцель. - М.: Наука,1980. – 208 с.
14. Вентцель Е.С. Элементы теории игр [Текст]: / Е.С. Вентцель. - М.: Физматгиз, 1969.
15. Вильямс Дж. Совершенный стратег или букварь по теории стратегических игр / Дж. Вильямс. - М.: Советское радио, 1960.
16. Воробьёв Н.И. Математическая теория игр [Текст]: / Н.И. Воробьёв. – Л.: Знание, 1963.
17. Гаас Е. Линейное программирование [Текст]: / Е. Гаас. - М.: ГФМЛ, 1961. - 309с.

18. Гантмахер Ф.Р. Теория матриц / Ф.Р. Гантмахер, М.: Наука, ГРФМЛ, 1988. – 552 с.
19. Гаррет Р. Основы анализа операций на море [Текст]: Пер. с английского И. Разумного под ред. Б. Балаева / Р. Гаррет, Дж. Лондон. - М.: Воениздат, 1974. - 270с.
20. Гельбранд И.М. Лекции по линейной алгебре [Текст]: / И.М. Гельбранд. - М.: Наука, 1971.
21. Гнеденко Б.В. Введение в теорию массового обслуживания [Текст]: / Б.В. Гнеденко, М.Н. Коваленко. - М.: Наука, 1986. – 400 с.
22. Гольштейн Е.Г. Задачи линейного программирования транспортного типа [Текст]: / Е.Г. Гольштейн, Д.Б. Юдин. - М.: Наука, 1976. – 384 с.
23. Горлач Б.А. Исследование операций [Текст]:/ Б.А. Горлач. – СПб.; М.; Краснодар: Лань, 2013. – 438 с.
24. Грешилов А.А. Математические методы принятия решений [Текст] / А.А. Грешилов, – М.: МГТУ им. И.Э. Баумана, 2006. – 584 с.
25. Губанов В.А. Введение в системный анализ [Текст]:/Т.С. Губанов. - Л.: Издательство ЛГУ, 1988. - 228с.
26. Данциг Дж. Линейное программирование, его обобщение и применение [Текст]: / Дж. Данциг. – М.: Прогресс, 1966. – 600 с.
27. Дегтярев Ю.И. Исследование операций [Текст]: / Ю.И. Дегтярев. - М.: Высшая школа, 1986. -320 с.
28. Денисов А.А. Теория больших систем управления [Текст]: / А.А. Денисов, Д.Н. Колесников - Ленинград: Энергоиздат, Ленинградское отделение, 1982. - 288 с.
29. Деордица Ю.Ф. Исследование операций в планировании управления [Текст]: / Ю.Ф. Деордица, Ю.М., Нефедов. - Киев: Вища школа, 1991. - 196 с.
30. Дрешер М. Стратегические игры [Текст]: / М. Дрешер. - М.: Советское радио, 1964.
31. Ермольев Ю.М. Математические методы исследования операций [Текст] / Ю.М. Ермольев, И.И. Ляшко, В.С. Михалевич, В.И. Тюптя. – К.: Вища школа, 1979. – 319 с.
32. Есипов Б.А. Методы исследования операций [Текст]:/ Б.А. Есипов – СПб.; М.; Краснодар: Лань, 2013. – 304 с.
33. Зайченко Ю.П. Исследование операций. [Текст]: учебное пособие. / Ю.П. Зайченко. -Киев: Вища школа, 1979. -392 с.
34. Зайченко Ю.П. Исследование операций. [Текст]: сборник задач / Ю.П. Зайченко, С.А. Шумилова. - Киев: Вища школа, 1990. - 239 с.
35. Калихман И.Л. Линейная алгебра и программирование [Текст]: / И.Л. Каллихман. – М.: Высшая школа, 1967. – 179 с.

36. Каллихман И.Л. Сборник задач по математическому программированию [Текст]:/ И.Л. Каллихман – М.: Высшая школа, 1975. – 270 с.
37. Карандаев И.С. Решение двойственных задач в оптимальном управлении [Текст]: / И.С. Карандаев. – М.: Статистика, 1976. – 87 с.
38. Карлин С. Математические методы в теории игр, программировании и экономике [Текст]: / С. Карлин. - М.: Мир.- 1964.
39. Карлусов В.Ю. Опыт преподавания базовых понятий игры двух персон с нулевой суммой [Текст]: / В.Ю. Карлусов. Математика, статистика и информационные технологии в экономике, управлении и образовании: материалы V-й Международной науч.-практ. конф. 31 мая 2016 г., г. Тверь. В 2 ч. Ч.2. Информационные технологии. Вопросы преподавания. – Тверь: Твер. гос. ун – т. – С. 151 – 156.
40. Колбин В.В. Специальные методы оптмизации[Текст]:/ В.В. Колбин. – СПб.; М.; Краснодар: Лань, 2014. – 384 с.
41. Конюховский П.В. Математические методы исследования операций [Текст]: / П.В. Конюховский. – СПб. и др.: Питер, 2001. – 192 с.
42. Коффман А. Массовое обслуживание, теория и применение [Текст]: / А. Коффман, Р. Крюон. - М.: Мир, 1965. – 302 с.
43. Коэн Дж. Граничные задачи в теории массового обслуживания [Текст]: / Дж. Коэн, О. Боксма. - М.: Мир.-1987.
44. Кремер Н.Ш. Исследование операций в экономике [Текст]: учеб. пособие для вузов. Под ред. проф. Н.Ш. Кремера / Н.Ш. Кремер и др. — М.: ЮНИТИ, 2002. — 407 с
45. Кюлян В.Р. Математическое программирование [Текст]: В.Р. Кюлян, Е.А. Юнькова, А.Б. Жильцов. – К.: МАУАП, 2000. - 124 с.
46. Кюнци Г.П. Нелинейное программирование [Текст]: / Г.П. Кюнци, В. Крелле. - М.: Советское радио, 1965. - 303 с.
47. Лесин В.В. Основы методов оптимизации [Текст]:/ В.В. Лесин, Ю.П. Лисовец. – СПб.; М.; Краснодар: Лань, 2012. – 342 с.
48. Льюис Р.Д. Игры и решения [Текст]: / Р.Д. Льюис, Х. Райфа. - М.: Иностранная литература, 1961.
49. Мак-Кинси Д. Введение в теорию игр [Текст]: / Д. Мак-Кинси. – М.: Физматгиз, 1960.
50. Моудер Дж. Основы исследования операций. [Текст]: в 2-х томах./ Дж. Моудер, С. Элмаграби. - М.: Мир, 1981. Т.1: - 347 с. Т.2: - 492 с.
51. Мулен Э. Теория игр [Текст]: / Э. Мулен. - М.: Мир, 1985.
52. Нейман фон Дж. Теория игр и экономическое поведение [Текст]: / Дж. Фон Нейман, О. Моргенштерн. – М.: Наука , 1970.
53. Овчаров Л.А. Прикладные задачи теории массового обслуживания [Текст]: / Л.А. Овчаров. – М.: Машиностроение, 1969.

54. Отнер С.Л. Системный анализ для решения деловых и промышленных проблем / С.Л. Отнер. - М.: Советское радио, 1969. – 436 с.
55. Оуэн Г. Теория игр [Текст] / Г. Оуэн. – М.: Мир, 1971.
56. Петросян Л.А. Теория игр [Текст]: / Л.А. Петросян, Н.А. Зенкевич, Е.А. Сёмина. – М.: Высшая школа, 1998.
57. Пробху Н. Статистические процессы теории запасов [Текст]: / Н. Пробху. - М.: Мир, 1985. – 184 с.
58. Прутков Козьма. Сочинения [Текст]: / Козьма Прутков. – М.: Правда, 1986. – 416 с.
59. Ржевский С.В. Исследование операций [Текст]:/ С.В. Ржевский – СПб.; М.; Краснодар: Лань, 2012. – 448 с.
60. Риордан Дж. Вероятностные системы обслуживания [Текст] / Дж. Риордан. – М.: Связь, 1966. – 184 с.
61. Розенберг В.Я. Что такое теория массового обслуживания [Текст]: / В.Я. Розенберг, А.И. Прохоров. - М.: Советское радио, 1982. - 200 с.
62. Романовский И.В. О сведении игры с полной памятью к матричной игре / И.В. Романовский. – М.: ДАН СССР, т. 144, 1962, № 1.
63. Сергиенко И.В. Математические модели и методы решения задач дискретной оптимизации [Текст] / И.В. Сергиенко. – К.: Наукова думка, 1988. – 472.
64. Суздаль В.Г. Теория игр для флота [Текст]: / В.Г Суздаль. - М.: Воениздат, 1974. - 270 с.
65. Таха Х.А Введение в исследование операций [Текст]: / Х. А. Таха. – М.: Издательский дом “Вильямс”, 2001. – 912 с.
66. Феллер В. Введение в теорию вероятностей и её приложения / В. Феллер. – М.: Т. 1, 1967.
67. Фиакко А.В. Нелинейное программирование: Методы последовательной безусловной оптимизации [Текст]:/ А.В. Фиакко, Г.П. Мак-Кормик –М.: Мир, 1972. – 240 с.
68. Химмельблау Д. Прикладное нелинейное программирование [Текст]: / Д. Химмельблау. - М.: Мир, 1975. – 534 с.
69. Чуев Ю.В. Исследование операций в военном деле [Текст]: / Ю.В. Чуев. - М.: Воениздат, 1970. – 256 с.
70. Экономико-математические методы и модели планирования и управления [Текст]: / Под общей редакцией В.Г. Шорина. – М.: Знание, 1973. – 240 с.
71. Юдин Д.Б. Задачи и методы линейного программирования [Текст]: / Д.Б. Юдин, Е.Г. Гольштейн. - М.: Советское радио, 1964. - 736 с.
72. Юдин Д.Б. Линейное программирование (Теория, методы, приложения) [Текст]: / Д.Б. Юдин, Е.Г. Гольштейн. - М.: Наука, 1969. - 424 с.

73. Little J.D.C. A Proof for Queuing Formulae L - λω. – Op. Res. 1961, v. 9, N 3, p. 383-387.

Электронные издания

74. Аттетков А.В. Введение в методы оптимизации [Электронный ресурс]: учебное пособие/ Аттетков А.В., Зарубин В.С., Канатников А.Н.— Электрон. текстовые данные.— М.: Финансы и статистика, 2014.— 272 с.— Режим доступа: http://www.iprbookshop.ru/18794
75. Грызина Н.Ю. Математические методы исследования операций в экономике [Электронный ресурс]: учебное пособие/ Грызина Н.Ю., Мастяева И.Н., Семенихина О.Н.— Электрон. текстовые данные.— М.: Евразийский открытый институт, 2009.— 196 с.— Режим доступа: http://www.iprbookshop.ru/10773 .— ЭБС «IPRbooks»
76. Исследование операций в экономике [Электронный ресурс]: учебное пособие/ Г.Я. Горбовцов [и др.].— Электрон. текстовые данные.— М.: Евразийский открытый институт, Московский государственный университет экономики, статистики и информатики, 2006.— 118 с.— Режим доступа: http://www.iprbookshop.ru/10690 .— ЭБС «IPRbooks».
77. Карманов В.Г. Математическое программирование [Электронный ресурс]/ Карманов В.Г.— Электрон. текстовые данные.— М.: ФИЗМАТЛИТ, 2011.— 264 с.— Режим доступа: http://www.iprbookshop.ru/17324
78. Колобашкина Л.В. Основы теории игр [Электронный ресурс]: учебное пособие/ Колобашкина Л.В.— Электрон. текстовые данные.— М.: БИНОМ. Лаборатория знаний, 2013.— 164 с.— Режим доступа: http://www.iprbookshop.ru/6541
79. Короткин, А. А. Модели и алгоритмы исследования операций : учеб. пособие / В. Г. Фокин, Яросл. гос. ун-т, А. А. Короткин .— Ярославль : ЯрГУ, 2006 http://rucont.ru/efd/200087?cldren=0
80. Костевич Л.С. Исследование операций. Теория игр [Электронный ресурс]: учебное пособие/ Костевич Л.С., Лапко А.А.— Электрон. текстовые данные.— Минск: Вышэйшая школа, 2008.— 368 с.— Режим доступа: http://www.iprbookshop.ru/20076
81. Линейное программирование. Дискретная математика. Теория вероятностей и математическая статистика : метод. указания / В. С. Альпина, Казан. гос. технол. ун-т .— Казань : КГТУ, 2009 http://rucont.ru/efd/292520?cldren=0
82. Лунгу К.Н. Линейное программирование. Руководство к решению задач [Электронный ресурс]/ Лунгу К.Н.— Электрон. текстовые данные.— М.: ФИЗМАТЛИТ, 2009.— 132 с.— Режим доступа: http://www.iprbookshop.ru/12905

83. Методы оптимизации [Электронный ресурс]: учебное пособие/ О.А. Васильева [и др.].— Электрон. текстовые данные.— М.: Московский государственный строительный университет, ЭБС АСВ, 2014.— 96 с.— Режим доступа: http://www.iprbookshop.ru/26859
84. Минько Э.В. Методы прогнозирования и исследования операций [Электронный ресурс]: учебное пособие/ Минько Э.В., Минько А.Э.— Электрон. текстовые данные.— М.: Финансы и статистика, 2012.— 480 с.— Режим доступа: http://www.iprbookshop.ru/18821
85. Пантелеев А.В. Методы оптимизации [Электронный ресурс]: учебное пособие/ Пантелеев А.В., Летова Т.А.— Электрон. текстовые данные.— М.: Логос, 2011.— 424 с.— Режим доступа: http://www.iprbookshop.ru/9093
86. Прокофьева С.И. Основы теории игр [Электронный ресурс]: учебное пособие/ Прокофьева С.И., Пак Э.Е., Ершов Е.К.— Электрон. текстовые данные.— СПб.: Санкт-Петербургский государственный архитектурно-строительный университет, ЭБС АСВ, 2014.— 64 с.— Режим доступа: http://www.iprbookshop.ru/30011
87. Розова В.Н. Методы оптимизации [Электронный ресурс]: учебное пособие/ Розова В.Н., Максимова И.С.— Электрон. текстовые данные.— М.: Российский университет дружбы народов, 2010.— 112 с.— Режим доступа: http://www.iprbookshop.ru/11536
88. Салмина Н.Ю. Теория игр [Электронный ресурс]: учебное пособие/ Салмина Н.Ю.— Электрон. текстовые данные.— Томск: Эль Контент, Томский государственный университет систем управления и радиоэлектроники, 2012.— 92 с.— Режим доступа: http://www.iprbookshop.ru/13870
89. Соколов А.В. Методы оптимальных решений. Общие положения. Математическое программирование. Том 1 [Электронный ресурс]/ Соколов А.В., Токарев В.В.— Электрон. текстовые данные.— М.: ФИЗМАТЛИТ, 2011.— 562 с.— Режим доступа: http://www.iprbookshop.ru/12922
90. Струченков В.И. Методы оптимизации в прикладных задачах [Электронный ресурс]: учебное пособие/ Струченков В.И.— Электрон. текстовые данные.— М.: СОЛОН-ПРЕСС, 2009.— 315 с.— Режим доступа: http://www.iprbookshop.ru/8722
91. Сухарев А.Г. Курс методов оптимизации [Электронный ресурс]: учебное пособие/ Сухарев А.Г., Тимохов А.В., Федоров В.В.— Электрон. текстовые данные.— М.: ФИЗМАТЛИТ, 2011.— 367 с.— Режим доступа: http://www.iprbookshop.ru/17283

ПРИЛОЖЕНИЕ А
(обязательное)
ПОСТОПТИМАЛЬНЫЙ АНАЛИЗ

Постоптимальный анализ ещё называется исследованием на чувствительность. Целью проведения такого анализа является изучение влияния изменения отдельных параметров модели на оптимальное решение, получаемое при статических условиях. Очевидно, что такие изменения способны как “улучшать” так и “ухудшать” оптимальное решение.

В первом случае, целесообразно изменить условия задачи, рекомендовать руководителю, ответственному за организацию производственного процесса, соответствующим образом “подогнать” структуру производства под изменения с целью получения “более оптимального” решения. Подобный подход позволяет адаптировать математическую модель к реальным процессам.

Дополнительным фактором, понуждающим к проведению такого анализа, является наличие погрешностей вычислений, которые обусловлены особенностями представления и обработки чисел в ЭВМ.

Задачи, связанные с анализом моделей на чувствительность, изложены в работах [8, 29, 31, 71, 75, 76].

В ходе постоптимального анализа, как правило, необходимо рассмотреть влияние на устойчивость модели, которое окажут изменения правых частей системы ограничений и вектора коэффициентов целевой функции.

А.1. Пример содержательной постановки задачи

Артель “Босой подбородок” специализируется на производстве брутальных кремов для бритья (“Shaving cream for men”): “Compass” и “Aggression”. Продукция обоих видов поступает в бытовую продажу. Для производства кремов, помимо прочего, используются компоненты: “Stearic acid”, “Myristic acid” и “Potassium Hydroxide”.

Максимально возможные суточные запасы этих компонентов составляют 4, 6, 7 кГ соответственно. Расходы этих ингредиентов на одну тонну соответствующих кремов приведены в таблице 1.1.

Таблица А.1 — Параметры производства

Компоненты	Расход ингредиентов (кг / т)		Запас (кг)
	Compass	Aggression	
Stearic acid	2	1	4
Myristic acid	3	5	6
Potassium Hydroxide	1	5	7

Оптовые цены одной тонны косметики равны между собой и составляют 1 тыс. экю.

Какова должна быть структура производства, то есть, какое количество Shaving cream for men каждого вида должна производить артель “Босой подбородок” за смену, чтобы доход от реализации продукции был максимальным?

По этому условию может быть выполнена постановка задачи в терминах линейного программирования. Определить план $X = (x_1, x_2)$, обеспечивающий оптимальное решение для следующей математической модели:

$$F(x_1, x_2) = 1 \cdot x_1 + 1 \cdot x_2 \to \max, \tag{A.1}$$

при ограничениях

$$\begin{cases} 2 \cdot x_1 + 1 \cdot x_2 \le 4, \\ 1 \cdot x_1 + 5 \cdot x_2 \le 6, \\ 3 \cdot x_1 + 5 \cdot x_2 \le 7, \\ x_1 \ge 0, x_2 \ge 0. \end{cases} \tag{A.2}$$

А.2. Результаты решения задачи линейного программирования

Ход решения задачи симплекс-методом приведём без будь-каких комментариев, поскольку данные методы достаточно подробно освещены как в специальной литературе [2, 3, 7, 10, 13,17, 23, 24, 27, 31 – 37, 41, 44, 45, 39, 59, 60, 71, 72, 79 – 82, 84], так и настоящем пособии. Помимо этого у читателя есть опыт практического применения указанных методов, приобретённый в ходе выполнения лабораторных (дневная форма обучения) либо контрольных (заочная форма обучения) работ.

Таблица А.2 – Начальная симплекс-таблица

Базис	$C_Б$	C_j	1	1	0	0	0
		A_0	A_1	A_2	A_3	A_4	A_5
←A_3	0	4	2	1	1	0	0
A_4	0	6	1	5	0	1	0
A_5	0	7	3	5	0	0	1
	δ	0	– 1↑	– 1	0	0	0

Таблица А.3 – Симплекс-таблица после первой итерации

Базис	$C_Б$	C_j	1	1	0	0	0
		A_0	A_1	A_2	A_3	A_4	A_5
A_1	1	2	1	1/2	1/2	0	0
A_4	0	4	0	9/2	– 1/2	1	0
←A_5	1	1	0	7/2	– 3/2	0	1
	δ	2	0	– 1/2↑	1/2	0	0

Окончательно приведём оптимальное решение задачи, выполненное вручную с обычными дробями (таблица А4) и компьютерное, округлённое с десятичными дробями (таблица А.5). Точность обоих результатов предоставим читателю сравнить самостоятельно.

Таблица А.4 – Результат оптимального решения вручную

Базис	$C_Б$	C_j	1	1	0	0	0
		A_0	A_1	A_2	A_3	A_4	A_5
A_1	1	13/7	1	0	5/7	0	– 1/7
A_4	0	19/7	0	0	10/7	1	– 9/7
A_2	1	2/7	0	1	– 3/7	0	2/7
	δ	15/7	0	0	2/7	0	1/7

Таблица А.5 – Симплекс-таблица решения, выполненного на ЭВМ

Базис	$C_Б$	C_j	1	1	0	0	0
		A_0	A_1	A_2	A_3	A_4	A_5
A_1	1	1,86	1	0	0,71	0	– 0,14
A_4	0	2,71	0	0	1,43	1	–1,286
A_2	1	0,286	0	1	– 0,43	0	0,286
	δ	2,143	0	0	0,286	0	0,143

В результате решения получен следующий оптимальный план производства, в тоннах:

$$X_{opt}^{T} = \left[\frac{13}{7} \quad \frac{2}{7}\right] \text{ или } X_{opt}^{T} = [1{,}86 \quad 0{,}29]/$$

при этом максимальное значение функции составит, в тыс. экю:

$$X_{\max} = \frac{15}{7} \cong 2{,}143.$$

А.3. Пример выполнения постоптимального анализа

Ниже будут рассмотрены влияние на устойчивость оптимального решения, которое окажут вариации правых частей системы ограничений и вектора коэффициентов целевой функции.

А.3.1. Определение устойчивости оптимального решения к изменениям элементов вектора ограничений

В ходе анализа модели на чувствительность к изменениям правых частей ограничений (измерению объемов ресурсов) необходимо дать ответы на следующие вопросы.

Каков статус ресурсов, какие из них являются “дефицитными”, а какие “недефицитными”?

Какова значимость “дефицитных” ресурсов, то есть, изменение объёма какого из ресурсов является наиболее выгодным, с точки зрения обеспечения наибольшего дохода (или наименьших потерь) для производства?

В каких пределах допустимо изменение запаса ресурсов, при которых их влияние на исходную модель задачи линейного программирования адекватно описывается двойственной задачей?

Как отразится на оптимальном плане увеличение (уменьшение) запаса ресурсов?

Проведение анализа модели на чувствительность связанно с использованием теории двойственности [22, 33 – 37]. Поэтому в курсовой работе необходимо построить двойственную задачу и найти ее решение.

Для рассматриваемой нами прямой задачи (А.1) и (А.2), по формальным правилам [33], можно получить модель соответствующей двойственной задачи:

$$Z(y_1, y_2, y_3) = 4 \cdot y_1 + 6 \cdot y_2 + 7 \cdot y_3 \to \min, \qquad \text{(А.3)}$$

$$\begin{cases} 2 \cdot y_1 + 1 \cdot y_2 + 3 \cdot y_3 \geq 1; \\ 1 \cdot y_1 + 5 \cdot y_2 + 5 \cdot y_3 \geq 1; \\ y_1 \geq 0, y_2 \geq 0, y_3 \geq 0. \end{cases} \qquad \text{(А.4)}$$

Оптимальное решение двойственной задачи, согласно известным теоремам [33] и следствиям из них, исходя из таблицы А.5, есть

$$Y_{\min}^T = [0{,}286 \quad 0{,}000 \quad 0{,}143].$$

Значение функции цели при этом составит $Z(Y_{\min}) = 2{,}143$.

Определение статуса ресурсов.

Ресурсы относятся к дефицитным, если оптимальный план предусматривает их полное использование, а при частичном использовании ресурсов они считаются недефицитными.

Статус ресурсов для любой модели линейного программирования можно установить непосредственно из оптимальной симплекс-таблицы исходной (прямой) задачи по значению дополнительных переменных.

Положительное значение дополнительной переменной указывает на ***неполное*** использование дополнительного ресурса, на его “недефицитность”.

Напротив, ***нулевое*** значение дополнительной переменной указывает на “дефицитность” ресурса, его ***полное*** использование.

В рассматриваемом нами примере дополнительные переменные x_3 и x_5 равны нулю, следовательно, исходные продукты Stearic acid и Potassium Hydroxide являются “дефицитными”, а продукт Myristic acid — “недефицитным” ($x_4 = 2{,}714$).

Аналогичный вывод можно сделать и из анализа оптимального решения двойственной задачи. Так как продукту Myristic acid соответствует нулевая оценка (y_2=0), он является “недефицитным”, а продукты Stearic acid и Potassium Hydroxide — “дефицитными”, поскольку соответствующие оценки положительны ($y_1 = 0{,}286$; $y_3 = 0{,}143$).

Определение значимости ресурсов.

Значимость ресурса можно охарактеризовать величиной улучшения оптимального значения целевой функции $F(x_1, x_2)$ прямой задачи, приходящейся на единицу прироста данного ресурса.

Как следует из теоремы о чувствительности [72], значимость ресурса всегда можно определить по значению двойственных переменных в

оптимальном решении двойственной задачи. Изменение свободного члена *i*-го ограничения исходной задачи на величину δ_{b_i} вызывает изменение целевой функции на величину $\Delta F = \delta_{b_i} \cdot y_i$, где y_i - двойственная оценка *i*-го ресурса. Если приращения запасов ресурса берутся из диапазона допустимых изменений, то изменение целевой функции будет линейно зависеть от изменений запасов ресурса, и при этом структура оптимального плана (то есть множество переменных совокупно с соответствующими базисными векторами) не будет меняться.

Таким образом, б**О**льшее значение двойственной переменной соответствует б**О**лее значимому ресурсу.

В данном примере (y_1 = 0,286; y_2 = 0,0; y_3 =0,143). Таким образом, из двух дефицитных ресурсов продукт Stearic acid имеет большую значимость, нежели Potassium Hydroxide, и увеличение объемов запасов этого продукта более выгодно с точки зрения влияния на значение целевой функции.

Действительно, если объем продукта Stearic acid увеличивается на $\delta_{b_1} = 1$, то целевая функция увеличится на величину $\Delta F = 1 \cdot y_1 = 0{,}286$.

А при увеличении объема запасов продукта Potassium Hydroxide на $\delta_{b_3} = 1$, то целевая функция увеличится только на величину $\Delta F = 1 \cdot y_3 = 0{,}143$.

Определение допустимых интервалов изменения запасов ресурсов.

Произвольное изменение запасов ресурсов (то есть правых частей ограничений) может привести к недопустимости текущего решения. Поэтому важно определить диапазон изменений компонент вектора ограничений, в котором допустимость решений не нарушается.

Пусть найдено оптимальное решение некоторой задачи линейного программирования $X^T_{opt} = [x_1 \quad \ldots \quad x_m \quad 0 \quad \ldots \quad 0] \equiv \left\| a^*_{i,0} \right\|$, $i = \overline{1,m}$, где $a^*_{i,0}$ — компоненты оптимального решения задачи, расположенные в столбце A_0. Для того, чтобы приращение левой части *r*-го ограничения δ_{b_r} не нарушало допустимости этого оптимального решения, необходимо, чтобы выполнялась система неравенств

$$a^*_{i,0} + \delta_{b_r} \cdot a_{i,n+r} \geq 0, \; i = \overline{1,m}, \qquad \text{(A.5)}$$

где $a_{i,n+r}$ — компоненты столбца A_{n+r} таблицы оптимального решения.

Определим диапазон изменения δ_{b_1} для более значимого ресурса Stearic acid. Согласно (A.5) имеем систему неравенств

$$\begin{cases} 1{,}86 + 0{,}71 \cdot \delta_{b_1} \geq 0, \\ 2{,}71 + 1{,}43 \cdot \delta_{b_1} \geq 0, \\ 0{,}286 - 0{,}43 \cdot \delta_{b_1} \geq 0, \end{cases} \Rightarrow \begin{cases} \delta_{b_1} \geq \dfrac{1{,}86}{0{,}71} \cong -2{,}6; \\ \delta_{b_1} \geq \dfrac{2{,}71}{1{,}43} \cong -1{,}9; \\ \delta_{b_1} \leq \dfrac{0{,}286}{0{,}43} \cong 0{,}66. \end{cases}$$

Обратите внимание, что в качестве коэффициентов при неизвестных использованы компоненты столбца A_3 таблицы А.5. В результате решения системы неравенств, получаем диапазон $-1{,}9 \leq \delta_{b_1} \leq 0{,}66$, откуда следует, что допустимый интервал изменения ресурса $2{,}1 \leq b_1 \leq 4{,}66$.

Таким образом, первоначальный запас продукта Stearic acid (4 кг) может быть увеличен до 4,66 кГ или уменьшен до 2,1 кГ без нарушения допустимости решения.

Однако уменьшение запаса ресурса, в данном случае, повлечет за собой уменьшение прибыли от реализации продукции, поэтому является нецелесообразным.

Аналогичные исследования для ресурса Potassium Hydroxide на основании системы (А.5), используя компоненты столбца A_5, таблицы 5 приведут к следующему:

$$\begin{cases} 1{,}86 - 0{,}14 \cdot \delta_{b_3} \geq 0, \\ 2{,}71 - 1{,}286 \cdot \delta_{b_3} \geq 0, \\ 0{,}286 + 0{,}286 \cdot \delta_{b_3} \geq 0, \end{cases} \Rightarrow \begin{cases} \delta_{b_3} \leq \dfrac{1{,}86}{0{,}14} \cong 13{,}285; \\ \delta_{b_3} \leq \dfrac{2{,}71}{1{,}286} \cong 2{,}107; \\ \delta_{b_3} \geq \dfrac{-0{,}286}{0{,}286} = -1{,}0. \end{cases}$$

Откуда $-1{,}0 \leq \delta_{b_3} \leq 2{,}107$, а допустимый интервал изменения ресурса Potassium Hydroxide есть $6 \leq b_3 \leq 9{,}107$. Как и для продукта Stearic acid, уменьшение ресурса Potassium Hydroxide нецелесообразно.

Если приращения вводятся сразу для нескольких (а в самом общем случае – для всех) ограничений, то в этом случае вместо системы (5) будет её обобщение вида

$$a^*_{i,0} + \sum_{r=1}^{m} \delta_{b_r} \cdot a_{i,n+r} \geq 0, \; i = \overline{1,m}. \qquad \text{(А.6)}$$

Система (А.6) для оптимального решения, представленного таблицей А.5, будет выглядеть так

$$\begin{cases} 1{,}86 + 0{,}71 \cdot \delta_{b_1} - 0{,}14 \cdot \delta_{b_3} \geq 0; \\ 2{,}71 + 1{,}43 \cdot \delta_{b_1} - 1{,}286 \cdot \delta_{b_3} \geq 0; \\ 0{,}286 - 0{,}43 \cdot \delta_{b_1} + 0{,}286 \cdot \delta_{b_3} \geq 0; \\ \delta_{b_1} \geq 0; \\ \delta_{b_3} \geq 0. \end{cases} \qquad \text{(А.7)}$$

Последние два ограничения неотрицательности формально в запись (А.6) не входят. Но ранее мы убедились, что отрицательные приращения дефицитных ресурсов (уменьшение их) нецелесообразны. Система (А.7), в данном случае, может быть легко решена графоаналитически (рисунок А.1, без учёта ограничений неотрицательности).

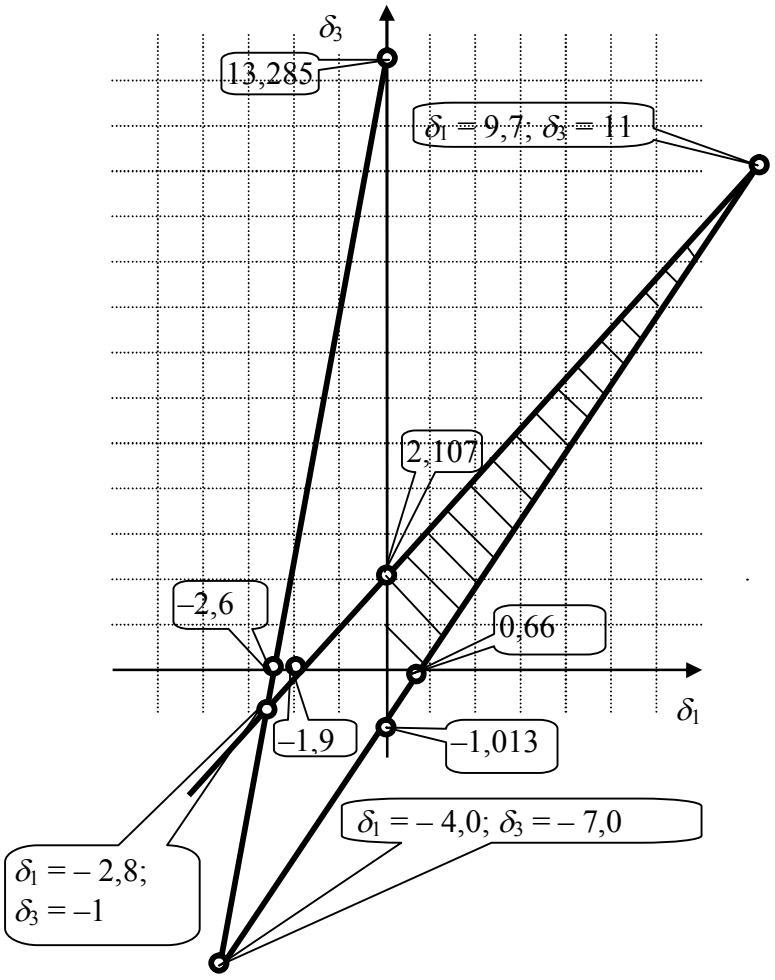


Рисунок А.1 – Решение системы ограничений (А.7)

В результате решения, мы получим область, в пределах которой ***одновременное изменение*** приращений дефицитных ресурсов ***не сказывается*** на структуре оптимального решения.

Ранее нами были получены ограничения на изменения одного из дефицитных ресурсов при неизменности другого:

$$\begin{cases} 0 \le \delta_{b_1} \le 0{,}66; \\ 0 \le \delta_{b_3} \le 2{,}107. \end{cases} \qquad \text{(A.8)}$$

Несложно показать, что усиление системы (A.7) добавлением к ним условий (A.6) приведёт к тому, что область одновременного изменения дефицитных ресурсов будет представлять собой прямоугольник, ограниченный осями координат δ_1, δ_3 и прямыми $\delta_1 = 0{,}66$, $\delta_3 = 2{,}107$. Заметим, что это решение не противоречит системе ограничений (A.6).

Таким образом, исследование устойчивости, проводимое по каждому дефицитному ресурсу в отдельности, позволяет построить прямоугольник или параллелепипед (в перспективе, *n*-мерный гиперпараллелепипед), в пределах которого можно варьировать приращения всех дефицитных ресурсов одновременно без риска выхода за пределы допустимости текущего оптимального решения.

Взаимозаменяемость дефицитных ресурсов.

Часто возникает задача о замене одного дефицитного ресурса другим, так, чтобы функция цели при этом не менялась. Очевидно, что в этом случае приращения для дефицитных ресурсов должны соотноситься как

$$\delta_{b_q} = \frac{\delta^*_{n+s}}{\delta^*_{n+q}} \cdot \delta_{b_s} \qquad \text{(A.9)}$$

или

$$\delta_{b_s} = \frac{\delta^*_{n+q}}{\delta^*_{n+s}} \cdot \delta_{b_q}. \qquad \text{(A.10)}$$

где δ_{b_q} и δ_{b_s} — приращение пары дефицитных ресурсов q и s соответственно, δ^*_{n+q} и δ^*_{n+s} — значения симплекс-разностей, соответствующих этим ресурсам в таблице оптимального решения.

Применительно к таблицам 4 и 5 замена дефицитного сырья Stearic acid на Potassium Hydroxide должна проводиться, в соответствии с (7) в пропорции

$$\delta_{b_1} = \frac{0{,}143}{0{,}286} \cdot \delta_{b_3} = 0{,}5,$$

а замена компонента Potassium Hydroxide на Stearic acid в пропорции

$$\delta_{b_3} = \frac{0{,}286}{0{,}143} \cdot \delta_{b_1} = 2{,}0\,.$$

Исследование зависимости оптимального решения от изменений запасов ресурсов.

Из математики известно, что линейная зависимость может быть представлена двумя точками в пространстве. Однако погрешность расчётов ведёт к искажению местоположения отдельных точек зависимости. Поэтому расчёты целесообразно проводить по точкам. Будем изменять запасы ресурса Stearic acid, начиная с 4 кг (начальный запас) с шагом h = 0,1 кг до 4,66 кг. Округленные результаты расчётов приведены в таблице А.6.

Таблица А.6 – Расчётные точки прямой

δ_{b_1}, [кг]	0	0,1	0,2	0,3	0,4	0,5	0,6	0,66
b_1, [кг]	4	4,1	4,2	4,3	4,4	4,5	4,6	4,66
ΔF, [тыс. экю]	0	0,03	0,06	0,09	0,11	0,14	0,17	0,19
F, [тыс. экю]	2,14	2,17	2,2	2,23	2,25	2,28	2,31	2,33

Для определения оптимальных решений, соответствующих ряду ограничений, приведённому во второй строке таблицы 6 (то есть оптимального плана производства кремов), можно решить задачу на ЭВМ с новыми значениями свободных членов ограничений, либо воспользоваться известным выражением

$$X' = A^{-1} \times B'\,,$$

где X' – вектор оптимального решения, A^{-1} – обратная матрица, составленная из столбцов оптимальной симплекс-таблицы для дополнительных переменных, B' – новый вектор правых частей ограничений. В частности, для рассматриваемого примера:

$$X' = \begin{bmatrix} x_1' \\ x_4' \\ x_2' \end{bmatrix} = \begin{bmatrix} 1{,}86 & 0 & 0{,}71 \\ 2{,}71 & 1 & 1{,}43 \\ 0{,}286 & 0 & -0{,}43 \end{bmatrix} \times \begin{bmatrix} 4 + \delta_{b_1} \\ 6 \\ 7 \end{bmatrix} = \begin{bmatrix} 1{,}86 + 0{,}71 \cdot \delta_{b_1} \\ 2{,}71 + 1{,}43 \cdot \delta_{b_1} \\ 0{,}286 - 0{,}43 \cdot \delta_{b_1} \end{bmatrix}.$$

Ниже, в таблице А.7, приводятся данные, соответствующие изменениям запаса добавки Stearic acid, с 4 кг до 4,66 кг, при шаге 0,1 кг.

Таблица А.7 — Данные об оптимальном выпуске кремов для бритья в зависимости от объёмов запаса добавки Stearic acid

b, [кг]		4	4,1	4,2	4,3	4,4	4,5	4,6
X'_{opt} [т] =	x'_1	1,86	1,93	2,0	2,07	2,14	2,20	2,28
	x'_2	0,28	0,24	0,2	0,16	0,11	0,07	0,03
F, [тыс. экю]		2,14	2,17	2,2	2,23	2,25	2,28	2,31

Проведённые расчёты позволяют сделать вывод, что с увеличением объема запасов добавки Stearic acid (на 0,6 кг) увеличивается выпуск крема “Compass” на 0,42 т, и уменьшается выпуск крема для бритья “Aggression” на 0,25 т. При этом общая прибыль от реализации продукции увеличивается на 170 экю.

А.3.2. Определение устойчивости оптимального решения к изменениям элементов вектора коэффициентов целевой функции

Вызывает интерес определение границ изменения коэффициентов функции цели, при которых структура оптимального плана остаётся без изменений. В экономической интерпретации это определение возможных изменений расценок (себестоимости или прибыли) по каждому типу выпускаемой продукции при неизменной структуре производства [36, 59].

Условию неизменности оптимального решения отвечает сохранение знаков симплекс-разностей в таблице, соответствующей этому решению. Указанное условие, в случае, когда задача решается на максимум, формулируется в виде системы неравенств

$$\delta_j + \sum_{r=1}^{m} a_{r,j} \cdot \delta_{C_r} \geq 0, \qquad j = \overline{1, n+m}, \tag{А.8}$$

в которых δ_j — симплекс-разности, обеспечивающие остановку алгоритма в точке достижения оптимума; δ_{C_r} — приращение коэффициента функции цели при переменной, находящейся в оптимальном решении в r-й строке симплекс-таблицы.

Для симплекс-таблицы А.5, согласно (А.8), можно записать такую систему:

$$\begin{cases} 0{,}286 + 0{,}71 \cdot \delta_{C_1} - 0{,}43 \cdot \delta_{C_2} \geq 0, \\ 0{,}143 - 0{,}14 \cdot \delta_{C_1} + 0{,}286 \cdot \delta_{C_2} \geq 0. \end{cases} \tag{А.9}$$

Для случая, когда изменяется первый коэффициент функции цели, система (9) примет вид

$$\begin{cases} 0{,}286 + 0{,}71 \cdot \delta_{C_1} \geq 0, \\ 0{,}143 - 0{,}14 \cdot \delta_{C_1} \geq 0, \end{cases} \tag{A.10}$$

а когда изменяется второй её коэффициент то

$$\begin{cases} 0{,}286 - 0{,}43 \cdot \delta_{C_2} \geq 0, \\ 0{,}143 + 0{,}286 \cdot \delta_{C_2} \geq 0. \end{cases} \tag{A.11}$$

Решение систем ограничений (10) и (11) даст следующие диапазоны приращений

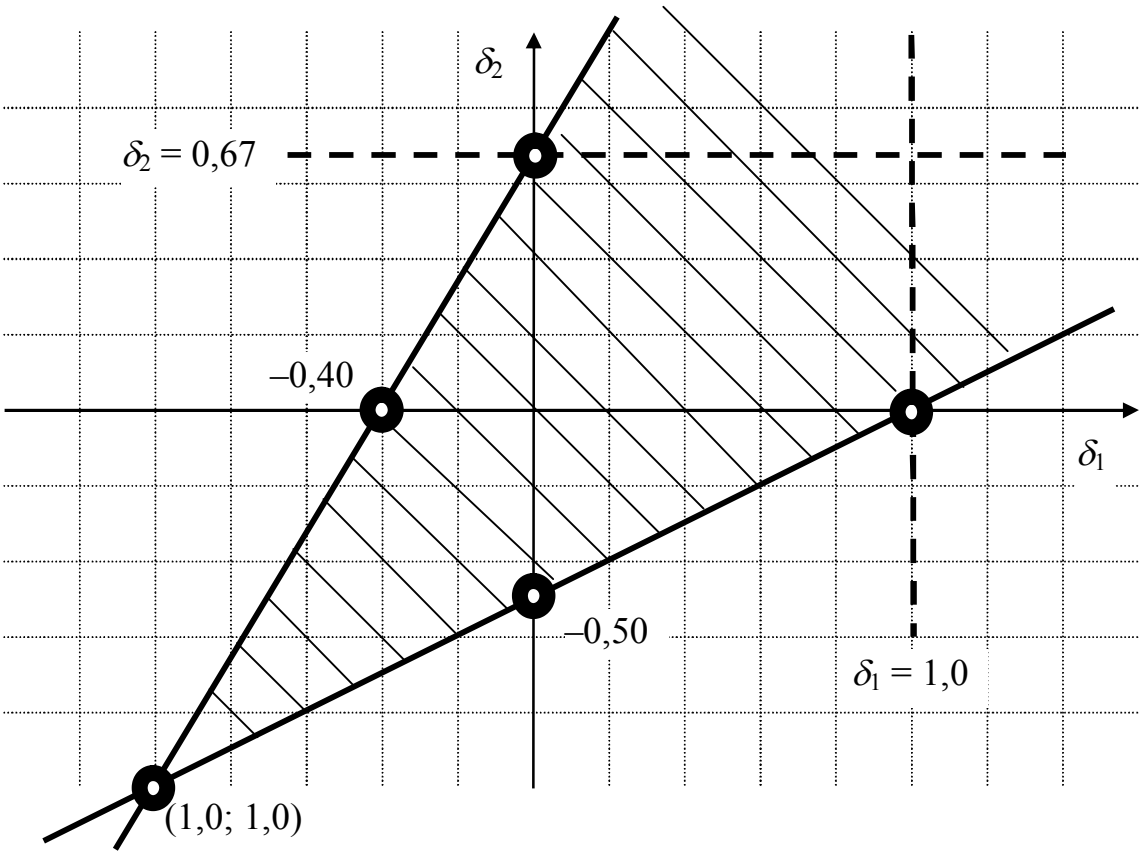


Рисунок А.2 – Область устойчивости модели по функции цели

$\begin{cases} -0{,}6 \le \delta_{c_1} \le 1{,}0; \\ -0{,}5 \le \delta_{c_2} \le 0{,}67 \end{cases}$ или с учётом не убывания функции цели $\begin{cases} 0 \le \delta_{c_1} \le 1{,}0; \\ 0 \le \delta_{c_2} \le 0{,}67. \end{cases}$

Решение (А.9) представим в графоаналитическом виде на рисунке А.2 выше.

Пунктиром показаны верхние границы областей допустимых решений систем неравенств (А.10) и (А.11). Таким образом, при величинах изменений коэффициентов функции цели, находящихся между двумя прямыми, структура оптимального решения задачи меняться не будет.

По результатам проведённых исследований можно порекомендовать руководству артели при разработке ценовой политики артели “Босой подбородок” удерживать колебания цен на кремы “Compass” и “Aggression” в рамках коридора, определяемого заштрихованной областью, как бы ни был велик соблазн выйти за её пределы. Получаемая прибыль может быть рассчитана по формуле (А.1) после введения соответствующих корректив в коэффициенты функции цели.

ЗАКЛЮЧЕНИЕ

Необходимо отметить, что построение детерминированных математических моделей лишь отчасти отвечает производственно-экономическим задачам. Строго говоря, на практике имеют место, в лучшем случае, параметрические зависимости в моделях. Подавляющее же большинство реальных ситуаций потребуют для моделирования учёта значительного числа параметров, с одной стороны, и учёта вероятностного характера их проявления и взаимодействия с другой.

Известно, что одним из практических подходов к решению задач большой размерности и задач стохастического программирования является их декомпозиция и, в конечном итоге, сведение, с учётом различных допущений, к задачам линейного программирования и их решению, чему, собственно и посвящено приложение.

У ч е б н о е и з д а н и е

Карлусов Вадим Юрьевич

ИССЛЕДОВАНИЕ ОПЕРАЦИЙ И МЕТОДЫ ОПТИМИЗАЦИИ

Учебное пособие

Издается в авторской редакции

Изд. № 204/19. Объем 19,75 п.л.
РИИЦМ ФГАОУ ВО «Севастопольский государственный университет»