



**Федеральное государственное образовательное бюджетное учреждение
высшего образования**

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ
РОССИЙСКОЙ ФЕДЕРАЦИИ»
(Финансовый университет)**

**Факультет информационных технологий и анализа больших данных
Департамент анализа данных и машинного обучения**

Г.В. Моисеев

**УЧЕБНОЕ ПОСОБИЕ ПО ДИСЦИПЛИНЕ
«Математические методы принятия решений»**

Для студентов,
обучающихся по направлению
38.03.05 «Бизнес-информатика»
(программа подготовки бакалавра)

Москва 2021

**Федеральное государственное образовательное бюджетное учреждение
высшего образования**

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ
РОССИЙСКОЙ ФЕДЕРАЦИИ»
(Финансовый университет)**

**Факультет информационных технологий и анализа больших данных
Департамент анализа данных и машинного обучения**

Г.В. Моисеев

**УЧЕБНОЕ ПОСОБИЕ ПО ДИСЦИПЛИНЕ
«Математические методы принятия решений»**

Для студентов,
обучающихся по направлению
38.03.05 «Бизнес-информатика»
(программа подготовки бакалавра)

*Одобрено Советом Департамента анализа данных и машинного обучения
(протокол №4 от 27.10.2021 г.)*

Москва – 2021

УДК 51(07)

ББК 22.18

M74

Автор:

Моисеев Г.В., канд. техн. наук, доцент Департамента анализа данных и машинного обучения факультета информационных технологий и анализа больших данных Финансового университета при Правительстве РФ.

Рецензент:

Золотова Т.В., доктор физ.-мат. наук, профессор Департамента анализа данных и машинного обучения факультета информационных технологий и анализа больших данных Финансового университета при Правительстве РФ.

Математические методы принятия решений Учебное пособие для организации самостоятельной работы студентов при прохождении дисциплины «Математические методы принятия решений» в 4,5 семестрах. - М.: Финансовый университет, департамент анализа данных и машинного обучения, 2021. - 224 с.

Дисциплина «Математические методы принятия решений» является обязательной дисциплиной базовой части профессионального цикла направления 38.03.05 «Бизнес-информатика». Изучение данной дисциплины нацелено на формирование у студентов теоретических сведений по применению математических методов в прикладных задачах принятия решений. В учебном пособии представлен материал лекций, раскрывающий в простой и доступной форме различные разделы прикладной математики из курса теории принятия решений для студентов непрофильных математических специальностей.

УДК 51(07)

ББК 22.18

Учебное издание

Моисеев Георгий Викторович

Учебное пособие по дисциплине «Математические методы принятия решений»

Для студентов, обучающихся по направлению 38.03.05 «Бизнес-информатика»,
(программа подготовки бакалавра)

Компьютерный набор, верстка Д.А. Чистякова

Формат 60x90/16. Гарнитура *Times New Roman*.
Усл. п.л. 13,37. Изд. № - 2021. Заказ № _____
Электронное издание

© ФГОБУ ВО «Финансовый университет при Правительстве Российской Федерации», 2021
© Моисеев Георгий Викторович, 2021

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
І МАТЕМАТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ	5
1. Производственная задача, симплекс-метод.....	5
2. Классическая транспортная задача	18
3. Задача о назначениях	30
4. Задача о рюкзаке	35
5. Задача об оптимальном портфеле	45
ІІ ТЕОРИЯ ГРАФОВ.....	49
6. Задача о кратчайшем пути.....	52
7. Задача коммивояжера	60
8. Задача о максимальном потоке.....	66
9. Задача о минимальном разрезе	75
10. Задача о паросочетаниях	82
11. Задача о выравнивании ресурсов проекта	91
ІІІ ТЕОРИЯ ИГР	100
12. Антагонистические игры.....	102
13. Биматричные игры	112
14. Игры с «природой»	118
15. Позиционные игры с противником	128
16. Позиционные игры с «природой»	140
ІV ДОПОЛНИТЕЛЬНЫЕ ГЛАВЫ	148
17. Регрессионное прогнозирование	148
18. Многокритериальная оптимизация	165
19. Экспертные решения	187
20. Системы массового обслуживания	204
ЗАКЛЮЧЕНИЕ	221

БЛАГОДАРНОСТИ

Данное пособие является результатом колективного труда группы студентов: Анастасии Пантелеимоновой, Полины Термышевой, Анны Чирвоной, Натальи Гушан, Никиты Комлева, Елизаветы Красотской, Анелии Клевлеевой, Дмитрия Бахтиграеева, Марии Андреевой, Марии Ксенофонтовой и Лидии Василевской. Особые труды были приложены Дарьей Чистяковой по доведению материала до окончательного вида.

ВВЕДЕНИЕ

Теория принятия решений является наследницей исследования операций, которая появилась во время Второй мировой войны, когда союзные силы привлекли на свою сторону математиков для оптимизации бомбардировочной авиации, «морской войны» и операций по освобождению Европы. Разработанные в это время математические модели оказались настолько эффективными, что со временем они оформились в несколько отраслей науки, за открытия в области которых до сих пор вручает нобелевские премии.

Первая часть пособия содержит исторически более ранние задачи, из которых родились модели линейного, нелинейного и дискретного программирования, объединенных общим названием «Математическое программирование». Позднее эти задачи записали графически в виде сетей, деревьев и других топологий, из которых родилась «Теория графов». Решению задач на графах посвящена вторая часть пособия. Третья часть описывает основополагающие модели на стыке математики, экономики и социологии, когда несколько агентов борются и конкурируют между собой. Этот раздел носит название «Теория игр». Завершают пособие дополнительные главы, каждая из которых является отдельной отраслью науки, но ознакомление с которыми необходимо каждому специалисту.

Автор постарался выбрать стиль изложения, понятный широкому кругу читателей, не имеющих серьезной математической подготовки. Модели и постановки задач намеренно даны в упрощенном виде, пригодном для практического использования программистами, бизнес-аналитиками и экономистами.

ЧАСТЬ I. МАТЕМАТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

1. Производственная задача, симплекс-метод

В обществе, как и в любой другой комплексной системе, проходят сложные экономические процессы, вызванные рядом факторов: социальных, политических и экономических. Анализ этих процессов представляется трудной задачей, поскольку он предполагает изучение немалого объема информации. В связи с этим стало популярно использование математических моделей, то есть схематичных представлений реально существующих процессов, описанных при использовании специальных символов. Чаще всего экономико-математические модели предстают в виде математических формул, графиков, систем уравнений и неравенств.

Математическую модель можно представить в виде уравнения или системы неравенств и уравнений, которые описывают взаимосвязи, происходящие в реальной системе.

Целью моделирования является поиск наилучших решений (лучших значений решения системы уравнений и неравенств). Выражаясь математическим языком, необходимо решить задачу на экстремум (найти минимум или максимум).

Решение такой задачи оптимизации называется математическим программированием. В данном контексте термин «программирование» не имеет ничего общего с тем программированием, где нужно писать код на определенном языке, создавать элементы управления пользовательского интерфейса и прочее, так как он появился еще до изобретения компьютера. Под этим термином следует понимать именно «вычисление».

Математическое программирование состоит из нескольких частей, одной из которых служит линейное программирование.

Задача линейного программирования

Если целевая функция и ограничения, представленные в виде неравенств и/или уравнений, состоят из переменных (x_1, x_2, \dots, x_n) в первой степени, то есть являются линейными, то мы имеем дело с задачей линейного программирования. В других случаях задачи могут относиться к нелинейному, дискретному, булеву и другим разделам математического программирования. Но поскольку задачи дискретного, нелинейного и др. программирования сложны в вычислениях, нужно стараться при возможности свести нелинейные модели к линейным, которые хорошо изучены и имеют стабильную работу (не дают сбои при вычислениях).

Для того, чтобы составить математическую модель задачи линейного программирования, необходимо выполнить четыре действия, исходя из условий задачи:

1. Записать исходные данные.
2. Определить переменные.
3. Сформулировать целевую функцию.
4. Записать систему ограничений.

Производственная задача

Изучение линейного программирования чаще всего начинают с производственной задачи, самой простой и понятной. Ее суть сводится к следующему: есть некое производственное предприятие, выпускающее несколько видов продукции. Обозначены ресурсы и их количество, необходимое для производства продукции. Количество ресурсов ограничено. Известно, за какую цену можно продать ту или иную продукцию. Необходимо понять, какие производственные изделия следует произвести и в каком количестве, чтобы прибыль предприятия была максимальной.

Разберем задачу на примере. Рассмотрим столярный цех, который производит кухни и офисную мебель. Для каждого из двух типов продукции у нас используются следующие материалы: ДСП, стекло и фурнитура (дверные ручки,

петли и прочее). В итоге мы имеем номенклатуру продукции (кухня и офисный комплект) и номенклатуру материалов, из которых они изготавливаются. Теперь мы можем составить матрицу, в которой будут описаны расходы материалов на изготовление одной кухни или одного офисного комплекта (таблица 1.1).

Например, для изготовления кухни нам нужно a_{11} листов ДСП, a_{12} квадратных метров стекла и a_{13} элементов фурнитуры. Для производства офисного комплекта нужно другое количество материалов: a_{21} листов ДСП, a_{22} м² стекла и a_{23} элементов фурнитуры. В таком случае матрица примет следующий вид:

Таблица 1.1

Исходные данные производственной задачи

	ДСП	Стекло	Фурнитура
Кухня	a_{11}	a_{12}	a_{13}
Офисный комплект	a_{21}	a_{22}	a_{23}

Далее, в производственной задаче задаются **ограничения** по складу, в котором хранятся материалы (таблица 1.2). Пусть на складе хранятся b_1 листов ДСП, b_2 квадратных метров стекла и b_3 элементов фурнитуры. Большее количество материалов склад столярного цеха уместить не может.

Таблица 1.2

Ограничения по складу в производственной задаче

	ДСП	Стекло	Фурнитура
Склад	b_1	b_2	b_3

Третий блок исходных данных – **стоимость продажи** готовой продукции (таблица 1.3). Цена, по которой столярный цех может реализовать один комплект кухонного гарнитура, составляет c_1 единиц денег.

В случае одного комплекта офисной мебели эта величина составляет c_2 единиц денег.

Таблица 1.3

Стоимость продажи готовой продукции

	Стоимость продажи
Кухня	c_1
Офисный комплект	c_2

Если объединить все исходные данные (таблица 1.4), то их можно записать в виде следующей таблицы:

Таблица 1.4

Объединение исходных данных производственной задачи

	ДСП	Стекло	Фурнитура	Стоимость продажи
Кухня	a_{11}	a_{12}	a_{13}	c_1
Офисный комплект	a_{21}	a_{22}	a_{23}	c_2
Материалов на складе	b_1	b_2	b_3	

Следующим шагом нам нужно определить искомые переменные (таблица 1.5). В данном случае мы будем искать оптимальное количество каждого вида продукции, которые нужно изготовить. Обозначим переменные x_1 – количество кухонных гарнитуров, x_2 – количество офисных комплектов.

Таблица 1.5

Исходные данные с искомыми переменными

	ДСП	Стекло	Фурнитура	Стоимость продажи	Количество продукции
Кухня	a_{11}	a_{12}	a_{13}	c_1	x_1
Офисный комплект	a_{21}	a_{22}	a_{23}	c_2	x_2
Склад	b_1	b_2	b_3		

Найти оптимальное количество производимой мебели того или иного вида можно по разным критериям: стоимость, скорость изготовления, минимальное количество отходов и др. Рассмотрим в качестве примера наиболее простой и понятный **критерий оптимальности** – максимальная прибыль от проданной мебели.

Вид целевой функции отвечает на вопрос: «Какое соотношение количества произведенной продукции разных видов можно назвать оптимальным?». В нашем случае целевая функция будет выглядеть следующим образом:

$$C = c_1x_1 + c_2x_2 \rightarrow \max \quad (1.1)$$

То есть мы стоимость одного кухонного гарнитура c_1 умножаем на количество произведенных кухонь x_1 , к этому числу мы прибавляем стоимость одного офисного комплекта c_2 , умноженную на количество офисных комплектов x_2 . Следует отметить, что под стоимостью имеется в виду стоимость продажи, а не стоимость изготовления.

Целевая функция имеет **линейный вид**, следовательно, мы можем использовать задачу линейного программирования.

Теперь можно перейти к описанию **ограничений**, которые не позволяют целевой функции устремиться в бесконечность. В качестве ограничений в этой задаче выступает вместимость склада материалов. Ограничения будут выглядеть следующим образом:

$$\begin{cases} a_{11}x_1 + a_{21}x_2 \leq b_1 \\ a_{12}x_1 + a_{22}x_2 \leq b_2 \\ a_{13}x_1 + a_{23}x_2 \leq b_3 \end{cases}$$

Опишем **ограничения**. Первое ограничение строится по первому столбцу таблицы № 5. Данное ограничение говорит нам о том, что количество материала ДСП, израсходованного на производство x_1 кухонь и x_2 офисных комплектов, должно быть не больше b_1 , то есть запасов данного материала на складе. Второе и третье ограничения строятся аналогично: по второму и третьему столбцу таблицы, соответственно.

Часто в задачах линейного программирования встречается еще одно ограничение – **неотрицательность** искомых переменных, в нашем случае, x_1 и x_2 . Оно обозначает, что столярный цех не может произвести, к примеру, -3 кухонных гарнитуров. Таким образом, запишем вторую группу ограничений:

$$x_1 \geq 0, \quad x_2 \geq 0.$$

На этом задачу можно считать сформулированной. Остается теперь найти способ ее решения, то есть отыскания таких значений переменных x_1 и x_2 , которые доставляют максимум *целевой функции* C и при этом не нарушают ограничений.

В общем виде такая математическая модель получила название «стандартная форма записи задачи линейного программирования»:

$$\begin{aligned} W &= \sum_{j=1}^n c_j x_j \rightarrow \max \\ \sum_{j=1}^n a_{ij} x_j &\leq b_i, \quad (i = \overline{1, m}) \end{aligned} \tag{1.2}$$

Графическое представление задачи линейного программирования

Задача линейного программирования помимо аналитического представления может быть представлена графически. Графическое представление может быть полезно для образного понимания функционирования задач линейного программирования.

В нашей задаче мы определили две искомые переменные x_1 и x_2 , значит мы можем взять две оси и изобразить задачу на плоскости. Любая точка на этой плоскости имеет две координаты (x_1, x_2) . Значит, зная стоимость одного кухонного гарнитура и одного офисного комплекта, мы можем в каждой точке плоскости посчитать значение целевой функции по формуле (1.1) как показано на рис. 1.1.

Очевидно, что, увеличивая значения переменных x_1 и x_2 , значение целевой функции будет *возрастать*, так как c_1 и c_2 являются положительными константами. Максимум целевой функции будет стремиться к *бесконечности* и будет находиться в правом верхнем углу. Следовательно, чтобы вычислить значение целевой функции нам ее нужно ограничить, то есть ввести ограничения.

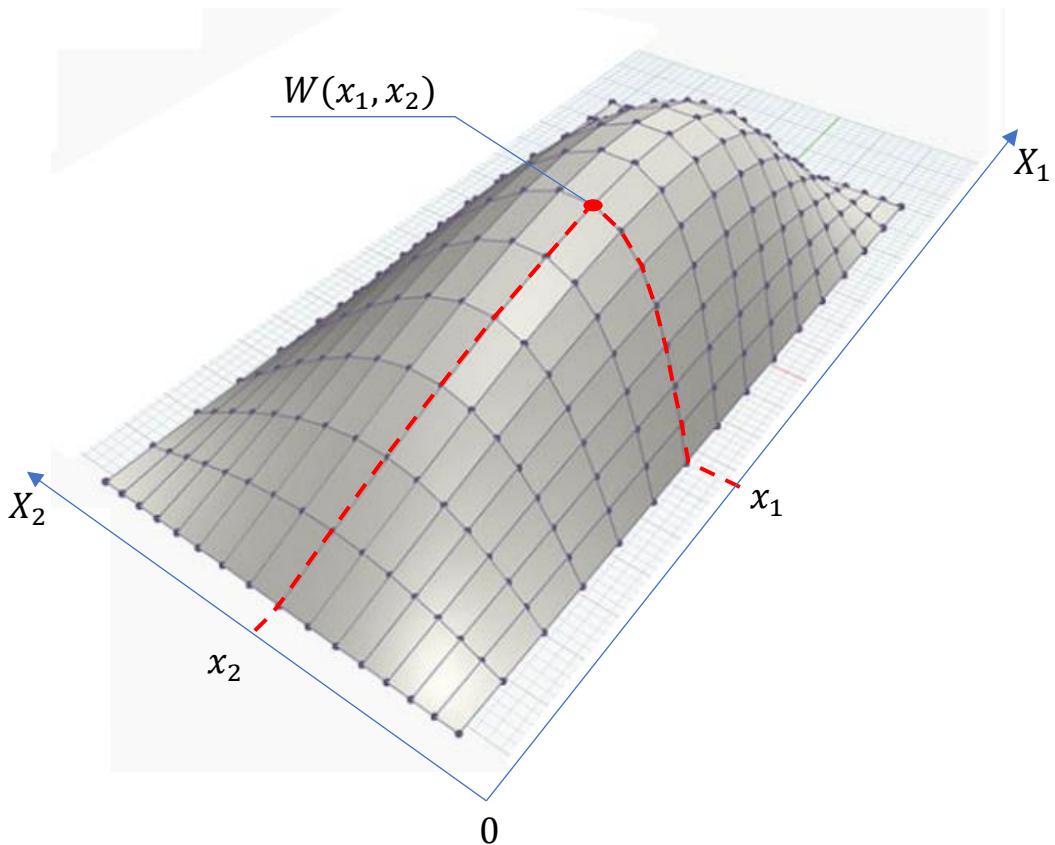


Рис. 1.1. Целевая функция

В графическом виде, если мы выражаем одну переменную через другую и заменяем знак неравенства на равенство, то получаем уравнение прямой $y = ax + b$. Например, выразим переменную x_1 из первого ограничения. Тогда мы получим следующее:

$$x_1 = \frac{b_1 - a_{12}x_2}{a_{11}}$$

Это означает, что каждое наше ограничение представляет собой какую-то прямую на плоскости.

А так как в ограничениях у нас находятся знаки неравенства, то нас интересует та область графика, которая находится выше или ниже этой прямой в зависимости от знака неравенства. Если **знак неравенства в ограничении \leq** , то мы должны рассматривать все, что находится ниже прямой, и наоборот.

Значит наши ограничения отрезают нам область, внутри которой и следует искать максимум (рис. 1.2). А часто забываемое четвертое ограничение на **неотрицательность** искомых переменных эту область замыкает.

В итоге мы ищем **максимум** в замкнутой области, что представляется несложной задачей. Как правило, максимум функции лежит в одной из точек пересечения ограничений. В редких случаях, когда целевая функция в двух точках одинакова, то максимум лежит на всей прямой. Но не может быть точки максимума, лежащей на середине прямой.

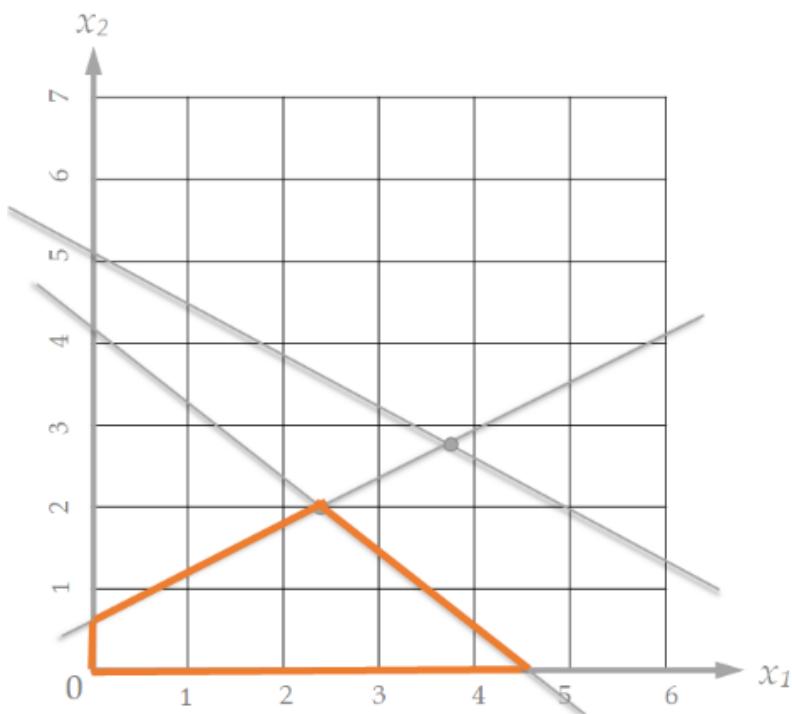


Рис. 1.2. Графики ограничений

Симплекс-метод

Вернемся к нашей модели и постараемся найти ее решение не в графическом виде, а в аналитическом. Для поиска максимума целевой функции существует множество способов, множество различных приемов и методов, но, наверное, самым универсальным и простым является так называемый симплекс-метод.

Начнем с начала и разберемся, что означает слово «симплекс». Он представляет собой **геометрическую фигуру**, количество углов которой на единицу больше размерности пространства (рис. 1.3). То есть в двумерном пространстве, на плоскости, симплекс будет являться треугольником. В случае одномерного пространства, например, одной оси x , симплекс будет представлять собой

отрезок, а в случае 0-мерного пространства симплекс будет точкой. Аналогично, в трехмерном пространстве симплекс будет являться *пирамидой* с треугольным основанием. Конечно, симплекс-метод существует и четырехмерном, и в n -мерном пространстве. Но дальше трехмерного пространства сложно представить себе, как будет выглядеть симплекс, хотя аналитически он существует.

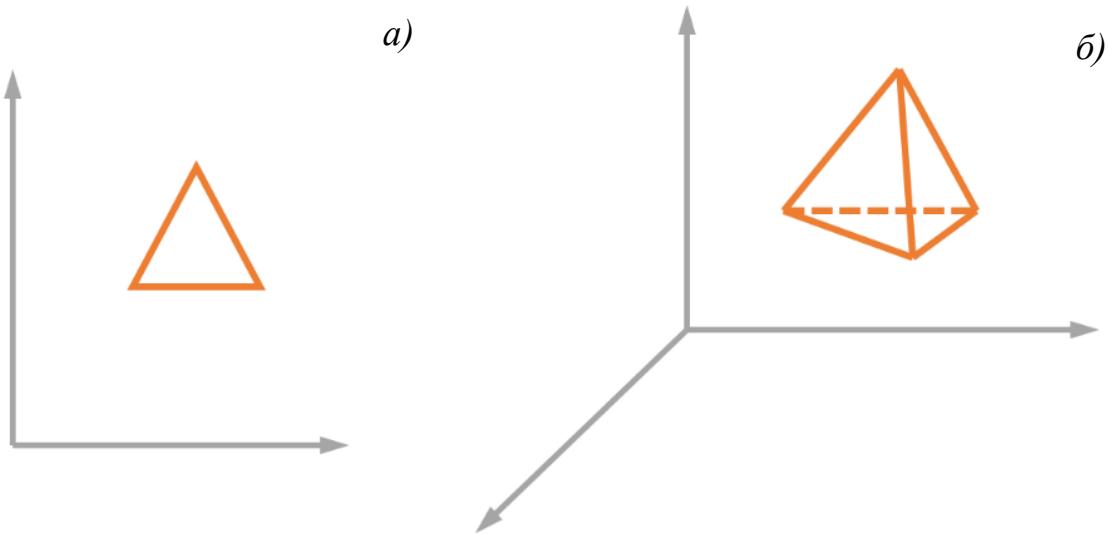


Рис. 1.3. Симплекс в двумерном (а) и трехмерном (б) пространствах

Мы разобрались в том, что из себя представляет симплекс. Теперь следует понять, как он работает. *Симплекс-метод* представляет собой алгоритм решения задачи линейного программирования посредством перебора вершин выпуклого многогранника в многомерном пространстве.

Вернемся к *графическому* представлению нашей математической модели и помещаем в наше рабочее пространство симплекс. Поскольку рабочее пространство у нас плоское (имеем только две координаты x_1 и x_2), симплекс будет представлен в качестве равностороннего треугольника (рис. 1.4). Длина стороны симплекса выбирается произвольно, как и его первоначальное положение. Помещаем данный треугольник abc в какую-то начальную точку (рис. 1.4) и дальше в каждой вершине треугольника считаем значение целевой функции по формуле (1.2). Так как каждая вершина треугольника – это точка на плоскости, значит она имеет свои **координаты** (x_1, x_2).

Сравниваем значения целевой функции в каждой вершине симплекса. В случае поиска максимального значения целевой функции отбрасывается та

вершина, в которой значение наименьшее, и наоборот. Поскольку наша **целевая функция** стремится к максимуму, ту вершину, в которой значение целевой функции самое маленькое, мы отбрасываем и перекидываем треугольник через противоположное ребро (рис. 1.5).

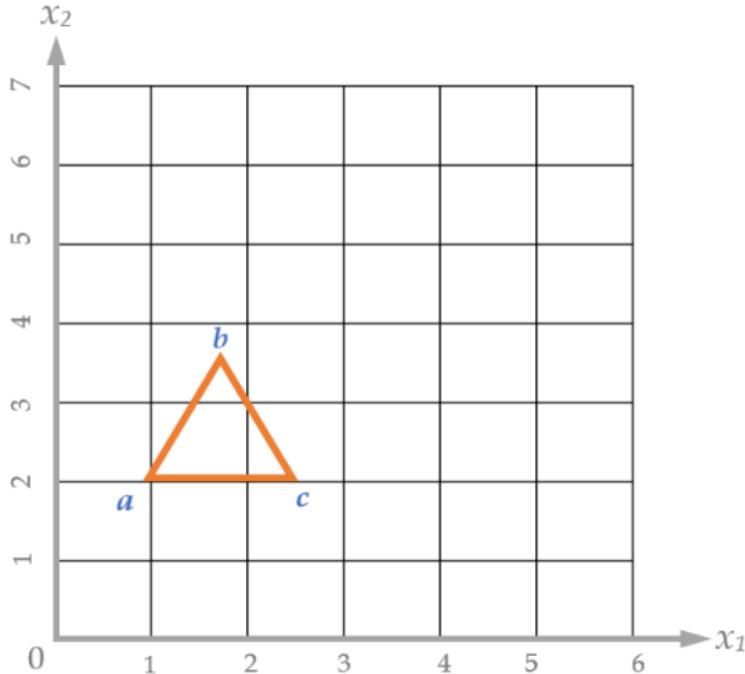


Рис. 1.4. Симплекс в двумерном пространстве

Таким образом получаем новый симплекс. Повторяем все шаги, пока не найдем максимальное значение (рис. 1.6).

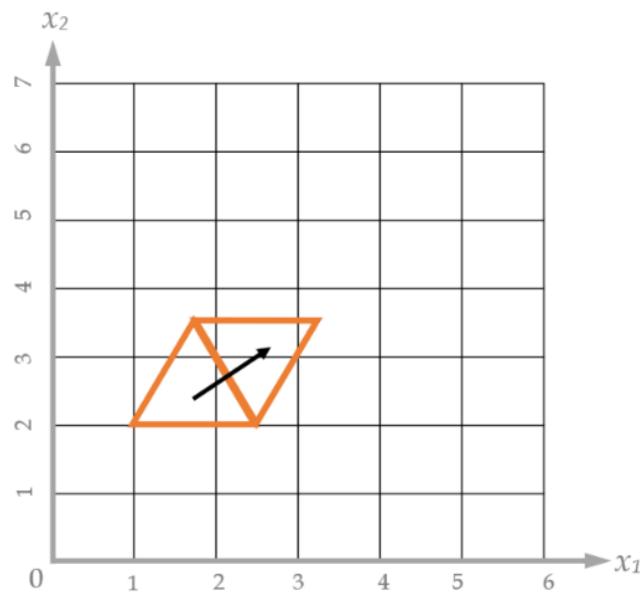


Рис. 1.5. Перекидывание симплекса в двумерном пространстве

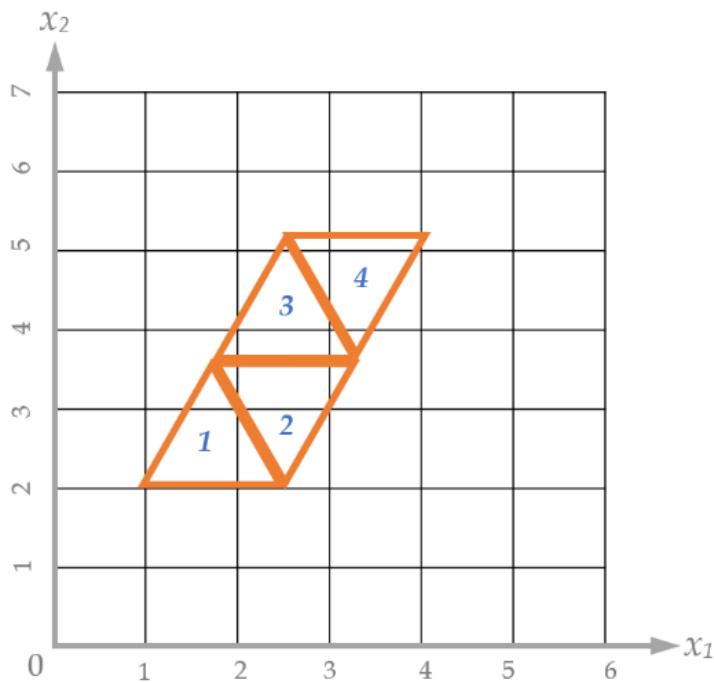


Рис. 1.6. Путь симплекса

Показателем того, что мы нашли **максимум**, являются два различных варианта развития событий: систематический поворот относительно одной из сторон (позиции 2-3, рис. 1.7), вращение вокруг какой-либо точки (например, точки D, рис. 1.7).

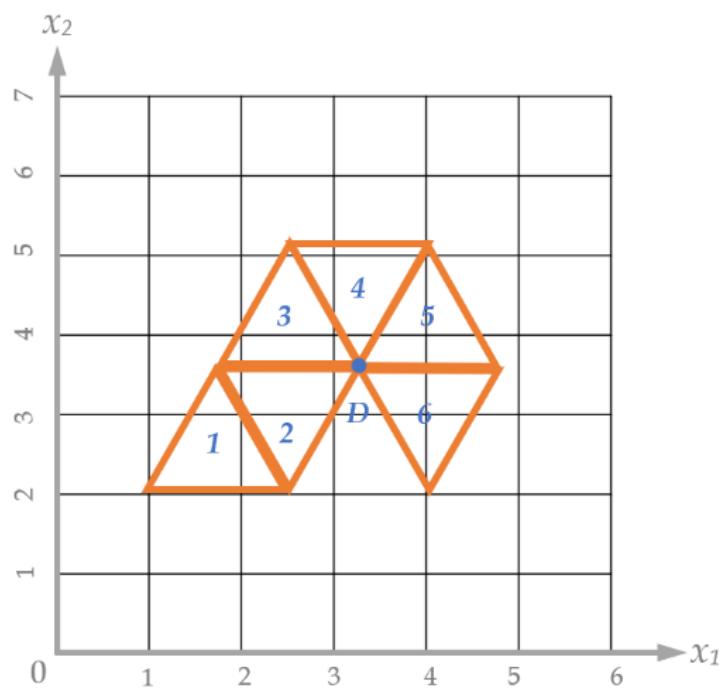
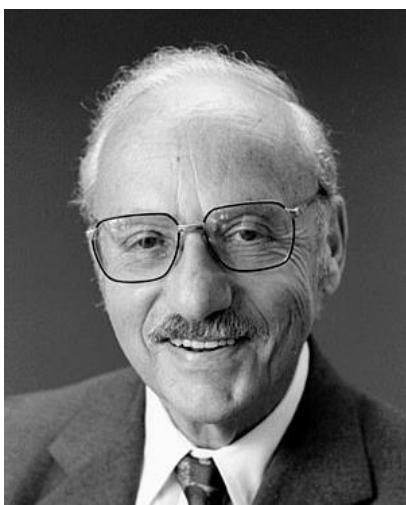


Рис. 1.7. Зацикливание симплекса вокруг максимума

В большинстве случаев таким событием является второй вариант, то есть попадание точки максимума внутрь одного из треугольников, то есть перекидываясь, треугольник накрыл точку максимума. В этот момент треугольник начинает закручиваться вокруг одной из вершин, которая находится ближе всего к точке максимума. Как только треугольник совершил полный оборот и пришел в ту позицию, где он уже был, значит, в окрестности точки и находится максимум. Таким образом находится окрестность точки максимума. Для поиска более точного результата следует уменьшить симплекс, например, можно взять одну пятую часть от него.



Изобретателем симплекс-метода считается американский математик **Джордж Бернард Данциг** (8 ноября 1914 - 13 мая 2005) - выдающийся математик США, автор и разработчик симплекс-метода. Считается «отцом линейного программирования», наряду с наряду с Леонидом Канторовичем и Джоном фон Нейманом. Родился в Портленде, США в семье еврейских эмигрантов из Польши. Учился в Мэрилендском и Мичиганском университете. Работал в Калифорнийском университете в Беркли и Стэнфордском университете. Будучи студентом университета, однажды опоздал на урок и принял написанные на доске уравнения за домашнее задание. Оно показалось ему сложнее обычного, но через несколько дней Бернард всё-таки смог его выполнить. Оказалось, что это были «нерешаемые в то время» задачи по статистике, над которыми безуспешно работали многие учёные. Эта история стала очень популярной, обросла легендами и была использована в первых кадрах фильма «Умница Уилл Хантинг».

Пример решения производственной задачи

Для наглядности решения производственной задачи возьмем такие исходные данные (таблица 1.6). Компания производит два вида мобильных телефонов. Для производства каждого используются следующие материалы: пластик, стекло, алюминий. Для производства первого вида мобильного телефона используется 3 единицы пластика, 5 единиц стекла и 8 единиц алюминия. Для производства второго – 7 единиц пластика, 5 единиц стекла, 1 единица алюминия. На складе компании находятся 100 единиц пластика, 150 единиц стекла и лишь 75 единиц алюминия. Первый вид телефона компания может продать за 25 единиц, а второй – за 13 единиц. Необходимо понять, в каком количестве нужно

произвести мобильные телефоны каждого вида, чтобы максимизировать выручку компании, при этом соблюдая ограничения по складским запасам.

Таблица 1.6

Исходные данные с искомыми переменными примера производственной задачи

	Пластик	Стекло	Алюминий	Стоимость продажи	Количество продукции
Мобильный телефон 1	3	5	8	25	x_1
Мобильный телефон 2	7	5	1	13	x_2
Склад	100	150	75		

В таблице 1.7 представлено решение задачи, полученное симплекс-методом. Наилучшим вариантом является производство 8 единиц мобильного телефона 1 и 10 единиц мобильного телефона 2. Для этого потребуется 94 номенклатурные единицы пластика, 90 номенклатурных единиц стекла и 74 номенклатурные единицы АI. При таком раскладе итоговая выручка компании составит 330 денежных единиц.

Таблица 1.7

Решение производственной задачи симплекс-методом

	Количество продукции	Выручка по продукции
Мобильный телефон 1	8	200
Мобильный телефон 2	10	130
Итого	18	330

2. Классическая транспортная задача

Производственная задача, рассмотренная в предыдущей главе – это базовый, основополагающий элемент линейного программирования, на котором строятся все остальные задачи путем модернизации и развитием которого они считаются. Двумя из этих развитий являются классическая транспортная задача и задача о назначениях.

Классическая транспортная задача впервые была сформулирована под решение проблемы перевозки грузов. Но ее методологию можно применить в различных сферах - от ИТ до подбора персонала.

В классической постановке она звучит так: есть город и в нем располагаются несколько складов однотипной продукции (рис. 2.1).

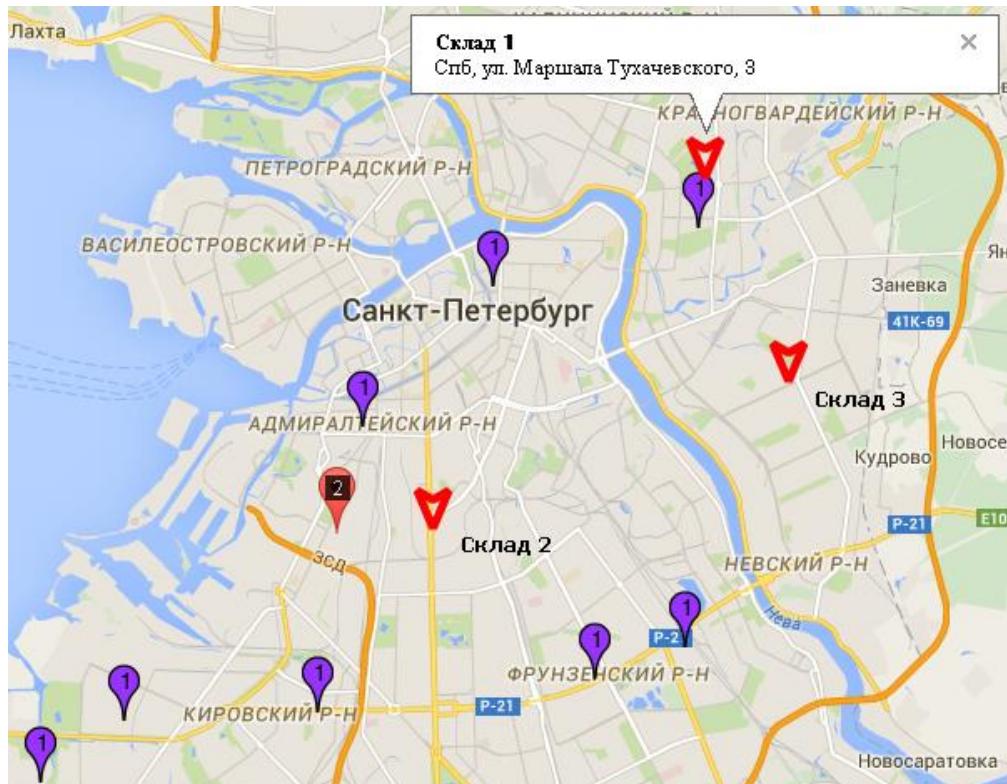


Рис. 2.1. Расположение складов и магазинов на карте города

Однотипность продукции является ключевым моментом. И есть несколько объектов на территории города, которые являются потребителями этой продукции. У каждого объекта имеется определенный спрос, то есть необходимое для осуществления деятельности количество продукции. В свою очередь, у

каждого склада имеется его определенный запас – положительный потенциал этого необходимого товара. Есть стоимость перевозки - везти со склада в ближайший магазин намного эффективнее (дешевле, быстрее или любой другой критерий стоимости, который важно рассмотреть в данной задаче). Задача сводится к тому, что необходимо построить ***оптимальный план*** перевозок продукции от поставщиков (например, склады) к потребителям (например, магазины), обеспечивающий ***минимальные транспортные расходы***.

Математическая постановка задачи

Эту задачу можно записать следующим образом – построить небольшой граф (рис. 2.2). Слева обозначены поставщики (фактические склады). Справа – потребители (магазины). Пусть имеется m поставщиков однородной продукции, у которых сосредоточено a_1, a_2, \dots, a_m единиц соответственно. Также имеется n потребителей продукции, потребности которых составляют b_1, b_2, \dots, b_n единиц продукции соответственно.

Далее от каждого поставщика к каждому потребителю, к которому существует дорога, строится ребро и обозначается его величина c_{ij} - стоимость перевозки единицы продукции от i -го поставщика j -му потребителю. Соответственно, необходимо найти маршруты и объемы перевозок продукции ***минимальной стоимости***.

В математической постановке имеется еще одно условие того, что данная система является ***замкнутой***, и все количество товара со склада должно переместиться в магазин, то есть запас на складах полностью удовлетворяет ***спрос*** магазинов – не превышает и не является меньшим:

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

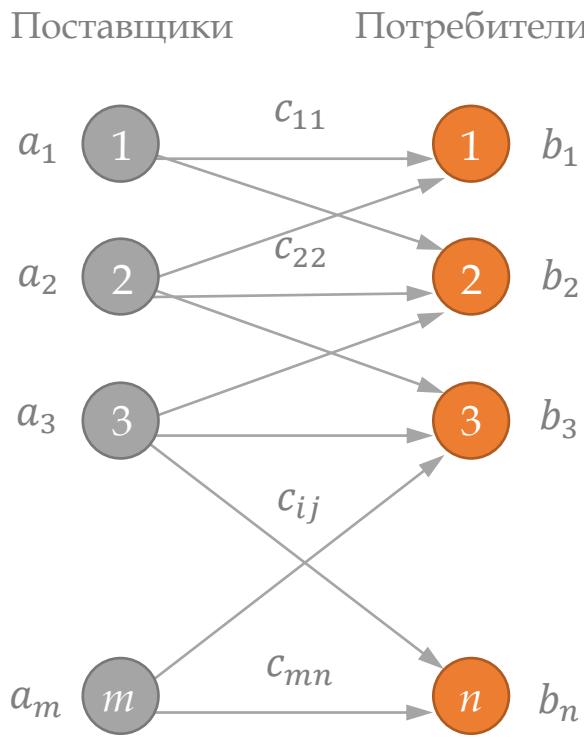


Рис. 2.2. Графическое представление задачи

Сначала рассмотрим в такой классической постановке, далее разберем, как действовать, когда происходит дефицит или профицит товара. **Равенство** является первым отличием от производственной задачи. Если в предыдущей задаче в условии и ограничениях неравенство, то здесь строгое равенство.

После того как нарисован граф, преобразуем его в *набор матриц*. Граф и матрица очень сильно связаны между собой, то есть граф можно преобразовать в матрицу и матрицу можно преобразовать в *граф*. Понятие графа разберем чуть позже.

Граф с рисунка 2.2. преобразован в три матрицы (таблица 2.1): **матрица-столбец**, в котором записано количество товаров у каждого поставщика a_i ; **матрица-строка**, в которой записаны потребности каждого магазина b_j ; матрица, в ячейках которой записана стоимость доставки c_{ij} . Нулевая стоимость перевозки означает, что перевести продукцию в данный пункт невозможно.

Таблица 2.1

Исходные данные

b_j	b_1	b_2	...	b_n
-------	-------	-------	-----	-------

a_i	c_{ij}	1	2	...	n
a_1	1	c_{11}	c_{12}	...	c_{1n}
a_2	2	c_{21}	c_{22}	...	c_{2n}
...
a_m	m	c_{m1}	c_{m2}	...	c_{mn}

После того как в модели обозначены исходные данные, следующий шаг – обозначение **переменных**, то есть тех значений, которые не известны, но необходимо найти. Искомые переменные, образующие матрицу x_{ij} - это количество единиц продукции, перевезенной от i -го поставщика к j -му потребителю (таблица 2.2).

Таблица 2.2

Искомые переменные

x_{ij}	1	2	...	n
1	x_{11}	x_{12}	...	x_{1n}
2	x_{21}	x_{22}	...	x_{2n}
...
m	x_{m1}	x_{m2}	...	x_{mn}

Далее указываем **ограничения**. Первое ограничение заключается в том, что для i -го склада количество продукции, вывезенное из данного склада в разные магазины, должно строго равняться общему запасу продукции на i -ом складе. Другими словами, запасы всех поставщиков должны быть вывезены.

$$\begin{cases} x_{11} + x_{12} + \dots + x_{1n} = a_1 \\ \dots \\ x_{m1} + x_{m2} + \dots + x_{mn} = a_m \\ \sum_{j=1}^n x_{ij} = a_i \quad (i = \overline{1, m}) \end{cases}$$

Второе ограничение аналогично первому – количество товаров, доставляемое к j -му потребителю с разных складов, должно строго равняться потребности

j -ого потребителя. Соответственно, потребности всех магазинов должны быть удовлетворены.

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

Стоит отметить, что есть и ограничение на неотрицательность переменных.

Следующий шаг – составление *целевой функции*. Целевая функция должна отражать следующее: суммарная стоимость перевозки по всем маршрутам должна быть минимальна.

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

Данная задача может быть записана в виде классической записи задачи линейного программирования (ЗЛП), то есть сначала идет целевая функция, затем – ограничения.

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

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

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

$$x_{ij} \geq 0$$

Решение задачи осуществляется с помощью симплекс-метода (метода направленного перебора), так как он самый универсальный и подходящий для любого типа задач линейного программирования.

Открытая транспортная задача

Но, прежде чем перейдем к решению конкретного примера, рассмотрим еще один частный случай. Если запасы на складах превышают потребности

потребителей (профицит) или напротив их не хватает для удовлетворения спроса (дефицит), такая постановка называется *открытой (несбалансированной) транспортной задачей*.

Данная система является открытой, и запас на складах не равен спросу магазинов:

$$\sum_{i=1}^m a_i <> \sum_{j=1}^n b_j$$

В случае дефицита:

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

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

В случае профицита:

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

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

Как же решать такую задачу? Для того чтобы выровнять баланс, необходимо ее свести к закрытой транспортной задаче, то есть ввести дополнительный **фиктивный** магазин или склад и списать скопившиеся излишки на него как показано на рис. 2.3.

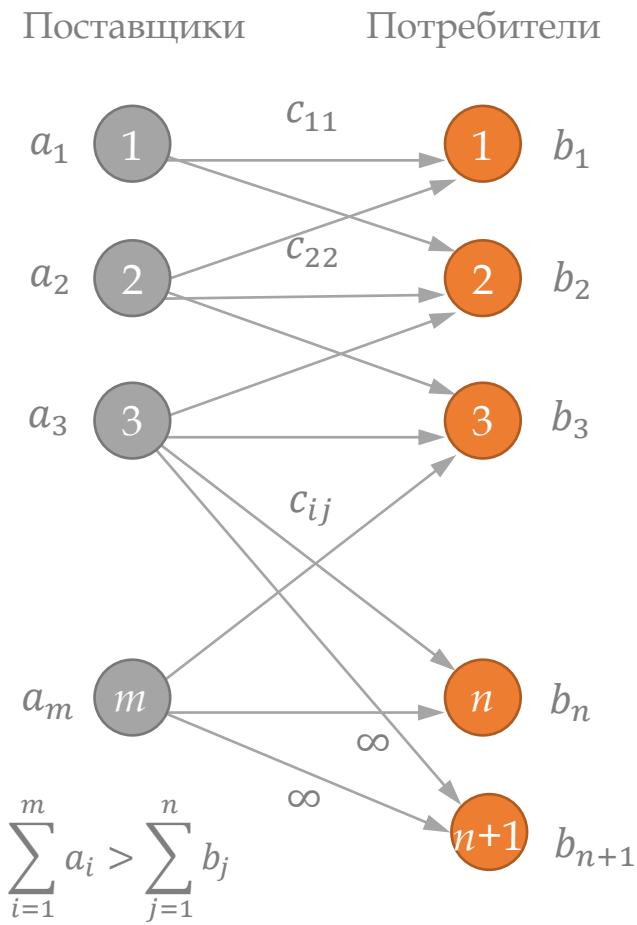


Рис. 2.3. Введение фиктивного потребителя

Введение подобного балансира переводит несбалансированную задачу в *сбалансированную*. В итоге, наше ограничение типа неравенство, когда продукции больше, чем магазинов, превращается в ограничение типа равенство. Спрос в этом магазине устанавливается равный разнице между положительным потенциалом товаров и отрицательным.

$$b_{n+1} = \sum_{i=1}^m a_i - \sum_{j=1}^n b_j$$

А стоимость перевозки в этот пункт полагаем равным бесконечности, чтобы в изначальных пунктах не возник недостаток.

$$c_{i,n+1} = \infty \quad (i = \overline{1, m})$$

Если существует обратный баланс и товаров больше, чем магазинов, вводится ***фиктивный склад***. Соответственно, когда спрос превышает предложение:

$$a_{m+1} = \sum_{i=1}^n b_j - \sum_{j=1}^m a_i$$

Таким образом, открытая транспортная задача сводится к **закрытой**, и ее можно решать вышеуказанными методами решения транспортной задачи.

Пример решения КТЗ

Рассмотрим на конкретном примере (рис. 2.4). На территории города имеется 3 кирпичных завода ($m = 3$) и 4 крупных стройки ($n = 4$), разбросанных по разным территориям. На стройках существует суточная потребность в определенном количестве кирпичей, и необходимо поставлять $\{400; 300; 350; 150\}$ единиц кирпича в день на каждую стройку соответственно. Производительность заводов составляет $\{600; 300; 500\}$ кирпичей в сутки соответственно. Стоимость заказа и доставки партии из 100 кирпичей представлена в таблице (таблица 2.3):

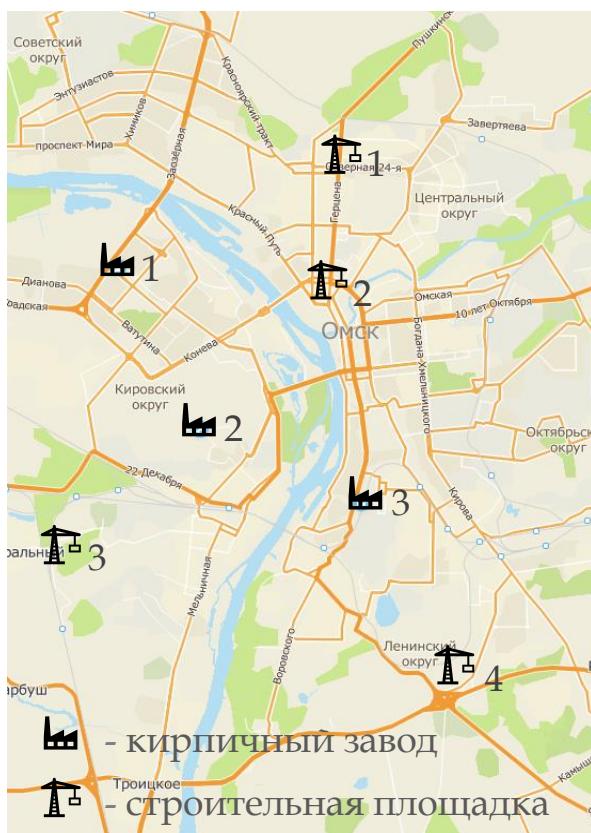


Рис. 2.4. Расположение заводов и строительных площадок

Таблица 2.3

Стоимость заказа и доставки партии из 100 кирпичей

Р тыс.	Стр. 1	Стр. 2	Стр. 3	Стр. 4
Завод 1	10	10	14	20
Завод 2	13	11	9	15
Завод 3	18	16	14	8

Данная матрица стоимости перевозок составлена с учетом всех экономических нюансов (отпускная цена, расстояние, наличие пробок и так далее).

Задача заключается в том, что необходимо найти с каких заводов на какие стройки дешевле всего будет поставлять кирпич.

Запишем формальную постановку задачи:

$$a_i = \{600; 300; 500\} \quad (m = 3)$$

$$b_j = \{400; 300, 350; 150\} \quad (n = 4)$$

В качестве массива a_i обозначим производственные мощности кирпичных заводов, в качестве b_j – потребность строек в кирпичах. Все стоимости перевозок преобразуем в матрицу c_{ij} , при этом c_{ij} - цена за 1 кирпич.

$$c_{ij} = \begin{bmatrix} 0,1 & 0,1 & 0,14 & 0,2 \\ 0,13 & 0,11 & 0,09 & 0,15 \\ 0,18 & 0,16 & 0,14 & 0,08 \end{bmatrix}$$

Искомые переменные, образующие матрицу x_{ij} , - это количество кирпичей, перевозимых с i -го завода на j -ю стройку.

$$x_{ij} = \begin{bmatrix} ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \end{bmatrix}$$

Перед формулировкой целевой функции необходимо проверить задачу на сбалансированность: 1400 кирпичей в день могут выдавать кирпичные заводы и 1200 кирпичей требуется на стройках в данный момент:

$$1400 = \sum_{i=1}^m a_i > \sum_{j=1}^n b_j = 1200$$

Соответственно, данная задача открытая и несбалансированная. Значит, необходима фиктивная стройка №5, чья производственная мощность будет равна переизбытку кирпичей – 200 штук.

$$b_5 = \sum_{i=1}^m a_i - \sum_{j=1}^n b_j = 200$$

$$b_j = \{400; 300, 350; 150; 200\} \quad (n = 5)$$

Стоимость перевозки с каждого завода на данную стройку поставим в разы больше, чем на остальные:

$$c_{ij} = \begin{bmatrix} 0,1 & 0,1 & 0,14 & 0,2 & 1 \\ 0,13 & 0,11 & 0,09 & 0,15 & 1 \\ 0,18 & 0,16 & 0,14 & 0,08 & 1 \end{bmatrix}$$

Целевая функция конкретизируется следующим образом: стоимость доставки кирпичей с заводов на стройки должна стремиться к **минимуму**.

$$W = \sum_{i=1}^3 \sum_{j=1}^5 c_{ij} x_{ij} \rightarrow \min$$

Запишем ограничения на **производительную мощность** заводов, на потребности строек и на **неотрицательность переменных** соответственно:

$$\sum_{j=1}^n x_{ij} = \{600; 300; 500\} \quad (i = \overline{1,3})$$

$$\sum_{i=1}^m x_{ij} = \{400; 300, 350; 150; 200\}$$

$$x_{ij} \geq 0 \quad (j = \overline{1,5})$$

Результаты решения задачи получим с помощью **симплекс-метода** (данные фиктивной стройки вычеркиваем):

$$x_{ij} = \begin{bmatrix} 400 & 200 & 0 & 0 & 0 \\ 0 & 0 & 300 & 0 & 0 \\ 0 & 100 & 50 & 150 & 200 \end{bmatrix}$$

Данное распределение иков доставляет **минимум** целевой функции стоимости и удовлетворяет ограничения по перевозке кирпичей. Но стоит отметить,

что решений данной задачи может быть несколько, однако у них всегда одно и тоже значение целевой функции.

Что нужно сделать после того, как получено выражение в таком виде? Необходимо интерпретировать результат в виде таблицы (таблица 2.4.) и следующих выводов:

1. Стройки №1 и №4 полностью обслуживаются заводами №1 и №3.
2. Стройка №2 закупает кирпич у заводов №1 (200 шт.) и №3 (100 шт.).
3. На стройку №3 большую часть кирпича доставляют с завода №2 (300 шт.) и недостающие 50 шт. с завода №3.
4. Суммарные затраты составляют 122 тыс. руб.

Таблица 2.4

Наиболее дешевый вариант заказа кирпича

шт.	Стр. 1	Стр. 2	Стр. 3	Стр. 4
Завод 1	400	200		
Завод 2			300	
Завод 3		100	50	150

Области применения модели. Несмотря на то, что эта задача называется транспортной, данную методологию постановки можно применить к различным областям и сферам:

1. Доставка бензина АИ-92, -95, -98 и ДТ с загородных хранилищ на заправки в городе или по области.
2. Доставка строительных материалов (ПГС, бетон, кирпич и др.) со складов поставщиков на объекты строительства.
3. Доставка продуктов питания (хлеб, молоко, питьевая вода и др.) по магазинам и торговым точкам.
4. Распределение кредитных потоков от банков организациям-заемщикам (в качестве «стоимости доставки» могут выступать условия кредитования).
5. Распределение хостинг-серверов между организациями-потребителями (в качестве «стоимости доставки» можно принять место на сервере или производительные мощности).

Математическая модель, рассмотренная выше, в более усложненном виде используется и в логистических системах, которые продаются и коммерчески применяются в различных службах транспортной перевозки. В одном из модулей подобных систем и решается задача о том, откуда, куда и что надо везти.

Классическую транспортную задачу можно усложнять/развивать следующим образом:

1. Введение дополнительных целевых функций (например, время).
2. Обобщенная постановка: нет производителей и поставщиков, есть однотипные пункты с положительным и отрицательным балансом и двусторонней перевозкой.
3. Введение промежуточных пунктов маршрута.
4. Учет максимальной пропускной способности каналов перевозки.
5. Перевозка нескольких видов продукции.
6. Перевозка дискретными блоками: контейнерами, грузовиками, вагонами.
7. Объединение с задачами прокладки маршрутов перевозок, оптимальной загрузки транспортных средств, учета очередности посещения и др.



Впервые КТЗ была разработана и опубликована **Леонидом Витальевичем Канторовичем** (06.01.1912, Санкт-Петербург - 07.04.1986, Москва) - советским математиком и экономистом. Один из редких советских ученых, которым разрешалось активно взаимодействовать с западными коллегами, в том числе и с американскими. Поэтому он являлся одним из реальных авторов большого количества задач линейного программирования. Лауреат Нобелевской премии по экономике 1975 года «за вклад в теорию оптимального распределения ресурсов». Доктор физико-математических наук в 22 года (1935 г.), академик Академии наук СССР (1964 г.), профессор. В 1939 году опубликовал работу «Математические методы организации и планирования производства», чем заложил основы линейного программирования. Участник обороны Ленинграда 1942 года и ядерной программы 1948 года. После внедрения в 1949 году методов оптимизации резки металла лишил целый завод премии из-за срыва плана по сдаче металломолома. Предложил и рассчитал всесоюзные тарифы такси с платой за посадку и километраж, чем устранил нежелание водителей совершать короткие поездки. Имя собственное «Леонид Канторович» присвоено воздушному судну «Аэрофлот» Airbus A-330-300.

3. Задача о назначениях

Постановка задачи

Одной из модификаций производственной задачи, рассмотренной в разделе 1, является задача о назначениях, однако ее можно считать и следующим шагом в классической транспортной задаче, рассмотренной в разделе 2.

Рассмотрим ее на примере. Существует такая методология разработки программного обеспечения как Scrum/Agile (рис. 3.1), когда маленькой команде программистов (максимум 10-15 человек)дается каждый день некоторый стек задач, который они должны распределить между собой, выполнить и отчитаться о своих результатах в конце дня. И в этой методологии существует некоторая проблема, которую нужно решить: кому какую задачу поручить, чтобы она была выполнена максимально эффективно. Что подразумевается под словом «эффективно»? Например, с высокой степенью достижения результата, с минимальным количеством скрытых багов, с высокой эффективностью самого кода (логичность, понятность, простота) или все вышеперечисленные показатели сразу.

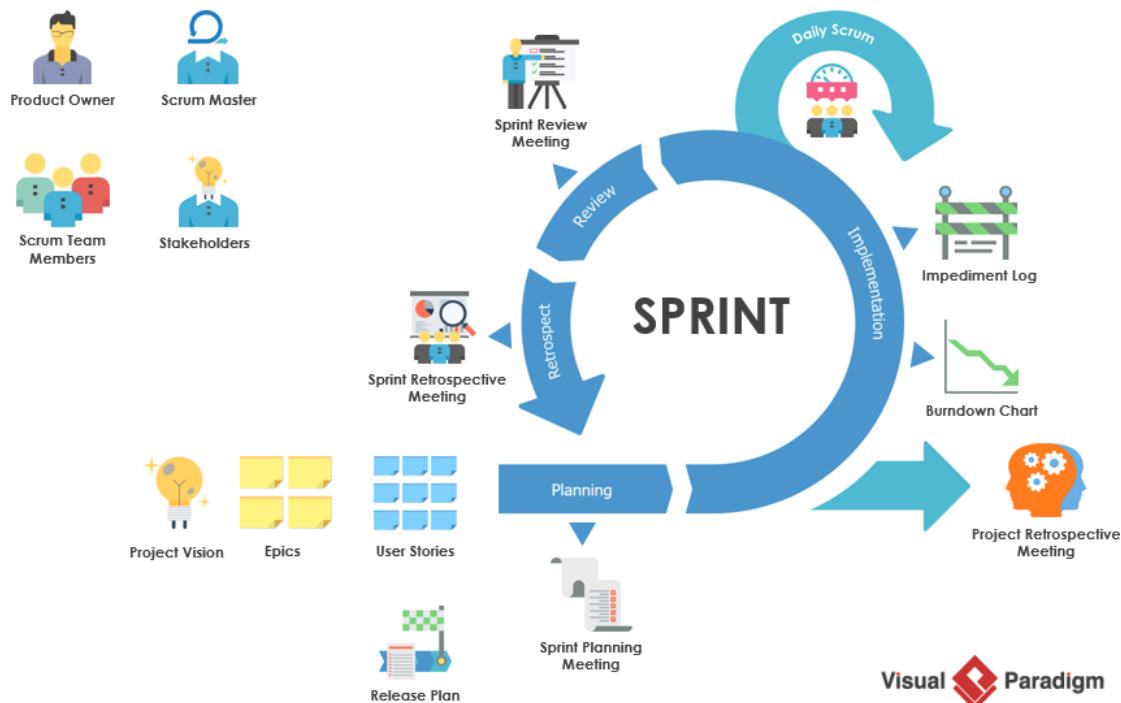


Рис. 3.1. Методология SCRUM/Agile

Пример решения задачи о назначениях

Разберем конкретный пример. Пусть имеется фирма с командой разработчиков из 3 человек (Вася, Фима и Тагир), обладающих разными компетенциями в 9 различных сферах: Python junior, Python middle, Python senior, Java junior, Java middle, Java senior, C++ junior, C++ middle, C++ senior. Под компетенцией разработчика подразумевается его эффективность решения задачи, которая определяется указанным в таблице 3.1 коэффициентом.

Таблица 3.1

Коэффициенты эффективности разработчиков

Эффективность	Вася	Дядя Фима	Тагир
Python jun	0,8	1,8	
Python mid	0,9	1,4	
Python sen	1,1	1,1	
Java jun	0,3	1,5	0,7
Java mid	0,5	1,7	0,3
Java sen	0,6	1,6	0,1
C++ jun	0,3	1,3	
C++ mid	0,1	1,5	
C++ sen		1,2	

При этом Вася способен решить за спринт 2 задачи, Дядя Фима – 3, а Тагир – 1. Задача заключается в том, что очередной спринт состоит из 7 задач различного уровня сложности и руководителю проекта требуется выбрать 6 из них на этот спринт и назначить их разработчикам так, чтобы эффективность их выполнения была максимальная.

Перейдем к непосредственному решению данной задачи и запишем ее формальную постановку. Исходные данные представлены в таблице 3.2.

Матрица-строка a_i обозначает количество задач на каждого разработчика:

$$a_i = \{2; 3; 1\} \quad (m = 3)$$

Матрица-столбец b_j – количество задач различных типов в текущем стеке:

$$b_j = \{2; 0, 1; 0; 1; 0; 0; 2; 1\} \quad (n = 9)$$

Таблица 3.2

Исходные данные

c_{ij}	Вася	Дядя Фима	Тагир	b_j
Pyt jun	0,8	1,8		2
Pyt mid	0,9	1,4		
Pyt sen	1,1	1,1		1
J jun	0,3	1,5	0,7	
J mid	0,5	1,7	0,3	1
J sen	0,6	1,6	0,1	
C++ jun	0,3	1,3		
C++ mid	0,1	1,5		2
C++ sen		1,2		1
a_i	2	3	1	

Матрица c_{ij}^T обозначает эффективность решения разработчиком текущей поставленной задачи:

$$c_{ij}^T = \begin{bmatrix} 0,8 & 0,9 & 1,1 & 0,3 & 0,5 & 0,6 & 0,3 & 0,1 & 0 \\ 1,8 & 1,4 & 1,1 & 1,5 & 1,7 & 1,6 & 1,3 & 1,5 & 1,2 \\ 0 & 0 & 0 & 0,7 & 0,3 & 0,1 & 0 & 0 & 0 \end{bmatrix}$$

Основное отличие задачи о назначениях от остальных заключается в том, что искомые переменные являются *булевского типа*, то есть могут принимать значения только 1 и 0. Получается *бинарная матрица* переменных x_{ij} , где единицы стоят в тех позициях, которые назначены данному программисту:

$$x_{ij} = \begin{cases} 1, & i \text{ исполнителю назначена } j \text{ задача} \\ 0, & \text{в противном случае} \end{cases}$$

Рассмотрим математическую запись этой задачи. Сначала проверим ее на сбалансированность: 7 задач в стеке и 6 задач для распределения между программистами.

$$6 = \sum_{i=1}^m a_i < \sum_{i=1}^n b_j = 7$$

Значит, существует дисбаланс и необходимо свести к закрытой постановке. Введем фиктивного программиста Лену с 1 задачей и эффективностью -1.

$$a_4 = \sum_{i=1}^n b_j - \sum_{i=1}^m a_i = 1$$

$$a_i = \{2; 2, 2; 1\} \quad (m = 4)$$

$$c_{ij}^T = \begin{bmatrix} 0,8 & 0,9 & 1,1 & 0,3 & 0,5 & 0,6 & 0,3 & 0,1 & 0 \\ 1,8 & 1,4 & 1,1 & 1,5 & 1,7 & 1,6 & 1,3 & 1,5 & 1,2 \\ 0 & 0 & 0 & 0,7 & 0,3 & 0,1 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

Запишем целевую функцию и ограничения в виде строгих равенств, при этом целевая функция эффективности стремится к максимуму, а ограничения сформулированы на продуктивность каждого разработчика и на количество в стеке задач определенного уровня сложности. Стоит отметить, что теперь x принимает только два значения: 0 и 1.

$$W = \sum_{i=1}^4 \sum_{j=1}^9 c_{ij} x_{ij} \rightarrow \max$$

$$\sum_{j=1}^n x_{ij} = \{2; 2, 2; 1\} \quad (i = \overline{1,4})$$

$$\sum_{i=1}^m x_{ij} = \{2; 0, 1; 0; 1; 0; 0; 2; 1\}$$

$$x_{ij} \in \{0, 1\} \quad (j = \overline{1,9})$$

Решаем задачу **симплекс-методом** и получаем следующий результат:

$$x_{ij}^T = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Для интерпретации результата удобнее будет переписать матрицу x_{ij} в виде таблицы 3.3. Получаем, что эффективнее всего будет, если:

1. Задачи Python jun распределить между Васей и Дядей Фимой;
2. Задачу Python sen отдать Васе;
3. Задачи Java mid и C++ mid решит Фима;
4. Задача же C++ sen достанется Тагиру, но она не будет выполнена из-за отсутствия у него компетенций по решению такого типа задач.

Таблица 3.3

Оптимальное распределение задач по исполнителям

Эффективность	Вася	Дядя Фима	Тагир
Python jun	1	1	
Python mid			
Python sen	1		
Java jun			
Java mid		1	
Java sen		Не будет выполнена из-за отсутствия компетенции	
C# jun			
C# mid		1	
C# sen			1

В итоге суммарная эффективность команды составит 5,9 единиц.

Распределив таким образом задачи и даже оставив одну задачу невыполненной, все равно получаем **эффективность большую, чем при любом другом распределении.**

Области применения модели

Математическую модель задачи о назначениях можно применить к:

1. Назначению спецтехники на объекты в зависимости от вида предстоящих работ (экскаваторы, грейдеры, грузовики, погрузчики и др.).
2. Назначению транспортных средств службы доставки или машин такси на ближайшие или важнейшие заявки.
3. Распределению менеджеров по клиентам исходя из их сильных сторон и потребностей клиентов (продавец, тех. поддержка, кризис-менеджер и др.).
4. Автоматической обработке заявок на кредит по банковским серверам.
5. Распределению клиентов облачных вычислений по серверам в зависимости от вида вычислений и доступных мощностей (видео, криптография, data processing, хранение, резервирование и др.).

4. Задача о рюкзаке

Постановка задачи

У нас имеется рюкзак и ряд предметов, каждый из которых обладает размерами и ценностью. Необходимо уложить как можно большее число ценных вещей в рюкзак при условии, что вместимость рюкзака ограничена.

Представим **математическую постановку** задачи (рис. 4.1).

Пусть имеется n предметов, которые необходимо взять с собой в рюкзак. Каждый предмет имеет определенный **объем** a_i и **ценность** c_i ($i = \overline{1, n}$). Рюкзак ограничен размером b . Требуется выбрать такое подмножество предметов x_i , которое помещалось бы в рюкзаке и имело бы **максимальную ценность**.

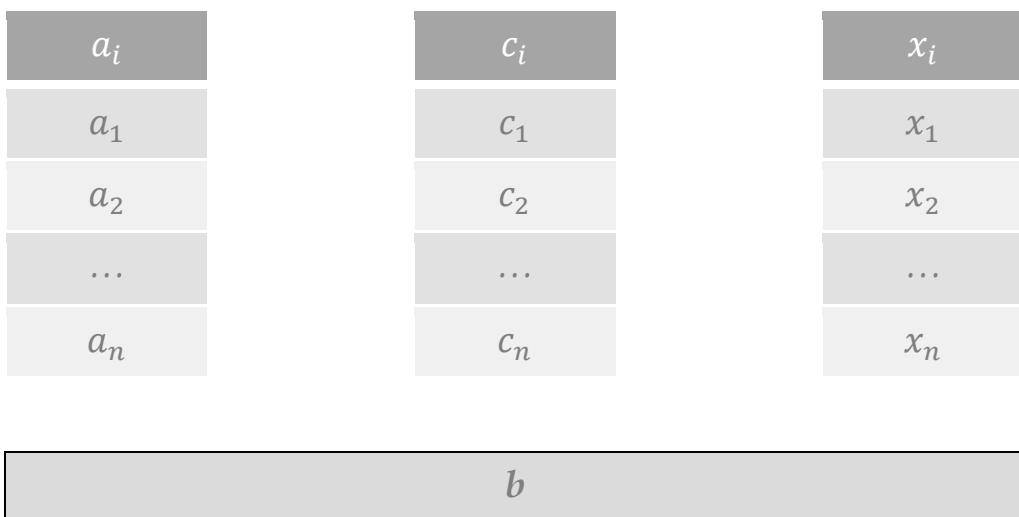


Рис. 4.1. Математическая постановка задачи

На рис. 4.2 представлена задача в графическом виде. Предметы – это прямоугольники разного объема и ценности. Рюкзак – черная рамка определенного размера¹.

Может показаться, что эта задача не такая сложная и ответ очевиден. Решения, которые сразу же «приходят на ум», называются эвристическими. Для поставленной задачи мы могли бы применить такие эвристические методы:

1. Сначала складываются наиболее ценные предметы. Если предмет не помещается, переходят к следующему.
2. Сначала складываются маленькие, а затем среди них наиболее ценные.

¹ В нашей задаче под размером подразумевается объем рюкзака, однако это может быть и его вес.

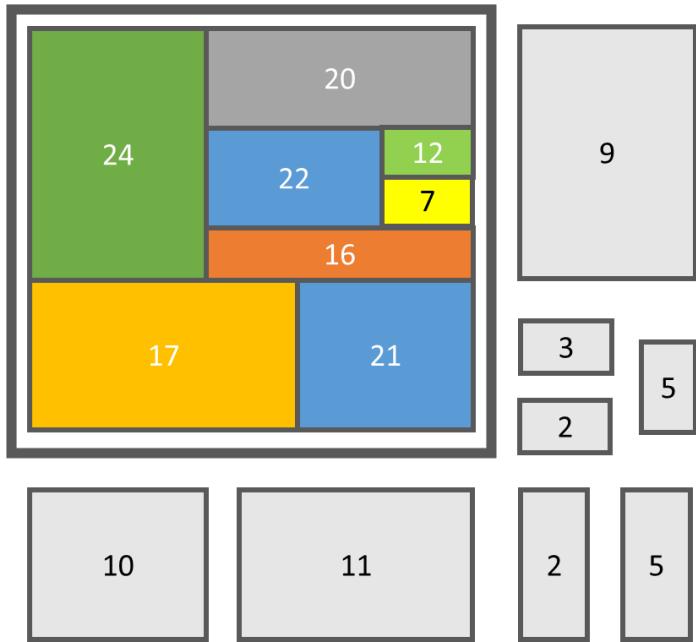


Рис. 4.2. Графическое представление задачи

3. Выбирать среди ценных предметов самые маленькие и среди маленьких – самые ценные.

4. Отсечь одновременно большие и не ценные предметы и т.д.

Эвристическими подходами можно найти решение. Но мы не можем быть уверенными, что мы получим его оптимальным и эффективным. Поэтому рассмотрим решение задачи с помощью ***дискретного программирования***.

Линейное программирование ищет решение задачи на непрерывной плоской поверхности, в то время как дискретное программирование работает с пространством, состоящим из набора отдельных точек (рис. 4.3). Функция существует только на определенном наборе точек, в промежутках между ними функцию определить нельзя.

Получается, что ***дискретное программирование*** – это поиск решения задачи линейного программирования на множестве дискретных (целочисленных) точек.

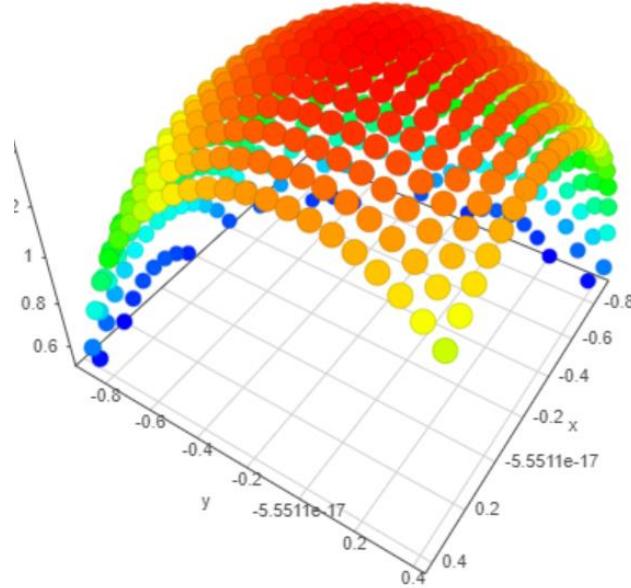


Рис. 4.3 Пространство дискретного программирования

По условию задачи нам необходимо выбрать подмножество предметов, поэтому входимость i -го предмета в рюкзак обозначим через булевский вектор-столбец. Если мы берем предмет в рюкзак, то обозначаем его единицей, если не берем – то нулем.

$$x_i \in \{0,1\}$$

Суммарная ценность отобранных предметов (обозначим ее W) должна быть максимальна:

$$W = \sum_{i=1}^n c_i x_i \rightarrow \max \quad (4.1)$$

По условию все выбранные предметы должны поместиться в рюкзаке. Иными словами, размер всех предметов, которые входят в рюкзак, должен быть в сумме меньше размера рюкзака:

$$\sum_{i=1}^n a_i x_i \leq b \quad (4.2)$$

Таким образом, необходимо выбрать такие предметы x_i , чтобы они удовлетворяли ограничению (4.2) и доставляли максимум целевой функции (4.1).

На рис. 4.4 видно, что наша задача является **комбинированным** вариантом производственной и транспортной задач (выделено оранжевым цветом).

$W = \sum_{i=1}^n c_i x_i \rightarrow \max$ Производственная задача $W = \sum_{i=1}^n c_i x_i \rightarrow \max$ $\sum_{i=1}^n a_i x_i \leq b$ $x_i \geq 0$	$\sum_{i=1}^n a_i x_i \leq b$ $x_i \in \{0,1\}$ Транспортная задача / задача о назначениях $W = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min$ $\sum_{j=1}^n x_{ij} = a_i \quad \sum_{i=1}^m x_{ij} = b_j$ $x_{ij} \geq 0 \quad / \quad x_{ij} \in \{0,1\}$
--	--

Рис. 4.4. Задача о рюкзаке

Перед нами типичная задача дискретного программирования. Для поиска целочисленных решений используются специальные методы: **метод Гомори** (отсечений), **метод ветвей и границ**, **метод Монте-Карло** и другие.

Рассмотрим принцип работы **метода Гомори** (метода отсечений):

Представим наше пространство как непрерывное и попробуем решить задачу на нем. У нас существуют все точки, поэтому мы можем найти решение симплекс-методом (рис. 4.5).

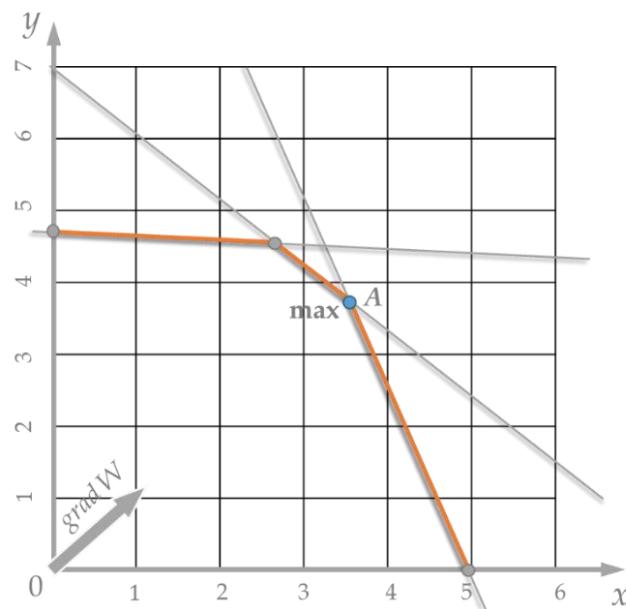


Рис. 4.5 Решение симплекс-методом

Дальше вокруг дробного решения определяем ближайшие целые значения (рис. 4.6). Значения Е и D не входят в ограниченную область и являются решениями не могут, а точки В и С могут быть потенциальными точками максимума.

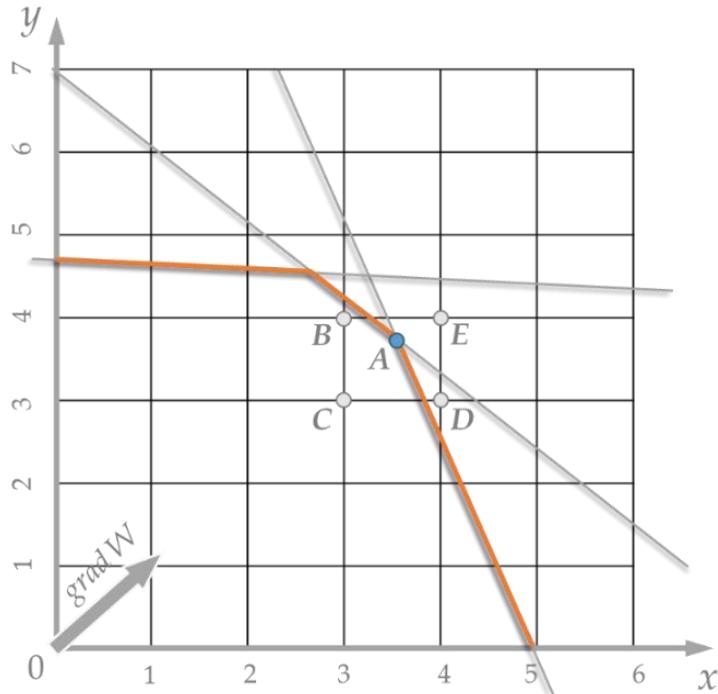


Рис. 4.6 Нахождение ближайших целых значений

С помощью дополнительного набора ограничений мы обрезаем область существования вдоль целых значений (рис. 4.7).

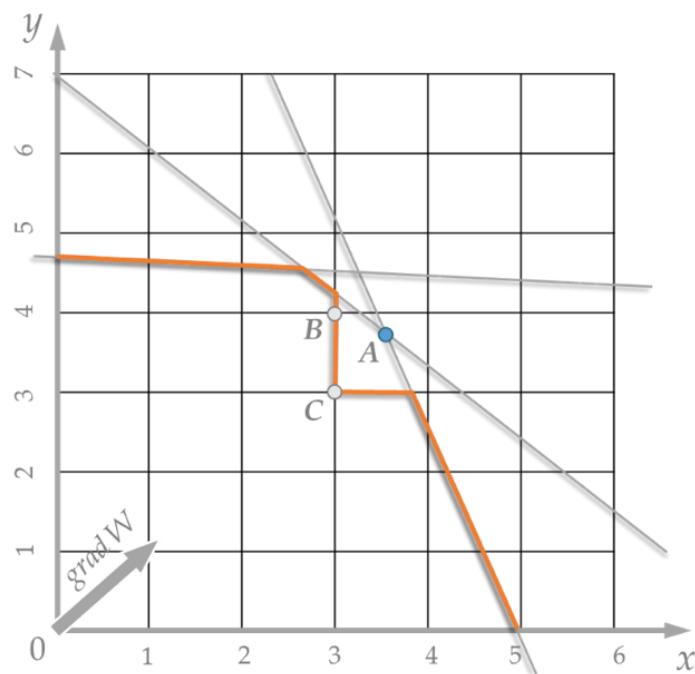


Рис. 4.7. Отсечение сегмента

Обратно представляем, что *пространство непрерывное*, применяем *симплекс-метод* и отсекаем от нашей области определения дробные участки. Алгоритм повторяется до тех пор, пока найденный симплекс-методом результат не будет строго целочисленным (рис. 4.8).

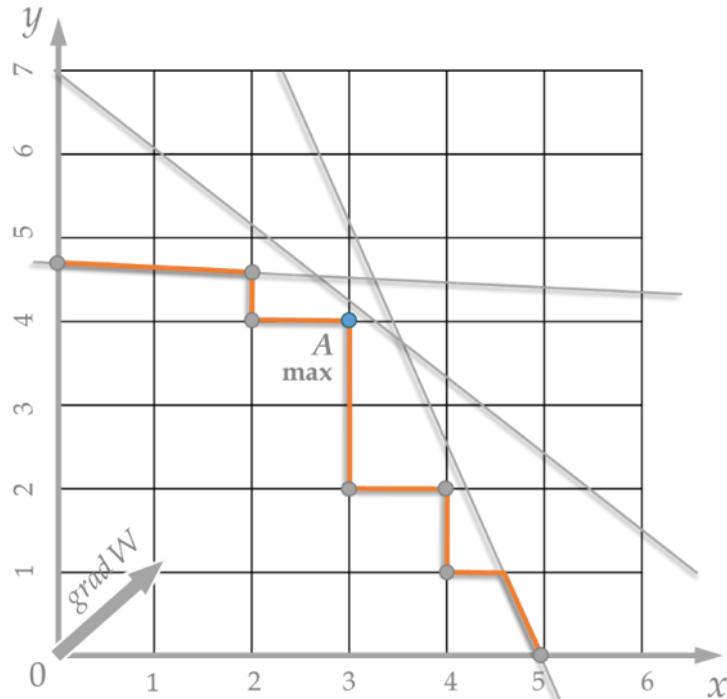
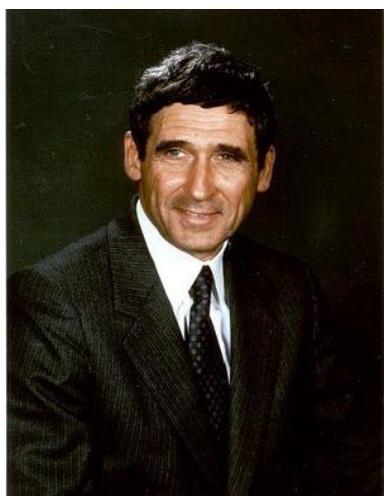


Рис. 4.8. Нахождение результата



Ральф Эдвард Гомори (род. 7 мая 1929, Brooklyn Heights, Нью-Йорк – жив до настоящего времени) - американский математик и менеджер, президент компании IBM, директор Washington Post и Bank of New York, профессор Нью-Йоркского университета. Защитил докторскую степень по математике в Принстонском университете в 1954 году. Во время службы в ВМС США с 1954 по 1957 годы, будучи специалистом по прикладной математике, сконцентрировался на исследовании операций и разработал область целочисленного программирования. В течение следующих 20 лет Гомори занимал руководящие должности в компании IBM, где под его руководством появились 2 нобелевских лауреата и были разработаны архитектура HDD, современная структура базы данных и архитектура процессоров RISC. После выхода на пенсию преподает в Нью-Йоркском университете и состоит в совете при президенте США.

На рис. 4.9 представлено решение задачи. Наибольшую ценность рюкзак будет иметь, если вместо предмета ценностью 17 он будет содержать четыре предмета ценностью 10, 5, 3 и 2. Мы нашли более оптимальное решение по сравнению с эвристическим.

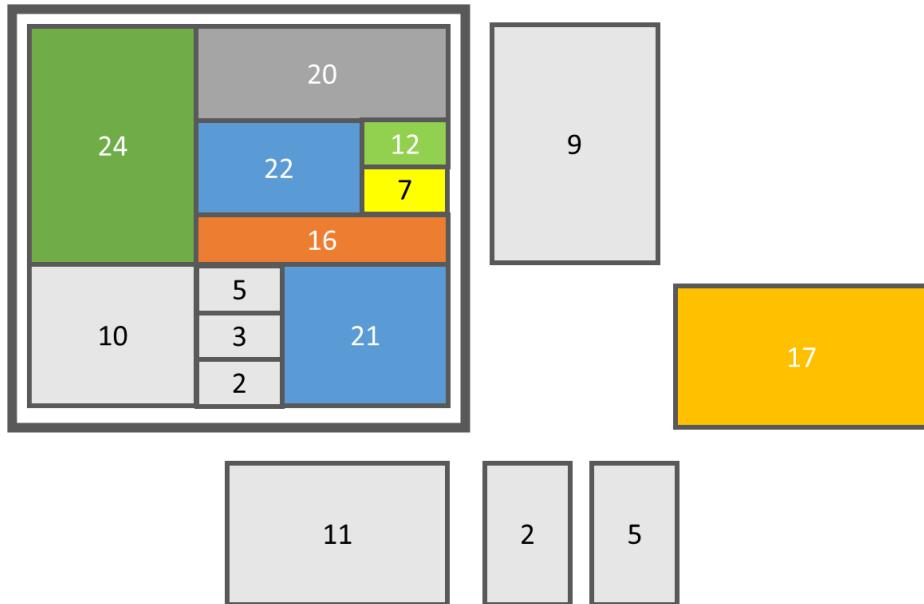


Рис. 4.9. Графическое представление задачи о рюкзаке

Области применения

Задача раскрайя. Необходимо напилить элементы шкафа или кухни из листа ДСП. Рисунок должен быть соблюден, повороты запрещены. Лист ДСП раскраивается сначала вдоль оси y , затем полосы пилятся вдоль оси x . Найти наилучший раскрой листа (рис. 4.10).

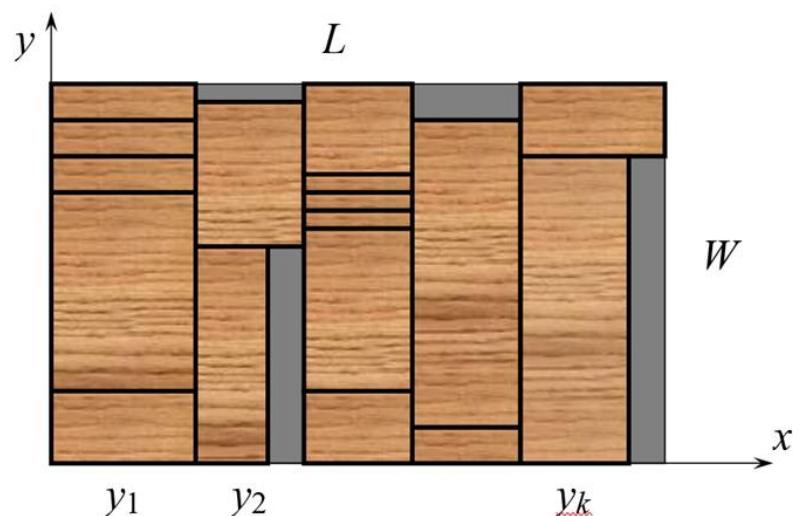


Рис. 4.10. Раскрой листа ДСП

$$W = \sum_{i=1}^k \sum_{j=1}^n c_i x_i \rightarrow \max$$

$$\sum_{j=1}^n w_i x_{ij} \leq W$$

$$\sum_{j=1}^n y_i \leq L$$

$$x_{ij} \in \{0, \dots, m_j\}$$

Дополнительные ограничения:

$$c_i x'_{ij} \leq y_i, \quad i = \overline{1, k}$$

$$m_j x'_{ij} \geq x_{ij}, \quad j = \overline{1, n}$$

$$x'_{ij} \in \{0, 1\}$$

$$y_i \geq 0$$

Задача минимизации обрезков. В организации имеются доски длиной $L=6$ м. Для изготовления деревянного изделия необходимо: 8 заготовок длиной $l_1 = 1,5$ м, 10 длиной $l_2 = 1$ м и 10 длиной $l_3 = 0,8$ м. Каким образом распилить доски на заготовки, чтобы количество отходов осталось минимальным (рис. 4.11).

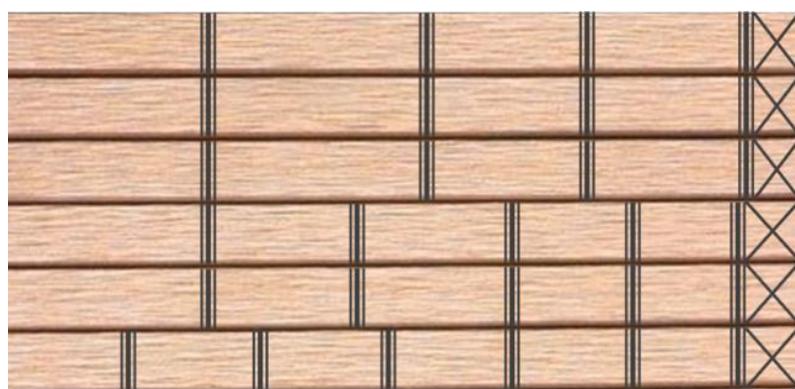


Рис. 4.11. Распил досок

Здесь C – суммарная длина отходов, X – количество используемых досок.

$$C = \sum_{i=1}^n c_i x_i \rightarrow \min$$

$$X = \sum_{i=1}^n x_i \rightarrow \min$$

$$\sum_{i=1}^k \sum_{j=1}^n a_{ij} x_i \geq b, \quad x_i \in N$$

Задача упаковки. Имеется $I = \{1, \dots, n\}$ автомобильных транспортировочных контейнеров и $J = \{1, \dots, m\}$ грузов различного объема. c_{ij} – стоимость перевозки j -го груза в i -м контейнере, w_j – объем j -го груза, v_i – вместимость контейнера i . $x_{ij} = 1$ означает, что груз j будет помещен в контейнер i . Найти оптимальное распределение грузов по контейнерам, минимизирующее стоимость перевозки (рис. 4.12).

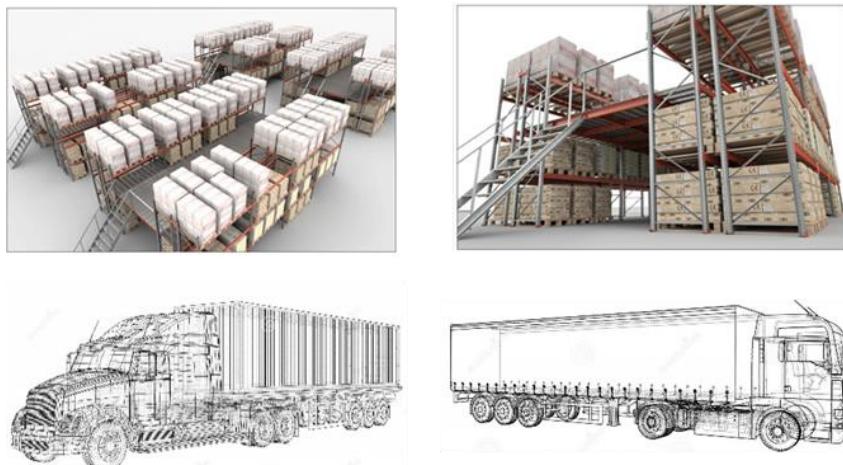


Рис. 4.12. Распределение грузов

Целевая функция и ограничения задаются следующими формулами.

$$C = \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \rightarrow \min$$

$$\sum_{i \in I} x_{ij} \leq 1$$

$$\sum_{j \in I} w_j x_{ij} \leq v_i$$

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

Задача выбора заказов для фирмы. Необходимо выбрать из большого количества поступивших заказов такое подмножество, чтобы их суммарная прибыль была максимальна, а общая трудоемкость выполнения не превышала возможности коллектива (рис. 4.13).

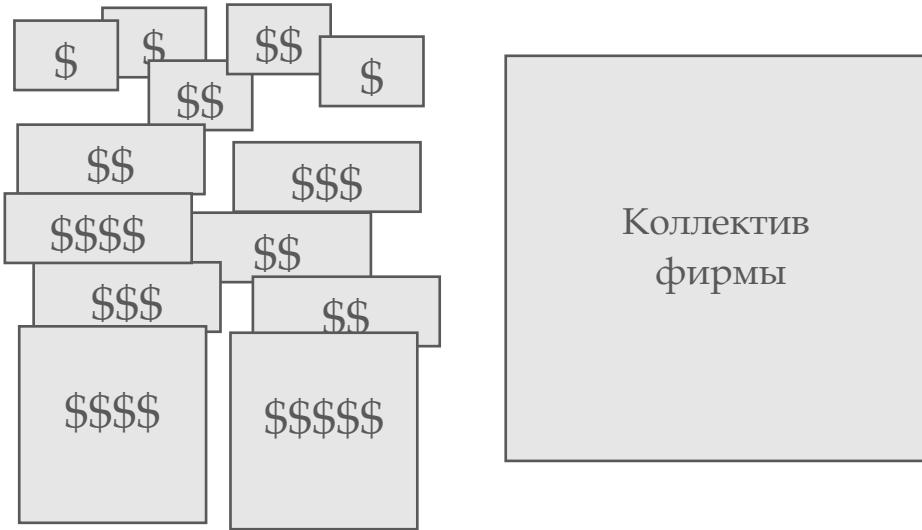


Рис. 4.13. Выбор заказов

В этой задаче размер рюкзака выражен через трудоемкость фирмы b , а предметы, которые необходимо вместить в рюкзак, – это заказы, каждый из которых обладает своей трудоемкостью a_i и ценностью c_i ($i = \overline{1, n}$). Необходимо выбрать проекты, которые принесут наибольшую прибыль.

Классический вид математической модели:

$$W = \sum_{i=1}^n c_i x_i \rightarrow \max$$

$$\sum_{i=1}^n a_i x_i \leq b$$

$$x_i \in \{0,1\}$$

5. Задача об оптимальном портфеле

Постановка задачи

У инвестора имеется 3 актива с ожидаемыми доходностями $m_1 = 11\%$, $m_2 = 15\%$ и $m_3 = 8\%$. Матрица ковариаций² доходностей имеет вид:

$$V = 10^{-5} \begin{bmatrix} 15 & 5 & -7 \\ 5 & 25 & -3 \\ -7 & -3 & 10 \end{bmatrix}$$

Ожидаемая доходность портфеля $m = 12\%$ ³ (таблица 5.1).

Таблица 5.1

Исходные данные

m_i	m	V_{ij}	ЦБ1	ЦБ2	ЦБ3
11%	12%	ЦБ1	15	5	-7
15%		ЦБ2	5	25	-3
8%		ЦБ3	-7	-3	10

Найти оптимальный портфель по модели Марковица. Необходимо набрать такой портфель бумаг, чтобы их совместная ковариация была минимальной.

В качестве искомых переменных возьмем оптимальные доли активов в портфеле (x_1, x_2, x_3).

Целевая функция V представляет суммарную ковариацию ценных бумаг, которая стремится к минимуму. Это позволяет ценным бумагам скомпенсировать друг друга в случае падения одной из них.

$$V(x_i) = \sum_{i=1}^n \sum_{j=1}^n V_{ij} x_i x_j \rightarrow \min$$

² Ковариация – зависимость поведения двух функций. Если две функции ведут себя одинаково (одновременно возрастают или убывают), то присутствует высокий коэффициент ковариации; если функции ведут себя противоположно, то между ними – существенная отрицательная ковариация; если между функциями не устанавливается никакой зависимости, то ковариация равна нулю.

³ Мы прописываем ожидаемую доходность портфеля, а не ищем максимальную доходность. Зачем так делается? Больше прибыль – большие риски. Большую прибыль могут обеспечить только нестабильные бумаги. Поэтому здесь должен быть набор бумаг, которые будут стабильны. Доходность портфеля ограничивается в разумных размерах

Сумма всех переменных равна единице, потому что они представляют доли портфеля:

$$\sum_{i=1}^n x_i = 1$$

А суммарная доходность акций должна быть равна ожидаемой:

$$\sum_{i=1}^n m_i x_i = m$$

Описанная задача является задачей нелинейного программирования, так как в целевой функции присутствует произведение переменных. Симплекс-метод здесь применить невозможно, поэтому мы воспользуемся методом градиента.

Нелинейное программирование

Линейное программирование работает с поверхностями, которые образованы линиями (рис. 5.1). А целевая функция решаемых задач имеет линейный вид:

$$W = \sum_{i=1}^3 c_i x_i = c_1 x_1 + c_2 x_2 + c_3 x_3$$

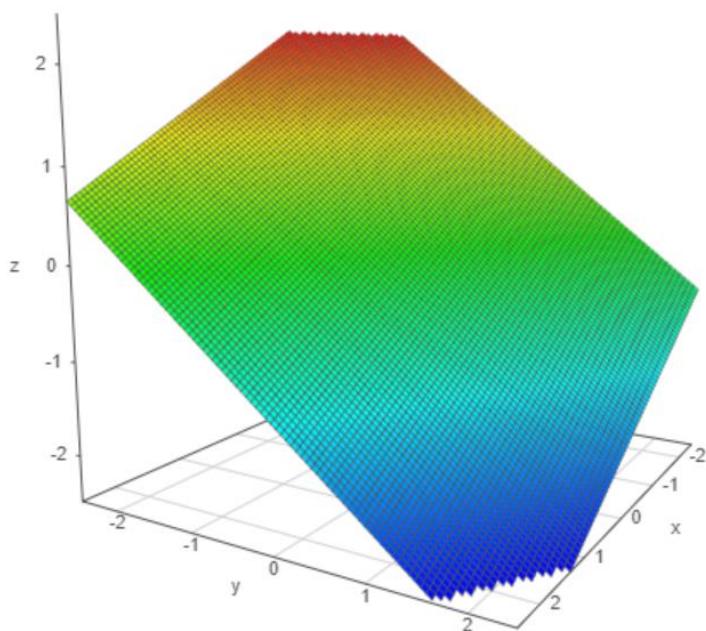


Рис. 5.1. Поверхность, где работает линейное программирование

Как только целевая функция принимает нелинейный вид, например, появляются степени (5.1), логарифмы (5.2) или переменные находятся в показателе (5.3), то ровная поверхность начинает изгибаться (рис. 5.2).

$$W = c_1 x_1^3 + c_2 x_2^2 + c_3 x_3^3 \quad (5.1)$$

$$W = c_1 x_1 + c_2 x_1 x_2 + \ln x_3 \quad (5.2)$$

$$W = c_1^{x_1} c_2^{x_2} \quad (5.3)$$

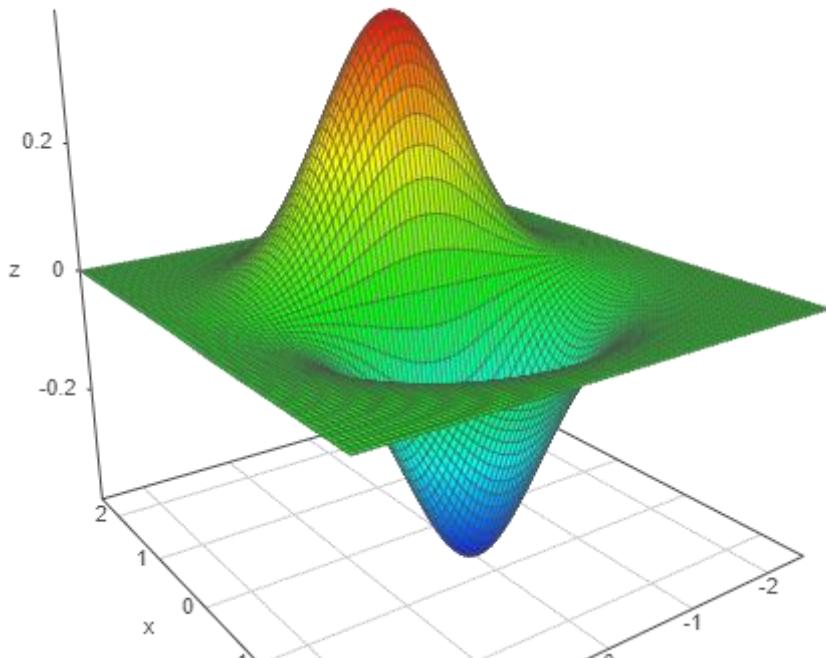


Рис. 5.2. Поверхность, где работает нелинейное программирование

Для решения таких задач симплекс-метод не может использоваться, а применяется, например, метод обобщенного приведенного градиента (ОПГ).

Обобщенный приведенный градиент является разновидностью методов градиента. **Градиент** – это вектор, который в любой точке пространства, направлен всегда в сторону максимального возрастания или убывания функции.

При решении задач нелинейного программирования методом градиента мы задаем начальное значение на поверхности и из него строим градиент: смотрим, в какую сторону функция возрастает быстрее всего. Переходим в точку, на которую указал градиент, и строим новый градиент из нее и т.д. (рис. 5.3).

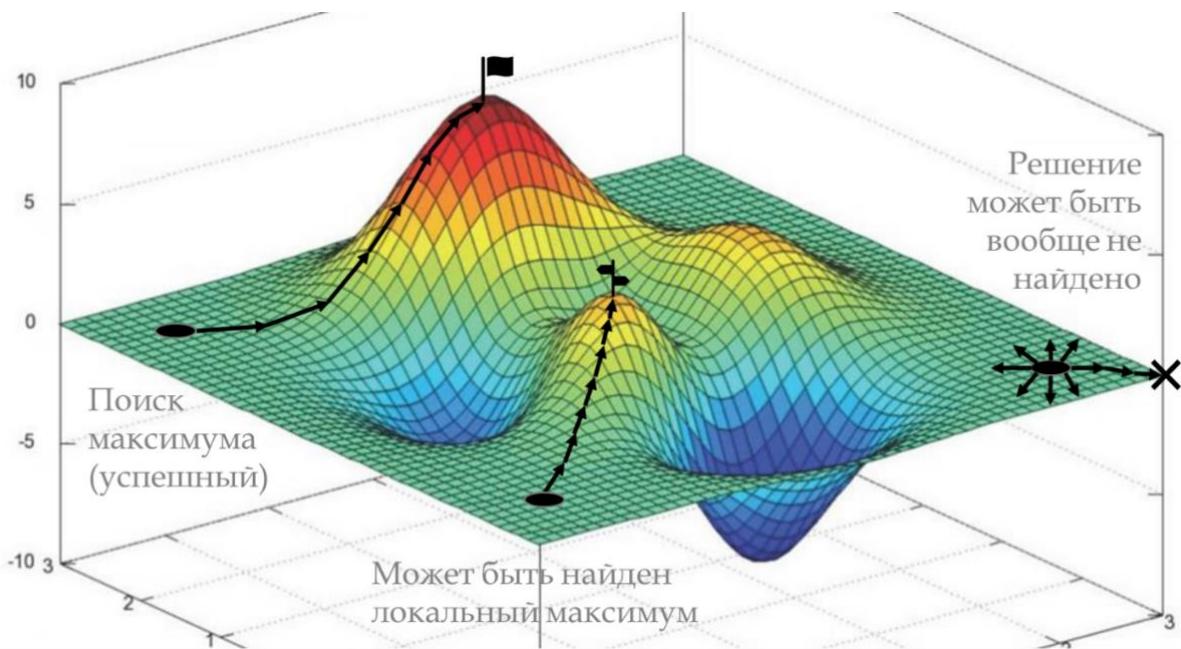


Рис. 5.3. Метод градиента

Метод градиента универсальный, однако имеет ряд недостатков. Главный – зависимость от начальной точки. На рис. 5.3 видно, что в зависимости от начального значения мы можем успешно найти глобальный максимум, или ложный локальный максимум или вообще ничего не найти. Поэтому на практике, применяя этот метод, берут сетку начальных значений: решают задачу из нескольких точек.

Результаты решения задачи об оптимальном портфеле с помощью метода обобщенного приведенного градиента (ОПГ) представлены ниже.

Целесообразнее всего инвестору сформировать портфель ценных бумаг в пропорциях:

$$(x_1 = 28,5\%; \quad x_2 = 45\%; \quad x_3 = 26,5\%)$$

В этом случае зависимость поведения на рынке всех трех бумаг будет минимальна ($V_{min} = 6,45 \times 10^{-5}$). **Околонулевое** значение суммарной ковариации говорит нам о том, что финансовый рынок сбалансирован. Нельзя подобрать такой портфель, который бы без рисков возрастил.

ЧАСТЬ II. ТЕОРИЯ ГРАФОВ

Основные определения теории графов

Теория графов – один из обширнейших разделов в дискретной математике.

Первой задачей в данной области считается задача о мостах, которая появилась благодаря немецкому математику Леонарду Эйлеру, который большую часть жизни провел в России и проработал в Санкт-Петербургской академии наук, и кайзеру Вильгельму, германскому императору и королю Пруссии.

Однажды знаменитый немецкий математик Леонард Эйлер гулял по г. Кенигсберг и задумался - сможет ли он построить маршрут своей прогулки так, чтобы выйти из дома, пройти по всем 7 мостам города по одному разу и вернуться домой (рис. 6.1). Задачу Леонарда Эйлера никто решить не мог до определенного момента.

Радикальное решение нашел кайзер Вильгельм: он взял перо и чернила и написал «Приказываю построить восьмой мост». Этот мост до сих пор существует в г. Калининград и называется мост Кайзера.

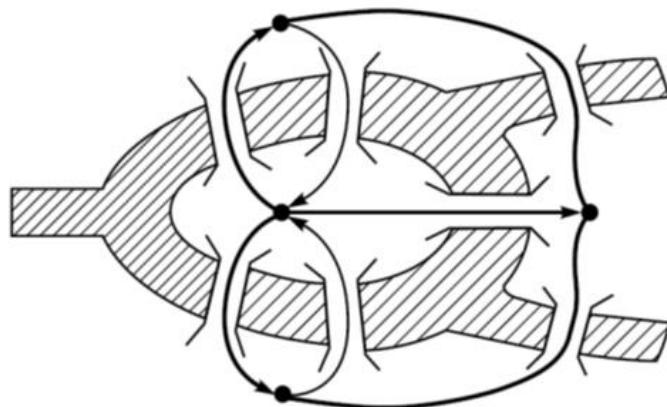


Рис. 6.1. Задача Эйлера о 7 мостах

Так в 1736 году была сформулирована первая задача теории графов и тут же доказано, что такой маршрут проложить невозможно.

Для изучения теории графов необходимо рассмотреть понятийный аппарат, так как термины в данном разделе математики специфичные.

Граф – это совокупность двух конечных множеств: множества вершин X и множества ребер U .

Если задавать граф в *аналитическом виде*, то он будет выглядеть следующим образом:

$$G = (X, U)$$

Вершины и ребра характеризуются выражениями:

$$X = \{x_1, x_2, x_3 \dots x_7\}$$

$$U = \{(x_1, x_1), (x_1, x_2), (x_2, x_3), \dots\}$$

Ребра в графе могут соединять разные вершины, могут соединять одну и ту же вершину, тем самым образовывая петлю, могут не соединять вершины вообще (тогда эти вершины называют несвязанными). Также ребро может быть ориентированным: у него задано направление движения. На графике его отображают с помощью знака стрелочки. В этом случае ребро называют дугой. Стоит отметить, что дуги могут быть получены как из ребер, так и из петель (рис. 6.2).

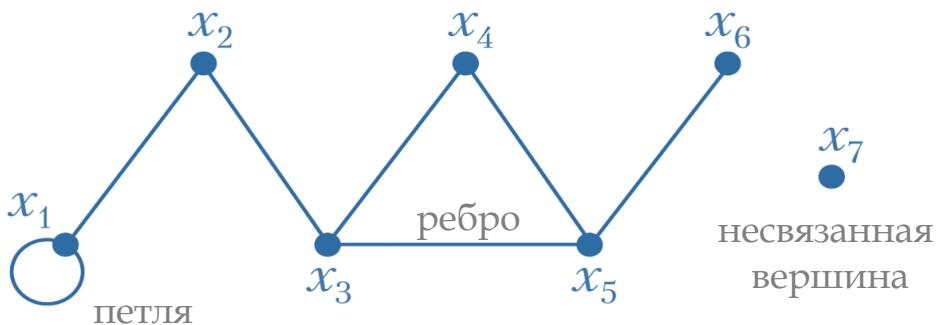


Рис. 6.2 Понятийный аппарат теории графов

В графике можно выделить цепи. *Цепь* – это маршрут по неповторяющимся вершинам: (x_1, x_2, x_3, x_4) . У цепи есть некое начало и конец, и если они совпадают, то эта цепь – цикл. Иными словами, *цикл* – это замкнутая цепь: (x_1, x_2, x_3, x_1) (рис. 6.3).

Стоит отметить, что ребра графа могут иметь вес. Он обозначается в виде цифры (или переменной), стоящей рядом с ребром. Данная цифра может задавать либо время поездки, либо длину дороги, либо стоимость перевозки по этому маршруту.

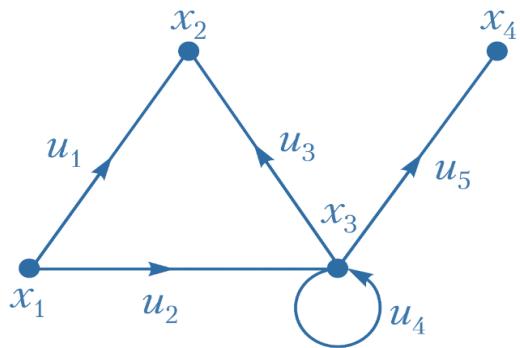


Рис. 6.3 Ориентированный граф

Мы перечислили базовые термины из теории графов. Сами графы бывают различных типов (рис. 6.4). Самый популярный в настоящее время граф – это **«Дерево»**. Мы его видим в различных задачах анализа данных, в банковской сфере, в алгоритмах из других сфер, в том числе в алгоритмах принятия решений.

Также существует тип графа – **«Сеть»**. «Сеть» - граф, как правило, ориентированный, где есть один вход и один выход. Например, этот граф можно интерпретировать как сеть автомобильных дорог или сеть передачи данных.

Графы бывают **двудольными**, то есть состоящими из двух частей.

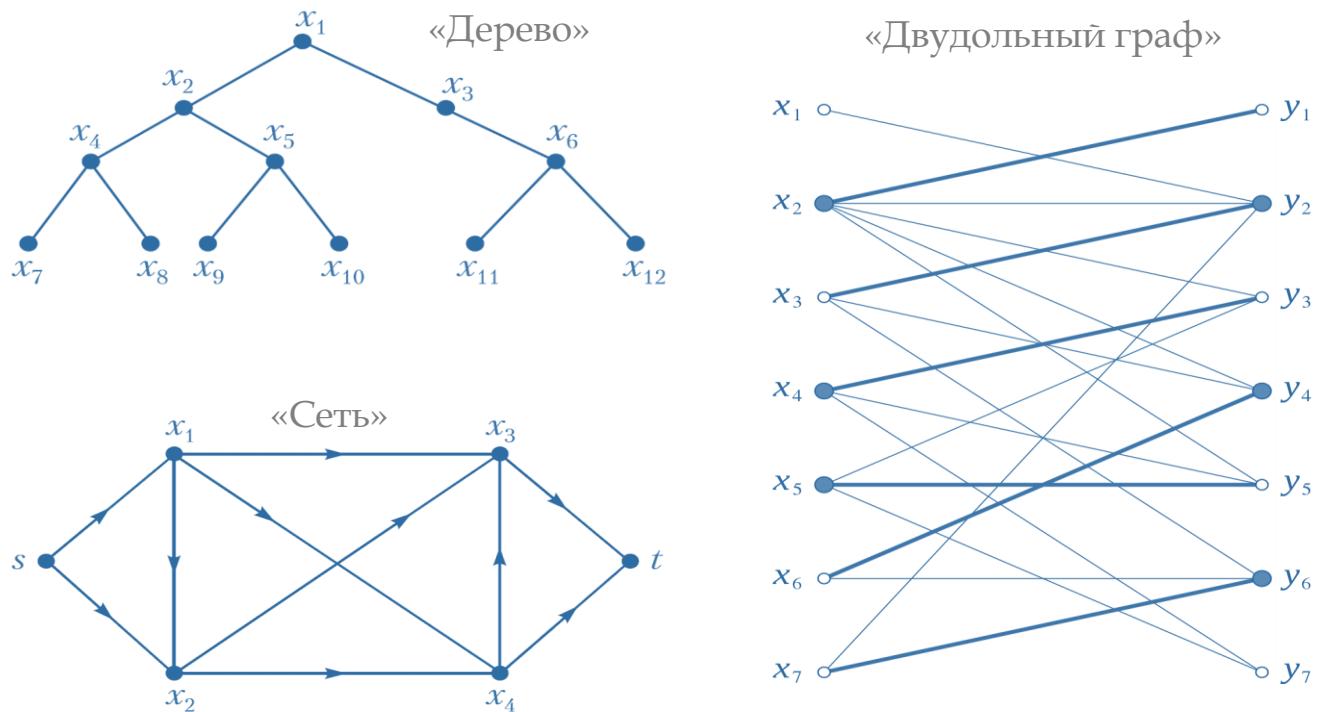


Рис. 6.4 Основные типы графов

6. Задача о кратчайшем пути

Какие задачи можно решать с помощью графов? Самая очевидная – *задача о кратчайшем пути*. Эта задача реализуется в автомобильных навигаторах. Графами можно представить любую сеть дорог, где перекрестки – вершины, а проезжая часть между перекрестками – ребра графа. В данном типе задач необходимо построить кратчайший маршрут из одной точки транспортной сети в другую (рис. 6.5).

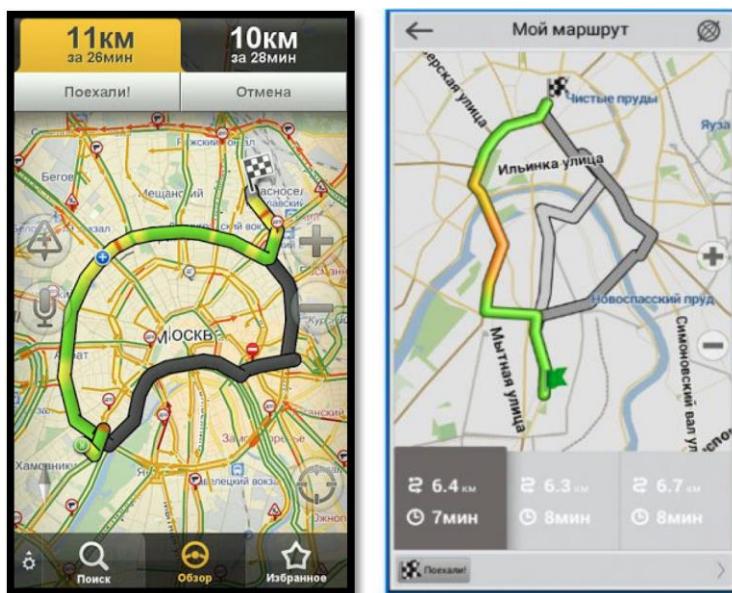


Рис. 6.1. Автомобильный навигатор

Критериями могут служить время перемещения, длина пройденного пути, стоимость проезда (для платных дорог), вероятность затора (количество полос, перекрестков и др.)

Рассмотрим задачу поиска кратчайшего пути в центре Москвы (с района Якиманка на Китай-город), которая представлена на рис. 6.6. В качестве вершин графа возьмем перекрестки, в качестве дуг – дороги между перекрестками. Все ребра будут обозначать приблизительное время проезда по участку дороги. Нам необходимо построить самый быстрый маршрут из пункта отправления А в пункт назначения В. На данном графе можно заметить и ребра, и дуги. Введение ориентированности связано с односторонним движением. Также особенной является вершина №7, в которую можно войти, но нельзя выйти.



Рис. 6.6. Условие задачи

Алгоритм Дейкстры

По алгоритму Дейкстры необходимо выполнение следующих действий:

1. Строим все возможные маршруты из начальной точки А в ближайшие точки: в данной задаче возможен только один путь из-за одностороннего движения в вершину №1. Находим длительности пути: ребро, соединяющее вершины А и №1, равняется 8 мин. Вершину №1 отмечаем как посещенную и рядом с ней подписываем значение 8.

2. Из первой посещенной вершины строим возможные пути перемещения в соседние: здесь так же возможен только один путь к вершине №2. Суммируем время движения: прибавляем 4 мин и получаем 12 мин. Вершину №2 отмечаем как посещенную и рядом с ней подписываем значение 12.

3. Из вершины №2 переходим в вершины №3 и №11. Записываем суммарное время перемещения рядом с вершинами: 21 мин и 34 мин соответственно.

4. Из вершины №11 строим маршрут к вершинам №8 и №10, записываем напротив них суммарное время проезда 58 мин и 46 мин соответственно.

5. Аналогично из вершины №3 мы переходим в вершины №4 и №8. Обращаем внимание на то, что вершину №8 мы уже посещали и продолжительность

пути к ней составила 58 мин. Однако, через вершину №3 найден более быстрый маршрут, поэтому вместо 58 мин записываем 41 мин.

6. Из вершины №10 переходим в №9 и №8, которая уже посещена. Новое время перехода в №8 составляет 58 мин, что является плохим результатом, по сравнению с имеющимся. Оставляем время перехода в вершину №8 равное 41 мин. К этому моменту у нас сформировано три типа вершин: не посещенные, вершины, где рассмотрены не все маршруты, и вершины, где рассмотрены все маршруты.

7. Аналогичный алгоритм применяем для вершины №4 (в том числе необходимо рассмотреть все пути выезда).

8. Ищем переходы для вершины №8. Найден более короткий маршрут в вершину №9, поэтому 62 мин заменяем на 57 мин.

9. Время переходов из вершины №5 больше, чем уже имеющееся в №6 и №8 вершинах, поэтому значения рядом с вершинами не изменяем.

10. В 6 вершине картина аналогичная. Вершина №7 тупиковая, из нее отсутствуют выходящие дуги.

11. Наконец переход из вершины №9 в пункт назначения В завершает работу алгоритма в связи с тем, что закончились вершины, в которые был осуществлен вход и выход (за исключением выхода из вершины В). В результате работы этого алгоритма мы получили схему как на рис. 6.7.

Рядом с пунктом В, таким образом, появляется 71 мин – это значение и будет минимальным. Для нахождения оптимального маршрута на графе необходимо совершить обратный проход по алгоритму, двигаясь из пункта В каждый раз в сторону вершины с наименьшим значением суммарного времени, то есть:

$$B(71) \rightarrow №9(57) \rightarrow №8(41) \rightarrow №3(21) \rightarrow №2(12) \rightarrow №1(8) \rightarrow A(0)$$

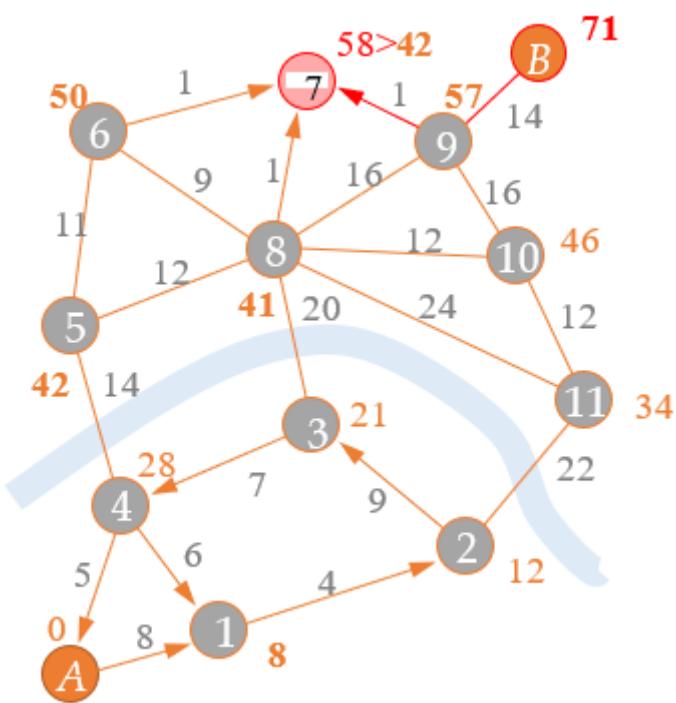


Рис. 6.7. Решение задачи (часть 1)

Таким образом, кратчайший путь из пункта А в пункт В пролегает через перекрестки №1, №2, №3, №8, №9. Время в пути составит 71 мин. Сам маршрут показан на рис. 6.8.

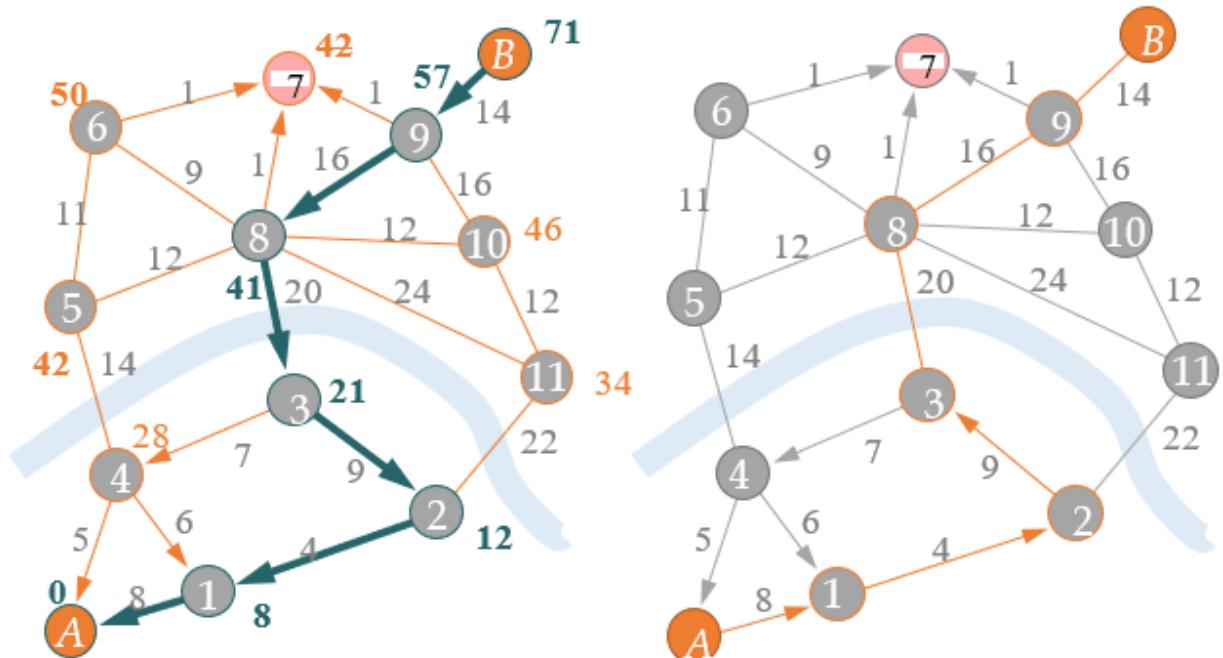


Рис. 6.8. Решение задачи (часть 2)



Эдсгер Вибе Дейкстра (11 мая 1930, Роттердам, Нидерланды - 6 августа 2002, Нюэнен, Нидерланды), повлиял на всю мировую ИТ-сферу - автор языка Алгол, алгоритма «Дейкстры», один из первых профессиональных программистов. Во второй половине 1950-х годов в поисках путей оптимизации разводки печатных плат разработал алгоритм поиска кратчайшего пути на графе, ставший известным как «алгоритм Дейкстры». В 1957 году женился, в графе «профессия» анкеты, которую положено заполнять при бракосочетании, написал «программист» - и его заставили переписывать документы, заявив, что такой профессии не существует, в результате пришлось указать «физик-теоретик». В 1970-е годы в Нидерландах вместе с Тони Хоаром и Николаусом Виртом разработал основные положения структурного программирования. Не пользовался телевизором и мобильным телефоном, предпочитал ручку и блокнот компьютеру. Однажды от руки написал 61 письмо с благодарностью за издание своего юбилейного сборника. Один из редких ученых, который получил премию имени самого себя, причем третьим по счету.

Решение как задачи линейного программирования

Известно, что ко всем задачам существуют различные подходы к решению. Алгоритм Дейкстры – это *алгоритмический* метод поиска кратчайшего пути на графике. Теперь рассмотрим *математический* численный метод решения данной задачи. Преобразуем график в матрицу, где вершины – номера столбцов/строк, вес ребер – элементы внутри самой матрицы.

Пусть дан график $G = (V, A)$, где V – множество вершин графа, A – множество ребер графа.

Вес ребер графа w_{ij} , задается матрицей смежности:

$$W = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1n} \\ w_{21} & w_{22} & \dots & w_{2n} \\ \dots & \dots & \dots & \dots \\ w_{n1} & w_{n2} & \dots & w_{nn} \end{bmatrix}$$

где n – количество вершин графа.

Матрица смежности может строиться на основании веса ребер, но может и иметь бинарный вид, то есть показывать наличия или отсутствия дороги (ребра в графике) (таблица 6.1).

Таблица 6.1

Матрица смежности графа $W = \{w_{ij}\}$

w_{ij}	A	1	2	3	4	5	6	7	8	9	10	11	B
A	∞	8											
1		∞	4										
2			∞	9									22
3				∞	7					20			
4	5	6			∞	14							
5					14	∞	11			12			
6						11	∞	1	9				
7							∞						
8				20		12	9	1	∞	16	12	24	
9								1	16	∞	16		14
10									12	16	∞	12	
11		22							24		12	∞	
B										14			∞

Наша задача – найти по каким ребрам графа x_{ij} проходит **кратчайший маршрут** из вершины A в вершину B .

Также создадим матрицу искомого маршрута $\{x_{ij}\}$, где 1 будет обозначать наш выбор данного ребра. Иными словами, это матрица переменных. Например она может выглядеть так как представлена в таблице 6.2.

Таблица 6.2

Матрица искомого маршрута $\{x_{ij}\}$

x_{ij}	A	1	2	3	4	5	6	7	8	9	10	11	B
A													
1													
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													
B													

Таким образом, мы сводим нашу задачу к задаче **линейного программирования**. Для решения зададим **целевую функцию – минимальную** суммарную **длину** пути:

$$F = \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_{ij} \rightarrow \min$$

То есть мы суммируем произведения единиц из матрицы искомого маршрута и значений из матрицы смежности. Наша целевая функция должна стремиться **к минимуму**.

Теперь необходимо задать ограничения:

a) Маршрут должен начинаться в пункта А и заканчиваться в пункт В:

$$\sum_{j=1}^n x_{Aj} = \sum_{i=1}^n x_{iB} = 1$$

Это значит, что в строке А и столбце В обязательно должна стоять единица.

b) Переход должен осуществляться только по существующим ребрам:

$$x_{ij} \leq w_{ij}, \forall i, j = \overline{1, n}$$

Единицу мы имеем право ставить только в тех ячейках, в которых в матрице смежности стоит число, отличное от 0. Если стоит 0, то ребро отсутствует.

c) Ограничение на связность маршрута – **вход** в вершину всегда должен **соответствовать выходу** из нее (сумма по строке j равна сумме по столбцу j), кроме начала A и конца B маршрута:

$$\sum_{i=2}^{n-1} x_{ij} = \sum_{i=2}^{n-1} x_{ji}, \quad j = \overline{2, n-1}$$

d) Вход и выход в каждую вершину производится **не более одного раза**, чтобы не возникало петель:

$$\begin{aligned} \sum_{i=1}^n x_{ij} &\leq 1, \quad \sum_{j=1}^n x_{ij} \leq 1 \\ j &= \overline{1, n}, \quad i = \overline{1, n} \\ x_{ij} &\in \{0, 1\} \end{aligned}$$

Как мы видим, данная задача имеет относительно много ограничений, причем эти ограничения разные по своей сути.

Данная задача решается **симплекс-методом**, как и любая другая задача линейного программирования. В результате мы получаем следующий результат (таблица 6.3):

Таблица 6.3

Результат решения симплекс-методом

w_{ij}	A	1	2	3	4	5	6	7	8	9	10	11	B
A		1											
1			1										
2				1									
3									1				
4													
5													
6													
7													
8										1			
9													1
10													
11													
B													

Целевая функция $F = 71$ мин. Ответ совпадает с решением алгоритмом Дейкстра.

Это обозначает, что мы последовательно проходим вершины A , №1, №2, №3, №8, №9, B в течение 71 мин.

Данный алгоритм заложен в работу, например, Яндекс.Навигатор (вершины у него не перекрестки, а адреса напротив каждого дома). Сложности в работе возникают с поворотами и разворотами, так как адреса по разные стороны дороги отличаются. Также Яндекс.Навигатор учитывает скопление машин, позволяя их объезжать.

7. Задача коммивояжера

В 1859 г. ирландский математик **Ульям Гамильтон** предложил задачу «Кругосветное путешествие». Каждой из 20 вершин графа соответствует один из городов мира. Необходимо проезжая по ребрам графа, посетить каждый город только один раз и вернуться назад. Такой маршрут существует, причем их несколько. Поэтому задача была усложнена поиском кратчайшего расстояния. Сам маршрут, представляющий собой замкнутую цепь, проходящую через все вершины графа, получил название «гамильтонов цикл», представленный на рис. 7.1.

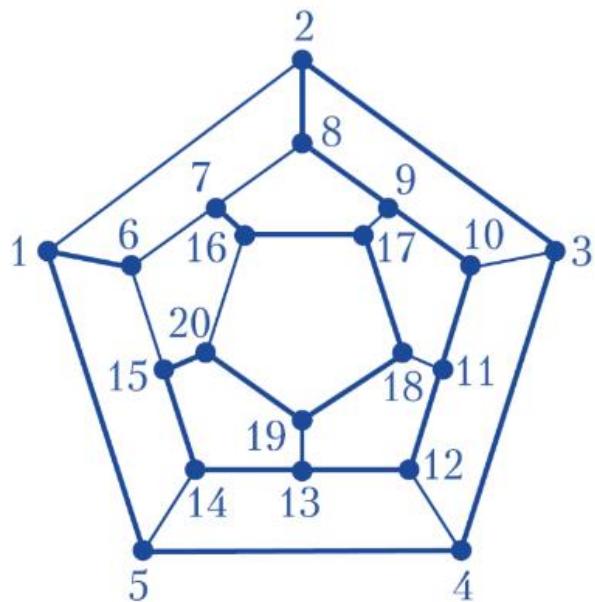


Рис. 7.1. Гамильтонов цикл

Впервые задачу коммивояжера (путешествующего торговца) сформулировали в следующей постановке: «Как объехать все города Германии и вернуться в исходную точку за минимальное время?» (рис. 7.2). Решить задачу направленным перебором нельзя, возможно только через комбинаторный перебор. Она относится к классу NP-трудных задач комбинаторной математики

трансвычислительной сложности: уже при 66+ числе городов время перебора вариантов приближается к нескольким миллиардам лет.



Рис. 7.2 Задача коммивояжера

Данная задача часто применяется в логистике, например, при доставке из магазинов (рис. 7.3). Рассмотрим одну из вариаций подобных задач.

Пусть имеется склад некоторых товаров (офис службы доставки). На территории города расположено несколько адресатов, которым необходимо доставить товар. Каким образом курьеру избрать маршрут обьезда заказчиков, чтобы развести все заказы и вернуться в офис за минимальное время.

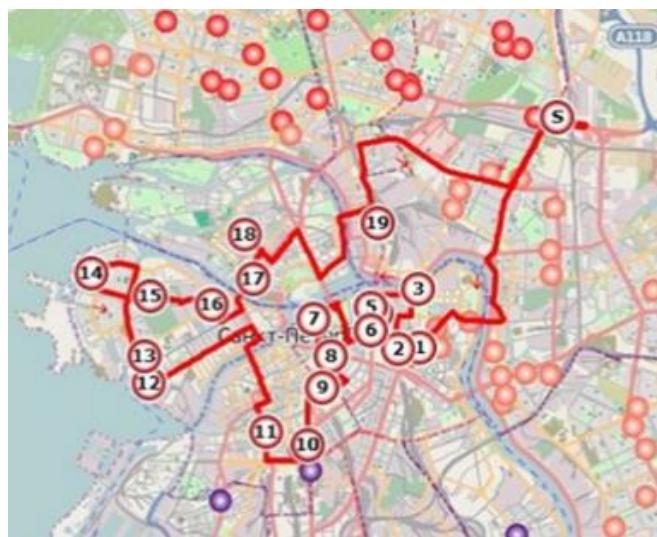


Рис. 7.3. Применение задачи коммивояжера в логистике

Постановка задачи

В математической постановке задача может звучать следующим образом:
«Необходимо найти на графике все гамильтоновы циклы минимальной длины».

Представим маршруты переездов между адресами в виде графа (ориентированные дуги обозначают последовательность посещений). Возьмем, для примера, график из предыдущей главы с небольшими изменениями (рис. 7.4).

Вершина А является началом и концом маршрута (офис доставки). Вес ребра обозначает время переезда между адресатами. Необходимо построить замкнутый маршрут, проходящий через все вершины графа не более одного раза и имеющий минимальную длину.

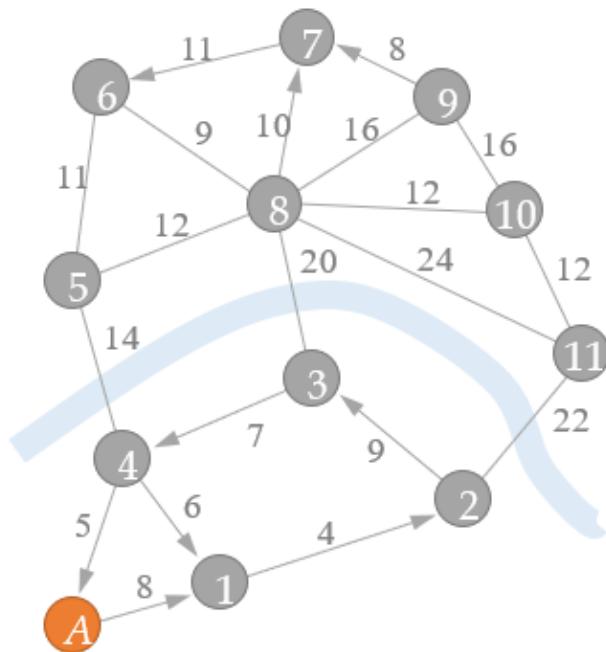


Рис. 7.4. Условие задачи

Решение как задачи дискретного комбинаторного программирования

Математическая модель остается схожей с моделью из прошлой задачи. Пусть дан график $G = (V, A)$, где V - множество вершин графа, A - множество ребер графа.

Вес ребер графа w_{ij} , задается матрицей смежности:

$$W = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1n} \\ w_{21} & w_{22} & \dots & w_{2n} \\ \dots & \dots & \dots & \dots \\ w_{n1} & w_{n2} & \dots & w_{nn} \end{bmatrix}$$

где n - количество вершин графа.

Необходимо найти замкнутый цикл минимальной длины.

Для решения составляем матрицу смежности (таблица 7.1).

Таблица 7.1

Матрица смежности графа $W=\{w_{ij}\}$

w_{ij}	A	1	2	3	4	5	6	7	8	9	10	11	B
A		8											
1			4										
2				9								22	
3					7				20				
4	5	6				14							
5					14		11		12				
6						11			9				
7							11						
8				20		12	9	10		16	12	24	
9									8	16		16	14
10										12	16		12
11			22							24		12	
B		8											

В данном случае, мы будем использовать другую вариацию матрицы маршрута. Так как нам необходимо обойти все вершины, то можно построить для переменных вектор-столбец (таблица 7.2), в который мы будем записывать номер вершины в определенной последовательности. В конце стоит та же вершина, что и в начале, так как нам необходимо «замкнуть» маршрут.

Вектор-столбец посещения вершин

x_i
0
1
2
11
10
8
...
0

Для решения зададим **целевую функцию** – минимальную суммарную длину всех дуг пути:

$$F = \sum_{i=1}^{n-1} w_{x_i x_{i+1}} \rightarrow \min$$

Наша целевая функция должна стремиться к минимуму.

Теперь необходимо задать **ограничения**:

1. Маршрут должен **начинаться и заканчиваться в п. А**:

$$x_0 = x_n = A$$

2. Переход должен осуществляться **только по существующим** ребрам:

$$w_{x_i x_{i+1}} > 0, \quad i = \overline{0, n-1}$$

3. Вершины **не должны повторяться**:

$$x_i \neq x_j, \quad \forall i \neq j$$

4. Искомые переменные **целочисленные**:

$$x_i \in N$$

Решить данную задачу симплекс-методом и методом ОПГ не представляется возможным, поэтому используем **эволюционный метод**. Результат решения задачи эволюционным методом в MS Excel (комбинаторный перебор вариантов) представлен на рис. 7.8.

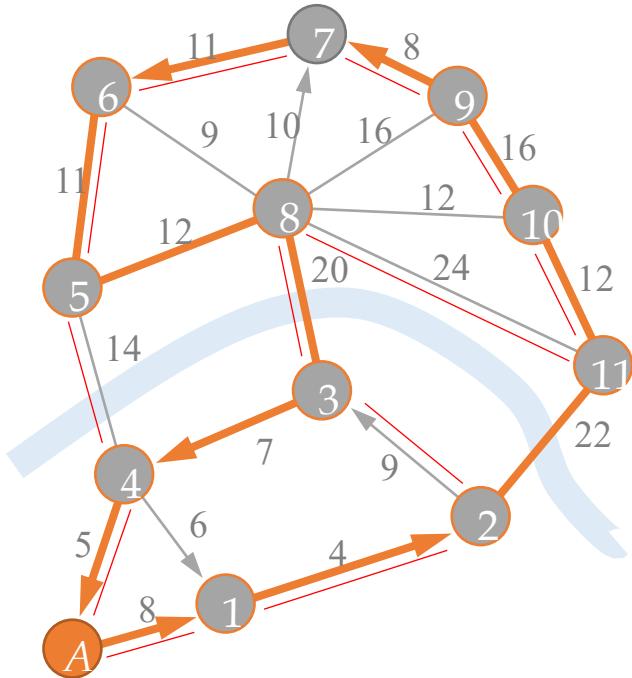


Рис. 7.8. Результат эволюционного метода

Кратчайший маршрут обьезда всех адресатов курьером службы доставки:

$$\begin{aligned}
 A &\rightarrow \text{№}1 \rightarrow \text{№}2 \rightarrow \text{№}11 \rightarrow \text{№}10 \rightarrow \text{№}9 \rightarrow \\
 &\quad \text{№}7 \rightarrow \text{№}6 \rightarrow \text{№}5 \rightarrow \text{№}8 \rightarrow \text{№}3 \rightarrow \text{№}4 \rightarrow A
 \end{aligned}$$

Время в пути составляет 136 мин.

Существует также неоптимальный маршрут продолжительностью 142 мин:

$$\begin{aligned}
 A &\rightarrow \text{№}1 \rightarrow \text{№}2 \rightarrow \text{№}3 \rightarrow \text{№}8 \rightarrow \text{№}11 \rightarrow \text{№}10 \rightarrow \\
 &\quad \rightarrow \text{№}9 \rightarrow \text{№}7 \rightarrow \text{№}6 \rightarrow \text{№}5 \rightarrow \text{№}4 \rightarrow A
 \end{aligned}$$

8. Задача о максимальном потоке

Графы можно рассматривать не только как статичные объекты, в которых можно искать разнообразные маршруты, но и как динамические объекты, внутри которых может что-то изменяться.

В качестве таких графов можно рассмотреть, например, сети, по которым протекают некоторые потоки.

Сеть – это конечный граф без циклов и петель, ориентированный в одном направлении от входа к выходу (рис. 8.1).

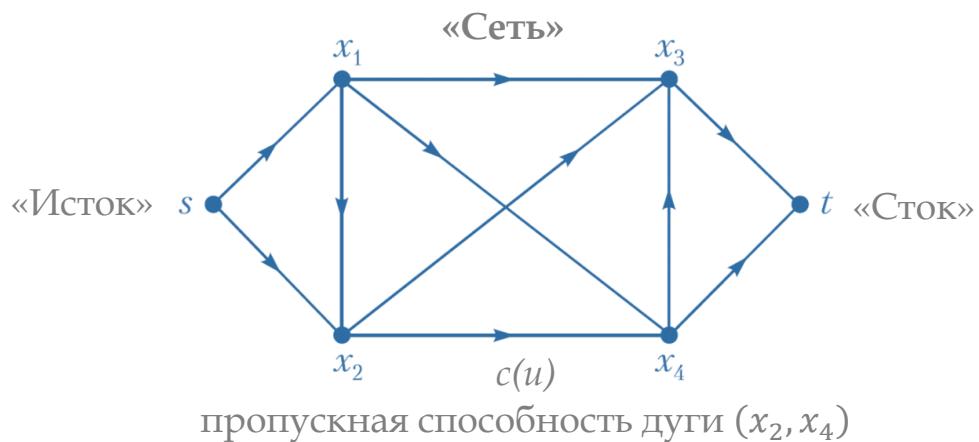


Рис. 8.1. Граф типа «сеть»

В сети всегда имеется «исток» (вход в граф) и «сток» (выход из графа). Ребра в сети всегда ориентированы от истока к стоку и носят название «дуги», а вершины в сети называются «узлами».

Чтобы проще представить, что такое сеть возьмем древний водопровод – акведук, в котором вода течет с гор в долину под действием силы тяжести. Исток ее находится в горном роднике, вода бежит по трубам – дугам сети, заполняя каждую из них целиком или на какую-то часть, и выливается в стоке - на центральной площади города.

Диаметр трубы, по которой течет вода, называется пропускной способностью дуги и обозначает количество потока, которая она способна пропустить в единицу времени.

Здесь мы подбираемся ко второму ключевому понятию этого раздела – потоку в сети. Потоком называется количество воды, протекающее по всем трубам водопровода, или выражаясь в терминологии теории графов:

Потоком в сети $S = (N, U)$ от входа $s \in N$ к выходу $t \in N$ называется неотрицательная функция φ , определенная на множестве дуг сети U со следующими свойствами:

1. Величина входящего и выходящего потока в каждой вершине **одинакова** (сколько в вершину затекло воды, столько же и должно вылиться):

$$\sum_{j \in N_i} \varphi^-(i, j) = \sum_{j \in N_i} \varphi^+(j, i)$$

2. Величина потока по каждой дуге **не может превосходить** пропускной способности дуги (нельзя лить воды больше, чем позволяет диаметр трубы):

$$0 \leq \varphi(i, j) \leq c(i, j)$$

где $c(i, j)$ – пропускная способность дуги из i в j .

Например, как показано на рис. 8.2, в вершину x_4 втекает $\varphi^- = 8$ единиц потока. При этом пропускная способность дуги составляет 10 единиц, то есть дуга заполнена не полностью. Поток в вершине никуда не исчезает, и ниоткуда не берется. Поэтому вытекает также $\varphi^+ = 8$ единиц потока, по одной из дуг 3 единицы, по другой – 5. При этом дуга (x_4, x_3) заполнена полностью, а по дуге (x_4, t) остается еще 1 единица незадействованной пропускной способности.

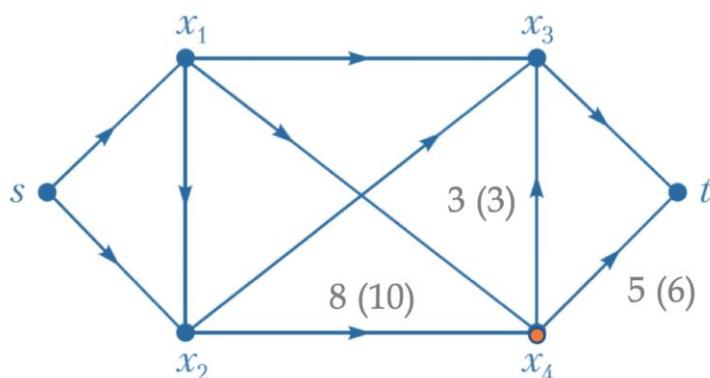


Рис. 8.2. Поток в сети

Для того, чтобы нагляднее понять, что из себя представляет поток в сети, рассмотрим пример сети, представленный на рис. 8.3. На вход s поступает 5

единиц потока, из которых 4 единицы потока уходят вниз по дуге $(s, 3)$ и занимают всю ее пропускную способность. Оставшуюся 1 единицу потока давление выдавливает в верхнюю часть сети по дуге $(s, 1)$.

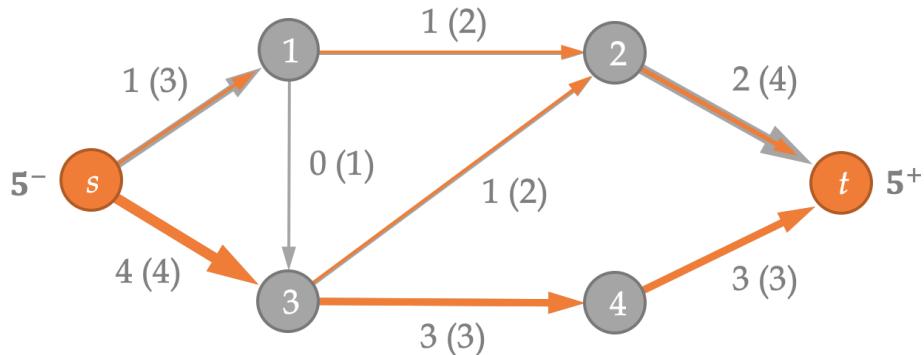


Рис. 8.3. Пример потока в сети

Из четырёх единиц потока к разветвлению 3 подходят все 4, но на рисунке видно, что водопровод сужается и пропускная способность становится равной 3. 3 единицы потока продолжают течь снизу, 1 единица потока выдавливается наверх, не заполняя полностью всю трубу. Первая единица потока продолжает течь наверху, 3 единицы потока внизу продолжают течь понизу. В разветвлении 2 поток из 1 и 3 объединяются, теперь его пропускная способность равна 2 и к финишу (стоку) протекает по альтернативному маршруту.

Возникает вопрос: какую максимальную величину потока способна обеспечить данная сеть?

Постановка задачи

Для этого разберём на примере *задачу о максимальном потоке*, построив транспортную сеть центрального района Москвы в виде дорог. Определим, какое количество автомобилей способна выдержать данная транспортная сеть, представленная на рис. 8.4.

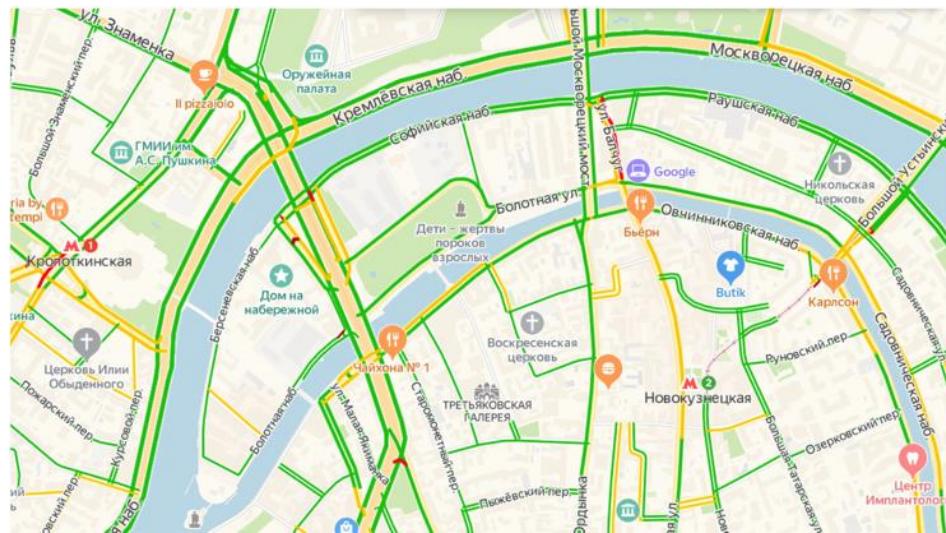


Рис. 8.4. Задача о максимальном потоке

Первоначально мы представим дорожную сеть в виде графа дорог, в котором перекрестки – вершины графа, дороги – соединяющие их дуги. Двустороннее движение в данном случае будем обозначать двумя дугами, так как сеть состоит только из направленных дуг (рис. 8.5).

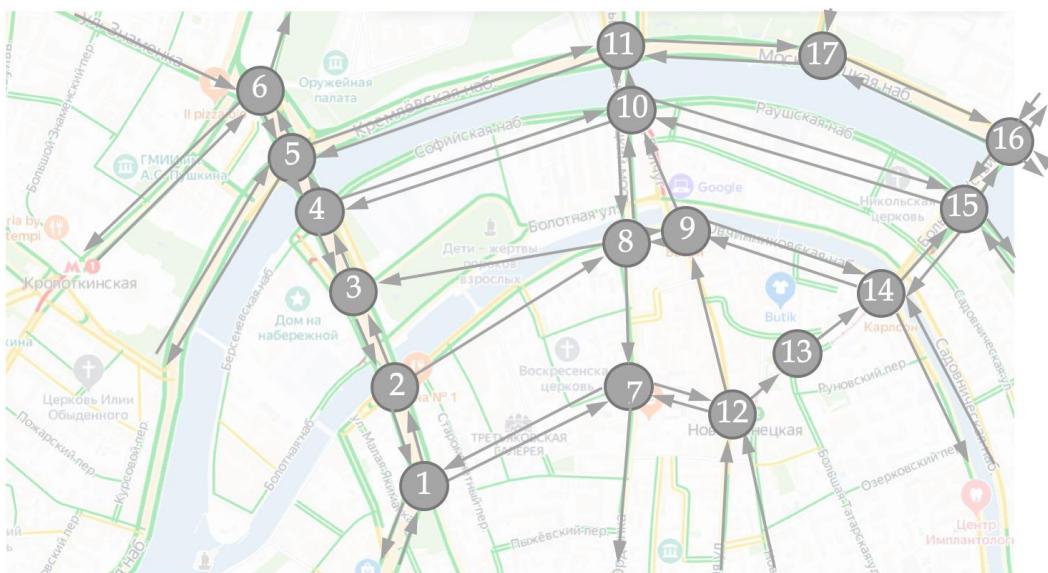


Рис. 8.5. Построение графа

Чтобы граф стал по-настоящему сетью, на наш рисунок необходимо добавить сток и исток. Все дороги, по которым можно заехать, связываются дугами с одной вершиной – исток. Все выезды – дороги, которые ведут на выезд из транспортной сети, привязываем к стоку.

Все входы объединяются в исток, а все выходы – в сток (рис. 8.6.):

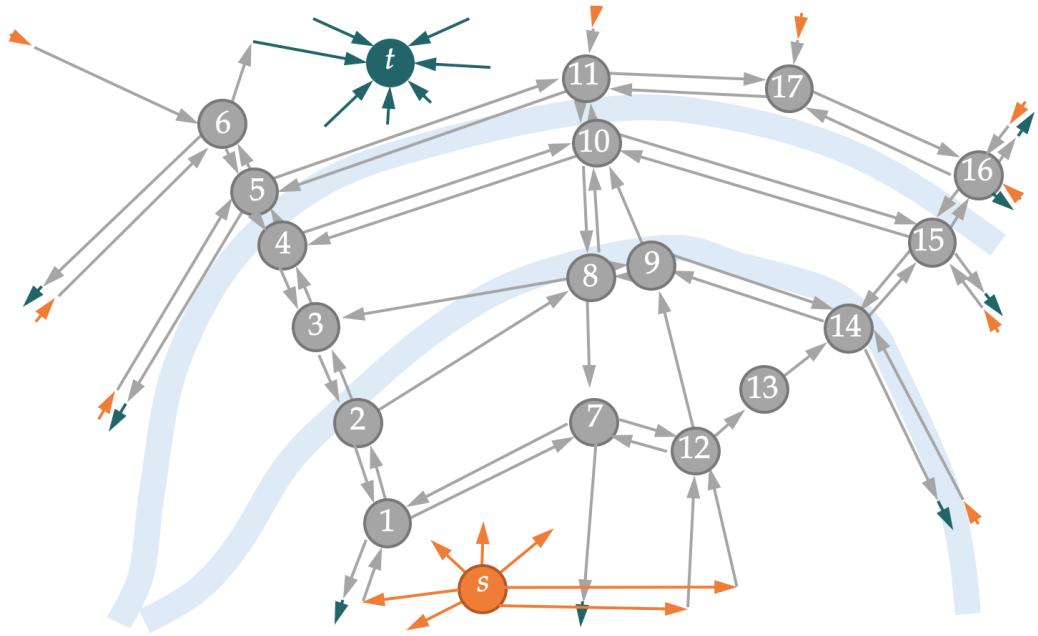


Рис. 8.6. Входы и выходы сети

Было замечено, что в данном районе движение нерегулярное: по утрам движение машин в направлении с юга на север, а по вечерам – с севера на юг. Поэтому данную задачу удобнее разбить на две: посчитать сначала утренние потоки, когда машины едут в одном направлении, а затем вечерние, когда поток едет в другом направлении. Остановимся на утренних потоках, и рассмотрим задачу, в которой мы будем ехать с юга на север (рис. 8.7).

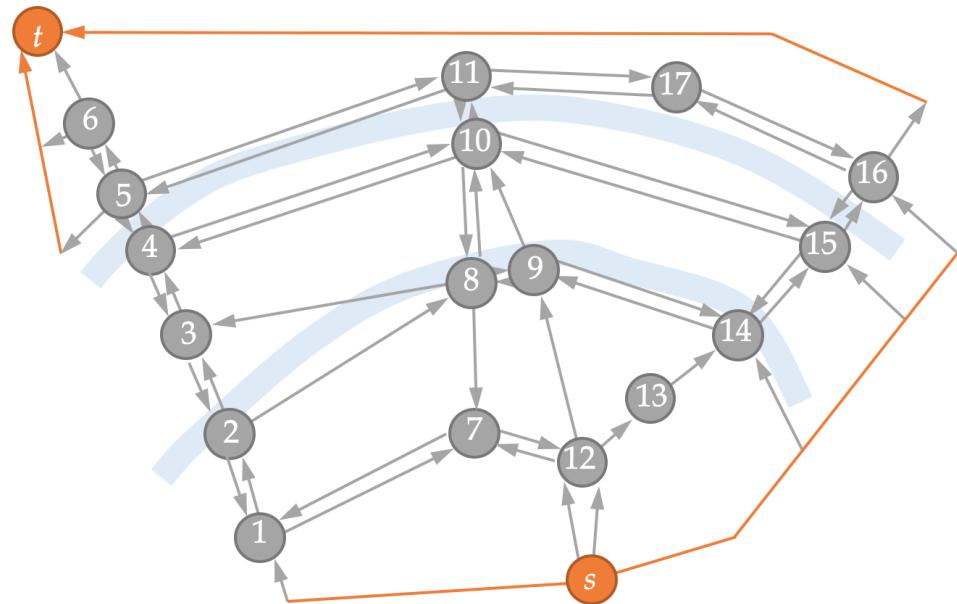


Рис. 8.7. Пропускные способности

В данной задаче у нас есть 6 въездов и 4 выезда, которые мы объединяем во вход и выход. Чтобы рассчитать пропускную способность дороги (количество автомобилей, которые могут проезжать в час по этой дороге), необходимо воспользоваться следующей формулой:

$$c_{ij} = k \times V/L$$

где k – количество полос, V – средняя скорость, L – динамический габарит автомобиля (габарит автомобиля + расстояние до следующей машины)

Получилось, что в данной задаче по одной полосе в среднем могут проезжать 12 000 автомобилей. Умножим на количество дорог и получим пропускную способность = 132 000 автомобилей в час.

Перейдем к *математическому виду* задачи и опишем граф матрицей смежности, у которой на пересечении соответствующих строки и столбца будет стоять пропускная способность дуг $W = \{c_{ij}\}_{18 \times 18}$, представленной в таблице 8.1:

Таблица 8.1

Матрица смежности

c_{ij}	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	t
S	36											36		12	24	24		
1		36					12											
2			36					36										
3				48														
4					48						12							
5						72						24						24
6																		72
7	12												24					
8			48				36		36	48								
9								60		12				12				
10				24				48			60				24			
11					36					24							36	
12						24		24					12					
13														0				
14									12					24				
15										24					36			
16																36	36	
17											36				36			

Нам необходимо найти матрицу такого же размера, в которой на пересечении соответствующих строки и столбца будет стоять величина **максимального потока** φ_{ij} , которую способна выдержать транспортная сеть. Это и будет результатом решения задачи.

Для решения этой задачи существуют математический и алгоритмический подходы. Алгоритмически эта задача решается с помощью алгоритма Форда-Фалкерсона. Математическое решение будет рассмотрено ниже.

Запишем нашу задачу в виде задачи линейного программирования, которое уже рассматривалось ранее.

Исходные данные записаны в виде матрицы смежности, переменными являются величины максимального потока.

Целевая функция – суммарный поток на входе или на выходе должен быть максимальным. Другими словами, чтобы в транспортную сеть въезжало как можно больше машин или по трубопроводам протекало как можно больше воды.

Целевая функция является так же *первым ограничением*:

$$F = \sum_j \varphi_{sj}^+ = \sum_i \varphi_{it}^- \rightarrow \max$$

Второе ограничение:

Для каждой вершины, кроме s и t входной поток должен равняться выходному:

$$\sum_{i,j} \varphi_{ij}^+ = \sum_{i,j} \varphi_{ij}^-, \quad i \neq s, j \neq t$$

Третье ограничение:

Величина потока должна быть неотрицательной и количество проезжающих машин не должно превышать пропускной способности дуги:

$$0 \leq \varphi_{ij} \leq c_{ij}, \quad i, j \in N$$

Результатом решения задачи симплекс-методом – максимальный поток в сети φ_{ij} - какое количество автомобилей должно ехать по определенным дорогам, чтобы полностью загрузить транспортную сеть. Максимальный поток представлен в таблице 8.2.

Таблица 8.2

Результат – матрица потока

c_{ij}	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	t
S	36											24		12	24	24		
1		36					0											
2			36					0										
3				48														
4					48					0								
5						60					0							24
6																		60
7	0												0					
8			12				0		0	0								
9								12		12				0				
10				0			0			12					0			
11					36					0							0	
12						0			24				0					
13													0					
14									0						12			
15										0						36		
16																24	36	
17											24					0		

По таблице видно, что не все дороги оказались востребованными и не все дороги максимально загружены. В ходе решения **симплекс-методом** мы получили количество автомобилей, которое способно проехать по нашей транспортной сети в час, без создания заторов (рис. 8.8):

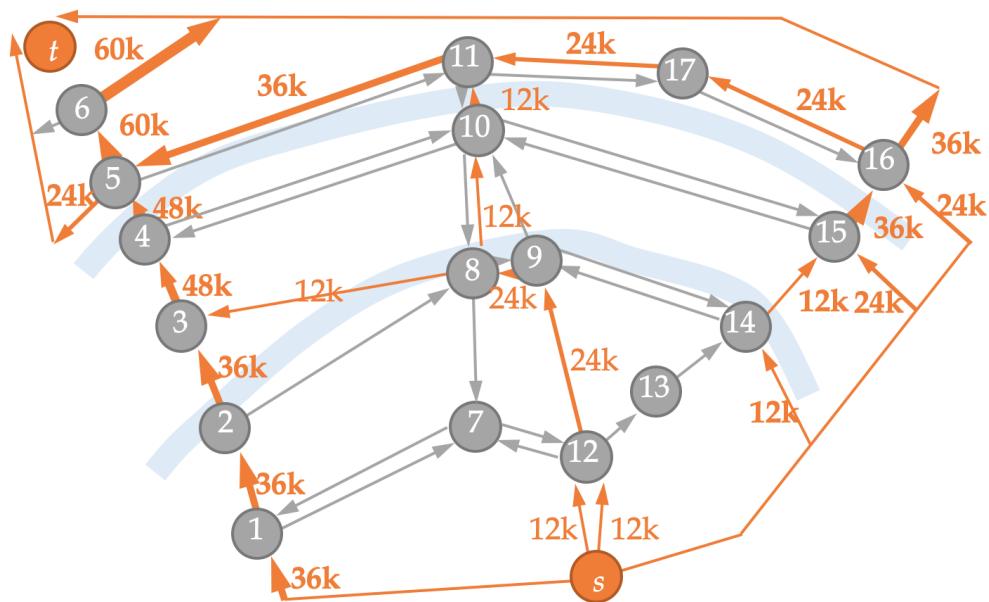


Рис. 8.8. Максимальный поток в сети

Максимальная величина потока $\varphi_{\max}^+ = \varphi_{\max}^- = 120\ 000$ автомобилей в час. Из них часть едет по левому широкому направлению, другая часть едет по набережной, и оставшаяся часть едет по центральному району.

Пропускная способность входов и выходов $c_{st} = 132\ 000$ автомобилей в час.

Возникает вопрос: если пропускная способность на входе 132 000, а на выходе только 120 000, то как сделать так, чтобы оставшиеся 12 000 автомобилей тоже смогли проехать? Ответ на этот вопрос дает задача о минимальном разрезе, которая будет рассмотрена в следующей главе.

9. Задача о минимальном разрезе

Для того, чтобы найти минимальный разрез, необходимо найти то самое «бутылочное горлышко» - наиболее узкую совокупность улиц, которые всегда максимально «забиты» машинами.

Множество вершин N сети $S = (N, U)$ можно разбить на два подмножества N_p и \bar{N}_p так, чтобы исток всегда был в одном из них, а сток – в другом. Например, $N_p = (s,)$ и $\bar{N}_p = (x_3, x_4, t)$ (рис. 9.1 *a*). Множество всех дуг U_p , выходящих из вершин множества N_p и входящих в вершины \bar{N}_p будет называться *разрезом*. Например, на рис. 9.1 *б* разрез проходит по дугам.

$$U_p = \{(x_1, x_3), (x_1, x_4), (x_2, x_3), (x_2, x_4)\}$$

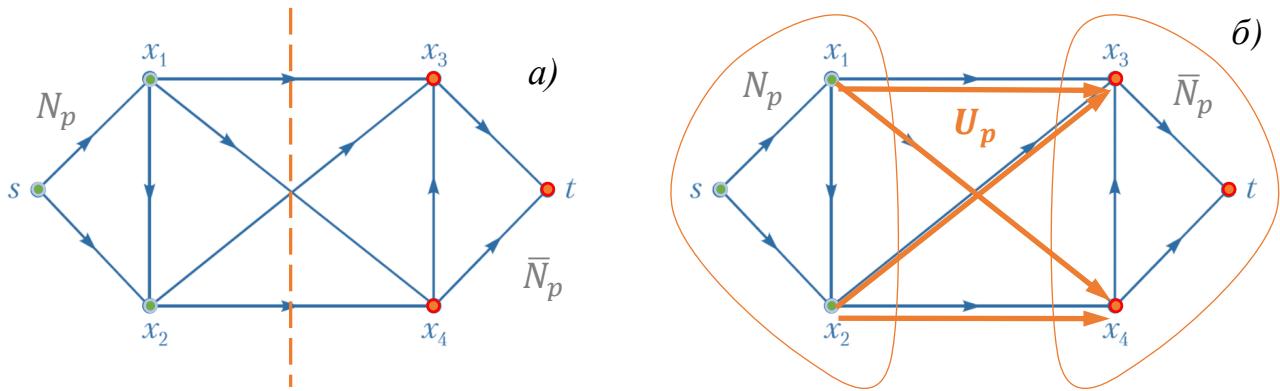


Рис. 9.1. Разрез сети (*а*) и разрез сети по дугам (*б*)

Разрез делает исходный граф двудольным графом. Разрез удаляет дуги U_p из графа и разделяет его на 2 половины (N_p, \bar{N}_p) (рис. 9.2).

Свойства разреза:

1. В одной половине разрезанного графа – всегда исток s , в другой – сток t .
2. При исключении дуг разреза поток по графу останавливается ($V = 0$).

У каждого разреза есть такое понятие, как величина разреза C . Величина разреза C – это сумма пропускных способностей всех дуг разреза (U_p).

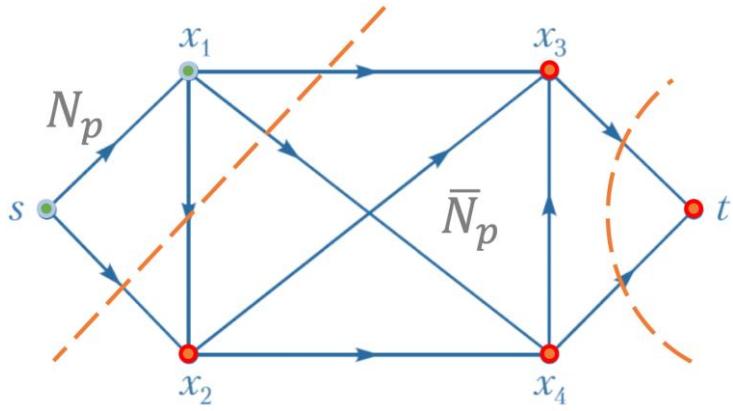


Рис. 9.2. Примеры разрезов

Граф можно разрезать различными способами так, что на выходе получим разный суммарный диаметр разреза дуг. В качестве примера можно привести трубы, суммарных диаметр которых будет различным.

Теорема Форда-Фалкерсона. Максимальный поток V_{\max} в сети равен минимальному разрезу $C(N_p, \bar{N}_p)$:

$$V_{\max} = \min_{U_p} C(N_p, \bar{N}_p)$$

В сети не может протекать поток больше, чем ширина самого узкого места, так называемого «бутылочного горлышка» (рис. 9.3):

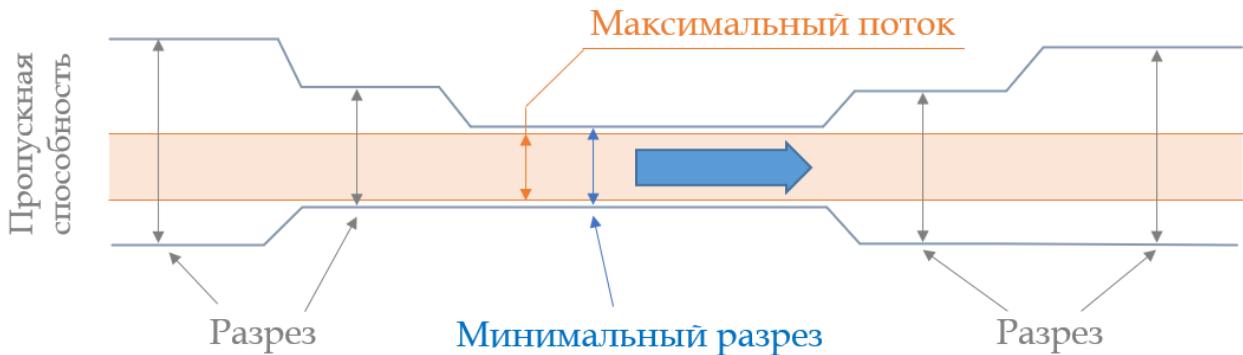


Рис. 9.3. Минимальный разрез

По рисунку видно, что величина максимального потока равна величине минимального разреза. Для того, чтобы повысить возможность проезда автомобилей необходимо найти, по каким дугам проходят **минимальные разрезы** (узкие места) и увеличить их **пропускную способность**.

Постановка задачи

Для решения этой задачи существуют множество различных *алгоритмов*, например:

1. Алгоритм Матильды Штор и Франка Вагнера – нахождение минимального разреза между любой парой вершин и «стягивание» в одну мультивершину. Повторяется до тех пор, пока не останется 2 вершины, между которыми и будет минимальный разрез графа.
2. Алгоритм Каргера-Штейна – рандомизированный перебор вариантов.
3. Алгоритмы Хао-Орлина, Намагочи-Ибараки, теория Менгера и др.

Однако мы с вами будем рассматривать применение линейного программирования для решения этой задачи.

Пусть дана сеть $G = (V, A)$, где V – множество узлов сети, A - множество дуг сети.

Кроме матрицы смежности, граф может быть задан различными способами. Например, для нашей задачи удобнее задать граф матрицей инцидентности – набором дуг (начало и конец дуги) и пропускными способностями (таблица 9.1):

$$C = \begin{bmatrix} 1 & 2 & 3 & \dots & n \\ 2 & 3 & 1 & \dots & n \\ c_{12} & c_{23} & c_{31} & \dots & c_{nn} \end{bmatrix}^T$$

где n – количество вершин графа.

Таблица 9.1

Матрица инцидентности $C = \{c_{ij}\}$

Начало дуги, i	Конец дуги, j	Пропускная способность, c_{ij}
s	1	36000
s	12	36000
s	14	12000
...
2	3	36000
2	8	36000
3	4	48000
...
16	t	36000

Переменные задачи (булевские) – дуги минимального разреза:

$$x_{ij} = \begin{cases} 1, & \text{дуга } ij \text{ входит в минимальный разрез} \\ 0, & \text{в противном случае} \end{cases}$$

Нововведением является то, что нам понадобятся дополнительные переменные – потенциалы узлов сети $\{\pi_i\}$, $i = \overline{1, n}$ (таблица 9.2):

Таблица 9.2

Потенциалы узлов сети

1	2	3	4	5	...	n
π_1	π_2	π_3	π_4	π_5	...	π_n

Потенциалом узла сети π называется максимальное количество потока φ , которое он способен через себя пропустить.

Потенциал узла определяется следующим образом: находится сумма пропускных способностей всех входящих в него дуг $\sum \varphi^+$ и всех выходящих $\sum \varphi^-$, затем из них выбирается *наименьшее* (рис. 9.4).

$$\pi = \min(\sum \varphi^+, \sum \varphi^-)$$

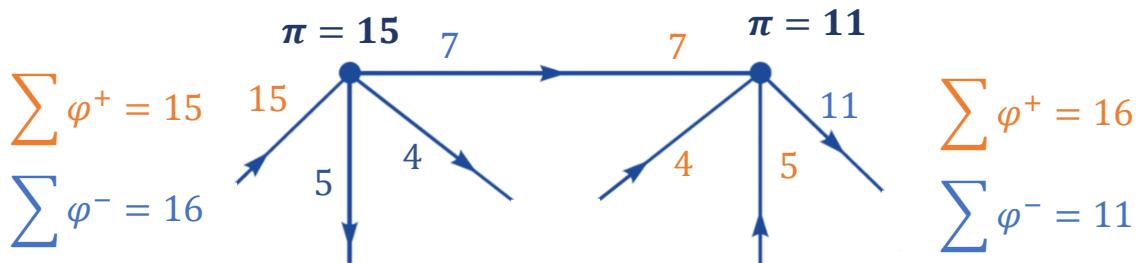


Рис. 9.4. Потенциал узлов сети

Целевая функция – минимальная пропускная способность разреза:

$$F = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min$$

Придаем небольшой дополнительный потенциал стоку сети t , равный 1. Т. е. на выход поступило больше воды или приехало больше автомобилей:

$$\pi_s = 0, \quad \pi_t = 1$$

Дополнительный поток по каждой дуге должен быть **положительным** (разность потенциалов на дуге сети) как показано на рис. 9.5:

$$\pi_i - \pi_j + x_{ij} \geq 0$$

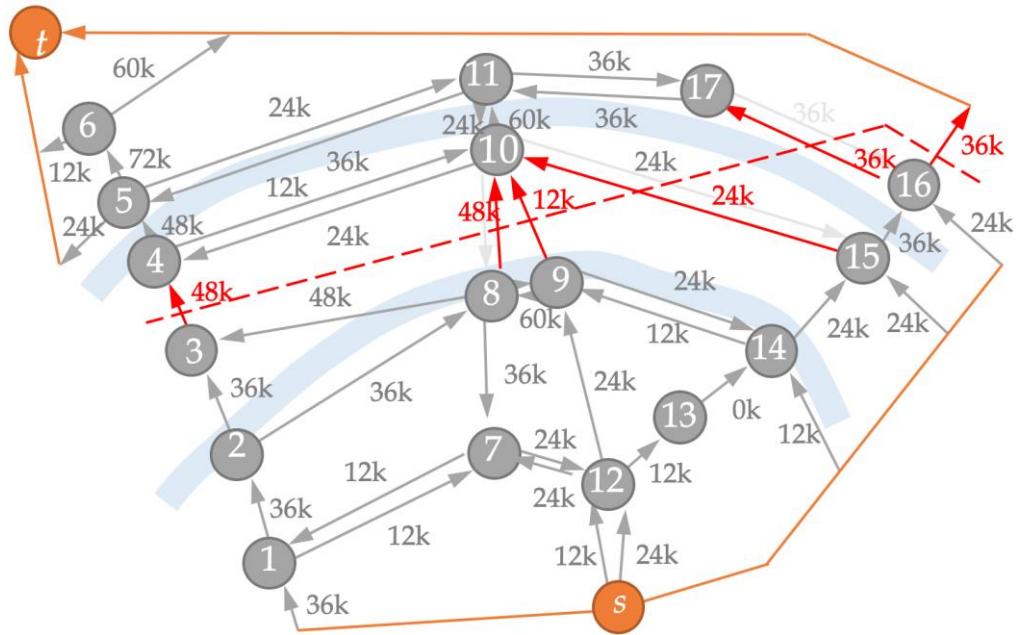


Рис. 9.5. Пример разреза дорожной сети

Мы видим, что данный набор формул представляет собой задачу линейного программирования, поэтому для решения данной задачи воспользуемся **симплекс-методом**. Результаты решения представлены в таблице 9.3.

Таблица 9.3

Результаты решения задачи симплекс-методом

Дуги минимального разреза №1:

Начало дуги, i	Конец дуги, j	Минимальный разрез, x_{ij}
...	...	0
4	5	1
...	...	0
...	...	0
...	...	0
11	5	1
...	...	0
16	t	1
...	...	0

Дуги минимального разреза №2:

Начало дуги, i	Конец дуги, j	Минимальный разрез, x_{ij}
...	...	0
1	2	1
...	...	0
12	9	0
...	...	0
13	14	1
...	...	0
s	16	1
...	...	0

В нашей задаче 2 минимальных разреза, проходящих по ребрам с пропускной способностью как представлено на рис. 9.6:

Любой другой разрез даст более широкую возможность проезда, поэтому для разрешения проблемы «бутылочного горлышка» необходимо работать сразу с двумя минимальными разрезами, так как при устранении проблемы с первым минимальным разрезом поток автомобилей встретится со вторым.

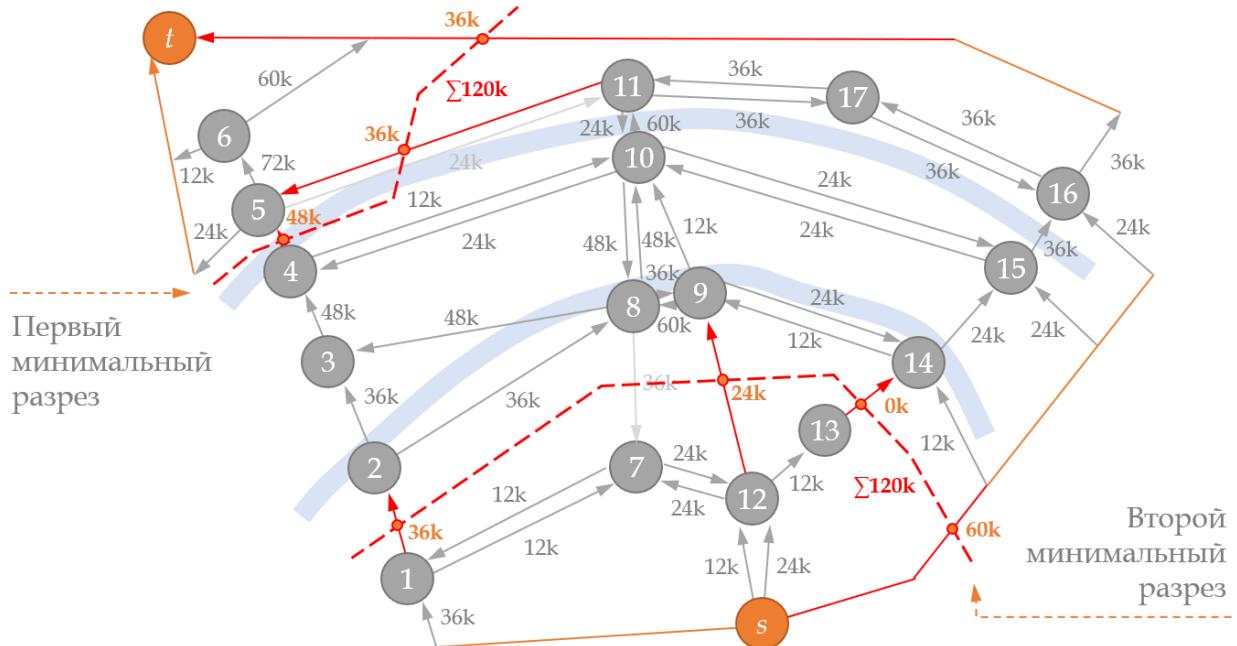


Рис. 9.6. Минимальные разрезы

Можно открыть проезд 13-14 для движения автомобилей одной полосой (0+12000), а также на мосту 4-5 забрать одну полосу от встречного движения (48000+12000). Таким образом мы повысим пропускную способность всей сети до 132000 автомобилей в час (рис. 9.7).

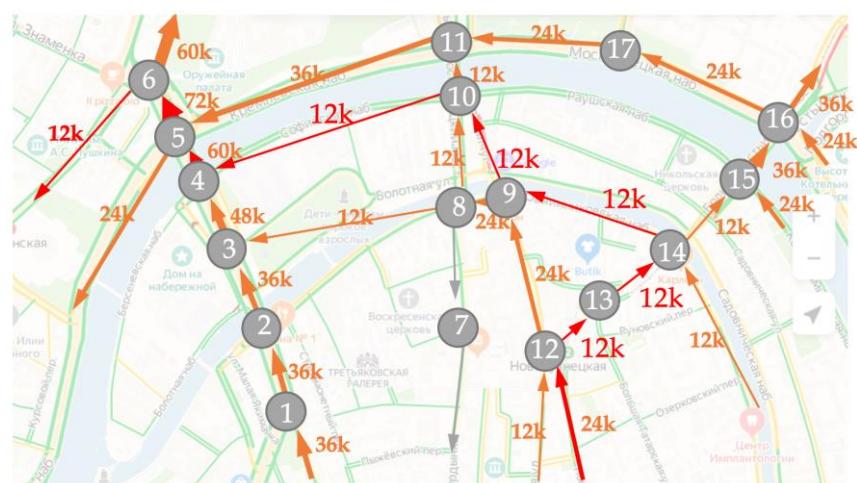
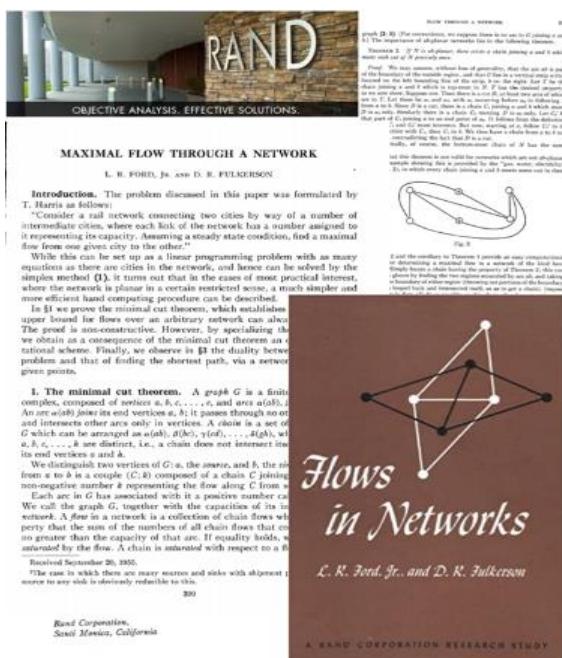


Рис. 9.7. Оптимизированный поток автомобилей по транспортной сети

Другие области применения задачи о максимальном потоке

1. Расчет маршрутов доставки грузов по железным дорогам.
2. Расчет пропускной способности и направлений прокладки веток трубопроводов (газ, нефть, водоснабжение и др.)
3. Анализ и оптимизация сетей передачи данных (Internet, LAN и др.).
4. Анализ мощности кабельной сети электроснабжения.
5. Анализ устойчивости любого типа систем к выходу из строя элементов (сохранение величины потока при удалении вершин или ребер).
6. Сохранение и распределение финансовых поток между организациями и отделениями банков и др.



Лестер Форд мл. и Делберт Рей Фалкерсон в 1954 г. впервые опубликовали технический доклад о проблеме максимального потока в графе и алгоритме его нахождения. Фалкерсон и Форд являлись сотрудниками корпорации RAND (Research and Development) - американской некоммерческой организации стратегических исследований правительства США. Основана в 1948 году BBC США и Douglas Aircraft в Санта-Монике. С начала 1960 занимается системным анализом, космическими расчетами и искусственным интеллектом. Создатели теоретической концепции и фактической облика современного Интернета. В её стенах работает более 30 нобелевских лауреатов. Значительная часть исследований корпорации RAND строго засекречена, однако не связанные с национальной безопасностью материалы выкладываются в публичный доступ.

10. Задача о паросочетаниях

Постановка задачи

Классическую задачу о паросочетаниях можно представить в виде задачи «о мальчиках и девочках». Традиционно предполагают, что мальчики и девочки собираются, например, на бал и необходимо сделать так, чтобы из заданного количества мальчиков и девочек было составлено максимальное количество пар. Одним из ограничений данной задачи является наличие и отсутствие симпатии между участниками бала, из-за этого составить пару можно только из тех ребят, между кем есть симпатия. Так же по правилам бала необходимо, чтобы один мальчик танцевал только с одной девочкой, а одна девочка только с одним мальчиком.

Исходя уже из данной формулировки, внимательный взгляд читателя мог заметить отчетливое присутствие классической задачи *линейного программирования*. В задаче имеются *исходные данные* (количество мальчиков и девочек и их предпочтения), *переменные* (образованные связи между мальчиками и девочками), *целевая функция* (количество образованных пар) и *ограничения* (зависимость паросочетания от симпатии молодежи и правил бала). Тогда задачу можно решить хорошо знакомым *симплекс-методом*, осталось лишь правильно ее формализовать.

Попробуем преобразовать исходные данные в понятный математический вид. Помочь в этом может построение *графа*. Исходя из ограничения о невозможности составлении пары между двумя мальчиками или двумя девочками, множество участников делится на два подмножества, между которыми не может быть смежности. Таким образом, граф будет являться *двуодольным*, его общий вид представлен на рис. 10.1.

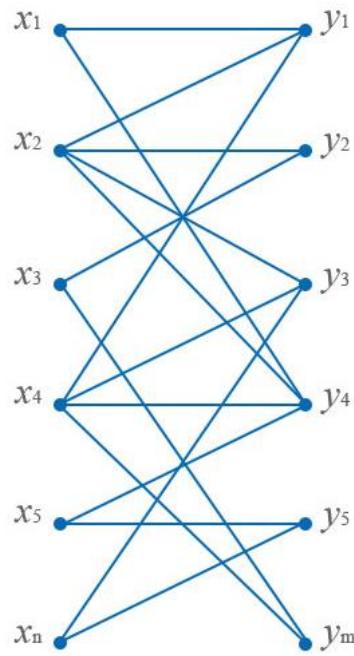


Рис. 10.1. Двудольный граф в общем виде

Для примера построим произвольный двудольный граф G (рис. 10.2), где вершины A_i – мальчики, вершины B_j – девочки, ребро (i, j) – наличие взаимной симпатии между i -м мальчиком и j -й девочкой.

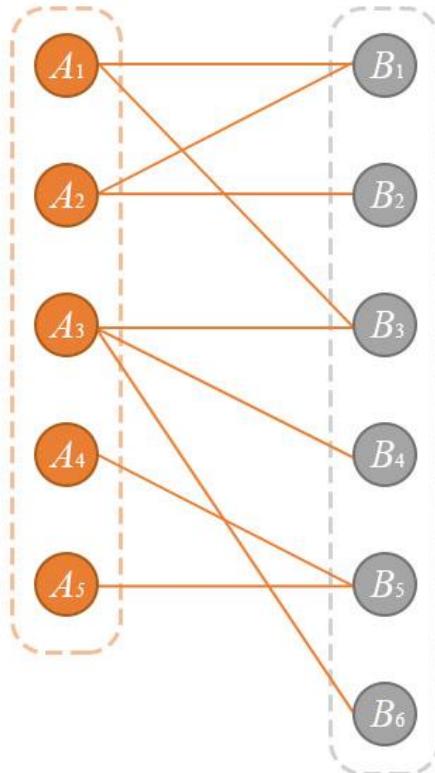


Рис. 10.2. Двудольный граф симпатии мальчиков и девочек

На основании построенного графа можно записать матрицу смежности:

$$C = \|c_{ij}\|_{n \times m} \quad (10.1)$$

где $c_{ij} \in \{0,1\}$ – наличие взаимной симпатии между i -м мальчиком и j -й девочкой,

n – количество мальчиков, m – количество девочек.

Отличаться данная матрица от привычной будет тем, что столбцы и строки не будут попарно соответствовать. Таким образом, в строках будут представлены мальчики, а в столбцах – девочки. Полученная матрица представлена в таблице 10.1.

Таблица 10.1

Матрица смежности графа симпатии

	B_1	B_2	B_3	B_4	B_5	B_6
A_1	1	0	1	0	0	0
A_2	1	1	0	0	0	0
A_3	0	0	1	1	0	1
A_4	0	0	0	0	1	0
A_5	0	0	0	0	1	0

Исходя из полученных данных можно объявить переменные в виде **аналогичной бинарной матрицы** составленных пар X , где единица на пересечении i -й строчки и j -го столбца будет обозначать наличие пары между i -м мальчиком и j -й девочкой, а ноль – ее отсутствие.

Тогда **целевой функцией** будет являться **максимальное** число единиц в матрице переменных, представляющее собой суммарное количество пар. Представить ее в математическом виде можно следующим образом:

$$F = \sum_{i=1}^n \sum_{j=1}^m x_{ij} \rightarrow \max$$

Для полной записи задачи в виде ЗЛП остается ввести ограничения, определенные предпочтениями участников в выборе партнера и правилами бала. Их можно задать так:

$$\left\{ \sum_{i=1}^n x_{ij} \leq 1; \quad j = \overline{1, m} \right\} \quad (10.2)$$

$$\left\{ \sum_{j=1}^m x_{ij} \leq 1; \quad i = \overline{1, n} \right\} \quad (10.3)$$

$$x_{ij} \leq c_{ij} \quad (10.4)$$

Ограничение (10.2) обозначает, что сумма элементов матрицы по строкам **не должна превышать** единицы, то есть каждая девочка должна танцевать не более чем с одним мальчиком.

Ограничение (10.3) обозначает, что сумма элементов матрицы по столбцам **не должна превышать** единицы, то есть каждый мальчик должен танцевать не более чем с одной девочкой.

Ограничение (10.4) обозначает, что единица в матрице переменных может располагаться только в том случае, если соответствующий элемент матрицы смежности **равен единице**. Другими словами, можно составить пару только при наличии ребра взаимной симпатии в исходном графе.

Можно заметить, что математическая постановка задачи идентична задаче о назначениях. И это не удивительно, ведь общий смысл назначения людей в пару друг другу аналогичен назначению людей «в пару» работе, для которой он подходит лучшим образом. Этот пример еще раз иллюстрирует тесную связь теории графов и математического программирования.

После решения поставленной задачи **симплекс-методом** имеем решение, представленное в таблице 10.2.

Таблица 10.2

Матрица переменных решения

	B_1	B_2	B_3	B_4	B_5	B_6
A_1	1	0	0	0	0	0
A_2	0	1	0	0	0	0
A_3	0	0	1	0	0	0
A_4	0	0	0	0	1	0
A_5	0	0	0	0	0	0

Также можем изобразить это решение на графике (рис. 10.3).

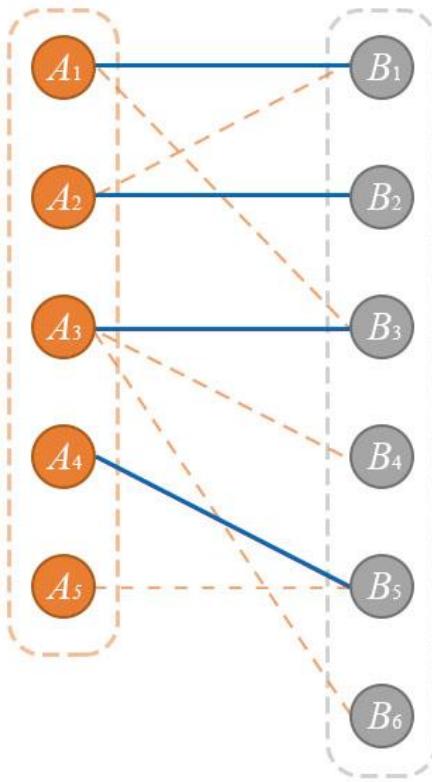


Рис. 10.3. Двудольный граф составленных пар

Как можем заметить, не каждому досталась пара. Из-за преобладания количества женского пола над мужской как минимум одна девочка не смогла пойти на бал за неимением свободных мальчиков, а из-за отсутствия взаимной симпатии как минимум одна девочка и один мальчик не смогли перебороть себя и отправиться танцевать с несимпатичным ему партнером.

Также следует отметить то, что данное решение, как это бывает, не единственное. Поменяй мы партнершу мальчика A₃ с B₃ на B₆, с математической точки зрения ничего бы не изменилось касательно целевой функции. Но это лишь потому, что и B₃ и B₆ одинаково симпатичны мальчику A₃. Но часто ли можно встретить такое совпадение? Нечасто, а значит, имеет смысл указывать не просто наличие взаимной симпатии, а действительное значение силы симпатии. Другими словами, можно задать вес связи между вершинами. Но это уже не классическая задача о паросочетаниях, а ее вариация. Рассмотрим ее и некоторые другие в следующем разделе.

Вариации задач о паросочетаниях

Выше была представлена базовая математическая модель задачи о паросочетаниях. Однако в жизни часто приходится сталкиваться с необходимостью доработать ее, чтобы учесть некоторые дополнительные условия задачи. Рассмотрим несколько таких возможных доработок.

1) Начнем с уже упомянутой задачи, когда взаимоотношения между партнерами определяются **не бинарным** значением (1 – есть симпатия, 0 – нет симпатии), а задается числом, обозначающим силу взаимной симпатии. Тогда сочтания будут строиться не на основе возможности ее построения, а на мощности полученной пары.

Условия задачи от данной постановки меняются некардинально. Изменений будет лишь два: первое заключается в том, что матрица симпатий C , описанная в выражении 10.1, будет содержать не бинарные значения, а **вещественные**, то есть $c \in N$; второе заключается в изменении целевой функции, теперь это не просто максимальное количество пар, а **суммарная сила** симпатии среди всех пар, другими словами, имеем:

$$F = \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} \rightarrow \max$$

2) Еще одной вариацией является задача, где из-за каких-либо обстоятельств позволяет мальчикам танцевать с несколькими девочками или наоборот, одной девочке с несколькими мальчиками. В таком случае преобразования затронут наши условия, касающиеся возможности танцевать только с одним партнером. В случае, если мальчикам позволено танцевать с несколькими партнершами, из условия будет вычеркнуто ограничение 10.3, аналогично с разрешением для девочек – вычеркиваем ограничение 10.2.

3) Третьей очень важной вариацией является задача, когда симпатии молодых людей могут не совпадать. В этом случае наш двудольный граф приобретает еще одну характеристику – теперь он **ориентированный** (рис. 10.4).

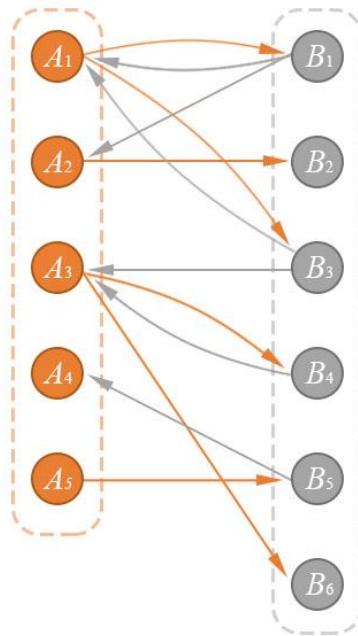


Рис. 10.4. Ориентированный двудольный граф симпатий

В этом случае изменения весьма значимы. Можно придумать несколько способов решения. Рассмотрим один из них. Разделим матрицу смежности на две матрицы: одна для обозначения симпатии мальчиков, вторая для обозначения симпатии девочек. Тогда матрица переменных X будет определять взгляды первых, а матрица Y – вторых:

$$Y = \|y_{ij}\|_{m \times n}$$

где $y_{ij} \in \{0,1\}$ – наличие симпатии у i -й девочки к j -му мальчику, n – количество мальчиков, m – количество девочек.

Соответственно, целевой функцией будет вновь суммарное количество возможных пар, которое надо **максимизировать**. Имеем:

$$F = \sum_{i=1}^n \sum_{j=1}^m x_{ij} y_{ji} \rightarrow \max$$

Стойте отдельно акцентировать внимание на порядке индексации элементов матриц в целевой функции. Мы задали матрицы таким образом, что в матрице симпатий мальчиков X в строках (под индексом i) расположены мальчики, а в столбцах (под индексом j) расположены девочки, но в матрице симпатий

девочек Y все наоборот: в строчках – девочки, в столбцах – мальчики. Тогда для того, чтобы правильно соотнести элементы матриц друг с другом, необходимо либо транспонировать матрицу Y , либо поменять порядок индексации. Поэтому в целевой функции можно увидеть, что из матрицы X берется элемент из i -й строки и j -го столбца, а из матрицы Y наоборот: берется элемент j -й строки и i -го столбца.

Построение сетевой диаграммы на основе задачи о паросочетаниях

В математике, как и в жизни, все взаимосвязано. Одни понятия переходят в другие, методы решения одних проблем справляются с другими методами. Перед нами очередной случай возможности представить одну задачу в виде другой и применить другие, свойственные ей, инструменты.

Как мы ранее показали связь задачи о паросочетаниях с задачей о назначении, находящейся вне теории графов, также и здесь можно показать связь между задачами внутри теории графов. Добавляя сток и исток к рассмотренному ранее графу, показанном на рис. 10.3, мы получаем из двудольного графа – **двуухполюсную (биполярную) сеть**, на которой можем решать задачу о максимальном потоке и искать **минимальные разрезы** (рис. 10.5).

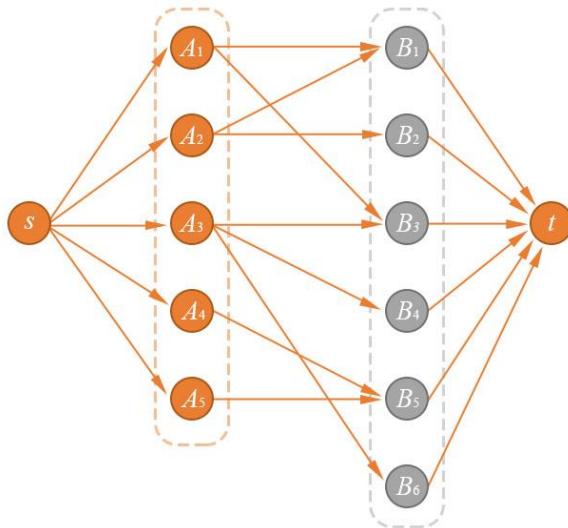


Рис. 10.5. Двуухполюсная сетевая диаграмма

Величина потока в данном примере будет обозначать количество возможных пар, а решение задачи минимального разреза поможет определить, между

кем из мальчиков и девочек следует добавить симпатию, чтобы увеличить значение потока на единицу.

Таким образом, можно комбинировать инструменты решения задач из разных областей математики, преобразуя их математически из одной в другую, и получать новые полезные результаты.

11. Задача о выравнивании ресурсов проекта

Постановка задачи выравнивания ресурсов

Проблемы проектного управления затрагивают факторы инициации, планирования, исполнения и контроля проектной деятельности. Эти проблемы начали тревожить трудовую сферу человечества еще с давних времен, собственно, с тех пор как люди научились организованно подходить к совместной работе. Стремление тратить меньше, но при этом делать больше, быстрее и качественнее развивало мысли математических умов о все более новых методах оптимизации каждого из этапов проектной деятельности. В рамках настоящей темы будет затронута проблема планирования, а в частности, задача выравнивания ресурсов проекта.

Представим, что менеджер проекта составил набор задач для достижения какой-то поставленной цели (таблица 11.1).

Таблица 11.1

Перечень задач проекта

	Наименование задачи
1	Составление сметы
2	Приобретение оборудования
3	Подбор кадров
4	Монтаж оборудования
5	Подготовка кадров
6	Оформление зала
7	Доставка товаров
8	Заказ и получение униформы
9	Заказ и получение ценников
10	Выкладка товаров
11	Заполнение ценников
12	Генеральная репетиция

Чтобы составить план работы необходимо обозначить последовательность выполнения обозначенных задач. Для этого следует построить *иерархическую зависимость* между работами, а именно, назначить для каждой работы предшественника. Результат можно увидеть на таблице 11.2.

Таблица 11.2

Иерархический план задач проекта

	Наименование задачи	Предшественник
1	Составление сметы	-
2	Приобретение оборудования	1
3	Подбор кадров	1
4	Монтаж оборудования	2
5	Подготовка кадров	3
6	Оформление зала	4
7	Доставка товаров	5
8	Заказ и получение униформы	5
9	Заказ и получение ценников	5
10	Выкладка товаров	6, 7
11	Заполнение ценников	9
12	Генеральная репетиция	8, 10, 11

На основании построенных зависимостей можно сказать, например, что возможность приобрести оборудование (задача 2) и подобрать кадры (задача 3) не имеет места быть до тех пор, пока не будет составлена смета (задача 1). Исходя из этого, можно отметить, что какие-то задачи являются последовательными, а какие-то параллельными, при этом существует первая задача (исток) и заключительная (сток). Попробуем представить полученный план в виде *сетевой диаграммы* (рис. 11.1).

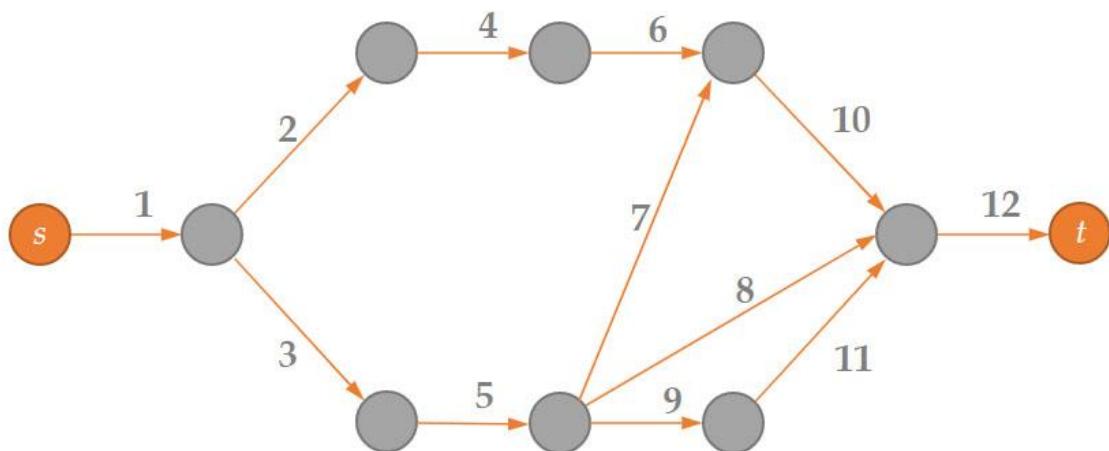


Рис 11.1. Сетевая диаграмма плана задач проекта

На данной диаграмме ребра – назначенные работы, номера соответствуют номерам задач, вершины – результаты, обозначающие выполнение одной задачи и начало выполнения другой (кроме вершин s и t , обозначающий начало и конец проекта соответственно).

Альтернативный способ построения графа – пронумеровать вершины и представить работы (ребра) парой смежных вершин (рис. 2). Пример: смежные вершины i и j связаны ребром (i, j) .

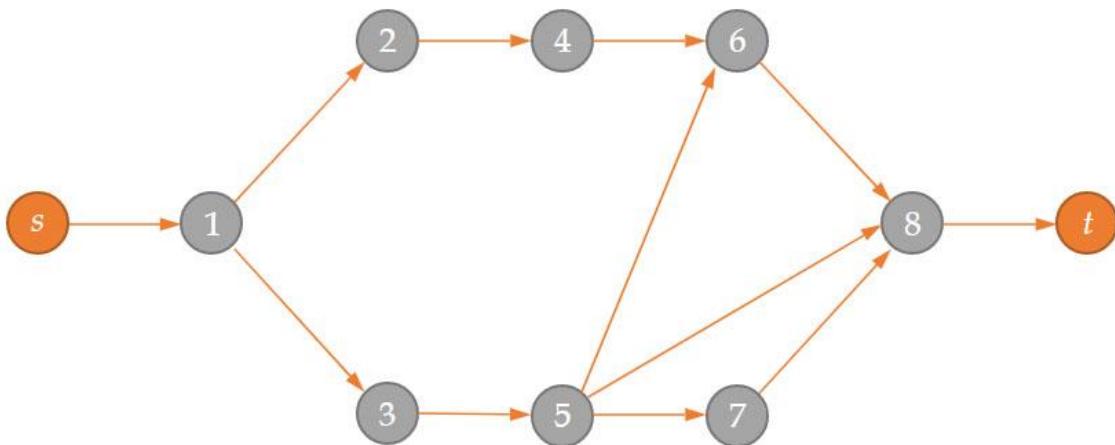


Рис 11.2. Сетевая диаграмма плана задач проекта с пронумерованными вершинами

Тогда соответствующий данной диаграмме план указан в таблице 11.3.

Таблица 11.3

Иерархический план задач проекта

	Наименование задачи
$(s, 1)$	Составление сметы
$(1, 2)$	Приобретение оборудования
$(1, 3)$	Подбор кадров
$(2, 4)$	Монтаж оборудования
$(3, 5)$	Подготовка кадров
$(4, 6)$	Оформление зала
$(5, 6)$	Доставка товаров
$(5, 8)$	Заказ и получение униформы
$(5, 7)$	Заказ и получение ценников
$(6, 8)$	Выкладка товаров
$(7, 8)$	Заполнение ценников
$(8, t)$	Генеральная репетиция

Для решения поставленных задач необходимы ресурсы. Они могут быть человеческими (кадры, сотрудники, работники и т. д.), финансовыми (денежные средства, капитал, инвестиции, заработка плата и прочее), материальные (оборудование, инструменты и т. д.) и другими. Предположим, менеджеру необходимо распределить сотрудников для выполнения некоторой работы.

Мы знаем, что если назначить на выполнение работы одного человека в первом случае и пять человек во втором, то, очевидно, во втором случае работа будет выполнена быстрее. Тогда в интересах менеджера проекта распределить работников таким образом, чтобы достигнуть цели проекта как можно быстрее. При этом, конечно же, стоит отметить, что численность сотрудников ограничена.

Предположим, менеджер распределил работников эмпирическим путем (исходя из своего богатого опыта), и результат распределения можно увидеть в таблице 11.4.

Таблица 11.4

План с эмпирически распределенными ресурсами

	Наименование задачи	Ресурсы, трудодни	Длительность, дни
(s, 1)	Составление сметы	10	15
(1, 2)	Приобретение оборудования	5	16
(1, 3)	Подбор кадров	2	6
(2, 4)	Монтаж оборудования	3,3	6
(3, 5)	Подготовка кадров	1,7	5
(4, 6)	Оформление зала	2,5	8
(5, 6)	Доставка товаров	1,4	6
(5, 8)	Заказ и получение униформы	1,1	14
(5, 7)	Заказ и получение ценников	1	8
(6, 8)	Выкладка товаров	1,3	2
(7, 8)	Заполнение ценников	0,9	4
(8, t)	Генеральная репетиция	0,8	3

Не стоит пугаться наличию «полутора землекопов», ведь дробное значение человеческих ресурсов обозначает дробное значение ставки (например, значение 3,3 будет означать, что кто-то из работников потратит лишь треть своего

рабочего дня на конкретную задачу). Из рассчитанной где-то за кулисами нормы трудодней по каждой задаче была выведена длительность выполнения работ. Обозначенные длительности можно также отразить на графике, для этого каждому ребру присвоим значение веса, равное длительности соответствующей работы (рис. 11.3).

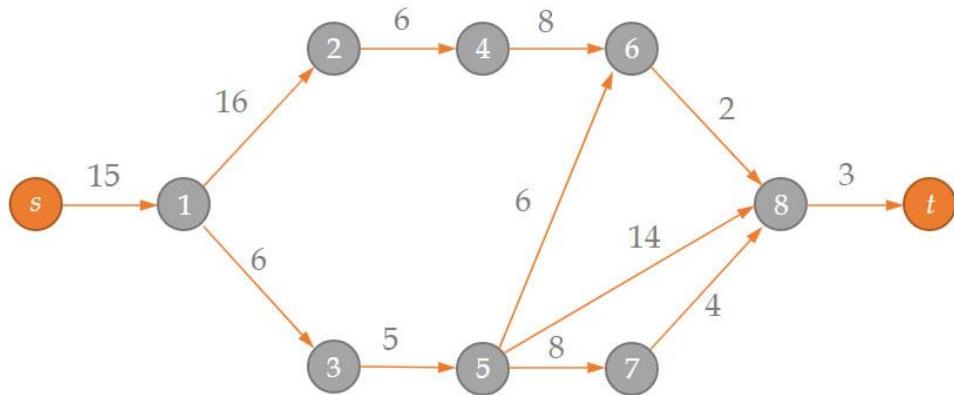


Рис. 11.3. Сетевая диаграмма плана задач с обозначенными длительностями

Следующим очевидным и интуитивно понятным действием хочется привнести в график наглядности и отрисовать длины ребер согласно длительности работы. Для этого следует построить ось времени и откладывать вершины графа с учетом поставленных временных условий (рис. 11.4).

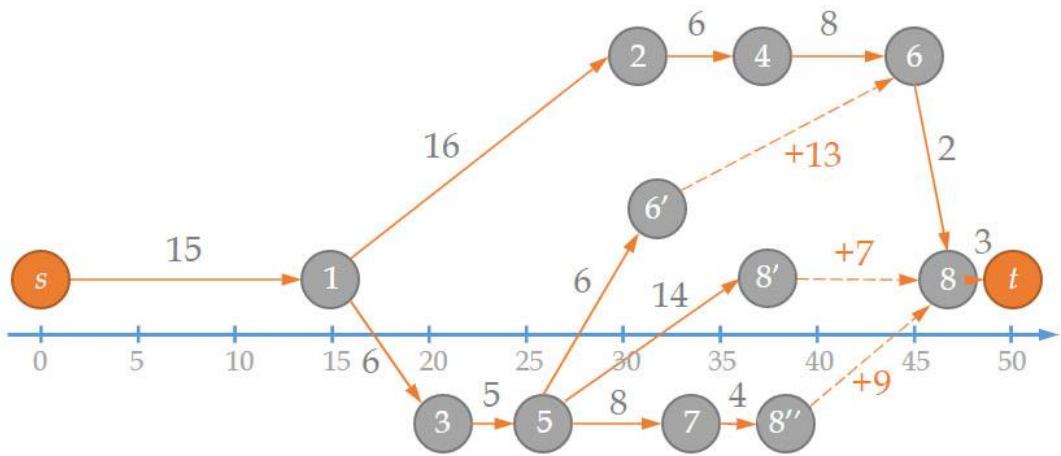


Рис. 11.4. Сетевая диаграмма с пропорциональными длинами ребер

Как можно увидеть из диаграммы, не каждая вершина располагается так, как хотелось бы. Из-за несогласованности расстояний появляются промежуточные вершины со штрихами, соединенные со своими реальными узлами

пунктирными линиями, длина которых несет в себе совершенно однозначный физический смысл – это длительность простоя. Другими словами, ресурсы, направленные на выполнение, например работы (5, 6) вынуждены 13 дней ожидать выполнения работ (2, 4) и (4, 6). Нетрудно понять, что любой простой – это потеря эффективности работы и как следствие увеличение сроков работы. По этой причине перед менеджером проекта стоит задача минимизировать подобные простои. Таким образом появляется уже упомянутая задача о выравнивании ресурсов, направленная на оптимальное распределение ресурсов с целью сокращения общей длительности проекта.

Решение задачи выравнивания ресурсов

Перед тем, как приступить к решению поставленной задачи, стоит рассмотреть такую характеристику графа, как критический путь. Данное понятие играет большую роль в сфере проектного управления. Критический путь – это путь от истока к стоку, который имеет максимальную длительность. По смыслу же значит, что критический путь в данном примере – это последовательность работ без запасов времени. То есть при затягивании какой-либо из работ в критическом пути на какой-то срок общая длительность проекта так же увеличится на этот срок (рис. 11.5).

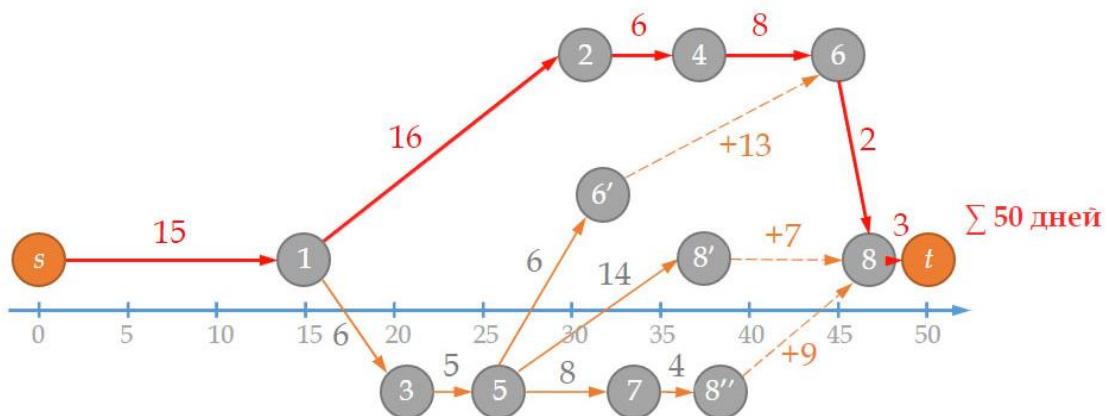


Рис. 11.5. Критический путь на диаграмме плана проекта

Критический путь всегда содержит только последовательные работы и его длительность равна длительности проекта. Из этого следует, что сокращение

критического пути сократит сроки выполнения проекта. Формализуем задачу в математическом виде.

В качестве переменных обозначим вектор количества перераспределенных ресурсов:

$$\bar{b} = \{b_{i,j}^{opt}\}$$

где $b_{i,j}^{opt}$ – оптимальное количество ресурсов, выделенное на задачу (i,j) .

Тогда целевая функция будет выглядеть следующим образом:

$$\tau = \sum_{i,j} t_{i,j}^{opt} \rightarrow \min; i,j \in (s, 1, 2, 4, 6, 8, t)$$

где $t_{i,j}^{opt}$ – оптимальное время выполнения задачи (i,j) .

В дополнение обратимся к нашему богатому опыту. «Если двое человек выкапывают яму в течение четырех часов, то четверо человек справится с задачей за два часа», – можно ли назвать данное утверждение абсолютно верным? С одинаковой долей вероятностью можно сказать и «да» и «нет», однако опыт подсказывает, что чем больше человек работает над одной задачей, тем больше они друг другу мешают. Другими словами: «девять женщин за месяц ребенка не рожают». Отсюда появляется такое явление как снижение эффективности каждого последующего помощника. Переводя на язык математики, зависимости изменения длительности работ от количества выделенных на нее дополнительных ресурсов как правило не является линейной, но имеет экспоненциальный характер (рис. 11.6). Отсюда имеем:

$$t_{i,j}^{opt} = t_{i,j} \cdot e^{\frac{b_{i,j} - b_{i,j}^{opt}}{b_{i,j}}}$$

где $b_{i,j}$ – неоптимальное время выполнения задачи (i,j) , $t_{i,j}$ – неоптимальное количество ресурсов, выделенное на решение задачи (i,j) .

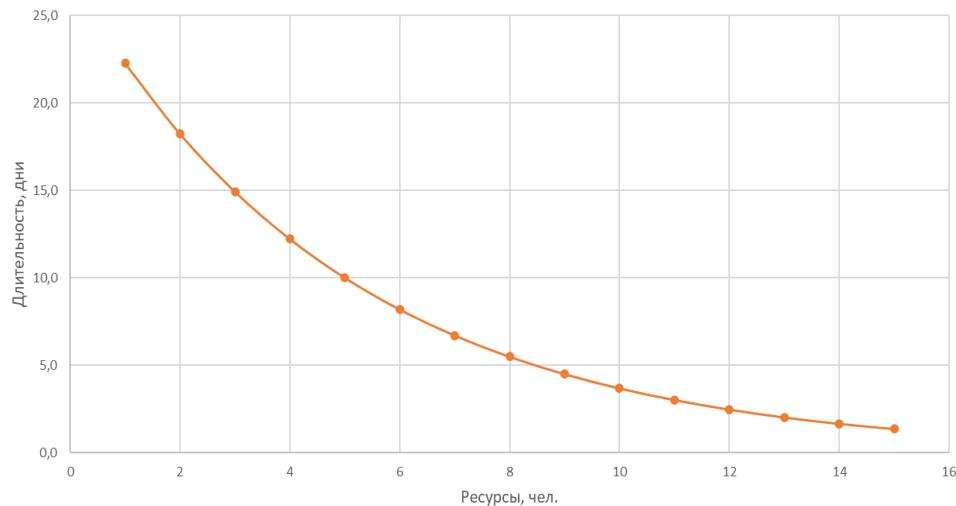


Рис. 11.6. График зависимости длительности работ от количества ресурсов

Из-за нелинейности функции, которой задается оптимальное время, задача выравнивания ресурсов является задачей нелинейного программирования, поэтому решается методом обобщенного приведенного градиента (ОПГ). Помимо определенных переменных и целевой функции имеются следующие ограничения:

$$\sum_{P_1} t_{i,j}^{opt} = \sum_{P_2} t_{i,j}^{opt} = \sum_{P_3} t_{i,j}^{opt} = \sum_{P_4} t_{i,j}^{opt} \quad (11.1)$$

$$P_1 \in (s, 1, 2, 4, 6, 8, t), \quad P_2 \in (s, 1, 3, 5, 6', 8, t)$$

$$P_3 \in (s, 1, 3, 5, 8', t), \quad P_4 \in (s, 1, 3, 5, 7, 8'', t)$$

$$\sum_{i,j} b_{i,j}^{opt} = \sum_{i,j} b_{i,j} \quad (11.2)$$

Ограничение (11.1) обозначает, что длина всех путей, ведущих от начала проекта к его завершению, должна быть одинакова, другими словами, запасы времени (простои) должны отсутствовать на каждой ветке. Ограничение (11.2) обозначает, что количество ресурсов до и после выравнивания должно оставаться постоянным. Применив метод ОПГ, мы получили результаты, которые можно увидеть в таблице 11.5, а на перерисованной диаграмме (рис. 11.7) можно увидеть отсутствие всякого присутствия каких-либо простоев. В итоге, видим, что по цепочке работ $(s, 1, 2, 4, 6, 8, t)$, что по цепочке $(s, 1, 3, 5, 7, 8, t)$, что по любым

другим проект будет выполняться оптимальные 43,6 дней, вместо 50 эмпирически запланированных дней.

Таблица 11.5

Результат решения задачи выравнивания ресурсов

	Наименование задачи	Ресурсы, трудодни	Длительность, дни	Выравн. ресурсы, трудодни	Оптималь. длительн., дни
(s, 1)	Составление сметы	10	15	10,6	14,1
(1, 2)	Приобретение оборудования	5	16	6,4	12,0
(1, 3)	Подбор кадров	2	6	1,8	6,8
(2, 4)	Монтаж оборудования	3,3	6	2,4	8,0
(3, 5)	Подготовка кадров	1,7	5	1,5	5,7
(4, 6)	Оформление зала	2,5	8	3,2	6,0
(5, 6)	Доставка товаров	1,4	6	0,3	13,6
(5, 8)	Заказ и получение униформы	1,1	14	1,0	16,0
(5, 7)	Заказ и получение ценников	1	8	1,0	8,4
(6, 8)	Выкладка товаров	1,3	2	1,0	2,4
(7, 8)	Заполнение ценников	0,9	4	0,3	7,6
(8, t)	Генеральная репетиция	0,8	3	1,6	1,2

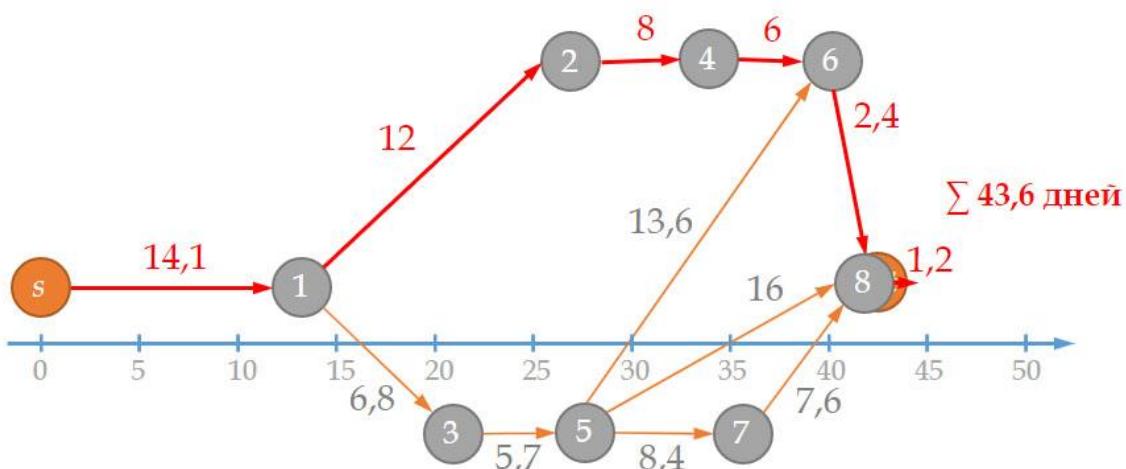
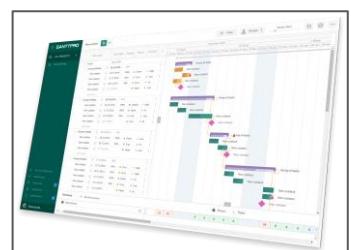


Рис. 11.7. Итоговая диаграмма оптимального плана проекта



Генри Лоуренс Ганнт (20 мая 1861 года - 23 ноября 1919 года) — один из наиболее успешных консультантов по управлению, автор диаграммы Ганнта и более 10 изобретений в современном менеджменте. Ганнт работал вместе с «отцом научного менеджмента» Фредериком Тейлором над количественными методами организации производства. Диаграммы Ганнта (англ. Gant Charts) и сегодня считается одним из стандартов де-факто. Диаграммы Ганнта были революционными в 20-х годах XX века и применялись в таких грандиозных инженерных проектах, как строительство дамбы Гувера (1931г.) и постройка сети скоростных шоссе США (1956г.), многие из которых закончились досрочно.



ЧАСТЬ III. ТЕОРИЯ ИГР

Теория игр выросла из простейших игр древности. Шашки, шахматы, карты и другие азартные игры положили начало большому и очень объемному разделу современной математики.

Сейчас принято считать, что теория игр – это наука на стыке математики и социологии; наука об оцифровке поведенческих взаимоотношений при конфликте двух или более сторон. В данном случае конфликт рассматривается с положительной стороны, с научной точки зрения, когда несколько сторон стремятся к одной и той же цели (например, стараются привлечь потенциальных покупателей или занять часть рынка).

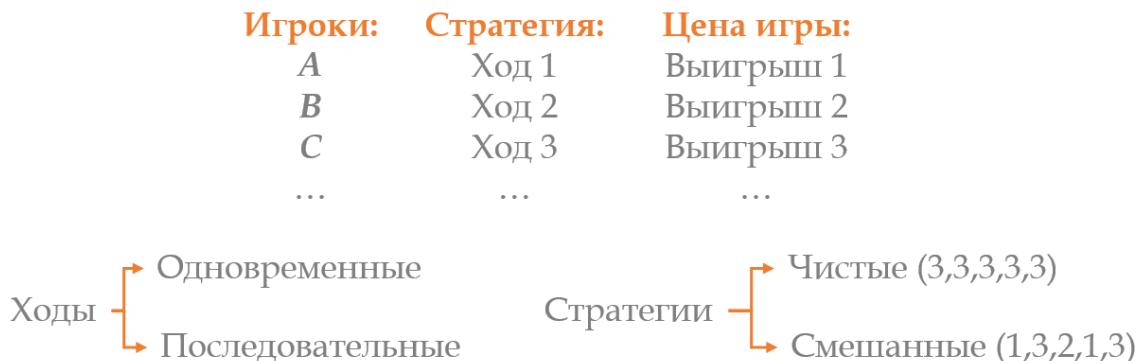
Теория игр – раздел математической экономики, изучающей решение конфликтов между двумя и более игроками и оптимальность их стратегий. Теория игр переносит игровые упрощенные модели на решение реальных, зачастую более сложных, задач.

Разновидностей игр очень много: под различные жизненные и бизнес-ситуации разработано некоторое количество математических моделей, каждая из которых подходит к описанию каждой ситуации более детально и профессионально.

Определим для себя компоненты игры (рис. III):

1. Непосредственные игроки (двоє і більше);
2. Ходы игроков, которые осуществляются с целью получения какого-то определенного выигрыша – могут быть одновременными (например, в игре «Камень – Ножницы – Бумага») или последовательными (например, шахматы, где каждый последующий ход делается с учетом крайнего хода противника);
3. Стратегии игроков – представляют из себя совокупность ходов; могут быть как чистыми, так и смешанными;
4. Выигрыши игроков («стоимость», назначающаяся на каждый ход, как, например, в преферанс, или в конце игры - шахматы);
5. Цена игры – совокупность выигрышей (может высчитываться, как сумма выигрышей каждого из ходов или назначаться в конце игры);

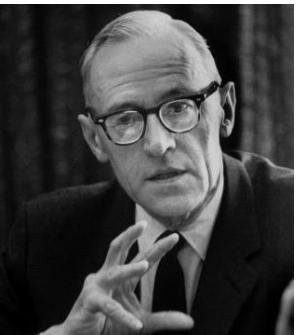
6. Платежная матрица – элемент, отражающий зависимость выигрыш от ходов (показывает количество выигрыша в ячейке на пересечении двух стратегий); ячейки платежной матрицы – все возможные варианты ходов и выигрышей.



Платежные матрицы игры «камень, ножницы, бумага»

A	К	Н	Б	B	К	Н	Б
К	0	1	-1	К	0	-1	1
Н	-1	0	1	Н	1	0	-1
Б	1	-1	0	Б	-1	1	0

Рис. III. Элементы любой игры



Теория игр получила свое начало на полях сражений Второй мировой войны. Применить полученные модели боевых столкновений к экономике впервые решились **Джон фон Нейман** и **Оскар Моргенштерн** в своей книге «**Теория игр и экономическое поведение**», Принстонский университет, США, 1944г. Вплоть до 1950-х теория игр оставалась формальной теорией в рамках математики. Но уже с 1950-х гг. начинаются попытки применить методы теории игр не только в экономике, но и в биологии, кибернетике, технике, антропологии. Во время Второй мировой войны и сразу после неё теорией игр серьёзно заинтересовались военные, которые увидели в ней мощный аппарат для исследования стратегических решений. Начиная с 1994 года огромное количество нобелевских премий было выдано в области теории игр. В период с 1994 по 2020 годы премию получили 18 человек. В 2020 году обладателями нобелевской премии стали Пол Милром и Роберт Уилсон за теорию аукционов (одно из ответвлений теории игр, определяющей правила оптимальных торгов для достижения максимальной выгоды).

12. Антагонистические игры

Перейдем к рассмотрению одного из самых простых видов игр в теории игр, а именно к антагонистическим играм, или так называемым играм с нулевой суммой.

Название антагонистических игр происходит от древнегреческого слова антагонист (ἀνταγωνιστής), обозначающего соперников, которые борются друг с другом. Данный тип игр применяется в случае, когда два или более игроков, реализуя свои стратегии на одном поле, пытаются завоевать большую (а желательно всю) его часть. В таких играх выигрыши игроков противоположны (выигрыш одного игрока является проигрышем другого).

Предположим, что два игрока делят между собой банк в 100 монет. Если один игрок выиграл 63 монеты, то второй эти 63 монеты проиграл, а выиграл только 37. Относительно равновесного состояния в 50 монет (ровно половина банка), первый игрок выиграл 13 монет, а второй – проиграл их. Таким образом, выигрыш первого игрока равняется +13, а выигрыш второго – -13. Если мы сложим выигрыши первого и второго игроков – получим ноль ($+13+(-13)=0$).

Именно поэтому игры данного типа также называют играми с нулевой суммой: сколько выиграл один игрок, столько же проиграл второй.

В данной игре используется только одна платежная матрица (в силу того, что платежная матрица второго игрока будет обратной к платежной матрице первого). В неё будут вноситься выигрыши первого игрока, а выигрыши второго будут вычисляться как разность между банком и выигрышем первого игрока.

Ходы в антагонистической игре одновременные. Стратегии могут быть как чистые, так и смешанные.

Чистые стратегии

Рассмотрим реализацию антагонистической игры на примере.

Пусть имеется офисный центр, в который каждое утро приходит на работу 1000 человек. К офисному центру ведут три дороги. Есть две компании,

занимающиеся продажей кофе – крупная известная компания А и начинающая неизвестная потребителю компания В.

Компания А располагает большим объемом капитала и может позволить себе содержать отдел рыночной аналитики. Маркетологи компании А подсчитали, какое количество клиентов они получат, если разместят свой киоск на одной из дорог с учетом того, как будет размещен киоск компании В.

Платежная матрица для компаний А выглядит следующим образом (таблица 12.1):

Таблица 12.1

Платежная матрица для А

<i>A</i>	<i>B</i> ₁	<i>B</i> ₂	<i>B</i> ₃
<i>A</i> ₁	800	700	600
<i>A</i> ₂	700	900	700
<i>A</i> ₃	600	700	800

Рассматривая платежную матрицу, можем сказать, например, следующее: если компания А разместит свой киоск на первой дороге (стратегия *A*₁), а компания В – также расположится на ней (стратегия *B*₁), то компания А привлечет 800 клиентов, оставив компании В только 200.

Целью данной антагонистической игры является поиск оптимальной для игрока А схемы размещения: на какой из дорог компании А необходимо расположиться для того, чтобы получить наибольшее количество клиентов и поставить компанию В в максимально невыгодное положение.

Для решения данной задачи используются принципы «Максимины» и «Минимакса».

Рассмотрим первый из них.

Принцип (или стратегия) «Максимин» обеспечивает максимальный из гарантированных выигрышей игрока А, вне зависимости от того, какие стратегии применяет игрок В.

При этом платежная матрица просматривается по строкам (для стратегий игрока А), и в каждой из них определяется минимальное значение. Далее из минимальных значений по строкам выбирается максимальное. Полученное значение и будет являться максимальным из гарантированных выигрышей (минимальным, но надежно обеспеченным).

$$\underline{v} = \max_j \min_i a_{ij}$$

Рассмотрим платежную матрицу из нашего примера.

Если компания А поставит свой киоск на первую дорогу (стратегия A_1), то она привлечет 600 клиентов, если на вторую дорогу (стратегия A_2) – 700 клиентов, если на третью (стратегия A_3) – снова 600. Максимальным из этих трех вариантов является вариант с выигрышем в 700 клиентов (таблица 12.2).

Таблица 12.2

Минимум по строкам у А

A	B_1	B_2	B_3	min по строке	
A_1	800	700	600	600	
A_2	700	900	700	700	max среди min
A_3	600	700	800	600	

Таким образом, если компания А поставит свой киоск на вторую дорогу, то гарантированно заработает 700 клиентов, и даже, возможно, 900, если компания В не сориентируется (выберет стратегию B_2).

$\underline{v} = 700$ – называется нижней ценой игры. Это значение является нижней границей игры в силу того, что игрок А, применяя данную стратегию не может заработать меньше, а напротив, может привлечь большее количество клиентов. Другими словами, гарантированный выигрыш игрока А не опустится ниже 700 человек при реализации, выявленной принципом «Максимины» стратегии.

Следующим принципом, рассматриваемым нами, является принцип «Минимакса».

Этот принцип (стратегия) обеспечивает минимальный проигрыш игрока В, вне зависимости от стратегий, применяемых игроком А. Принцип «Минимакса» является обратным к принципу «Максимины».

Если в предыдущем случае, мы просматривали платежную матрицу по строкам, то в этот раз мы рассматриваем её по столбцам (по стратегиям игрока В) в поисках максимальных значений. После этого из найденных максимальных значений по столбцам выбирается минимальное. Полученное значение и будет являться максимальным из гарантированных выигрышей (минимальным, но надежно обеспеченным).

$$\bar{v} = \min_i \max_j a_{ij}$$

Вернемся к нашему примеру. Если компания В расположит свой киоск на первой дороге (стратегия B_1), то компания А максимально привлечет 800 клиентов, если на вторую дорогу (стратегия B_2), то 900 клиентов, если на третью (стратегия B_3) – вновь 800. Минимальное значение из этих вариантов – 800 (таблица 12.3).

Таблица 12.3

Максимум по столбцам у В

A	B_1	B_2	B_3
A_1	800	700	600
A_2	700	900	700
A_3	600	700	800
max по столбцу	800	900	800

Это означает, что если игрок В поставит свой киоск на первую или на третью дорогу (то есть будет придерживаться стратегий (стратегия B_1 или B_2), то игроку А достанется минимум 800 клиентов (а оставшиеся 200 уйдут к игроку В).

$\bar{v} = 800$ – называется верхней ценой игры и означает, что количество клиентов игрока В гарантированно не опустится ниже 200 человек.

Стоит отметить, что оптимальное решение антагонистической игры будет находится где-то между нижней и верхней ценами игры. В нашем случае – в диапазоне между 700 и 800 клиентами.

Седловая точка

Предположим, что прошло какое-то время. Бренд компании В стал более известен, люди стали доверять ему и чаще покупать его кофе. Компания А всё ещё сильна, но расстановка сил в борьбе за клиента изменилась (таблица 12.4), причем, не в лучшую сторону для компании А (обратите внимание на обновленную платежную матрицу).

Таблица 12.4

Изменения в платежной матрице

A	B ₁	B ₂	B ₃
A ₁	600	500	400
A ₂	600	700	600
A ₃	400	500	600

Воспользуемся уже известными нам принципами «Минимакса» и «Максимины» (таблица 12.5).

Таблица 12.5

«Минимакса» и «Максимины»

A	B ₁	B ₂	B ₃	нижняя цена игры
A ₁	600	500	400	400
A ₂	600	700	600	600
A ₃	400	500	600	400
верхняя цена игры	600	700	600	

При измененной платежной матрице мы получили неординарный вариант: нижняя и верхняя цены игры совпали. Это означает, что получено оптимальное решение – найдена пара стратегий игроков, при которой оба получают

максимально возможный выигрыш, который называется чистой ценой игры. Если кто-то из игроков решит отклониться от выбранной оптимальной стратегии, то он проиграет.

$$v = \underline{v} = \bar{v} = \max_j \min_i a_{ij} = \min_i \max_j a_{ij}$$

Оптимальное решение игры называется «седловой точкой» – точкой, в которой сходятся минимум функции AXB и максимум функций DXC (рис. 12.1).

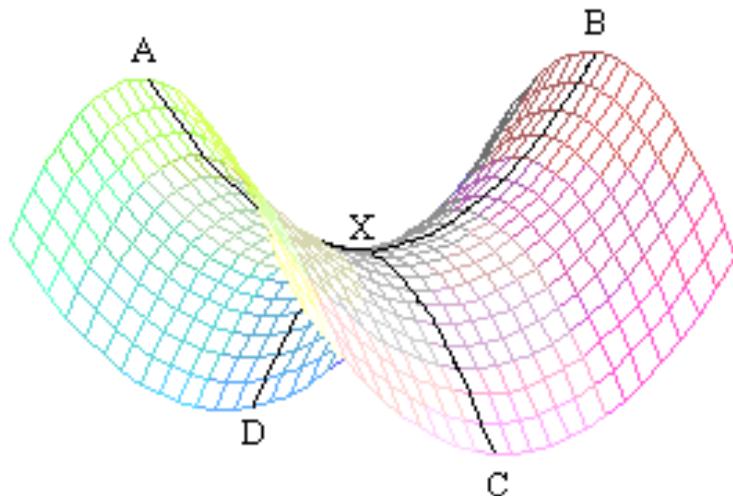


Рис. 12.1. Седловая точка X

В связи с этим была выведена теорема, по которой у игр с седловой точкой всегда есть определенная комбинация ходов, которая приводит к победе первого, то есть начавшего игру, игрока, вне зависимости от того, как будет ходить второй игрок.

Примером игры с седловой точкой можно назвать шахматы, так как в них объективно существует оптимальная комбинация ходов белых, при которой они все выигрывают. Однако она до сих пор не найдена, в связи с высокой вычислительной сложностью самой игры: древовидная структура всех возможных ходов шахмат настолько сильно ветвится, что на данных момент ни один суперкомпьютер не способен решить её полностью. Нахождение подобной стратегии означает конец шахмат, как таковых, ведь будет известно, какую стратегию выбирать, чтобы гарантировано выиграть.

Смешанные стратегии

Разошлем уже знакомый нам пример дальше.

Компания А, поняв, что теряет рынок, провела рекламную компанию и вернула своему кофе популярность. Судя по платежной матрице (таблица 12.6), привлекательность компании А приблизилась к своему начальному уровню. В ответ на действия компании А компания В придумала совершенно новую схему: передвижной киоск.

Таблица 12.6

Изменения в матрице

A	B ₁	B ₂	B ₃
A ₁	700	600	400
A ₂	500	800	600
A ₃	400	500	700

«Что, если мы будем ставить киоск каждый раз на новое место; найдется ли такая комбинация перемещений, которая заставит компанию А оставаться в невыгодном для них положении?» – подумали маркетологи компании В.

Действительно, в теории игр кроме чистых стратегий, которые мы рассматривали ранее, существуют также и смешанные стратегии.

Каждый день или каждую неделю меняя положение киоска, мы получим смешение стратегий, которое, возможно, поможет нам увеличить цену игры.

Стратегии смешиваются в определенных пропорциях для игроков А и В:

$$S_A = (p_1, p_2, p_3), S_B = (q_1, q_2, q_3)$$

Предположим, что компания А реализует свои стратегии в следующей последовательности (таблица 12.7):

Таблица 12.7

Последовательность реализации стратегий

Ходы:	A ₁	A ₂	A ₃	A ₁	A ₃	A ₂	A ₁	A ₁	A ₃	A ₁	A ₃	A ₂	A ₁	A ₃
-------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

Тогда вероятности для каждой из стратегий A_1, A_2, A_3 будут соответственно равны:

$$p_1 = \frac{6}{14} = 0,43; p_2 = \frac{3}{14} = 0,21; p_3 = \frac{5}{14} = 0,36$$

Значения вероятностей можно трактовать следующим образом: 43% времени компания А будет реализовывать стратегию A_1 , 21% времени – стратегию A_2 , а оставшиеся 36% времени – стратегию A_3 .

Но как узнать, что найденная смесь стратегий является оптимальной? В этом нам поможет решение задачи линейного программирования (ЗЛП), рассмотренной в части I.

Исходными данными для решения ЗЛП является транспонированная платежная матрица A^T .

Искомые переменные – x_1, x_2, x_3 (замененные вероятности p_1, p_2, p_3).

В качестве целевой функции выступает максимальная величина входного/выходного потока:

$$L = x_1 + x_2 + x_3 \rightarrow \min$$

Ограничения:

$$700x_1 + 500x_2 + 400x_3 \geq 1;$$

$$600x_1 + 800x_2 + 500x_3 \geq 1;$$

$$400x_1 + 600x_2 + 700x_3 \geq 1.$$

Далее осуществляется оптимизация с помощью симплекс-метода, выявляются искомые переменные, обеспечивающие минимум целевой функции.

Производится обратная замена переменных, после которой мы получаем искомые вероятности смешивания стратегий и цену игры:

$$\nu = \frac{1}{L_{\min}}, \quad p_1 = x_1\nu, \quad p_2 = x_2\nu, \quad p_3 = x_3\nu$$

Решение данной задачи покажет оптимальный вариант для игрока А.

Поиск оптимального решения для игрока В осуществляется через решение обратной задачи линейного программирования.

Исходными данными в этом случае будет нетранспонированная матрица А.

Искомые переменные – y_1, y_2, y_3 (замененные вероятности q, q_2, q_3).

В качестве целевой функции выступает максимальная величина входного/выходного потока:

$$L = x_1 + x_2 + x_3 \rightarrow \max$$

Ограничения:

$$700y_1 + 500y_2 + 400y_3 \leq 1;$$

$$600y_1 + 800y_2 + 500y_3 \leq 1;$$

$$400y_1 + 600y_2 + 700y_3 \leq 1.$$

Обратная замена переменных:

$$\nu = \frac{1}{L_{\max}}, \quad q_1 = y_1\nu, \quad q_2 = y_2\nu, \quad q_3 = y_3\nu$$

В рамках антагонистической игры, при нахождении оптимального варианта смешения стратегий для игрока В мы ищем наихудший вариант смешения стратегий игрока А.

Результаты решения трактуются следующим образом. Если игрок В будет каждый день перемещаться между первой и третьей дорогами, то сможет получить 450 клиентов ($1000 - 550$), в отличие от чистой стратегии, рассчитанной по принципу «Минимакс», при которой он получал бы гарантировано 300 клиентов.

Этим он вынуждает игрока А 3 дня из 4 (75%) выбирать вторую дорогу, но 1 день из 4 (25%) переходить на первую дорогу для того, чтобы получить хотя бы 550 клиентов. Если он не будет перемещаться, он получит 500 клиентов (таблица 12.8).

Таблица 12.8

Оптимальные смешанные стратегии

A	B_1	B_2	B_3	p_i
A_1	700	500	400	25%
A_2	600	800	500	75%
A_3	400	600	700	0
q_j	50%	0	50%	

Использование смешанных стратегий – оптимальный способ для нахождения какой-то точки, которая будет лежать между нижней и верхней ценами игры в чистых стратегиях. Нахождение этой точки позволит улучшить результаты как первого, так и второго игроков.

В смешанных стратегиях всегда присутствует только чистая цена игры.

Если привести аналогию с целыми и дробными значениями, то верхняя и нижняя цены игры – это целые значения, а цена игры в смешанных стратегиях – дробное.

Поясним: если решать какую-то задачу в целочисленной логике, то получится, что наилучшее значение находится, к примеру, между 3 и 4 (включая концы интервала). В каких-то случаях этого может быть вполне достаточно. Но если мы хотим получить лучшее решение, то необходимо взять дробное число, допустим, 3,85. В последнем случае мы получим больший процент оптимальности.

13. Биматричные игры

Как легко догадаться из названия, биматричные игры означают игры с двумя матрицами.

В каких же случаях нужны именно две матрицы, а не одна? В тех случаях, когда выигрыш одного игрока совершенно не означает проигрыш другого. Игроκи работают не в одном пуле преференций и выгод, а в параллельных областях, где каждый зарабатывает своё количество очков, клиентов, денег, при том, что заработка каждого зависит от действий противника.

Ходы также одновременные. Стратегии могут быть как чистыми, так и смешанными.

Самая знаменитая биматричная игра – это, так называемая, «дilemma заключенного».

Классическая формулировка теоремы звучит так: Двоє преступників – А і В – попались примерно в одно и то же время на сходных преступлениях. Есть основания полагать, что они действовали по сговору, и полиция, изолировав их друг от друга, предлагает им одну и ту же сделку: если один свидетельствует против другого, а тот хранит молчание, то первый освобождается за помощь следствию, а второй получает максимальный срок лишения свободы (10 лет). Если оба молчат, их деяние проходит по более лёгкой статье, и каждый из них приговаривается к году тюрьмы. Если оба свидетельствуют друг против друга, они получают по 5 лет. Каждый заключённый выбирает, молчать или свидетельствовать против другого. Однако ни один из них не знает точно, что сделает другой. Что произойдёт?

Платежная матрица выглядит следующим образом (рис. 13.1):

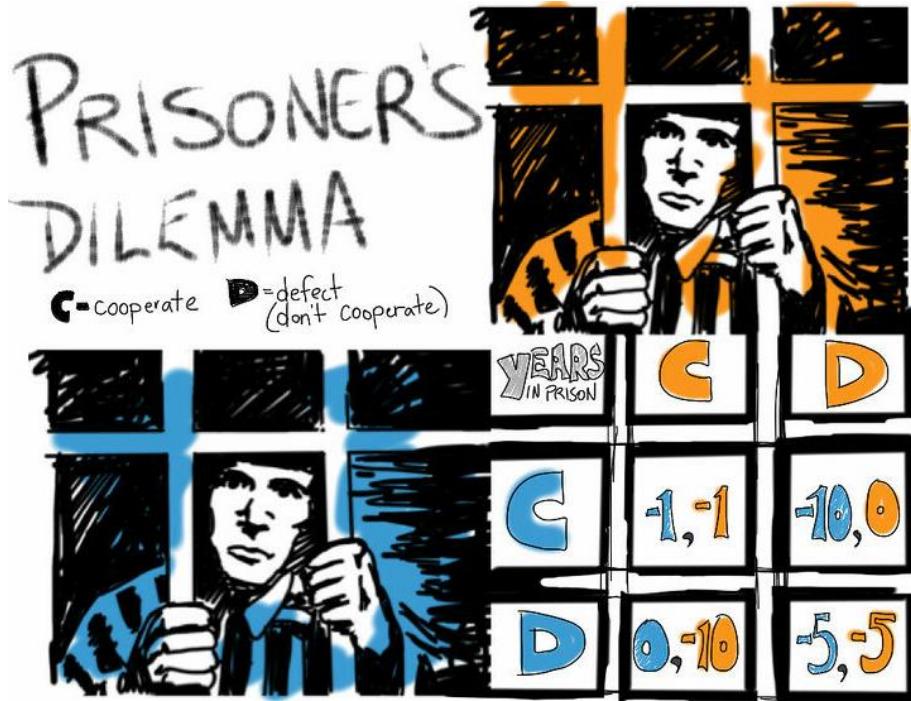


Рис. 13.1. Платежная матрица

Задачу можно решать с помощью поиска двух вариантов равновесия: равновесия по Нэшу (устойчивое равновесие) и простого равновесия (неустойчивое равновесие, или парето-эффективное равновесие) (рис. 13.2).

$[A,B]$	Молчать	Сдать
Молчать	[1,1]	[10,0]
Сдать	[0,10]	[5,5]

$[A,B]$	Молчать	Сдать
Молчать	[1,1]	[10,0]
Сдать	[0,10]	[5,5]

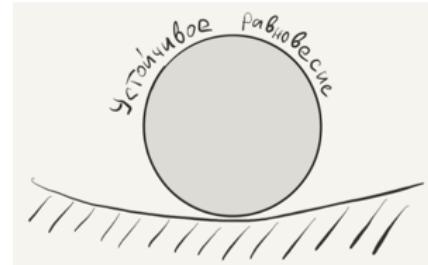
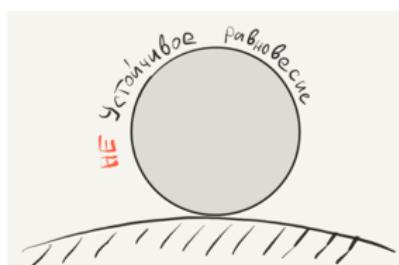


Рис. 13.2. Варианты равновесия

Равновесие по Нэшу – такое сочетание стратегий, при котором игроки не доверяют друг другу и выбирают такой вариант, который гарантированно

позволит им выйти с минимальными потерями. При этом каждому из них не выгодно отступать от выбранной стратегии вне зависимости от обстоятельств.

Противовесом равновесия по Нэшу выступает равновесие по Парето – такая ситуация, когда игроки доверяют друг другу и договариваются выбирать такой вариант, при котором оба получат максимальный выигрыш. Но есть один недостаток – если игрок отклонится от выбранной стратегии, его выигрыш увеличится, а выигрыш противника уменьшится.

Простое равновесие в теории игр практически не используется в силу того, что подобные ситуации встречаются крайне редко.



Джон Форбс Нэш (англ. John Forbes Nash, 13.06.1928 - 23.05.2015). В 1950–1953 годах Нэш будучи учеником Джона фон Неймана в Принстонском университете опубликовал четыре революционные работы в области игр с ненулевой суммой. Он обнаружил возможность «некооперативного равновесия», при которой обе стороны используют стратегию, приводящую к устойчивому равновесию («равновесие Нэша»). За свою 28-страничную диссертацию о некооперативных играх

Нэш получил в 1950 году докторскую степень. Лауреат Нобелевской премии 1994 года по экономике за «Фундаментальный анализ равновесия в теории некооперативных игр».

Чистые стратегии

Вернемся к уже знакомым нам компаниям А и В. Компания А поняла, что конкуренция разрушает бизнес и отнимает клиентов у обеих, и предложила договориться: компании А и В будут обособленно реализовывать кофе и снеки, что позволит им обслуживать одних и тех же клиентов.

Маркетологи обеих компаний рассчитали количество клиентов в зависимости от расположения киосков. На какие дороги необходимо поставить киоски, чтобы обе компании получили максимальное число клиентов.

В данном случае нет необходимости делить 1000 клиентов: в силу того, что компании реализуют различные товары (например, компания А продает кофе, а компания В – снеки), их клиентская база может пересекаться.

Если компании выберут правильные стратегии взаимодействия, то привлекут большее количество клиентов и получат большую прибыль.

Платежные матрицы будут выглядеть следующим образом (таблица 13.1).

Таблица 13.1

Платежные матрицы игроков А и В

A	B ₁	B ₂	B ₃
A ₁	500	400	300
A ₂	600	700	600
A ₃	300	400	500

B	B ₁	B ₂	B ₃
A1	500	300	200
A2	400	600	400
A3	200	300	500

Для отыскания оптимальных стратегий, найдем равновесие по Нэшу.

Как происходит поиск равновесия по Нэшу? Игрок А ищет наиболее выгодные варианты в своей матрице для игрока В (максимумы по столбцам матрицы А). Игрок В также ищет наиболее выгодные для оппонента варианты в своей матрице (максимумы по строкам матрицы В). В тех ячейках, где отмеченные выгодные варианты совпали в обоих матрицах и наблюдается ситуация равновесия по Нэшу.

Если компания А будет применять первую стратегию (стратегия A₁), компания В максимально может привлечь 500 клиентов, если будет выбрана вторая стратегия (стратегия A₂) – 600 клиентов, если третья (стратегия A₃) – также 500. Таким же образом происходит выбор во второй платежной матрице, но уже для игрока А.

В нашем случае пара, образовавшая равновесие по Нэшу, и соответственно самая выгодная пара для обоих игроков – стратегии A₂ и B₂ (таблица 13.2)

Таблица 13.2

Наиболее выгодная пара для обоих игроков

A	B ₁	B ₂	B ₃
A ₁	500	400	300
A ₂	600	700	600
A ₃	300	400	500
max	600	700	600

B	B ₁	B ₂	B ₃	max
A1	500	300	200	500
A2	400	600	400	600
A3	200	300	500	500

Если какая-то из компаний нарушит договоренность, получит меньшее количество клиентов.

Смешанные стратегии

Через некоторое время предпочтения работников офиса немного изменились, и ситуация равновесия пропала. Теперь появились две ситуации «полуравновесия», каждая из которых выгоднее только одной из фирм А или В.

Отношения между ними немного охладели, было принято решение чередовать ситуации равновесия, но о графике расстановки киосков они не договорились. Как каждому из них оптимально расставлять киоски на своих дорогах?

Новые платежные матрицы для компаний выглядят следующим образом (таблица 13.3):

Таблица 13.3

Новые платежные матрицы

A	B ₁	B ₂	B ₃
A ₁	500	400	300
A ₂	500	700	500
A ₃	600	300	600

B	B ₁	B ₂	B ₃
B ₁	500	700	500
B ₂	400	600	300
B ₃	700	300	500

Теперь, как можно видеть, возникло два варианта потенциального равновесия, то есть возникла ситуация «полуравновесия» или «качающегося равновесия». В одном случае компания А получает больше прибыли, в другом – компания В привлекает большее количество клиентов.

В подобной ситуации можно применить методологию смешанных стратегий.

Для того, чтобы определить пропорцию, необходимо решить задачу нелинейного программирования.

Целевая функция в данном случае нелинейного вида, и представляет собой сумму выигрышей обоих игроков.

$$H_A(p, q) + H_B(p, q) \rightarrow \max$$

где ситуации равновесия:

$$H_A(p, q) = \sum_{i=1}^m \sum_{j=1}^n a_{ij} p_i q_j, \quad H_B(p, q) = \sum_{i=1}^m \sum_{j=1}^n b_{ij} p_i q_j$$

Ограничения:

$$\left\{ \sum_{j=1}^n a_{ij} q_j \leq H_A(p, q) \mid i = 1 \dots m \right\}, \quad \left\{ \sum_{i=1}^m b_{ij} p_i \leq H_B(p, q) \mid j = 1 \dots n \right\},$$

$$\sum_{i=1}^m p_i = 1, \quad \sum_{j=1}^n q_j = 1, \quad q_j \geq 0, \quad p_i \geq 0$$

Такая задача может быть решена различными методами, например, методом градиента.

По результатам решения задачи нелинейного программирования (ЗНЛП) был получен баланс между двумя ситуациями «полуравновесия».

Игрок А будет проводить на второй дороге 2 из 3 дней (67%), а на третьей дороге оставшийся день (33%). Игрок В должен проводить на первой дороге 4 из 5 дней (80%) и на 1 день выставляться на второй дороге. В случае такого варианта расстановки киосков игрок А получает 540 клиентов, а игрок В -500 клиентов. В связи с тем, что киоски расставляются случайно (без графика), происходят потери клиентов при выпадении пар стратегий (A_2, B_2) и (A_3, B_1) (таблица 13.4).

Таблица 13.4

Оптимальные смешанные стратегии

<i>A</i>	<i>B</i> ₁	<i>B</i> ₂	<i>B</i> ₃	<i>p</i> _{<i>i</i>}
<i>A</i> ₁	500	400	300	0
<i>A</i> ₂	500	700	500	67%
<i>A</i> ₃	600	300	600	33%
<i>q</i> _{<i>j</i>}	80%	20%	0%	

<i>B</i>	<i>B</i> ₁	<i>B</i> ₂	<i>B</i> ₃	<i>p</i> _{<i>i</i>}
<i>A</i> ₁	500	700	500	0
<i>A</i> ₂	400	600	300	67%
<i>A</i> ₃	700	300	500	33%
<i>q</i> _{<i>j</i>}	80%	20%	0%	

Таким образом, компании А нужно уделять 80% времени стратегии, выгодной для неё, и 20% времени – стратегии, выгодной для компании В. А компании В нужно 67% своего времени уделять выгодной для себя стратегии, а 33% – стратегии, выгодной для соседа.

14. Игры с «природой»

Перейдем к еще одному типу игры в теории игр – к играм с природой, в которых в качестве второго игрока выступает так называемая «природа».

В данном типе игр существует большая доля неопределенности и вариативности в поведении «природы». «Природа» не стремится сделать хуже или лучше, она живет по своим законам, которые часто нам неизвестны, и имеет свои задачи, которые нас не касаются.

Примеры «природы»:

- поведение диких животных;
- реальные погодные условия;
- изменение потребительского спроса;
- колебание курса ценных бумаг и др.

Данному типу игр в экономике отдается предпочтение, так как все большинство ситуаций, рассматриваемых экономической наукой, подпадают под условия риска или неопределенности (например, колебания курса акций).

Принятие решений в условиях полной или частичной неопределенности – задача теории игр с «природой».

От того, знаем ли мы поведение «природы», зависит тип игры:

1. Игра с риском (или с частичной неопределенностью) – в данном случае мы приблизительно, с определенной долей вероятности, знаем, какое состояние природы наступит.

2. Игра с неопределенностью (полной неопределенностью) – мы не знаем вероятность возникновения состояний природы.

Начнем с игр с природой в условиях риска (вероятности наступления состояний природы известны).

В принятии решений нам помогут три человека: Байес, Лаплас и Гермейер, которые с разницей в 100 лет придумывали критерии, позволяющие определить наилучший вариант действий в условиях риска.

Принятие решений в условиях риска

Владелец небольшого магазина в начале каждого дня закупает для реализации некий скоропортящийся продукт по цене 50 рублей за единицу. Цена реализации этого продукта — 60 рублей за единицу. Из наблюдений за работой магазина известно, что спрос на этот продукт за день может быть равен 1, 2, 3 или 4 единицам.

Пусть известно, что на практике спрос 1 наблюдался 15 раз, спрос 2 наблюдался 30 раз, спрос 3 наблюдался 30 раз, спрос 4 наблюдался 25 раз. Если продукт в течение дня не распродан, то в конце дня его всегда покупают по цене 30 рублей за единицу. В данном случае мы имеем возможность определить какое-то процентное отношение возникновения каждого случая.

Сколько единиц этого продукта должен закупать владелец магазина каждый день, чтобы минимизировать свой риск и повысить свою прибыль?

Для начала необходимо составить платежную матрицу, в которой будут расписаны стратегии игрока А и стратегии «природы». Заполняем матрицу соответствующими значениями исходя из условия задачи – на пересечении строк и столбцов матрицы находятся финансовые последствия (таблица 14.1).

Таблица 14.8

Стратегии игрока А и состояния «природы»

Стратегии игрока А	Состояния «природы»			
Объем закупки, ед./день	Спрос в течение дня, единиц продукта / день			
	1	2	3	4
1	10	10	10	10
2	-10	20	20	20
3	-30	0	30	30
4	-50	-20	10	40
	$p_1 = 0,15$	$p_2 = 0,3$	$p_3 = 0,3$	$p_4 = 0,25$

Так как объемы спроса известны, мы можем просчитать вероятность наступления каждого состояния природы:

$$p(1) = \frac{15}{15 + 30 + 30 + 25} = 0,15$$

$$p(2) = \frac{30}{15 + 30 + 30 + 25} = 0,30$$

$$p(3) = \frac{30}{15 + 30 + 30 + 25} = 0,30$$

$$p(4) = \frac{25}{15 + 30 + 30 + 25} = 0,25$$

Исходя из этого, нам необходимо определить, какую стратегию поддержать (какая из них даст наибольшую прибыль).

Критерий Байеса

Воспользуемся критерием Байеса: определим стратегию, которая суммарно принесет нам максимальную прибыль. Для этого вычисляем средневзвешенное значение выигрыша каждой стратегии игрока А. Наибольшее значение из полученных будет являться нашей оптимальной стратегией (таблица 14.2).

$$B_i = \sum_{j=1}^n p_j a_{ij}, i = \overline{1, m}$$

Таблица 14.2

Оптимальная стратегия по критерию Байеса

	Природа, x	Вероятность, p	Ср. взвешенное, $\Sigma(x \cdot p)$
Стратегия 1	10	0,15	$10 \cdot 0,15 = 1,5$
	10	0,30	$10 \cdot 0,30 = 3$
	10	0,30	$10 \cdot 0,30 = 3$
	10	0,25	$10 \cdot 0,25 = 2,5$
	Итого	1,00	$B_i = 10$
Стратегия 2	-10	0,15	$-10 \cdot 0,15 = -1,5$
	20	0,30	$20 \cdot 0,30 = 6$
	20	0,30	$20 \cdot 0,30 = 6$
	20	0,25	$20 \cdot 0,25 = 5$
	Итого	1,00	$B_i = 15,5$
Стратегия 3	-30	0,15	$-30 \cdot 0,15 = -4,5$
	0	0,30	$0 \cdot 0,30 = 0$
	30	0,30	$30 \cdot 0,30 = 9$
	30	0,25	$30 \cdot 0,25 = 7,5$
	Итого	1,00	$B_i = 12$
Стратегия 4	-50	0,15	$-50 \cdot 0,15 = -7,5$
	-20	0,30	$-20 \cdot 0,30 = -6$
	10	0,30	$10 \cdot 0,30 = 3$
	40	0,25	$40 \cdot 0,25 = 10$
	Итого	1,00	$B_i = -0,5$

Критерий Лапласа

Данный критерий применяется в том случае, если вероятности наступления состояний природы неизвестны. Мы принимаем их как равновероятные и вычисляем среднеарифметическое (а не средневзвешенное, как в критерии Байеса) значение выигрыша игрока А по строке:

$$B_i = \frac{1}{n} \sum_{j=1}^n a_{ij}, i = \overline{1, m}$$

Вновь выбираем стратегию, по которой средний выигрыш будет максимальным (таблица 14.3).

Таблица 14.3

Оптимальная стратегия по критерию Лапласа

	Природа, x	Вероятность, p	Среднее арифметическое
Стратегия 1	10	0,25	10
	10	0,25	10
	10	0,25	10
	10	0,25	10
	Итого	0,25	$B_1 = 10$
Стратегия 2	-10	0,25	-10
	20	0,25	20
	20	0,25	20
	20	0,25	20
	Итого	0,25	$B_2 = 12,5$
Стратегия 3	-30	0,25	-30
	0	0,25	0
	30	0,25	30
	30	0,25	30
	Итого	0,25	$B_3 = 7,5$
Стратегия 4	-50	0,25	-50
	-20	0,25	-20
	10	0,25	10
	40	0,25	40
	Итого	0,25	$B_4 = -5$

Критерий Гермейера

Критерий Гермейера несколько сложнее уже рассмотренных нами критериев, однако он позволяет найти некие альтернативные варианты.

По критерию Гермейера игрок может получить свой выигрыш a_{ij} с вероятностью p_j . В связи с этим рассматривается элемент Гермейера $p_j a_{ij}$, из которых строится матрица Гермейера (таблица 14.4).

Таблица 14.4

Матрица Гермейера

Стратегии игрока А		Состояния «природы»			
Объем закупки, ед./день	Спрос в течение дня, единиц продукта / день				
	1	2	3	4	
1	1,5	3	3	2,5	
2	-1,5	6	6	5	
3	-4,5	0	9	7,5	
4	-7,5	-6	3	10	

Элементы этой новой матрицы содержат уже не просто значения выигрыша, а значения выигрыша, умноженного на вероятность этого выигрыша. К этой матрице можно применять все методы, применяемые к платежным матрицам в матричных играх, рассмотренных ранее. Например, мы можем использовать принцип максимина, чтобы получить наибольший гарантированный выигрыш (таблица 14.5):

$$G = \max_i \min_j p_j a_{ij}$$

Таблица 14.5

Оптимальная стратегия по критерию Гермейера

Стратегии игрока А		Состояния «природы»				
Объем закупки, ед./день	Спрос в течение дня, единиц продукта / день				min по стр.	
	1	2	3	4	max из min	
1	1,5	3	3	2,5	1,5	
2	-1,5	6	6	5	-1,5	
3	-4,5	0	9	7,5	-4,5	
4	-7,5	-6	3	10	-7,5	

Критерий Гермейера показывает гарантированный выигрыш в самом неблагоприятном состоянии природы.

Ситуации риска и неопределенности возникают по причине того, что мы не обладаем полной информацией. Так как информация частичная мы вынуждены рисковать. Если бы мы знали, какой ход сделает природа – наша игра стала бы игрой обычной антагонистической игрой с полной информацией. Для того, чтобы уменьшить долю риска, необходимо провести тщательное исследование.



Юрий Борисович Гермейер (18 июля 1918, Аткарск – 24 июня 1975, Москва) — советский учёный, специалист в области прикладной математики, исследования операций и теории игр. 3 июля 1941 года Ю. Б. Гермейер получил диплом с отличием и был направлен в Сталинград инженером-расчётом. В 1966 году перешёл в Вычислительный центр Академии наук СССР, организовал научный семинар по исследованию операций и теории игр и до конца жизни руководил его работой. В 1970 году основал кафедру исследования операций на факультете вычислительной математики и кибернетики МГУ им. М.В. Ломоносова. Награжден орденами «Трудового Красного Знамени», «Знак Почёта», медалью «За доблестный труд в Великой Отечественной войне 1941-1945 гг.».

Принятие решений в условиях неопределенности

В данном случае количество рассматриваемых критериев значительно больше, так как задача, стоящая перед нами, сложнее в силу того, что мы ничего не знаем об окружающей среде. Решение игр в условиях неопределенности, при которой вероятности наступления состояний природы неизвестны, осуществляется с помощью критерием оптимизма (или «максимакса»), пессимизма (или «минимина»), критерия Вальда (или «максимины»), критерия Гурвица (линейная свертака) или критерия Сэвиджа (на матрице рисков).

Для получения наилучшего результата и принятия максимального решения рекомендуется решать задачу по каждому критерию и затем выбирать стратегию, на которую указывает большинство. В этом случае постановка задачи

остается такой же, с той лишь разницей, что вероятности наступления состояний природы неизвестны.

Владелец небольшого магазина в начале каждого дня закупает для реализации некий скоропортящийся продукт по цене 50 рублей за единицу. Цена реализации этого продукта — 60 рублей за единицу. Если продукт в течение дня не распродан, то в конце дня его всегда покупают по цене 30 рублей за единицу. Сколько единиц этого продукта должен закупать владелец магазина каждый день?

Критерии оптимизма и пессимизма

Для начала можно оценить нижнюю и верхнюю границы игры — сколько мы максимум можем потратить и сколько максимум заработать. Это возможно определить с помощью критериев «Пессимизма» (или крайнего пессимизма) и «Оптимизма» (или крайнего оптимизма).

$W = \min_i \min_j a_{ij}$ — критерий пессимизма или "минимин"

$W = \max_i \max_j a_{ij}$ — критерий оптимизма или "максимакс"

При пессимистическом сценарии мы получаем самые большие издержки, а при оптимистическом — максимизацию прибыли (таблица 14.6).

Таблица 14.6

Пессимистический и оптимистический сценарий

Объем закупки, ед. / день	Спрос в течение дня, единиц продукта / день				$\min \min$
	1	2	3	4	
1	10	10	10	10	10
2	-10	20	20	20	-10
3	-30	0	30	30	-30
4	-50	-20	10	40	-50

Объем закупки, ед. / день	Спрос в течение дня, единиц продукта / день				$\max \max$
	1	2	3	4	
1	10	10	10	10	10
2	-10	20	20	20	20
3	-30	0	30	30	30
4	-50	-20	10	40	40

Критерий Вальда

Где-то между результатами каждого из критериев находится та самая золотая середина, которая обнаруживается при помощи других, например, критерия Вальда (таблица 14.7).

Критерий Вальда предлагает искать самый надежный сценарий по принципу «Максимина» (искать гарантированную прибыль от каждой стратегии).

$$W = \max_i \min_j a_{ij}$$

Таблица 14.7

Сценарий по критерию Вальда

Объем закупки, ед. / день	Спрос в течение дня, единиц продукта / день				max min
	1	2	3	4	
1	10	10	10	10	10
2	-10	20	20	20	-10
3	-30	0	30	30	-30
4	-50	-20	10	40	-50

Новым предприятиям рекомендуется придерживаться стратегии Вальда, чтобы излишне не рисковать и получать минимальную, но гарантированную прибыль. Однако выбор нижней границы может быть неподходящим для других предприятий, которые готовы рискнуть, чтобы получить больше.

Критерий Гурвица

Для того, чтобы учесть склонность человека к риску существует следующий критерий – критерий Гурвица. Все математические приемы, которые были до этого, применяются и здесь.

Гурвиц предлагает использовать линейную свертку, чтобы выйти на компромиссный вариант, настраивая весовой коэффициент пессимизма и оптимизма, который регулирует отношение к риску (таблица 14.8).

$$H = \max_i (\alpha \max a_{ij} + [1 - \alpha] \min a_{ij})$$

Так, например, значение коэффициента равное 0,4 означает, что нам на 40% важны потери, а на 60% – прибыль, и мы готовы рискнуть.

Отношение к риску – это функция полезности из одноименной теории полезности, которая определяет, насколько какой-то результат является для нас более полезным.

Таблица 14.8

Сценарий по критерию Гурвица

Объем за- купки, ед. /день	Спрос в течение дня, ед./день				\max avg
	1	2	3	4	
1	10	10	10	10	10
2	-10	20	20	20	8
3	-30	0	30	30	6
4	-50	-20	10	40	4

Человек принимает решение не в условиях того, сколько он получит, а в условия того, насколько это для него полезно. Если игрок безразличен к риску, функция полезности примет вид прямой. Для человека, который более склонен к риску, эта функция выгнута вниз. Он готов получить большую прибыль, понеся при этом большие потери. Если функция выгнута вверх, то для каждого значения выигрыша, он готов понести меньшие потери. Мы меняем изгиб функции, меняя значения параметра.

Критерий Сэвиджа

Из ряда всех этих критериев выбивается ещё один критерий – критерий Сэвиджа, который так же, как и критерий Гермейера, создает свою собственную матрицу. Это так называемая матрица упущенной выгоды (значение того, сколько денег мы могли бы заработать максимально, но не заработали).

Среди элементов исходной матрицы ищем максимальное значение по столбцам (таблица 14.9).

Заполняем матрицу упущенной выгоды (или матрицу рисков) соответствующими значениями – разностями максимума по столбцу и значения текущей ячейки (таблица 14.10).

$$r_{ij} = \max_i a_{ij} - a_{ij}$$

Таблица 14.9

Исходная матрица

Объем за- купки, ед. /день	Спрос в течение дня, единиц продукта / день			
	1	2	3	4
1	10	10	10	10
2	-10	20	20	20
3	-30	0	30	30
4	-50	-20	10	40

Если в ячейках матрицы стоят 0 – значит, упущенной выгоды нет. К данной матрице вновь можно применить один из методов, работы с матрицами в антагонистической игре (например, метод «минимакса») (таблица 14.10).

Таблица 14.10

Применение метода «минимакса» к матрице упущенной выгоды

Объем за- купки, ед. /день	Спрос в течение дня, единиц продукта / день				min max
	1	2	3	4	
1	0	10	20	30	30
2	20	0	10	20	20
3	40	20	0	10	40
4	60	40	20	0	60



Леонид Соломонович Гурвич (англ. **Leonid Hurwicz**; 21 августа 1917 - 24 июня 2008) — американский экономист российского происхождения, почётный профессор Университета Миннесоты. Работал в Комиссии Коулса, лауреат премии по экономике памяти Альфреда Нобеля за 2007 год за работу совместно с Эриком Маскинами Роджером Майерсоном над теорией оптимальных механизмов. Старейший экономист, получивший премию памяти Альфреда Нобеля (это произошло в 90 лет). Гурвичу и его коллегам удалось создать теорию, помогающую выявлять эффективные торговые механизмы и схемы регулирования экономики, а также определять, насколько в той или иной ситуации необходимо вмешательство государства.

15. Позиционные игры с противником

Следующая категория игр теории игр, которую мы рассмотрим – позиционные игры.

Позиционные игры – одни из самых распространённых и наиболее приближенных к реальности математических моделей. Это не означает, что рассмотренные ранее модели оторваны от реальности. Напротив, многие математические модели и инструменты, с которыми мы ранее познакомились, необходимы для понимания позиционных игр и применимы к ним.

Позиционная игра – игра, в которой противники ходят по очереди. В этом заключается основное отличие их от рассмотренных ранее игр: в матричных или биматричных играх, играх с природой или без неё игроки ходят одновременно (вспоминаем игру «Камень–Ножницы–Бумага»), происходит выбор одной из стратегий в надежде, что противник выберет наименее выгодную для себя в сложившихся условиях стратегию поведения. Такие ситуации одновременных ходов противников безусловно встречаются, но ситуации, когда противники ходят по очереди встречаются значительно чаще. При последовательных ходах игроки могут оценить состояние игрового поля после предшествующего хода противника и ответить самой лучшей своей стратегией.

Шахматы – яркий пример позиционной игры: противники ходят друг за другом по очереди, оценивая сложившуюся ситуацию.

Ходы позиционной игры строго последовательны и могут быть представлены в виде дерева, в котором каждый вариант стратегии разворачивается друг за другом, отражая зависимость всех вариантов. Именно поэтому позиционные игры по-другому называются играми в развернутой форме.

Дерево (в теории графов) – связанный ациклический (без наличия замкнутых контуров) граф, имеющий «корень», «ветви» и «листья». «Корень» – вершина, дающая начало дереву, «ветви» – промежуточные вершины, «листья» – набор терминальных (конечных) вершин.

Дерево в виде графа может быть, как бинарным (состоит из вершин, имеющих не более двух выходов – да или нет, 0 или 1), так и ориентированным (указывает направление движения).

Дерево игры – это совокупность всевозможных ходов игроков из всех возможных позиций и выигрышней/проигрышней, получаемых в результате этих ходов. Конечным элементам дерева игры (листьям) соответствует цена игры данной совокупности ходов.

Дерево игры может быть изображено для двух и более игроков. Для каждого хода отображаются возможные варианты развития событий и для каждого варианта рассчитываются прибыли, которые получит каждый из игроков после конкретного хода. Стоит отметить, что под прибылью также могут пониматься отрицательные значения (потери от хода).

Стратегии игроков могут быть только чистыми: смешивать стратегии нет необходимости в силу того, что игроки ходят друг за другом и могут адекватно реагировать на изменяющуюся ситуацию.

По своей идеологии позиционные игры близки к антагонистическим – противники действуют в одном поле и делят между собой один банк (клиентскую базу, прибыль, сегмент рынка), или могут быть подобны биматричным – каждый игрок набирает очки индивидуально.

Построение дерева игры

Любую игру, где противники ходят друг за другом, любую последовательность ходов можно представить в виде дерева, на каждом шаге которого, располагаются ходы противников и результаты этих ходов.

В качестве примера рассмотрим игру в крестики-нолики. Несмотря на то, что игра достаточно простая и количество возможных ходов не столь велико, дерево игры достаточно обширно. Поэтому рассмотрим только кусочек этого дерева для формирования понимания вопроса построения дерева игры.

Ситуация на поле, от которой мы отталкиваемся, перед ходом «крестика» представлена на рис. 15.1.

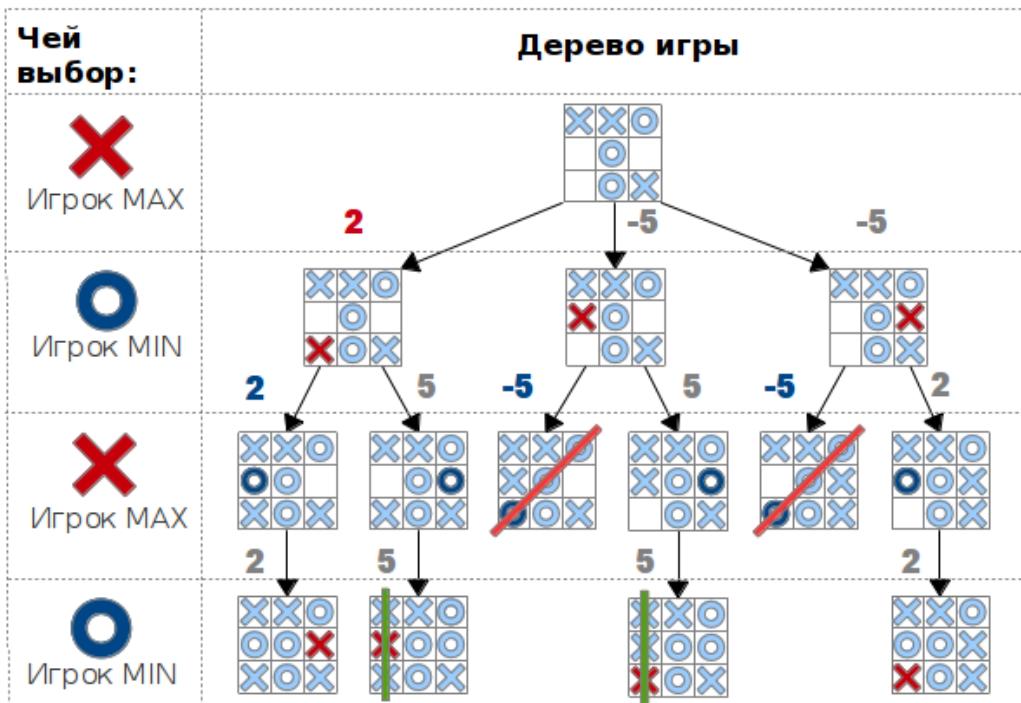


Рис. 15.1. Дерево завершения игры

Возможных вариантов хода в данной ситуации всего три (по количеству оставшихся на поле клеток). Каждый ход имеет свою стоимость: в первом случае «крестик» получает 2 условных балла, во втором и третьем – теряет 5 очков.

В данном дереве цена игры указана только для «крестика».

Следующая горизонталь дерева игры – ход «нолика». В каждой образовавшейся ветке существует два возможных варианта (суммарно шесть возможностей хода). Благодаря ходу «нолика» «крестик» может заработать 2 балла (в двух вариантах развития событий), получить 5 баллов (также в двух вариантах) или потерять их (оставшиеся два варианта). Варианты, в которых «крестик» теряет 5 баллов являются терминальными (несмотря на то, что расположены на предпоследнем ряду), так как заканчиваются для него проигрышем.

Важно отметить, что в дереве игры отображаются все возможные варианты: логичные и нелогичные, приводящие к успеху и влекущие за собой убытки.

Последний уровень дерева вновь отображает ход «крестика». На данном уровне отображены четыре варианта, два из которых заканчиваются ничьей, а два оставшихся – выигрышем «крестика».

Возникает вполне логичный вопрос: какой вариант выбрать? какой сделать ход, чтобы прийти к желаемому результату?

В классической позиционной игре в теории игр считается, что оба игрока стремятся к выигрышу. Поэтому при выборе между ничьей и выигрышем будет выбран второй вариант.

Также необходимо принимать во внимание такой аспект, как горизонт планирования (прогнозирования) – количество уровней дерева, на которых игрок может предвидеть возможные варианты развития событий. Горизонт планирования индивидуален для каждого игрока и может зависеть от внешних условий.

Ответ на вопрос «Какой же вариант выбрать?» дается в двух вариантах игр: позиционной игре с противником и позиционной игре с природой.

Позиционная игра с противником

Пусть имеются два производителя мобильных устройств – компании А и В. Обе компании работают на одном рынке и конкурируют друг с другом. Компании могут производить как мобильные телефоны, так и планшеты. Первая компания разрабатывает уникальный мобильный телефон и выходит с ним на рынок (емкость рынка телефонов \$200M). В ответ на это вторая компания может либо разработать аналогичный аппарат, либо попытаться захватить рынок планшетов (емкость рынка планшетов \$150M). После этого первая компания снова может вложить средства в развитие мобильного направления, либо выйти на рынок планшетов. Компания В в ответ на это может совершить аналогичные действия. Экономический эффект от каждого действия известен (отображен на дереве игры).

Необходимо найти оптимальные стратегии компаний А и В, приводящие их к захвату рынков.

Действие игры происходит в течение двух периодов времени (например, 2 лет), причем, каждые полгода фирмы по очереди предпринимают какие-то действия. В первый период времени – компания А делает свой ход, через полгода компания В делает ответный шаг (полгода – время реагирования на действие первой компании).

Для того, чтобы решить конкретную задачу необходимо создать дерево игры. Его построение начинается с указания корня – в нашем случае, выбора хода компании А.

Предположим, компания А решила начать с выхода на рынок мобильных телефонов, заполучив тем самым 150M рынка. В ответ на это компания В также решает заняться выпуском мобильных телефонов и забирает у первого игрока 50M (рис. 15.2).

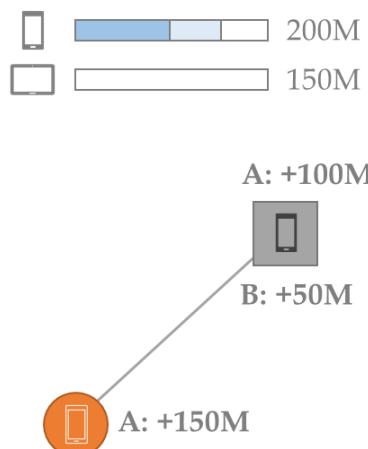


Рис. 15.2. Ход компании В во вторые полгода

Спустя полгода компания А вновь вкладывается в телефонное направление. В результате этого хода у компании А – 150M, у компании В – 40M. Компания В следует примеру компании А и продолжает работать на рынке мобильных телефонов (рис. 15.3).

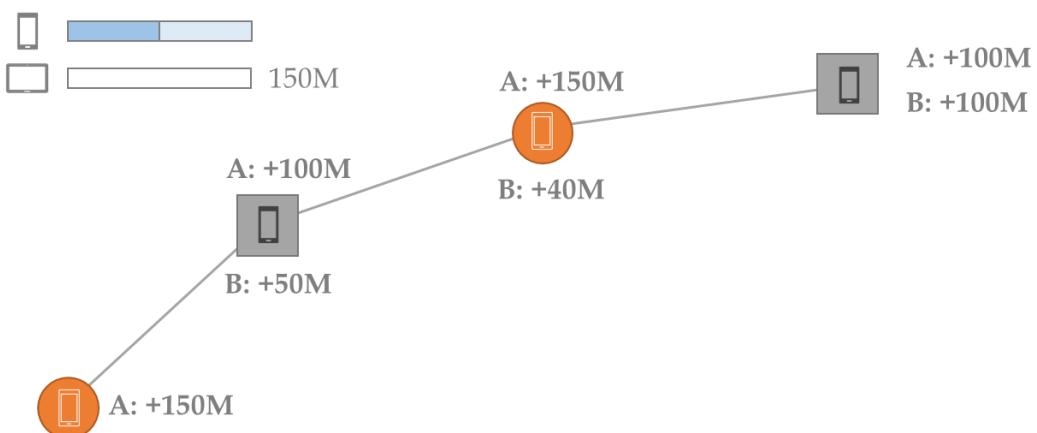


Рис. 15.3. Первый вариант стратегии игры

Таким образом в терминальной точке компании А и В поровну поделили между собой рынок и теперь владеют 100М каждая.

Попробуем реализовать другой вариант развития событий. Скажем, игрок А после первого хода игрока В решает сменить рынок и выпустить планшет. В результате хода первого игрока получает 200М рынка, второй – 70М. В ответ на это компания В также переходит на рынок планшетов. Прибыли компаний А и В в терминальной точке 180М и 130М соответственно (рис. 15.4).

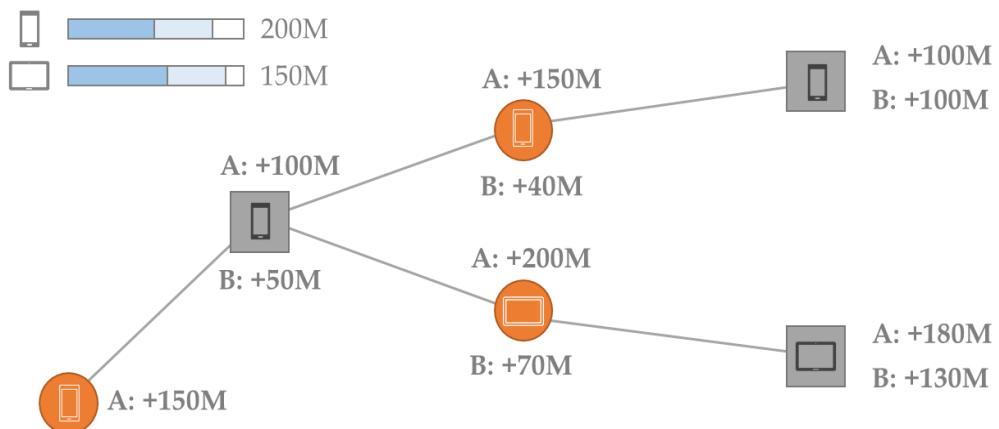


Рис. 15.4. Второй вариант стратегии

Если же компания В решает сразу выпускать планшеты и не мешать компании А производить телефоны, то у компании А – 170М, а у компании В – 100М рынка. Несмотря на этот маневр, компания А продолжает совершенствовать свой смартфон, работая на рынке мобильных устройств, выигрывая тем самым 200М.

Компания В решает поддержать компанию А в стратегии невмешательства и вновь вкладывается в выпуск планшетов. В итоге каждый из производителей максимально реализуется на своем рынке. Прибыль компании А – 220М, компании В – 170М (рис. 15.5).

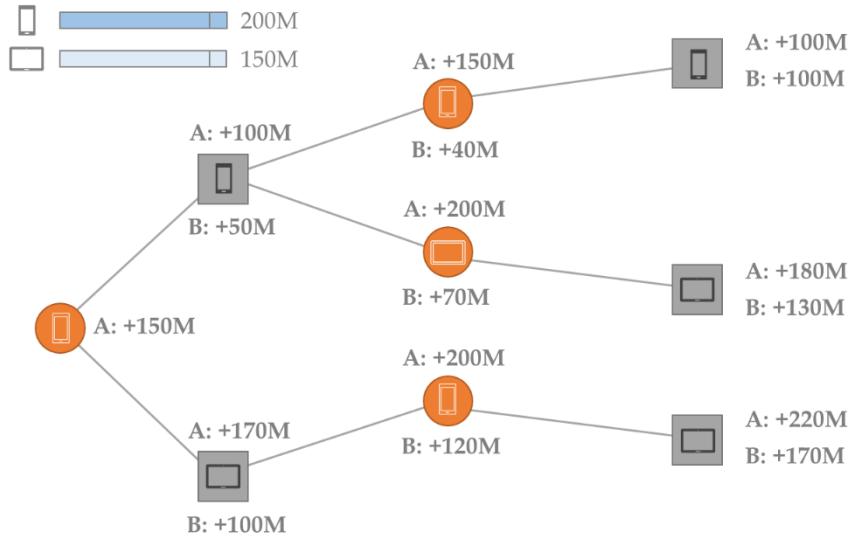


Рис. 15.5. Третья стратегия игры

Представим ситуацию, в которой после выпуска планшета компанией В, компания А решает потеснить конкурента и также вкладывается в планшеты. В ответ на это, игрок В переходит на рынок телефонов, что сказывается на выигрышах игроков: 200М у компании А, 150М – у компании В (рис. 15.6).

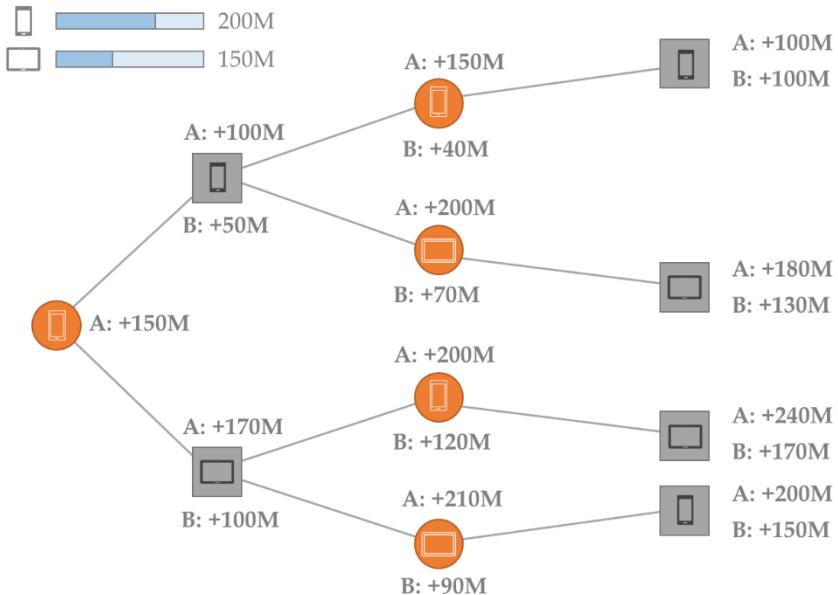


Рис. 15.6. Четвертый вариант стратегии

Схожим образом оставшиеся ветви достраиваются до терминальных точек. В итоге все возможные варианты развития событий предстают перед нами в виде дерева решений (рис. 15.7).

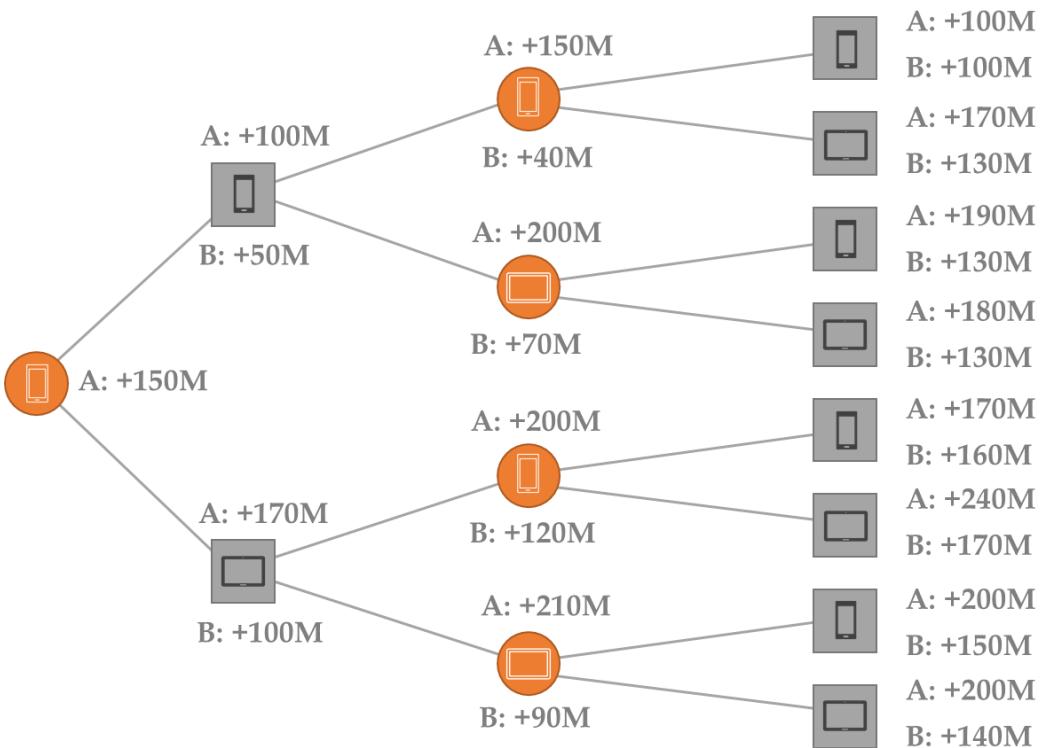


Рис. 15.7. Конечное дерево игры

Опираясь на полученное дерево игры, мы вычисляем наилучшую последовательность ходов из представленных, руководствуясь выбранной стратегией.

1-я стратегия – выбор худшего варианта для конкурента (классический антагонистический сценарий). Для этого суммируем выигрыши каждого из ходов игроков на каждом шагу и выбираем вариант с наименьшим выигрышем компании B.

2-я стратегия – сохранение наименьшего отставания от конкурента (случай, когда отстающий игрок стремится к минимизации разницы выигрышей на каждом ходу).

3-я стратегия – наиболее понятная и простая – выбор лучшего варианта для себя на каждом ходу.

4-я стратегия – стремление к взаимовыгодной ситуации, когда будут выбираться лучшие для обеих сторон ходы.

Варианты стратегий не заканчиваются на представленных выше – их может быть великое множество. Выбор будет зависеть от личных предпочтений игроков, от рыночной ситуации, от других внешних и внутренних факторов.

В силу того, что дерево в нашем конкретном случае достаточно маленькое, его можно охватить одним взглядом и аналитически прийти к тому или иному решению (применять упомянутые стратегии). Однако, если дерево более обширное, есть необходимость задуматься об универсальном варианте, или хотя бы о его наличии.

Алгоритм Куна

Нам повезло и такой алгоритм действительно существует – это алгоритм Куна, который позволяет выявить оптимальную стратегию путем сравнения исходов игры на каждой разветвке и отсечением непродуктивных веток.

Прямым ходом алгоритма Куна (двигаемся от корня к листьям) суммируем все выигрыши каждого игрока на каждой ветке и указываем их на листьях (мы видим связанную пару исходов игры для двух игроков) (рис. 15.8).

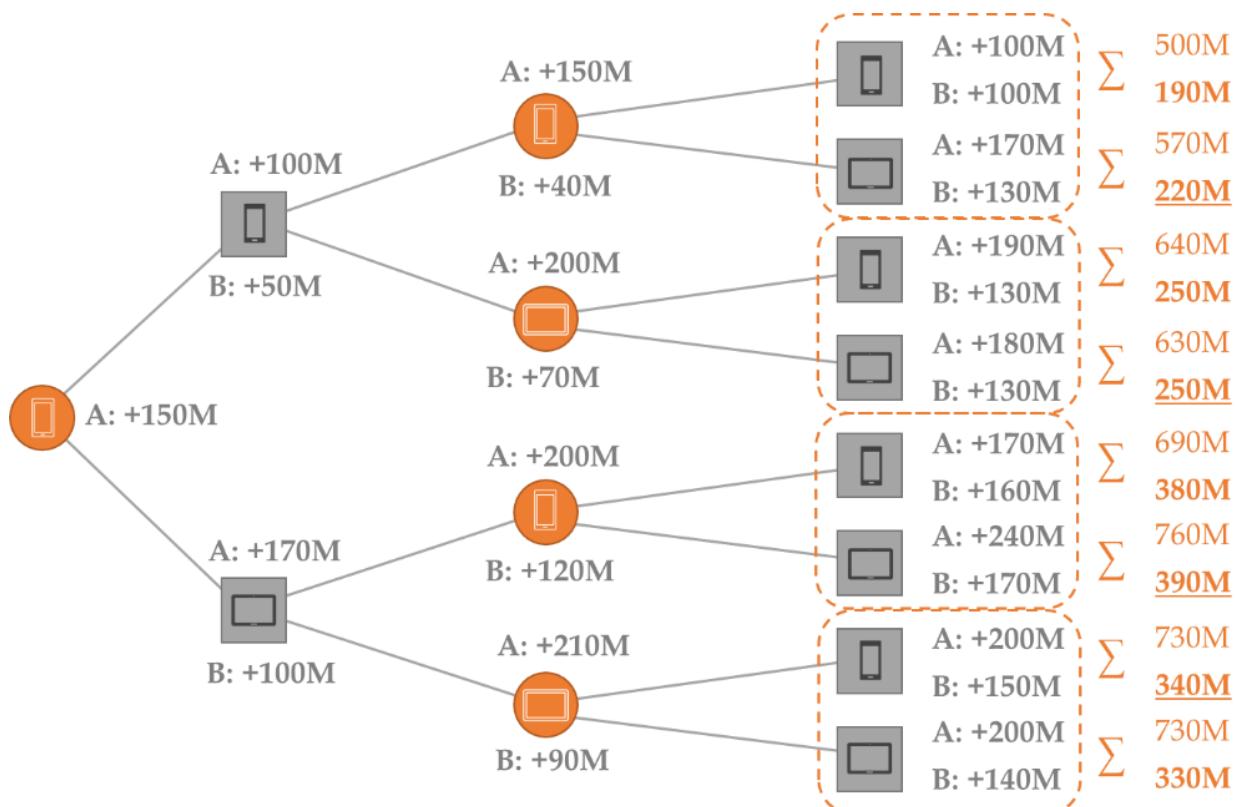


Рис. 15.8. Прямой ход алгоритма Куна

Обратным ходом алгоритма (направление движения – от листьев к корню) отсекаем наихудшие варианты на каждом уровне с позиции выбирающего игрока. При этом мы ориентируемся не на сиюминутный выигрыш, а на суммарный

исход. Если суммарные выигрыши на развилке равнозначны, то выбираем тот вариант, который будет наименее выгоден для соперника (рис. 15.9)

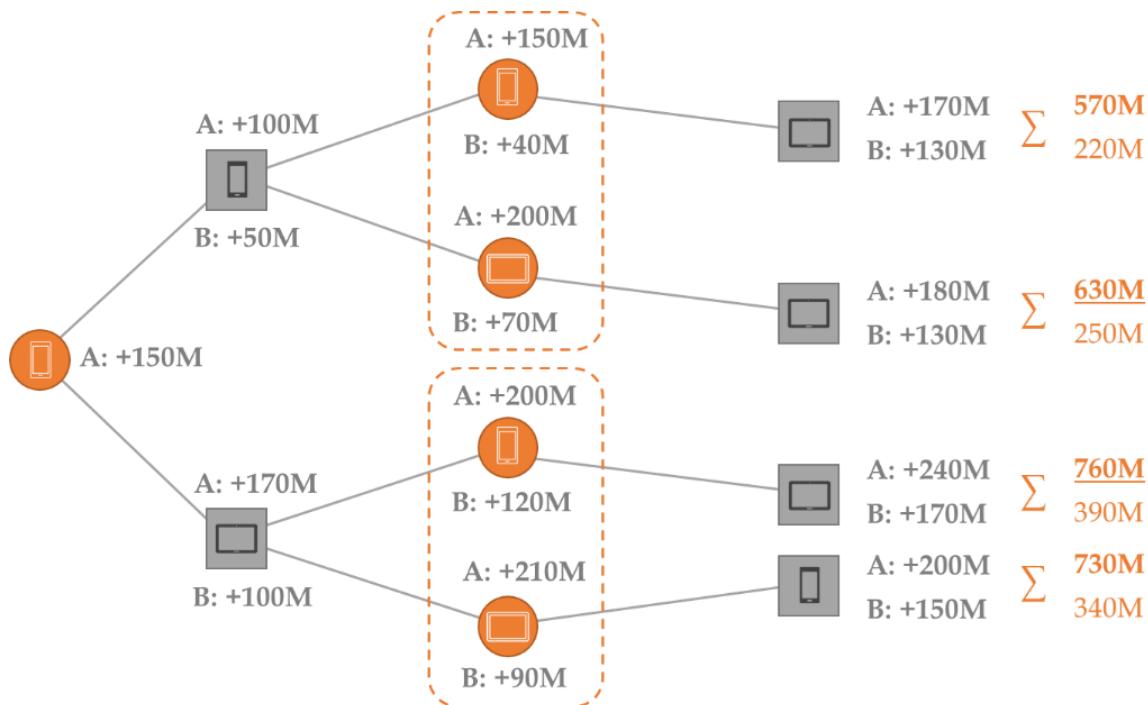


Рис. 15.9. Обратный ход алгоритма Куна

Чем выше к корню мы поднимаемся, тем больше веток мы отсекаем. По окончании работы алгоритма мы придём к корню и оставим единственную необходимую ветку стратегии, которая будет для нас оптимальной (рис. 15.10).

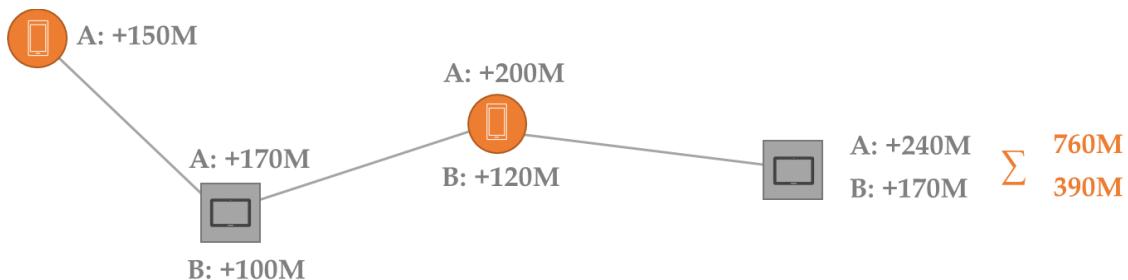
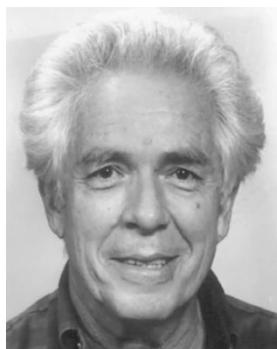


Рис. 15.10. Окончательный вариант после применения алгоритма Куна



Гарольд Уильям Кун (англ. Harold William Kuhn, 29.07.1925 – 02.07.2014, США) – заслуженный профессор математики в Принстонском университете, автор теоремы Куна, покера Куна, условия Куна–Таккера, специалист по теории игр. Лауреат премии фон Неймана за 1980 год совместно с Альбертом Таккером и Дэвидом Гейлом. Был другом и коллегой Джона Нэша. Оба плодотворно сотрудничали с Альбертом Таккером. Кун был соавтором книги «The Essential John Nash» и консультантом в фильме «Игры разума».

Сведение к матричной игре

Любую позиционную игру можно представить в матричной форме и работать с ней, используя уже знакомые инструменты. Платежная матрица строится как комбинация всех возможных сочетаний стратегий обоих игроков и финального выигрыша от применения этих стратегий. Конечные цены игры будут находиться в ячейках матрицы.

Таким образом, мы можем представить позиционную игру в виде двух антагонистических игр для игроков А и В, для каждой из которых с помощью принципов «максимина» и «минимакса» найти нижнюю и верхнюю цены игры (таблица 15.1 и таблица 15.2).

Таблица 15.1

Платежная матрица игрока А

A	■ ■	■ ■	■ ■	■ ■	мак- симин
■ ■	500	570	690	760	500
■ ■	640	630	730	730	630
мини макс	640	630	730	760	

Таблица 15.2

Платежная матрица игрока В

	■ ■	■ ■	■ ■	■ ■	мак- симин
■ ■	190	220	380	390	190
■ ■	250	250	340	330	250
мини макс	250	250	380	410	

В нашем случае, верхняя и нижняя цена антагонистических игр строится по ходам, которые соответствуют стратегии минимального отставания от

конкурента – задача игроков выбирать такие стратегии, при которых разница между выигрышем их конкурента минимальна.

Также можно рассмотреть биматричную игру (таблица 15.3) и найти ситуацию равновесия по Нэшу.

Таблица 15.3

Платежные матрицы игроков А и В

A	□ □	□ □□	□□ □	□□□ □
□ □	500	570	690	760
□ □□	640	630	730	730
B	□ □	□ □□	□□ □	□□□ □
□ □	190	220	380	390
□ □□	250	250	340	330

В рассматриваемом примере равновесие по Нэшу в биматричной форме игры строится по ходам, которые соответствуют стратегии выбора лучшего следующего хода для себя и для оппонента, что гарантировано приводит к более высоким результатам для обоих.

Стоит отметить, что обратный вариант «перевоплощения» игр также возможен – матричная игра может быть представлена в виде дерева и решена посредством алгоритма Куна или другим аналитическим способом.

16. Позиционные игры с «природой»

Как нам уже известно из раздела 14, играть можно не только с одушевлённым противником. Противостоять можно так называемой «природе» (стихии, рынку, природному явлению, потребительскому спросу, экономической ситуации и т.д.).

Предположим, что одна компания, занимающаяся космическими полетами, решает на чем ей сконцентрировать усилия – производить пилотируемые корабли для доставки в космос людей или беспилотные судна для доставки грузов.

Пилотируемые корабли сложны и сопряжены с рисками, но приносят большую прибыль, беспилотные – проще и дешевле, но не столь прибыльны. Компания имеет возможность продать корабль и свернуть программу, а также в случае успешного первого года купить второй корабль. Известны вероятности возникновения высокого и низкого спроса на каждую из предоставляемых услуги. Необходимо определить какую программу выбрать, чтобы через два года получить наибольшую прибыль.

Природой в данной ситуации является коммерческий спрос, который может быть или высоким, или низким. От спроса зависит выигрыш, другими словами, заработка компании. Корень дерева в данном случае не определен. Первый ход представляет собой две альтернативы – покупка грузового или пилотируемого корабля. Стоимость пилотируемого корабля составляет 550М, грузового корабля – 250М (рис. 16.1).

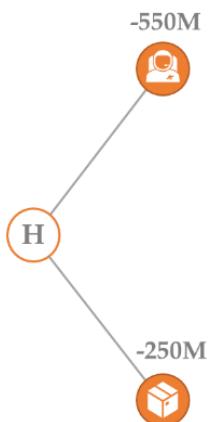


Рис. 16.1. Ходы компании в первый год

В первый год высокий спрос ожидается с вероятностью 0,6, низкий – с вероятностью 0,4 (рис. 16.2).

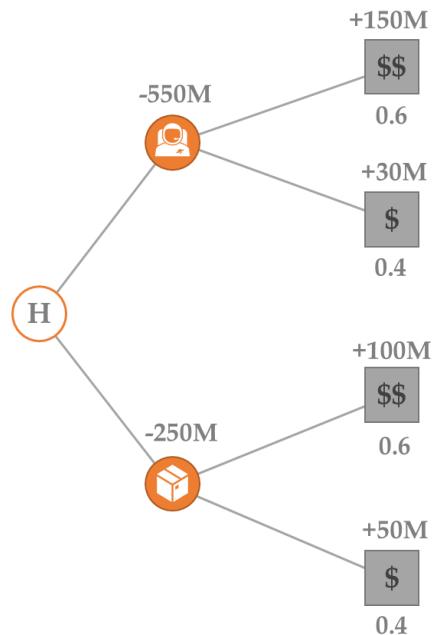


Рис. 16.2. Ходы компании в первый год

После этого мы можем продать построенные корабли и вернуть себе 500М и 150М за пилотируемый и грузовой варианты соответственно, или же купить ещё один корабль того или иного типа (рис. 16.3).

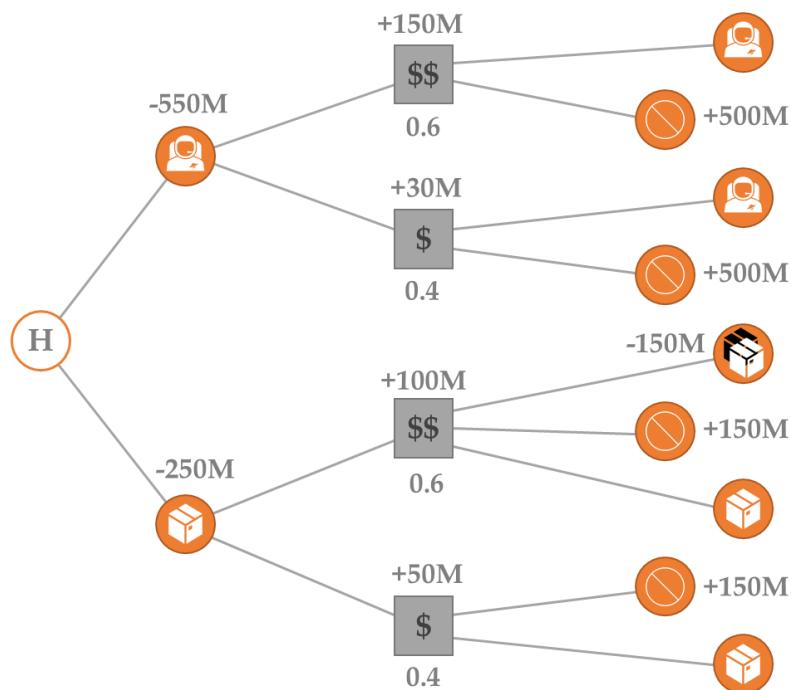


Рис. 16.3. Ходы компании во второй год

Ситуация со спросом напрямую зависит от результатов прошлого года: если спрос в первый год был высокий, то с вероятностью 0,8 он сохранится, а если был низкий – вероятность хорошего спроса на следующий год всего 0,4.

Таким образом мы построили дерево решений для позиционной игры с природой (рис. 16.4). Остается только выбрать наилучший вариант. В данном случае мы будем использовать немного другой алгоритм.

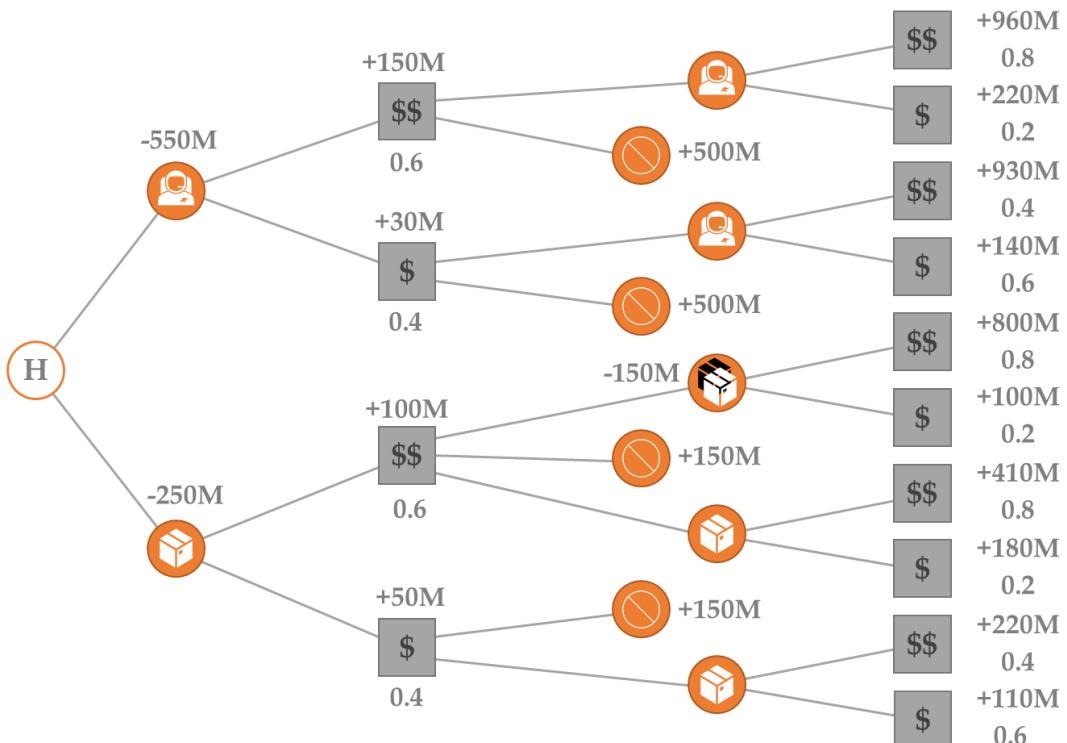


Рис. 16.4. Дерево игры с природой

Алгоритм поиска решения

В силу того, что во время ходов природы мы ничего не знаем наверняка и можем только предполагать (действовать на основе вероятности наступления того или иного состояния природы), мы будем работать со средневероятным значением выигрыша – будем «стягивать» ветви и находить средневзвешенное решение (рис. 16.5).

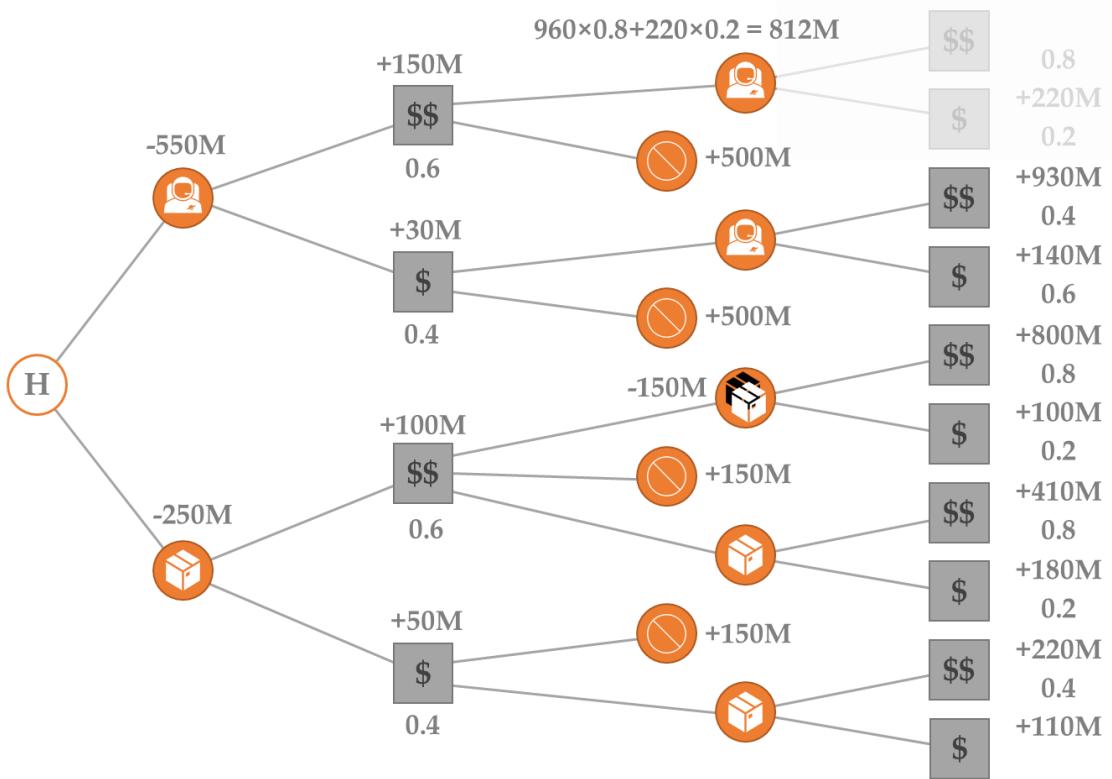


Рис. 16.5. Расчет средневзвешенного значения для второго года

Нам необходимо высчитать средневзвешенное значение выигрыша для каждой ветки, которую мы таким образом «сворачиваем» на один шаг (рис. 16.6).

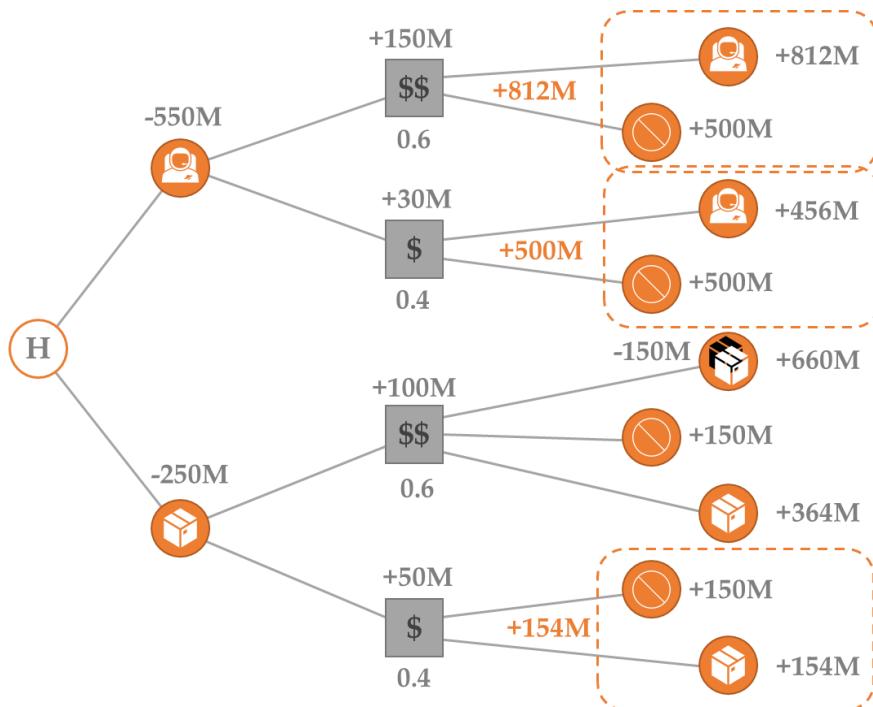


Рис. 16.6. Свертка веток

Далее выбираем лучший вариант в каждой паре стратегий и повторяем шаги до тех пор, пока не дойдем до корня (рис. 16.7). Средневзвешенным значением мы руководствуемся только когда действует природа, ведь ходах игрока мы действуем осмысленно и можем выбрать лучшую альтернативу.

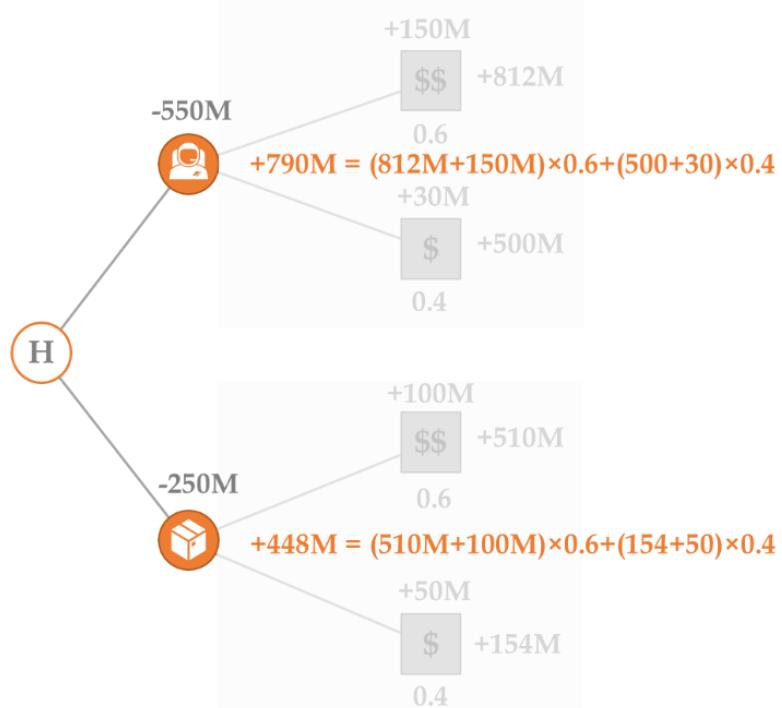


Рис. 16.7. Расчет средневзвешенных значений

После того, как мы полностью свернули наше дерево, получилось, что pilotируемая программа обеспечит нам чистую прибыль в 240M, а грузовая – только 198M (рис. 16.8).

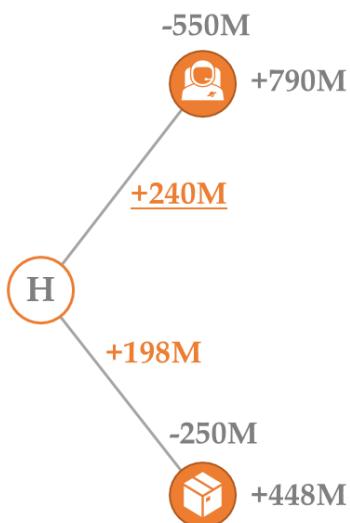


Рис. 16.8. Результат свертки дерева

Таким образом, оптимальным вариантом будет изначально купить пилотируемый корабль, чтобы в дальнейшем купить ещё один (если будет высокий спрос) или продать имеющийся (если спрос будет низкий) (рис. 16.9).

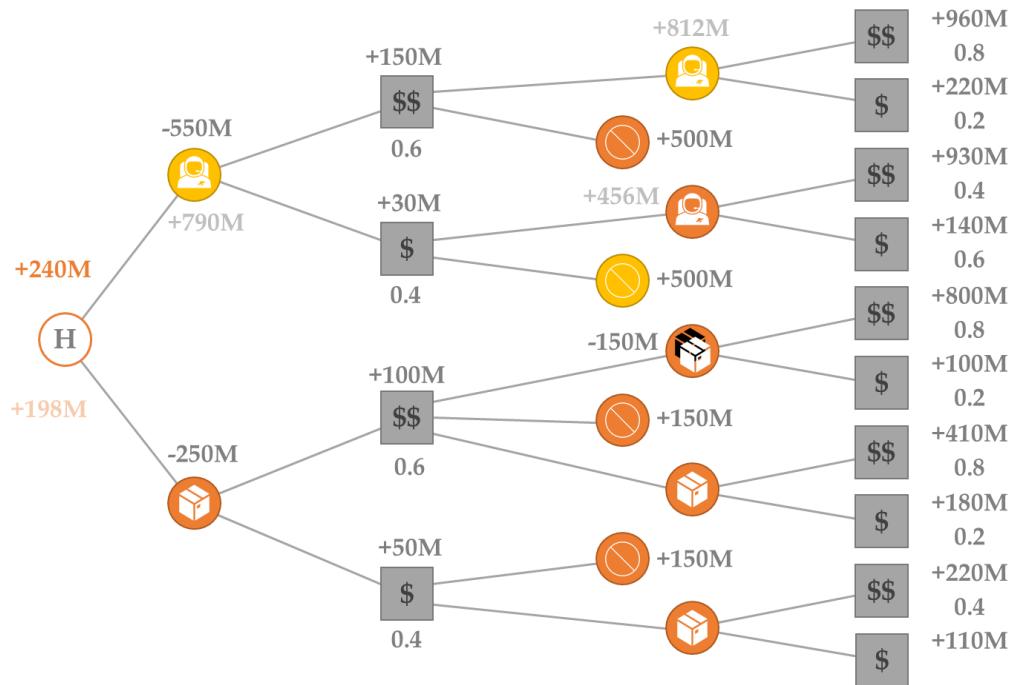


Рис. 16.9. Вариант стратегии компании с наибольшей прибылью

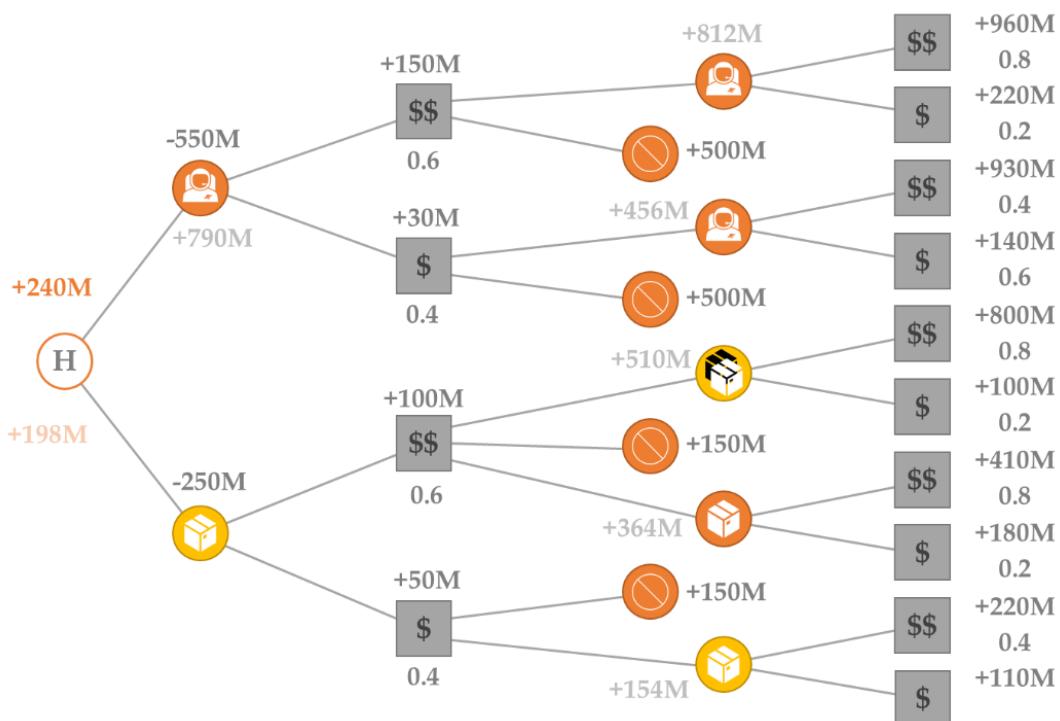


Рис. 16.10. Варианты стратегии компании с минимальным риском

Конечно же не стоит забывать о горизонте планирования, который также может внести свои корректизы. Другими словами, возможно компромиссное решение. Выбор грузового корабля принесет меньшую прибыль, но позволит продолжать проект при любом спросе с приемлемой выручкой (см. рис. 16.10).



Альберт Уильям Таккер (Albert William Tucker; 28.11.1905, Канада — 25.01.1995 года, США) — глава математического факультета Принстонского университета с 1933 г. по 1974 г. (рис. 17.23). Его аспиранты — Джон Нэш, Ллойд Шепли и Марвин Мински, ученики — Гарольд Кун и Джон Кемени. В 1950 году дал название и формулировку «Дilemma заключенного». Соавтор условия Куна-Такера для нелинейного программирования. В 1980-х годах записал знаменитую серию устных историй о Принстонском математическом обществе 1930-х годов.

Сведение к матричной форме

Позиционную игру с природой также можно свести к матричному виду (таблица 16.1). В строках платежной матрицы будут все комбинации действий игрока, а в столбцах — комбинации состояний природы за 2 года. В силу того, что ходы могут привести к совершенно различным исходам, мы можем получить отрицательные значения выигрышей. Далее, применяя ранее изученные критерии матричных игр, можно вычислить оптимальное решение. Например, по критерию Байеса наилучшим вариантом будет покупка двух грузовых кораблей.

Таблица 16.1

Критерий Байеса

A	\$\$ \$\$	\$\$ \$	\$ \$\$	\$ \$	Байеса
👤👤	560	-180	410	-380	222
👤🚫	100	100	-20	-20	52
📦📦	500	-200	0	0	216
📦📦	260	30	20	-90	110
📦🚫	0	0	-50	-50	-20
$p =$	0.48	0.12	0.16	0.24	

По критерию Гермейера (таблица 16.2) гарантированным вариантом минимального проигрыша будет вариант покупки пилотируемого корабля и продажи его во второй год. Этот вариант обеспечит минимальные потери в неблагоприятной ситуации.

Таблица 16.2

Критерий Гермейера

A	\$\$ \$\$	\$\$ \$	\$ \$\$	\$ \$	Гермейера
👤👤	268,8	-21,6	65,6	-91,2	-91,2
👤🚫	48	12	-3,2	-4,8	-4,8
📦🚀	240	-24	0	0	-24
📦📦	124,8	3,6	3,2	-21,6	-21,6
📦🚫	0	0	-8	-12	-12

Так как при построении матрицы Гермейера мы избавились от вероятностей наступления состояний природы, мы можем также применить на ней метод Гурвица (таблица 16.3 и рис. 16.11), то есть посчитать наилучшие решения в зависимости от склонности игрока к риску.

Таблица 16.3

Метод Гурвица

Риск, α	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
👤👤	-91,2	-55,2	-19,2	16,8	52,8	88,8	124,8	160,8	196,8	232,8	268,8
👤🚫	-4,8	0,48	5,76	11,04	16,32	21,6	26,88	32,16	37,44	42,72	48
📦🚀	-24	2,4	28,8	55,2	81,6	108	134,4	160,8	187,2	213,6	240
📦📦	-21,6	-6,96	7,68	22,32	36,96	51,6	66,24	80,88	95,52	110,16	124,8
📦🚫	-12	-10,8	-9,6	-8,4	-7,2	-6	-4,8	-3,6	-2,4	-1,2	0

Получаем, что, покупатель, не склонный к риску, должен выбирать покупку грузового корабля в первый год и во второй год. Если же он желает рискнуть и заработать, то ему необходимо купить два пилотируемых корабля.

ЧАСТЬ IV. ДОПОЛНИТЕЛЬНЫЕ ГЛАВЫ

17. Регрессионное прогнозирование

С точки зрения теории принятия решений, прогнозирование – принятие решений в условии отложенных результатов, то есть необходимость принятия решения в настоящий момент времени, результат которого будет получен в определенном будущем. Например, принятие решений о покупке акций компаний или вложений в инвестиционные проекты. Чтобы вычислить результат подобных действий, применяются определенные методы прогнозирования.

Регрессия – самый простой метод прогнозирования, основой которого является метод наименьших квадратов.

Рассмотрим простейший пример.

Имеется статистика по спросу и цене подарочных товаров в период новогодних праздников. Необходимо построить зависимость между спросом и ценой. Исходные данные приведены в таблице 17.1.

Таблица 17.1

Объем и цена продаж подарков

	Дата	Спрос (x)	Цена (y)
1	25.дек	400 шт.	850,00 ₽
2	26.дек	500 шт.	900,00 ₽
3	27.дек	600 шт.	1 000,00 ₽
4	28.дек	700 шт.	1 100,00 ₽
5	29.дек	800 шт.	1 200,00 ₽
6	30.дек	900 шт.	1 500,00 ₽
7	31.дек	1000 шт.	1 900,00 ₽
8	01.янв	150 шт.	1 500,00 ₽
9	02.янв	50 шт.	900,00 ₽
10	03.янв	100 шт.	750,00 ₽
11	04.янв	200 шт.	800,00 ₽
12	05.янв	300 шт.	900,00 ₽
13	06.янв	550 шт.	1 300,00 ₽
14	07.янв	200 шт.	1 100,00 ₽
15	08.янв	950 шт.	600,00 ₽

Представим все точки таблицы на координатной плоскости, как показано на рис. 17.1. Рассмотрим, что этот ряд представляет из себя – разрозненное множество точек, объединенное общей тенденцией. Так, при увеличении спроса

цена возрастает экспоненциально. Несмотря на общий тренд графика, существуют точки, не поддающиеся общему описанию функции, – выбросы. Для управления такими данными применяется регрессия.



Рис. 17.1. Зависимость цены от объема продаж

Регрессия (от латинского *regressio* — обратное движение, отход) в теории вероятностей и математической статистике — односторонняя стохастическая зависимость, устанавливающая соответствие между случайными переменными. Это математическое выражение, отражающее связь между переменными y и x при наличии между ними статистической зависимости.

Регрессия строится обратным образом – от известных точек к функции, определяет, какая функция описывает все точки на графике. Таким образом, получив функцию, мы можем определить ее поведение на отдельном участке. Изображение подобной функции на рис. 17.2.

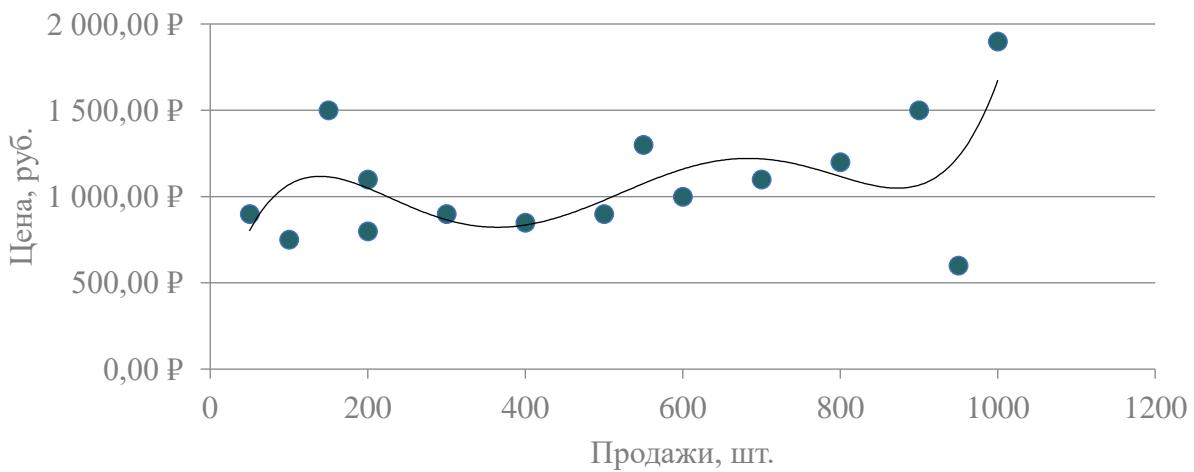


Рис. 17.2. Регрессионная функция, описывающая зависимость цен от продаж

Функция, проходящая через каждую из точек, не является точной и верно описывающей ситуацию на графике. Кривая, отражающая тенденцию поведения всей статистики в целом, будет являться лучшим решением. Для того, чтобы найти такую кривую, необходимо использовать следующие методы.

Линейная регрессия

Линейная регрессия – функция некоторой прямой, которая отражает характер поведения всей статистики. Она имеет следующий вид:

$$y_k = (ax_k + b) + e_k$$

где $(ax_k + b)$ – истинная зависимость (тренд), e_k – случайные колебания (шум). Из всех точек, расположенных на графике, для точного описания прямой подходят только те, не выбивающиеся из тренда точки. Изобразим полученную функцию y_k на рис. 17.3.

Благодаря нахождению коэффициентов a и b линейной регрессии $y = ax + b$ становится возможным найти наиболее адекватно отражающие зависимость переменные x_k и y_k . Коэффициенты находятся методом наименьших квадратов.

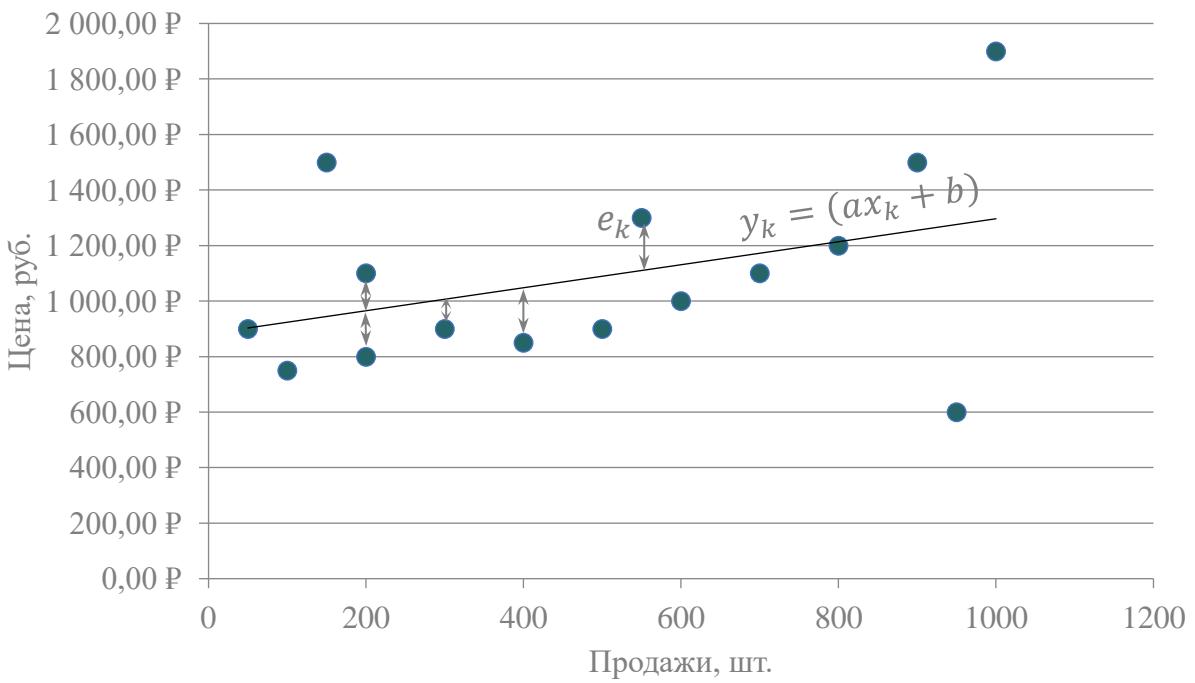


Рис. 17.3. График линейной регрессии

Метод наименьших квадратов

Для каждой точки, описываемой x_k и y_k , применяется следующая функция – наименьшее расстояние до каждой точки:

$$(y_k - ax_k - b)^2 \rightarrow \min$$

Для нахождения значений a и b запишем функцию $f(a, b)$:

$$f(a, b) = \sum_{k=1}^n (y_k - ax_k - b)^2$$

Найдем частные производные $f(a, b)$ по a и по b :

$$\frac{\partial f(a, b)}{\partial a} = \sum_{k=1}^n 2(y_k - ax_k - b)(-x_k)$$

$$\frac{\partial f(a, b)}{\partial b} = \sum_{k=1}^n 2(y_k - ax_k - b)(-1)$$

Раскроем скобки внутри суммы и вынесем множители за скобки:

$$\frac{\partial f(a, b)}{\partial a} = -2 \left(\sum_{k=1}^n y_k x_k - a \sum_{k=1}^n x_k^2 - b \sum_{k=1}^n x_k \right)$$

$$\frac{\partial f(a, b)}{\partial b} = -2 \left(\sum_{k=1}^n y_k - a \sum_{k=1}^n x_k - bn \right)$$

Приравняем частные производные к 0:

$$\sum_{k=1}^n y_k x_k - a \sum_{k=1}^n x_k^2 - b \sum_{k=1}^n x_k = 0$$

$$\sum_{k=1}^n y_k - a \sum_{k=1}^n x_k - bn = 0$$

Запишем среднее арифметическое x_k и y_k , а также x_k^2 и $y_k x_k$ в виде:

$$\bar{x} = \frac{1}{n} \sum_{k=1}^n x_k; \quad \bar{y} = \frac{1}{n} \sum_{k=1}^n y_k; \quad \bar{x}^2 = \frac{1}{n} \sum_{k=1}^n x_k^2; \quad \bar{yx} = \frac{1}{n} \sum_{k=1}^n y_k x_k$$

Подставляя средние арифметические в уравнения, умножаем на n и получаем:

$$\bar{yx} - a\bar{x}^2 - b\bar{x} = 0$$

$$\bar{y} - a\bar{x} - b = 0$$

Решая систему линейных уравнений относительно a и b , получаем ответ:

$$a^* = \frac{\bar{yx} - \bar{y} \cdot \bar{x}}{\bar{x}^2 - \bar{x}^2}, \quad b^* = \bar{y} - a^* \bar{x}$$

Таким образом, искомая регрессия принимает вид

$$y = a^* x + b^*$$

где коэффициенты вычисляются как:

$$a^* = \frac{\bar{yx} - \bar{y} \cdot \bar{x}}{\bar{x}^2 - \bar{x}^2}, \quad b^* = \bar{y} - a^* \bar{x}$$

График регрессии с новыми коэффициентами изображен на рис. 17.4.

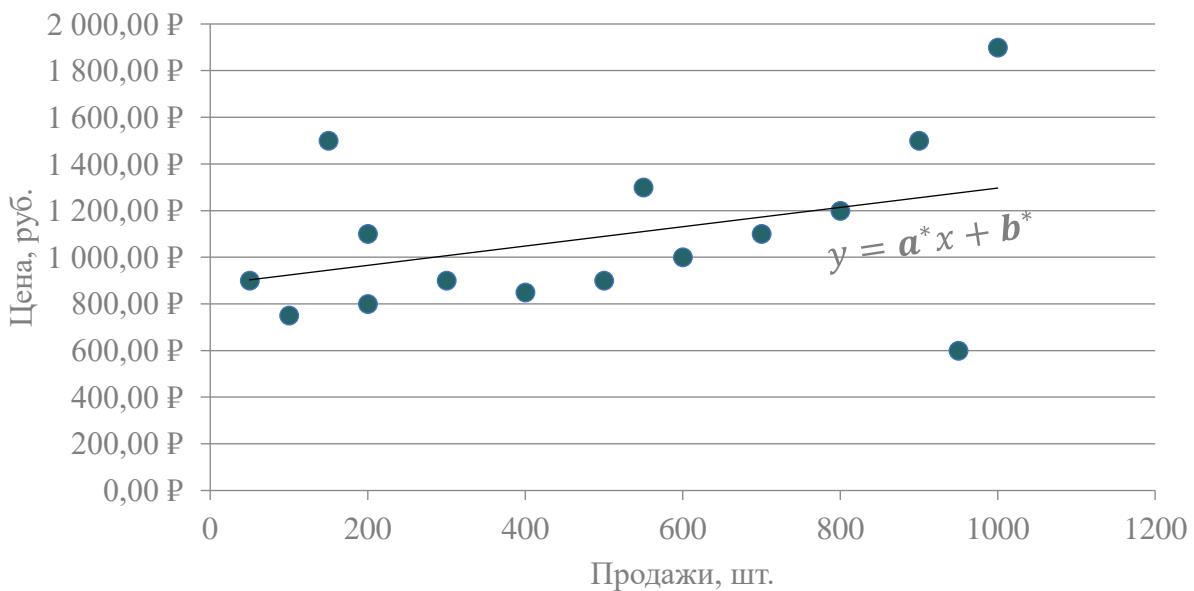


Рис. 17.4. График линейной регрессии

Приведем задачу о новогодних подарках к виду функции линейной регрессии. Уравнение прямой принимает следующий вид:

$$y = a^*x + b^*$$

которое минимизирует квадраты отклонений $(\hat{y} - y)^2$, где \hat{y} – восстановленное значение y , полученное в результате вычисления функции регрессии. Коэффициенты примут следующий вид:

$$a^* = \frac{\bar{yx} - \bar{y}\bar{x}}{\bar{x}^2 - \bar{x}^2} = 0,41$$

$$b^* = \bar{y} - a^*\bar{x} = 882$$

Уравнение линейной регрессии:

$$y = 0,41x + 882$$

Все рассчитанные данные занесём в таблицу 17.2.

Основное свойство линейной регрессии $y = a^*x + b^*$ – сумма всех расстояний от точек (x_k, y_k) до прямой $y = a^*x + b^*$ равна 0:

$$\sum_{k=1}^n (y_k - a^*x_k - b) = 0$$

Отклонение значений восстановленных \hat{y} , изображенных на рис. 17.5, сравнивается с изначальным значением y . Таким образом вычисляется дисперсия набора данных, то есть ее разброс, расстояние между \hat{y} и y .

Таблица 17.2

Исходные данные и значение функции регрессии

<i>k</i>	Дата	Спрос (<i>x</i>)	Цена (<i>y</i>)	Восст. \hat{y}	$\hat{y} - y$
1	25.дек	400 шт.	850,00 ₽	1 048,00 ₽	-198,00 ₽
2	26.дек	500 шт.	900,00 ₽	1 089,43 ₽	-189,43 ₽
3	27.дек	600 шт.	1 000,00 ₽	1 130,85 ₽	-130,85 ₽
4	28.дек	700 шт.	1 100,00 ₽	1 172,28 ₽	-72,28 ₽
5	29.дек	800 шт.	1 200,00 ₽	1 213,70 ₽	-13,70 ₽
6	30.дек	900 шт.	1 500,00 ₽	1 255,13 ₽	244,87 ₽
7	31.дек	1000 шт.	1 900,00 ₽	1 296,55 ₽	603,45 ₽
8	01.янв	150 шт.	1 500,00 ₽	944,44 ₽	555,56 ₽
9	02.янв	50 шт.	900,00 ₽	903,02 ₽	-3,02 ₽
10	03.янв	100 шт.	750,00 ₽	923,73 ₽	-173,73 ₽
11	04.янв	200 шт.	800,00 ₽	965,15 ₽	-165,15 ₽
12	05.янв	300 шт.	900,00 ₽	1 006,58 ₽	-106,58 ₽
13	06.янв	550 шт.	1 300,00 ₽	1 110,14 ₽	189,86 ₽
14	07.янв	200 шт.	1 100,00 ₽	965,15 ₽	134,85 ₽
15	08.янв	950 шт.	600,00 ₽	1 275,84 ₽	-675,84 ₽
		$\bar{x} = 493$ шт.	$\bar{y} = 1087$ ₽		$\Sigma = 0,00$ ₽

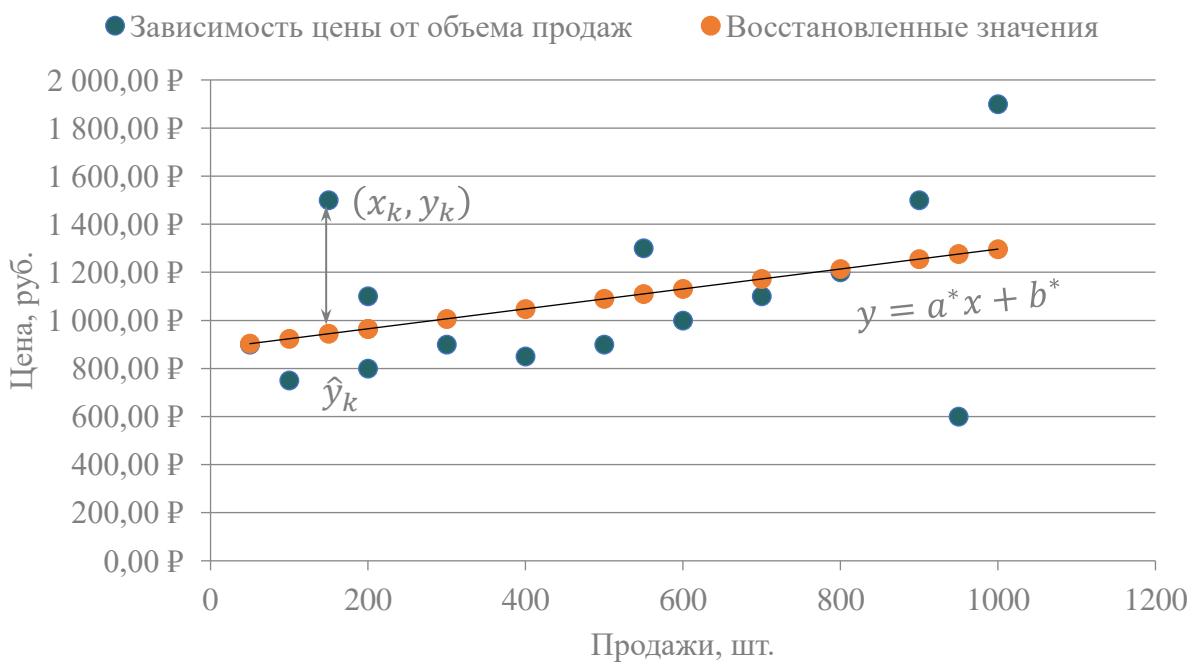


Рис. 17.5. Изображение восстановленных значений на прямой регрессии

Дисперсия определяется формулой

$$\sigma^2 = \frac{1}{n} \sum_{k=1}^n (\hat{y} - y)^2$$

Применение дисперсии рассмотрим позже.

Фильтрация

Прежде чем проводить прогнозирование, временной ряд необходимо подготовить. Существует множество методов подготовки данных: дифференцирование, интегрирование, устранение различных факторов и т. п. Рассмотрим однозначно необходимый и самый простой метод – фильтрацию.

Фильтрация – это исключение выбросов, значений, сильно выпадающих из общего тренда, не описывающих характер поведения функции. Такие значения обнаруживаются по трем признакам: неадекватные по расчету, неадекватные по смыслу и случайные возмущения (рис. 17.6).

Неадекватные по расчету значения – такие значения, отклонение которых превышает максимальное допустимое. Максимальное допустимое значение отклонения тренда рассчитывается по следующей формуле:

$$|\hat{y} - y| \leq 1,5\sigma$$

Фильтрация неадекватных по расчету значений является наиболее популярным методом очистки временного ряда.

Случайные возмущения – ошибки, актуальные для механических измерений и сбоев технического оборудования. Такие ошибки редко наблюдаются в данных, описывающих рыночную или любую другую экономическую ситуацию.

Неадекватные по смыслу значения – точки, не вписывающиеся логически. Исключение таких точек возможно экспертами и опытными аналитиками в именно в той сфере, для которой проводится прогнозирование.

<i>k</i>	Дата	Спрос (x)	Цена (y)	Восст. \hat{y}	$ \hat{y} - y $	
1	25.дек	400 шт.	850,00 ₽	1 048,00 ₽	198,00	
2	26.дек	500 шт.	900,00 ₽	1 089,43 ₽	189,43	
3	27.дек	600 шт.	1 000,00 ₽	1 130,85 ₽	130,85	
4	28.дек	700 шт.	1 100,00 ₽	1 172,28 ₽	72,28	
5	29.дек	800 шт.	1 200,00 ₽	1 213,70 ₽	13,70	
6	30.дек	900 шт.	1 500,00 ₽	1 255,13 ₽	244,87	
7	31.дек	1000 шт.	1 900,00 ₽	1 296,55 ₽	603,45	
8	01.янв	150 шт.	1 500,00 ₽	944,44 ₽	555,56	
9	02.янв	50 шт.	900,00 ₽	903,02 ₽	3,02	
10	03.янв	100 шт.	750,00 ₽	923,73 ₽	173,73	
11	04.янв	200 шт.	800,00 ₽	965,15 ₽	165,15	
12	05.янв	300 шт.	900,00 ₽	1 006,58 ₽	106,58	
13	06.янв	550 шт.	1 300,00 ₽	1 110,14 ₽	189,86	
14	07.янв	200 шт.	1 100,00 ₽	965,15 ₽	134,85	
15	08.янв	950 шт.	600,00 ₽	1 275,84 ₽	675,84	

Праздничный день
Спрос высокий
Цена неадекватно высокая

Послепраздничный день
Спрос низкий
Цена неадекватно высокая

Последний день
Распродажа
Спрос высокий
Цена неадекватно низкая

Рис. 17.6. Выявление и устранения выбросов, неадекватных по значению

После исключения выбросов значение уравнения регрессии изменяется.

Исходный тренд (рис. 17.7):

$$y = 0,41x + 882$$

Отфильтрованный тренд (рис. 17.7):

$$y = 0,61x + 755$$

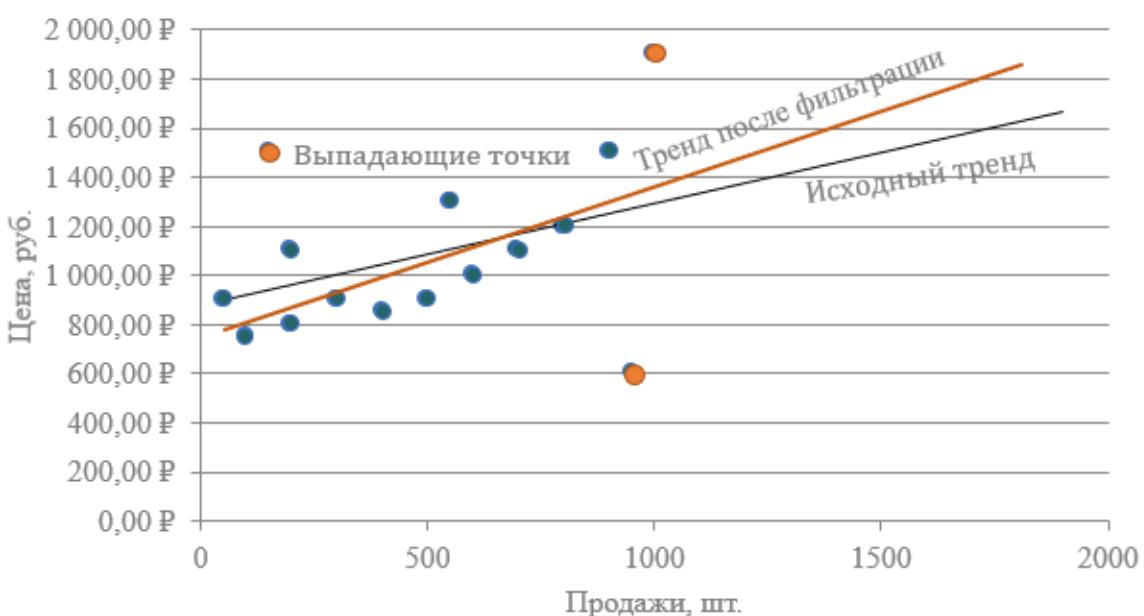


Рис. 17.7 Сравнение тренда после фильтрации и исходного тренда

Новая функция описывает временной ряд адекватнее: после фильтрации уточнились как угловое, так и пространственное положение функции.

Доверительные интервалы

Вернемся к вопросу дисперсии. Итак, дисперсия описывает разброс данных в выборке. Если дисперсия вокруг тренда невелика, то результатам прогнозирования можно доверять. Если дисперсия велика – то вероятность того, что прогноз сбудется, мала.

Стоит отметить, что прогноз сам из себя представляет поиск примерного диапазона возможных значений, а не одно единственного значения. Таким образом, ошибка прогноза вычисляется из прогнозируемого диапазона и находится как квадратный корень из дисперсии (рис. 17.8):

$$\sigma = \sqrt{D}$$

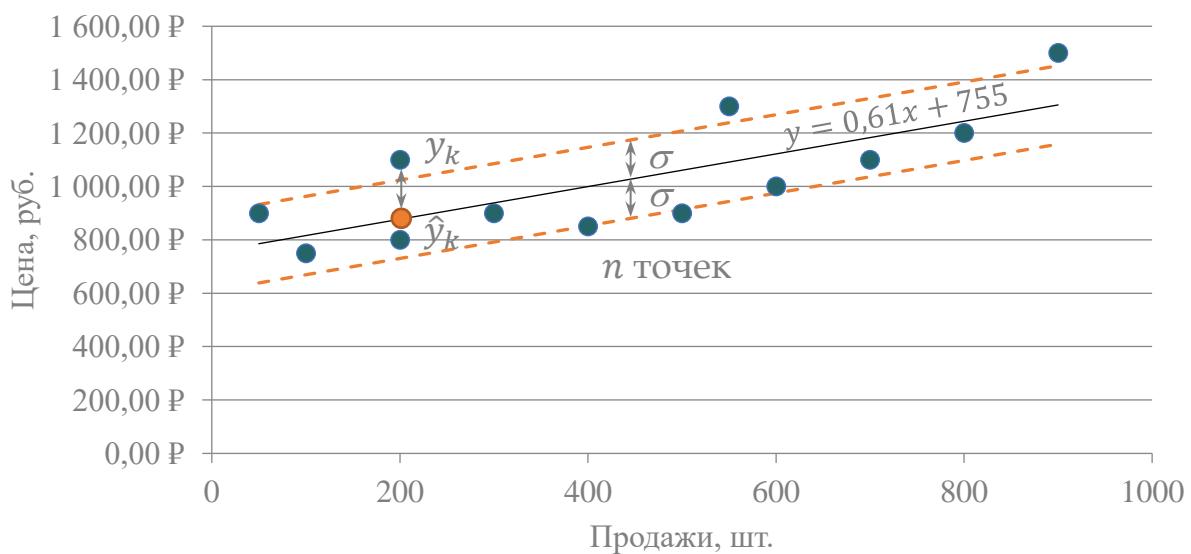


Рис. 17.8. Графическое изображение дисперсии регрессии

Прогнозируемое значение, лежащее в диапазоне, будет вероятно колебаться по определенному закону распределения. Самым частовстречающимся в природе распределением является нормальное распределение Гаусса, изображенное на рис. 17.9. Плотность вероятности зависит от среднеквадратичного отклонения следующим образом:

1. В доверительный интервал $(+\sigma; -\sigma)$ попадает ~68% всех точек.

2. В доверительный интервал $(+2\sigma; -2\sigma)$ попадает **~95%** всех точек.
3. В доверительный интервал $(+3\sigma; -3\sigma)$ попадает **~99%** всех точек.

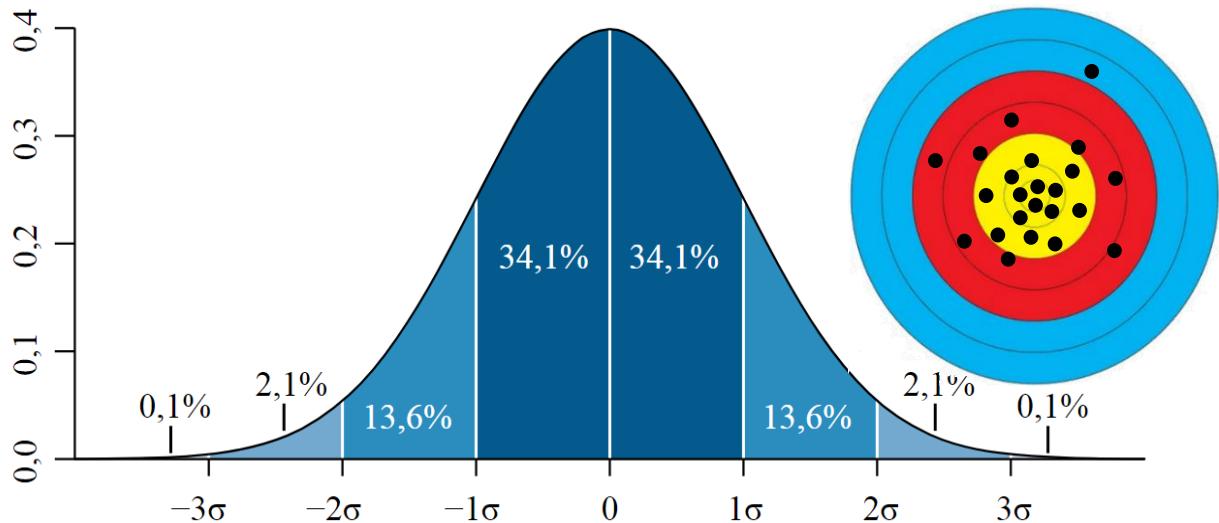


Рис. 17.9. График нормального распределения

В данной задаче доверительные интервалы можно изобразить как на рис. 17.10.

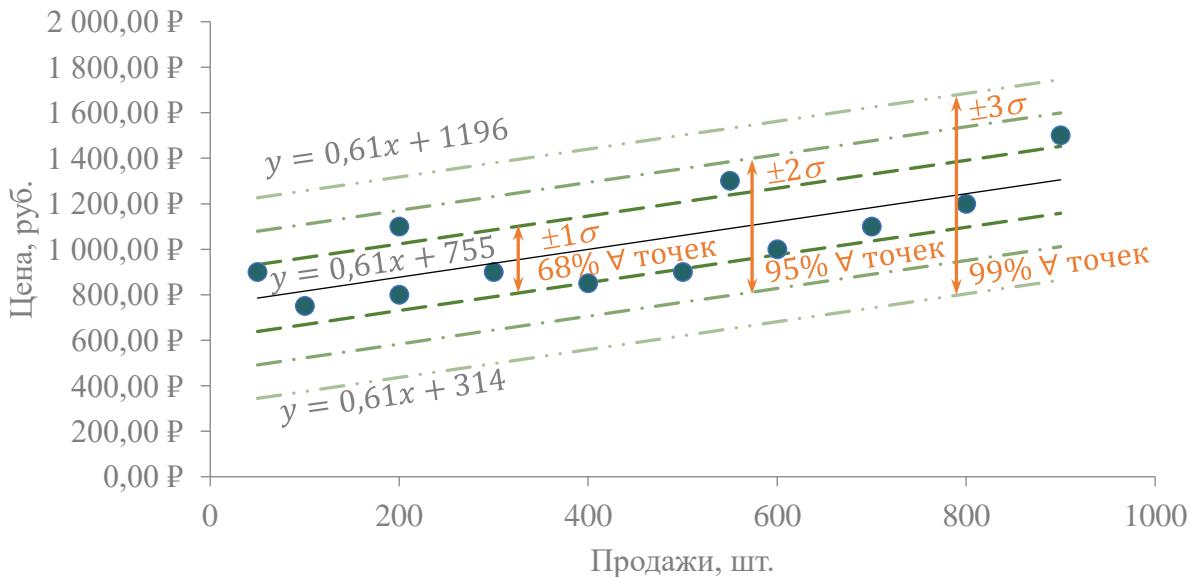


Рис. 17.10. Доверительные интервалы регрессии

Уравнение границ доверительного интервала ($i = \overline{1,3}$):

$$y^{\text{верх}} = a^*x + b^* + i\sigma$$

$$y_{\text{ниж}} = a^*x + b^* - i\sigma$$

Прогнозирование

Итак, функция регрессии задает бесконечную прямую, которая продолжается и вне диапазона заданных значений. Значит, этот тренд можно продлить – экстраполировать на более широкий диапазон значений.

Экстраполяция (от лат. *extrā* - вне, снаружи, и лат. *polire* - приглаживать, выправлять) — особый тип аппроксимации, при котором функция восстанавливается вне границ заданного интервала. С помощью экстраполяции возможно прогнозировать значения для будущего диапазона.

Рассчитав дисперсию, можно сделать вывод, что при уровне продаж в $x = 1600$ шт. цена продукции будет находиться с вероятностью $p = 68\%$ в интервале от 1586 до 1733 рублей, с вероятностью 95% - в интервале от 1440 до 2027 рублей, с вероятностью 99% - в интервале от 1292 до 2074 рублей. Таким образом, чем вероятнее прогноз, тем шире спектр возможных значений. Дисперсии изображены на рис. 17.11.

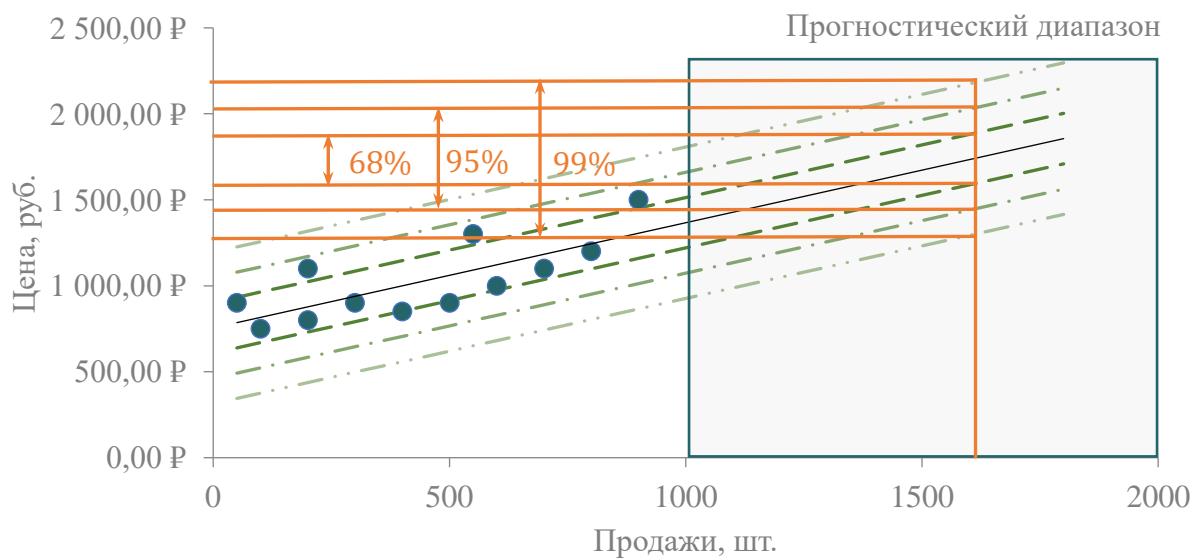


Рис. 17.11. Доверительные интервалы регрессии в прогностическом диапазоне

Так, точный прогноз содержит в себе высокий риск, что он не сбудется, а широкий прогноз содержит в себе минимальные риски.

Полиномиальная регрессия

Рассматривая предыдущую задачу, заметим, что тренд имеет тенденцию изгибаться, то есть ведет себя нелинейно. Для более точного описания поведения тренда вычисляют полиномы второго и третьего порядков, что имеют формы параболы и параболы с двумя перегибами. Чем выше степень полинома, тем подробнее он повторяет поведение значений на графике. На рис. 17.12 изображены регрессии в виде полиномов высшего порядка.

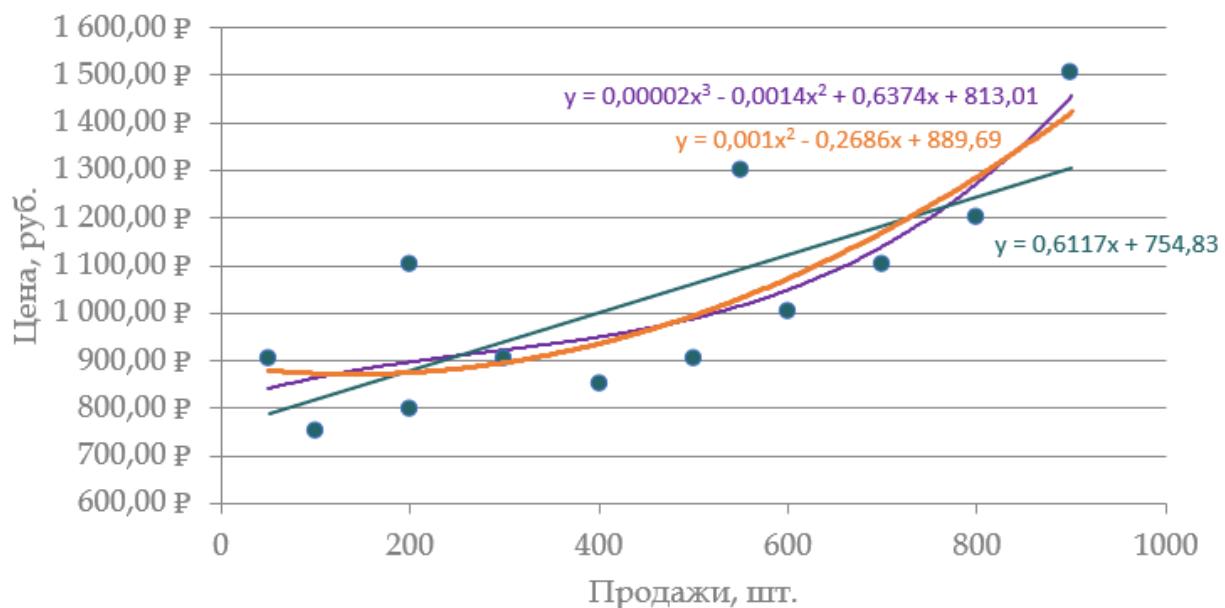


Рис. 17.12. Графики полиномиальной регрессии для полиномов высшего порядка

Вычислим среднеквадратические отклонения для каждой функции:

$$y = ax + b \quad \sigma = 146 \text{ руб.}$$

$$y = ax^2 + bx + c \quad \sigma = 120 \text{ руб.}$$

$$y = ax^3 + bx^2 + cx + d \quad \sigma = 105 \text{ руб.}$$

На практике при прогнозировании полиномы степенью выше второй не применяются, так как их поведение на прогностическом диапазоне неадекватно. Рассмотрим полиномы высших степеней на графиках рис. 17.13.

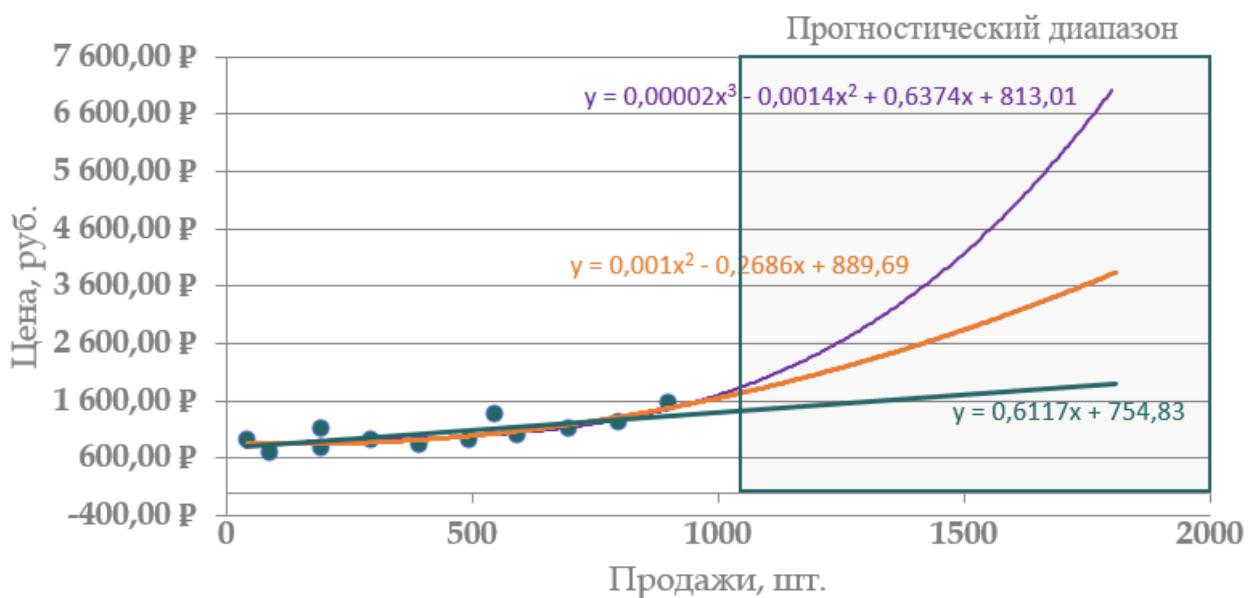


Рис. 17.13. Полиномиальная регрессия на прогностическом диапазоне

Чем выше степень полинома, тем короче срок прогноза, на который его можно применять, чтобы получить адекватный результат. Соответственно, линейная регрессия применяется для долгосрочного прогнозирования, а полином второй степени – для краткосрочного.

Существует более универсальный, общий вид регрессии. Любую регрессию можно составить из нескольких компонентов: линейной составляющей, нелинейной составляющей и сезонных колебаний, а также другие функции, которые описывают состояние конкретной области, в которой строится прогноз (рис.17.14).

$$y(x) = \underbrace{a_0 + a_1x + a_2x^2 + a_3x^3 + \dots + a_nx^n}_{\text{Полиномиальная составляющая}} +$$

$$+ \underbrace{b_1 \sin b_2x + b_3 \cos b_4x}_{\text{Сезонные колебания}} + \underbrace{c_1 e^{c_2x}}_{\text{Индекс инфляции}} + \underbrace{\sqrt{d_1x^2 + d_2x + d_3} + \frac{1}{i_1x^2 + i_2x + i_3}}_{\text{Различные функции}}$$

Рис. 17.14. Состав регрессии

Решение такой регрессии аналитически осуществлялось приведением ее к линейному виду, заменяя переменные.

Множественная регрессия

Множественная регрессия – построение зависимости одного параметра от множества различных параметров.

Рассматривая задачу, спрос на подарки зависит не только от цены, но и от даты. Например, 26 декабря и 6 января спрос примерно равен, но цены значительно отличаются. Значит, действительно, на спрос влияет не один фактор, а несколько, приведём их в таблице 17.3.

Таблица 17.3

Исходные данные по трем переменным

<i>k</i>	Дата (<i>t</i>)	Спрос (<i>x</i>)	Цена (<i>y</i>)
1	25.дек	400 шт.	850,00 ₽
2	26.дек	500 шт.	900,00 ₽
3	27.дек	600 шт.	1 000,00 ₽
4	28.дек	700 шт.	1 100,00 ₽
5	29.дек	800 шт.	1 200,00 ₽
6	30.дек	900 шт.	1 500,00 ₽
7	31.дек	1000 шт.	1 900,00 ₽
8	01.янв	150 шт.	1 500,00 ₽
9	02.янв	50 шт.	900,00 ₽
10	03.янв	100 шт.	750,00 ₽
11	04.янв	200 шт.	800,00 ₽
12	05.янв	300 шт.	900,00 ₽
13	06.янв	550 шт.	1 300,00 ₽
14	07.янв	200 шт.	1 100,00 ₽
15	08.янв	950 шт.	600,00 ₽

Тогда можно построить множественную регрессию, где цена зависит и от спроса, и от даты. График будет выглядеть следующим образом как на рис. 17.15.

Уравнение множественной регрессии, где спрос – x_k , дата продажи – t_k :

$$y_k(x_k, t_k) = ax_k + bt_k$$

Воспользуемся методом наименьших квадратов для нахождения коэффициентов a и b .

Для нахождения значений a и b запишем функцию $f(a, b)$:

$$f(a, b) = \sum_{k=1}^n (y_k - ax_k - bt_k)^2$$

Найдем частные производные $f(a, b)$ по а и по b:

$$\frac{\partial f(a, b)}{\partial a} = \sum_{k=1}^n 2(y_k - ax_k - bt_k)(-x_k)$$

$$\frac{\partial f(a, b)}{\partial b} = \sum_{k=1}^n 2(y_k - ax_k - bt_k)(-t_k)$$

Приравняем частные производные к 0:

$$a \sum_{k=1}^n x_k^2 + b \sum_{k=1}^n x_k t_k = \sum_{k=1}^n y_k x_k$$

$$a \sum_{k=1}^n x_k t_k + b \sum_{k=1}^n t_k^2 = \sum_{k=1}^n y_k t_k$$

Решая систему, уравнений найдем значения a^* и b^*

Итак, построив график полученной функции, получаем поверхность или совокупность прямых. Эта поверхность изображена на рис. 17.16.

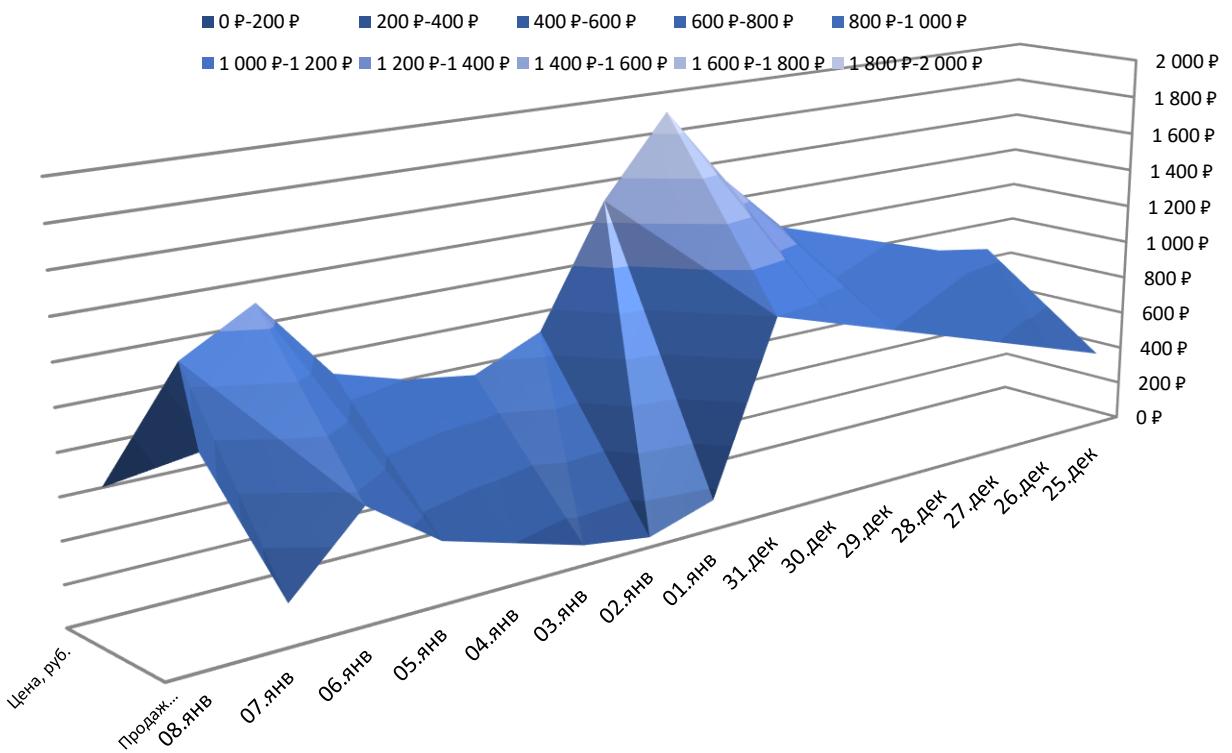


Рис. 17.15. Зависимость цены от спроса и даты

Рассмотрим еще один пример применения множественной регрессии на следующей задаче. Фирма-перевозчик хочет достичь максимального объема продаж. Фирма выписала все характеристики погрузчиков и все возможные варианты, существующие на рынке. Характеристиками являются грузоподъемность, цена, наличие напольной рамы, наличие синхронизации, число двигателей, суммарная мощность двигателей, высота подхвата в нижнем положении, высота подхвата в нижнем положении, максимальная высота подъема, скорость подъема, гарантийный срок, срок службы, время на рынке, внешний вид, срок поставки, доступность запчастей, наличие системы смазки, масса.

По результатам сбора статистики по рынку была построена регрессия:

$$V = -1769,77 - 65,09X_1 - 0,03X_2 + 68,79X_3 + 147,54X_5 + 2,53X_7 \\ + 1,06X_8 + 25,75X_{12} - 132,26X_{13} - 12,41X_{14} + 107,78X_{15} \\ + 397X_{16}$$

Не вошедшие параметры X_i незначительно влияют на объем продаж V .

Результатом работы явился прогноз объема продаж погрузчиков в 1010 шт., при этом с вероятностью 95% отклонение от прогнозируемого значения не выйдет за границы диапазона от 695 шт. до 1332 шт.

18. Многокритериальная оптимизация

До этого момента оптимизационная задача сводилась к системе с множеством ограничений и одной оптимизируемой целевой функцией. Однако представим, что, исследуя какой-то предмет или область необходимо оценивать его по нескольким критериям, которые невозможно уместить в рамках одной функции. Например, обращаясь к задаче на графы, будет необходимо найти не только кратчайший путь, но и самый дешевый, а протяженность и цена – это разные критерии оценки пути. Тогда происходит переход уже к многокритериальной оптимизации. Многокритериальная оптимизация – поиск оптимизированного решения по двум или более целевым функциям, область определения которых может не соотноситься друг с другом.

Задачи многокритериальной оптимизации сводят к двум направлениям:

1. Многокритериальная оптимизация по Парето;
2. Сведение к однокритериальной задаче.

Причем во втором методе присутствует весомое количество подходов к сведению многокритериального выбора к поиску по одному эффективному показателю. Далее будет выборочно представлено три наиболее популярных метода, а именно: линейная свертка критериев; метод «идеальной точки», метод «контрольных показателей».



Вильфредо Федерико Дамасо Парето (15.07.1848, Париж - 20.08.1923, Швейцария) — итальянский инженер, экономист и социолог. Один из основоположников теории элит. Он разработал теории, названные впоследствии его именем: статистическое Парето-распределение (80/20) и Парето-оптимум, широко используемые в экономической теории и иных научных дисциплинах.



Многокритериальная оптимизация по Парето

Рассмотрим на примере задачи о футболистках двухкритериальную задачу, в таблице 18.1 представлены футболисты с их характеристиками. Постановка задачи, следующая:

Выбрать лучших нападающих в футбольный клуб по критериям максимальной личной результативности (голы) и командной сыгранности (голевые передачи).

Таблица 18.1

Футбольные игроки с показателями голов и пасов

<i>k</i>	Фамилия	Голы	Пасы
1	Перваков	12 шт.	12 шт.
2	Вторянский	9 шт.	12 шт.
3	Третьяк	12 шт.	5 шт.
4	Четвертак	1 шт.	10 шт.
5	Пятаков	10 шт.	2 шт.
6	Шестерман	7 шт.	7 шт.
7	Семидзе	6 шт.	13 шт.
8	Восьмушкин	14 шт.	0 шт.
9	Девятуллин	0 шт.	12 шт.
10	Децимал	13 шт.	10 шт.
11	Двупалкин	5 шт.	4 шт.
12	Дюжев	12 шт.	6 шт.
13	Чертодюжев	2 шт.	15 шт.
14	Четыркин	2 шт.	5 шт.
15	Фифтин	9 шт.	12 шт.
16	Нехов	1 шт.	14 шт.

Представим эти точки на плоскости (рис. 18.1). На плоскости каждый претендент в футбольный клуб значится точкой, по оси абсцисс у которой голы в шт., а по оси ординат – переданные пасы в шт. Таким образом, можно наглядно сравнивать игроков по данным показателям. Например, претендент №5, забивший 10 голов и сделавший 2 паса является лучшим по результативности, сравнивая его с игроком №9, который забил ни одного гола, однако 9 игрок сильно опережает его по командной игре благодаря 12 переданным пасам.

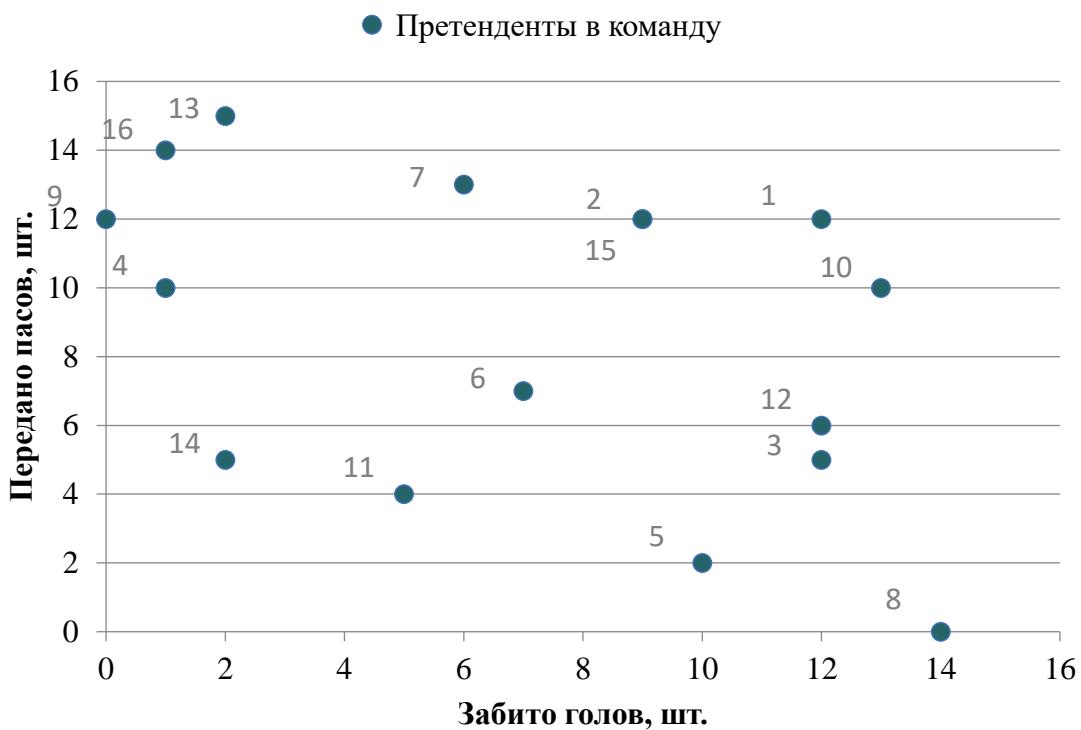


Рис. 18.1. Диаграмма соотношения голов и пасов претендентов

Выделим среди данных точек не улучшаемые, то есть для каждой точки выделим подпространство в направлении к максимуму: наибольшему количеству голов и пасов – рис. 18.2.

На рис. 18.2 отмечено 6 точек, 5 из которых – неулучшаемые. Рассмотрим 6 и сравним ее с остальными для точного понимания неулучшаемого значения. 11 футболист забил 5 голов и передал 4 паса, что может являться неплохим результатом, однако в команде есть игрок под номером 7, который имеет 13 пасов и 6 голов, что явно лучше результатов 11 игрока. Сравнивая 7 игрока с остальными, нельзя найти такого, кто забил бы больше голов и в то же время передал больше пасов. То есть если для 11 игрока мы можем найти другого, который будет лучше него по обоим показателям, то для 13, 7, 1, 10, 8 – таких игроков нет.

Все, что находится в первой четверти, ограниченной оранжевым цветом, устремлено к максимуму по голам и пасам, а для вышеперечисленных точек в их четвертях других значений нет. В соответствии с этим они по определению не улучшаемы.

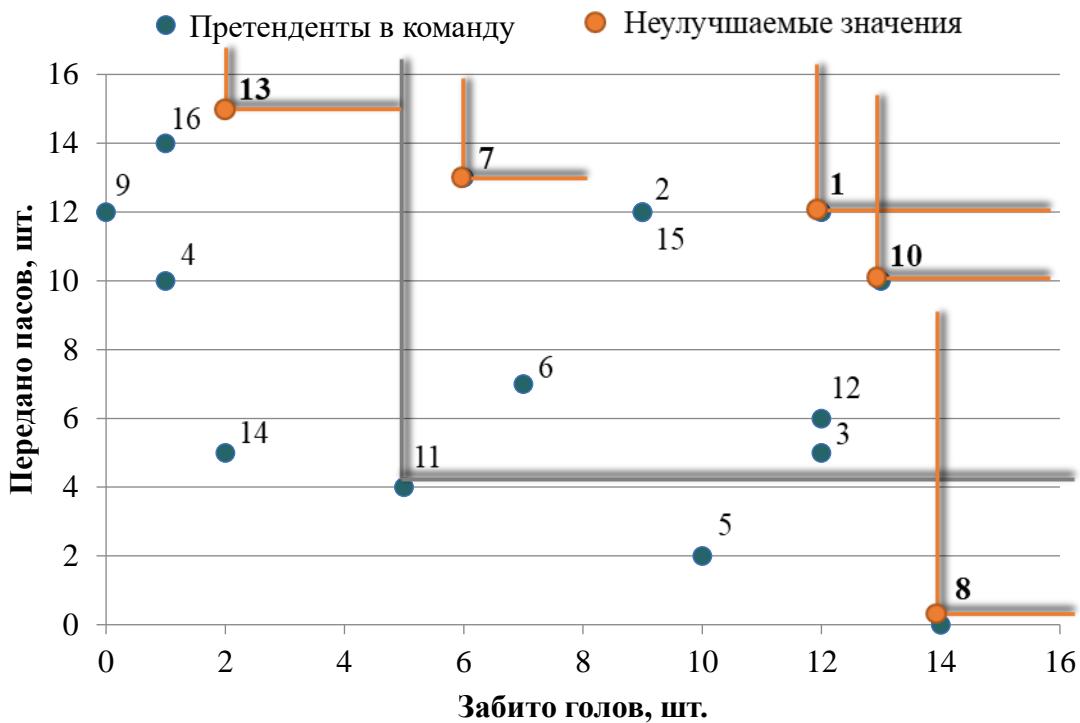


Рис. 18.2. Диаграмма неулучшаемых претендентов

Данные точки образует поверхность Парето, которая содержит в себе оптимальные по Парето или паретооптимальные значения, то есть неулучшаемые (таблица 18.2).

Хотя эти точки уже несравнимы, однако из них можно выделить Лучшего бомбардира игрока №8, а также лучшего ассистента – игрока №9, так как эти точки являются крайними. А в отношении крайних можно найти только компромиссы, учитывая необходимый баланс по голам и пасам, поэтому все не крайние точки называются компромиссными решениями.

Таблица 18.2

Оптимальные по Парето игроки

<i>k</i>	Фамилия	Голы	Пасы
8	Восьмушкин	14 шт.	0 шт.
10	Децимал	13 шт.	10 шт.
1	Перваков	12 шт.	12 шт.
7	Семидзе	6 шт.	13 шт.
13	Чертодюжев	2 шт.	15 шт.

В данном случае поверхность представлена дискретными точками, это можно увидеть на рис. 18.3, однако возможны показатели не дискретные, тогда поверхность будет являться непрерывной моделью. Нахождение точек такой модели будет сложнее, однако аналитически они в любом случае рассчитываются посредством специальных методов. Однако разницы в оценке не будет, так как выше было определено, что значения лучше, чем те, что представлены на паретооптимальной плоскости, найти невозможно. С экономической точки зрения она встречается как поверхность безразличия, подчеркивая факт равнозначности представленных неулучшаемых результатов.

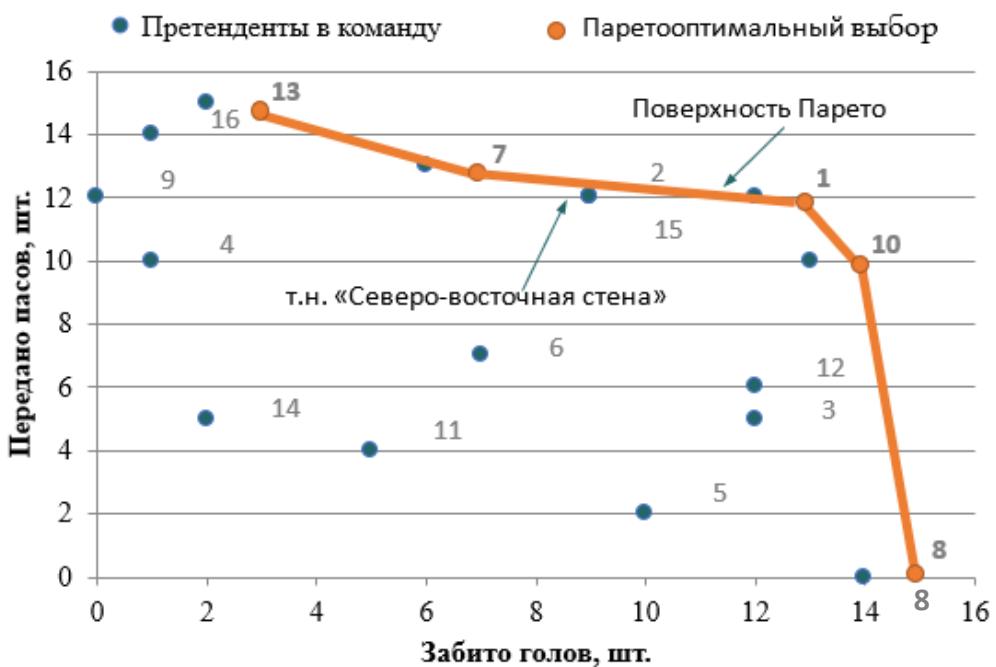


Рис. 18.3. Поверхность Парето

Представим математическую интерпретацию данной задачи для возможности поиска решения не только ручным перебором.

Математическая запись задачи.

Найти множество игроков x_k , лучших по критериям:

1) Наибольшего количества забитых голов:

$$W_1 = f_1(x_k) \rightarrow \max$$

2) Выполненных голевых передач:

$$W_2 = f_2(x_k) \rightarrow \max$$

Решением задачи является паретооптимальное множество игроков x_k :

$$\{x_1, x_7, x_8, x_{10}, x_{13}\},$$

доставляющее максимум критериям W_1 и W_2 .

Так как решений много эксперту необходимо из них выбрать те, которые будут являться удовлетворительными исходя из соображений выгоды каждого из них в конкретный момент времени.

Выше представлены функции, устремленные в сторону максимума, однако в жизни вполне реальны ситуации обратные. Например, когда мы хотим устремить к максимуму забитые голы, но при этом к минимуму количество переданных пасов как на рис. 18.4. Или же ситуация, когда нам нужно устремить к минимуму обе функции, чтобы найти худших игроков – рис. 18.5.

Максимум - минимум:

$$W_1 = f_1(x_k) \rightarrow \max$$

$$W_2 = f_2(x_k) \rightarrow \min$$

Минимум - минимум:

$$W_1 = f_1(x_k) \rightarrow \min$$

$$W_2 = f_2(x_k) \rightarrow \min$$

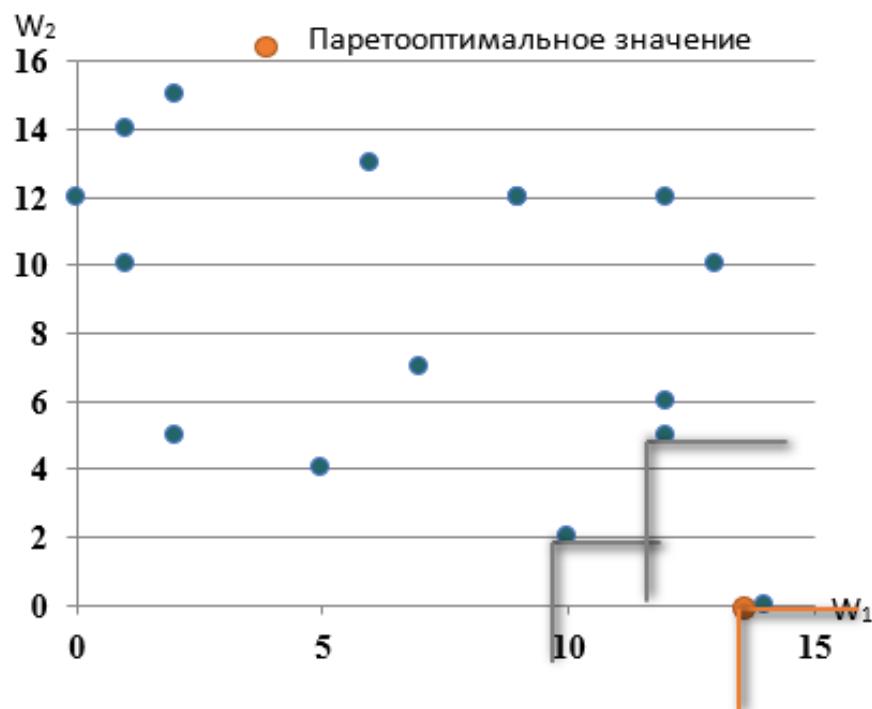


Рис. 18.4. Функции разноименных знаков

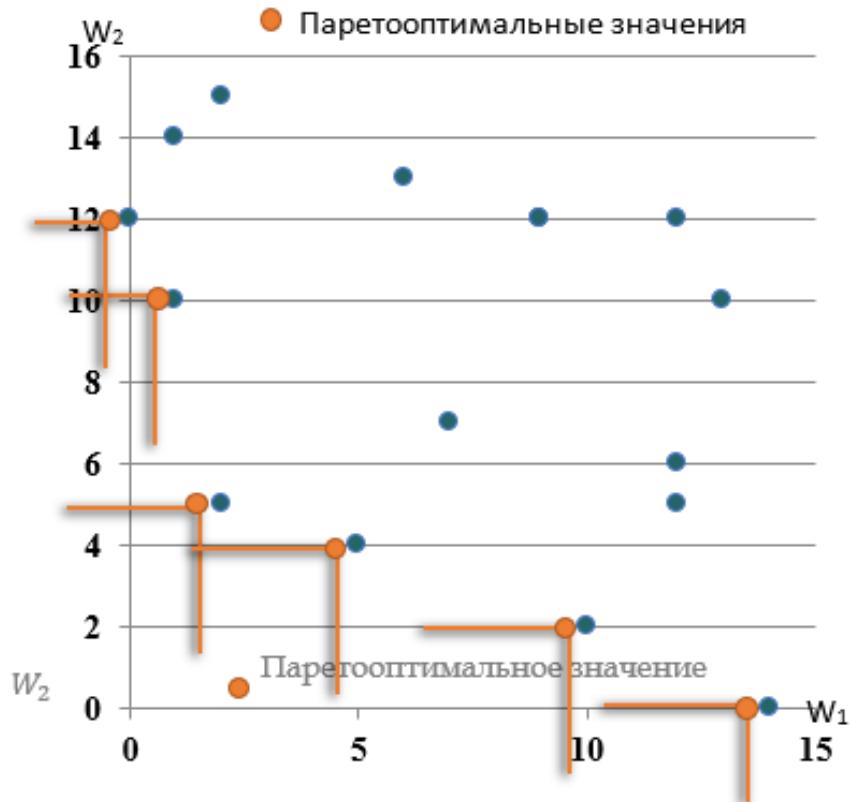


Рис. 18.5. Функции, стремящиеся к \min

При разноименных знаков мы имеем устремление к началу координат в 4 координатной четверти, так как устремляем голы к максимуму, а пасы к минимуму, при этом отрицательных значений быть не может.

В случае поиска худших игроков поиск будет осуществляться по третьей четверти, устремляя голы и пасы к нулю.

Так как критериев может быть много, соответственно поиск по каждому реализовывается своей функцией.

Запись задачи в общем виде.

Найти x , доставляющий экстремум критериям:

$$W_1 = f_1(x) \rightarrow \text{extr}$$

$$W_2 = f_2(x) \rightarrow \text{extr}$$

...

$$W_n = f_n(x) \rightarrow \text{extr}$$

при выполнении ограничений:

$$\varphi(x) \leq 0, \quad x \geq 0$$

Соответственно, если достигается минимум — точка экстремума называется точкой минимума, а если максимум — точкой максимума. Если $f(x) \rightarrow \max$, то критерий называется позитивным, в противном случае — негативным.

Сделать аналогичный выбор игроков по 7 критериям, представленным в таблице 18.3. Выбор футболиста команду следует осуществлять по обширному количеству показателей. Например, учесть и скорость бега, которая должна устремляться к максимуму, так же выносливость, большой опыт практики, наименьший возраст игрока для его потенциала в дальнейшем, и его стоимость его содержание, которая должно сводиться к минимуму.

Таблица 18.3

Исходные данные для многокритериальной оптимизации

№ <i>k</i>	Фамилия <i>x</i>	Гол, ед. <i>f</i> ₁ (<i>x</i>)	Пас, ед. <i>f</i> ₂ (<i>x</i>)	Скор., <i>f</i> ₃ (<i>x</i>)	Выносл., <i>f</i> ₄ (<i>x</i>) мин	Опыт, <i>f</i> ₅ (<i>x</i>) лет	Возраст, <i>f</i> ₆ (<i>x</i>) лет	Цена, \$M <i>f</i> ₇ (<i>x</i>)
1	Перваков	12	12	4,6	80	7	24	10,2
2	Вторянский	9	12	5,4	81	1	16	15
3	Третьяк	12	5	3,9	75	3	19	8,8
4	Четвертак	1	10	4,9	63	21	36	10
5	Пятаков	10	2	4,3	66	20	35	9,9
6	Шестерман	7	7	4,5	86	18	35	11,5
7	Семидзе	6	13	5,5	79	5	22	8,6
8	Восьмушкин	14	0	5,5	61	9	26	11,4
9	Девятуллин	0	12	3,9	63	5	22	11,8
10	Децимал	13	10	5,5	65	21	36	12,2
11	Двупалкин	5	4	3,3	86	17	31	11,1
12	Дюжев	12	6	2,6	84	17	33	10,8
13	Чертодюжев	2	15	3,5	59	15	30	7,1
14	Четыркин	2	5	4,2	69	6	23	6,8
15	Фифтин	9	12	5,1	83	7	24	7,8
16	Нехов	1	14	4,5	65	5	19	10,5

Данную задачу можно решать, применяя методы, аналогичные двухкритериальной задаче, однако в связи с увеличением количества критериев данная

задача будет решаться в аналитическом виде. Excel предоставляет инструмент, который способен представить семикритериальную задачу в виде лепестковой диаграммы, которую можно увидеть на рис. 18.6. На ней изображены 7 осей, и каждая точка отложена своей координатой по соответствующей оси, то есть каждый футболист является некоторой фигурой на плоскости с семью вершинами.

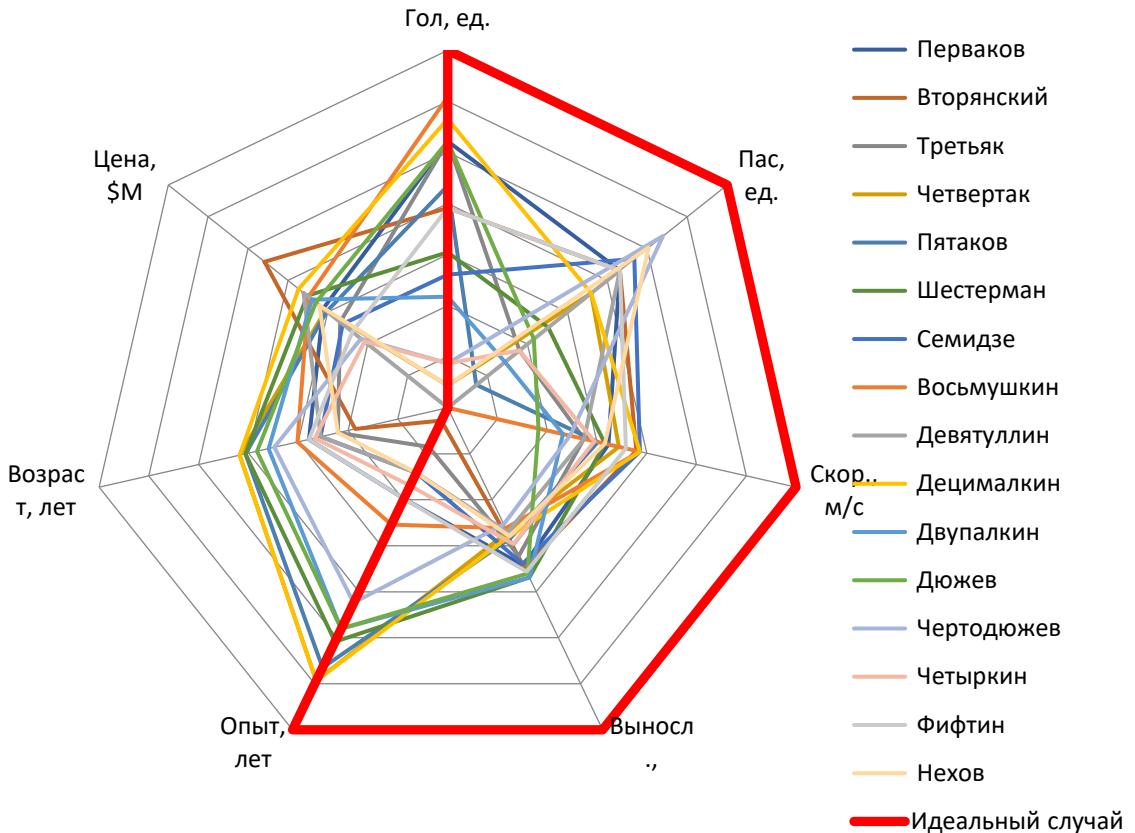


Рис. 18.6. Представление результатов в виде лепестковой диаграммы

Идеальным случаем, к которому должна стремиться каждая фигура, является устремление голов, пасов, скорости, выносливости, опыта к максимуму, а возраста и цены к минимуму. Поэтому каждая фигура должна повторять этот идеальный случай. Однако такое отношение не даст четкого представления, каким образом искать наилучшие точки при множестве критерии. Поэтому мы будем искать, как и при случае двухкритериальной задачи, фигуры, которые будут лучше по всем показателям. Для того чтобы выбрать такие, необходимо рассмотреть каждую и проверить, что она не находится внутри других фигур, а наоборот, покрывает их собой. Только в этом случае она будет их превосходить

по критериям. А пересекающиеся фигуры, которые не накрывают друг друга, будут составлять паретооптимальное множеством. Но стоит учесть, что в случае с устремлением к минимуму нам нужно рассматривать фигуры иначе. Для наглядности можно инвертировать функции, тогда минус минимум станет максимумом, и можно воспользоваться вышеописанным способом, ища фигуры, перекрывающие друг друга.

Сведение к однокритериальной задаче

Переход от многокритериальной задачи к однокритериальной помогает выявить из множества единственное решение. Так как у однокритериальной задаче есть только одно решение, то мы сводим задачу к простоте, оптимизируя только его. Если оптимизация по Парето требует полного присутствия принимающего решения лица, то в однокритериальных задачах такой человек либо совсем не требуется, либо его участия минимально.

Линейная свертка критериев

Первый способ сведения задачи к одному критерию является *линейной сверткой*. Он получил широкое распространение во многих областях, в том числе и в экономике.

Допустим, у нас так же есть два критерия: количество забитых голов и переданных пасов, каждому из них будет назначен соответствующий коэффициент важности, причем в общей сумме все коэффициенты дают единицу. Складывая произведением коэффициентов на значения их функций, получается некоторый показатель, который заменяет два других.

Переход от двух критериев к одному

$$W_1 = f_1(x) \rightarrow \max$$

$$W_2 = f_2(x) \rightarrow \max$$

производится путем их «сворачивания» в выражение:

$$W = \alpha_1 f_1(x) + \alpha_2 f_2(x) \rightarrow \max$$

где α_1 и α_2 - весовые коэффициенты, отражающие значимость критериев.

$$\alpha_1 + \alpha_2 = 1$$

Далее решается классическая задача по одному обобщенному критерию.

Несмотря на различие двух критериев и их несравнимость, как, например, килограммы и секунды, такая сумма функций имеет место быть в силу того, что в итоге получается некоторое среднее значение. И в математике оно часто упоминается как средневзвешенное значение.

Пример. Выбрать лучших нападающих в футбольный клуб по критериям максимальной личной результативности (голы) и командной сыгранности (головые передачи). Важность личной результативности определить в 70%, командной – в 30%.

Это та же самая задача, что была представлена выше, однако для каждого критерия назначен свой коэффициент значимости, в соответствии с ним наша уже линейная сверточная функция принимает следующий вид:

$$\alpha_1 = 0,7 \quad \alpha_2 = 0,3$$

$$W = 0,7 \cdot f_1(x_k) + 0,3 \cdot f_2(x_k) \rightarrow \max$$

Представление линейной свертки в общем виде:

$$W = \sum_{i=1}^n \alpha_i f_i(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \dots + \alpha_n f_n(x) \rightarrow \max$$

при условии, что все критерии $f_i(x)$ стремятся к максимуму

$$f_i(x) \rightarrow \max, \quad \forall i = 1 \dots n$$

Разноименные критерии:

Если некоторые критерии стремятся к минимуму, то свертка видоизменяется.

Например, при $f_1(x) \rightarrow \max, \quad f_2(x) \rightarrow \min$ свертка приобретает один из следующих видов:

$$W = \frac{f_1(x)}{f_2(x)} \rightarrow \max$$

$$W = \alpha_1 f_1(x) - \alpha_2 f_2(x) \rightarrow \max$$

Теперь каждый человек обозначается один числом, которое складывается из количества голов, умноженное на 0,7 и количество пасов, умноженное на 0,3. Значения свертки представлены в таблице 18.4 в столбце «W». Это число уже можно оценивать привычным способом, решая обычную оптимизационную задачу.

Таблица 18.4

Игроки с линейной сверткой

№ <i>k</i>	Фамилия <i>x_k</i>	Гол, ед. <i>f₁(x_k)</i>	Пас, ед. <i>f₂(x_k)</i>	Свертка <i>W</i>
1	Перваков	12 шт.	12 шт.	12,0
2	Вторянский	9 шт.	12 шт.	9,9
3	Третьяк	12 шт.	5 шт.	9,9
4	Четвертак	1 шт.	10 шт.	3,7
5	Пятаков	10 шт.	2 шт.	7,6
6	Шестерман	7 шт.	7 шт.	7,0
7	Семидзе	6 шт.	13 шт.	8,1
8	Восьмушкин	14 шт.	0 шт.	9,8
9	Девятуллин	0 шт.	12 шт.	3,6
10	Децимал	13 шт.	10 шт.	12,1
11	Двупалкин	5 шт.	4 шт.	4,7
12	Дюжев	12 шт.	6 шт.	10,2
13	Чергодюжев	2 шт.	15 шт.	5,9
14	Четыркин	2 шт.	5 шт.	2,9
15	Фифтин	9 шт.	12 шт.	9,9
16	Нехов	1 шт.	14 шт.	4,9

Пример. В аналогичной задаче найти самых некомандных игроков, которые забили больше всего голов $f_1(x) \rightarrow \max$ но отдали меньше всего передач $f_2(x) \rightarrow \min$.

Свертка критериев:

$$W = f_1(x) - f_2(x) \rightarrow \max$$

Свертка всех критериев представлена в таблице 18.5. По результатам этой свертки можно сразу определить, что игроки №5, 8 самые «некомандные», так как у них максимальные значения.

Таблица 18.5

Игроки с max голов и min пасов

№ <i>k</i>	Фамилия x_k	Гол, ед. $f_1(x_k)$	Пас, ед. $f_2(x_k)$	Свертка W
1	Перваков	12 шт.	12 шт.	1,0
2	Вторянский	9 шт.	12 шт.	0,8
3	Третьяк	12 шт.	5 шт.	2,4
4	Четвертак	1 шт.	10 шт.	0,1
5	Пятаков	10 шт.	2 шт.	5,0
6	Шестерман	7 шт.	7 шт.	1,0
7	Семидзе	6 шт.	13 шт.	0,5
8	Восьмушкин	14 шт.	0 шт.	∞
9	Девятуллин	0 шт.	12 шт.	0,0
10	Децимал	13 шт.	10 шт.	1,3
11	Двупалкин	5 шт.	4 шт.	1,3
12	Дюжев	12 шт.	6 шт.	2,0
13	Чертодюжев	2 шт.	15 шт.	0,1
14	Четыркин	2 шт.	5 шт.	0,4
15	Фифтин	9 шт.	12 шт.	0,8
16	Нехов	1 шт.	14 шт.	0,1

Метод линейной свертки избавляет человека от участия в решении почти полностью, так как это участие требуется лишь на начальной стадии, когда расставляются коэффициенты критериям. С учетом того, что такую задачу не нужно анализировать при выдаче результата, то ее вполне можно отдать на вычисление компьютеру. Однако в некоторых случаях нам важно учесть факторы, которые компьютеру не могут быть представлены на машинном языке, тогда оптимизация по Парето будет подходящим методом для таких ситуаций.

Однако возникает вопрос следующий вопрос: что будет, если изменять значения α от 0 до 1?

Запишем свертку множества критериев в виде:

$$W^{(k)} = \alpha f_1(x) + (1 - \alpha) f_2(x) \rightarrow \max$$

Где $f_1(x)$ – количество переданных пасов, $f_2(x)$ – количество забитых голов.

Параметр свертки последовательно принимает значения от 0 до 1:

$$0 = \alpha_0 < \alpha_1 < \alpha_2 < \dots < \alpha_m = 1, \quad k = \overline{0, m}$$

Значимость критериев $f_1(x)$ и $f_2(x)$ постепенно меняется от одного к другому, главное, что в сумме они дают единицу, как показано на рис. 18.7.

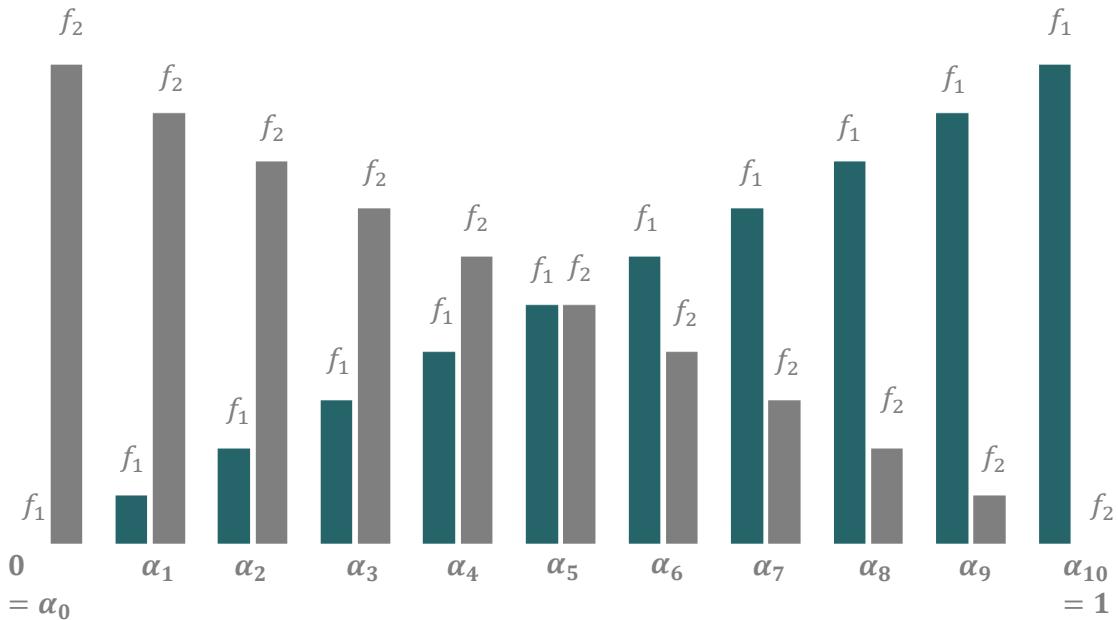


Рис. 18.7. Изменение весовых коэффициентов

Проследим изменения оптимального решения в зависимости от значения параметра свертки α (параметр значимости критериев).

В этом случае получается серия решения со своим значением свертки, из которых выбирается максимальное, в таблице 18.6 показаны все рассматриваемые изменения параметра.

На одном диапазоне параметра свертки, где важность забитых голов ниже, футболист №13 получает максимальное значение свертки, на другом – где важность гола уже начинает преобладать, равняясь важности паса, футболист №1 лидирует. Когда важность гола значительно преобладает над пасом, то на первом месте оказывается футболист №10. И при необходимости безусловного бомбардира, лидирует футболист №8.

Таблица 18.6

Значения свертки при различных весах

№ <i>k</i>	Фамилия <i>x</i>	Значение свертки $W = \alpha f_1(x) + (1 - \alpha)f_2(x)$										
		$\alpha = 0$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
1	Перваков	12	12	12	12	12	12	12	12	12	12	12
2	Вторянский	12	11,7	11,4	11,1	10,8	10,5	10,2	9,9	9,6	9,3	9
3	Третьяк	5	5,7	6,4	7,1	7,8	8,5	9,2	9,9	10,6	11,3	12
4	Четвертак	10	9,1	8,2	7,3	6,4	5,5	4,6	3,7	2,8	1,9	1
5	Пятаков	2	2,8	3,6	4,4	5,2	6	6,8	7,6	8,4	9,2	10
6	Шестерман	7	7	7	7	7	7	7	7	7	7	7
7	Семидзе	13	12,3	11,6	10,9	10,2	9,5	8,8	8,1	7,4	6,7	6
8	Восьмушкин	0	1,4	2,8	4,2	5,6	7	8,4	9,8	11,2	12,6	14
9	Девятуллин	12	10,8	9,6	8,4	7,2	6	4,8	3,6	2,4	1,2	0
10	Децимал	10	10,3	10,6	10,9	11,2	11,5	11,8	12,1	12,4	12,7	13
11	Двупалкин	4	4,1	4,2	4,3	4,4	4,5	4,6	4,7	4,8	4,9	5
12	Дюжев	6	6,6	7,2	7,8	8,4	9	9,6	10,2	10,8	11,4	12
13	Чертодюжев	15	13,7	12,4	11,1	9,8	8,5	7,2	5,9	4,6	3,3	2
14	Четыркин	5	4,7	4,4	4,1	3,8	3,5	3,2	2,9	2,6	2,3	2
15	Фифтин	12	11,7	11,4	11,1	10,8	10,5	10,2	9,9	9,6	9,3	9
16	Нехов	14	12,7	11,4	10,1	8,8	7,5	6,2	4,9	3,6	2,3	1

Вышеописанные решения, рассмотренные при изменении коэффициента, очень сильно напоминают оптимальные по Парето решения.

Если вернуться назад, то можно увидеть, что это они и есть, по таблице 18.7 и рис. 18.8 можно отметить те же решения при заданных уровнях параметра.

Паретооптимальные решения – это решения, которые получаются при переборе параметра линейной свертки от 0 до 1, то есть при изменении значимости от одного критерия к другому.

Замечание. При самом наименьшем шаге коэффициента линейной свертки, например, в модели только с целочисленными значениями переменных теряется часть решений, которая составляет невыпуклую фигуру.

При этом данное замечание не распространяется на непрерывную модель.

Таблица 18.7

Оптимальные решения

<i>k</i>	Фамилия	Голы	Пасы	α
8	Восьмушкин	14 шт.	0 шт.	0...0,2
10	Децимал	13 шт.	10 шт.	0,3...0,6
1	Перваков	12 шт.	12 шт.	0,7...0,9
7	Семидзе	6 шт.	13 шт.	1
13	Чернодюжев	2 шт.	15 шт.	α

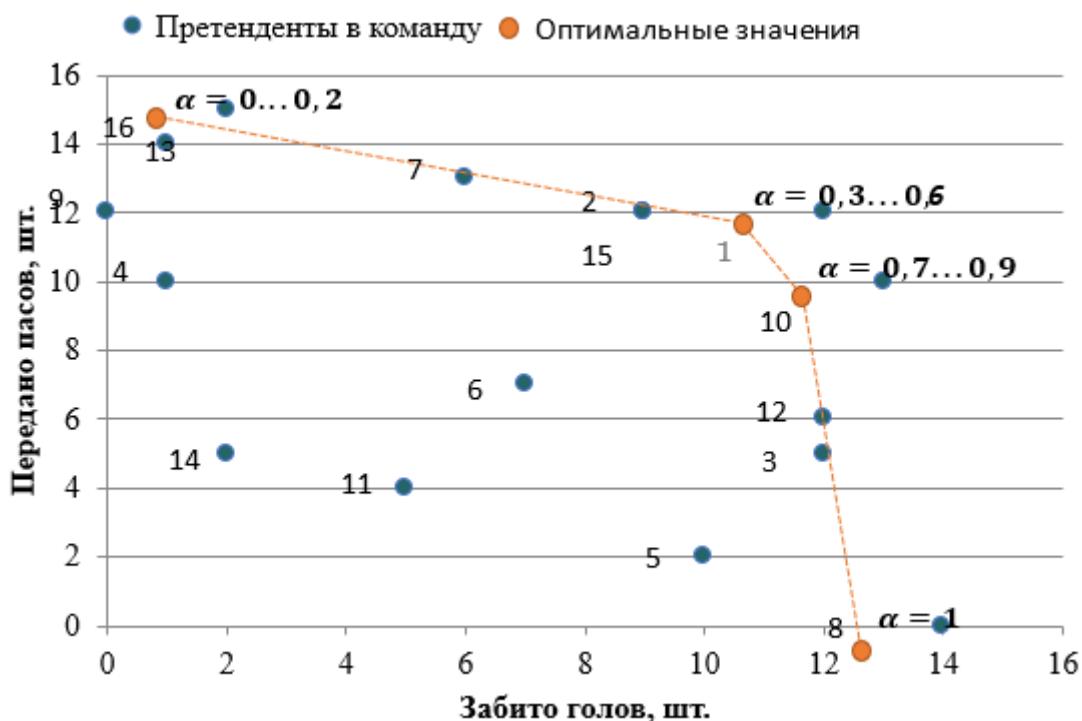


Рис. 18.8. Плоскость Парето через линейную свертку

То есть меняя коэффициенты можно переходить к другим решениям, которые изначально составляют плоскость Парето.

Метод идеальной точки

Если взглянуть на многоокритериальную задачу с другого угла, то можно рассмотреть ее решение не посредством линейной свертки, а через расстояние до наилучшего решения.

При рассмотрении поверхности Парето мы стремились к некоторой идеальной точки, которая была недосягаема, но в целом все точки стремились к ней и в рамках задачи были неулучшаемыми. Перейдя к еще одному методу сведения к однокритериальной задаче – метод идеальной точки, мы также будем ее выбирать, но учитывать ее в решении более практически.

Алгоритм будет следующий.

1. Выбирается точка, которая лучше остальных по всем критериям, причем она должна быть относительно приближена к ним.
2. Из каждой точки строится расстояние до идеальной, соответственно для двух критериев поиск будет сводится к нахождению гипотенузы через построения проекции для прямоугольного треугольника.

Такой алгоритм используется для поиска *сбалансированного* решения, наиболее близкого к идеальному.

Замечание. Так как значения по критериям могут сильно отличаться по своим шкалам, то для корректности результатов следует провести нормализацию значений, чтобы каждый параметр, например, измерялся по шкале от 0 до 1.

Например, воспользовавшись формулой минимакс нормализации:

$$x_i = \frac{x_i - x_{\min}}{x_{\max}}$$

Математическая интерпретация:

Пусть имеется n -критериальная задача:

$$W_i = f_i(x) \rightarrow \max \quad , \quad i = \overline{1, n}$$

Выберем «идеальную точку» - наилучшие значения каждого критерия:

$$(f_1^*, f_2^*, \dots, f_n^*)$$

В случае с двумерной пространством поиск расстояния осуществляется по теореме Пифагора, где для футболиста №1 формула будет следующей:

$$W_1 = \sqrt{a^2 + b^2}$$

Если взять

$$a = \frac{f_1(x) - f_1^*}{\max f_i(x)}, \quad b = \frac{f_2(x) - f_2^*}{\max f_i(x)}$$

как на рис. 18.9, то от теоремы Пифагора можно перейти к формуле расстояния между двумя точками с известными координатами, причем за каждую координату делится на $\max f_i(x)$ соответственно нормируя каждую шкалу критерия, так как координата в данном случае является критерием. И тогда нормированная идеальная точка будет иметь следующие координаты $(1,1,\dots,1)$.

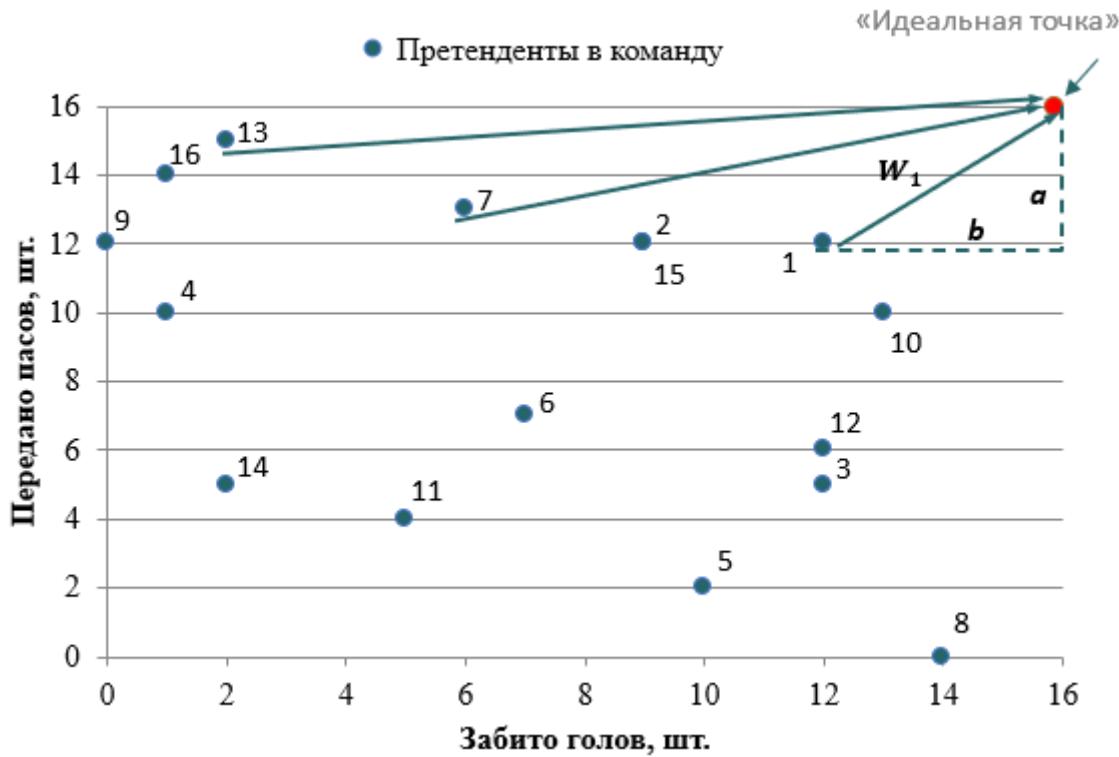


Рис. 18.9. Идеальная точка

Переходя к поиску расстояния по всему множеству критериев, формула будет содержать в себе соответственно множество разниц между всеми координатами критерия и идеальной точки. Для этого ведем глобальный критерий W , который минимизирует нормированное расстояние до идеальной точки:

$$W = \sqrt{\sum_{i=1}^n \left(\frac{f_i^* - f_i(x)}{\max f_i(x)} \right)^2} \rightarrow \min$$

Решая задачу, находим такое W , при котором все функции $f_i(x)$ ближе всего к своим идеальным значениям f_i^{max} , то есть к нормализованной 1.

Преимущество метода идеальной точки заключается в том, что присутствие эксперта, который анализирует какие-либо параметры, не требуется, поэтому данный способ может быть полностью автоматизирован.

Возвращаясь к задаче о футболистах, необходимо выбрать наиболее сбалансированных игроков в футбольный клуб, все показатели которых значительно выше минимальных границ – таблица 18.8.

Таблица 18.8

Нормированные критерии с расстоянием до идеальной точки

$\# k$	Фамилия x	Гол, ед.	Пас, ед.	Скор., м/с	Выносл.	Опыт, лет	Возраст, лет	Цена, \$M	Расст. до идеальн.
1	Перваков	0,16	0,16	0,12	0,01	0,52	0,11	0,26	1,16
2	Вторянский	0,30	0,16	0,05	0,01	0,92	0,00	0,44	1,38
3	Третьяк	0,16	0,56	0,20	0,03	0,77	0,02	0,19	1,39
4	Четвертак	0,90	0,25	0,09	0,09	0,03	0,31	0,25	1,38
5	Пятаков	0,25	0,81	0,15	0,07	0,04	0,29	0,24	1,36
6	Шестерман	0,42	0,42	0,13	0,00	0,08	0,29	0,32	1,29
7	Семидзе	0,49	0,12	0,05	0,01	0,64	0,07	0,18	1,25
8	Восьмушкин	0,09	1,00	0,05	0,10	0,41	0,15	0,32	1,45
9	Девятуллин	1,00	0,16	0,20	0,09	0,64	0,07	0,33	1,58
10	Децимал	0,12	0,25	0,05	0,08	0,03	0,31	0,35	1,09
11	Двупалкин	0,56	0,64	0,28	0,00	0,10	0,23	0,30	1,46
12	Дюжев	0,16	0,49	0,40	0,00	0,10	0,27	0,29	1,31
13	Чертодюжев	0,81	0,06	0,25	0,12	0,16	0,22	0,09	1,31
14	Четыркин	0,81	0,56	0,16	0,05	0,58	0,09	0,07	1,53
15	Фифтин	0,30	0,16	0,07	0,01	0,52	0,11	0,13	1,14
16	Нехов	0,90	0,09	0,13	0,08	0,64	0,02	0,27	1,46
								min:	1,09

После нормирования каждого из 7 параметров и подсчета расстояния с учетом критерия W получается, что игрок № 10 наиболее близкий по всем параметрам к идеальному варианту. Игроκи № 15 и № 1 занимают второе и третье место соответственно, что и видно и по лепестковой диаграмме на рис. 18.10.

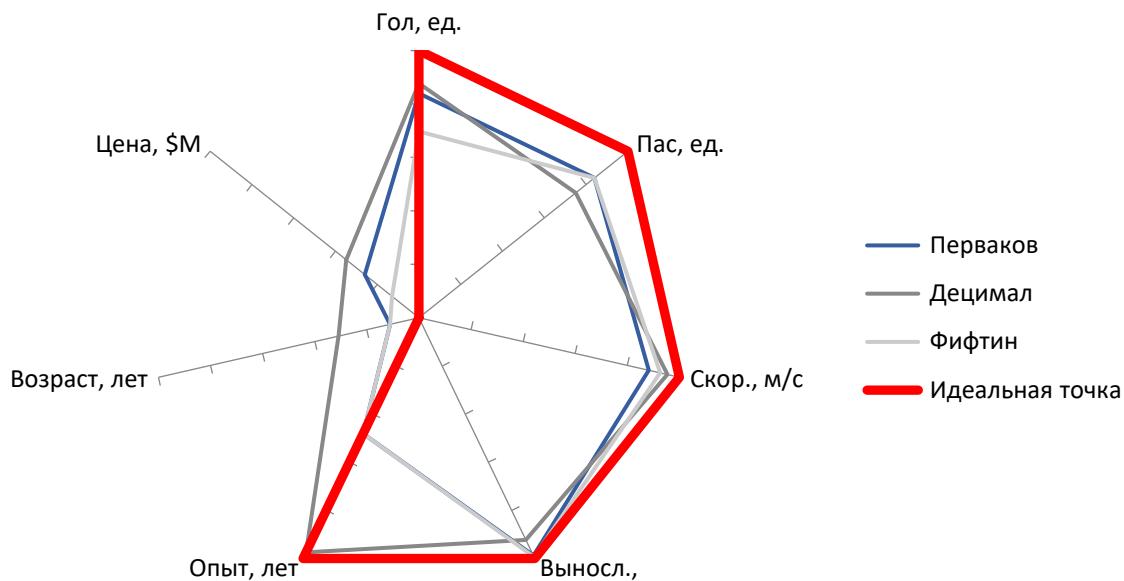


Рис. 18.10. Наиболее близкие к идеальной точке футболисты

Метод контрольных показателей

Рассмотрим еще один метод сведения задачи к одному критерию, который предлагает решение посредством взгляда на задачу с другой стороны.

Если в методе идеальной точки находилось расстояние для каждого решения, то в случае с контрольными показателями по каждой координате назначается нижняя граница, которую нельзя переступить. Например, не отбирать футболистов в команду, если показатель голов ниже 5, а пасов ниже 3.

Алгоритм действий

1. Определяется точка – нижняя граница как контрольный показатель, который значения не должны пересекать

2. Для каждой точки находится та координата, по которой данное решение ближе всего к проходному баллу. То есть выбирается расстояние до контрольного показателя не по всем координатам, а по одному самому близкому к нижней границе критерию.

3. Среди всех таких подсчетов расстояний выбирается то решение, которое по своей самой близкой к проходному баллу координате, находится как можно дальше от него. Иными словами, выбирается точка, наиболее отдаленная от нижней границы по худшему показателю.

Данный метод используется для поиска сбалансированного решения, которое максимально удалено от опасной границы по самому малому показателю.

Запишем в математическом виде:

Пусть имеется n -критериальная задача:

$$W_i = f_i(x) \rightarrow \max \quad , \quad i = \overline{1, n}$$

Назначим каждой функции $f_i(x)$ ее контрольный показатель – нижнюю границу:

$$f_i(x) \geq f_i^*$$

Введем глобальный критерий W , который максимизирует минимальное расстояние от функции $f_i(x)$ до нижней границы f_i^* :

$$W = \min_{i=1,n} \left\{ \frac{f_i(x)}{f_i^*}; \frac{f_j^*}{f_j(x)} \right\} \rightarrow \max_x,$$

где $f_i(x) \rightarrow \max, f_j(x) \rightarrow \min$. Здесь имеется ввиду, что для критериев, стремящихся к максимуму применяется первая часть выражения в фигурных скобках, а для критериев, стремящихся к минимуму – вторая.

Решая задачу, находим такое W , при котором все функции $f_i(x)$ дальше всего от своих нижних границ f_i^* .

Если рассмотреть этот метод в рамках задачи о футболистах, то для каждого футболиста мы подсчитаем значение расстояния до нижней границе по каждой координате. Среди этих расстояний находится минимум, неважно по какому именно показателю. И далее среди этих решений выбирается самое безопасное отдаленное значение, которое является решением.

Причем, те значения, для которых нормированное расстояние меньше единицы, либо оно ненормированное отрицательное, исключаются из поиска, значит, они уже не прошли проходной балл.

Таким образом, игроки №№ 4, 8, 9, 13, 14, 16 – не прошли по условию ограничений $f_i(x) \geq f_i^* W$. Игроки №№ 3, 15 имеют характеристики наиболее отличные от минимальных, что видно по таблице 18.9 и лепестковой диаграмме на рис. 13.11. А игрока №1 стоит рассмотреть, как близкого к оптимальным точкам.

Таблица 18.9

Критерии с расстоянием до контрольного показателя

Нижняя граница:		5	1	2,5	50	1	36	15
№ k	Фамилия x	Гол, ед. $f_1(x)$	Пас, ед. $f_2(x)$	Скор., м/с $f_3(x)$	Вы- носл., мин $f_4(x)$	Опыт, лет $f_5(x)$	Возраст, $f_6(x)$ лет	Цена, \$M $f_7(x)$
1	Перваков	12	12	4,6	80	7	24	10,2
2	Вторянский	9	12	5,4	81	1	16	15
3	Третьяк	12	5	3,9	75	3	19	8,8
4	Четвертак	1	10	4,9	63	21	36	10
5	Пятаков	10	2	4,3	66	20	35	9,9
6	Шестерман	7	7	4,5	86	18	35	11,5
7	Семидзе	6	13	5,5	79	5	22	8,6
8	Восьмушкин	14	0	5,5	61	9	26	11,4
9	Девятуллин	0	12	3,9	63	5	22	11,8
10	Децимал	13	10	5,5	65	21	36	12,2
11	Двупалкин	5	4	3,3	86	17	31	11,1
12	Дюжев	12	6	2,6	84	17	33	10,8
13	Чертодюжев	2	15	3,5	59	15	30	7,1
14	Четыркин	2	5	4,2	69	6	23	6,8
15	Фифтин	9	12	5,1	83	7	24	7,8
16	Нехов	1	14	4,5	65	5	19	10,5

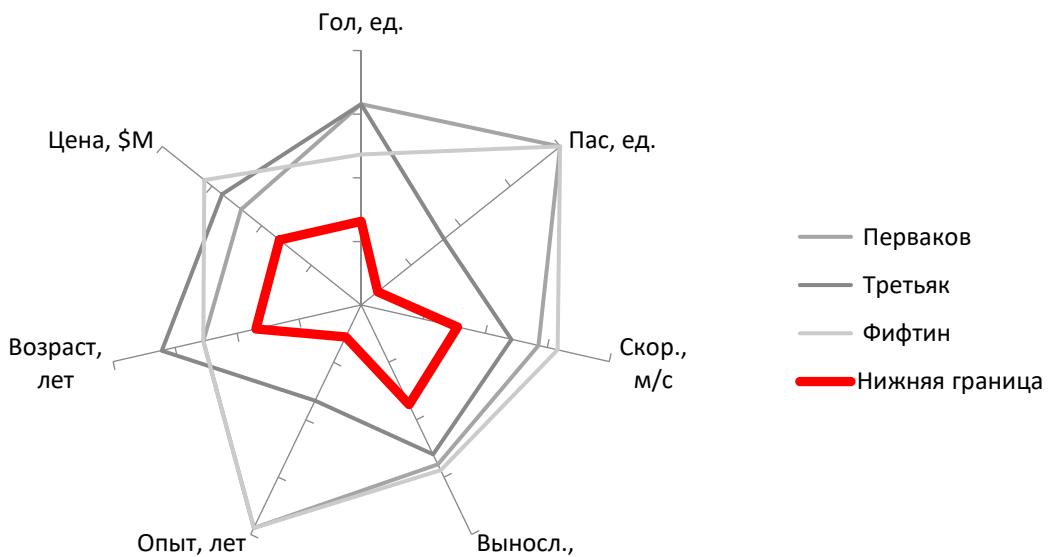


Рис. 18.11. Отдаленные от нижней границы футболисты

На лепестковой диаграмме (рис. 18.11) показано, что Фифтин сильнее всего удален от нижней границы.

19. Экспертные решения

В самых различных ситуациях возникает необходимость определенной оценки возможных вариантов и принятия решений. Примерами таких ситуаций являются суд присяжных, на котором принимается решение о виновности или невиновности человека, путем учета мнений присяжных, фигурное катание, где судьи выставляют оценки каждому спортсмену и таким образом выявляют лучшего и в других подобных случаях.

В случае, когда математика и вычислительная техника не справляются с решением задачи, приходится прибегать к человеческому мнению – опросу экспертов в необходимой области. Это происходит когда:

1. Невозможно описать формулами постановку задачи и ход ее решения.
2. Задача слишком сложная для существующих вычислительных средств.
3. Необходимо выбрать единственное решение из паретооптимальных.

В таких случаях применяются экспертные технологии принятия решений, они позволяют использовать субъективные оценки, основанные на опыте и мнениях специалистов рассматриваемой предметной области.

Существует классификация методов экспертных оценок:

1. По стадии применения:
 - Методы организации работы группы экспертов. Применяются при подборе экспертов и определении формата их взаимодействия.
 - Методы обработки результатов экспертизы. Применяются при работе с результатами заключений экспертов для выработки на их основе определенного итога.
2. По количеству экспертов:
 - Индивидуальные – каждый эксперт предоставляет свое видение правильного решения.
 - Коллективные – эксперты выносят коллективное решение.
3. По разновидности:
 - Эмпирические (нематематические) – использующие разнообразные методы, не основанные на математике.

- Математические – основанные на математике, ее методах и теориях.

Процесс проведения экспертных оценок можно описать с помощью обобщенного алгоритма:

1. Принятие решения о необходимости проведения экспертного опроса и формулировка его цели.
2. Формирование рабочей группы по проведению опроса.
3. Разработка технического задания (ТЗ) на экспертный опрос.
4. Разработка подробного сценария (регламента) проведения, сбора экспертных мнений и анализа экспертных оценок.
5. Подбор экспертов в соответствии с их компетентностью.
6. Проведение экспертизы.
7. Обработка и анализ результатов.
8. Интерпретация результатов.
9. Выработка рекомендаций, оформление отчета, завершение работы.

Возможно
несколько
тур

Несколько этапов следует описать с большим вниманием, они приведены ниже.

Разработка регламентов

Существует несколько параметров, которые важно определить на этапе разработки регламента проведения экспертных оценок:

1. Количество туров экспертизы:

- Один – эксперты однократно выбирают подходящее, по их мнению, решение.
- Несколько – проводится несколько итераций экспертизы для корректировки оценки на каждой следующей, с учетом итогов предыдущей.
- Неопределенное количество (до нахождения компромиссного решения) – экспертизы повторяют до тех пор, пока все эксперты не сойдутся на каком-либо компромиссном решении.

2. Степень общения экспертов:

- Отсутствует – эксперты не имеют каналов общения, как следствие, их мнения не зависят друг от друга.

- Заочно анонимное – эксперты взаимодействуют заочно и анонимно, тем самым исключается воздействие авторитетов при обмене мнениями.
- Заочно личностное – эксперты взаимодействуют заочно и знают, с кем именно они общаются. При таком варианте может иметь воздействие авторитет экспертов, но он может быть эффективен при определенных обстоятельствах.
- Очное с ограничениями – эксперты имеют возможность взаимодействовать очно, однако только в установленном режиме.
- Очное без ограничений – эксперты имеют возможность взаимодействовать очно и в том формате, в котором сочтут нужным.

При разработке регламента сбора экспертных оценок необходимо учитывать:

1. Степень подключения экспертов к работе:

- Использование всех экспертов сразу.
- Поочередное подключение экспертов (например, постепенно подключая к работе более дорогостоящих/удаленных экспертов, если не удается выработать решение с меньшими затратами).

2. Степень рассогласованности мнений:

- Хорошо, если мнения экспертов не согласованы: удается получить большее количество различных вариантов, среди которых найдется верный.
- Плохо, если мнения экспертов не согласованы: отсутствует единство мнений, возможно эксперты некомпетентны или задача не точно сформулирована.

При разработке регламента интерпретации результатов экспертных оценок следует помнить о двух исключениях из правил, которые встречаются на практике:

1. «Догма согласованности» - говорит о том, что не обязательно большее количество экспертов выражают верное мнение, иногда мнение «диссидентов» является правильным.

2. «Догма одномерности» - говорит о том, что не все результаты можно выразить одним числом или упорядочиванием по единственному даже интегральному признаку.

Подбор экспертов

Для поиска решения с применением экспертных технологий необходимо задействовать в этом процессе компетентных в исследуемой предметной области людей. Другими словами, нужно провести подбор экспертов. Он включает в себя два этапа, на каждом из которых возможны различные принципы организации:

1. Составление списка возможных экспертов:

- Самостоятельное составление реестра экспертов.
- Набор по принципу «снежного кома» - каждый приглашенный эксперт рекомендует еще нескольких.

2. Выбор экспертов необходимой компетенции (в зависимости от сложности задачи, бюджета и других факторов):

- По результатам научной деятельности. В данном случае необходимы эксперты для оценки деятельности других экспертов.
- Самооценка экспертов. При таком подходе существует риск необъективной, то есть заниженной или завышенной самооценки.
- Оценка друг друга. В данном случае возможно возникновение проблемы «клановости» (рис. 19.1).



Рис. 19.1. Опасность «клановости»

Перейдем к методам, используемым при применении экспертных технологий принятия решений.

Нематематические методы

Прежде чем мы перейдем к рассмотрению математических методов следует отметить, что существуют нематематические методы принятия решений:

1. Метод «Дельфи»
2. Метод сценариев
3. Метод ассоциаций
4. Метод фокальных объектов
5. Экспертное интервью
6. Ситуационный анализ
7. Мозговой штурм

Они приведены для общего ознакомления, однако прежде всего нас интересуют математические методы принятия решений при использовании экспертных оценок

Математические методы обработки результатов экспертных оценок

Существует несколько семейств математических методов обработки экспертных оценок:

1. Назначение баллов. Применяется, когда можно сказать, что один вариант лучше другого и оценить насколько. Пример приведен в таблице 19.1.

Таблица 19.1

Назначение баллов

	Экс. 1	Экс. 2	Экс. 3
Вар. 1	10	9	10
Вар. 2	8	8	8
Вар. 3	7	5	9
Вар. 4	3	2	4

2. Ранжирование. Применяется, когда можно сказать, что один вариант лучше другого, но нельзя оценить насколько. В таблице 19.2 отражено, как может выглядеть результат ранжирования.

Таблица 19.2

Ранжирование

	Экс. 1	Экс. 2	Экс. 3
Вар. 1	1	1	1
Вар. 2	2	2	3
Вар. 3	3	3	2
Вар. 4	4	4	4

3. Попарные сравнения (бинарные отношения). Применяется, когда невозможно ранжировать все варианты по порядку, но можно попарно сравнить их между собой. Например, как в таблице 19.3.

Таблица 19.3

Попарные сравнения

Вар. 1	>	Вар. 2		Вар. 2	>	Вар. 3		Вар. 3		Вар. 4	<	Вар. 1
--------	---	--------	--	--------	---	--------	--	--------	--	--------	---	--------

Метод средних баллов

Наиболее простой метод. Позволяет сравнивать объекты, назначая каждому из них определенное количество баллов по заданной шкале измерений. Подходит для сравнения простых объектов, для которых можно сказать, что один вариант лучше другого и указать количественно на сколько.

Пример. Фокус-группа из 7 экспертов выбирает лучшее мороженое из 5 видов (таблица 19.4):

Таблица 19.4

Метод средних баллов

	Эксп. 1	Эксп. 2	Эксп. 3	Эксп. 4	Эксп. 5	Эксп. 6	Эксп. 7
Мороженое 1	10	9	10	8	10	9	8
Мороженое 2	8	8	8	7	6	5	9
Мороженое 3	7	5	9	7	3	4	7
Мороженое 4	3	2	4	5	5	4	6
Мороженое 5	1	1	3	6	2	4	3

Обработка результатов проводится методами математической статистики:
Среднее количество баллов образца – математическое ожидание выборки:

$$M = \frac{1}{n} \sum_{i=1}^n x_i$$

Данный показатель отражает средний балл, который получил образец с учетом оценок всех экспертов.

Степень рассогласованности мнений экспертов – дисперсия по выборке:

$$D = \frac{1}{n-1} \sum_{i=1}^n (x_i - M)^2$$

Если дисперсия маленькая – мнения экспертов согласованы, в противном случае, их мнения на счет оцениваемых предметов сильно отличаются. С помощью этого показателя можно определить, насколько правильно подобраны эксперты и верная ли перед ними поставлена задача.

Интервальная оценка баллов – доверительный интервал, например $\pm 2\sigma$:

$$[M - 2\sigma; M + 2\sigma], \text{ где } \sigma = \sqrt{D} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - M)^2}$$

Интервальная оценка отражает, насколько сильно разбросаны мнения экспертов, относительно среднего значения. На основе этой информации можно судить, корректно ли в итоге оценены сравниваемые объекты.

В таблице 19.5 приведены расчеты на основе данных методов для описанного ранее примера:

Таблица 19.5

Метод средних баллов – расчеты

	Средний балл	Рассогласованность мнений	Интервальная оценка баллов
Мороженое 1	9,1	0,8	7,3 ... 10,9
Мороженое 2	7,3	1,9	4,5 ... 10,0
Мороженое 3	6,0	4,3	1,8 ... 10,2
Мороженое 4	4,1	1,8	1,5 ... 6,8
Мороженое 5	2,9	3,1	-0,7 ... 6,4

На рис. 19.2 мы видим, что средние баллы распределились практически равномерно от 1 места к 5. Но при этом относительно 1, 2 и 4 места мнения экспертов практически совпадали, а относительно 3 и 5 мест очень сильно различались, что понижает степень достоверности. Интервальная оценка показывает, что 2 – 3 – 4 места недостаточно однозначно оценены и могут менять местами, например, в зависимости от набора экспертов. Верхняя граница 4 – 5 мест также очень близки. Для повышения степени достоверности желательна повторная экспертиза альтернативными методами.

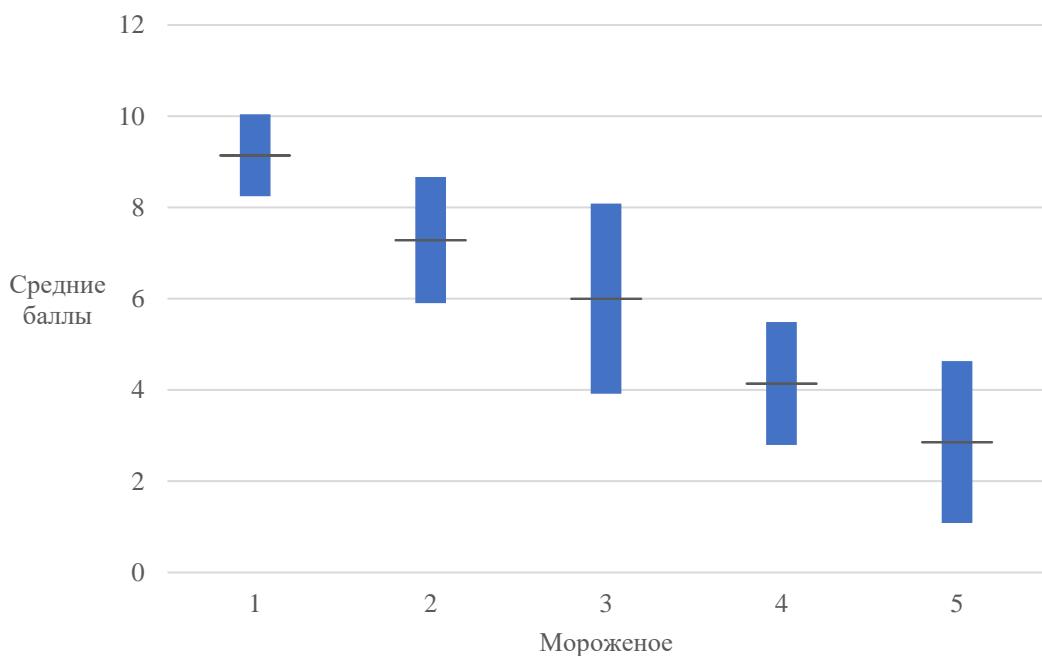


Рис. 19.2. Метод средних баллов

Метод медианных рангов

Применяется в том случае, если невозможно назначить баллы, но можно расставить по рангу – отранжировать объекты. Подходит для сравнения сложных объектов, для которых можно сказать, что один вариант лучше другого, но нельзя указать количественно на сколько.

Пример. Выборы главы государства и отдельных городов в Австралии, Ирландии, Индии, Мальте и др. (преференциальное голосование) – таблица 19.6.

Таблица 19.6

Метод медианных рангов

	Эксп. 1	Эксп. 2	Эксп. 3	Эксп. 4	Эксп. 5	Эксп. 6	Эксп. 7	...
Кандидат 1	2	2	1	2	2	1	1	...
Кандидат 2	1	1	2	1	1	4	2	...
Кандидат 3	3	3	4	3	3	2	3	...
Кандидат 4	4	4	3	4	4	3	4	...

Медиана (Me) – это некоторая отметка, делящая ранжированные данные (отсортированные по возрастанию или убыванию) на две равные части. Половина исходных данных меньше этой отметки, а половина – больше. Расчет медианы представлен в таблице 19.7. Наглядно медиана изображена на рис. 19.3.

Таблица 19.7

Вычисление медианы

Исходные данные:	1	2	3	1	5	4	6	1	4
Вычисление медианы:	1	1	1	2	3	4	4	5	6



Рис. 19.3. Медиана

Медиана, в отличие от матожидания, устойчива к аномальным отклонениям (рис. 19.4).



Рис. 19.4. Сравнение медианы и среднего арифметического

Найдем медиану и матожидание выборки преференциального голосования:

Таблица 19.9

Вычисление матожидания для примера

				Me				M
Кандидат 1	1	1	1	2	2	2	2	1,6
Кандидат 2	1	1	1	1	2	2	4	1,7
Кандидат 3	2	3	3	3	3	3	4	3,0
Кандидат 4	3	3	4	4	4	4	4	3,7

Большинство выборщиков (экспертов) поставило второго кандидата на 1 место, но за счет наличия «диссидентов», поставившего его на 4 место, средний ранг M опустился на 2 место. При этом медианный ранг Me остался на 1 месте, так как он не подвержен случайным выбросам и отражает мнение большинства.

Метод средневзвешенных рангов

Средневзвешенные ранги определяются с учетом компетентности всех экспертов, которые рассчитываются по следующему алгоритму:

1. Определяется среднее арифметическое рангов кандидатов (матожидание):

$$M = \frac{1}{n} \sum_{i=1}^n x_i$$

Среднее арифметическое рангов (матожидание) кандидатов представлено в таблице 19.10.

Таблица 19.10

Вычисление матожидания

	Эксп. 1	Эксп. 2	Эксп. 3	Эксп. 4	Эксп. 5	Эксп. 6	Эксп. 7	M
Кандидат 1	2	2	1	2	2	1	1	1,6
Кандидат 2	1	1	2	1	1	4	2	1,7
Кандидат 3	3	3	4	3	3	2	3	3,0
Кандидат 4	4	4	3	4	4	3	4	3,7

2. По каждому эксперту рассчитывается степень отклонения его ответа от среднего значения M – то есть дисперсия мнений каждого эксперта:

$$D = \frac{1}{n-1} \sum_{i=1}^n (x_i - M)^2$$

Отклонение представлено в таблице 19.11.

Таблица 19.11

Дисперсия по экспертам

D эксперта	0,3	0,3	0,6	0,3	0,3	2,4	0,2
------------	-----	-----	-----	-----	-----	-----	-----

3. Чем больше дисперсия, тем сильнее мнение эксперта отличается от мнения большинства. Следовательно, его компетентность в данном вопросе можно считать ниже, чем у коллег (исключая догму «согласованности»).

Компетентность эксперта K будет обратно пропорциональна его дисперсии D :

$$K = \frac{1}{D}$$

Компетентность эксперта представлена в таблице 19.12.

Таблица 19.12

Компетентность экспертов

k эксперта	3,9	3,9	1,6	3,9	3,9	0,4	6,1	23,6
------------	-----	-----	-----	-----	-----	-----	-----	------

4. Выразим значимость мнения каждого эксперта в процентах (нормируем) и приведем его к виду коэффициента компетентности k :

$$k = \frac{K}{\sum K}$$

Значимость мнений экспертов представлена в таблице 19.13.

Таблица 19.13

Компетентность экспертов в процентах

k эксперта	16%	16%	7%	16%	16%	2%	26%	100%
--------------	-----	-----	----	-----	-----	----	-----	------

5. Далее найдем средневзвешенное мнение экспертов относительно кандидатов с учетом коэффициента их компетентности (важности их мнения) (таблица 19.14):

$$Mk = k_1x_1 + k_2x_2 + k_3x_3 + \cdots + k_nx_n$$

Таблица 19.14

Средневзвешенное мнение экспертов

	Эксп. 1	Эксп. 2	Эксп. 3	Эксп. 4	Эксп. 5	Эксп. 6	Эксп. 7	Mk
Кандидат 1	2	2	1	2	2	1	1	1,7
Кандидат 2	1	1	2	1	1	4	2	1,4
Кандидат 3	3	3	4	3	3	2	3	3,0
Кандидат 4	4	4	3	4	4	3	4	3,9
K эксперта	16%	16%	7%	16%	16%	2%	26%	

Характерно, что средневзвешенный ранг Mk дал более точный результат, чем среднеарифметической ранг M . Он также поставил второго кандидата на 1 место, как и медианна рангов Me (таблица 19.15).

Таблица 19.15

Результаты оценки

Me	M
2	1,6
1	1,7
3	3,0
4	3,7

Бинарные отношения

Метод попарных сравнений - один из наиболее сложных методов.

Применяется в случаях, когда объекты настолько комплексные и многомерные, что нет возможности отранжировать их по одному признаку, но есть возможность попарно сравнить их между собой. Каждый эксперт должен сравнить каждую пару вариантов друг с другом. Результат записывается в виде матриц бинарных отношений (таблица 19.16). Принцип применения метода, следующий: например, имеется два объекта с условными названиями a и b и метрика $[0, 1]$. В случае, если объект a лучше или равен объекту b , метрика приобретает значение 1, в противном случае – значение 0:

$$a \geq b \quad x = 1$$

$$a < b \quad x = 0$$

Если сравниваемых объектов несколько a_1, a_2, \dots, a_n , то результатом их попарных сравнений будет являться матрица бинарных отношений:

$$\|X_{ij}\| = \begin{cases} 1, & a_i \geq a_j \\ 0, & a_i < a_j \end{cases}$$

Таблица 19.16

Матрица бинарных отношений

	a_1	a_2	a_3	a_4
a_1	1	0	1	1
a_2	1	1	1	1
a_3	0	0	1	1
a_4	0	0	0	1

Пример. Выбор места строительства крупного инфраструктурного объекта. Результатом сравнения между собой четырех инфраструктурных объектов тремя экспертами будут следующие матрицы (таблица 19.17):

Таблица 19.9

Матрицы бинарных отношений для примера

Эксперт 1				
	П1	П2	П3	П4
П1	1	1	1	1
П2	0	1	1	1
П3	0	0	1	1
П4	0	0	0	1

Эксперт 2				
	П1	П2	П3	П4
П1	1	0	1	1
П2	1	1	1	1
П3	0	0	1	0
П4	0	0	1	1

Эксперт 3				
	П1	П2	П3	П4
П1	1	0	1	1
П2	1	1	1	1
П3	0	0	1	0
П4	0	0	1	1

Теперь необходимо найти некоторое усредненное значение мнений экспертов. Для того, чтобы сделать это на матрицах введем несколько новых понятий.

Определение. Расстоянием Кемени между бинарными отношениями X и Y называется число, характеризующее количество несовпадающих элементов матриц X и Y :

$$D(X, Y) = \sum_{i=1}^n \sum_{j=1}^n |x_{ij} - y_{ij}|$$

Возвращаясь к примеру, расстояние Кемени между мнениями эксперта 1 и эксперта 2 рассчитывается как вычитание матриц их бинарных отношений, взятие модуля и сложение элементов получившейся матрицы. Как мы видим из таблицы 13.17 мнения первого и второго экспертов отличается по 2 позициям. Так как матрица симметрична относительно главной диагонали, то складываются и «зеркальные» отличия ниже главной диагонали. Таким образом расстояние Кемени равно $D = 4$ (таблица 19.18).

Таблица 19.18

Расчет расстояния Кемени для примера

	П1	П2	П3	П4	—		П1	П2	П3	П4	=	0	1	0	0
П1	1	1	1	1		П1	1	0	1	1		1	0	0	0
П2	0	1	1	1		П2	1	1	1	1		0	0	0	1
П3	0	0	1	1		П3	0	0	1	0		0	0	1	0
П4	0	0	0	1		П4	0	0	1	1		0	0	1	0
															Σ
															$D = 4$

Определение. Медианой Кемени называется такая матрица A , суммарное расстояние Кемени от которой до всех остальных матриц X_1, X_2, \dots, X_m является минимальным:

$$\arg \min_A \sum_{i=1}^m D(A, X_i)$$

Медиана Кемени является той самой усредненной матрицей, которую мы задались найти. Она обобщает мнение всех экспертов следующим образом.

Как известно из геометрии, чтобы объект находился в центре, сумма расстояний от него до всех остальных объектов должна быть минимальна (рис. 19.5).

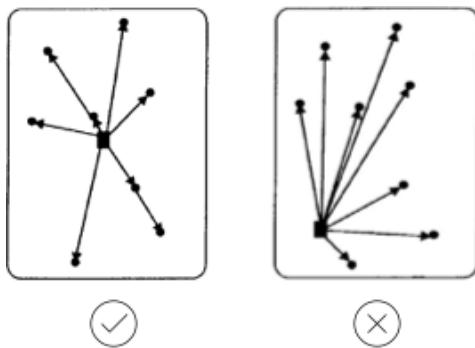


Рис. 19.5. Расстояние между объектами

Аналогичное правило применимо и к матрицам, только используется не геометрическое расстояние, а расстояние Кемени (рис. 19.6).

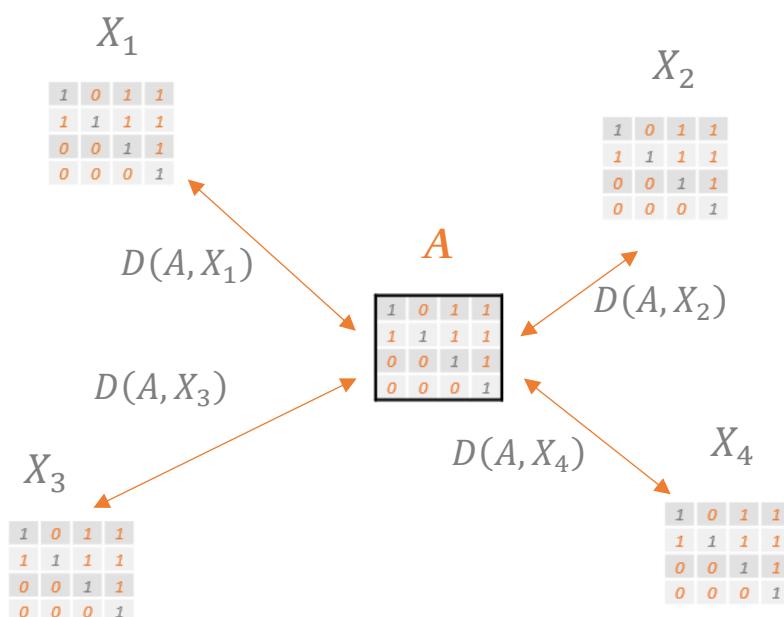


Рис. 19.6. Расстояние Кемени между матрицами

Таким образом, медиана Кемени A является той самой матрицей, которая находится на «геометрической середине» по отношению ко всем остальным матрицам X_1, X_2, X_3, X_4 и за счет этого является усредненным мнением всех экспертов.

Найти медиану Кемени можно с помощью **задачи нелинейного программирования:**

Исходные данные: матрицы бинарных отношений X_1, X_2, \dots, X_m .

Искомая переменная: медиана Кемени A .

Целевая функция – минимальное суммарное расстояние до всех матриц:

$$\sum_{i=1}^m D(A, X_i) \rightarrow \min$$

Ограничения:

$$A \in \{0,1\}$$

$$\text{Маска } A = \begin{vmatrix} 1 & a & b & \dots \\ \neg a & 1 & c & \dots \\ \neg b & \neg c & 1 & \dots \\ \dots & \dots & \dots & \dots \end{vmatrix}$$

Решая задачу нелинейного программирования одним из известных методов, например, методом градиента, описанным в разделе ___, получаем значения результирующей матрицы A . В нашем примере медиана Кемени A совпала с матрицей эксперта 3 (таблица 19.19):

Таблица 19.19

Медиана Кемени

Эксперт 1					Эксперт 3					Эксперт 2				
	П1	П2	П3	П4		П1	П2	П3	П4		П1	П2	П3	П4
П1	1	1	1	1		1	0	1	1		1	0	1	1
П2	0	1	1	1		1	1	1	1		1	1	1	1
П3	0	0	1	1		0	0	1	1		0	0	1	0
П4	0	0	0	1		0	0	0	1		0	0	1	1

$D = 2$ $D = 2$ $D = 4$

Матрица А

Медиана Кемени (усредненное мнение экспертов) совпало с мнением эксперта 3, поскольку расстояние от его матрицы до двух других в сумме равно $2+2$, а расстояние от матриц экспертов 1 и 3 до других равно $2+4 = 6$. Таким образом, мнение эксперта 3 можно принимать как усредненное мнение всех членов экспертной группы.

Полученный результат можно **интерпретировать в ранговый вид** (таблица 19.20), расставив предприятия в порядке выгодности для покупки активов путем суммирования элементов медианы Кемени (матрицы A) по строкам:

Таблица 19.20

Интерпретация медианы Кемени в ранговый вид

	П1	П2	П3	П4			Ранги
П1	1	0	1	1		Положение 1	3
П2	1	1	1	0		Положение 2	3
П3	0	0	1	1		Положение 3	2
П4	0	1	0	1		Положение 4	2

Таким образом, в методе бинарных отношений фокус внимания эксперта сужен до решения простой задачи – какой из двух вариантов лучше. Это сделать проще, чем охватить область всех вариантов, еще и назначив каждому из них баллы или ранги.

Поэтому метод бинарных отношений дает более точные ответы, когда множество альтернатив велико или используются сложные объекты, но трудозатрачен в обработке.



Джон Джордж Кемени (Янош Дьёрдь Кемень; 31.05.1926, Будапешт, Венгрия - 26.12.1992, Нью-Гэмпшир, США) — американский математик и специалист по информатике. После начала войны эмигрировал в Нью-Йорк в январе 1940 года. Учился в Принстонском университете с 1943 г., участвовал в Манхэттенском проекте, где познакомился с Джоном фон Нейманом. Был консультантом у Альберта Эйнштейна по математическим вопросам. Вместе с Томасом Курцем разработал язык Бейсик в 1964 г. В 1970 году был избран 13-м президентом Дартмутского колледжа (одного из старейших и авторитетнейших университетов США) и занимал эту должность 11 лет. В этом колледже он впервые начал систематическое применение компьютеров в учебном процессе.

20. Системы массового обслуживания

Данный раздел является частью теории массового обслуживания (теории очередей) (queueing theory) – раздела математической экономики, занимающийся вопросами изучения моделирования и оптимизации систем массового обслуживания с большим количеством клиентов, вне зависимости от их возникновения.

Рассмотрим несколько типов систем массового обслуживания. К первому типу можно отнести организации социального обслуживания – супермаркеты, банки и больницы. Очереди здесь состоят из людей, которым необходимо получить ту или иную услугу. Ко второму типу относятся социальные сети и мессенджеры. Очередь находится на сервере и состоит только из обращений клиентов к данному серверу. В данном случае обращения могут иметь несколько статусов (в зависимости от используемого приложения) – сообщение отправлено, получено, прочитано. Третий тип систем массового обслуживания – такси и общественный транспорт. В случае рассмотрения такси очередь является очередью заявок на получение услуг такси. Она распределяется между таксистами, которые из предложенного списка ожидающих клиентов выбирают одного. В случае же с общественным транспортом очередь может выстраиваться на остановке в ожидании необходимого транспортного средства. К четвертому типу относятся колл-центры и экстренные службы. В данном типе очередь состоит из двух последовательных очередей – очередь на обработку заявки оператором и очередь на обработку заявки необходимой службой (например, пожарным отделением, где имеется свой набор обрабатываемых заявок). В случае с таким типом систем массового обслуживания особенно важно уменьшить время ожидания клиента на линии и обработки его заявки.

В качестве примера рассмотрим маленький магазин, где имеется 3 кассы. По результатам наблюдения за посетителями, в утреннее время с 9 до 12 часов магазин посещают 50 покупателей, в обеденное время с 12 до 15 часов – 300 покупателей, днем с 15 до 18 часов – 150 покупателей, а вечером с 18 до 21 часа – 500 покупателей. Среднее время обслуживания одного клиента на кассе

составляет 1,5 минуты. Одновременно в очередь на все кассы в сумме помещается не более 9 человек (если в очереди уже находится 9 человек, то 10 пришедший либо покинет магазин из-за нежелания долго стоять в очереди, либо уйдет к конкурентам в соседний магазин). Необходимо найти нагрузку кассы магазина и оптимизировать систему обслуживания.

Структура системы массового обслуживания

Математическая модель для всех систем массового обслуживания имеет одинаковую структуру (рис. 20.1). Ее компонентами являются:

- Поступающие новые заявки (входящие покупатели)
- Ожидающие в очереди заявки (очередь на кассу)
- Обслуживающие устройства (кассы)
- Обслуженные исходящие заявки (выходящие с покупками)

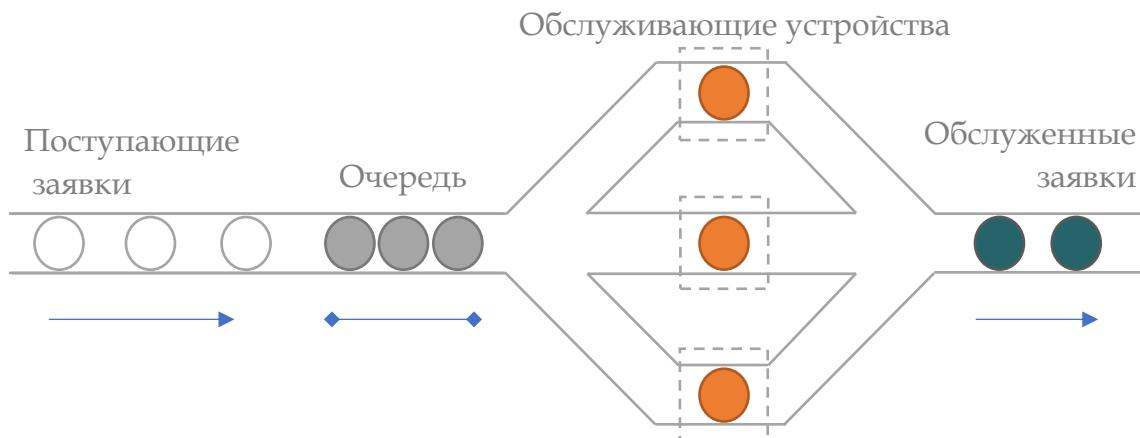


Рис. 20.1. Структура системы массового обслуживания

Потоки заявок

Входной поток заявок, представленный на рис. 20.2, характеризуется скоростью поступления заявок в систему (интенсивностью входного потока заявок). Данный показатель показывает сколько заявок приходит в систему за определенный интервал времени. Если известен средний интервал времени между потоком заявок, то используется 2 формула. Если нет, то необходимо определить количество заявок, пришедшее за определенный период времени.

$$\lambda = \frac{N}{t} = \frac{50 \text{ чел}}{180 \text{ мин}} = 0,28 \text{ мин}^{-1}$$

$$\lambda = \frac{1}{t_{\text{вх}}}$$

где N – количество заявок, пришедшее за период времени t , $t_{\text{вх}}$ – средний интервал времени между поступлением заявок.

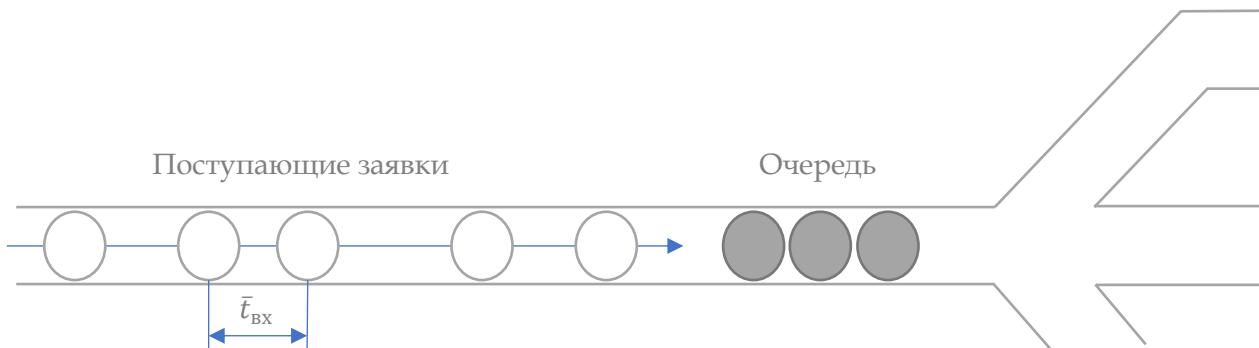


Рис. 20.2. Входной поток заявок

Применяя формулу к вышесказанному условию задачи, получаем данные, приведенные в таблице 20.1.

Таблица 20.1

Данные по времени суток

	Утро	Обед	День	Вечер
$\lambda, \text{мин}^{-1}$	0,28	1,67	0,83	2,78

Выходной поток заявок, представленный на рис. 20.3, характеризуется скоростью обработки заявок системой (интенсивностью выходного потока заявок от одного прибора). Данный показатель показывает сколько заявок обрабатывает один прибор за единицу времени и рассчитывается по формуле. Если известно среднее время, необходимое на обработку 1 заявки одним прибором, то используется 2 формула. Если нет, то необходимо определить количество заявок, обработанное за период времени некоторым количеством приборов.

$$\mu = \frac{N}{nt}$$

$$\mu = \frac{1}{t_{\text{вых}}} = \frac{1}{1,5 \text{ мин}} = 0,66 \text{ мин}^{-1}$$

где N – количество заявок, обработанное за период времени t всеми n приборами, $t_{\text{вых}}$ – среднее время на обработку одной заявки одним прибором.

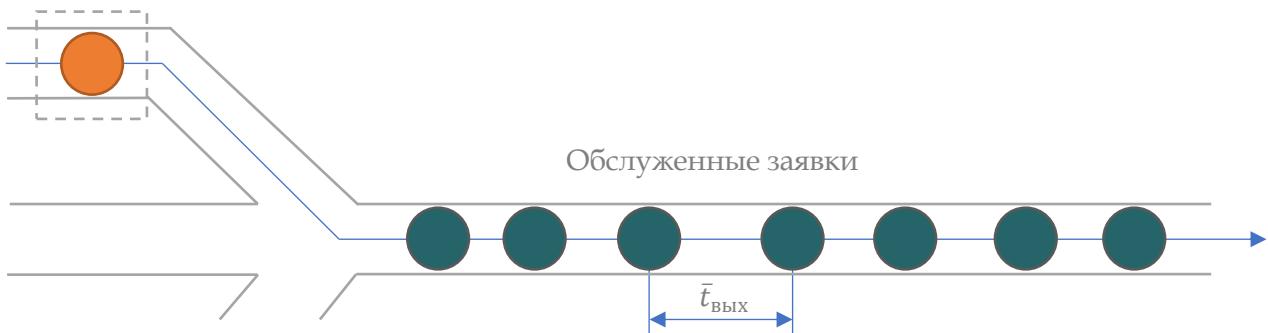


Рис. 20.3. Поток обслуженных заявок

Применяя формулу к вышесказанному условию задачи, получаем данные, приведенные в таблице 20.2.

Таблица 20.2

Данные по времени суток

	Утро	Обед	День	Вечер
$\mu, \text{мин}^{-1}$	0,66	0,66	0,66	0,66

Чтобы узнать количество покупателей, находящихся на кассах и в очереди, необходимо построить граф состояний системы. Обозначим за вершины графа состояния системы (рис. 20.4). Первая вершина соответствует состоянию S_0 , когда покупателей в магазине нет и система бездействует. Вторая, третья и четвертая вершины соответствуют состояниям S_1 , S_2 и S_3 и показывает, что на кассах суммарно находятся один, два и тря человека соответственно, но очереди в магазине еще нет. С пятой вершины графа начинается состояние очереди, то есть все кассы заняты, а в очереди находится один человек. Подобным образом происходит, пока в общей очереди не будет находиться 9 человек. То есть последний график будет иметь состояние S_{12} – в каждой кассе находится по человеку, а в очереди стоит 9 покупателей.

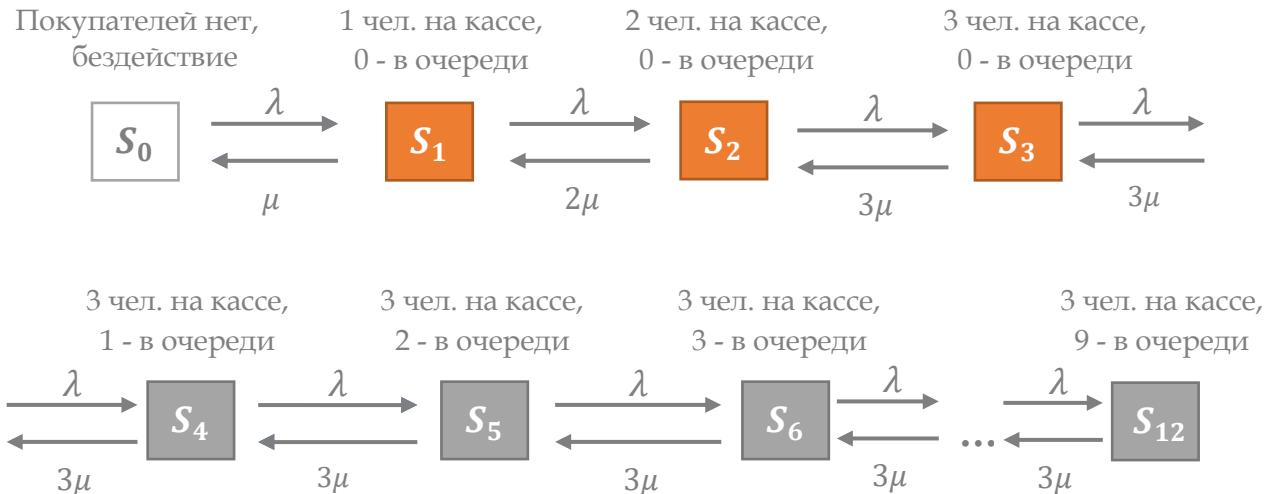


Рис. 20.4. Граф состояний системы

Дуги же будут показывать интенсивности перехода между состояниями. Перемещение по графу в сторону увеличения количества человек происходит с интенсивностью λ (то есть λ раз в минуту в магазин заходит один покупатель). При этом существует и перемещение по графу в сторону уменьшения количества человек в магазине – в среднем 3μ раз за минуту с кассы уходит один покупатель. Стоит отметить, что между состояниями S_2 , S_1 и S_0 перемещение происходит с интенсивностью 2μ и μ соответственно. Происходит это по причине того, что в состоянии S_2 работает две кассы и из них выходят 2 человека, а в состоянии S_1 работает одна касса и магазин покидает один человек, обслуженный на этой кассе.

Вероятности состояний системы

Следующим шагом рассмотрим состояния системы. Как показывает практика система будет функционировать так, как представлено на рис. 20.5. Каждый синий прямоугольник показывает обработку клиента – от его прихода до ухода из магазина. Стоит отметить, что длина прямоугольников разная, так как процесс обработки заявок является случайным процессом (у каждого покупателя разная по величине корзина покупок).

Если сгруппировать все участки и посчитать в сумме какое количество времени система находилась в каждом из состояний, то представляется возможным рассчитать вероятность нахождения системы в состояниях от S_0 до S_3 (то есть с

какой вероятностью загружены 1, 2 или 3 кассы, или же они простоявают без покупателей). Эти вероятности обозначены соответственно p_0 , p_1 , p_2 , p_3 и в сумме составляют 100% (рис. 20.6).

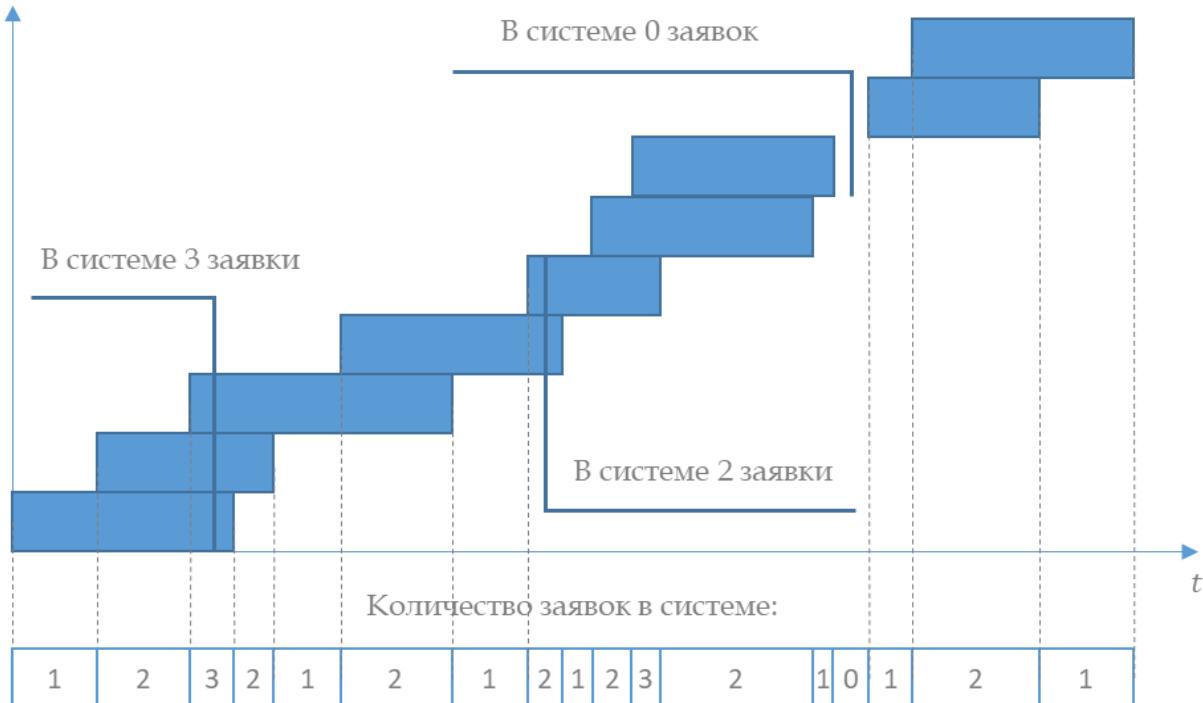


Рис. 20.5. Вероятности состояний системы



Рис. 20.6. Вероятности нахождения системы в различных состояниях

Рассмотрим каким же образом рассчитываются данные вероятности. Вероятность того, что в магазине нет покупателей и система бездействует, определяется по следующей формуле:

$$p_0 = \left[\sum_{k=0}^n \frac{\rho^k}{k!} + \sum_{k=n+1}^{n+m} \frac{\rho^k}{n^{k-n} n!} \right]^{-1}, \rho = \frac{\lambda}{\mu}$$

Для вероятностей наступления состояний обслуживания используется представленная ниже формула.

$$p_k = \frac{\rho^k}{k!} p_0, k = 1 \dots n$$

Для вероятностей наступления состояний очереди используются следующая формула:

$$p_k = \frac{\rho^k}{n^{k-n} n!} p_0, k = (n+1) \dots (n+m)$$

В рассмотренных выше формулах $n=3$ – количество касс обслуживания, $m=9$ – максимальная длина очереди.

Найдем соответствие различных состояний и соответствующие им вероятности наступления (рис. 20.7)

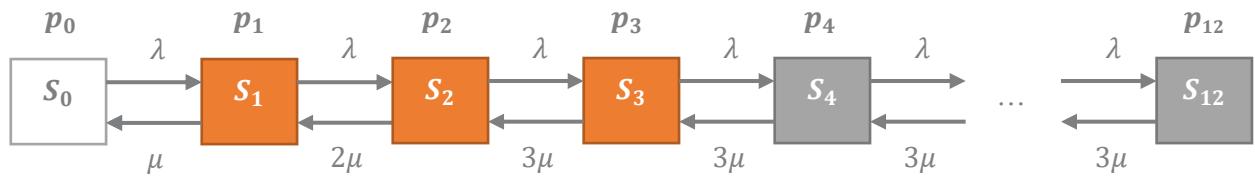


Рис. 20.7. Вероятности наступления различных состояний

Сделав расчёты, получим таблицу с вероятностями нахождения системы в разные промежутки времени в различных состояниях. (Таблица 20.3)

Таблица 20.3

Вероятности нахождения системы в разные промежутки времени

	p_0	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	\dots	p_{12}
Утро	66%	27%	6%	1%	0%	0%	0%	0%	0%	0%	...	0%
Обед	5%	13%	16%	13%	11%	9%	8%	6%	5%	4%	...	3%
День	28%	35%	22%	9%	4%	2%	1%	0%	0%	0%	...	0%
Вечер	0%	1%	1%	1%	2%	3%	4%	6%	8%	11%	...	29%

Визуализируем полученные вероятности с помощью графиков (рис. 20.8–20.11).

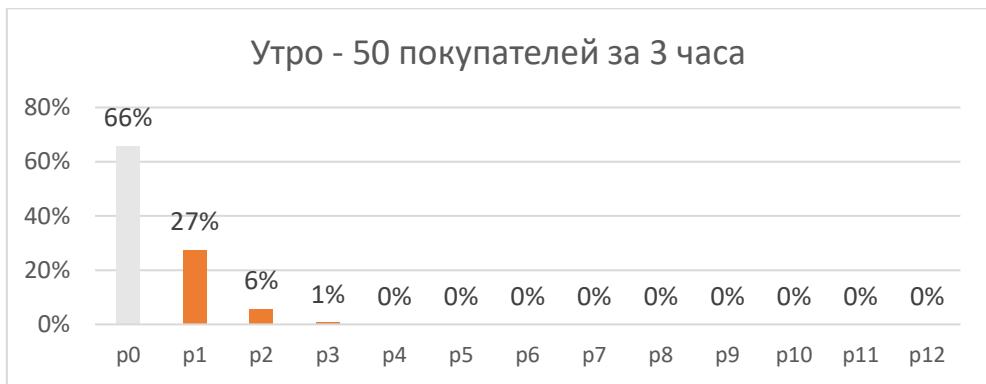


Рис. 20.8. Вероятности состояний системы утром

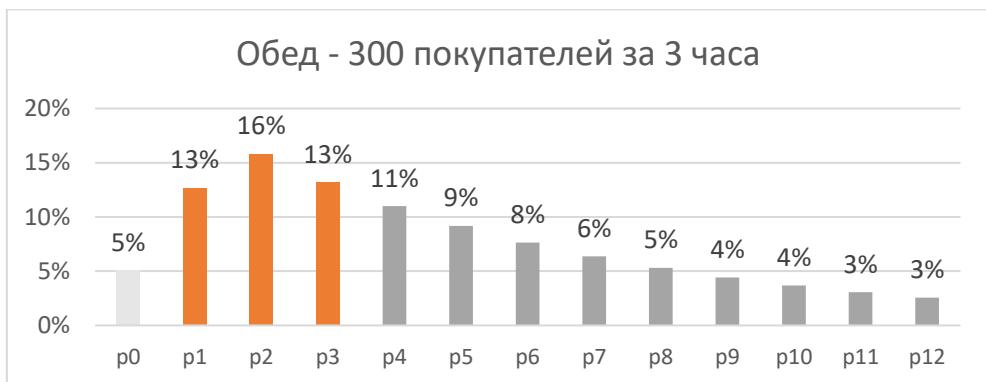


Рис. 20.9. Вероятности состояний системы в обед



Рис. 20.10. Вероятности состояний системы днем



Рис. 20.11. Вероятности состояний системы вечером

Показатель загруженности системы

Величина ρ является показателем загруженности системы и дает представление о том, насколько система справляется с потоком клиентов.

$$\rho = \frac{\lambda}{\mu}$$

- $\rho \ll n$ (показатель загруженности сильно меньше количества касс) – система недогружена, что выгодно для клиента, так как нет очередей, но невыгодна для владельца, так как есть лишние кассы и большой простой по времени;
- $\rho < n$ (показатель загруженности меньше количества касс) – система сбалансирована для клиента, так как есть приемлемые очереди и допустимый простой касс;
- $\rho \leq n$ – (показатель загруженности приближен к количеству касс) система сбалансирована для владельца, так как есть большие очереди клиентов, кассы заняты практически полностью;
- $\rho > n$ (показатель загруженности больше количества касс) – система перегружена, что выгодно для владельца, так как заявок больше, чем можно обработать, но невыгодно для клиента по причине бесконечно растущей очереди.

Для рассматриваемой задачи таблица значений ρ и n представлена в таблице 20.4

Таблица 20.4

Значения ρ и n

		Утро	Обед	День	Вечер
Показатель загруженности	ρ	0,42	2,5	1,25	4,16
Количество касс магазина	n	3	3	3	3

Визуализируем полученные данные с помощью графиков (рис. 20.12–20.15).

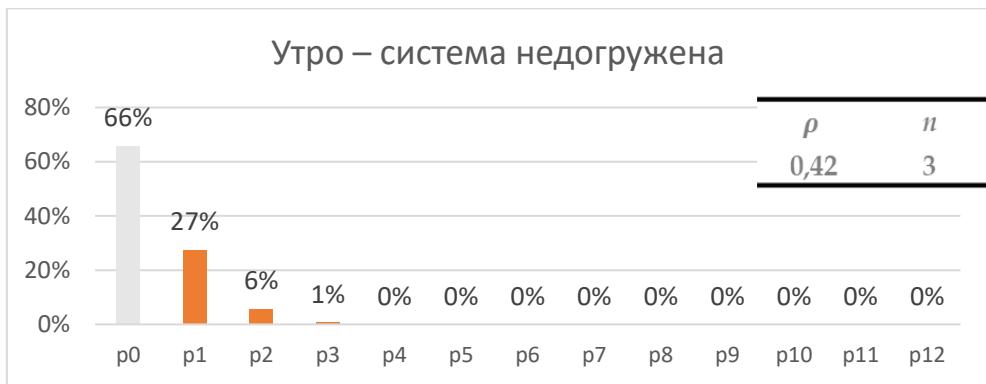


Рис. 20.12. Нагруженность системы утром

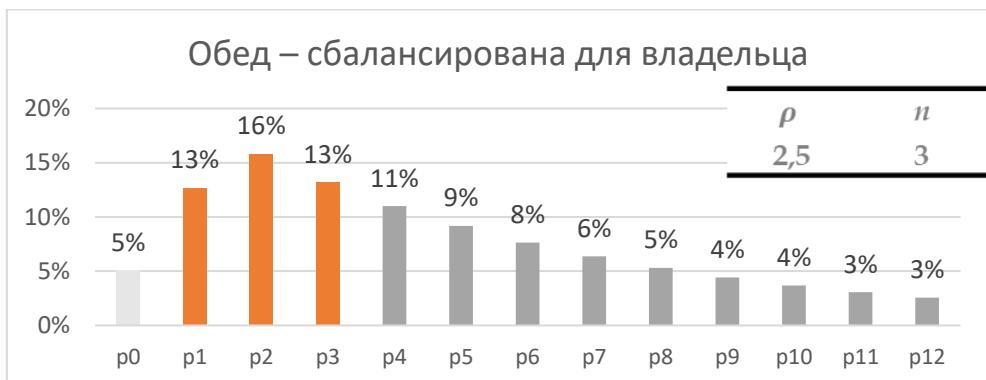


Рис. 20.13. Нагруженность системы в обед



Рис. 20.14. Нагруженность системы днем



Рис. 20.15. Нагруженность системы вечером

Таким образом, утром система недогружена (показатель загруженности сильно меньше количества касс), в обед сбалансирована для владельца (показатель загруженности приближен к количеству касс), днем сбалансирована для клиента (показатель загруженности меньше количества касс), а вечером перегружена (показатель загруженности больше количества касс).

Сбалансированность системы для владельца и клиента

Изменим количество касс так, чтобы система была сбалансированная для владельца (показатель загруженности приближен к количеству касс) (рис. 20.16–20.19). Тогда утром необходимо чтобы работала одна касса, днем – две, в обед – три, а вечером – пять касс.

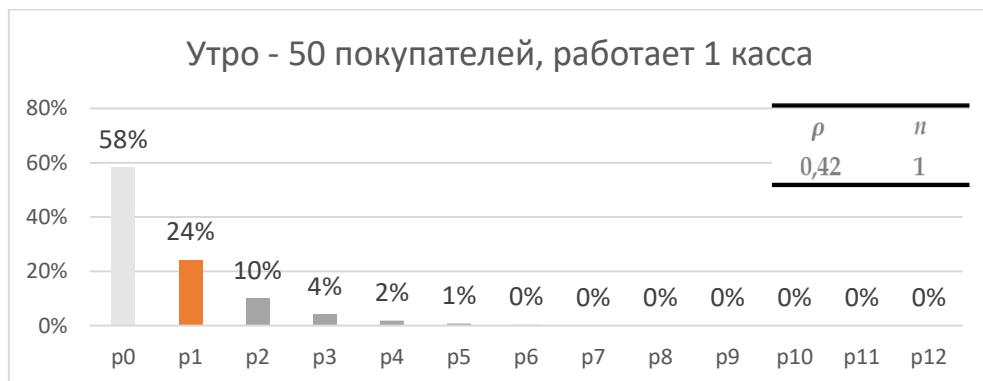


Рис. 20.16. Система, сбалансированная для владельца, утром

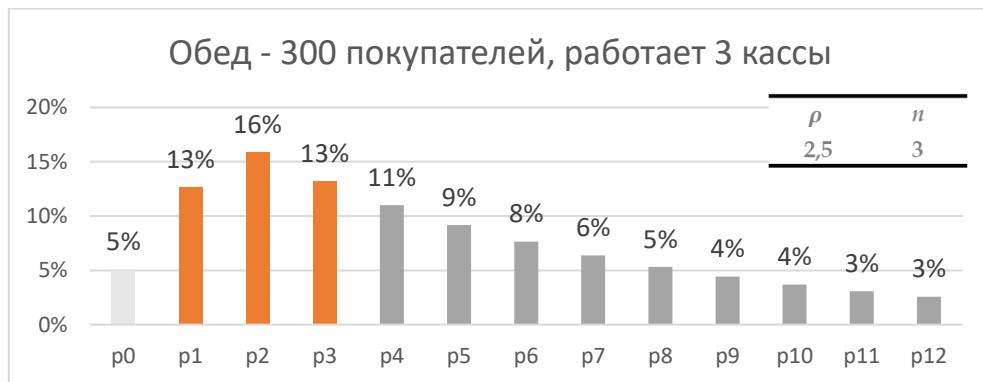


Рис. 20.17. Система, сбалансированная для владельца, в обед



Рис. 20.18. Система, сбалансированная для владельца, днем

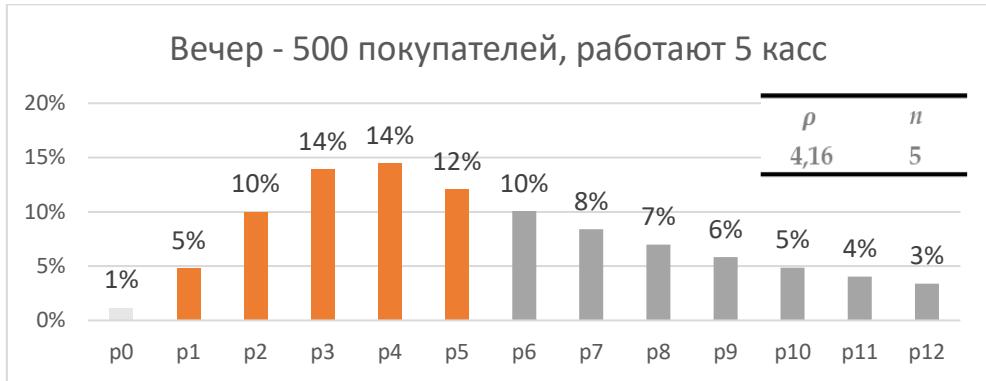


Рис. 20.19. Система, сбалансированная для владельца, вечером

Изменим количество касс так, чтобы система была сбалансированная для клиента (показатель загруженности меньше количества касс) (рис. 20.20–20.23). Тогда утром необходимо чтобы работало две кассы, днем – четыре, в обед – шесть, а вечером – девять касс. При таком количестве касс клиент практически не будет вставать в очередь.

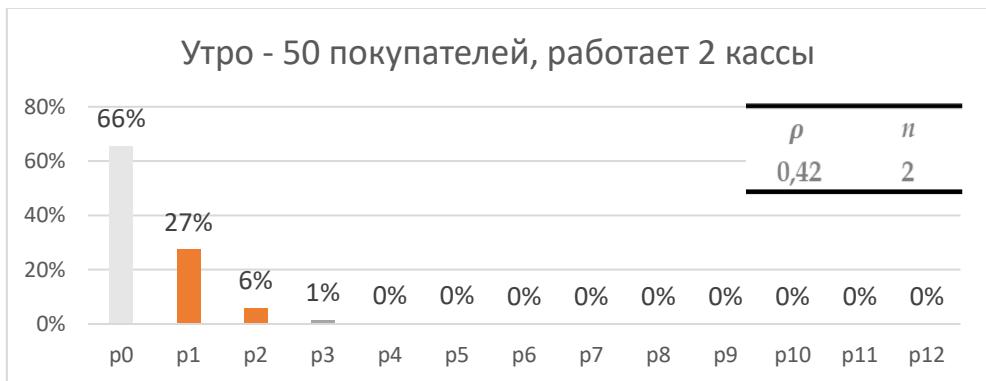


Рис. 20.20. Система, сбалансированная для клиента, утром



Рис. 20.21. Система, сбалансированная для клиента, в обед



Рис. 20.22. Система, сбалансированная для клиента, днем



Рис. 20.23. Система, сбалансированная для клиента, вечером

Показатели для клиентов и владельцев

Рассмотрим характеристики системы, важные для клиентов.

Вероятность отказа в обслуживании показывает вероятность, что новая заявка не помещается в систему (заняты все приборы и заполнена вся очередь – последняя «правая» вероятность):

$$p_{\text{отк}} = p_{n+m} = p_{12}$$

Вероятность встать в очередь показывает вероятность того, что новая заявка встречает в системе очередь (заняты все приборы и есть место в очереди - сумма всех вероятностей очередей):

$$p_{\text{оч}} = \sum_{k=n+1}^{n+m-1} p_k = p_4 + \cdots + p_{11}$$

Средняя длина очереди показывает среднее количество заявок, ожидающих в очереди:

$$L_{\text{оч}} = \frac{\rho^{n+1}}{n \cdot n!} \cdot \frac{1 - (\rho/n)^m (m + 1 - m \rho/n)}{(1 - \rho/n)^2} \cdot p_0$$

Среднее время ожидания в очереди рассчитывается по формуле Литтла:

$$T_{\text{оч}} = \frac{L_{\text{оч}}}{\lambda (1 - p_{\text{отк}})}$$

В рамках рассматриваемой задачи показатели для клиентов относительно времени суток будут выглядеть следующим образом (рис. 20.24–20.27).

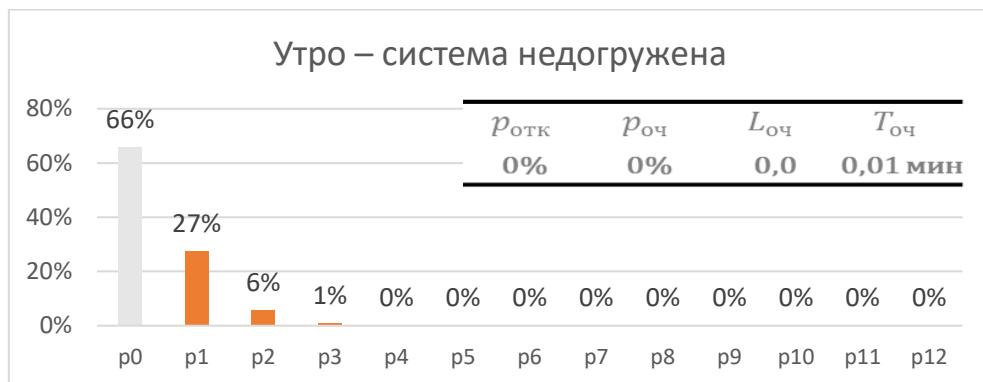


Рис. 20.24. Показатели для клиентов утром

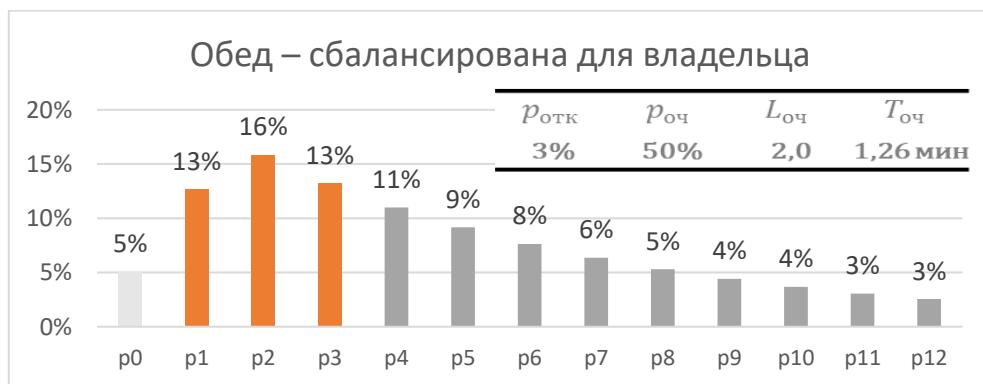


Рис. 20.25. Показатели для клиентов в обед

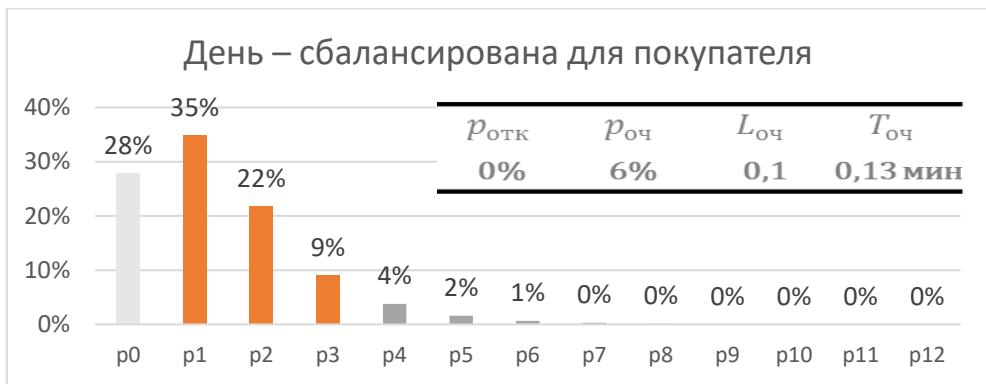


Рис. 20.26. Показатели для клиентов днем

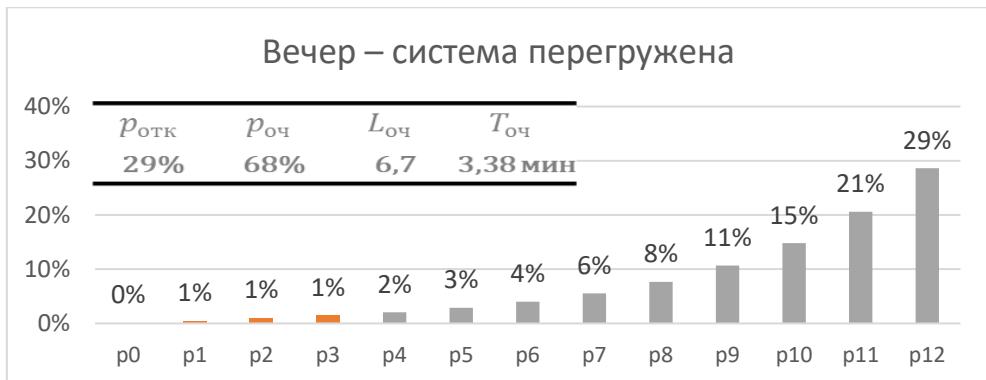


Рис. 20.27. Показатели для клиентов вечером

Рассмотрим характеристики системы, важные для владельцев.

Абсолютная пропускная способность показывает скорость обслуживания заявок (сколько заявок успевает обрабатываться системой в единицу времени):

$$A = \lambda (1 - p_{отк})$$

Относительная пропускная способность показывает процент обслуженных заявок (какой процент заявок не успевает обрабатываться и получает отказ):

$$Q = 1 - p_{отк}$$

Среднее количество занятых приборов:

$$\bar{n}_{зан} = \frac{A}{\mu}$$

Коэффициент простоя показывает процент времени простоя обслуживающих приборов:

$$K_{пр} = 1 - \frac{\bar{n}_{зан}}{n}$$

В рамках рассматриваемой задачи показатели для владельцев относительно времени суток будут выглядеть следующим образом (рис. 20.28–20.31)

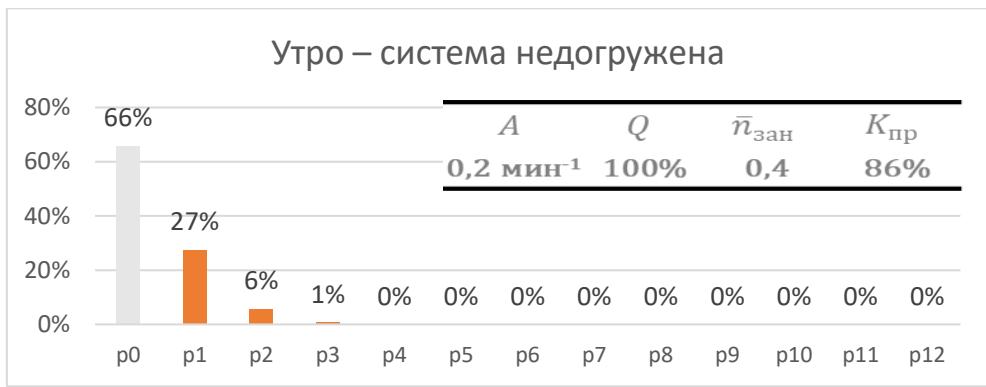


Рис. 20.28. Показатели для владельцев утром

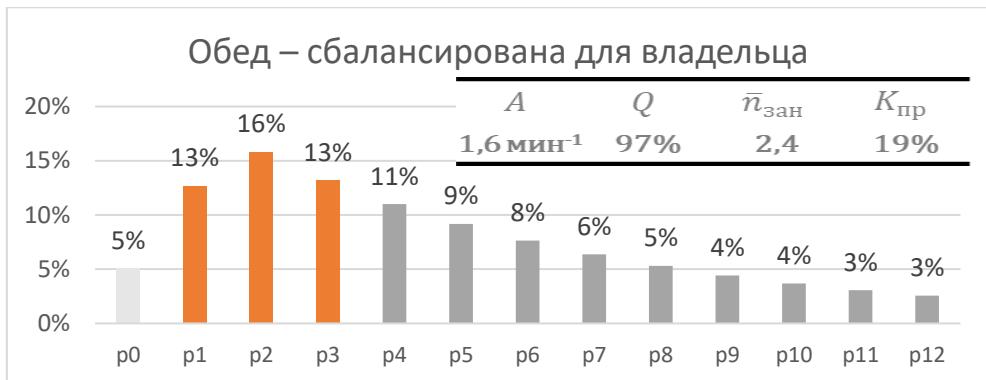


Рис. 20.29. Показатели для владельцев в обед

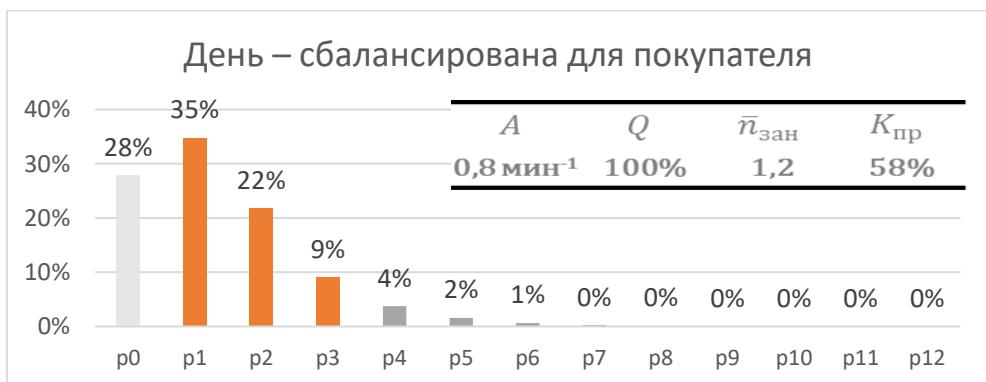


Рис. 20.30. Показатели для владельцев днем

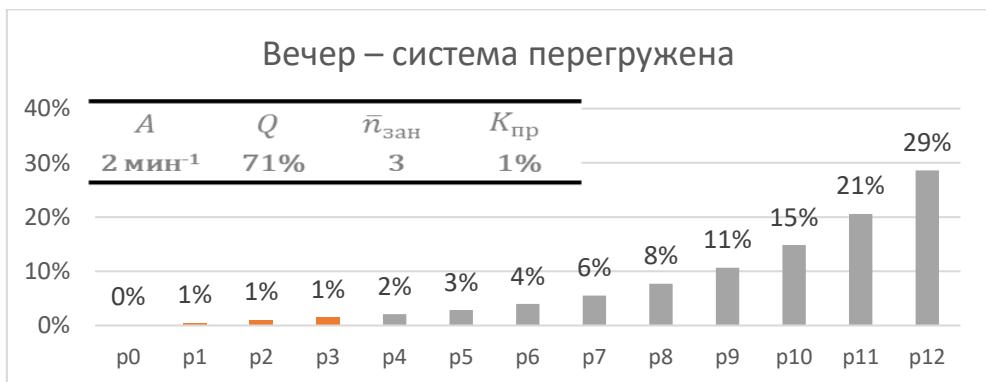


Рис. 20.31. Показатели для владельцев вечером

Результаты оптимизации

В результате анализа магазина (таблица 20.5) было обнаружено, что оптимальнее всего в утренние часы использовать одну кассу, в обед – три кассы, днем – две кассы, а для вечернего часа-пик дооснастить магазин еще двумя кассами самообслуживания вместо пространства для очереди, потому что средняя длина очереди не превышает двух человек из девяти возможных.

В таком случае из-за очередей будет уходить не более 3% клиентов (ожидание 1 мин).

Простой касс – не более 20% времени в нагруженные часы, и около 40–60% в свободные.

Таблица 20.5

Итоговый анализ

	ρ	n	$p_{\text{отк}}$	$p_{\text{оч}}$	$L_{\text{оч}}$	$T_{\text{оч}}$	A	Q	$\bar{n}_{\text{зан}}$	$K_{\text{пр}}$
Утро	0,41	1	0%	3%	0,3	1,07 мин	0,28 мин ⁻¹	100%	0,4	58%
Обед	2,5	3	3%	53%	2,0	1,26 мин	1,62 мин ⁻¹	97%	2,4	19%
День	1,25	2	0%	19%	0,8	0,90 мин	0,83 мин ⁻¹	100%	1,2	38%
Вечер	4,16	5	3%	70%	1,9	0,70 мин	2,68 мин ⁻¹	97%	4,0	19%



Агнер Крагуп Эрланг (дат. Agner Krarup Erlang) 1 января 1878 - 3 февраля 1929 — датский математик, статистик и инженер, основатель научного направления по изучению телекоммуникационных систем и теории массового обслуживания. Сотрудник телефонной компании в Копенгагене. Книга «Теория вероятностей и телефонные разговоры» (1909 г.) заложила основы теории массового обслуживания. Эрланг (1 Erl) - единица измерения трафика в телекоммуникационных системах. Erlang – язык программирования, разработанный фирмой Ericsson для создания распределенных вычислительных систем (ERicsson LANGuage). Распределение Эрланга применяется для моделирования случайного потока звонков и обращений в телекоммуникационных системах.

ЗАКЛЮЧЕНИЕ

Представленные к изучению материалы являются только знакомством с рассмотренными областями, каждая из которых является отдельной ветвью науки под названием «Теория принятия решений». Многие из них, такие как теория графов или теория игр из-за своей масштабности выделились в отдельные отрасли знаний, но все они направлены на то, что помогает человеку с помощью математических методов принимать оптимальные решения в различных жизненных ситуациях. Так, например, математическое программирование называется принятием решений в условиях полной информации, игры с «природой» - в условиях риска и неопределенности, в целом теория игр описывает принятие решений в условиях противодействия и т.д.

Чтобы систематизировать изученный материал и правильно уложить его в голове далее на рисунке приводится ментальная карта задач, методов их решения и разделов, к которым они относятся, а также возможные переходы между ними.

Мы надеемся, что изучение материалов курса прошло для вас легко и без лишних усилий, а знания курса надежно условились и пригодятся в дальнейшей профессиональной деятельности.

Плейлист с видеолекциями можно найти [по ссылке](#) или по ключевым словам «Математические методы принятия решений» на канале автора на Youtube.



Рис. Структура изученного материала

УЧЕБНОЕ ПОСОБИЕ ПО ДИСЦИПЛИНЕ
«Математические методы принятия решений»

Автор:

Моисеев Георгий Викторович, канд. техн. наук, доцент Департамента анализа данных и машинного обучения факультета информационных технологий и анализа больших данных Финансового университета при Правительстве РФ.

Компьютерный набор, верстка Д.А. Чистякова

Формат 60x90/16. Гарнитура Times New Roman

п.л. 13,37. Изд. № - 2021. Электронное издание

Вычитка и корректура выполнены автором