МИНИСТЕРСТВО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ВОЛГОГРАДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Г.Н. КОПЫЛОВ, Н.Н. СУХАНОВА

МАТЕМАТИЧЕСКИЕ МЕТОДЫ В ЭКОНОМИКЕ

Методическое пособие
(для студентов экономического факультета)

Волгоград 2002

ББК 66в641я73
К65

Рецензенты:
канд. физ.-мат. наук, профессор *В.К. Карташов*,
д-р. физ.-мат. наук *А.Г. Лосев*

Рекомендовано к печати
редакционно-издательским советом университета

Копылов Г.Н., Суханова Н.Н.

К65 Математические методы в экономике: Методическое пособие (для студентов экономического факультета). — Волгоград: Издательство ВолГУ, 2002. — 108 с.

ISBN 5-85534-533-5

В работе приведены конспекты лекций, читаемых авторами студентам экономических специальностей Волгоградского государственного университета. Рассматриваются модели из разных сфер экономической деятельности, посвященные проблеме выбора наилучшего варианта из возможных. В каждом параграфе описывается конкретная практическая ситуация, строится математическая модель и обсуждаются точные и эвристические алгоритмы ее решения. Предложенные алгоритмы подробно разбираются на примерах. В заключительном параграфе содержатся задачи, предназначенные для индивидуальных заданий.

ISBN 5-85534-533-5

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ 3
Основные понятия 8
§1. ЗАДАЧА О ДВУХ ГОРОДАХ 12
§2. СЕТЕВЫЕ ГРАФИКИ 22
§3. ЗАДАЧА О КРАТЧАЙШЕМ ПУТИ В ГРАФЕ 35
§4. ЗАДАЧА О НАЗНАЧЕНИЯХ 42
§5. ТРАНСПОРТНАЯ ЗАДАЧА 52
§6. ЗАДАЧА КОММИВОЯЖЕРА 80
§7. ТИПОВЫЕ РАСЧЕТЫ 89
СПИСОК РЕКОМЕНДОВАННОЙ ЛИТЕРАТУРЫ. 105

ВВЕДЕНИЕ

Что изучает математика? Можно выделить два ответа на этот вопрос.

1) Математика изучает математику, то есть свой собственный математический мир, созданный самими математиками. Если бы математики (Ньютон, Лейбниц и др.) не разработали интегральное и дифференциальное исчисление, то как школьники и студенты могли бы изучать производные и интегралы?

2) Математика изучает окружающий нас мир, который, как говорят философы, существует вне нас и независимо от нас.

Верны оба ответа. Да, математика изучает свои математические объекты, но они являются отражением реального мира. Прежде чем человек придумал число «два», он много раз наблюдал двух людей, двух рыб и т. д., пока не заметил между ними общее и это общее назвал числом «два».

Для применения математики к реальному миру необходимо построить математическую модель, сформулировать некоторую математическую задачу.

Что такое модель? Моделью объекта A является другой объект B, имеющий некоторые общие свойства с A. Рассмотрим примеры. Фотография или портрет человека дает представление о его внешнем виде и частично о характере. Многие детские игрушки можно назвать моделями реальных объектов. При этом степень приближения к оригиналу может быть разной. Книги являются моделями реального мира. Человек, никогда не бывавший в Англии, по книгам может познакомиться с ней. При разработке новых типов кораблей, самолетов, при строительстве часто предварительно делают уменьшенный макет объекта и на макете изучают будущие свойства. План любых действий можно тоже назвать моделью. Грамотный, опытный специалист может по финансовым документам воссоздать весьма полную картину деятельности того или иного предприятия. Здесь документация предприятия является его моделью. Эти примеры можно легко продолжить.

Зачем строить и изучать модели, если можно изучать реальный объект? Модель упрощает и фиксирует объект. В модели учитываются те свойства, которые важны в данной ситуации для принятия определенного решения. В разных моделях один и тот же объект может и должен быть описан с разной степенью подробности. При назначении стипендии студенту его модель сводится к оценкам в его зачетке. При рекомендации выпускника в аспирантуру оценки в его дипломе, конечно, учитываются, но одних хороших оценок может быть недостаточно. Модель не должна быть и чересчур простой, и чересчур сложной. Подробность описания объекта диктуется нашими возможностями и нашими целями. По усложненной модели часто трудно и даже невозможно получить точное решение получившейся математической задачи. Физика Ньютона в большинстве случаев прекрасно описывает окружающий нас мир, но иногда она его искажает, и тогда должна учитываться физика Эйнштейна.

Для решения любой экономической задачи математическими методами необходимо построение математической модели. Анализируя имеющуюся информацию, человек принимает решение. Качество принятого решения зависит от многих факторов: от опыта, знаний, точности имеющейся информации, интеллекта, интуиции и т. д. Во многих ситуациях мы применяем модели, разработанные ранее, но иногда надо построить новую модель (или модифицировать старую) самостоятельно. Правильный выбор модели является первым шагом на пути получения решения. Математические модели в экономике должны строиться в процессе диалога экономиста — практика с математиком. После того как модель построена, мы решаем полученную математическую задачу. Полученное решение надо проверить на правдоподобность и с точки зрения математика, и с точки зрения экономиста. На этой стадии модель часто уточняется.

Выбору наилучших решений в различных ситуациях, возникающих в практической деятельности, и посвящен курс лекций, читаемый одним из авторов на 3-м курсе экономического факультета Волгоградского государственного университета, краткому изложению которого служит это пособие. Оно должно помочь студентам ознакомиться с некоторыми математическими моделями экономических задач и освоить алгоритмы решения получающихся математических задач.

Как построить математическую модель? При построении модели надо выбрать основные, существенные признаки объекта и записать их на языке математики. В простейших случаях построение модели напоминает задачи на составление уравнений, знакомые по школьному курсу. Сначала надо понять ситуацию так, чтобы ее можно было рассказать другому человеку. Затем определить математический язык, на котором будет строиться наша модель. Вместо того, чтобы сказать «надо найти план действий», следует указать, как его можно формализовать. План может быть задан определением значения переменных x, y либо вектора $\mathbf{z} = (z_1, z_2, ..., z_n)$, матрицы A и т. д. Далее необходимо определить множество допустимых вариантов, то есть задать множество ограничений, которым допустимые варианты должны удовлетворять. После этого надо ответить на вопрос «как сравнивать варианты?». Обычно определяется некоторая функция (она называется целевой функцией или функцией полезности). Для сравнения двух допустимых вариантов мы для каждого из них подсчитываем значение этой функции и объявляем лучшим то значение, при котором значение функции больше (или меньше).

В результате многие модели можно записать в виде задачи *математического программирования.*

Найти $max\, f(x),\ x \in A.$

Здесь A — множество допустимых вариантов, $f(x)$— целевая функция.

В случае, если функция $f(x)$ линейна, а множество ограничений задано системой линейных уравнений и неравенств, задача математического программирования превращается в задачу линейного программирования.

Рассуждения о математических моделях закончим фразой Гексли: «Математика, подобно жернову, перемалывает то, что под нее засыпают, и, как, засыпав лебеду, вы не получите пшеничной муки, так, исписав целые страницы формулами, вы не получите истины из ложных предпосылок».

Пособие состоит из введения и 6 параграфов. Параграфы посвящены различным оптимизационным задачам. В §1 рассматривается задача выбора места строительства железнодорожной станции для обслуживания двух городов. Здесь строятся три различные модели формализации одной ситуации. В §2 рассматривается задача построения сетевых графиков. В §3 решается задача о кратчайшем пути в графе. §4 посвящен решению задачи о назначениях. В §5 рассмотрена транспортная задача. Для ее решения предлагаются метод потенциалов и венгерский алгоритм. В §6 предложен алгоритм решения задачи коммивояжера. В каждом параграфе приводится ряд задач с ответами для решения на семинарских занятиях. По каждой теме предложены типовые задачи для самостоятельной работы (§7). Для генерирования задач на ЭВМ были составлены программы, некоторые из них написал А.В. Ионов, которому авторы выражают свою благодарность.

Авторы хотят выразить свою признательность Соросовскому профессору Анатолию Юрьевичу Левину, на кафедре которого они проработали около 10 лет. Большую помощь в наборе текста оказали студенты экономического факультета.

Основные понятия

О графах

В ряде случаев нужную информацию можно представить в виде графа.

Граф состоит из двух множеств: множества вершин и множества ребер (дуг), соединяющих эти вершины. В общем случае вершины графа соответствуют каким-то объектам и изображаются на схеме в виде точек, кругов, прямоугольников, а ребра соединяют эти объекты и изображаются в виде отрезков, дуг, линий, стрелок. Например, схема железных или автомобильных дорог может быть представлена в виде графа. Здесь вершины графа — это города, а ребра — соединяющие их дороги. Алгоритмы решения различных задач часто представляют в виде блок-схем. Отдельные блоки соответствуют шагам алгоритма и являются вершинами графа. Стрелки указывают порядок выполнения этих блоков и являются ребрами графа. Графы можно использовать, чтобы объяснить структуру какой-либо организации, влияние различных факторов в сложной системе и т. д.

В ряде случаев вершинам и (или) ребрам графа приписаны числа. Иногда их называют весами, а сами графы взвешенными. Например, при планировании путешествия можно построить граф, вершинам которого соответствуют города, которые планируется посетить, ребрам соответствуют возможности переезда из города в город, а вес ребра равен стоимости проезда по этому ребру.

Существуют **ориентированные** и **неориентированные** графы. В ориентированном графе мы различаем ребро (a,b) и ребро (b,a). Здесь ориентированное ребро (a,b) идет из вершины a в вершину b: $a \to b$, а ориентированное ребро (b,a) идет из вершины b в вершину a: $b \to a$. В неориентированном графе эти ребра не различаются и ребро (a,b) в равной степени соединяет обе вершины: $a \leftrightarrow b$. Ориентированные ребра часто называют дугами. Если рассматривать схему автомобильных дорог, то ее, как правило, надо считать неориентированным графом. А если рассматривать схему водного транспорта, то (с учетом течения) лучше применить ориентированный граф. Путешествовать на лодках или плотах можно только по течению. Даже если мы можем плыть против течения, то скорость, затраченные усилия и время могут существенно отличаться. Если, например, психолог изучает отношения в коллективе, то он может построить граф, в котором вершины — это люди, а ребра соединяют друзей. В этом случае у нас неориентированный граф. Если же рассмотреть граф знакомств, то есть вершины графа люди, а дуга идет от A к B, если A знает B, то такой граф во многих случаях надо рассматривать ориентированным. Например, все студенты университета знают, кто в университете ректор и кто в стране президент, но ректор не может знать всех студентов, а президент России не может знать всех жителей. Часто ориентация графа связана с перевозкой товаров. Изучая грузопотоки товаров, можно построить граф, где вершины — производители и потребители товаров, а стрелки указывают перевозки товаров. Отметим, что в этих случаях граф возможных перевозок часто является неориентированным, но граф реально осуществляемых перевозок обязательно ориентированный.

О трудоемкости алгоритмов

Важной характеристикой алгоритмов является их трудоемкость, то есть число элементарных операций, необходимых для решения задачи этим алгоритмом.

Рассмотрим пример. Можно ли пешком дойти от Волгограда до Москвы? Алгоритм решения задачи очень прост: сделай шаг правой ногой, сделай шаг левой ногой; так поступай до тех пор, пока задача не будет решена.

Алгоритм простой, но число шагов (как алгоритма, так и его исполнителя) очень велико. Поэтому этот алгоритм вряд ли будет применен на практике. Задача практически неразрешима.

Рассмотрим другой пример. Пусть студент потратил час на перемножение двух матриц размера 5×5. Сколько примерно времени ему понадобится на перемножение двух матриц размера 10×10? Трудоемкость $T(n)$ стандартного алгоритма растет как Cn^3. Здесь n — размер матрицы, C — константа, которая не зависит от n. При переходе от 5 до 10 размеры матрицы увеличились в два раза. Поэтому на перемножение двух матриц размера 10×10 студент потратит в 8 раз больше времени, то есть примерно 8 часов. Если он применит компьютер, то это изменит величину C, но все равно (если не изменить алгоритм), перемножение двух матриц размера 10×10 займет у него в 8 раз больше времени, чем перемножение двух матриц размера 5×5. Если размер матрицы увеличить в 10 раз, то время счета увеличится в 1000 раз.

Если трудоемкость алгоритма оценивается полиномом (многочленом) от размеров задачи, то такие алгоритмы называются полиномиальными. При наличии полиномиального алгоритма практически на компьютере можно решить задачу любых размеров. Основная трудность — получить данные и ввести их в компьютер.

О невозможности большого перебора

Компьютеры работают очень быстро, и с каждым годом их быстродействие увеличивается. Увеличение скорости работы компьютера приводит к надежде, что компьютер может решить любые задачи. Насколько обоснован этот оптимизм?

Быстродействие компьютеров не беспредельно. Современная физика считает, что никакой процесс не может протекать быстрее скорости света $3 \cdot 10^{10}$ см/с. Если для совершения одной операции хотя бы один электрон должен передвинуться на расстояние диаметра атома водорода (1 ангстрем = 10^{-8} см), то время выполнения операции не меньше чем (расстояние делим на скорость) $3 \cdot 10^{-19}$ с. Поэтому быстродействие компьютера не может быть больше $3 \cdot 10^{18}$ оп./с.

Рассмотрим пример. Пусть две бригады должны выполнить 100 работ. Как распределить эти работы между двумя бригадами? Выполнение любой работы можно доверить любой из бригад. Пусть мы разработали алгоритм, и компьютер для любого распределения работ может определить, возможно ли такое распределение работ, а также умеет сравнивать различные варианты. Можно ли получить лучший вариант, рассмотрев все возможные варианты?

Очевидно, что общее число этих способов распределить 100 работ между 2 бригадами равно 2^{100}. Оценим эту величину. $2^{10} \approx 1000$, $2^{100} \approx 10^{30}$. При быстродействии 10^{18} оп./с компьютеру потребуется 10^{12} с > 10^{10} мин >10^{8} ч > 10^{4} лет. Таким образом, перебрать все варианты мы не сможем.

Быстродействие компьютера можно увеличить за счет одновременного выполнения алгоритма на нескольких компьютерах. Предположим, что один компьютер имеет быстродействие 10^{18} оп./с и объем 1 см^3. Из таких компьютеров сделан суперкомпьютер размером с Земной шар. Каково быстродействие такого суперкомпьютера? Объем Земли $V = 4\pi R^3/3$. $R = 6400$ км $= 2^6\cdot 100$ км $= 2^6\cdot 10^7$ см. Поэтому $V\approx 4\cdot(2^6\cdot 10^7 \text{см})^3 = 2^{20}\cdot 10^{21}$ см$^3 \approx 10^{27}$ см^3. Поэтому такой суперкомпьютер может сделать $10^{18}\cdot 10^{27} = 10^{45}$ оп./с. Это, конечно, очень много. Но всегда ли достаточно?

Пусть из Волгограда в сторону Саратова отправляется товарный поезд. В нем около 50 вагонов. Мы хотим сформировать состав, выбрав оптимальный порядок вагонов. Общее число вариантов $50!\approx 10^{65}$. При быстродействии 10^{45} оп/с компьютеру потребуется $10^{65}/10^{45} = 10^{20}$ с $> 10^{18}$ мин $>10^{16}$ ч $> 10^{12}$ лет. Таким образом, перебрать все варианты мы не сможем и на таком суперкомпьютере.

Подведем итоги. Важной характеристикой алгоритма является его трудоемкость. Если трудоемкость какого-либо алгоритма растет как полином (многочлен) от размеров задачи, то решить задачу можно практически для любой размерности. Если же трудоемкость какого-либо алгоритма растет как 2^n или $n!$, то при больших n применение этого алгоритма практически невозможно.

§1. ЗАДАЧА О ДВУХ ГОРОДАХ

В некоторых случаях, рассматривая одну и ту же практическую ситуацию, можно предложить разные математические модели ее формализации. Рассмотрим такую ситуацию. Мэры двух городов A и B решили построить на близлежащей железной дороге L станцию для совместного использования. Как выбрать место для станции? Каждый мэр заинтересован в том, чтобы станция располагалась возможно ближе к его городу. Поэтому один мэр хотел бы построить станцию в точке A_1, а другой — в B_1 (см. рис. 1). Здесь точки A_1 и B_1 — ближайшие к городам A и B точки железной дороги. Очевидно, отрезки AA_1 и B_1B перпендикулярны линии L железной дороги. Как найти компромиссное решение?

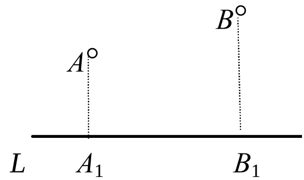


Рис. 1.

Модель 1. *Принято решение построить станцию на равном расстоянии от городов.*

Возникла следующая математическая задача.

На прямой L выбрать точку S такую, что $AS = BS$.

Для нахождения места расположения станции в этом случае достаточно знаний школьной геометрии. Чтобы найти все точки, равноудаленные от двух данных, надо построить перпендикуляр к середине соединяющего эти точки отрезка AB. Станцию нужно строить на пересечении этого срединного перпендикуляра с линией железной дороги.

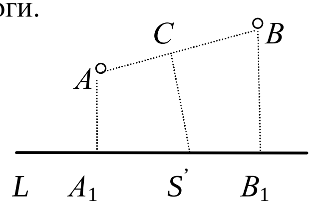


Рис. 2.

Введем систему координат. Пусть линия железной дороги совпадает с осью абсцисс. Положение городов определяется координатами точек A и B. A (a_1, a_2), B (b_1, b_2). Тогда координаты точки C, являющейся серединой отрезка AB, определяются по формулам:

$$c_1 = (a_1 + b_1)/2, \quad c_2 = (a_2 + b_2)/2. \tag{1}$$

Уравнение прямой AB определяется формулой:

$$(x - a_1)(b_2 - a_2) = (y - a_2)(b_1 - a_1).$$

Найдем k— угловой коэффициент этой прямой.

$$k = (b_2 - a_2)/(b_1 - a_1).$$

Найдем k_1 — угловой коэффициент прямой CS. Для взаимно перпендикулярных прямых

$$k \cdot k_1 = -1.$$

Отсюда

$$k_1 = (b_1 - a_1)/(b_2 - a_2). \tag{2}$$

Если прямая проходит через точку (c_1, c_2), а ее угловой коэффициент равен k_1, то ее уравнение можно задать в виде

$$y = c_2 + k_1(x - c_1). \tag{3}$$

Таким образом, прямая CS, точки которой равноудалены от точек A и B, определяется уравнением *(3)*. Для нахождения точки пересечения с осью абсцисс подставим $y = 0$. Найденное значение x задает положение станции S в модели 1. Легко найти расстояние до каждого города. Длины отрезков AS и BS вычисляются по формулам:

$$AS = \sqrt{(a_1 - x)^2 + (a_2 - y)^2} = \sqrt{(a_1 - x)^2 + a_2^{\,2}}, \quad BS = \sqrt{(b_1 - x)^2 + b_2^{\,2}}.$$

При численном решении примера желательно проверить, что $AS = BS$.

В ряде случаев эта модель дает вполне удовлетворительное решение, но может дать и явно невыгодное (см. рис. 3). Почему это происходит? Основным критерием решения было равенство расстояний от данных городов. Психологически это оправдано, так как удовлетворяет претензиям двух сторон в равной степени, но в экономических задачах важно минимизировать расходы (или максимизировать доходы). В данном примере (рис.3) можно видеть, что решение, полученное в рамках модели, не является выгодным ни для одной стороны.

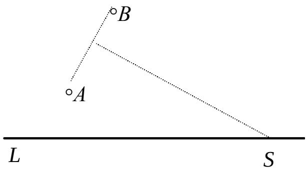


Рис. 3.

Модель 2. *Выберем место строительства станции так, чтобы суммарная длина дорог от станции до обоих городов была минимальной.*

Возникает следующая математическая задача.

На прямой L выбрать точку S так, чтобы величина AS + BS имела наименьшее значение.

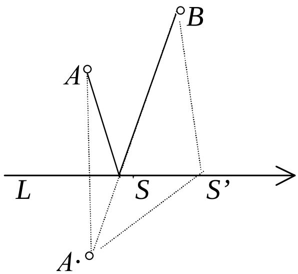


Рис. 4.

Для решения этой задачи одну из данных точек *A* или *B* (например, А) отобразим симметрично относительно прямой *L*. Полученную точку *A′* соединим с точкой *B* (см. рис.4). Точку S выберем на пересечении прямой L с прямой *A′B*. Очевидно, что это соответствует минимальной величине *AS* + *BS*, так как длина пути *AS* + *BS* равна длине отрезка *A′B*, а любая другая точка *S′* определяет ломаную *A′S′B*, длина которой больше длины отрезка *A′B*.

Например, заданы точки $A(a_1, a_2)$ и $B(b_1, b_2)$. Тогда *A′* имеет координаты $(a_1, -a_2)$, Найдем уравнение прямой *A′B:*

$$\frac{x-a_1}{b_1-a_1}=\frac{y+a_2}{b_2+a_2}.$$

Подставив в последнее уравнение значение $y = 0$, найдем значение *x*.

$$x=(a_1b_2+a_2b_1)/(b_2+a_2)\cdot$$

Точка

$$S((a_1b_2 + a_2b_1)/(b_2 + a_2);0) \qquad (4)$$

определяет место расположения станции.

Модель 3. *В ряде случаев невыгодно вести к станции две отдельные дороги. Можно выбрать где-либо перекресток P, провести дороги из A и B к перекрестку P, а затем от перекрестка P к станции S (см. рис. 5).*

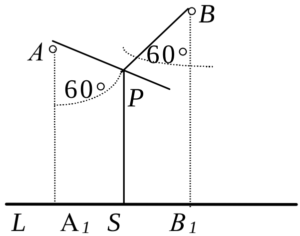


Рис. 5.

Получили следующую математическую задачу.

Определить расположение точек P и S так, чтобы S лежала на прямой L и сумма длин отрезков AP + BP + PS была бы минимальной.

Как в этом случае выбрать P и S? Очевидно, что $PS \perp L$ линии железной дороги. Можно доказать, что если P не совпадает ни с A, ни с B, ни с S, то углы APB, APS и BPS равны 120°. Следовательно, углы A_1AP и B_1BP равны 60°. Поэтому, для нахождения P надо опустить перпендикуляры AA_1 и BB_1 из точек A и B на линию L и провести прямые из точек A и B под углом 60° к отрезкам AA_1 и BB_1 внутрь вертикальной полосы, заключенной между ними. Точку пересечения прямых обозначим буквой P. Для того, чтобы определить координаты этой точки, проведем следующие выкладки. Пусть даны точки $A(a_1, a_2)$ и $B(b_1, b_2)$. Будем считать, что $a_1 < b_1$. Угловой коэффициент прямой AP, проходящей через точку A под углом 60° к AA_1, равен $\mathrm{tg}150° = -1/\sqrt{3}$, а угловой коэффициент прямой PB, проходящей под углом 60° к BB_1, равен $\mathrm{tg}30° = -1/\sqrt{3}$ (см. рис. 5). Координаты точки $P(x,y)$ определяются из системы уравнений:

$$\begin{cases} y - a_2 = -\dfrac{1}{\sqrt{3}}(x - a_1) \\ y - b_2 = \dfrac{1}{\sqrt{3}}(x - b_1). \end{cases} \tag{5}$$

Решим систему линейных уравнений (5). Сложив уравнения, найдем значение y, вычитая из второго первое, найдем значение x.

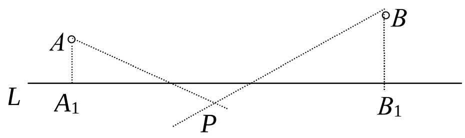


Рис. 5a.

Проанализируем решение системы уравнений (5). Здесь возможны различные случаи. Если $y < 0$, то точка P расположена ниже линии L (см. рис. 5a). В этой точке явно нет смысла строить перекресток. Что делать? В этом случае перекресток не нужен. Точка P совпадает с S. Оптимальное расположение станции S нужно искать, как в модели 2.

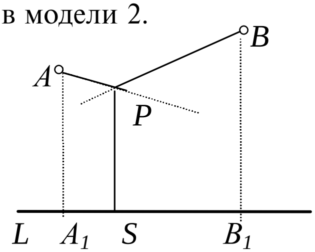


Рис. 5b.

Если $y > 0$ и при этом $a_1 < x < b_1$, то точка $P(x,y)$ расположена над линией L и находится между отрезками AA_1 и BB_1. В точке P строим перекресток, а станцию располагаем в точке $S(x,0)$ (см. рис. 5b). Дорога прокладывается от линии железной дороги вдоль отрезка SP от станции S к перекрестку P, а затем вдоль отрезка PA к городу A и вдоль отрезка PB к городу B.

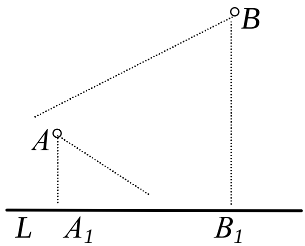


Рис. 5с.

Наконец, при $y > 0$ возможен случай, что $x < a_1$, либо $x > b_1$ (см. рис. 5с). В этом случае соединяем дорогой города A и B, а станцию S располагаем на линии железной дороги на ближайшем расстоянии от одного из городов. На рис. 5d это точка A_1. Точка P совпала с точкой A.

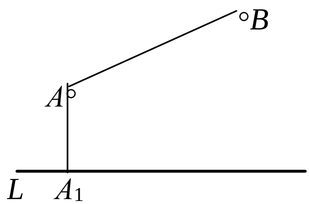


Рис. 5d.

Пример. Города заданы на координатной плоскости точками *A* и *B*. *A(1; 4), B(5; 3)*. Как определить наиболее выгодное расположение станции *S* на железной дороге, если линия железной дороги совпадает с осью *OX*?

Модель 1. *Будем искать точку расположения станции S так, чтобы AS = BS (см. рис.6).*

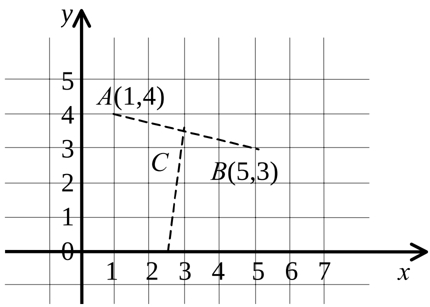


Рис. 6.

1 способ.

Определим координаты точки C — середины отрезка AB и коэффициент k_1. Используем формулы (1) и (2).

$$c_1 = (1 + 5)/2=3,\ c_2 = (4 + 3)/2 = 3{,}5,\ k_1 = (5 - 1)/(4 - 3) = 4.$$

Выпишем уравнение прямой, все точки которой равноудалены от точек A и B. Подставим в *(3)* значения k_1, c_1, c_2:

$$y = 3{,}5 + 4(x - 3).$$

Подставим $y = 0$ и получим x_S. $4x_S = 8{,}5$; $x_S = 2{,}125$. Таким образом, определили координаты точки S: $S(2{,}125;\ 0)$.

2 способ.

Это решение основано на применении векторного исчисления. Определим координаты точки C — середины отрезка AB.

$$c_1 = (1 + 5)/2=3,\ c_2 = (4 + 3)/2 = 3{,}5.$$

Вектора $\boldsymbol{AB}$ и $\boldsymbol{CS}$ перпендикулярны и вершина S лежит на оси OX. $S(x_S; 0)$.

$$\boldsymbol{AB} = (5 - 1;\ 3 - 4) = (4,\ -1),$$
$$\boldsymbol{CS} = (x_S - 3;\ 0 - 3{,}5) = (x_S - 3;\ -3{,}5).$$

Скалярное произведение этих векторов равно нулю. Рассмотрим уравнение

$$(\boldsymbol{AB},\ \boldsymbol{CS}) = 0.$$
$$4(x_S - 3) - 1(-3{,}5) = 0.$$

Откуда $4x_S = 8{,}5$, $x_S = 2{,}125$.

Получили такое же решение: $S(2{,}125;\ 0)$.

3 способ.

Найдем x из уравнения $AS = BS$.

$$AS = \sqrt{(x-1)^2 + (4-0)^2},\ BS = \sqrt{(x-5)^2 + 3^2}$$
$$\sqrt{(x-1)^2 + (4-0)^2} = \sqrt{(x-5)^2 + 3^2}.$$

Возведем обе части равенства в квадрат, приведем подобные и определим значение x.

$$x = 2{,}125.$$

Проверим, что расстояния AS и BS одинаковы.

$$AS = \sqrt{(1-2{,}125)^2 + (4-0)^2} = \sqrt{17{,}2656} = 4{,}16.$$

$$BS = \sqrt{(5-2{,}125)^2 + 3^2} = \sqrt{17{,}2656} = 4{,}16.$$

Суммарный путь $AS + BS = 8{,}32$.

Модель 2. *Ищем такое расположение станции* S, *чтобы минимизировать общую длину пути AS + BS (см. рис. 7).*

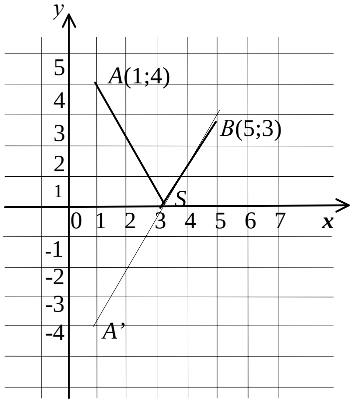


Рис. 7.

По формуле *(4)* сразу найдем расположение станции.

$$x_S = (1 \cdot 3 + 4 \cdot 5)/(3 + 4) = 23/7 = 3\tfrac{2}{7}.$$

Здесь также можно рассмотреть другое решение. Определим координаты точки *A′*. *A′ (1; -4)*. Вектор ***A′B*** коллинеарен вектору ***A′S***.

$$\boldsymbol{A'B} = (5 - 1;\ 3 - (-4)) = (4;\ 7).$$

$$\boldsymbol{A'S} = (x_S - 1;\ 0 - (-4)) = (x_S - 1;\ 4).$$

Из условия коллинеарности имеем:

$$(x_S - 1)/4 = 4/7.\quad 7x_S = 23.\quad x_S = 3.$$

Получили $S(3\tfrac{2}{7};\ 0)$.

Вычислим *AS + BS*.

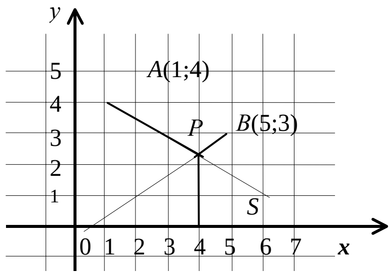
$$AS + BS = A'S + BS = A'B = \sqrt{(1-5)^2 + ((-4)-3)^2} =$$
$$= \sqrt{65} = 8{,}06.$$

Модель 2 дала лучшее решение, чем модель 1: 8,06 < 8,32, как и предполагалось.

Модель 3. И*щем расположение станции S и перекрестка P, используя формулы (1)—(5).*

Прямая *AP* расположена под углом *150°* к оси *OX*. Следовательно, угловой коэффициент $k = tg150° = -\frac{1}{\sqrt{3}}$. Подставляем в уравнение прямой $y - y_1 = k(x - x_1)$ координаты точки *A* и выписываем уравнение прямой *AB:*

$$y - 4 = -\frac{1}{\sqrt{3}}(x - 1)$$



Угловой коэффициент прямой *BP:* $k_1 = tg30 = \frac{1}{\sqrt{3}}$. Подставляем координаты точки *B* и выписываем уравнение прямой *BP*:

$$y - 3 = \frac{1}{\sqrt{3}}(x - 5).$$

Решая систему уравнений $\begin{cases} y - 4 = -\frac{1}{\sqrt{3}}(x - 1) \\ y - 3 = \frac{1}{\sqrt{3}}(x - 5), \end{cases}$ найдем точку *P* пересечения этих прямых.

Прибавим к первому уравнению второе и получим:

$$2y - 7 = -\frac{4}{\sqrt{3}}.\ y = \frac{7}{2} - \frac{4}{2\sqrt{3}}.\ y = 2{,}34.$$

Вычитая от первого уравнения второе, получим значение *x*:

$$\frac{2x}{\sqrt{3}} - \frac{6}{\sqrt{3}} = 1.\ x = \frac{6 + \sqrt{3}}{2} = 3{,}87.$$

Получили: *P* (3,87; 2,34), *S* (3,87;0). Так как точка *P* лежит выше оси *X* и в полосе между городами *A* и *B*, то строим здесь перекресток.

Вычислим суммарную длину дорог *AP* + *BP* + *PS*.

$$AP = \sqrt{(1-3{,}87)^2 + (4-2{,}34)^2} = 3{,}32.$$

$$BP = \sqrt{(5-3{,}87)^2 + (3-2{,}34)^2} = 1{,}31.$$

$$PS = 2{,}34.$$

$$AP + BP + PS = 3{,}32 + 1{,}31 + 2{,}34 = 6{,}97.$$

Модель 3 дала наименьшее суммарное расстояние: $6{,}97 < 8{,}06 < 8{,}32$.

Рассмотрим еще пример применения модели 3. Пусть города заданы на координатной плоскости точками *A* (1; 9)*, B* (5; 3). Опустим перпендикуляры на ось *X* из точек *A* и *B*. Обозначим их AA_1 и BB_1. Рассмотрим угол A_1AB. Его тангенс равен 4/6. Это меньше, чем тангенс 60 градусов. Поэтому луч, проведенный из *A* под углом в 60 °, пройдет выше точки *B*. В этом случае точка *P* совпадает с *B*. Надо соединить *A* с *B* и *B* с B_1.

Выводы. Часто в одной и той же ситуации мы можем построить разные модели. Модель 1 является примером плохо продуманной модели. Ее использование может привести к явно нелепым результатам. Модели 2 и 3 направлены на экономию средств. Модель 3 дает наилучший результат, он может совпадать с результатом, полученным при использовании модели 2.

Задачи для самостоятельного решения

Заданы координаты двух городов на плоскости. Линия железной дороги совпадает с осью *OX*. Для каждой из трех моделей найти оптимальное расположение железнодорожной станции и суммарную длину дорог, связывающих станцию с городами.

1. *A* (1, 4), *B* (6, 6). **2**. *A* (4, 3), *B* (5, 5). **3**. *A* (1, 5), *B* (6, 6).
4. *A* (2, 5), *B* (5, 7). **5**. *A* (4, 8), *B* (6, 7). **6**. *A* (4, 8), *B* (7, 4).

Ответы

1. l_1= 12,04; l_2= 11,18; l_3= 9,33. **2**. l_1= 18,03; l_2= 8,06; l_3= 5,24.
3. l_1= 12,32; l_2= 12,08; l_3= 9,83. **4**. l_1= 14,87; l_2= 12,37; l_3= 8,61.
5. l_1= 16,92; l_2= 15,13; l_3= 9,23. **6**. l_1= 20,62; l_2= 12,37; l_3= 9.

§2. СЕТЕВЫЕ ГРАФИКИ

Задача о сетевом графике возникает при планировании сроков выполнения комплекса работ. При выполнении сложного проекта — строительстве завода, запуске космического корабля и т. д. — необходимо за наименьшее время выполнить много разных работ. При планировании сроков выполнения комплекса работ будем считать известным, какие работы надо выполнить, и время, необходимое для выполнения каждой работы.

Рассмотрим простую ситуацию. Пусть надо выполнить две работы. Продолжительность одной равна 10, а другой 7 дней. За сколько дней можно выполнить обе работы?

Если работы должны выполняться последовательно, то время выполнения всего комплекса работ равно суммарному времени выполнения всех работ, то есть $10 + 7 = 17$.

Если работы можно поручить разным исполнителям и выполнять одновременно, то время выполнения комплекса равно времени выполнения самой длительной работы комплекса, то есть $max\ (10,\ 7) = 10$.

В общем случае существуют ограничения на порядок выполнения работ — какие-то работы могут выполняться одновременно, а какие-то можно начинать только по завершению других. Например, можно вести отделочные работы на первом этаже строящегося дома и одновременно строить 5-й этаж.

Информацию о комплексе работ можно записать в таблицу, где будет указано время выполнения каждой работы и номера предшествующих работ (то есть таких, которые необходимо завершить до начала данной).

Рассмотрим *пример 1.* Пусть мы собираем устройство, состоящее из двух узлов. Выделим 4 работы. Необходимо закупить материалы, изготовить оба узла и собрать устройство. При этом изготовлять узлы можно независимо один от другого, но после закупки материала, а собирать устройство можно после изготовления обоих узлов. Исходные данные можно представить в виде таблицы.

Таблица

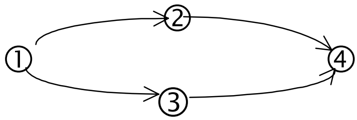
№ п/п	Содержание работ	Продолжительность работы	№ предшествующей работы
1.	Закупка материалов	3 дня	
2.	Изготовление узла 1	8 дней	1
3.	Изготовление узла 2	4 дня	1
4.	Сборка	2 дня	2,3

Известен перечень работ, время, необходимое для выполнения каждой работы, а также отношение предшествования между работами.

Информацию о комплексе работ можно также представить в виде графа. Применяются два способа построения и интерпретации графа при планировании комплекса работ.

Модель 1. *Вершины графа соответствуют работам, которые необходимо выполнить. Дуги (ориентированные ребра) содержат информацию о порядке выполнения работ. Если работу В нельзя начинать раньше, чем закончится работа А, то в графе есть дуга из вершины А в вершину В.*

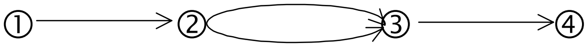
Построим граф, соответствующий примеру 1 и модели 1. Вершины графа соответствуют работам. Так как в примере 4 работы, то в графе будут 4 вершины.



Ребра графа указывают, что работы 2 и 3 нельзя начинать до окончания работы 1, а работу 4 нельзя начинать до окончания работ 2 и 3.

Модель 2. *Вершины графа соответствуют определенным моментам времени (срокам начала и окончания отдельных работ). Дуги соответствуют работам. Работа, соответствующая дуге, выходящей из вершины i, может начинаться только после того, как будут выполнены все работы, заканчивающиеся в вершине i.*

Построим граф, соответствующий примеру 1 и модели 2. Вершины графа соответствуют моментам времени начала и окончания отдельных работ. Работам соответствуют дуги. Работа 1 начнется в вершине 1 и закончится в вершине 2. Работы 2 и 3 начнутся в вершине 2 и закончатся в вершине 3. Работа 4 начнется в вершине 3 и закончится в вершине 4. Здесь вершина 1 соответствует началу работ. Вершина 2 соответствует окончанию покупки материалов и началу изготовления узлов. Вершина 3 соответствует началу сборки. Вершина 4 соответствует окончанию всех работ.



В дальнейшем будем придерживаться только модели 1.

В рассмотренном выше примере найдем время выполнения комплекса работ и составим график их выполнения. Для каждой работы определим самый ранний срок ее выполнения. Работа 1 не имеет предшественников. Начнем ее в момент 0 и, так как время ее выполнения 3 дня, закончим в момент 3. Работа 2 имеет предшественником работу 1, поэтому самое раннее время начала работы 2 равно самому раннему времени окончания работы 1, то есть 3. Так как длительность ее равна 8, то самое раннее время ее окончания 11(=3 + 8). Работу 3 можно начать в момент 3 и закончить в момент 7. У работы 4 два предшественника. Самое раннее время ее начала определяется ранними сроками окончания тех работ, которые являются ее предшественниками. Оно равно максимальному из них. Поэтому ее можно начать в 11 и закончить в 13. Время окончания этой работы и будет временем окончания всего комплекса работ (13 дней). Запишем результаты нашего обсуждения в таблицу.

Введем некоторые обозначения:

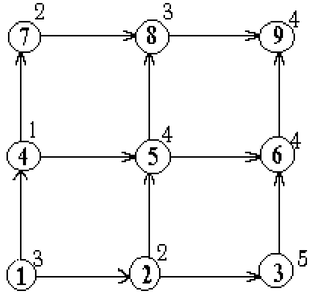
$T_i^{рн}$ — самый ранний срок начала i-й работы;

$T_i^{ро}$ — самый ранний срок окончания i-й работы; $i = 1, 2, 3, 4$.

№ п/п	Содержание работ	Продолжительность работы	№ предшествующей работы	Самый ранний срок начала работы	Самый ранний срок окончания работы
1.	Закупка материалов	3 дня		0	3
2.	Изготовление узла 1	8 дней	1	3	11
3.	Изготовление узла 2	4 дня	1	3	7
4.	Сборка	2 дня	2,3	11	13

Два последних столбца нашей таблицы могут рассматриваться как график выполнения комплекса работ.

Пример 2. Пусть комплекс работ представлен в виде графа.



Здесь у нас 9 работ и 12 ограничений на порядок выполнения работ. Работы соответствуют вершинам нашего графа, а ограничения его дугам. Например, работу 5 нельзя начать, пока не закончатся работы 2 и 4, так как в вершину 5 входят ребра из 2-й и 4-й вершины. В свою очередь, до завершения работы 5 нельзя начинать работы 6 и 8.

Мы хотим составить график выполнения всего комплекса работ за минимальное время.

Найдем самые ранние сроки начала каждой работы. Так как работы занумерованы так, что номер предшественника всегда меньше номера самой работы, то определять самые ранние сроки будем в порядке увеличения номеров работ.

Работу 1 начнем в момент 0 и закончим в момент 3. Работу 2 можно начать после окончания работы 1. Поэтому самое раннее начало работы 2 равно времени самого раннего окончания работы 1, то есть 3. Так как время выполнения равно 2, то самое раннее окончание работы 2 равно 5. Для работы 3 самые ранние сроки выполнения 5 — 10. Работу 4 можно начать после выполнения работы 1. Поэтому самые ранние сроки ее выполнения 3 — 4. У работы 5 два предшественника. Из моментов их окончания надо взять максимум. Самые ранние сроки выполнения 5 — 9. У работы 6 самые ранние сроки 10 — 14. Аналогично для работы 7: 4 — 6, работы 8: 9 — 12, работы 9: 14 — 18.

Для отыскания самого раннего срока начала некоторой работы надо найти всех ее предшественников и определить максимум из самых ранних сроков окончания этих работ. Для отыскания самого раннего срока окончания работы надо к сроку самого раннего ее начала прибавить время ее выполнения.

Мы нашли самые ранние сроки выполнения каждой работы и тем самым минимальное время выполнения всего комплекса. В данном случае комплекс из 9 работ можно выполнить за 20 единиц времени. Выпишем график выполнения в виде строки:

0—3, 3—5, 5—10, 3—4, 5—9, 10—14, 4—6, 9—12, 14—18.

Здесь первый интервал задает самые ранние сроки выполнения 1-й работы, второй интервал — 2-й работы и т. д.

Подсчитаем самые поздние сроки выполнения каждой работы при условии, что мы должны уложиться в самый ранний срок выполнения всего комплекса работ.

Работу 9 необходимо закончить к 18. Поэтому самый поздний срок ее окончания равен 18. Самый поздний срок начала равен 18 — 4 = 14.

Самый поздний срок окончания любой работы определяется самыми поздними сроками начала ожидающих ее работ и равен минимальному из этих сроков. Самый поздний срок начала работы равен самому позднему сроку ее окончания минус время ее выполнения.

Работу 8 необходимо закончить к 14. Поэтому самый поздний срок ее окончания равен 14. Самый поздний срок начала равен 14 — 3 =11.

Работу 7 необходимо закончить к 11. Поэтому самый поздний срок окончания равен 11. Самый поздний срок начала равен 11 — 2 = 9.

Работу 6 необходимо закончить к 14. Поэтому самый поздний срок ее окончания равен 14. Самый поздний срок начала равен 14 — 4 =10.

Работу 5 необходимо закончить к 10. Поэтому самый поздний срок ее окончания равен 10. Самый поздний срок начала равен 10 — 4 = 6.

Работу 4 необходимо закончить к 6. Поэтому самый поздний срок ее окончания равен 6. Самый поздний срок начала равен 6 — 1 = 5.

Работу 3 необходимо закончить к 10. Поэтому самый поздний срок ее окончания равен 10. Самый поздний срок начала равен 10 — 5 = 5.

Работу 2 необходимо закончить к 5. Поэтому самый поздний срок ее окончания равен 5. Самый поздний срок начала равен 5 — 2 = 3.

Работу 1 необходимо закончить к 3. Поэтому самый поздний срок ее окончания равен 3. Самый поздний срок начала равен 3 — 3 = 0.

Выпишем результаты в виде строки:

0 — 3, 3 — 5, 5 — 10, 5 — 6, 6 — 10, 10 — 14, 9 — 11, 11 — 14, 14 — 18.

Аналогично с предыдущим, здесь последовательно указаны сроки самого позднего начала и окончания каждой работы.

Важной характеристикой комплекса работ являются резервы, то есть время, на которое без больших неприятных последствий можно задержать ту или иную работу.

Определим два вида резервов[1].

Полный резерв $R_i^{пол}$ — время, на которое можно задержать выполнение работы i без изменения времени окончания всего комплекса работ:

$$R_i^{\text{пол}} = T_i^{\text{пн}} - T_i^{\text{рн}}.$$

Свободный резерв $R_i^{св}$ — время, на которое можно задержать выполнение i-й работы без изменения самых ранних сроков начала последующих работ:

$$R_i^{\text{св}} = \min\{T_k^{\text{рн}}\} - T_i^{\text{ро}}.$$

Здесь минимум берется по всем тем работам k, для которых работа i является предшественником.

Легко заметить, что $R_i^{пол} \geq R_i^{св}$. Отсюда следует, что если полный резерв равен нулю, то и свободный резерв равен нулю. Обратное может быть неверным.

Для нахождения полного резерва нужно от самых поздних сроков выполнения работы отнять самые ранние. Для нахождения свободного резерва надо найти минимальное время, когда эта работа кому-либо понадобится, и отнять самое раннее время окончания данной работы.

Найдем полные и свободные резервы для всех работ в нашем примере:

У работы 1 самые ранние сроки выполнения совпадают с самыми поздними. Полный резерв этой работы равен 0. У работы 2 самые ранние сроки совпадают с самыми поздними. Полный

[1] Иногда рассматриваются и другие виды резервов.

резерв этой работы равен 0. У работы 3 самые ранние сроки совпадают с самыми поздними. Полный резерв этой работы равен 0. У работы 4 самые ранние сроки 3 — 4, а самые поздние 5 — 6. Поэтому полный резерв этой работы равен 2. У работы 5 самые ранние сроки 5 — 9, а самые поздние 6 — 10. Поэтому полный резерв этой работы равен 1. У работы 6 самые ранние сроки совпадают с самыми поздними. Полный резерв этой работы равен 0. У работы 7 самые ранние сроки 4 — 6, а самые поздние 9 — 11. Поэтому полный резерв этой работы равен 5. У работы 8 самые ранние сроки 9 — 12, а самые поздние 11 — 14. Поэтому полный резерв этой работы равен 3. У работы 9 самые ранние сроки совпадают с самыми поздними. Полный резерв этой работы равен 0. Запишем полные резервы вершин: 0, 0, 0, 2, 1, 0, 5, 3, 0.

Свободный резерв не может быть больше полного. Поэтому у тех работ, у которых полный резерв равен 0, свободный резерв тоже равен 0. Подсчитаем свободный резерв у тех работ, у которых полный резерв больше 0. Работу 4 можно закончить в 4. В этот момент ее ждет работа 7. Поэтому свободный резерв работы 4 равен 0. Работу 5 можно закончить в 9. В этот момент ее ждет работа 8. Поэтому свободный резерв работы 5 равен 0. Работу 7 можно закончить в 6. Ее ждет только работа 8. Но она не может начаться ранее 9. Поэтому свободный резерв работы 7 равен 3 (разность между 9, временем, когда она понадобится, и 6, временем самого раннего окончания). Работу 8 можно закончить в 12. Ее ждет только работа 9. Свободный резерв работы 8 равен 2. (разность между 16, временем, когда она понадобится, и 12, временем самого раннего окончания). Получили: 0, 0, 0, 0, 0, 0,3, 2, 0.

Работы, полный резерв которых равен нулю, называются *критическими*. Последовательность этих работ образует один или несколько *критических путей* [2], то есть последовательностей вершин, соединяющих начало комплекса с его окончанием. В рассматриваемом примере критический путь такой: 1-2-3-6-9. Иногда в графе может быть несколько критических путей, но каждая работа, у которой полный резерв равен нулю, обязательно входит хотя бы в один из критических путей.

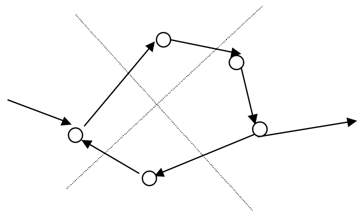
---

[2] Критический путь является путем максимальной длины в графе.

Условие наличия решения

Можно ли выполнить комплекс работ, заданный произвольной таблицей или графом? Если, например, у каждой работы есть хотя бы один предшественник, то весь комплекс невыполним. Мы не можем начать ни одной работы. Если, например, у работы 5 предшественник работа 3, а у работы 3 предшественник работа 5, то весь комплекс невыполним. Мы не можем начать работу 3, пока не закончим работу 5, и не можем начать работу 5, пока не закончим работу 3. Обычно такие ситуации возникают из-за ошибок в начальных данных. Сетевой график не должен содержать ориентированных циклов, то есть в нем не должно быть такого множества вершин, в котором в каждую последующую вершину идет ориентированная дуга из предыдущей.

Такого фрагмента не должно быть в сетевом графике:



Если в графе имеется такой цикл, то ни одну из работ, соответствующих вершинам цикла, выполнить невозможно, так как для этого нужно выполнить ее предшественника.

Как проверить, есть ли в ориентированном графе циклы?

Теорема. *В графе наверняка нет ориентированных циклов, если вершины занумерованы так, что каждое ребро идет из вершины с меньшим номером в вершину с большим номером.*

Доказательство очевидно.

Для проверки отсутствия циклов и нахождения такой нумерации вершин можно воспользоваться следующим алгоритмом:

1) Найти в графе вершину, в которую не входит ни одна дуга. Присвоить ей номер 1.

2) Среди вершин, не имеющих номера, найти такую, в которую либо не входит ни одна дуга, либо входят дуги только из уже пронумерованных вершин. Найденная вершина получает следующий свободный номер.

Этот шаг повторяется пока это возможно.

Если в результате все вершины получат номера, то в графе нет ориентированных циклов. Если же не все вершины получили номера, то среди оставшихся вершин есть цикл. Это означает, что требования противоречивы. Наш комплекс работ выполнить невозможно.

Подведем итоги. В этом параграфе была рассмотрена следующая задача.

Известен перечень работ, время, необходимое для выполнения каждой работы, а также отношение предшествования между работами.

Нас интересуют ответы на следующие вопросы:

Можно ли выполнить данный комплекс работ?

За какое минимальное время можно выполнить все работы?

Есть ли резервы сроков выполнения отдельных работ?

Какие работы самые важные? (Срыв сроков выполнения каких работ приведет к срыву сроков сдачи всего комплекса?).

Для ответов на эти вопросы были рассмотрены быстрые (полиноминальные) алгоритмы. Это означает, что в практической ситуации с применением ЭВМ можно решить задачу практически любой размерности.

Отметим два момента, которые могут привести к более сложным задачам и существенно затруднить поиск оптимального решения:

1. В ряде задач могут возникнуть другие ограничения на порядок выполнения работ. Например, «работы А и В могут выполняться в любом порядке, но не одновременно».

2. Мы считали, что сроки выполнения каждой работы фиксированы. В реальных задачах они могут быть случайными величинами, а также зависеть от распределения некоторого ресурса (например, денег или количества работников).

При учете этих факторов могут получиться задачи, для которых не известны быстрые и точные алгоритмы решения.

Задачи для самостоятельного решения

Задан комплекс из 20 работ, расположенных в виде прямоугольника 4×5. Каждую работу можно начинать выполнять после окончания ее соседа слева и соседа сверху. Найти: 1) минимальное время выполнения комплекса работ; 2) самые ранние и самые поздние возможные сроки выполнения каждой работы; 3) полный и свободный резервы каждой работы; 4) критический путь; 5) число критических работ; 6) число работ, у которых свободный резерв равен 0; 7) число работ, у которых свободный резерв равен полному резерву.

1. 6 5 5 3 3
6 2 6 6 6
2 4 7 6 4
7 7 2 7 4

2. 5 6 2 5 4
3 5 5 3 3
6 6 5 7 7
3 6 7 3 5

3. 2 7 6 2 5
2 2 6 3 2
3 4 3 7 7
4 3 2 2 5

4. 4 4 6 3 5
3 3 5 2 4
7 3 6 4 3
7 5 5 4 2

5. 5 6 7 7 2
5 7 4 6 5
5 4 4 4 3
2 3 7 2 4

6. 4 4 2 4 3
5 5 2 7 5
5 4 2 6 4
6 5 6 2 3

Ответы

1. Ранние сроки выполнения работ:

0—6	6—11	11—16	16—19	19—22
6—12	12—14	16—22	22—28	28—34
12—14	14—18	22—29	29—35	35—39
14—21	21—28	29—31	35—42	42—46.

Критический путь: (1, 1) (1, 2) (1, 3) (2, 3) (3, 3) (3, 4) (4, 4) (4, 5). Указаны номера строк и столбцов работ одного критического пути.

Поздние сроки выполнения работ:

0—6	6—11	11—16	20—23	29—32
8—14	14—16	16—22	23—29	32—38
16—18	18—22	22—29	29—35	38—42
19—26	26—33	33—35	35—42	42—46.

Номер работы 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20.
Полные резервы: 0 0 0 4 10 2 2 0 1 4 4 4 0 0 3 5 5 4 0 0.
Свободные резервы: 0 0 0 0 6 0 0 0 0 1 0 3 0 0 3 0 1 4 0 0.

Время выполнения: 46.
Число работ, у которых свободный резерв равен 0: 14.
Число критических работ: 8.
Число работ, у которых свободный резерв равен полному: 10.

2. Ранние сроки выполнения работ:

0—5	5—11	11—13	13—18	18—22
5—8	11—16	16—21	21—24	24—27
8—14	16—22	22—27	27—34	34—41
14—17	22—28	28—35	35—38	41—46.

Критический путь: (1, 1) (1, 2) (2, 2) (3, 2) (3, 3) (3, 4) (3, 5) (4, 5). Указаны номера строк и столбцов работ одного критического пути.

Поздние сроки выполнения работ:

0—5	5—11	15—17	19—24	27—31
7—10	11—16	17—22	24—27	31—34
10—16	16—22	22—27	27—34	34—41
22—25	25—31	31—38	38—41	41—46.

Номер работы 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20.
Полные резервы: 0 0 4 6 9 2 0 1 3 7 2 0 0 0 0 8 3 3 3 0.
Свободные резервы: 0 0 0 0 2 0 0 0 0 7 0 0 0 0 0 5 0 0 3 0.

Время выполнения 46.
Число работ, у которых свободный резерв равен 0: 16.
Число критических работ 8.
Число работ, у которых свободный резерв равен полному: 10.

3. Ранние сроки выполнения работ:

0—2	2—9	9—15	15—17	17—22
2—4	9—11	15—21	21—24	24—26
4—7	11—15	21—24	24—31	31—38
7—11	15—18	24—26	31—33	38—43

Критический путь: (1, 1) (1, 2) (1, 3) (2, 3) (3, 3) (3, 4) (3, 5) (4, 5). Указаны номера строк и столбцов работ одного критического пути.

Поздние сроки выполнения работ:

0—2	2—9	9—15	19—21	24—29
11—13	13—15	15—21	21—24	29—31
14—17	17—21	21—24	24—31	31—38
27—31	31—34	34—36	36—38	38—43.

Номер работы 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20.
Полные резервы: 0 0 0 4 7 9 4 0 0 5 10 6 0 0 0 20 16 10 5 0.
Свободные резервы: 0 0 0 0 2 0 0 0 0 5 0 0 0 0 0 4 6 5 5 0.

Время выполнения: 43.
Число работ, у которых свободный резерв равен 0: 14.
Число критических работ 9.
Число работ, у которых свободный резерв равен полному: 11.

4. Ранние сроки выполнения работ:

0—4	4—8	8—14	14—17	17—22
4—7	8—11	14—19	19—21	22—26
7—14	14—17	19—25	25—29	29—32
14—21	21—26	26—31	31—35	35—37.

Критический путь: (1, 1) (2, 1) (3, 1) (4, 1) (4, 2) (4, 3) (4, 4) (4, 5). Указаны номера строк и столбцов работ одного критического пути.

Поздние сроки выполнения работ:

0—4	5—9	9—15	20—23	23—28
4—7	12—15	15—20	25—27	28—32
7—14	17—20	20—26	27—31	32—35
14—21	21—26	26—31	31—35	35—37.

Номер работы 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20.
Полные резервы: 0 1 1 6 6 0 4 1 6 6 0 3 1 2 3 0 0 0 0 0.
Свободные резервы: 0 0 0 0 0 0 3 0 1 3 0 2 0 0 3 0 0 0 0 0.
Время выполнения: 37.
Число работ, у которых свободный резерв равен 0: 15.
Число критических работ 8.
Число работ, у которых свободный резерв равен полному: 9.

5. Ранние сроки выполнения работ:

0—5	5—11	11—18	18—25	25—27
5—10	11—18	18—22	25—31	31—36
10—15	18—22	22—26	31—35	36—39
15—17	22—25	26—33	35—37	39—43.

Критический путь: (1, 1) (1, 2) (1, 3) (1, 4) (2, 4) (2, 5) (3, 5) (4, 5). Указаны номера строк и столбцов работ одного критического пути.

Поздние сроки выполнения работ:

0—5	5—11	11—18	18—25	29—31
9—14	14—21	21—25	25—31	31—36
17—22	22—26	26—30	32—36	36—39
25—27	27—30	30—37	37—39	39—43.

Номер работы 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20.
Полные резервы: 0 0 0 0 4 4 3 3 0 0 7 4 4 1 0 10 5 4 2 0.
Свободные резервы: 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 5 1 2 2 0.

Время выполнения: 43.
Число работ, у которых свободный резерв равен 0: 15.
Число критических работ: 8
Число работ, у которых свободный резерв равен полному: 10.

6. Ранние сроки выполнения работ:

0—4	4—8	8—10	10—14	14—17
4—9	9—14	14—16	16—23	23—28
9—14	14—18	18—20	23—29	29—33
14—20	20—25	25—31	31—33	33—36.

Критический путь: (1, 1) (2, 1) (3, 1) (4, 1) (4, 2) (4, 3) (4, 4) (4, 5). Указаны номера строк и столбцов работ одного критического пути.

Поздние сроки выполнения работ:

0—4	5—9	10—12	12—16	21—24
4—9	9—14	14—16	16—23	24—29
9—14	16—20	21—23	23—29	29—33
14—20	20—25	25—31	31—33	33—36.

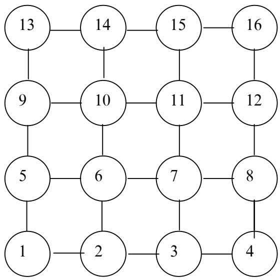
Номер работы 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20.
Полные резервы: 0 1 2 2 7 0 0 0 0 1 0 2 3 0 0 0 0 0 0 0.
Свободные резервы: 0 0 0 0 6 0 0 0 0 1 0 0 3 0 0 0 0 0 0 0.

Время выполнения: 36.
Число работ, у которых свободный резерв равен 0: 17.
Число критических работ: 13.
Число работ, у которых свободный резерв равен полному: 15.

§3. ЗАДАЧА О КРАТЧАЙШЕМ ПУТИ В ГРАФЕ

При перевозке груза из пункта *A* в пункт *B* мы хотим выбрать оптимальный маршрут. Эта задача может рассматриваться как задача поиска пути в графе. Здесь вершинами графа могут быть города, железнодорожные станции, перекрестки и т. д., а ребрами графа являются дороги.

Рассмотрим *пример.* Задан неориентированный граф, имеющий 16 вершин.



Как можно попасть из вершины *1* в вершину *16*? Здесь можно воспользоваться путем $1 \to 2 \to 3 \to 4 \to 8 \to 12 \to 16$, а можно $1 \to 2 \to 6 \to 10 \to 11 \to 15 \to 16$. Если на каждом шаге мы идем только вверх или вправо, то всего в этом примере имеются 20 возможных путей из вершины *1* в вершину *16.* Иногда может быть выгодно разрешить двигаться не только вверх или вправо. В этом случае число путей увеличится. Так, возможен вариант $1 \to 2 \to 6 \to 5 \to 9 \to 10 \to 11 \to 12 \to 16$.

Как сравнивать возможные варианты? Это зависит от поставленной цели. Например, при выборе маршрута поездки с работы домой мы учитываем много факторов: наличие свободного времени, денег, состояние погоды, время суток и т. д. Здесь выбор происходит по многим критериям. В формализованной модели будем считать, что каждому ребру графа приписано определенное число. Назовем это число длиной ребра. Длина пути определяется как сумма длин входящих в него ребер. Нашей задачей является нахождение кратчайшего пути, ведущего из одной вершины в другую.

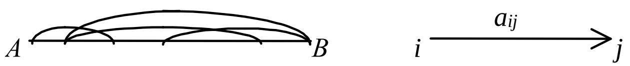
Получили следующую математическую задачу.

Дан ориентированный граф, каждому ребру приписана некоторая длина. Длина пути из одной вершины в другую равна сумме длин ребер, входящих в этот путь. Необходимо найти кратчайший из путей, ведущих из вершины *A* в вершину *B*.

Такая задача может возникнуть в различных ситуациях. Например, если мы захотим перевезти груз, то ребра графа — это дороги, а вершины графа — города, стоящие на перекрестках путей. Длины ребер определяются условиями задачи. Если мы хотим найти самый дешевый путь, то за длину ребра нужно взять стоимость перевозки груза по этому ребру. Если мы хотим найти самый быстрый путь, то в качестве длины надо взять время проезда. Если самый короткий — расстояния между соответствующими городами.

В частности, задача о кратчайшем пути возникает при рассмотрении задачи об использовании оборудования.

Рассмотрим *пример*. Пусть наше производство в течение года нуждается в автомобиле. Нам представляются различные возможности. Например, мы можем арендовать автомобиль на определенный срок по определенной стоимости. Мы можем купить автомобиль, какое-то время им пользоваться, а затем продать, и т. д. Ставится задача разработать самый дешевый план использования оборудования на указанный период времени. Здесь вершинам графа соответствуют моменты времени, а каждой имеющейся возможности (в каждый момент времени) будет соответствовать ребро графа. Может быть несколько ребер, соединяющих одну и ту же пару вершин. За длину ребра нужно принять стоимость выбранного варианта. Можно проиллюстрировать сказанное следующим рисунком. Здесь точки на прямой соответствуют последовательным моментам времени, отрезки и дуги соответствуют наличию ребра (варианта), a_{ij} стоимость соответствующего варианта использования автомобиля на интервале времени $[i, j]$.



Алгоритм расстановки меток

Предложенный ниже алгоритм нахождения кратчайшего пути состоит из двух частей. Сначала определяются длины кратчайших путей из вершины *A* во все вершины графа, затем, зная длины, находим кратчайший путь, то есть список вершин, через которые он проходит.

Нахождение длин кратчайших путей, выходящих из вершины A, будем производить, используя *алгоритм расстановки меток*. В процессе работы алгоритма каждой вершине будет приписано некоторое число, в дальнейшем эти числа мы будем называть метками. Если какой-то вершине приписана метка, равная d, то это значит, что существует путь из вершины A в эту вершину длиной d. Если найден более короткий путь, то значение метки заменяется на меньшее. Метки могут быть временными или постоянными. Постоянные метки будем отмечать значком *, не отмеченные таким значком — временные. Например, l_i — временная, а l_i^* — постоянная. Метка становится постоянной в том случае, если она соответствует самому короткому пути из вершины A в данную вершину. Задача решена, если все метки стали постоянными.

Итак, выпишем шаги нашего алгоритма.

Шаг 1. Каждой вершине i приписываем некоторое число l_i, равное длине самого короткого, известного на данный момент пути из вершины A в данную вершину. Вначале это $l_A = 0$, а $l_2 = l_3 = \ldots = l_B = \infty$. Не ограничивая общности можно считать, что вершине A соответствует вершина с номером 1, а вершине B — вершина с самым большим номером. На данном этапе все метки временные.

Шаг 2. Среди временных меток ищем минимальную. Например, это вершина i. Просматриваем все ребра, выходящие из вершины i, и проверяем выполнение неравенства

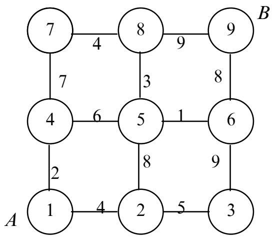
$$l_i + a_{ij} < l_j \,. \tag{3.1}$$

Для тех вершин j, для которых оно справедливо, заменяем метку l_j на $l_i + a_{ij}$, эта величина соответствует пути из A в вершину i длиной l_i, а затем по ребру a_{ij} в вершину j. Метка l_i становится постоянной.

Задача решена, если все метки постоянные. Если же среди меток есть временные, то повторяем *шаг 2.*

Длина кратчайшего пути из A в B равна l_B^*.

Пример. Рассмотрим граф, имеющий девять вершин и 12 ребер (см. чертеж). В нем выделены вершины A и B, необходимо найти кратчайший путь из A в B.



Выполняя первый шаг, вершине A припишем число $l_A = 0$, всем остальным вершинам — бесконечность:

$l_A = l_1 = 0$, $l_2 = \infty$, $l_3 = \infty$, $l_4 = \infty$, $l_5 = \infty$, $l_6 = \infty$, $l_7 = \infty$, $l_8 = \infty$, $l_B = l_9 = \infty$.

На следующем этапе выбираем вершину A, ей соответствует минимальная метка. Из вершины A выходят ребра в вершину 2 и 4.

Очевидное неравенство (3.1) выполняется как для вершины 2:

$$l_1 + a_{12} < l_2 \Rightarrow 0 + 4 < \infty,$$

так и для вершины 4:

$$l_1 + a_{14} < l_4 \Rightarrow 0 + 2 < \infty.$$

Положим:

$$l_2 = 4,\ l_4 = 2.$$

Метка l_A становится постоянной, $l_A^* = 0$. Выпишем новые значения меток:

$l_A^* = 0$, $l_2 = 4$, $l_3 = \infty$, $l_4 = 2$, $l_5 = \infty$, $l_6 = \infty$, $l_7 = \infty$, $l_8 = \infty$, $l_B = l_9 = \infty$.

Повторяем *шаг 2.* Среди временных меток выбираем наименьшую. Это l_4. Из нее выходят ребра в вершины 5 и 7.

Проверяем неравенство (3.1) для вершин 5 и 7:

$$l_4 + a_{45} < l_5 \Rightarrow 2 + 6 < \infty.$$

$$l_4 + a_{47} < l_7 \Rightarrow 2 + 7 < \infty.$$

Заменяем l_5 и l_7. $l_5 = 8$, $l_7 = 9$. Метка l_4 становится постоянной. $l_4^* = 2$.

Значения меток: $l_A^* = 0$, $l_2 = 4$, $l_3 = \infty$, $l_4^* = 2$, $l_5 = 8$, $l_6 = \infty$, $l_7 = 9$, $l_8 = \infty$, $l_B = l_9 = \infty$.

Вновь выполняем *шаг 2.* Здесь постоянные метки l_A и l_4. Среди временных наименьшее значение имеет l_2 ($l_2 = 4$, $l_5 = 8$, $l_7 = 9$).

Проверяем вершины 5 и 3. Изменяется значение l_3, $l_3 = 9$. Постоянной становится метка l_2, $l_2^* = 4$.

$l_A^* = 0$, $l_2^* = 4$, $l_3 = 9$, $l_4^* = 2$, $l_5 = 8$, $l_6 = \infty$, $l_7 = 9$, $l_8 = \infty$, $l_B = l_9 = \infty$.

Аналогично, среди меток l_3, l_5, l_7 ($l_3 = 9$, $l_5 = 8$, $l_7 = 9$) выбираем l_5. Из 5-й вершины есть ребра в вершины 6 и 8. Заменяем $l_6 = 9$, $l_8 = 11$. $l_5^* = 8$.

$l_A^* = 0$, $l_2^* = 4$, $l_3 = 9$, $l_4^* = 2$, $l_5^* = 8$, $l_6 = 9$, $l_7 = 9$, $l_8 = 11$, $l_B = l_9 = \infty$.

Далее выбор производится из вершин 3, 6, 7 и 8. Может быть выбрана любая из вершин 3, 6, 7($l_3 = 9$, $l_6 = 9$, $l_7 = 9$, $l_8 = 11$). Например, 3-я. В этом случае все метки сохраняют свои значения. $l_3^* = 9$.

$l_A^* = 0$, $l_2^* = 4$, $l_3^* = 9$, $l_4^* = 2$, $l_5^* = 8$, $l_6 = 9$, $l_7 = 9$, $l_8 = 11$, $l_B = l_9 = \infty$.

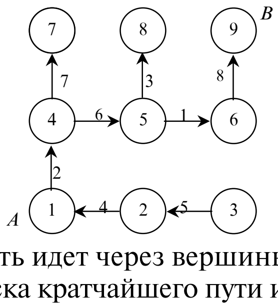
Из вершин 6, 7 и 8 выберем вершину 6. Тогда $l_B = 17$. $l_6^* = 9$.

$l_A^* = 0$, $l_2^* = 4$, $l_3 = 9$, $l_4^* = 2$, $l_5^* = 8$, $l_6^* = 9$, $l_7 = 9$, $l_8 = 11$, $l_B = l_9 = 17$.

Наконец, из вершин 7, 8 и 9(B) выбираем вершину с минимальной меткой. Это вершина 7. Все метки сохраняют свои значения. $l_7^* = 9$.

Остались вершины 8-я и B (9-я). Их метки делаются постоянными, сохраняя свои значения. $l_8^* = 11$, $l_B^* = 17$. Таким образом, $l_B^* = 17$.

Задача решена. Найдены длины кратчайших путей. Для отыскания самих путей надо, начиная с вершины B, для каждой вершины j отметить такое ребро (i, j), для которого $l_i + a_{ij} = l_j$. Здесь ребра ориентированы от i к j (!). В нашем примере получим следующие отмеченные (см. чертеж) ребра. При этом в каждую вершину, кроме начальной, входит по крайней мере одно ребро.

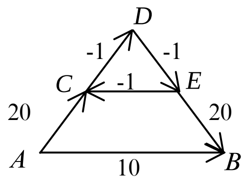


Кратчайший путь идет через вершины $A \to 4 \to 5 \to 6 \to B$.

В процессе поиска кратчайшего пути из A в B определяются и кратчайшие пути из A в любую другую вершину графа. Трудоемкость этого алгоритма Cn^2, где n — число вершин, C — константа.

В реальных экономических задачах могут возникать ситуации, когда некоторые ребра имеют отрицательную длину. Например, при перевозке попутного груза из i в j можно не только не тратить денег на переезд, но и получать прибыль. В этом случае, необходимо определить наличие в графе циклов отрицательной длины. Если таких циклов нет, то алгоритм можно применять практически без изменений. Отличие состоит в том, что может измениться метка со значком *. В графе без отрицательных длин ребер такие метки мы называли постоянными, и они не могли меняться. Здесь такая метка может измениться. В этом случае мы убираем у нее * и продолжаем работу алгоритма. Если же в графе существует цикл отрицательной длины (см. чертеж), то кратчайшего пути может не существовать.

Рассмотрим пример, показывающий, как это может произойти.



На рисунке видно, что длина пути из A в B, состоящего из одного ребра, равна 10. Если же сначала пойти из A в C, затем N раз пройти по циклу CDE, а лишь потом пойти в B, то длина пути будет равна $(38 — 3N)$. Попадая на цикл, можем прокручиваться по нему неоднократно, получая тем самым длину пути из A в B меньше любого, наперед заданного, числа.

Задачи для самостоятельного решения

Задан граф с 20 вершинами, расположенными в виде прямоугольника 4×5 и изображенными кружочками. Каждая вершина соединена с соседями справа, слева, снизу, сверху неориентированными ребрами. Длины ребер указаны числами, находящимися между вершинами. Найти длины кратчайших путей из левой верхней вершины в каждую вершину графа.

1.

°	5	°	7	°	3	°	7	°
2		4		3		6		3
°	4	°	6	°	5	°	6	°
6		5		3		3		2
°	5	°	2	°	5	°	6	°
7		4		7		4		7
°	2	°	6	°	2	°	2	°

2.

°	2	°	6	°	7	°	6	°
3		7		4		7		5
°	4	°	2	°	4	°	2	°
3		3		6		6		3
°	6	°	6	°	5	°	5	°
7		3		4		7		3
°	7	°	5	°	6	°	2	°

3.

°	2	°	5	°	7	°	4	°
2		3		5		5		5
°	3	°	7	°	4	°	4	°
3		3		5		3		5
°	7	°	4	°	5	°	6	°
3		5		7		5		5
°	4	°	3	°	2	°	2	°

4.

°	6	°	3	°	3	°	4	°
2		6		2		2		7
°	4	°	7	°	2	°	2	°
3		4		5		7		3
°	3	°	3	°	4	°	5	°
4		6		6		4		7
°	5	°	6	°	6	°	6	°

5.

°	2	°	3	°	2	°	3	°
3		7		4		4		2
°	2	°	3	°	6	°	3	°
6		6		4		4		4
°	2	°	7	°	2	°	5	°
5		3		6		6		7
°	2	°	6	°	2	°	4	°

6.

°	2	°	4	°	6	°	7	°
6		2		7		3		3
°	6	°	6	°	4	°	2	°
5		3		4		3		4
°	6	°	2	°	7	°	5	°
4		5		7		4		2
°	4	°	3	°	6	°	5	°

Ответы

1.

0	5	12	15	22
2	6	12	17	23
8	11	13	18	24
15	15	20	22	24

-1	1	2	3	4
1	6	7	8	9
6	7	12	13	14
11	12	13	14	19

2.

0	2	8	15	20
3	7	9	13	15
6	10	15	19	18
13	13	18	23	21

-1	1	2	3	10
1	6	7	8	9
6	7	8	9	10
11	12	17	20	15

3.

0	2	7	14	18
2	5	12	16	20
5	8	12	17	23
8	12	15	17	19

-1	1	2	3	4
1	2	7	8	9
6	7	12	13	14
11	16	17	18	19

4.

0	6	9	12	16
2	6	11	13	15
5	8	11	15	18
9	14	17	19	25

-1	1	2	3	4
1	6	3	8	9
6	11	12	13	10
11	12	13	14	15

5.

0	2	5	7	10
3	5	8	11	12
9	11	12	14	16
14	14	18	20	23

-1	1	2	3	4
1	6	7	4	5
6	7	8	13	10
11	12	13	14	15

6.

0	2	6	12	18
6	4	10	14	16
11	7	9	16	20
15	12	15	20	22

-1	1	2	3	10
1	2	7	8	9
6	7	12	13	10
11	12	17	14	15

§4. ЗАДАЧА О НАЗНАЧЕНИЯХ

Предположим, в отделе есть *n* должностей и *n* кандидатов на эти должности. Как назначить кандидатов на должности оптимальным образом?

Какую информацию мы будем использовать? Если мы знаем, на каких должностях может работать каждый человек, то начальные данные можно задать в виде квадратной матрицы *C*, имеющей размерность $(n \times n)$. Элементы этой матрицы — нули и единицы.

$$c_{ij} = \begin{cases} 1, & \text{если } i\text{-й человек может работать на } j\text{-й должности;} \\ 0, & \text{если } i\text{-й человек не может работать на } j\text{-й должности.} \end{cases}$$

Существует ли такое распределение работ, чтобы каждый человек справлялся с работой на своей должности? Как лучше назначить людей на работы?

Назначение человека на должность можно рассматривать, как выбор определенного элемента матрицы С. Каждый человек должен работать только на одной должности, и на каждой должности должен работать только один человек. Поэтому в каждой строчке и в каждом столбце матрицы С должно быть выбрано ровно по одной единице. Получили следующую математическую задачу:

В матрице $C(n \times n)$ выбрать n элементов, стоящих в разных строках и разных столбцах, так, чтобы их сумма была максимальной[3].

В идеале эта сумма должна быть равна *n*. Это возможно в том случае, если для каждого *к* $(1 \le k \le n)$ и любых *к* должностей, найдется не менее *к* человек, каждый из которых может справиться хотя бы с одной из этих *к* должностей (теорема Холла).

Рассмотрим другие способы рассмотрения этой задачи. Зададим распределение по должностям с помощью матрицы $X(n \times n)$, состоящей из нулей и единиц.

$$x_{ij} = \begin{cases} 1, & \text{если } i\text{-й человек будет назначен на } j\text{-ю должность;} \\ 0, & \text{если } i\text{-й человек не будет назначен на } j\text{-ю должность} \end{cases}$$

---

[3] При другой практической ситуации эту сумму надо минимизировать.

Каждый человек должен работать только на одной должности, и на каждой должности должен работать только один человек. Поэтому в каждой строчке и в каждом столбце матрицы X должно быть ровно по одной единице, а остальные элементы равны нулю. Следовательно, матрица X должна удовлетворять следующим условиям:

$$\sum_{i=1}^{n} x_{ij} = 1 \quad (j = 1,2,\ldots, n). \tag{1}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \quad (i = 1,2,\ldots,n). \tag{2}$$

$$x_{ij} \in \{0,1\}. \tag{3}$$

Рассмотрим сумму

$$\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}c_{ij}. \tag{4}$$

Если элементы матрицы C нули и единицы, то эта сумма равна количеству людей, назначенных на те должности, с которыми они справятся. Эту сумму мы хотим максимизировать. Возникла задача: среди всех матриц X, удовлетворяющих условиям (1), (2), (3), найти такую, для которой сумма (4) имеет наибольшее значение. Если условие (3) заменить условием

$$0 \le x_{ij} \le 1, \tag{5}$$

то возникает задача линейного программирования, где $L = \sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}c_{ij}$ — целевая функция, которую необходимо максимизировать, условия (1), (2), (5) задают систему линейных ограничений. Можно доказать, что среди оптимальных решений этой задачи обязательно есть целочисленные решения. Для таких решений условие (3) будет выполнено.

Возможны и другие постановки данной задачи. Если можно оценить эффективность любого назначения и дать количественную оценку работы, то матрица C может содержать любые числа, а не только 0 и 1. В этом случае c_{ij} — результат работы i—го человека на j-й должности. Если элементы матрицы C произвольные числа, то сумма (4) — некий критерий эффективности назначений. Если матрица C задает наши прибыли в связи с назначениями, то ищем максимум функции $L = \sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}c_{ij}$, если же издержки — минимум.

Другой пример. Нам необходимо распределить заказы между фирмами. Если c_{ij} плата, которую мы платим фирме i за выполнение заказа j, тогда функцию затрат L надо сводить к минимуму. Здесь предполагается, что каждая фирма выполняет заказ качественно. Если же c_{ij} — цена, которую платит нам фирма i за получение заказа j, то функцию L необходимо максимизировать.

Полученная задача является также задачей линейного программирования. Как и всякую задачу линейного программирования, ее можно решить симплекс-методом.

Таким образом, у нас есть две эквивалентные формулировки задачи о назначениях.

1) В матрице $C(n\times n)$ выбрать n элементов, стоящих в разных строках и разных столбцах, так, чтобы их сумма была максимальной (минимальной).

2) Среди всех матриц X, удовлетворяющих условиям (1), (2), (3), найти такую, для которой сумма (4) имеет наибольшее (наименьшее) значение.

Мы рассмотрим алгоритм, названный ***венгерским.***

В основе венгерского алгоритма лежит лемма об оптимальности, позволяющая в ряде случаев гарантировать оптимальность найденного решения, и эквивалентные преобразования матрицы стоимостей, позволяющие воспользоваться леммой.

Лемма об оптимальности. *Если в матрице $C(n\times n)$ нет отрицательных элементов и удалось выбрать n нулей, стоящих в разных строках и разных столбцах, то этот выбор оптимален в задаче на минимум.*

Доказательство леммы очевидно. В самом деле, сумма любых элементов матрицы неотрицательна. Сумма выбранных нами чисел равна 0, так как все выбранные числа равны 0. Поэтому нельзя выбрать элементы с меньшей суммой. Наш выбор оптимален.

Лемма правильна, но на первый взгляд бесполезна, так как в матрице может быть мало или совсем не быть нулевых элементов.

Применение леммы обеспечивается возможностью преобразовывать матрицу, создавая в нужных местах нулевые элементы.

Что произойдет, если мы от какой-нибудь строки матрицы C отнимем число *a*? Так как в этой строке должен быть выбран один и только один элемент, то стоимость любого варианта назначений уменьшится на *a*. Но если стоимость любого варианта изменилась на одну и ту же величину, то отношение предпочтения между вариантами не изменилось. Если два варианта стоили по старой матрице одинаково, то они будут стоить одинаково и по новой матрице. Если один из сравниваемых вариантов по старой матрице стоил меньше (больше) другого, то и по новой матрице он стоит на ту же величину меньше (больше). Эти рассуждения являются обоснованием для следующих преобразований матрицы стоимостей *C*.

Эквивалентные преобразования. *Мы имеем право от любой строки (столбца) матрицы C отнять (прибавить) любое число. Решение, оптимальное для одной матрицы, остается оптимальным для всех матриц, получаемых при любой последовательности эквивалентных преобразований матрицы C.*

При решении венгерским алгоритмом задачи о назначениях на минимум мы, имея в виду лемму об оптимальности, будем работать с матрицами без отрицательных элементов и выбирать в этих матрицах только нулевые элементы. Если мы сможем выбрать по одному нулевому элементы в каждой строке и каждом столбце матрицы (может быть уже преобразованной с помощью эквивалентных преобразований), то мы нашли оптимальное решение как для последней, так и для первоначальной матрицы. Если мы не сможем выбрать решение, состоящее только из нулевых элементов, то матрицу надо преобразовать.

Алгоритм решения задачи о назначениях на минимум

Шаг 1. Преобразуем матрицу. От каждой строки отнимем минимальный элемент этой строки. В результате все элементы матрицы будут неотрицательны, и в каждой строке появится хотя бы один нуль.

Шаг 2. Аналогично вычитаем минимальный элемент из каждого столбца получившейся матрицы. В результате в каждом столбце и в каждой строке появился хотя бы один *0*.

Шаг 3. Осуществляем назначения, выбирая и отмечая звездочкой нули, стоящие в разных строках и разных столбцах. Нули можно выбирать произвольно, но лучше в первую очередь, выбирать те, которые единственны в своих строках или столбцах [4].

Если удастся выбрать *n* нулей, стоящих в разных строках и разных столбцах, то задача решена. Эти нули соответствуют оптимальному назначению. Необходимо подсчитать значение целевой функции *по первоначальной* матрице C.

Если нулей меньше [5], то переход к шагу 4.

Рассмотрим приведенные три шага на примере.

Пример. Пусть задана матрица C. Элементы этой матрицы определяют затраты на обучение кандидатов. А именно: если i-й человек будет назначен на j-ю должность, то для его переобучения необходимо затратить сумму c_{ij} (здесь учитываются образование i-го кандидата, опыт, необходимые навыки для выполнения j-й должности и пр.). Решим задачу о назначениях, минимизируя целевую функцию (4).

Так как решаем задачу на минимум, то надо выбрать 5 элементов матрицы C таким образом, чтобы их сумма была минимальна. Количество возможных вариантов выбора равно *5!* (в общем случае *n!*).

$$C = \begin{pmatrix} 7 & 9 & 3 & 4 & 0 \\ 8 & 5 & 1 & 2 & 4 \\ 6 & 3 & 2 & 5 & 3 \\ 3 & 7 & 1 & 8 & 6 \\ 5 & 4 & 2 & 0 & 3 \end{pmatrix}.$$

Произведем эквивалентные преобразования матрицы так, чтобы в каждой строке и каждом столбце были нули. Отнимем от 2-й, 3-й, и 4-й строки соответственно 1, 2, 1. Затем, 2 и 1 от первого и второго столбца. Отметим в каждой строке и каждом столбце преобразованной матрицы нули.

---

[4] Единственный выбор не может быть ошибкой!

[5] Этот шаг выполняется, если мы не смогли отметить нули во всех строках и столбцах матрицы. При этом мы заранее не знаем, почему это произошло – то ли такие нули выбрать невозможно, то ли мы плохо их искали.

$$\begin{array}{cc}\begin{pmatrix}7 & 9 & 3 & 4 & 0\\ 8 & 5 & 1 & 2 & 4\\ 6 & 3 & 2 & 5 & 3\\ 3 & 7 & 1 & 8 & 6\\ 5 & 4 & 2 & 0 & 3\end{pmatrix} & \begin{array}{c}\\ -1\\ -2\\ -1\\ \\ \end{array}\end{array} \Rightarrow \begin{array}{c}\begin{pmatrix}7 & 9 & 3 & 4 & 0\\ 7 & 4 & 0 & 1 & 3\\ 4 & 1 & 0 & 3 & 1\\ 2 & 6 & 0 & 7 & 5\\ 5 & 4 & 2 & 0 & 3\end{pmatrix}\\ \begin{array}{ccccc}-2 & -1 & & & \end{array}\end{array} \Rightarrow \begin{pmatrix}5 & 8 & 3 & 4 & 0^*\\ 5 & 3 & 0^* & 1 & 3\\ 2 & 0^* & 0 & 3 & 1\\ 0^* & 5 & 0 & 7 & 5\\ 3 & 3 & 2 & 0^* & 3\end{pmatrix}.$$

Мы смогли отметить пять нулей, стоящих в разных столбцах и разных строках. Следовательно, задача решена. В первой строке отмечен нуль в 5 столбце. Поэтому первого человека назначим на 5 должность. Во второй строке отмечен нуль в третьем столбце. Поэтому второго человека назначим на третью должность, 3-го на вторую, 4-го на первую и, наконец, 5-го на 4-ю. Значение целевой функции будет минимальным. Вычислим его по первоначальной матрице

$$L = a_{15} + a_{23} + a_{32} + a_{41} + a_{54} = 0 + 1 + 3 + 3 + 0 = 7.$$

Не всегда решение получается так сразу. Рассмотрим дальнейшие шаги алгоритма.

Шаг 4. «Расстановка меток». Строки и столбцы, в которых нет отмеченных звездочкой нулей, будем называть «недовольными».

Отметим «недовольную» строку.

4.1. Если отмечена строка, то отметим все те столбцы, у которых в данной строке есть нули.

4.2. Если отмечен какой-либо столбец, то отметим ту строку, у которой в этом столбце есть 0^*.

Шаги 4.1 и 4.2 повторяются до тех пор, пока в этой процедуре появляются новые строки и столбцы.

Здесь могут возникать две различные ситуации. Назовем их ситуацией «прорыва» и ситуацией «узкое место».

Ситуация «прорыва». Такая ситуация возникает в том случае, если, начав со строки, мы в какой-то момент отметили «недовольный» столбец. Это значит, что мы не очень удачно выбирали нули. Из имеющихся нулей можно было выбрать и отметить больше нулей, чем мы выбрали. Покажем, как перераспределить часть звездочек и назначить еще одну. Помеченному «недовольному» столбцу должна дать звездочку та строка, которая его пометила. Если при этом в строке окажется более одной звездочки, то надо убрать звездочку в столбце, пометившем эту строку. Это перераспределение звездочек может затронуть и другие помеченные строки и столбцы. Если в помеченном столбце нет звездочки, то ее должна дать та строка, которая его пометила. Если в строке окажется более одной звездочки, то надо убрать звездочку у столбца, пометившего эту строку. При этом строки и столбцы, получившие звездочку один раз, всегда останутся со звездочкой. Она может перейти на другое место в строке или столбце, но не должна пропасть.

Если, в результате перераспределения, число отмеченных звездочками нулей станет равно n, то задача решена. Если же меньше n, то необходимо повторить сначала шаг 4.

Ситуация «узкое место». Она возникает в том случае, если процедура расстановки меток завершилась, но при этом не был отмечен ни один «недовольный» столбец. В этом случае число помеченных строк больше числа помеченных столбцов. Поэтому по данной матрице осуществить все назначения по нулевым элементам невозможно. Изменим матрицу стоимостей. Найдем минимальный элемент среди чисел, стоящих в отмеченных строках вне отмеченных столбцов. Этот элемент вычтем от всех отмеченных строк и прибавим ко всем отмеченным столбцам. В результате в матрице появятся новые нули. Вернемся к шагу 3 [6].

Пример. Предположим, решая задачу о назначениях, получили следующую матрицу.

$$\begin{pmatrix} 2 & 0^* & 3 & 5 & 7 \\ 0 & 3 & 4 & 6 & 2 \\ 0^* & 3 & 1 & 1 & 1 \\ 5 & 1 & 4 & 0 & 0^* \\ 0 & 1 & 0^* & 5 & 0 \end{pmatrix}.$$

Здесь нам удалось расставить всего 4 звездочки. Сначала мы отметим строку 2, которой не хватило 0*. Из-за нее мы отметим столбец 1. Затем отметим третью строчку. Больше отметить ничего нельзя. Отмечен столбец 1, но его нельзя назвать недовольным. Поэтому имеет место ситуация «узкое место».

---

[6] Можно перейти к шагу 4 и преобразовать решение. Должна возникнуть ситуация прорыва. Перейдя к шагу 3, мы просто отбросили старое решение и начали поиск нового.

$$\begin{array}{l} \begin{pmatrix} 2 & 0^* & 3 & 5 & 7 \\ 0 & 3 & 4 & 6 & 2 \\ 0^* & 3 & 1 & 1 & 1 \\ 5 & 1 & 4 & 0 & 0^* \\ 0 & 1 & 0^* & 5 & 0 \end{pmatrix} \begin{matrix} \\ \leftarrow \\ \leftarrow \\ \\ \\ \end{matrix}. \\ \quad\uparrow \end{array}$$

И второй, и третий человек хотят работать только на первой должности. Но 2 > 1. Поэтому выбрать нули и во второй и в третьей строке так, чтобы они стояли в разных столбцах, невозможно. Изменяем матрицу стоимостей. Минимальный элемент строк 2 и 3, вне первого столбца, равен единице. Отнимем единицу от строк 2 и 3 и прибавим к столбцу 1.

$$\begin{array}{l} \begin{pmatrix} 2 & 0 & 3 & 5 & 7 \\ 0 & 3 & 4 & 6 & 2 \\ 0 & 3 & 1 & 1 & 1 \\ 5 & 1 & 4 & 0 & 0 \\ 0 & 1 & 0 & 5 & 0 \end{pmatrix} \begin{matrix} \\ -1 \\ -1 \\ \\ \\ \end{matrix} \\ \;+1 \end{array} \Rightarrow \begin{pmatrix} 3 & 0^* & 3 & 5 & 7 \\ 0^* & 2 & 3 & 5 & 1 \\ 0 & 2 & 0 & 0 & 0^* \\ 6 & 1 & 4 & 0^* & 0 \\ 1 & 1 & 0^* & 5 & 0 \end{pmatrix}.$$

В последней матрице уже можно осуществить пять назначений. Найдено оптимальное решение. Запишем его в виде матрицы.

$$X = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

Алгоритм решения задачи на максимум

Для решения задачи на максимум можно модифицировать алгоритм решения задачи на минимум. Мы поступим по-другому. Сведем задачу на максимум к задаче на минимум.

$\max\limits_{x \in S} f(x) = A$ означает, что

1) $\exists x_0 \in S$, такое, что $f(x_0) = A$;

2) $\forall x \in S$ справедливо: $f(x) \le A$.

Последнее неравенство умножим на (-1): $\forall x \in S, \quad -f(x) \geq -A.$ Следовательно, если число *A* является максимумом для функции *f(x)*, то (*—A)* является минимумом для функции *-f(x)* на том же множестве допустимых вариантов.

$$\min_{x \in S}(-f(x)) = -A.$$

Точки максимума и минимума при этом совпадают.

Таким образом, чтобы решить задачу на максимум с матрицей *C*, достаточно решить задачу на минимум с матрицей *(-C)*. Для этого все элементы матрицы *C* умножим на -1, а затем для новой матрицы решим задачу на минимум.

Обычно, для решения задачи на максимум, меняют первый шаг изложенного выше алгоритма решения задачи на минимум.

Шаг 1. Преобразовать матрицу, заменив каждый элемент матрицы разностью максимального элемента его строки и самого элемента.

Остальные шаги алгоритма не меняются.

Трудоемкость решения задачи о назначениях порядка n^3, что позволяет с помощью ЭВМ решать задачи практически любой размерности.

Задачи для самостоятельного решения

Решить задачу о назначениях на максимум.

1.

14	6	6	10	7	8
11	3	6	2	6	14
14	9	11	13	4	15
2	5	7	15	7	6
14	4	1	1	12	13
3	8	15	1	15	3

2.

11	9	9	8	10	4
6	14	10	6	9	8
7	13	13	4	11	14
2	3	9	10	2	9
8	14	10	11	12	5
4	15	2	4	1	4

3.

1	2	13	15	14	8
13	1	9	14	5	10
9	10	13	4	4	4
2	2	8	13	9	12
6	7	10	14	10	13
2	9	2	13	14	14

4.

3	2	13	11	1	7
5	7	11	6	12	15
8	4	11	6	9	10
6	2	4	7	7	5
14	10	7	13	11	12
12	12	12	4	3	13

5.

6	3	2	3	14	8
12	2	11	2	4	12
6	5	15	13	3	2
9	4	11	1	14	12
12	8	2	8	11	9
7	7	3	6	2	13

6.

1	6	10	12	8	9
4	3	7	5	4	14
10	6	1	2	13	1
5	7	5	2	9	12
14	14	15	10	14	1
4	8	10	2	15	14

Ответы

1. $L = 79$. Назначения: $x_{26} = 1$; $x_{32} = 1$; $x_{44} = 1$; $x_{63} = 1$; $x_{11} = 1$; $x_{55} = 1$.
2. $L = 72$. Назначения: $x_{11} = 1$; $x_{36} = 1$; $x_{44} = 1$; $x_{55} = 1$; $x_{62} = 1$; $x_{23} = 1$.
3. $L = 76$. Назначения: $x_{21} = 1$; $x_{32} = 1$; $x_{13} = 1$; $x_{44} = 1$; $x_{56} = 1$; $x_{65} = 1$.
4. $L = 70$. Назначения: $x_{26} = 1$; $x_{51} = 1$; $x_{62} = 1$; $x_{13} = 1$; $x_{35} = 1$; $x_{44} = 1$.
5. $L = 71$. Назначения: $x_{15} = 1$; $x_{21} = 1$; $x_{34} = 1$; $x_{52} = 1$; $x_{66} = 1$; $x_{43} = 1$.
6. $L = 73$. Назначения: $x_{14} = 1$; $x_{26} = 1$; $x_{53} = 1$; $x_{65} = 1$; $x_{31} = 1$; $x_{42} = 1$.

Решить задачу о назначениях на минимум.

7.
$$\begin{matrix} 1 & 5 & 4 & 10 & 10 & 11 \\ 2 & 12 & 6 & 12 & 9 & 11 \\ 5 & 2 & 13 & 14 & 7 & 1 \\ 14 & 7 & 14 & 14 & 2 & 1 \\ 1 & 9 & 8 & 8 & 4 & 2 \\ 1 & 11 & 2 & 4 & 6 & 2 \end{matrix}$$

8.
$$\begin{matrix} 7 & 4 & 1 & 14 & 6 & 8 \\ 5 & 3 & 6 & 14 & 3 & 13 \\ 8 & 5 & 1 & 9 & 14 & 14 \\ 2 & 15 & 2 & 11 & 10 & 8 \\ 6 & 10 & 2 & 8 & 12 & 11 \\ 5 & 7 & 5 & 3 & 13 & 8 \end{matrix}$$

9.
$$\begin{matrix} 8 & 12 & 14 & 6 & 10 & 3 \\ 14 & 7 & 5 & 11 & 14 & 13 \\ 10 & 7 & 14 & 9 & 13 & 8 \\ 14 & 13 & 3 & 15 & 9 & 11 \\ 12 & 6 & 5 & 9 & 12 & 5 \\ 10 & 11 & 14 & 12 & 1 & 5 \end{matrix}$$

10.
$$\begin{matrix} 4 & 9 & 2 & 4 & 11 & 4 \\ 10 & 8 & 13 & 8 & 14 & 7 \\ 3 & 14 & 10 & 9 & 13 & 8 \\ 3 & 2 & 12 & 9 & 11 & 1 \\ 11 & 14 & 1 & 14 & 12 & 3 \\ 9 & 11 & 15 & 8 & 9 & 10 \end{matrix}$$

11.
$$\begin{matrix} 9 & 4 & 1 & 12 & 9 & 14 \\ 1 & 8 & 1 & 2 & 9 & 1 \\ 9 & 1 & 9 & 2 & 8 & 2 \\ 11 & 8 & 14 & 2 & 14 & 10 \\ 13 & 7 & 10 & 1 & 10 & 11 \\ 3 & 9 & 12 & 10 & 5 & 3 \end{matrix}$$

12.
$$\begin{matrix} 4 & 6 & 10 & 2 & 15 & 11 \\ 12 & 10 & 5 & 9 & 12 & 14 \\ 1 & 11 & 10 & 13 & 1 & 1 \\ 2 & 2 & 2 & 14 & 14 & 9 \\ 9 & 5 & 14 & 15 & 14 & 9 \\ 5 & 1 & 14 & 6 & 7 & 15 \end{matrix}$$

Ответы

7. $L = 16$. Назначения: $x_{21} = 1$; $x_{32} = 1$; $x_{45} = 1$; $x_{56} = 1$; $x_{64} = 1$; $x_{13} = 1$.
8. $L = 23$. Назначения: $x_{16} = 1$; $x_{25} = 1$; $x_{41} = 1$; $x_{64} = 1$; $x_{32} = 1$; $x_{53} = 1$.
9. $L = 32$. Назначения: $x_{31} = 1$; $x_{43} = 1$; $x_{65} = 1$; $x_{14} = 1$; $x_{22} = 1$; $x_{56} = 1$.
10. $L = 26$. Назначения: $x_{31} = 1$; $x_{53} = 1$; $x_{65} = 1$; $x_{22} = 1$; $x_{46} = 1$; $x_{14} = 1$.
11. $L = 18$. Назначения: $x_{13} = 1$; $x_{21} = 1$; $x_{32} = 1$; $x_{44} = 1$; $x_{55} = 1$; $x_{66} = 1$.
12. $L = 20$. Назначения: $x_{14} = 1$; $x_{35} = 1$; $x_{41} = 1$; $x_{62} = 1$; $x_{23} = 1$; $x_{56} = 1$.

§5. ТРАНСПОРТНАЯ ЗАДАЧА

В стоимости многих товаров большую долю составляют транспортные расходы. Из многих моделей, которые относятся к минимизации транспортных расходов, одна получила настолько большое распространение, что именно ее имеют в виду, когда говорят о *транспортной задаче*. Эта модель используется при разработке плана перевозок одного вида продукции из нескольких пунктов отправления в несколько пунктов потребления. Задача ставится следующим образом.

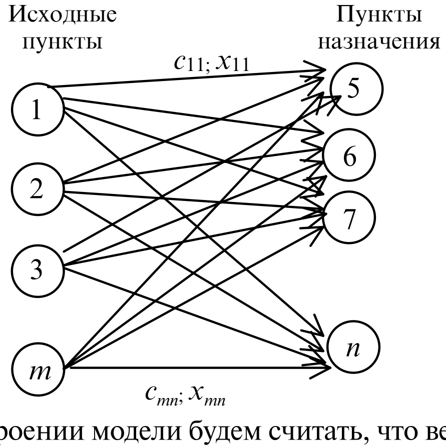
Есть m пунктов, где производится или хранится товар (пункты производства), и n пунктов потребления однородного товара. Пусть:

$a_1, a_2, a_3,\dots, a_m$ — количество единиц товара, имеющегося в соответствующих пунктах производства;

$b_1, b_2, b_3,\dots, b_n$ — количество единиц товара, требующееся в пунктах потребления;

$c_{11}, c_{12}, c_{13},\dots, c_{mn}$ — стоимость перевозки единицы товара от i-го производителя к j-му потребителю.

Необходимо найти такой план перевозок, чтобы суммарные транспортные расходы были наименьшими.



При построении модели будем считать, что величина транспортных расходов на каждом маршруте прямо пропорциональна объему перевозимой продукции.

Задать план перевозок можно с помощью матрицы $\boldsymbol{X} = \{x_{ij}\}$, где x_{ij} — количество единиц товара, которое планируется перевезти из пункта производства i в пункт потребления j.

Каким ограничениям должны удовлетворять эти переменные?

Из условия ясно, что

$$x_{ij} \geq 0 \text{ при любых } i, j.$$

Очевидно также, что перевозки, производимые из 1-го пункта, в сумме составляют величину a_1, то есть

$$x_{11} + x_{12} + x_{13} + \ldots + x_{1n} = a_1.$$

Аналогичное равенство справедливо для каждого пункта производства:

$x_{i1} + x_{i2} + x_{i3} + \ldots + x_{in}$ — общее количество товара, которое будет вывезено из i -го пункта.

$$\sum_{j=1}^{n} x_{ij} = a_i; \qquad i = 1,\ 2,\ \ldots,\ m.$$

$x_{1j} + x_{2j} + x_{3j} + \ldots + x_{mj}$ — общее количество товара, которое необходимо привезти в пункт j.

$$\sum_{i=1}^{m} x_{ij} = b_j; \qquad j = 1,\ 2,\ \ldots,\ n.$$

Так как стоимость перевозки линейно зависит от количества перевозимого товара, то функция издержек определяется как $L = c_{11}x_{11} + c_{12}x_{12} + c_{13}x_{13} + \ldots + c_{mn}x_{mn}$.

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

Целевую функцию необходимо минимизировать.

Окончательно наша модель выглядит следующим образом:

Минимизировать

$$L = \sum_{i,j} c_{ij} x_{ij}$$

при следующих ограничениях

$$\sum_{j=1}^{n} x_{ij} = a_i \ \ (i = 1, 2, \ldots, m);$$
$$\sum_{i=1}^{m} x_{ij} = b_j \ \ (j = 1, 2, \ldots, n); \tag{2}$$

$$x_{ij} \geq 0,\ (i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n). \tag{3}$$

Имеет ли задача решение? В каком случае?

Необходимо минимизировать линейную функцию (1) при линейных ограничениях (2), (3). В общем случае множество допустимых вариантов может быть пустым. Если множество вариантов не пусто, то целевая функция может быть ограничена (в этом случае есть оптимальное решение) и не ограничена (в этом случае для каждого решения есть лучшее). В поставленной транспортной задаче (1), (2), (3) множество допустимых вариантов всегда ограничено, каждое x_{ij} неотрицательно и $x_{ij} \leq \min\{a_i, b_j\}$. По теореме Вейерштрасса непрерывная (в частности линейная) функция многих переменных на замкнутом, ограниченном множестве ограничена и достигает в некоторой точке своего минимума. Выясним, в каком случае существует решение.

Предположим, что X— допустимое решение задачи (1), (2), (3). Обозначим сумму всех перевозок S.

$$S = \sum_{i,j} x_{ij} .$$

Тогда, с одной стороны,

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

с другой стороны,

$$S = \sum_{j=1}^{n} (\sum_{i=1}^{m} x_{ij}) = \sum_{j=1}^{n} b_j .$$

То есть задача (1), (2), (3) имеет решение только в том случае, если

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j . \tag{4}$$

Таким образом, это условие является необходимым. Оно требует равенства общего запаса продукта и общей потребности в нем. Часто это условие называется условием баланса. Покажем, что оно же является и достаточным. Пусть условие (4) выполняется. Возьмем

$$x_{ij} = \frac{a_i b_j}{S} . \tag{5}$$

Легко видеть, что такой план удовлетворяет всем условиям (2), (3). Следовательно, множество решений системы не пусто, это означает, что транспортная задача всегда имеет решение при выполнении условия баланса (4).

Итак, мы должны забрать весь товар и удовлетворить все заявки. Такая задача называется *сбалансированной*, или *замкнутой*. На практике условие баланса может нарушаться. Суммарные потребности могут не совпадать с наличием товара. В этом случае задача называется *открытой*, или *несбалансированной*. Если задача несбалансированна, то тем или иным способом она сводится к сбалансированной. Рассмотренные ниже алгоритмы решения относятся только к сбалансированным задачам[7].

Рассмотрим *пример.* Имеется 3 пункта производства и 4 пункта потребления. В пунктах производства находится 50, 20 и 30 единиц продукции, а в пунктах потребления требуется 10, 40, 15 и 35 единиц. Условие баланса выполняется, 50 + 20 + 30 = 10 + 40 + 15 + 35. Данные разместим в так называемой *транспортной таблице*, в которой строки соответствуют пунктам производства, а столбцы пунктам потребления. Таблица представлена матрицей стоимостей $\boldsymbol{C} = \{c_{ij}\}$. В данной таблице значения x_{ij} стоят в правом верхнем углу соответствующей ячейки, а в самой ячейке записывается значение стоимости. Иногда удобно расположить данные иначе, поменяв местами перевозки и стоимости. Это зависит от того, какой шаг алгоритма решения задачи рассматривается.

---

[7] Что делать, если задача не является сбалансированной? В случае нехватки товара невозможно удовлетворить спрос потребителей. Поэтому здесь необходимо некоторое административное решение. Может быть предпринята некоторая корректировка заявок, могут быть найдены новые поставщики или потребители товара. В ряде случаев может быть принято следующее решение. Имеется, например, 100 единиц товара, требуется 120. При любом распределении перевозок 20 единиц товара не хватает. Можно искать такой план перевозок, при котором нехватка товара распределится так, чтобы суммарные транспортные расходы были минимальными. Для поиска такого плана к матрице стоимостей C добавим фиктивную строку (или столбец). Припишем добавочной строке (или столбцу) нулевые стоимости перевозок и количество товара, делающее задачу сбалансированной.

	B_1	B_2	B_3	B_4	
A_1	x_{11} 3	x_{12} 5	x_{13} 9	x_{14} 4	50
A_2	x_{21} 7	x_{22} 1	x_{23} 6	x_{24} 8	20
A_3	x_{31} 0	x_{32} 9	x_{33} 2	x_{34} 3	30
	10	40	15	35	

Математическая модель для данной задачи имеет следующий вид:

Min $L = 3x_{11} + 5x_{12} + 9x_{13} + 4x_{14} + 7x_{21} + x_{22} + 6x_{23} + 8x_{24} + 9x_{32} + 2x_{33} + 3x_{34}$ при следующих ограничениях:

$$x_{11} + x_{12} + x_{13} + x_{14} = 50;$$
$$x_{21} + x_{22} + x_{23} + x_{24} = 20;$$
$$x_{31} + x_{32} + x_{33} + x_{34} = 30;$$
$$x_{11} + x_{21} + x_{31} = 10;$$
$$x_{12} + x_{22} + x_{32} = 40;$$
$$x_{13} + x_{23} + x_{33} = 15;$$
$$x_{14} + x_{24} + x_{34} = 35;$$
$$x_{ij} = 0 \ (i = 1,\ 2,\ 3;\ j = 1,\ 2,\ 3,\ 4).$$

Обычно эти ограничения не выписывают в явном виде, их держат в уме. Все расчеты записывают в транспортной таблице.

Как найти дешевый план перевозок? В плане, полученном с помощью формулы *(5)*, все перевозки положительны. Мы используем и самые дорогие, и самые дешевые элементы матрицы стоимостей. Попробуем найти план перевозок, используя в первую очередь самые дешевые возможности.

В таблице в клетках расположим значения матрицы стоимостей, а в верхнем правом углу — ненулевые значения перевозок. В первой строке выберем самую маленькую стоимость — это $c_{11} = 3$. Назначим возможно большую перевозку, допускаемую ограничениями на спрос и объем производства. Поскольку каждая перевозка в этой строке составляет не более 50 единиц товара, а в столбце не более 10, то назначаем перевозку $x_{11} = \min(50, 10) = 10$. Ограничения, представляемые первым столбцом, выполнены, поэтому в первом столбце остальные перевозки должны быть равны нулю, а в строке необходимо распределить еще 40 единиц товара. Следующая по малости величина в строке $c_{14} = 4$. Здесь наибольшее возможное значение перевозки равно 35, $x_{14} = 35$.

Аналогично остальные перевозки в четвертом столбце равны нулю, а в строке необходимо распределить еще пять единиц товара. Из оставшихся значений матрицы стоимостей в первой строке самое маленькое $c_{12} = 5$, не распределено еще 5 единиц. Положим $x_{12} = 5$. Теперь ограничения, представляемые первой строкой, выполнены, все неназванные переменные в первой строке полагаются равными нулю. Переходим к следующей строке. Повторяя аналогичные действия и учитывая, что значения некоторых переменных уже определены (равны нулю), положим $x_{22} = 20$. Теперь выполнены ограничения, представляемые второй строкой, полагаем остальные переменные в строке равными нулю, переходим к третьей строке. Здесь $x_{33} = 15$ и $x_{32} = 15$.

Получен некоторый план перевозок $X^{(1)}$, запишем его в таблице и подсчитаем его стоимость[8].

	B_1	B_2	B_3	B_4	
A_1	3 (10)	5 (5)	9	4 (35)	50
A_2	7	1 (20)	6	8	20
A_3	0	9 (15)	2 (15)	3	30
	10	40	15	35	

Стоимость плана $X^{(1)}$: $L^{(1)} = 3{\cdot}10 + 5{\cdot}5 + 4{\cdot}35 + 20 + 9{\cdot}15 + 2{\cdot}15 = 380$ (ед.).

При применении этого алгоритма мы получим план перевозок, содержащий не более чем $m + n - 1$ ненулевую перевозку[9]. План, содержащий ровно $m + n - 1$ ненулевых перевозок, называется невырожденным.

Рассмотренный алгоритм позволяет находить достаточно хороший план, который, к сожалению, не всегда оптимален, зато его поиск не требует больших временных затрат. Часто такой план является неплохим первым приближением. Алгоритмы такого рода, дающие достаточно разумные варианты, но не всегда приводящие к точному решению, называются *эвристическими*. Данный эвристический алгоритм относится к так называемым

---

[8] В клетках матрицы, где не указана величина перевозки, перевозка равна нулю.

[9] Число ненулевых перевозок связано с числом базисных переменных системы уравнений (см. симплекс-метод).

«жадным алгоритмам». В нем мы смотрим на один шаг вперед и выбираем решение, оптимальное на этом шаге. Этот алгоритм не гарантирует оптимальности, так как, истратив меньше на первом шаге, мы, может быть, больше истратим в дальнейшем.

Мы могли бы получить другой план, если бы применили тот же самый алгоритм, начав с другой строки (например, с третьей) или со столбцов, а не со строк.

c_1 x_1	c_2 x_2
c_3 x_3	c_4 x_4

Новый план перевозок:

c_1 x_1-d	c_2 x_2+d
c_3 x_3+d	c_4 x_4-d

Подсчитаем стоимость нового плана и сравним со стоимостью старого.

$L_{ст} = c_1x_1 + c_2x_2 + c_3x_3 + c_4x_4$, $L_{нов} = c_1(x_1 — \mathrm{d}) + c_2(x_2 + \mathrm{d}) + c_3(x_3 + \mathrm{d}) + c_4(x_4 — \mathrm{d}) = c_1x_1 + c_2x_2 + c_3x_3 + c_4x_4 + \mathrm{d}\ (c_2 + c_3 — (c_1 + c_4)) = L_{ст} + \mathrm{d}\ (c_2 + c_3 — (c_1 + c_4))$. Стоимость плана уменьшится, если сумма элементов главной диагонали больше суммы элементов побочной диагонали, и увеличится в противном случае. План станет дешевле, если уменьшать перевозки на диагонали с большей суммой стоимостей. Алгоритмы, в которых небольшими изменениями улучшается рассматриваемый вариант решения, называются локальными.

Обобщая вышесказанное, выделим в матрице стоимостей две произвольные строки и два столбца. Пусть это строки с номерами i_1, i_2 и столбцы k_1, k_2. Рассмотрим прямоугольник, образованный элементами матрицы стоимостей $c_{i_1k_1}$, $c_{i_1k_2}$, $c_{i_2k_1}$, $c_{i_2k_2}$.

$c_{i_1k_1}$	$c_{i_1k_2}$
$c_{i_2k_1}$	$c_{i_2k_2}$

Назовем прямоугольник *неправильным*, если на диагонали с большей суммой тарифов нет нулевых перевозок. Неправильный прямоугольник исправим, уменьшив перевозки на диагонали с большей суммой стоимостей и увеличив на диагонали с меньшей суммой. Величина, на которую изменяются перевозки, равна минимальной перевозке на диагонали с большей суммой тарифов. Такой план по-прежнему удовлетворяет условиям задачи.

В рассматриваемом примере, план $X^{(1)}$ может быть улучшен. Неправильный прямоугольник составляют 1-я, 3-я строки и 1-й, 2-й столбцы. $c_{11} + c_{32} = 3 + 9 = 12$, $c_{12} + c_{31} = 0 + 5 = 5$. $12 > 5$, а обе перевозки, отвечающие большей диагонали, положительны: $x_{11} = 10$, а $x_{32} = 15$.

3 ^{10}	5 ^{5}
0	9 ^{15}

Поскольку в плане все перевозки могут быть только неотрицательными, то перевозка x_{11} может быть уменьшена не более, чем на 10 единиц, а x_{32} не более, чем на 15. Min (10,15) = 10. Поэтому внутри данного четырехугольника перераспределим 10 единиц товара следующим образом. Перевозку x_{11} уменьшим на 10 единиц, тогда x_{12} увеличим на 10, x_{32} уменьшим на 10, а x_{31} увеличим на 10. Получим $x_{11} = 0$, $x_{12} = 15$, $x_{31} = 10$, $x_{32} = 5$.

3	5 ^{15}
0 ^{10}	9 ^{5}

Разность стоимостей диагоналей равна (3+9) — (0+5) = 7. В прямоугольнике перераспределено 10 единиц товара. Наш выигрыш в стоимости равен $10 \cdot 7 = 70$. Убедимся в этом, используя непосредственный посчет стоимости нового плана.

Обозначим новый план $X^{(2)}$. Его стоимость $L^{(2)} = 5 \cdot 15 + 4 \cdot 35 + 20 + 9 \cdot 5 + 2 \cdot 15 = 310$ (ед.). Здесь снова выделим неправильный прямоугольник c_{12}, c_{14}, c_{32}, c_{34} и изменяем план, перераспределив 5 единиц товара.

Неправильный прямоугольник.

5 ^{15}	4 ^{35}
9 ^{5}	3

Измененный прямоугольник.

5 ^{20}	4 ^{30}
9	3 ^{5}

Новый план $X^{(3)}$ дешевле предыдущего на 25 единиц. $L^{(3)} = 285$. Выпишем его в виде матрицы.

$$X^{(3)} = \begin{pmatrix} 0 & 20 & 0 & 30 \\ 0 & 20 & 0 & 0 \\ 10 & 0 & 15 & 5 \end{pmatrix}.$$

В плане $X^{(3)}$ все прямоугольники правильные. Итак, если в плане имеются неправильные прямоугольники, то он не оптимален. Можно ли утверждать обратное? Оказывается, нет. Возможна ситуация, когда все прямоугольники правильные, а план не является оптимальным. В этом случае изменения должны затронуть более четырех элементов матрицы перевозок.

Точные алгоритмы

Мы научились решать транспортную задачу эвристическими алгоритмами. Сначала с помощью жадного алгоритма мы получили первоначальный план, а потом с помощью локального алгоритма (исправление неправильных прямоугольников) попытались его улучшить. Но отсутствие неправильных прямоугольников не гарантирует оптимальность найденного решения.

Ниже мы рассмотрим два точных алгоритма — метод потенциалов и венгерский алгоритм. Оба метода являются некоторой модификацией симплекс-метода для решения транспортной задачи. Но излагаются эти методы так, что знание симплекс-метода не является необходимым для понимания. Оба метода опираются на лемму об оптимальности и эквивалентные преобразования матрицы стоимостей.

Лемма об оптимальности. *Если стоимость плана перевозок X, подсчитанная по матрице стоимостей C, равна нулю, а в матрице C нет отрицательных элементов, то этот план оптимален.*

Суммарная стоимость издержек L определяется как сумма произведений стоимости c_{ij} на соответствующую перевозку x_{ij}. Так как все перевозки неотрицательны по условиям задачи, а элементы матрицы стоимостей неотрицательны по условию леммы, то $L = \sum_{i,j} c_{ij} x_{ij}$ на любом плане. Следовательно, нулевой (по стоимости) план оптимален.

На первый взгляд кажется, что лемма об оптимальности бесполезна, так как в реальности не бывает планов перевозок с нулевой стоимостью. Чтобы ее применить, надо уметь преобразовывать матрицу стоимостей.

Следующие преобразования матрицы стоимостей назовем эквивалентными. От каждой строки и каждого столбца матрицы вычтем (или прибавим) некоторую постоянную величину. Для каждой строки (столбца) это может быть своя величина. Получим новую матрицу C'. Сравним стоимости произвольного плана, подсчитанные по матрице C и по матрице C'. Пусть

$$C = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \dots & \dots & \dots & \dots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{pmatrix}$$

матрица стоимостей. Стоимость плана перевозок в этом случае равна $L = \sum\limits_{i,j} c_{ij}\, x_{ij}$.

Проведем преобразования матрицы стоимостей. От i-й строки отнимем некоторое число a_i, а от j-го столбца отнимем некоторое число b_j.

$$\begin{array}{cc} \begin{array}{cccc} -\beta_1 & -\beta_2 & \dots & -\beta_m \end{array} & \\ C' = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \dots & \dots & \dots & \dots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{pmatrix} & \begin{array}{c} -\alpha_1 \\ -\alpha_2 \\ \vdots \\ -\alpha_n \end{array} \end{array}$$

Стоимость перевозок с матрицей стоимостей C' составит

$$L' = \sum_{i=1}^{m}\sum_{j=1}^{n}(c_{ij} - \alpha_i - \beta_j)\, x_{ij} = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}\, x_{ij} - \sum_{i=1}^{m}\sum_{j=1}^{n}(\alpha_i + \beta_j)\, x_{ij} =$$

$$= L - \sum_{i=1}^{m}\alpha_i \sum_{j=1}^{n} x_{ij} - \sum_{j=1}^{n}\beta_j \sum_{i=1}^{m} x_{ij} = L - \sum_{i=1}^{m}\alpha_i a_i - \sum_{j=1}^{n}\beta_j b_j .$$

То есть новая стоимость каждого плана отличается от прежней на величину, которая не зависит от перевозок x_{ij}. Эта величина одинакова для любого плана. Поэтому план, оптимальный для одной матрицы, оптимален и для другой.

Метод потенциалов

Пусть найден некоторый план *X*. Попробуем доказать его оптимальность. Для этого преобразуем матрицу стоимостей так, чтобы все перевозки осуществлялись по элементам нулевой стоимости. Тогда этот план по новой матрице будет иметь нулевую стоимость. Если в преобразованной матрице стоимостей не окажется отрицательных элементов, то согласно лемме об оптимальности этот план оптимален.

Рассмотрим следующее преобразование матрицы стоимостей. Вычтем из каждой строки i некоторое α_i, а из каждого столбца j некоторое β_j. α_i и β_j выберем так, чтобы для любой ненулевой перевозки x_{ij} выполнялось равенство

$$\alpha_i + \beta_j = c_{ij}\,.$$

Числа $\alpha_1, \alpha_2, \ldots, \alpha_m, \beta_1, \beta_2, \ldots, \beta_n$ обычно называют потенциалами строк и столбцов, а сам метод — методом потенциалов. Выписывая соответствующее уравнение для каждой ненулевой перевозки, получаем систему линейных уравнений. В этой системе число переменных равно $(m + n)$, а число уравнений равно числу ненулевых перевозок нашего плана.

Проанализируем полученную систему. Может ли эта система иметь единственное решение? Нет! Если потенциалы всех строк, то есть все α_i, увеличить, например, на 5, а потенциалы всех столбцов, то есть все β_j, уменьшить на 5, то новые потенциалы тоже удовлетворяют всем уравнениям системы. Если эта система имеет хотя бы одно решение, то она имеет бесконечное множество решений.

Найдем какое-либо частное решение системы. Например, положим[10] $\alpha_1 = 0$. На каждом следующем шаге выбираем уравнение, в котором одна переменная уже определена, а другая еще нет, и определяем новую переменную. Решив систему, вычтем от каждой строки и каждого столбца матрицы соответствующий потенциал. Матрица стоимостей изменится. В ней появятся нулевые стоимости там, где имеются ненулевые перевозки. Стоимость этого плана по этой измененной матрице равна 0. В случае отсутствия в матрице стоимостей отрицательных элементов найденный план оптимален.

---

[10] Можно произвольное значение.

$$
\begin{cases}
\alpha_1 + \beta_2 = 5; \\
\alpha_1 + \beta_4 = 4; \\
\alpha_2 + \beta_2 = 1; \\
\alpha_3 + \beta_1 = 0; \\
\alpha_3 + \beta_3 = 2; \\
\alpha_3 + \beta_4 = 3.
\end{cases}
\tag{6}
$$

Вернемся к примеру. Введем потенциалы. По переменным, соответствующим ненулевым перевозкам плана $X^{(3)}$, составим систему линейных уравнений (6). Полагая $\alpha_1 = 0$, получаем: $\beta_2 = 5$, $\beta_4 = 4$, $\alpha_2 = -4$, $\alpha_3 = -1$, $\beta_1 = 1$, $\beta_3 = 3$.

Преобразуем матрицу стоимостей.

$$
\begin{array}{cc}
 & \begin{array}{cccc} -1 & -5 & -3 & -4 \end{array} \\
C = & \begin{pmatrix} 3 & 5 & 9 & 4 \\ 7 & 1 & 6 & 8 \\ 0 & 9 & 2 & 3 \end{pmatrix} \begin{array}{c} -0 \\ +4 \\ +1 \end{array}
\end{array}
$$

Получаем

$$
C' = \begin{pmatrix} 2 & 0 & 6 & 0 \\ 10 & 0 & 7 & 8 \\ 0 & 5 & 0 & 0 \end{pmatrix}.
$$

Матрица стоимостей C' получена из матрицы C преобразованиями, не изменяющими оптимальное решение. C' не содержит отрицательных элементов, стоимость плана $X^{(3)}$, вычисленная по этой матрице, равна нулю. Следовательно, он оптимален для матрицы C', а следовательно и для первоначальной матрицы C. Задача решена.

Замечание. Считать потенциалы можно прямо на таблице, не выписывая системы уравнений (6), см. чертеж.

	$\beta_1=1$	$\beta_2=5$	$\beta_3=3$	$\beta_4=4$	
$\alpha_1=0$	3	20 5	9	30 4	50
$\alpha_2=-4$	7	20 1	6	8	20
$\alpha_3=-1$	10 0	9	15 2	5 3	30
	10	40	15	35	

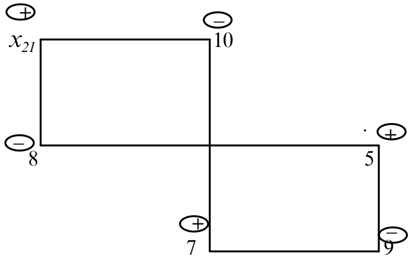
Если в новой матрице стоимостей есть хотя бы один отрицательный элемент, то условия леммы не выполнены, и мы не можем быть уверены в оптимальности нашего плана. Стоимость этого плана равна нулю, но может существовать план отрицательной стоимости (во многих случаях отрицательные элементы свидетельствуют о незамеченных неправильных прямоугольниках).

Выберем в транспортной таблице клетку, которой соответствует отрицательное значение в матрице стоимостей. Нам выгодно изменить план, увеличив перевозку в этой клетке. Изменив (увеличив или уменьшив) перевозку в какой-то клетке, мы должны компенсировать это изменение (уменьшив или увеличив соответственно какую-то перевозку в этой строке и какую-то перевозку в этом столбце). Клетки, в которых изменились перевозки, образуют цикл. Цикл — это замкнутая ломаная линия, состоящая из горизонтальных и вертикальных отрезков. Вершины ломаной находятся в тех клетках таблицы, в которых находятся ненулевые перевозки. Отрезки ломаной могут соединять только элементы, имеющие ненулевые перевозки и стоящие в одном столбце или в одной строке. Такой цикл обязательно существует, если план был невырожденный, то есть содержал $(n+m-1)$ ненулевую перевозку. Найдя такой цикл, помечаем его клетки знаками «+» и «-», начиная со знака «+» в начальной клетке с отрицательной стоимостью. Затем знаки должны чередоваться. Мы увеличим перевозку во всех клетках, помеченных знаком «+», и уменьшим перевозку во всех клетках, помеченных знаком «-». Изменение перевозки надо сделать на величину минимальной из перевозок, отмеченных знаком «-». Новый план будет иметь меньшую стоимость.

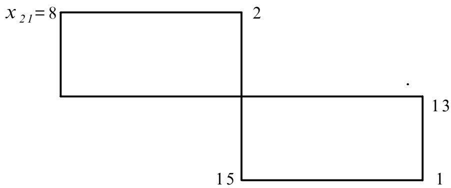
Рассмотрим *пример*. Пусть после преобразований получили следующую таблицу. $c_{21} < 0$. Попробуем увеличить перевозку x_{21}.

1	0 6	2	7	2	6
-2 x_{21}	5	0 10	0 4	6	14
0 8	3	4	2	0 5	13
3	0 8	0 7	4	0 9	24
8	14	17	4	14	

Начинаем построение цикла с вершины x_{21}. Найденный цикл имеет вид, как на рисунке.



Среди вершин, перевозки в которых должны быть уменьшены, ищем минимальную перевозку. Min (10, 9, 8) = 8. Изменяем план, увеличивая на 8 единиц перевозки в вершинах со знаком плюс и уменьшая на 8 единиц в вершинах со знаком минус. В изменяемой части план станет таким:



Новый план:

1	0 6	2	7	2	6
-2 8	5	0 2	0 4	6	14
0	3	4	2	0 13	13
3	0 8	0 15	4	0 1	24
8	14	17	4	14	

Является ли он оптимальным? Для того чтобы ответить на этот вопрос, необходимо вновь преобразовать матрицу стоимостей. При этом для дальнейших преобразований можно выбрать начальную матрицу, а можно и измененную.

Алгоритм решения транспортной задачи методом потенциалов

Шаг 1. Подобрать некоторый первоначальный план перевозок.

Шаг 2. Используя неправильные прямоугольники (если они есть), улучшить имеющийся план (метод локального поиска).

Шаг 3. С помощью потенциалов преобразовать матрицу стоимостей согласно последнему (самому лучшему из найденных) плану. В случае отсутствия отрицательных элементов в новой матрице стоимостей, найденный план оптимален. Если появился отрицательный элемент, то изменяем план так, чтобы на этом месте была ненулевая перевозка. Получаем новый план. Повторяем шаг 3.

Тонкости алгоритма

Всегда ли система уравнений для определения потенциалов имеет решение? Система содержит $(n + m)$ переменных. При решении системы (6) возможны три случая. Рассмотрим их более детально.

Случай 1. Положив $\alpha_1 = 0$, найдем частное решение системы, однозначно определяя все остальные переменные. В этом случае в системе $(n + m - 1)$ уравнение.

Случай 2. Положив $\alpha_1 = 0$, мы не смогли однозначно определить все потенциалы. Оставшиеся уравнения содержат только неизвестные величины. Тогда снова выберем некоторую переменную и зададим ей некоторое значение (обычно равное нулю). Продолжим вычисления. Этот шаг может быть придется повторить несколько раз. Такая ситуация возникает в том случае, если число уравнений меньше $(n + m - 1)$. План, в котором число положительных перевозок меньше $(n + m - 1)$, называют вырожденным.

Случай 3. Система может не иметь решений. В этом случае в матрице перевозок есть цикл (см. описание цикла), вершины которого соответствуют ненулевым перевозкам. Для проверки того, что система имеет решение, применим следующую процедуру. Назовем ее процедурой развала. Вычеркнем в матрице все строки и столбцы с одной перевозкой. Будем повторять это, пока возможно. Если мы вычеркнули все строки и все столбцы матрицы, то цикла нет, и система имеет решение.

На примере это выглядит следующим образом. Рассмотрим матрицу перевозок

$$\begin{pmatrix} 0 & 20 & 0 & 30 \\ 0 & 20 & 0 & 0 \\ 10 & 0 & 15 & 5 \end{pmatrix}.$$

Число ненулевых перевозок равно 6. Это равно числу строк плюс число столбцов минус 1. Вычеркнем строку 2, столбец 1, столбец 3. После вычеркивания появились новые строки и столбцы с одной ненулевой перевозкой. Сохраняя прежнюю нумерацию, вычеркнем столбец 2 и строку 3. В матрице остался один ненулевой элемент a_{14}, вычеркнем и его. Матрица развалилась. Система уравнений (6) разрешима.

Если ненулевых перевозок в плане больше, чем ($n+m-1$), то матрица наверняка не развалится, то есть мы не сможем удалить все строчки и столбцы. В оставшейся части матрицы обязательно существует цикл. Выделим его. Отметим вершины цикла знаками «+» и «-», чередуя их. Число тех и других в цикле всегда одинаково. Подсчитаем сумму стоимостей во всех вершинах, отмеченных одинаковыми знаками. Сравним их и выберем большую сумму. Уменьшим все перевозки в вершинах с тем знаком, которому соответствует большая сумма. Уменьшение произведем на минимальную перевозку среди этих вершин. В вершинах, отмеченных другим знаком, перевозки увеличим на ту же величину. Например, если суммарная стоимость больше у плюсов, то мы получим план меньшей стоимости, если уменьшим перевозки у всех плюсов и увеличим перевозки у всех минусов. Если суммы стоимостей одинаковы, то выберем сумму с произвольным знаком и проделаем такие же действия. Во всех случаях изменение перевозок делается на величину, минимальную из уменьшаемых перевозок. Эта процедура аналогична исправлению неправильных прямоугольников.

При появлении в матрице стоимостей отрицательных элементов, добавляем к клеткам с ненулевыми перевозками клетку с отрицательной стоимостью и ищем цикл. Для поиска цикла можно снова применить процедуру развала. Если граф развалится полностью, то цикла нет. В этом случае надо добавить эту клетку к системе уравнений для поиска потенциалов и заново решить систему. Если цикл найден, то перераспределим перевозки.

Венгерский алгоритм решения транспортной задачи

Выше нами был рассмотрен один из методов решения транспортной задачи — метод потенциалов. Применяя этот метод, мы сначала находим какой-либо план перевозок, а затем преобразуем матрицу стоимостей так, чтобы все перевозки осуществлялись по элементам нулевой стоимости. Если при этом все элементы новой матрицы неотрицательны, то найденный план оптимален. При решении задачи венгерским алгоритмом будем искать план, используя перевозки только нулевой стоимости. Если мы не можем назначить все перевозки по элементам нулевой стоимости, то будем производить преобразования матрицы стоимостей C: отнимать или прибавлять к строкам и столбцам соответствующие числа. При этом в преобразованной матрице не должно быть отрицательных элементов. Ранее было показано, что план, оптимальный для исходной матрицы стоимостей, оптимален и для преобразованной матрицы и наоборот. Если мы перевезли все, что требуется, то, согласно лемме об оптимальности (*в матрице без отрицательных стоимостей план нулевой стоимости заведомо оптимален*), задача решена.

Описание алгоритма будем сопровождать примером его реализации. Рассмотрим *пример 1.* Этот пример был решен нами ранее методом потенциалов. Пусть задана транспортная таблица.

3	5	9	4	50
7	1	6	8	20
0	9	2	3	30
10	40	15	35	

Все перевозки нужно осуществлять только по элементам нулевой стоимости. Однако в первой строке матрицы стоимостей нет ни одного нуля. Вычтем из всех элементов первой строки матрицы стоимостей ее минимальный элемент, то есть 3. Так как в первой строке матрицы надо распределить 50 единиц товара, а стоимость перевозки каждой единицы товара станет по новой матрице на 3 единицы дешевле, то стоимость любого плана по новой матрице дешевле на 150 (3×50) единиц, чем по старой.

Введем обозначения.

Пусть $L(X)$ — стоимость плана перевозок X ($X = \{x_{ij}\}$), подсчитанная по первоначальной матрице стоимостей C;

C_k — матрица стоимостей, полученная после k-го преобразования;

$L_k(X)$ — стоимость плана перевозок X, подсчитанная по матрице стоимостей C_k.

Тогда, при рассмотренном преобразовании матрицы, для любого X:

$$L_1(X) = L(X) - 150.$$

Алгоритм

Шаг 1. Преобразуем матрицу стоимостей C, отнимая от каждой строки минимальный элемент этой строки [11].

Шаг 2. Отнимем от каждого столбца, не имеющего нулевых элементов, его минимальный элемент [12].

После выполнения первого шага алгоритма матрица стоимостей примет вид:

$$\begin{array}{cccc|c} 3 & 5 & 9 & 4 & -3 \\ 7 & 1 & 6 & 8 & -1 \\ 0 & 9 & 2 & 3 & \\ \hline \end{array} \Longrightarrow \begin{array}{cccc|} 0 & 2 & 6 & 1 \\ 6 & 0 & 5 & 7 \\ 0 & 9 & 2 & 3 \\ \hline \end{array}$$

$$L_1(X) = L(X) - 3{\cdot}50 = L(X) - 150.$$

$$L_2(X) = L_1(X) - 1{\cdot}20 = L(X) - 170.$$

После второго шага алгоритма:

$$\begin{array}{cccc|} 0 & 2 & 6 & 1 \\ 6 & 0 & 5 & 7 \\ 0 & 9 & 2 & 3 \\ \hline & & (-2) & (-1) \end{array} \Longrightarrow \begin{array}{cccc|} 0 & 2 & 4 & 0 \\ 6 & 0 & 3 & 6 \\ 0 & 9 & 0 & 2 \\ \hline \end{array}$$

$$L_3(X) = L_2(X) - 2{\cdot}15 = L_2(X) - 30 = L(X) - 200.$$

$$L_4(X) = L_3(X) - 35 = L(X) - 235.$$

---

[11] После этого в каждой строке окажется хотя бы один 0.

[12] После этого в каждой строке и каждом столбце имеется хотя бы один 0. На всех этапах алгоритма в матрице стоимостей не должно быть отрицательных элементов.

0	2	4	0	50
6	0	3	6	20
0	9	0	2	30
10	40	15	35	

Попытаемся выяснить, можно ли по имеющимся нулям реализовать весь план перевозок. В полученной матрице в каждой строке и каждом столбце есть хотя бы один нуль. Но иногда строке или столбцу одного нуля недостаточно. Например, в третьей строке есть даже два нуля. Но нуль в первом столбце может обеспечить не более 10 единиц товара, а нуль в третьем столбце не более 15. Вместе они дадут не более 25, а строке надо 30. В данном случае осуществить все перевозки по имеющимся нулям невозможно. Надо сделать в третьей строке еще нули. Для этого отнимем от третьей строки ее минимальный ненулевой элемент, то есть 2. Чтобы в новой матрице не появилось отрицательных элементов, к столбцам, имеющим в этой строке нули, прибавим 2.

При всех преобразованиях матрицы стоимость любого плана должна уменьшаться. Вычитая 2 от 4-го столбца, мы уменьшим стоимость плана на 2×30 единиц. Прибавив 2 к 1-му и 2-му столбцам, мы увеличим стоимость на величину 2× (10+15). Поэтому

$$L_5(X) = L_4(X) - 10 = L(X) - 245.$$

Матрица стоимостей примет вид:

$$\begin{array}{cccc|l} 0 & 2 & 4 & 0 & \\ 6 & 0 & 3 & 6 & \\ 0 & 9 & 0 & 2 & (-2) \\ \hline (+2) & & (+2) & & \end{array} \Longrightarrow \begin{array}{cccc|} 2 & 2 & 6 & 0 \\ 8 & 0 & 5 & 6 \\ 0 & 7 & 0 & 0 \\ \hline \end{array}$$

Первой строке надо перевезти 50 единиц товара. Там есть только один нуль, который может обеспечить не более 35 единиц. Сделаем в первой строке еще нули. Отнимем 2 от первой строки и прибавим 2 к 4-му столбцу. Матрица стоимостей примет вид:

$$\begin{array}{cccc|l} 2 & 2 & 6 & 0 & (-2) \\ 8 & 0 & 5 & 6 & \\ 0 & 7 & 0 & 0 & \\ \hline & & & (+2) & \end{array} \Longrightarrow \begin{array}{cccc|} 0 & 0 & 4 & 0 \\ 8 & 0 & 5 & 8 \\ 0 & 7 & 0 & 2 \\ \hline \end{array}$$

$$L_6(X) = L_5(X) - 2{\cdot}50 + 2{\cdot}35 = L(X) - 275.$$

3-й строке опять не хватает перевозки. Надо сделать в третьей строке еще нули. Для этого можно отнять от третьей строки ее минимальный ненулевой элемент, то есть 2. Чтобы в новой матрице не появились отрицательные элементы, надо к тем столбцам, где в этой строке стояли нули, прибавить 2. Матрица стоимостей примет вид:

$$\begin{array}{cccc|l} 0 & 0 & 4 & 0 & \\ 8 & 0 & 5 & 8 & \\ 0 & 7 & 0 & 2 & (-2) \\ \hline (+2) & & (+2) & & \end{array} \Longrightarrow \begin{array}{cccc|} 2 & 0 & 6 & 0 \\ 10 & 0 & 7 & 8 \\ 0 & 5 & 0 & 0 \\ \hline \end{array}$$

$$L_7(X) = L_6(X) - 2 \cdot 30 + 2 \cdot (15{+}10) = L_6(X) - 10 = L(X) - 285.$$

Шаг 3. Проверить, хватает ли имеющихся нулей каждой строке и столбцу в отдельности. Для этого сравнить потребности строки (столбца) с суммарной возможностью всех нулей этой строки (столбца). Если не хватает, сделать новые нули в соответствующей строке (столбце).

В рассмотренном примере мы могли заметить, что во втором столбце был только один нуль, который мог дать 20 единиц товара, а столбцу надо 40. Чтобы разрешить этот конфликт, мы могли бы отнять от второго столбца минимальный элемент этого столбца и прибавить этот элемент ко второй строке. В данном случае этот конфликт разрешился сам при устранении других конфликтов.

Шаг 4. По преобразованной матрице стоимостей ищем план, назначая возможно большие перевозки там, где стоят нулевые стоимости. При этом рекомендуется в первую очередь выбирать такие строки (столбцы), в которых единственный нуль, не заполненный перевозкой. Если такой план найден, то задача решена, найденный план оптимален. Необходимо подсчитать стоимость плана перевозок по первоначальной матрице стоимостей. Если же нам не удалось построить такой план, то перейдем к шагу 5.

В рассматриваемом примере в первой строке 2 нуля. Пока ее пропустим. Во второй строке только один нуль. Назначим на него максимально возможную перевозку. Строке надо 20. Столбец может дать 40. Назначим $x_{22} = 20$. В третьей строке 3 нуля. Пока пропустим. Перейдем к столбцам. В первом столбце один нуль. Назначим $x_{31} = 10$. Во втором столбце два нуля, но один уже занят. На оставшийся нуль назначим $x_{12} = 20$. В третьем столбце один нуль. Назначим $x_{33} = 15$. Оставшиеся перевозки назначим в 4-й столбец ($x_{34} = 5$, $x_{14} = 30$). Мы смогли назначить все перевозки на элементы нулевой стоимости.

$$\begin{array}{cccc|c} 2 & 0^{20} & 6 & 0^{30} & 50 \\ 10 & 0^{20} & 7 & 8 & 20 \\ 0^{10} & 5 & 0^{15} & 0^{5} & 30 \\ \hline 10 & 40 & 15 & 35 & \end{array}$$

Все требования на перевозки выполнены, следовательно, это допустимый план. Его стоимость по последней матрице равна 0. Так как в ней нет отрицательных элементов, то план нулевой стоимости оптимален. А так как мы делали только эквивалентные преобразования, то этот план оптимален и для первоначальной матрицы. Мы нашли оптимальный план X^*. Его стоимость по последней матрице стоимостей равна нулю.

$$L_7(X^*) = 0; \; L(X^*) = 285.$$

Проверьте, что стоимость этого плана по исходной матрице равна 285.

Если план не найден, то не всегда понятно, почему это произошло — то ли из-за того, что плохо были назначены перевозки по имеющимся нулям, то ли имеющихся нулей недостаточно для перевозки всего товара. Выполнение шага 5 позволит ответить на этот вопрос.

Шаг 5. Алгоритм расстановки меток:

A) Выделим недовольную[13] строку и отметим ее.

B) Если отмечена строка, то отметим все те столбцы, у которых в этой строке нулевые стоимости.

C) Если отмечен столбец, то отмечаем все те строки, у которых в этом столбце ненулевые перевозки.

Пункты *B* и *C* повторяются. Если мы отметили недовольный столбец, то перейти к пункту *D1.* Если мы не можем сделать новых отметок, то перейти к пункту *D2.*

---

[13] Будем называть строку «недовольной», если предложение в этой строке реализовано не полностью. Аналогично будем называть столбец «недовольным», если спрос в этом столбце полностью не удовлетворен.

Алгоритм расстановки меток приводит к одной из двух ситуаций. Одну из них назовем ситуацией прорыва. Она возникает в том случае, если мы пометили «недовольный» столбец. Это означает, что мы плохо распределили перевозки по имеющимся нулям. План перевозок будет изменен и общее количество перевезенного товара будет увеличено. Другую ситуацию назовем «узкое место». В этом случае по имеющимся нулям невозможно реализовать все перевозки. В этом случае матрица стоимостей будет преобразована. В матрице появятся новые нули, а стоимость любого плана по новой матрице будет меньше, чем по старой. Рассмотрим каждую из этих ситуаций более подробно.

D_1) Ситуация «прорыва». Перераспределим перевозки. На пересечении «недовольного» столбца и строки, из-за которой он был отмечен, перевозку увеличим. Чтобы сохранить величину перевозки в этой строке, уменьшим перевозку в строке и столбце, из-за которого она была отмечена. Чтобы сохранить величину перевозки в этом столбце, увеличим перевозку в столбце и пометившей его строке. И т. д., пока не дойдем до «недовольной» строки. Мы перевезли большее количество товара. Если перевезли все, что надо, то получили оптимальный план. Если опять перевезли не все, то повторяем шаг 5.

D_2) Ситуация «узкое место». Находим минимальный элемент в отмеченных строках матрицы стоимостей, исключая элементы отмеченных столбцов. Отнимаем его от каждой отмеченной строки и прибавляем к каждому отмеченному столбцу. После этого выполняем шаг 3.

Рассмотрим *пример 2.* Данные задачи представлены в транспортной таблице матрицей стоимостей, указаны также спрос и предложение. В таблице четыре пункта предложения, и пять пунктов потребления. Решим задачу, применяя венгерский алгоритм.

11	12	11	12	10	20
9	10	7	8	9	15
14	12	13	12	14	17
9	10	8	10	11	18
12	15	14	20	9	

Проверим условие баланса. Суммарный спрос равен (12 + 15 + 14 + 20 + 9) = 70. Суммарное предложение равно (20 + 15 + 17 + 18) = 70. Задача сбалансирована.

Выполним шаг 1. Вычтем из каждой строки ее минимальный элемент.

$$\begin{array}{ccccc|l} 11 & 12 & 11 & 12 & 10 & (-10) \\ 9 & 10 & 7 & 8 & 9 & (-7) \\ 14 & 12 & 13 & 12 & 14 & (-12) \\ 9 & 10 & 8 & 10 & 11 & (-8) \\ \hline \end{array} \Longrightarrow \begin{array}{ccccc|} 1 & 2 & 1 & 2 & 0 \\ 2 & 3 & 0 & 1 & 2 \\ 2 & 0 & 1 & 0 & 2 \\ 1 & 2 & 0 & 2 & 3 \\ \hline \end{array}$$

В каждой строке преобразованной матрицы есть нули.

$L_1(X) = L(X)$ — $(10{\cdot}20 + 7{\cdot}15 + 12{\cdot}17 + 8{\cdot}18) = L(X)$ — 653.

Здесь и далее $L(X)$ — функция стоимости, подсчитанная по исходной матрице стоимостей, а $L_k(X)$ — функция стоимости после k-го изменения матрицы стоимостей.

Выполним шаг 2. Поскольку в полученной матрице в первом столбце нет нулевых элементов, отнимем от первого столбца его минимальный элемент 1.

$$\begin{array}{ccccc|} 1 & 2 & 1 & 2 & 0 \\ 2 & 3 & 0 & 1 & 2 \\ 2 & 0 & 1 & 0 & 2 \\ 1 & 2 & 0 & 2 & 3 \\ \hline (-1) & & & & \end{array} \Longrightarrow \begin{array}{ccccc|} 0 & 2 & 1 & 2 & 0 \\ 1 & 3 & 0 & 1 & 2 \\ 1 & 0 & 1 & 0 & 2 \\ 0 & 2 & 0 & 2 & 3 \\ \hline \end{array}$$

В каждой строке и столбце матрицы есть нули.

$$L_2(X) = L_1(X) - 12 = L(X) - 665.$$

Выполним шаг 3. Проверяем, хватает ли имеющихся нулей каждой строке и столбцу в отдельности.

Видим, что во второй строке нуль находится только в третьем столбце, которому требуется 14 единиц товара. Наибольшая перевозка здесь составит 14 единиц, в то время как 2-й строке надо 15 единиц. Это означает, что план нулевой стоимости по имеющимся нулям построить невозможно. Второй строке не хватает нулей. Определим минимальный ненулевой элемент этой строки. Он равен 1. Вычитаем единицу из второй строки и прибавляем ее к третьему столбцу.

$$L_3(X) = L_2(X) - 15 + 14 = L_2(X) - 1 = L(X) - 666.$$

0	2	1	2	0			0	2	2	2	0
1	3	0	1	2	(-1)	$\Longrightarrow$	0	2	0	0	1
1	0	1	0	2			1	0	2	0	2
0	2	0	2	3			0	2	1	2	3
		(+1)									

Повторим шаг 3. Проверим, хватает ли имеющихся нулей каждой строке и столбцу в отдельности.

0	2	2	2	0	20
0	2	0	0	1	15
1	0	2	0	2	17
0	2	1	2	3	18
12	15	14	20	9	

Видим, что в четвертой строке нуль находится только в первом столбце. Наибольшая перевозка здесь составит 12 единиц, в то время как 4-й строке надо 18 единиц. Вычтем из 4-й строки матрицы стоимостей 1, а к первому столбцу прибавим 1.

0	2	2	2	0			1	2	2	2	0
0	2	0	0	1			1	2	0	0	1
1	0	2	0	2		$\Longrightarrow$	2	0	2	0	2
0	2	1	2	3	(-1)		0	1	0	1	2
(+1)											

$$L_4(X) = L_3(X) - 18 + 12 = L_3(X) - 6 = L(X) - 672.$$

Вновь повторим шаг 3.

1	2	2	2	0	20
1	2	0	0	1	15
2	0	2	0	2	17
0	1	0	1	2	18
12	15	14	20	9	

Точно так же выделим первую строку, вычтем из нее 1 и прибавим 1 к пятому столбцу.

$$\begin{array}{ccccc|l} 1 & 2 & 2 & 2 & 0 & (-1) \\ 1 & 2 & 0 & 0 & 1 & \\ 2 & 0 & 2 & 0 & 2 & \\ 0 & 1 & 0 & 1 & 2 & \\ \hline & & & & (+1) & \end{array} \Longrightarrow \begin{array}{ccccc|} 0 & 1 & 1 & 1 & 0 \\ 1 & 2 & 0 & 0 & 2 \\ 2 & 0 & 2 & 0 & 3 \\ 0 & 1 & 0 & 1 & 3 \\ \hline \end{array}$$

$$L_5(X) = L_4(X) - 20 + 9 = L_4(X) - 11 = L(x) - 683.$$

Выполним шаг 4. Попробуем по имеющимся нулям построить план перевозок.

0^{8}	1	1	1	0^{9}	20
1	2	0	0^{15}	2	15
2	0^{15}	2	0^{2}	3	17
0^{4}	1	0^{14}	1	3	18
12	15	14	20	9	

Во втором столбце один нуль. Назначим максимально возможную перевозку в 15 единиц (x_{32} = 15). Тогда в третьей строке остается один нулевой элемент. Здесь назначим перевозку в две единицы, чтобы выполнить требования, налагаемые строкой (x_{34} = 2). Теперь в 4-м столбце остался один нуль. Чтобы выполнить требования, налагаемые строкой, необходимо назначить перевозку в 18 единиц, но вторая строка, в которой стоит нуль, позволяет поставить только 15 единиц. Назначим перевозку в 15 единиц (x_{24} = 15). Поскольку предложение, представляемое 2-й строкой удовлетворено, то на нулевой элемент, стоящий во 2-й строке и 3-м столбце, ничего не назначаем. Теперь в третьем столбце и четвертой строке остался один нуль, на который можно назначить перевозку. Здесь мы назначим 14 единиц (x_{43} = 14). Четвертой строке не хватает 4-х единиц, назначим их на оставшийся нуль в четвертой строке и первом столбце (x_{41} = 4). Продолжая, в первом столбце зададим значение x_{11} равное 8 единицам. И на оставшийся нуль в первой строке (пятый столбец) назначим максимально возможную перевозку в 9 единиц (x_{15} = 9). Уже при назначении перевозки x_{24}, стало ясно, что план нельзя построить. Мы не смогли перевезти все, что нужно. Поэтому переходим к **выполнению шага 5**.

Отметим «недовольную» строку. Это строка 1-я. Поставим около нее звездочку.

По этой строке отметим первый и пятый столбец, так как они имеют нули в первой строке.

В первом столбце две ненулевые перевозки, в строке 1 и строке 4. Первая строка уже отмечена, отметим четвертую строку.

По четвертой строке отметим третий столбец, так как на их пересечении стоит нуль. Наконец, по третьему столбцу можно отметить только строку 4, она уже отмечена.

0^{8}	1	1	1	0^{9}	20*
1	2	0	0^{15}	2	15
2	0^{15}	2	0^{2}	3	17
0^{4}	1	0^{14}	1	3	18*
12*	15	14*	20	9*	

Процесс расстановки меток закончился. При этом не выделился «недовольный» столбец. Значит, у нас обозначилась ситуация «узкое место». В самом деле: отмеченные строки (первая и четвертая) вместе требуют 20+18=38 единиц товара. Перевозки в этих строках возможны только в отмеченных столбцах (первый, третий и пятый). Но вместе эти столбцы могут дать только 12+14+9=35 единиц товара. Так как 38>35, то реализовать все требования на перевозки невозможно. Выполняем пункт D2. Изменяем матрицу стоимостей. В строках 1 и 4 ищем минимальный элемент, исключая элементы 1, 3 и 5-го столбцов. Это 1. Вычитаем единицу из первой и четвертой строки и прибавляем единицу к 1, 3 и 5-му столбцам.

$$L_6(X) = L_5(X) - 20 - 18 + 12 + 14 + 9 = L_5(X) - 3 = L(X) - 686.$$

Перейдем к выполнению шага 4. Будем строить план нулевой стоимости по последней матрице стоимостей.

0^{11}	0	2	0	0^{9}	20
2	2	1	0^{15}	3	15
3	0^{15}	3	0^{2}	4	17
0^{1}	0	0^{14}	0^{3}	2	18
12	15	14	20	9	

В третьем столбце единственный нуль. Назначаем перевозку в 14 единиц ($x_{33} = 14$). В пятом столбце единственный нуль. Назначаем перевозку в 9 единиц ($x_{15} = 9$). Во второй строке один нуль. Назначим на него перевозку в 15 единиц ($x_{24} = 15$). В других строках и столбцах нет одновариантности. Продолжим построение плана, ориентируясь на предыдущий. Величину x_{11} положим равной 11 единицам, удовлетворив при этом потребности 1-й строки, а x_{41} — одной единице. В третьей строке и втором столбце оставим перевозку в 15 единиц ($x_{32} = 15$), а в четвертом — 2 единицы ($x_{34} = 2$). Завершим построение плана, задав значение x_{44} равное 3, удовлетворив тем самым ограничения четвертого столбца и четвертой строки. Получили план нулевой стоимости.

$$X = \begin{pmatrix} 11 & 0 & 0 & 0 & 9 \\ 0 & 0 & 0 & 15 & 0 \\ 0 & 15 & 0 & 2 & 0 \\ 1 & 0 & 14 & 3 & 0 \end{pmatrix}.$$

План оптимален по последней матрице стоимостей. Так как мы делали только разрешенные преобразования, то он оптимален и для перевоначальной матрицы стоимостей. Подсчитаем его стоимость по первоначальной матрице. $L = 11{\cdot}11 + 10{\cdot}9 + 8{\cdot}15 + 12{\cdot}15 + 12{\cdot}2 + 9{\cdot}1 + 8{\cdot}14 + 10{\cdot}3 = 686$.

Получили, что $L_6(X) = 0 = L(X) - 686$. Совпадение результатов увеличивает нашу уверенность в правильности проделанных вычислений и в ответе.

Замечание: Мы рассмотрели 2 точных алгоритма решения транспортной задачи. Отметим, что для каждого из них выполняется

Теорема о целочисленности: *Если все запасы товаров и все потребности в них целые числа, то алгоритм найдет решение, где все перевозки являются целыми числами.*

Задачи для самостоятельного решения

1.
$$\begin{array}{ccccc|c} 4 & 11 & 4 & 4 & 1 & 30 \\ 12 & 1 & 3 & 15 & 6 & 40 \\ 5 & 12 & 10 & 7 & 8 & 20 \\ 9 & 5 & 15 & 8 & 10 & 510 \\ \hline 120 & 130 & 160 & 170 & 20 & \end{array}$$

2.
$$\begin{array}{ccccc|c} 8 & 7 & 6 & 8 & 11 & 190 \\ 1 & 5 & 15 & 8 & 9 & 60 \\ 9 & 10 & 3 & 7 & 6 & 70 \\ 7 & 7 & 7 & 14 & 3 & 380 \\ \hline 150 & 60 & 90 & 200 & 200 & \end{array}$$

3.
$$\begin{array}{ccccc|c} 6 & 4 & 7 & 12 & 14 & 130 \\ 7 & 11 & 11 & 2 & 15 & 50 \\ 1 & 3 & 10 & 3 & 7 & 110 \\ 7 & 15 & 10 & 4 & 11 & 200 \\ \hline 140 & 200 & 10 & 10 & 130 & \end{array}$$

4.
$$\begin{array}{ccccc|c} 4 & 1 & 6 & 13 & 1 & 150 \\ 15 & 12 & 12 & 8 & 13 & 100 \\ 11 & 4 & 12 & 3 & 8 & 100 \\ 14 & 4 & 5 & 13 & 7 & 160 \\ \hline 90 & 80 & 110 & 40 & 190 & \end{array}$$

5.
$$\begin{array}{ccccc|c} 6 & 11 & 15 & 15 & 10 & 160 \\ 7 & 4 & 1 & 8 & 1 & 10 \\ 10 & 14 & 6 & 12 & 10 & 180 \\ 7 & 2 & 1 & 4 & 6 & 230 \\ \hline 110 & 70 & 180 & 50 & 170 & \end{array}$$

6.
$$\begin{array}{ccccc|c} 5 & 1 & 1 & 8 & 12 & 70 \\ 6 & 8 & 5 & 3 & 13 & 140 \\ 12 & 15 & 8 & 15 & 14 & 190 \\ 3 & 3 & 13 & 2 & 2 & 150 \\ \hline 40 & 150 & 110 & 80 & 170 & \end{array}$$

Ответы

1. L = 4780. x_{13}=30; x_{23}=40; x_{33}=20; x_{41}=120; x_{42}=130; x_{43}=70; x_{44}=170; x_{45}=20.

2. L = 3690. x_{14}=190; x_{21}=60; x_{33}=60; x_{34}=10; x_{41}=90; x_{42}=60; x_{43}=30; x_{45}=200.

3. L = 3020. x_{12}=130; x_{21}=40; x_{24}=10; x_{31}=40; x_{32}=70; x_{41}=60; x_{43}=10; x_{45}=130.

4. L = 2820. x_{11}=10; x_{15}=140; x_{21}=80; x_{24}=20; x_{32}=80; x_{34}=20; x_{43}=110; x_{45}=50.

5. L = 3140. x_{11}=110; x_{15}=50; x_{25}=10; x_{33}=70; x_{35}=110; x_{42}=70; x_{43}=110; x_{44}=50.

6. L = 3030. x_{12}=70; x_{22}=60; x_{24}=80; x_{31}=40; x_{32}=20; x_{33}=110; x_{35}=20; x_{45}=150.

§6. ЗАДАЧА КОММИВОЯЖЕРА

Коммивояжер (бродячий торговец) должен объехать n городов, побывав в каждом по одному разу, и вернуться обратно. Мы хотим из множества возможных способов объезда городов выбрать наилучший.

Что может являться критерием оптимальности?

В качестве критерия оптимальности могут быть выбраны разные величины. Например, время объезда городов, суммарная стоимость или длина пройденного пути. Мы в качестве критерия оптимальности выберем суммарную длину пройденного пути.

Рассмотрим *пример*. Предположим, необходимо объехать 5 городов. Нужно определить самый короткий способ объезда этих городов.

Будем считать, что возможен переезд из любого города в любой другой и известна длина каждого переезда. Эта информация задается в виде матрицы.

$$\begin{pmatrix} \times & 4 & 2 & 8 & 3 \\ 6 & \times & 9 & 7 & 1 \\ 5 & 0 & \times & 4 & 1 \\ 6 & 7 & 9 & \times & 3 \\ 5 & 8 & 3 & 7 & \times \end{pmatrix}$$

Назовем эту матрицу матрицей стоимостей и будем обозначать буквой C. Обычно считают, что номер строки матрицы указывает, откуда мы едем, а номер столбца — куда. Так, стоимость переезда из города 1 в город 2 определяется элементом c_{12}, он расположен в 1-й строке и 2-м столбце. Эта стоимость равна 4 ($c_{12} = 4$). На диагонали чисел нет, так как диагональный элемент соответствует переезду из города в тот же город. Иногда крестики можно считать нулями, иногда бесконечностью. Матрица может не быть симметричной, так как стоимость проезда из A в B не всегда равна стоимости проезда из B в A. Не ограничивая общности, можно считать, что объезд начинается из 1-го города.

Дадим формальную постановку задачи.

Задано число n и матрица $C(n \times n)$. Назовем перестановкой чисел $\{1, 2, \ldots, n\}$ n-мерный вектор $\boldsymbol{x} = (x_1, x_2, x_3, \ldots, x_n)$ такой, что каждое x_i — это целое число от 1 до n и при $i \neq j$ $x_i \neq x_j$. На множестве всех перестановок рассмотрим функцию

$$f(x) = \sum_{i=1}^{n-1} c_{x_i x_{i+1}} + c_{x_n} c_{x_1} .$$

Ставится задача: минимизировать функцию *f(x)* на множестве перестановок.

Иногда задачи с различными словесными формулировками приводят к одной и той же математической модели. Рассмотрим задачу о переналадке станка. Пусть на одном станке надо обработать n деталей. Нас интересует время выполнения задания, которое складывается из суммарного времени, затраченного на обработку всех деталей и времени, затраченного на переналадки станка. Если сразу после i-й детали обрабатывается j-я деталь, то необходимо затратить время t_{ij} на переналадку станка.

Надо выбрать такой порядок обработки деталей, чтобы суммарное время переналадок было минимальным.

С математической точки зрения задача о переналадке эквивалентна задаче коммивояжера. Достаточно детали назвать городами, а время переналадки временем или стоимостью переезда. В дальнейшем будем говорить о задаче коммивояжера.

Вернемся к нашему примеру. Выберем некоторый способ объезда городов. Например, можно проехать по городам в порядке возрастания их номеров. Такому объезду соответствует перестановка $\boldsymbol{x} = (1,\ 2,\ 3,\ 4,\ 5)$.

$$1 \xrightarrow{4} 2 \xrightarrow{9} 3 \xrightarrow{4} 4 \xrightarrow{3} 5 \xrightarrow{5} 1$$

Здесь стрелки означают путь из города в город. Над стрелкой написана стоимость переезда. Суммарная стоимость этого варианта объезда

$$f(x) = c_{12} + c_{23} + c_{34} + c_{45} + c_{51} = 4 + 9 + 4 + 3 + 5 = 25.$$

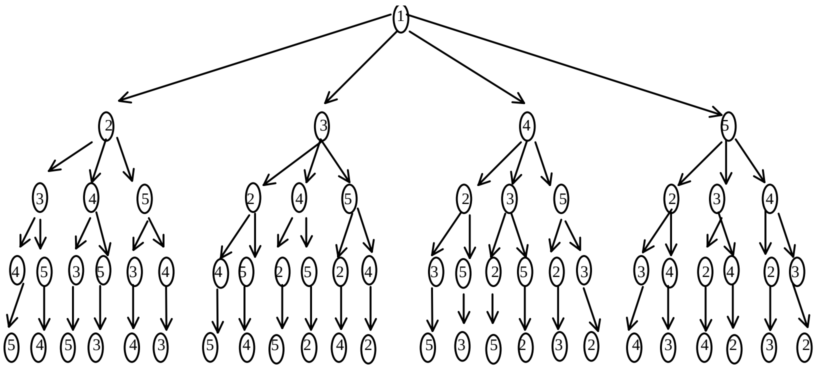
Мы рассмотрели произвольный объезд. Маловероятно, что он будет наилучшим. Попробуем действовать иначе, рассчитывая на более удачное решение. Стоимости переезда из 1-го города указаны в первой строке матрицы стоимостей. В 1-й строке минимальный элемент равен 2, это элемент c_{13}. Тогда из города 1 поедем в город 3. В 3-й строке минимальный элемент равен 0, это c_{32}. Тогда из города 3 поедем в город 2. Во 2-й строке минимальный элемент равен 1, это c_{25}. Тогда из города 2 поедем в город 5. В строке 5 минимальный элемент 3, это c_{53}, но в 3-м городе мы были. Мы были везде кроме города 4. Тогда из города 5 поедем в 4-й, а оттуда вернемся в город 1. Получили $\boldsymbol{x} = (1, 3, 2, 5, 4)$.

$$\mathbf{1} \xrightarrow{\mathbf{2}} \mathbf{3} \xrightarrow{\mathbf{0}} \mathbf{2} \xrightarrow{\mathbf{1}} \mathbf{5} \xrightarrow{\mathbf{7}} \mathbf{4} \xrightarrow{\mathbf{6}} \mathbf{1}$$

Здесь $f(x) = c_{13} + c_{32} + c_{25} + c_{54} + c_{41} = 16$. Этот результат существенно лучше предыдущего. Мы применили алгоритм «иди в ближайший», который достаточно разумен, но не гарантирует нахождение оптимального варианта. Такие алгоритмы называют *эвристическими.* В данном случае алгоритм смотрит на один шаг вперед и выбирает решение, оптимальное на этом шагу. Такие эвристические алгоритмы называют *жадными.* Для некоторых задач жадные алгоритмы дают оптимальное решение. Но можно придумать пример, когда алгоритм «иди в ближайший» дает наихудшее из всех возможных решений. Однако в большинстве случаев он дает достаточно хорошее решение.

Если множество вариантов в какой-то задаче является конечным, то теоретически можно найти наилучший вариант перебором всех возможных вариантов. Иногда такой перебор удобно делать с помощью построения дерева решений. Вершины его соответствуют моментам принятия решений, а ребра — возможным решениям.

Построим дерево вариантов для рассматриваемой задачи. В этой задаче мы ищем цикл, проходящий через все города. Варианты «$1 \to 2 \to 3 \to 4 \to 5 \to 1$» и «$3 \to 4 \to 5 \to 1 \to 2 \to 3$» определяют один и тот же цикл. Поэтому, без ограничения общности, можно считать, что все варианты начинаются с 1. Предположим, мы выехали из города 1. Далее можно поехать в город 2, 3, 4 или 5. Пусть мы выбрали город 4. Так как мы уже были в городе 1, то из города 4 можно поехать в город 2, 3 или 5. Выбираем, например, 2. Куда мы можем поехать из него? Так как мы были уже в городах 1 и 4, то из 2 можно поехать либо в 3, либо в 5. Выбираем город 3. Последний город, в котором мы не были — это 5, поэтому из 3 мы вынуждены поехать в 5 и затем возвращаемся в 1. Легко проследить такой путь следования на нашем дереве вариантов.



Общее число вариантов равно *(n-1)!*

Такие деревья можно использовать при решении сложных задач даже в том случае, когда мы не можем формализовать процесс решения. Например, мы решаем, как встретить Новый год. При этом нужно выбрать, с кем его встретить, где, какие подарки дарить и т. д. Ответить на эти вопросы мы должны в каком-то порядке, и возможные ответы на очередной вопрос могут зависеть от того, как мы ответили на предыдущий вопрос.

Построение дерева вариантов имеет свои недостатки и преимущества. «Прогулявшись» по всем ветвям нашего дерева, мы обязательно найдем лучший вариант, однако дерево может оказаться слишком большим. Для уменьшения перебора применяется метод ветвей и границ. В нем мы также будем строить ветви дерева вариантов. Поэтому метод называется методом ветвей. Мы не будем рисовать все ветви. Заведомо плохие ветви будем отбрасывать. Поэтому метод называется методом границ. Это метод разумного перебора. С одной стороны, это перебор, и мы не потеряем оптимальный вариант. С другой стороны, это разумный перебор. Мы не рассматриваем заведомо плохие варианты.

Мы будем рассматривать матрицы стоимостей без отрицательных элементов. Пусть лучший из найденных способов объехать все города имеет стоимость *a*. Если в дальнейшем, при рассмотрении дерева вариантов, на уже сделанных шагах мы затратили не меньше *a*, то такой вариант можно отбросить. Так как в матрице нет отрицательных чисел, то стоимость варианта не может уменьшиться, а может только увеличиться.

Для уменьшения перебора рекомендуется преобразовать матрицу стоимостей. Это можно сделать, применяя те же процедуры, которые использовались при решении задачи о назначениях венгерским алгоритмом.

Задача коммивояжера тесно связана с задачей о назначениях. Данные определяются матрицей, каждый элемент которой задает стоимость соответствующего переезда из одного пункта в другой. Коммивояжер должен приехать в каждый город ровно один раз и ровно один раз должен выехать из каждого города. Если мы отметим в матрице сделанные поездки, то в каждой строке и каждом столбце будет выбран ровно один элемент. Поэтому *любое допустимое решение задачи о коммивояжере дает нам допустимое решение задачи о назначениях.* Обратное неверно, так как если первого человека назначить на 2 должность, а второго на первую, то коммивояжер должен был бы из города 1 ехать в 2, а из 2 в 1, что невозможно.

Как и при решении задачи о назначениях, мы имеем право прибавлять к строкам и столбцам матрицы стоимостей произвольные числа. При таких преобразованиях множество возможных вариантов не меняется, а целевая функция меняется на константу. План, оптимальный для новой матрицы, оптимален для старой и наоборот.

От каждой строки вычитаем ее минимальный элемент, считая, что на диагонали стоят бесконечности. Затем вычтем минимальный элемент из тех столбцов, где нет 0.

$$\begin{pmatrix}\times&4&2&8&3\\6&\times&9&7&1\\5&0&\times&4&1\\6&7&9&\times&3\\5&8&3&7&\times\end{pmatrix}\begin{matrix}-2\\-1\\-0\\-3\\-3\end{matrix}\longrightarrow\begin{array}{c}\begin{pmatrix}\times&2&0&6&1\\5&\times&8&6&0\\5&0&\times&4&1\\3&4&6&\times&0\\2&5&3&4&\times\end{pmatrix}\\\begin{matrix}-2&\quad&\quad&-4&\quad\end{matrix}\end{array}\longrightarrow\begin{array}{c}\begin{pmatrix}\times&2&0&2&1\\3&\times&8&2&0\\3&0&\times&0&1\\1&4&6&\times&0\\0&5&3&0&\times\end{pmatrix}\\\begin{matrix}\quad&\quad&\quad&\quad&+1\end{matrix}\end{array}\begin{matrix}\\-1\\\\-1\\\\\end{matrix}\longrightarrow\begin{pmatrix}\times&2&0^*&2&2\\2&\times&7&1&0^*\\3&0^*&\times&0&2\\0^*&3&5&\times&0\\0&5&0&0^*&\times\end{pmatrix}$$

После первого преобразования, вторая и четвертая строка претендовали на 5-й столбец. Чтобы в этих строках появились еще нули, мы отняли 1 от 2-й и 4-й строки и прибавили 1 к 5-му столбцу. Выбирая единственные нули в строках и столбцах, удалось пометить 5 нулей, стоящих в разных строках и столбцах. Мы решили задачу о назначениях, но соответствует ли это решение задачи о назначениях решению задачи коммивояжера?

В 1-й строке мы отметили 3-й столбец. Значит, коммивояжер из города 1 поедет в город 3. В 3-й строке мы отметили 2-й столбец. Значит, коммивояжер из города 3 поедет в город 2. Во 2-й строке мы отметили 5-й столбец. Значит, коммивояжер из города 2 поедет в город 5. В 5-й строке мы отметили 4-й столбец. Значит, коммивояжер из города 5 поедет в город 4. В строке 4 мы отметили 1-й столбец. Значит, коммивояжер из города 4 поедет в город 1. Данные переезды образуют цикл. Его длина по последней матрице равна 0. Все элементы матрицы неотрицательны. План заведомо оптимален для преобразованной матрицы, значит он оптимален и для первоначальной матрицы.

Рассмотрим другой *пример*. Пусть после решения задачи о назначениях мы получили следующую матрицу

$$\begin{pmatrix} \times & 2 & 2 & 0^* & 2 \\ 2 & \times & 7 & 1 & 0^* \\ 3 & 0^* & \times & 0 & 2 \\ 0^* & 3 & 5 & \times & 0 \\ 0 & 5 & 0^* & 0 & \times \end{pmatrix}.$$

Решение задачи о назначениях не может рассматриваться как решение задачи коммивояжера. В самом деле, в строке 1 мы отметили столбец 4. Значит, коммивояжер из города 1 поедет в город 4. В 4-й строке мы отметили столбец 1. Значит, коммивояжер из города 4 поедет в город 1. Мы получили цикл, не объехав все города. Продолжим путешествие из какого-нибудь другого города. В строке 2 мы отметили столбец 5. Значит, коммивояжер из города 2 поедет в город 5. В строке 5 мы отметили 3-й столбец. Значит, коммивояжер из города 5 поедет в город 3. В строке 3 мы отметили столбец 2. Значит, коммивояжер из города 3 поедет в город 2. Это решение задачи о назначениях соответствует не одному, а двум циклам. «$1 \to 4 \to 1$» и «$2 \to 5 \to 3 \to 2$».

Попробуем «склеить» эти циклы. Например, это можно сделать с помощью эвристического алгоритма «иди в ближайший» . Получим цикл «$1 \to 4 \to 5 \to 3 \to 2 \to 1$». Его длина по последней матрице равна 2.

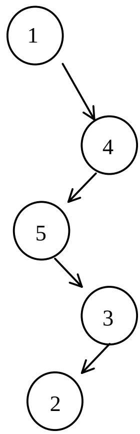
Для поиска оптимального решения применим метод ветвей и границ. В нем мы будем рисовать дерево вариантов. Но мы не будем рисовать все ветви. Заведомо плохие ветви мы будем отбрасывать. Это метод разумного перебора.

Это перебор и мы обязательно найдем оптимальный вариант. Но это разумный и быстрый перебор, так как заведомо плохие варианты отбрасываются.

В данном случае мы нашли вариант стоимости 2. Если его и можно улучшить, то не более чем на 2. Существует ли вариант лучше? В такой вариант не могут войти элементы матрицы ≥ 2. Все такие элементы можно удалить. Получим матрицу

$$\begin{pmatrix} \times & \times & \times & 0 & \times \\ \times & \times & \times & 1 & 0 \\ \times & 0 & \times & 0 & \times \\ 0 & \times & \times & \times & 0 \\ 0 & \times & 0 & 0 & \times \end{pmatrix}.$$

В первой строке остался только один элемент. Значит, можно рассматривать только варианты, начинающиеся $1 \to 4$, а ветки $1 \to 2$, $1 \to 3$, $1 \to 5$ отбросить. В строке 4 осталось 2 элемента, $4 \to 1$ и $4 \to 5$. Но в 1-й мы можем вернуться, только побывав во всех городах. Поэтому можно рассматривать только варианты, начинающиеся $1 \to 4 \to 5$. В строке 5 осталось 3 элемента, $5 \to 1$, $5 \to 3$, $5 \to 4$. Можно выбрать только $5 \to 3$. Далее только из $3 \to 2$. Из 2 можно только вернуться в 1, но такой шаг запрещен. Поэтому вариантов стоимости меньше 2 не существует. Полученный цикл $1 \to 4 \to 5 \to 3 \to 2 \to 1$ оптимален.



Дерево вариантов

Алгоритм решения задачи коммивояжера

Шаг 1. Решаем задачу о назначении венгерским алгоритмом. Если при решении задачи о назначении получился один цикл, то он будет решением задачи коммивояжера. Переходим к шагу 4. В противном случае переходим к шагу 2.

Шаг 2. Склеиваем полученные циклы в один большой.

Шаг 3. Применяем метод ветвей и границ. Строим дерево решений для новой матрицы, продолжая только те ветви, которые могут давать меньшее значение $L_{нов}$, по сравнению с предыдущими ветвями. Начинаем с любой вершины. Около ребер указываем затраты на путь. Выбираем наиболее перспективную ветвь (с наименьшей стоимостью) и продолжаем строить дерево до следующего уровня. Затем повторяем наши действия: выбираем наиболее перспективную ветвь. Поступаем таким образом, пока она не будет достроена полностью. Подсчитываем стоимость выбранного пути и далее используем это число за основу при анализе оставшихся ветвей. Если есть ветви, которые могут дать конечную стоимость меньше основной, то продолжаем эти ветви указанным выше способом. Остальные ветви ограничиваем. Полученный путь будет являться оптимальным, а стоимость — минимальной.

Шаг 4. Подсчитываем стоимость по первоначальной матрице.

Трудоемкость

Эта задача относится к числу труднорешаемых. При большом числе городов возникает слишком много вариантов (порядка $n!$). Пересмотреть все возможные варианты при больших n невозможно даже на суперЭВМ. Тогда рекомендуется применение эвристического алгоритма. В этом случае мы не можем гарантировать, что предложенный нами способ объезда городов оптимален.

Задачи для самостоятельного решения

Задана матрица расстояний между 6 городами. Найти кратчайший способ циклического объезда всех городов и длину кратчайшего пути.

1.

	5	13	9	6	8
2		2	7	20	8
4	13		3	1	8
2	13	9		19	1
10	4	17	16		17
3	11	13	8	11	

2.

	4	5	18	13	5
6		2	15	9	6
2	3		17	16	12
6	4	6		11	19
1	12	17	6		6
4	19	15	13	9	

3.

	9	15	5	4	11
16		19	13	1	7
1	19		10	18	14
19	11	3		19	2
14	13	2	17		12
17	4	16	8	12	

4.

	8	2	7	7	1
15		4	5	9	3
2	14		13	4	2
6	4	13		6	8
2	1	4	5		15
18	13	7	10	3	

5.

	16	2	14	9	10
20		7	11	3	3
1	20		15	17	14
18	6	19		17	7
5	20	15	6		1
16	16	8	16	4	

6.

	14	7	14	5	9
3		10	3	11	2
2	1		18	16	7
2	3	3		13	16
6	12	19	5		6
18	2	15	19	2	

Ответы

1. Кратчайший путь: 1→5→2→3→4→6→1; длина пути = 19.

2. Кратчайший путь: 1→6→5→4→2→3→1; длина пути = 28.

3. Кратчайший путь: 1→4→6→2→5→3→1; длина пути = 15.

4. Кратчайший путь: 1→3→6→5→2→4→1; длина пути = 19.

5. Кратчайший путь: 1→4→2→5→6→3→1; длина пути = 33.

6. Кратчайший путь: 1→3→2→6→5→4→1; длина пути = 19.

§7. ТИПОВЫЕ РАСЧЕТЫ

Задача о двух городах

Заданы координаты городов *A* и *B*. Используя модели 1, 2 и 3, найти наиболее выгодное расположение станции. Считать, что линия железной дороги лежит на оси *OX*.

1.1. A(3, 6), B(7, 5). **1.2**. A(2, 7), B(5, 7). **1.3**. A(4, 7), B(5, 5).
1.4. A(4, 7), B(6, 7). **1.5**. A(4, 3), B(8, 5). **1.6**. A(3, 7), B(5, 6).
1.7. A(2, 4), B(7, 6). **1.8**. A(2, 4), B(8, 7). **1.9**. A(3, 8), B(8, 4).
1.10. A(3, 8), B(5, 6). **1.11**. A(1, 8), B(7, 4). **1.12**. A(3, 3), B(5, 7).
1.13. A(2, 3), B(6, 5). **1.14**. A(2, 8), B(8, 5). **1.15**. A(2, 3), B(5, 6).
1.16. A(2, 5), B(7, 5). **1.17**. A(3, 4), B(5, 6). **1.18**. A(1, 5), B(8, 5).
1.19. A(4, 5), B(6, 7). **1.20**. A(3, 6), B(6, 4). **1.21**. A(3, 7), B(8, 7).
1.22. A(1, 6), B(8, 5). **1.23**. A(3, 6), B(7, 5). **1.24**. A(2, 5), B(6, 4).
1.25. A(1, 8), B(5, 5). **1.26**. A(2, 5), B(5, 5). **1.27**. A(2, 6), B(7, 4).
1.28. A(4, 6), B(7, 5). **1.29**. A(1, 7), B(6, 7). **1.30**. A (3, 7), B(5, 4)
1.31. A(1, 3), (6, 4). **1.32**. A(1, 6), (7, 6). **1.33**. A (4, 3), B(5, 6).
1.34. A(2, 5), (5, 6). **1.35**. A(4, 6), (5, 7). **1.36**. A (2, 5), B(6, 4).
1.37. A(2, 3), (7, 4). **1.38**. A(4, 3), (6, 5). **1.39**. A (4, 7), B(6, 6).
1.40. A(2, 3), (5, 5). **1.41**. A(4, 4), (5, 5). **1.42**. A(2, 5), B(5, 5).
1.43. A(1, 8), (7, 7). **1.44**. A(4, 4), (7, 4). **1.45**. A(3, 5), B(8, 4).

Задача о сетевом графике

Необходимо выполнить 20 работ. Данные о затратах времени, необходимых для выполнения каждой работы, заданы в виде матрицы. Каждая работа может начинаться только после окончания работ, расположенных выше и левее данной. Надо найти минимальное время выполнения всего комплекса, свободный и полный резерв каждой работы, критический путь и все критические работы.

1. 6 5 3 2 6
2 7 4 7 7
5 2 4 5 3
6 5 7 6 3

2. 2 6 5 2 3
3 3 7 3 2
4 6 5 3 2
7 6 3 5 4

3. 5 7 2 7 6
5 4 3 3 2
4 2 3 5 7
5 2 4 5 2

4. 2 3 2 5 6
3 4 2 7 2
4 4 3 5 7
4 7 3 3 2

5. 6 3 3 2 3
5 7 5 5 7
7 4 4 7 2
5 5 7 6 7

6. 4 6 4 6 4
3 4 7 3 2
3 4 4 6 5
2 3 2 6 4

7. 2 2 5 3 6
4 6 5 2 4
7 6 6 4 7
5 2 2 7 4

8. 3 4 5 5 5
5 6 6 4 3
6 5 6 7 6
7 2 6 4 3

9. 5 2 5 4 7
3 3 3 5 4
5 5 7 3 4
6 3 4 6 2

10. 4 3 2 7 7
7 5 5 7 4
3 4 3 3 7
5 3 7 4 7

11. 3 4 5 3 5
7 2 3 7 3
2 4 7 5 2
2 6 3 5 5

12. 7 7 4 7 3
3 5 5 5 5
2 6 2 4 3
6 4 7 6 4

13. 5 3 6 7 6
3 2 7 5 3
6 2 4 4 4
7 5 5 3 2

14. 4 5 2 2 5
5 7 5 6 2
6 4 6 6 3
7 5 5 2 2

15. 5 2 5 4 6
3 4 6 5 4
6 6 6 6 3
3 7 3 5 6

16. 6 7 4 4 5
2 5 5 3 7
4 5 2 7 2
6 4 4 5 3

17. 7 5 7 6 3
4 5 3 3 5
6 5 7 5 6
6 3 5 3 4

18. 5 4 4 5 3
2 6 4 6 2
2 6 6 4 6
4 3 5 5 6

19. 2 7 5 4 3
4 3 5 2 2
4 3 7 4 3
5 3 7 5 5

20. 7 4 3 4 4
3 7 7 7 5
2 5 4 5 4
3 2 5 2 3

21. 3 3 3 6 5
7 3 6 5 4
7 5 2 6 4
4 2 2 3 5

22. 4 2 4 3 5
5 3 5 5 4
4 2 2 3 2
7 6 5 5 3

23. 4 3 2 5 6
3 6 3 7 4
3 5 2 2 6
7 6 3 5 5

24. 2 2 5 3 7
3 2 7 6 3
5 4 7 3 4
7 7 4 5 2

25.

4	4	3	2	6
6	3	7	3	7
3	5	5	6	7
7	6	4	5	3

26.

7	4	5	4	3
7	7	6	2	3
2	7	5	3	2
5	2	5	3	3

27.

4	6	4	5	2
2	3	4	2	3
6	2	3	3	5
5	7	4	7	4

28.

3	2	4	4	6
3	4	2	6	6
2	5	4	2	4
2	2	7	4	2

29.

7	6	7	6	3
7	7	2	2	7
6	2	6	2	2
4	3	5	3	2

30.

6	4	7	6	4
3	6	3	7	4
4	7	6	4	4
3	7	7	7	3

31.

6	6	5	2	7
5	2	4	3	5
2	4	2	7	7
3	6	3	7	4

32.

3	3	3	4	6
3	3	7	5	3
6	4	5	3	6
5	2	7	7	6

33.

7	5	7	6	2
5	4	2	6	4
7	5	7	4	5
6	2	5	7	3

34.

6	3	6	5	5
5	3	7	2	3
5	4	6	6	2
3	7	4	6	7

35.

4	3	7	3	3
2	4	2	2	6
4	3	3	3	2
2	7	7	6	3

36.

5	6	4	6	6
3	6	2	2	6
2	5	3	5	5
5	6	6	2	6

37.

7	7	3	2	6
4	3	4	2	6
3	7	6	6	6
2	3	3	4	4

38.

5	7	5	5	4
7	6	7	3	5
7	7	7	3	4
3	3	2	2	4

39.

2	2	7	2	6
5	6	5	5	4
2	6	6	3	5
2	3	5	7	3

40.

4	5	7	2	4
3	4	7	7	3
2	6	4	3	3
7	6	3	5	5

Задача о кратчайшем пути

Найти длины кратчайших путей из левой верхней вершины графа в каждую вершину. Найти кратчайший путь из верхней, левой вершины в правую, нижнюю. Вершины графа обозначены кружочками. Число между вершинами обозначает длину неориентированного ребра, соединяющего эти вершины, изображение ребра опущено.

1. ° 6 ° 5 ° 3 ° 2 °
5 3 6 6 6
° 2 ° 7 ° 4 ° 7 °
4 6 7 2 4
° 5 ° 2 ° 4 ° 5 °
6 5 3 5 3
° 6 ° 5 ° 7 ° 6 °

2. ° 2 ° 6 ° 5 ° 2 °
7 2 2 6 2
° 3 ° 3 ° 7 ° 3 °
4 7 4 2 5
° 4 ° 6 ° 5 ° 3 °
4 3 3 5 4
° 7 ° 6 ° 3 ° 5 °

3. ° 5 ° 7 ° 2 ° 7 °
6 7 5 4 5
° 5 ° 4 ° 3 ° 3 °
4 4 2 7 4
° 4 ° 2 ° 3 ° 5 °
4 4 5 2 5
° 5 ° 2 ° 4 ° 5 °

4. ° 2 ° 3 ° 2 ° 5 °
2 2 5 5 2
° 3 ° 4 ° 2 ° 7 °
6 4 6 5 6
° 4 ° 4 ° 3 ° 5 °
4 3 2 3 2
° 4 ° 7 ° 3 ° 3 °

5. ° 6 ° 3 ° 3 ° 2 °
3 4 4 4 7
° 5 ° 7 ° 5 ° 5 °
6 3 2 4 2
° 7 ° 4 ° 4 ° 7 °
5 2 3 2 4
° 5 ° 5 ° 7 ° 6 °

6. ° 4 ° 6 ° 4 ° 6 °
4 3 5 5 5
° 3 ° 4 ° 7 ° 3 °
5 6 4 3 5
° 3 ° 4 ° 4 ° 6 °
5 7 3 6 6
° 2 ° 3 ° 2 ° 6 °

7. ° 2 ° 2 ° 5 ° 3 °
3 6 5 4 6
° 4 ° 6 ° 5 ° 2 °
7 2 6 4 3
° 7 ° 6 ° 6 ° 4 °
6 4 3 4 2
° 5 ° 2 ° 2 ° 7 °

8. ° 3 ° 4 ° 5 ° 5 °
4 3 2 7 7
° 5 ° 6 ° 6 ° 4 °
4 2 6 2 3
° 6 ° 5 ° 6 ° 7 °
2 6 3 6 7
° 7 ° 2 ° 6 ° 4 °

9. ° 5 ° 2 ° 5 ° 4 °
4 3 7 3 4
° 3 ° 3 ° 3 ° 5 °
6 4 7 4 7
° 5 ° 5 ° 7 ° 3 °
7 3 3 3 2
° 6 ° 3 ° 4 ° 6 °

10. ° 4 ° 3 ° 2 ° 7 °
6 6 5 3 2
° 7 ° 5 ° 5 ° 7 °
5 7 3 2 6
° 3 ° 4 ° 3 ° 3 °
4 5 5 2 2
° 5 ° 3 ° 7 ° 4 °

11. ° 3 ° 4 ° 5 ° 3 °
3 2 2 6 3
° 7 ° 2 ° 3 ° 7 °
4 4 6 6 2
° 2 ° 4 ° 7 ° 5 °
2 3 7 6 3
° 2 ° 6 ° 3 ° 5 °

12. ° 3 ° 5 ° 5 ° 5 °
4 4 4 2 3
° 7 ° 7 ° 4 ° 7 °
7 7 6 3 6
° 2 ° 6 ° 2 ° 4 °
4 2 3 5 3
° 6 ° 4 ° 7 ° 6 °

13. ° 5 ° 3 ° 6 ° 7 °
5 5 4 3 7
° 3 ° 2 ° 7 ° 5 °
4 3 6 5 5
° 6 ° 2 ° 4 ° 4 °
7 5 2 6 4
° 7 ° 5 ° 5 ° 3 °

14. ° 4 ° 5 ° 2 ° 2 °
5 6 4 3 5
° 5 ° 7 ° 5 ° 6 °
3 6 4 2 3
° 6 ° 4 ° 6 ° 6 °
7 6 2 4 5
° 7 ° 5 ° 5 ° 2 °

15. ° 5 ° 2 ° 5 ° 4 °
6 2 3 5 4
° 3 ° 4 ° 6 ° 5 °
2 5 5 6 6
° 6 ° 6 ° 6 ° 3 °
6 3 2 7 3
° 3 ° 7 ° 3 ° 5 °

16. ∘ 6 ∘ 7 ∘ 4 ∘ 4 ∘ 6 3 5 3 5 ∘ 2 ∘ 5 ∘ 5 ∘ 3 ∘ 2 7 4 4 6 ∘ 4 ∘ 5 ∘ 2 ∘ 7 ∘ 4 6 5 4 4 ∘ 6 ∘ 4 ∘ 4 ∘ 5 ∘	**17**. ∘ 7 ∘ 5 ∘ 7 ∘ 6 ∘ 3 3 3 3 2 ∘ 4 ∘ 5 ∘ 3 ∘ 3 ∘ 4 2 2 6 7 ∘ 6 ∘ 5 ∘ 7 ∘ 5 ∘ 5 3 3 5 6 ∘ 6 ∘ 3 ∘ 5 ∘ 3 ∘	**18**. ∘ 5 ∘ 4 ∘ 4 ∘ 5 ∘ 5 2 2 2 4 ∘ 2 ∘ 6 ∘ 4 ∘ 6 ∘ 5 4 4 3 7 ∘ 2 ∘ 6 ∘ 6 ∘ 4 ∘ 4 6 3 7 4 ∘ 4 ∘ 3 ∘ 5 ∘ 5 ∘
19. ∘ 2 ∘ 7 ∘ 5 ∘ 4 ∘ 3 3 4 3 3 ∘ 4 ∘ 3 ∘ 5 ∘ 2 ∘ 5 6 5 2 3 ∘ 4 ∘ 3 ∘ 7 ∘ 4 ∘ 5 7 5 2 2 ∘ 5 ∘ 3 ∘ 7 ∘ 5 ∘	**20**. ∘ 7 ∘ 4 ∘ 3 ∘ 4 ∘ 2 6 6 7 7 ∘ 3 ∘ 7 ∘ 7 ∘ 7 ∘ 6 3 7 5 4 ∘ 2 ∘ 5 ∘ 4 ∘ 5 ∘ 5 7 2 2 4 ∘ 3 ∘ 2 ∘ 5 ∘ 2 ∘	**21**. ∘ 3 ∘ 3 ∘ 3 ∘ 6 ∘ 4 4 5 3 5 ∘ 7 ∘ 3 ∘ 6 ∘ 5 ∘ 7 6 4 4 2 ∘ 7 ∘ 5 ∘ 2 ∘ 6 ∘ 3 6 5 6 2 ∘ 4 ∘ 2 ∘ 2 ∘ 3 ∘
22. ∘ 4 ∘ 2 ∘ 4 ∘ 3 ∘ 4 4 7 3 4 ∘ 5 ∘ 3 ∘ 5 ∘ 5 ∘ 4 2 4 7 5 ∘ 4 ∘ 2 ∘ 2 ∘ 3 ∘ 2 2 6 2 6 ∘ 7 ∘ 6 ∘ 5 ∘ 5 ∘	**23**. ∘ 4 ∘ 3 ∘ 2 ∘ 5 ∘ 4 7 4 5 2 ∘ 3 ∘ 6 ∘ 3 ∘ 7 ∘ 2 7 3 3 4 ∘ 3 ∘ 5 ∘ 2 ∘ 2 ∘ 7 5 7 6 3 ∘ 7 ∘ 6 ∘ 3 ∘ 5 ∘	**24**. ∘ 2 ∘ 2 ∘ 5 ∘ 3 ∘ 2 2 7 4 4 ∘ 3 ∘ 2 ∘ 7 ∘ 6 ∘ 7 2 7 7 5 ∘ 5 ∘ 4 ∘ 7 ∘ 3 ∘ 7 2 2 5 5 ∘ 7 ∘ 7 ∘ 4 ∘ 5 ∘
25. ∘ 4 ∘ 4 ∘ 3 ∘ 2 ∘ 4 5 4 7 2 ∘ 6 ∘ 3 ∘ 7 ∘ 3 ∘ 6 6 7 4 7 ∘ 5 ∘ 5 ∘ 6 ∘ 7 ∘ 7 4 6 2 5 ∘ 7 ∘ 6 ∘ 4 ∘ 5 ∘	**26**. ∘ 7 ∘ 4 ∘ 5 ∘ 4 ∘ 7 7 7 6 2 ∘ 7 ∘ 7 ∘ 6 ∘ 2 ∘ 5 6 7 6 3 ∘ 2 ∘ 7 ∘ 5 ∘ 3 ∘ 7 3 4 3 5 ∘ 5 ∘ 2 ∘ 5 ∘ 3 ∘	**27**. ∘ 4 ∘ 6 ∘ 4 ∘ 5 ∘ 6 5 5 5 7 ∘ 2 ∘ 3 ∘ 4 ∘ 2 ∘ 2 4 7 7 2 ∘ 6 ∘ 2 ∘ 3 ∘ 3 ∘ 2 6 7 3 5 ∘ 5 ∘ 7 ∘ 4 ∘ 7 ∘
28. ∘ 3 ∘ 2 ∘ 4 ∘ 4 ∘ 5 6 5 2 3 ∘ 3 ∘ 4 ∘ 2 ∘ 6 ∘ 3 6 4 5 3 ∘ 2 ∘ 5 ∘ 4 ∘ 2 ∘ 5 6 5 7 6 ∘ 2 ∘ 2 ∘ 7 ∘ 4 ∘	**29**. ∘ 7 ∘ 6 ∘ 7 ∘ 6 ∘ 2 2 7 6 3 ∘ 7 ∘ 7 ∘ 2 ∘ 2 ∘ 6 2 3 3 3 ∘ 6 ∘ 2 ∘ 6 ∘ 2 ∘ 6 6 3 5 5 ∘ 4 ∘ 3 ∘ 5 ∘ 3 ∘	**30**. ∘ 6 ∘ 4 ∘ 7 ∘ 6 ∘ 4 4 4 4 3 ∘ 3 ∘ 6 ∘ 3 ∘ 7 ∘ 2 6 4 6 7 ∘ 4 ∘ 7 ∘ 6 ∘ 4 ∘ 2 5 5 7 3 ∘ 3 ∘ 7 ∘ 7 ∘ 7 ∘

31.
```
 ° 6 ° 6 ° 5 ° 2 °
2   4   2   7   2
 ° 5 ° 2 ° 4 ° 3 °
7   4   7   6   3
 ° 2 ° 4 ° 2 ° 7 °
7   6   2   3   4
 ° 3 ° 6 ° 3 ° 7 °
```

32.
```
 ° 3 ° 3 ° 3 ° 4 °
7   2   5   5   6
 ° 3 ° 3 ° 7 ° 5 °
2   3   3   7   3
 ° 6 ° 4 ° 5 ° 3 °
6   2   2   7   4
 ° 5 ° 2 ° 7 ° 7 °
```

33.
```
 ° 7 ° 5 ° 7 ° 6 °
2   4   4   6   5
 ° 5 ° 4 ° 2 ° 6 °
6   7   6   6   7
 ° 7 ° 5 ° 7 ° 4 °
3   3   6   2   2
 ° 6 ° 2 ° 5 ° 7 °
```

34.
```
 ° 6 ° 3 ° 6 ° 5 °
6   7   4   4   4
 ° 5 ° 3 ° 7 ° 2 °
6   3   3   2   3
 ° 5 ° 4 ° 6 ° 6 °
7   3   3   3   7
 ° 3 ° 7 ° 4 ° 6 °
```

35.
```
 ° 4 ° 3 ° 7 ° 3 °
2   3   2   6   3
 ° 2 ° 4 ° 2 ° 2 °
3   7   6   5   6
 ° 4 ° 3 ° 3 ° 3 °
6   7   2   5   7
 ° 2 ° 7 ° 7 ° 6 °
```

36.
```
 ° 5 ° 6 ° 4 ° 6 °
7   5   3   3   5
 ° 3 ° 6 ° 2 ° 2 °
7   5   4   5   4
 ° 2 ° 5 ° 3 ° 5 °
5   7   3   4   5
 ° 5 ° 6 ° 6 ° 2 °
```

37.
```
 ° 7 ° 7 ° 3 ° 2 °
7   7   7   6   7
 ° 4 ° 3 ° 4 ° 2 °
7   4   5   4   2
 ° 3 ° 7 ° 6 ° 6 °
2   3   7   4   3
 ° 2 ° 3 ° 3 ° 4 °
```

38.
```
 ° 5 ° 7 ° 5 ° 5 °
6   2   2   3   4
 ° 7 ° 6 ° 7 ° 3 °
5   3   7   7   3
 ° 7 ° 7 ° 7 ° 3 °
4   2   7   3   6
 ° 3 ° 3 ° 2 ° 2 °
```

39.
```
 ° 2 ° 2 ° 7 ° 2 °
4   7   5   5   4
 ° 5 ° 6 ° 5 ° 5 °
2   3   7   7   5
 ° 2 ° 6 ° 6 ° 3 °
6   7   5   4   2
 ° 2 ° 3 ° 5 ° 7 °
```

40.
```
 ° 4 ° 5 ° 7 ° 2 °
3   4   6   4   5
 ° 3 ° 4 ° 7 ° 7 °
7   3   6   4   6
 ° 2 ° 6 ° 4 ° 3 °
2   3   6   4   4
 ° 7 ° 6 ° 3 ° 5 °
```

Задача о назначениях

Решить задачу о назначениях на максимум и на минимум.

1. $\begin{pmatrix} 2 & 6 & 8 & 11 & 6 & 8 & 10 \\ 12 & 1 & 1 & 15 & 4 & 12 & 6 \\ 14 & 11 & 4 & 1 & 9 & 4 & 15 \\ 12 & 9 & 15 & 10 & 15 & 14 & 4 \\ 14 & 14 & 15 & 10 & 6 & 4 & 12 \\ 1 & 5 & 11 & 9 & 8 & 1 & 4 \\ 4 & 1 & 3 & 10 & 1 & 2 & 7 \end{pmatrix}$

2. $\begin{pmatrix} 10 & 13 & 14 & 13 & 5 & 13 & 6 \\ 2 & 10 & 12 & 11 & 2 & 8 & 11 \\ 11 & 3 & 13 & 15 & 12 & 2 & 12 \\ 11 & 12 & 5 & 15 & 3 & 1 & 6 \\ 8 & 1 & 6 & 15 & 7 & 9 & 2 \\ 3 & 4 & 2 & 4 & 14 & 8 & 4 \\ 5 & 6 & 5 & 13 & 1 & 2 & 9 \end{pmatrix}$

3. $\begin{pmatrix} 12 & 9 & 5 & 2 & 11 & 10 & 2 \\ 4 & 8 & 6 & 14 & 15 & 15 & 15 \\ 5 & 9 & 3 & 3 & 15 & 8 & 9 \\ 11 & 9 & 14 & 15 & 4 & 14 & 2 \\ 14 & 14 & 2 & 2 & 4 & 10 & 15 \\ 15 & 1 & 12 & 7 & 7 & 11 & 3 \\ 13 & 9 & 11 & 3 & 14 & 11 & 1 \end{pmatrix}$

4. $\begin{pmatrix} 11 & 15 & 4 & 6 & 2 & 10 & 9 \\ 4 & 13 & 8 & 1 & 11 & 10 & 15 \\ 7 & 14 & 11 & 11 & 4 & 8 & 14 \\ 2 & 15 & 15 & 14 & 10 & 2 & 8 \\ 1 & 7 & 1 & 9 & 12 & 12 & 11 \\ 10 & 5 & 2 & 5 & 6 & 1 & 9 \\ 8 & 5 & 4 & 6 & 7 & 4 & 6 \end{pmatrix}$

5. $\begin{pmatrix} 4 & 15 & 8 & 14 & 4 & 5 & 12 \\ 2 & 10 & 14 & 8 & 6 & 7 & 14 \\ 13 & 1 & 10 & 6 & 9 & 8 & 10 \\ 3 & 8 & 15 & 3 & 1 & 9 & 2 \\ 12 & 8 & 12 & 5 & 15 & 5 & 12 \\ 11 & 10 & 7 & 3 & 13 & 3 & 9 \\ 12 & 7 & 5 & 1 & 9 & 10 & 4 \end{pmatrix}$

6. $\begin{pmatrix} 10 & 3 & 1 & 8 & 3 & 10 & 12 \\ 15 & 8 & 12 & 6 & 2 & 9 & 14 \\ 11 & 14 & 1 & 10 & 12 & 14 & 10 \\ 7 & 15 & 13 & 14 & 2 & 14 & 10 \\ 1 & 4 & 4 & 11 & 10 & 13 & 12 \\ 14 & 14 & 8 & 8 & 14 & 7 & 7 \\ 10 & 11 & 4 & 5 & 9 & 10 & 1 \end{pmatrix}$

7. $\begin{pmatrix} 12 & 1 & 7 & 3 & 1 & 4 & 3 \\ 9 & 5 & 14 & 4 & 3 & 4 & 9 \\ 8 & 7 & 1 & 13 & 9 & 5 & 10 \\ 8 & 14 & 6 & 5 & 14 & 3 & 12 \\ 15 & 13 & 2 & 2 & 8 & 3 & 8 \\ 10 & 5 & 12 & 3 & 14 & 11 & 15 \\ 8 & 3 & 14 & 6 & 5 & 3 & 7 \end{pmatrix}$

8. $\begin{pmatrix} 10 & 14 & 10 & 3 & 15 & 7 & 11 \\ 15 & 1 & 8 & 13 & 6 & 11 & 1 \\ 1 & 15 & 10 & 1 & 13 & 7 & 14 \\ 1 & 7 & 3 & 11 & 1 & 4 & 3 \\ 5 & 15 & 2 & 14 & 5 & 7 & 5 \\ 1 & 1 & 7 & 13 & 8 & 13 & 7 \\ 2 & 15 & 12 & 2 & 8 & 11 & 15 \end{pmatrix}$

9. $\begin{pmatrix} 1 & 11 & 4 & 14 & 3 & 6 & 6 \\ 11 & 4 & 1 & 5 & 3 & 12 & 14 \\ 2 & 3 & 2 & 6 & 4 & 3 & 6 \\ 2 & 4 & 8 & 11 & 10 & 5 & 4 \\ 14 & 15 & 4 & 5 & 4 & 5 & 7 \\ 14 & 3 & 8 & 13 & 11 & 2 & 4 \\ 11 & 8 & 2 & 4 & 5 & 1 & 4 \end{pmatrix}$

10. $\begin{pmatrix} 11 & 6 & 7 & 11 & 1 & 2 & 5 \\ 8 & 4 & 5 & 7 & 6 & 2 & 1 \\ 1 & 7 & 11 & 6 & 2 & 13 & 13 \\ 6 & 2 & 13 & 13 & 1 & 1 & 1 \\ 12 & 8 & 7 & 5 & 3 & 15 & 7 \\ 6 & 4 & 3 & 2 & 3 & 15 & 4 \\ 12 & 15 & 4 & 8 & 10 & 13 & 11 \end{pmatrix}$

11. $\begin{pmatrix} 11 & 15 & 3 & 10 & 14 & 9 & 6 \\ 15 & 4 & 9 & 13 & 12 & 6 & 11 \\ 8 & 13 & 7 & 13 & 14 & 10 & 10 \\ 12 & 6 & 1 & 3 & 11 & 14 & 8 \\ 13 & 9 & 5 & 2 & 5 & 15 & 12 \\ 9 & 6 & 2 & 2 & 2 & 10 & 10 \\ 15 & 12 & 5 & 9 & 10 & 9 & 6 \end{pmatrix}$

12. $\begin{pmatrix} 11 & 2 & 6 & 3 & 8 & 14 & 15 \\ 3 & 11 & 13 & 12 & 1 & 14 & 4 \\ 1 & 5 & 4 & 3 & 8 & 12 & 13 \\ 13 & 5 & 8 & 7 & 11 & 6 & 12 \\ 8 & 1 & 12 & 4 & 15 & 6 & 2 \\ 7 & 2 & 10 & 10 & 5 & 14 & 12 \\ 10 & 12 & 14 & 7 & 3 & 15 & 1 \end{pmatrix}$

13. $\begin{pmatrix} 11 & 2 & 6 & 3 & 7 & 13 & 12 \\ 15 & 2 & 2 & 7 & 8 & 4 & 14 \\ 10 & 2 & 4 & 4 & 8 & 2 & 8 \\ 1 & 14 & 8 & 5 & 11 & 8 & 6 \\ 7 & 4 & 10 & 11 & 11 & 11 & 6 \\ 15 & 7 & 15 & 7 & 6 & 2 & 9 \\ 5 & 14 & 12 & 12 & 5 & 4 & 10 \end{pmatrix}$

14. $\begin{pmatrix} 9 & 9 & 7 & 1 & 2 & 6 & 8 \\ 6 & 8 & 1 & 3 & 11 & 9 & 8 \\ 11 & 2 & 12 & 4 & 2 & 13 & 15 \\ 11 & 6 & 12 & 8 & 10 & 15 & 14 \\ 13 & 12 & 8 & 11 & 3 & 10 & 11 \\ 4 & 3 & 2 & 8 & 14 & 12 & 8 \\ 11 & 7 & 8 & 4 & 6 & 1 & 11 \end{pmatrix}$

15. $\begin{pmatrix} 9 & 6 & 3 & 5 & 7 & 7 & 10 \\ 14 & 11 & 13 & 5 & 12 & 1 & 5 \\ 12 & 8 & 14 & 2 & 8 & 5 & 13 \\ 5 & 10 & 4 & 11 & 11 & 5 & 6 \\ 15 & 5 & 4 & 11 & 11 & 6 & 12 \\ 4 & 8 & 11 & 6 & 10 & 11 & 4 \\ 2 & 1 & 10 & 5 & 10 & 4 & 1 \end{pmatrix}$

16. $\begin{pmatrix} 11 & 10 & 10 & 14 & 1 & 9 & 1 \\ 11 & 15 & 1 & 8 & 11 & 9 & 8 \\ 1 & 15 & 1 & 15 & 1 & 13 & 12 \\ 1 & 1 & 7 & 12 & 9 & 1 & 10 \\ 10 & 6 & 5 & 11 & 8 & 3 & 2 \\ 4 & 5 & 13 & 10 & 15 & 6 & 11 \\ 15 & 8 & 9 & 1 & 15 & 2 & 15 \end{pmatrix}$

17. $\begin{pmatrix} 13 & 5 & 7 & 3 & 9 & 5 & 4 \\ 8 & 13 & 11 & 14 & 13 & 14 & 7 \\ 15 & 1 & 7 & 3 & 2 & 13 & 1 \\ 3 & 11 & 13 & 11 & 8 & 10 & 14 \\ 7 & 12 & 5 & 9 & 14 & 8 & 6 \\ 14 & 8 & 13 & 9 & 2 & 14 & 9 \\ 11 & 2 & 1 & 10 & 13 & 6 & 5 \end{pmatrix}$

18. $\begin{pmatrix} 6 & 15 & 13 & 15 & 12 & 10 & 1 \\ 3 & 14 & 6 & 6 & 5 & 1 & 8 \\ 9 & 5 & 2 & 8 & 9 & 1 & 1 \\ 10 & 5 & 13 & 13 & 15 & 1 & 2 \\ 3 & 13 & 3 & 8 & 11 & 12 & 7 \\ 7 & 1 & 10 & 11 & 13 & 3 & 4 \\ 6 & 14 & 13 & 15 & 6 & 11 & 8 \end{pmatrix}$

19. $\begin{pmatrix} 13 & 13 & 13 & 10 & 14 & 5 & 12 \\ 6 & 15 & 15 & 5 & 7 & 11 & 2 \\ 3 & 7 & 1 & 2 & 3 & 3 & 2 \\ 15 & 11 & 6 & 8 & 4 & 10 & 13 \\ 9 & 14 & 1 & 1 & 4 & 4 & 9 \\ 12 & 2 & 9 & 6 & 13 & 13 & 6 \\ 4 & 1 & 15 & 7 & 6 & 11 & 12 \end{pmatrix}$

20. $\begin{pmatrix} 2 & 3 & 13 & 1 & 1 & 3 & 13 \\ 8 & 12 & 12 & 10 & 12 & 1 & 8 \\ 10 & 6 & 7 & 13 & 14 & 14 & 5 \\ 8 & 1 & 2 & 10 & 12 & 3 & 7 \\ 8 & 2 & 7 & 8 & 2 & 3 & 7 \\ 8 & 5 & 10 & 13 & 2 & 2 & 10 \\ 4 & 3 & 9 & 1 & 12 & 3 & 5 \end{pmatrix}$

21. $\begin{pmatrix} 4 & 7 & 15 & 14 & 2 & 7 & 3 \\ 3 & 12 & 1 & 9 & 14 & 14 & 14 \\ 2 & 12 & 4 & 5 & 4 & 15 & 3 \\ 11 & 7 & 14 & 8 & 6 & 5 & 1 \\ 8 & 1 & 3 & 1 & 6 & 11 & 4 \\ 5 & 5 & 12 & 1 & 4 & 12 & 4 \\ 11 & 3 & 15 & 5 & 10 & 15 & 7 \end{pmatrix}$

22. $\begin{pmatrix} 9 & 1 & 15 & 3 & 3 & 2 & 6 \\ 14 & 13 & 14 & 2 & 13 & 13 & 2 \\ 1 & 15 & 9 & 14 & 5 & 3 & 4 \\ 3 & 14 & 7 & 15 & 7 & 11 & 8 \\ 9 & 3 & 15 & 5 & 1 & 6 & 7 \\ 14 & 6 & 4 & 10 & 11 & 12 & 14 \\ 2 & 3 & 8 & 11 & 5 & 9 & 10 \end{pmatrix}$

23. $\begin{pmatrix} 15 & 11 & 11 & 7 & 3 & 6 & 3 \\ 9 & 5 & 9 & 4 & 15 & 5 & 11 \\ 8 & 12 & 11 & 7 & 10 & 8 & 8 \\ 8 & 15 & 2 & 6 & 5 & 7 & 4 \\ 14 & 1 & 14 & 7 & 4 & 2 & 14 \\ 10 & 15 & 10 & 9 & 2 & 7 & 4 \\ 1 & 13 & 10 & 10 & 1 & 14 & 13 \end{pmatrix}$

24. $\begin{pmatrix} 6 & 10 & 9 & 1 & 3 & 8 & 5 \\ 2 & 6 & 6 & 2 & 1 & 14 & 5 \\ 2 & 1 & 7 & 3 & 15 & 2 & 11 \\ 2 & 2 & 7 & 9 & 2 & 12 & 1 \\ 13 & 13 & 10 & 9 & 9 & 3 & 14 \\ 7 & 3 & 5 & 11 & 4 & 3 & 9 \\ 3 & 9 & 12 & 14 & 10 & 10 & 14 \end{pmatrix}$

25. $\begin{pmatrix} 5 & 11 & 1 & 9 & 10 & 5 & 8 \\ 11 & 5 & 12 & 15 & 8 & 8 & 8 \\ 12 & 3 & 2 & 15 & 10 & 8 & 1 \\ 8 & 6 & 4 & 7 & 14 & 1 & 15 \\ 6 & 3 & 14 & 7 & 9 & 13 & 14 \\ 10 & 4 & 12 & 14 & 5 & 3 & 1 \\ 3 & 14 & 11 & 6 & 11 & 12 & 11 \end{pmatrix}$

26. $\begin{pmatrix} 1 & 7 & 10 & 2 & 4 & 2 & 5 \\ 4 & 7 & 3 & 5 & 11 & 13 & 7 \\ 9 & 2 & 2 & 4 & 13 & 15 & 11 \\ 2 & 5 & 13 & 10 & 3 & 5 & 1 \\ 4 & 8 & 11 & 7 & 15 & 6 & 9 \\ 4 & 13 & 1 & 13 & 3 & 15 & 3 \\ 5 & 7 & 5 & 1 & 15 & 11 & 3 \end{pmatrix}$

27. $\begin{pmatrix} 14 & 4 & 5 & 10 & 2 & 7 & 13 \\ 11 & 14 & 5 & 3 & 9 & 12 & 8 \\ 6 & 5 & 13 & 11 & 14 & 5 & 15 \\ 3 & 4 & 2 & 6 & 12 & 15 & 14 \\ 6 & 4 & 8 & 1 & 8 & 14 & 11 \\ 2 & 1 & 3 & 12 & 13 & 6 & 9 \\ 10 & 14 & 7 & 4 & 2 & 10 & 13 \end{pmatrix}$

28. $\begin{pmatrix} 7 & 10 & 14 & 3 & 7 & 1 & 6 \\ 1 & 14 & 3 & 11 & 12 & 3 & 6 \\ 13 & 15 & 14 & 8 & 8 & 12 & 3 \\ 1 & 14 & 9 & 7 & 10 & 3 & 15 \\ 13 & 8 & 2 & 10 & 1 & 13 & 2 \\ 9 & 4 & 14 & 4 & 2 & 1 & 1 \\ 9 & 3 & 12 & 13 & 7 & 1 & 10 \end{pmatrix}$

29. $\begin{pmatrix} 4 & 3 & 13 & 7 & 6 & 13 & 3 \\ 12 & 14 & 13 & 10 & 5 & 2 & 10 \\ 4 & 13 & 7 & 7 & 8 & 6 & 7 \\ 11 & 6 & 3 & 6 & 8 & 3 & 11 \\ 1 & 6 & 15 & 13 & 11 & 7 & 13 \\ 15 & 7 & 13 & 13 & 2 & 14 & 1 \\ 5 & 11 & 1 & 12 & 9 & 2 & 2 \end{pmatrix}$

30. $\begin{pmatrix} 7 & 12 & 5 & 14 & 9 & 9 & 6 \\ 3 & 5 & 11 & 2 & 14 & 11 & 14 \\ 11 & 13 & 15 & 5 & 14 & 15 & 1 \\ 1 & 1 & 13 & 3 & 7 & 9 & 3 \\ 7 & 10 & 11 & 15 & 8 & 12 & 9 \\ 6 & 14 & 10 & 12 & 10 & 3 & 9 \\ 1 & 15 & 9 & 1 & 11 & 7 & 15 \end{pmatrix}$

31. $\begin{pmatrix} 8 & 10 & 9 & 7 & 9 & 5 & 9 \\ 15 & 6 & 7 & 9 & 4 & 10 & 11 \\ 6 & 13 & 7 & 1 & 12 & 5 & 2 \\ 15 & 14 & 3 & 13 & 11 & 5 & 3 \\ 5 & 15 & 5 & 15 & 11 & 8 & 13 \\ 14 & 6 & 9 & 14 & 8 & 2 & 15 \\ 4 & 11 & 10 & 1 & 12 & 14 & 11 \end{pmatrix}$

32. $\begin{pmatrix} 11 & 11 & 15 & 15 & 10 & 9 & 15 \\ 3 & 9 & 2 & 9 & 1 & 13 & 12 \\ 13 & 2 & 12 & 3 & 6 & 1 & 11 \\ 9 & 5 & 8 & 3 & 1 & 2 & 3 \\ 10 & 8 & 4 & 1 & 10 & 6 & 5 \\ 4 & 4 & 12 & 3 & 8 & 1 & 9 \\ 7 & 8 & 11 & 1 & 12 & 2 & 15 \end{pmatrix}$

33. $\begin{pmatrix} 11 & 8 & 3 & 8 & 8 & 5 & 10 \\ 6 & 8 & 5 & 8 & 15 & 15 & 5 \\ 15 & 15 & 2 & 7 & 1 & 10 & 10 \\ 1 & 11 & 4 & 3 & 9 & 9 & 1 \\ 10 & 4 & 6 & 4 & 15 & 15 & 10 \\ 2 & 13 & 2 & 6 & 15 & 5 & 4 \\ 2 & 1 & 2 & 5 & 13 & 13 & 10 \end{pmatrix}$

34. $\begin{pmatrix} 6 & 7 & 15 & 9 & 14 & 8 & 11 \\ 14 & 3 & 8 & 10 & 2 & 10 & 7 \\ 5 & 8 & 11 & 2 & 6 & 4 & 12 \\ 11 & 9 & 14 & 8 & 13 & 11 & 15 \\ 12 & 3 & 4 & 2 & 14 & 12 & 2 \\ 9 & 7 & 5 & 14 & 8 & 9 & 12 \\ 1 & 15 & 7 & 14 & 7 & 5 & 10 \end{pmatrix}$

35. $\begin{pmatrix} 8 & 5 & 7 & 13 & 15 & 4 & 13 \\ 15 & 1 & 6 & 11 & 7 & 13 & 3 \\ 12 & 15 & 1 & 3 & 3 & 13 & 7 \\ 15 & 4 & 2 & 9 & 9 & 3 & 6 \\ 12 & 11 & 8 & 6 & 4 & 11 & 7 \\ 3 & 7 & 1 & 1 & 5 & 12 & 11 \\ 3 & 2 & 13 & 6 & 11 & 8 & 12 \end{pmatrix}$

36. $\begin{pmatrix} 15 & 9 & 6 & 13 & 5 & 9 & 8 \\ 13 & 15 & 6 & 13 & 1 & 5 & 4 \\ 9 & 3 & 6 & 4 & 1 & 10 & 5 \\ 9 & 5 & 8 & 11 & 1 & 4 & 15 \\ 14 & 4 & 2 & 4 & 10 & 15 & 8 \\ 13 & 2 & 11 & 7 & 4 & 15 & 6 \\ 8 & 4 & 10 & 5 & 10 & 5 & 8 \end{pmatrix}$

37. $\begin{pmatrix} 8 & 6 & 14 & 7 & 7 & 4 & 9 \\ 6 & 7 & 5 & 6 & 4 & 1 & 8 \\ 7 & 12 & 11 & 15 & 9 & 12 & 12 \\ 15 & 1 & 8 & 15 & 13 & 4 & 6 \\ 2 & 12 & 8 & 13 & 9 & 15 & 6 \\ 1 & 5 & 12 & 4 & 2 & 8 & 9 \\ 8 & 9 & 8 & 10 & 11 & 14 & 9 \end{pmatrix}$

38. $\begin{pmatrix} 5 & 8 & 10 & 15 & 15 & 12 & 7 \\ 4 & 4 & 15 & 8 & 1 & 3 & 8 \\ 11 & 2 & 4 & 4 & 11 & 6 & 2 \\ 10 & 9 & 14 & 13 & 9 & 4 & 2 \\ 7 & 8 & 3 & 13 & 9 & 8 & 6 \\ 13 & 13 & 11 & 14 & 11 & 8 & 10 \\ 8 & 14 & 8 & 7 & 6 & 11 & 3 \end{pmatrix}$

39. $\begin{pmatrix} 15 & 6 & 3 & 3 & 12 & 3 & 8 \\ 13 & 13 & 9 & 2 & 6 & 4 & 7 \\ 11 & 14 & 12 & 4 & 14 & 12 & 1 \\ 7 & 12 & 4 & 8 & 11 & 6 & 10 \\ 12 & 12 & 14 & 3 & 2 & 8 & 7 \\ 14 & 9 & 14 & 8 & 1 & 10 & 3 \\ 14 & 14 & 6 & 1 & 11 & 11 & 1 \end{pmatrix}$

40. $\begin{pmatrix} 7 & 6 & 5 & 9 & 1 & 4 & 14 \\ 12 & 1 & 14 & 10 & 9 & 1 & 1 \\ 7 & 6 & 11 & 4 & 9 & 7 & 11 \\ 5 & 13 & 11 & 1 & 8 & 6 & 6 \\ 6 & 9 & 5 & 2 & 14 & 9 & 10 \\ 5 & 7 & 9 & 8 & 5 & 8 & 12 \\ 10 & 3 & 8 & 14 & 13 & 14 & 7 \end{pmatrix}$

Транспортная задача

Решить транспортную задачу.

1. $$\begin{array}{ccccc|c} 13 & 5 & 13 & 2 & 5 & 100 \\ 1 & 13 & 6 & 7 & 8 & 130 \\ 15 & 7 & 3 & 3 & 5 & 140 \\ 2 & 6 & 12 & 3 & 10 & 180 \\ \hline 50 & 160 & 130 & 10 & 200 & \end{array}$$

2. $$\begin{array}{ccccc|c} 3 & 10 & 2 & 4 & 10 & 170 \\ 2 & 14 & 5 & 5 & 15 & 90 \\ 5 & 3 & 11 & 14 & 7 & 30 \\ 2 & 5 & 1 & 5 & 15 & 170 \\ \hline 10 & 60 & 200 & 40 & 150 & \end{array}$$

3. $$\begin{array}{ccccc|c} 4 & 4 & 8 & 6 & 4 & 90 \\ 15 & 10 & 12 & 4 & 1 & 100 \\ 12 & 4 & 15 & 12 & 1 & 200 \\ 5 & 12 & 7 & 2 & 1 & 170 \\ \hline 180 & 80 & 70 & 160 & 70 & \end{array}$$

4. $$\begin{array}{ccccc|c} 12 & 5 & 15 & 13 & 10 & 10 \\ 1 & 2 & 14 & 4 & 8 & 50 \\ 12 & 15 & 2 & 5 & 11 & 170 \\ 9 & 3 & 1 & 2 & 5 & 330 \\ \hline 20 & 100 & 140 & 170 & 130 & \end{array}$$

5. $$\begin{array}{ccccc|c} 6 & 13 & 2 & 7 & 2 & 60 \\ 14 & 7 & 13 & 11 & 9 & 90 \\ 11 & 2 & 11 & 5 & 15 & 50 \\ 10 & 10 & 9 & 14 & 10 & 490 \\ \hline 120 & 110 & 90 & 190 & 180 & \end{array}$$

6. $$\begin{array}{ccccc|c} 14 & 12 & 8 & 1 & 3 & 190 \\ 8 & 7 & 13 & 2 & 10 & 90 \\ 3 & 7 & 11 & 1 & 14 & 100 \\ 10 & 7 & 3 & 4 & 14 & 400 \\ \hline 90 & 180 & 140 & 180 & 190 & \end{array}$$

7. $$\begin{array}{ccccc|c} 8 & 6 & 6 & 13 & 7 & 50 \\ 5 & 8 & 12 & 9 & 6 & 200 \\ 5 & 1 & 13 & 12 & 3 & 40 \\ 8 & 5 & 8 & 2 & 4 & 320 \\ \hline 40 & 150 & 200 & 120 & 100 & \end{array}$$

8. $$\begin{array}{ccccc|c} 13 & 12 & 13 & 15 & 1 & 40 \\ 10 & 11 & 12 & 14 & 13 & 20 \\ 4 & 4 & 9 & 13 & 7 & 140 \\ 5 & 7 & 13 & 7 & 1 & 160 \\ \hline 70 & 80 & 150 & 40 & 20 & \end{array}$$

9. $$\begin{array}{ccccc|c} 9 & 11 & 12 & 10 & 15 & 170 \\ 8 & 2 & 1 & 7 & 14 & 70 \\ 6 & 12 & 4 & 6 & 9 & 70 \\ 13 & 9 & 7 & 9 & 15 & 160 \\ \hline 60 & 100 & 50 & 70 & 190 & \end{array}$$

10. $$\begin{array}{ccccc|c} 2 & 10 & 10 & 3 & 10 & 90 \\ 1 & 9 & 1 & 3 & 3 & 190 \\ 15 & 11 & 8 & 13 & 11 & 20 \\ 6 & 2 & 4 & 6 & 7 & 310 \\ \hline 50 & 160 & 180 & 60 & 160 & \end{array}$$

11. $$\begin{array}{ccccc|c} 5 & 10 & 13 & 2 & 5 & 130 \\ 7 & 9 & 14 & 9 & 1 & 80 \\ 7 & 5 & 5 & 12 & 14 & 10 \\ 11 & 1 & 8 & 2 & 15 & 360 \\ \hline 100 & 200 & 110 & 20 & 150 & \end{array}$$

12. $$\begin{array}{ccccc|c} 14 & 4 & 9 & 1 & 11 & 100 \\ 14 & 5 & 4 & 5 & 12 & 40 \\ 9 & 4 & 5 & 10 & 15 & 180 \\ 3 & 5 & 6 & 8 & 15 & 340 \\ \hline 200 & 190 & 60 & 10 & 200 & \end{array}$$

13.

3	14	4	10	6	80
8	8	15	15	3	170
4	15	14	8	10	180
5	12	6	5	6	190
140	50	120	160	150	

14.

12	10	5	5	15	10
12	4	11	11	12	200
1	15	2	10	1	100
5	1	2	5	11	260
110	100	140	110	110	

15.

2	2	6	14	7	200
9	13	11	8	7	60
12	13	1	5	1	140
2	6	10	5	2	70
130	180	80	70	10	

16.

8	15	4	11	9	50
3	5	8	12	6	170
11	14	4	14	13	180
7	8	13	9	12	180
100	40	10	10	420	

17.

5	6	5	3	12	100
15	2	7	12	4	200
14	12	10	4	1	80
15	15	6	8	11	210
90	10	190	180	120	

18.

9	3	3	15	4	60
11	11	3	12	4	50
11	7	9	2	9	160
12	15	11	2	3	180
20	20	120	170	120	

19.

6	6	3	11	1	130
2	3	2	4	4	120
7	15	3	3	11	160
2	8	7	10	12	150
140	200	60	60	100	

20.

7	14	1	2	13	50
9	10	3	1	2	110
9	11	7	12	10	30
5	10	5	1	3	430
120	170	150	20	160	

21.

7	7	5	12	7	100
6	11	6	4	15	90
15	15	15	8	7	30
7	10	7	8	15	350
170	150	40	130	80	

22.

6	5	13	7	2	50
12	9	13	2	12	160
9	2	9	12	7	130
3	5	8	1	2	310
180	90	90	180	110	

23.

12	8	3	10	8	200
1	11	3	5	1	120
9	2	4	7	15	180
7	4	12	3	8	100
90	120	180	50	160	

24.

4	11	4	7	12	140
8	6	3	2	8	110
14	13	4	2	10	60
11	9	2	7	12	110
10	20	140	70	180	

25.

13	8	5	1	5	120
1	8	10	10	2	180
13	2	8	2	15	130
8	2	4	12	13	90
120	100	60	10	230	

26 .

7	7	14	9	2	180
9	8	9	12	10	50
6	14	14	7	6	170
1	5	7	1	1	120
90	160	10	170	90	

27.

2	10	11	3	7	120
5	8	10	2	11	200
1	7	13	9	4	190
3	5	1	3	6	110
110	130	190	100	90	

28.

13	2	6	13	14	190
5	10	4	1	4	30
5	11	5	2	9	160
9	5	12	9	5	130
60	20	190	10	230	

29.

3	1	1	9	3	150
5	8	1	12	2	140
3	10	2	4	3	40
5	2	4	10	8	210
150	10	170	190	20	

30.

4	1	13	2	15	90
8	6	2	13	8	110
12	13	2	15	10	120
9	1	15	15	2	260
10	130	100	180	160	

31.

7	14	14	11	2	110
5	3	8	5	15	160
2	7	12	14	12	160
3	7	14	10	6	300
100	140	160	190	140	

32.

7	3	14	15	6	10
15	11	9	10	7	80
6	3	9	11	2	50
14	6	14	12	5	260
30	70	180	30	90	

33.

1	11	15	5	10	110
10	8	7	12	9	120
4	7	15	12	7	200
2	12	8	12	2	200
140	10	60	40	380	

34.

12	13	11	14	5	70
4	15	13	8	5	130
11	1	10	3	6	20
7	7	4	6	4	390
10	200	130	130	140	

35.

2	5	3	11	4	180
7	10	3	10	11	30
4	14	14	12	7	150
3	4	7	2	4	330
200	100	80	140	170	

36.

4	1	5	13	12	50
4	7	10	8	12	170
13	8	4	14	12	60
4	11	14	2	15	340
160	30	110	120	200	

37.

1	12	12	15	3	100
10	6	6	10	11	100
1	7	12	8	7	40
7	13	13	7	15	450
30	180	110	180	190	

38

3	12	10	13	4	140
2	10	5	1	9	30
10	2	4	1	5	200
11	5	8	4	3	240
60	100	50	200	200	

39.

1	1	12	15	8	200
12	5	1	15	9	190
15	7	5	12	12	140
15	5	14	5	9	170
190	190	170	40	110	

40.

4	2	1	5	1	80
9	14	8	15	10	180
5	2	1	6	14	140
7	12	6	7	9	180
100	180	20	160	120	

Задача коммивояжера

Решить задачу коммивояжера.

1.

×	13	6	18	11	13
13	×	10	7	16	20
20	8	×	2	10	18
1	9	9	×	10	15
13	9	15	16	×	12
8	13	10	14	11	×

2.

×	9	16	17	7	15
15	×	8	15	18	13
18	5	×	3	14	6
15	7	6	×	20	3
9	2	6	10	×	11
5	19	1	16	10	×

3.

×	3	1	15	4	9
12	×	2	10	9	4
1	16	×	5	20	8
1	6	8	×	3	18
2	9	9	11	×	17
1	15	5	16	11	×

4.

×	2	3	8	15	3
20	×	3	5	1	4
4	11	×	15	6	14
16	8	10	×	10	13
1	19	16	10	×	6
20	10	10	6	10	×

5.

×	12	1	1	7	2
20	×	16	8	2	17
16	3	×	20	4	8
8	20	16	×	15	10
3	14	14	5	×	18
13	16	2	20	12	×

6.

×	11	7	4	16	3
9	×	19	13	11	3
13	3	×	4	3	12
19	19	9	×	14	7
19	13	3	12	×	15
16	16	12	2	1	×

7.

×	11	20	1	6	1
16	×	15	4	6	9
4	3	×	4	11	19
8	14	4	×	10	15
18	16	16	4	×	4
10	16	18	1	11	×

8.

×	16	10	16	17	10
17	×	7	7	3	1
13	3	×	10	16	14
7	15	11	×	15	4
16	16	7	19	×	7
17	6	6	17	5	×

9.

×	17	11	15	20	14
10	×	11	3	9	16
20	20	×	15	7	1
19	6	12	×	11	5
13	16	9	13	×	1
14	9	5	7	4	×

10.

×	16	1	20	3	14
10	×	14	16	19	13
20	9	×	12	2	13
17	2	19	×	17	3
7	5	17	4	×	10
14	3	3	5	19	×

11.

×	20	16	16	8	5
5	×	9	1	20	5
10	7	×	4	17	10
10	11	5	×	20	1
11	14	15	5	×	10
17	14	10	15	16	×

12.

×	3	8	1	10	18
11	×	17	17	9	12
4	14	×	17	19	9
17	15	5	×	1	11
15	9	4	19	×	12
1	10	1	5	6	×

13.

×	19	11	16	11	14
6	×	13	13	5	9
9	3	×	9	8	3
4	19	15	×	1	11
19	3	11	6	×	3
13	5	18	11	1	×

14.

×	8	13	4	18	15
17	×	7	20	13	12
11	6	×	13	8	8
9	4	20	×	17	12
19	3	12	1	×	5
5	14	11	3	12	×

15.

×	12	6	12	11	15
10	×	20	16	7	4
3	7	×	9	10	7
15	18	14	×	20	16
4	3	9	7	×	8
15	3	20	1	11	×

16. $$\begin{array}{cccccc} \times & 15 & 8 & 10 & 12 & 6 \\ 19 & \times & 3 & 20 & 8 & 20 \\ 1 & 10 & \times & 19 & 8 & 17 \\ 2 & 13 & 12 & \times & 8 & 16 \\ 9 & 10 & 19 & 18 & \times & 17 \\ 14 & 20 & 11 & 20 & 8 & \times \end{array}$$

17. $$\begin{array}{cccccc} \times & 18 & 18 & 2 & 6 & 5 \\ 4 & \times & 15 & 12 & 11 & 10 \\ 8 & 19 & \times & 18 & 18 & 3 \\ 10 & 19 & 11 & \times & 15 & 13 \\ 5 & 11 & 11 & 16 & \times & 6 \\ 8 & 7 & 7 & 10 & 2 & \times \end{array}$$

18. $$\begin{array}{cccccc} \times & 3 & 7 & 13 & 4 & 18 \\ 6 & \times & 16 & 9 & 14 & 8 \\ 10 & 14 & \times & 9 & 12 & 7 \\ 6 & 5 & 10 & \times & 6 & 13 \\ 2 & 2 & 18 & 19 & \times & 13 \\ 1 & 10 & 16 & 1 & 15 & \times \end{array}$$

19. $$\begin{array}{cccccc} \times & 17 & 7 & 10 & 18 & 1 \\ 10 & \times & 16 & 2 & 7 & 14 \\ 2 & 1 & \times & 20 & 15 & 7 \\ 16 & 6 & 2 & \times & 4 & 10 \\ 1 & 10 & 7 & 20 & \times & 11 \\ 12 & 11 & 2 & 18 & 14 & \times \end{array}$$

20. $$\begin{array}{cccccc} \times & 9 & 8 & 17 & 18 & 11 \\ 8 & \times & 13 & 11 & 19 & 7 \\ 13 & 4 & \times & 13 & 14 & 16 \\ 12 & 16 & 12 & \times & 5 & 10 \\ 1 & 8 & 4 & 13 & \times & 6 \\ 11 & 16 & 14 & 2 & 18 & \times \end{array}$$

21. $$\begin{array}{cccccc} \times & 5 & 20 & 5 & 18 & 1 \\ 20 & \times & 2 & 1 & 10 & 20 \\ 13 & 2 & \times & 13 & 13 & 3 \\ 2 & 16 & 9 & \times & 14 & 18 \\ 2 & 11 & 3 & 8 & \times & 5 \\ 17 & 4 & 11 & 4 & 5 & \times \end{array}$$

22. $$\begin{array}{cccccc} \times & 9 & 19 & 18 & 13 & 8 \\ 12 & \times & 4 & 2 & 20 & 16 \\ 18 & 19 & \times & 19 & 2 & 8 \\ 14 & 20 & 5 & \times & 6 & 15 \\ 19 & 6 & 5 & 16 & \times & 1 \\ 14 & 16 & 13 & 2 & 8 & \times \end{array}$$

23. $$\begin{array}{cccccc} \times & 7 & 16 & 5 & 13 & 18 \\ 15 & \times & 20 & 7 & 17 & 12 \\ 1 & 19 & \times & 10 & 13 & 16 \\ 1 & 7 & 9 & \times & 13 & 15 \\ 6 & 7 & 12 & 11 & \times & 2 \\ 5 & 15 & 20 & 3 & 19 & \times \end{array}$$

24. $$\begin{array}{cccccc} \times & 15 & 7 & 1 & 5 & 15 \\ 16 & \times & 8 & 14 & 1 & 11 \\ 8 & 15 & \times & 9 & 5 & 11 \\ 18 & 3 & 3 & \times & 10 & 19 \\ 16 & 2 & 1 & 3 & \times & 14 \\ 3 & 5 & 19 & 8 & 13 & \times \end{array}$$

25. $$\begin{array}{cccccc} \times & 8 & 17 & 15 & 9 & 19 \\ 12 & \times & 1 & 9 & 10 & 10 \\ 9 & 15 & \times & 4 & 1 & 9 \\ 15 & 17 & 18 & \times & 13 & 15 \\ 5 & 18 & 16 & 15 & \times & 10 \\ 9 & 7 & 9 & 5 & 12 & \times \end{array}$$

26. $$\begin{array}{cccccc} \times & 16 & 2 & 1 & 1 & 8 \\ 1 & \times & 13 & 14 & 9 & 11 \\ 10 & 7 & \times & 10 & 4 & 7 \\ 19 & 8 & 4 & \times & 7 & 1 \\ 8 & 2 & 8 & 15 & \times & 18 \\ 13 & 3 & 8 & 13 & 11 & \times \end{array}$$

27. $$\begin{array}{cccccc} \times & 20 & 2 & 20 & 2 & 7 \\ 17 & \times & 4 & 12 & 2 & 13 \\ 10 & 15 & \times & 1 & 5 & 12 \\ 9 & 11 & 10 & \times & 10 & 1 \\ 19 & 16 & 11 & 19 & \times & 18 \\ 10 & 17 & 12 & 2 & 7 & \times \end{array}$$

28. $$\begin{array}{cccccc} \times & 9 & 16 & 9 & 1 & 17 \\ 9 & \times & 13 & 5 & 3 & 20 \\ 3 & 15 & \times & 19 & 10 & 7 \\ 13 & 19 & 10 & \times & 20 & 14 \\ 13 & 9 & 12 & 8 & \times & 14 \\ 1 & 12 & 9 & 20 & 3 & \times \end{array}$$

29. $$\begin{array}{cccccc} \times & 6 & 1 & 10 & 4 & 13 \\ 11 & \times & 20 & 9 & 1 & 1 \\ 3 & 2 & \times & 19 & 17 & 13 \\ 1 & 19 & 9 & \times & 7 & 12 \\ 16 & 14 & 2 & 16 & \times & 11 \\ 4 & 20 & 3 & 7 & 20 & \times \end{array}$$

30. $$\begin{array}{cccccc} \times & 20 & 13 & 16 & 3 & 20 \\ 10 & \times & 6 & 18 & 13 & 6 \\ 17 & 3 & \times & 12 & 4 & 11 \\ 2 & 8 & 7 & \times & 12 & 19 \\ 4 & 12 & 1 & 7 & \times & 18 \\ 9 & 10 & 9 & 15 & 10 & \times \end{array}$$

31. $$\begin{array}{cccccc} \times & 20 & 13 & 19 & 1 & 5 \\ 13 & \times & 6 & 7 & 14 & 15 \\ 13 & 10 & \times & 13 & 15 & 18 \\ 7 & 4 & 8 & \times & 4 & 4 \\ 16 & 9 & 3 & 7 & \times & 12 \\ 9 & 7 & 6 & 10 & 17 & \times \end{array}$$

32. $$\begin{array}{cccccc} \times & 13 & 8 & 6 & 16 & 3 \\ 6 & \times & 11 & 12 & 15 & 13 \\ 11 & 19 & \times & 10 & 6 & 11 \\ 15 & 14 & 6 & \times & 15 & 10 \\ 5 & 9 & 10 & 12 & \times & 16 \\ 5 & 8 & 15 & 16 & 12 & \times \end{array}$$

33. $$\begin{array}{cccccc} \times & 14 & 9 & 1 & 2 & 7 \\ 3 & \times & 7 & 3 & 2 & 13 \\ 13 & 5 & \times & 8 & 16 & 9 \\ 8 & 19 & 5 & \times & 20 & 6 \\ 7 & 16 & 6 & 1 & \times & 16 \\ 14 & 3 & 14 & 7 & 19 & \times \end{array}$$

34.

×	18	20	9	10	13
13	×	9	6	15	20
14	18	×	11	16	13
4	5	3	×	15	7
13	10	5	2	×	2
8	11	4	16	5	×

35.

×	11	17	14	1	19
16	×	2	17	2	18
17	20	×	2	14	2
1	1	10	×	2	20
1	14	15	15	×	13
20	1	19	3	20	×

36.

×	8	3	6	1	14
4	×	12	14	3	13
20	1	×	19	20	3
9	4	5	×	13	2
17	15	6	15	×	13
20	19	19	8	14	×

37.

×	10	12	4	2	11
3	×	18	12	14	11
6	1	×	3	10	15
1	2	14	×	10	3
9	15	20	11	×	3
6	4	7	8	20	×

38.

×	11	15	17	19	13
8	×	3	8	6	20
17	17	×	20	20	7
5	6	17	×	1	9
5	6	7	17	×	13
14	12	19	16	17	×

39.

×	19	10	12	2	20
8	×	17	16	1	10
9	12	×	16	1	14
6	20	6	×	2	2
16	13	11	1	×	13
9	10	10	10	11	×

40.

×	14	8	14	20	4
8	×	5	12	13	17
4	16	×	19	16	6
14	17	8	×	10	15
5	13	19	8	×	19
17	13	13	7	19	×

СПИСОК РЕКОМЕНДОВАННОЙ ЛИТЕРАТУРЫ

Ряд вопросов, рассмотренных в этом пособии, можно найти в учебниках по высшей математике, по математическому программированию и по исследованию операций.

1. Кузнецов А.В., Сакович В.А., Холод Н.И. Высшая математика. Математическое программирование. Минск: Вышэйш. шк., 1994.

2. Таха Х. Введение в исследование операций. Т 1, 2. М.: Мир, 1985.

3. Вагнер Г. Основы исследования операций. Т 1—3. М.: Мир, 1972.

4. Зайченко Ю.П. Исследование операций. Киев.: Выща шк., 1979.

ДЛЯ ЗАМЕТОК

ДЛЯ ЗАМЕТОК

Учебное издание

Копылов Георгий Николаевич, **Суханова** Наталья Николаевна

МАТЕМАТИЧЕСКИЕ МЕТОДЫ В ЭКОНОМИКЕ

Методическое пособие
(для студентов экономического факультета)

Главный редактор *А.В. Шестакова*
Технический редактор *М.Н. Растегина*
Корректор *О.В. Рукосуева*
Художник *Н.Н. Захарова*

ЛР № 020406 от 12.02.97

Подписано в печать 20.10. 2001 г. Формат 60×84/16.
Бумага типографская № 1. Гарнитура Таймс. Усл. печ. л. 6,28.
Уч.-изд. л. 6,75. Тираж 150 экз (1-й завод 70 экз.). Заказ . «С» 6.

Издательство Волгоградского государственного университета.
400062, Волгоград, ул. 2-я Продольная, 30.