

Министерство образования Республики Беларусь

Учреждение образования
«Полоцкий государственный университет»

А. В. Капусто, И. П. Кунцевич

ДИСКРЕТНАЯ МАТЕМАТИКА

Учебно-методический комплекс
для студентов машиностроительных специальностей

В двух частях

Часть 2

Новополоцк
ПГУ
2010

УДК 51(075.8)
ББК 22.11я73
К20

Рекомендовано к изданию методической комиссией
машиностроительного факультета в качестве учебно-методического комплекса
(протокол № 1 от 1.02.2010)

РЕЦЕНЗЕНТЫ:

профессор кафедры уравнений математической физики БГУ,
доктор педагогических наук, доцент О. И. МЕЛЬНИКОВ;
кандидат физико-математических наук, доцент, профессор
кафедры высшей математики УО «ПГУ» И. Е. АНДРУШКЕВИЧ

Капусто, А. В.

К20 Дискретная математика : учеб.-метод. комплекс для студентов машиностроительных специальностей. В 2 ч. Ч. 2 / А. В. Капусто, И. П. Кунцевич. – Новополоцк : ПГУ, 2010. – 248 с.
ISBN 978-985-531-039-7.

Во второй части изложен теоретический материал по разделам «Основы теории графов» и «Элементы математического моделирования».

Содержит большое число примеров с подробным решением; приведены задания для практических занятий и самостоятельного индивидуального выполнения.

Предназначен для студентов машиностроительных специальностей.

УДК 51(075.8)
ББК 22.11я73

ISBN 978-985-531-039-7 (ч. 2)
ISBN 978-985-531-035-9

© Капусто А.В., Кунцевич И.П., 2010
© УО «Полоцкий государственный университет», 2010

РАЗДЕЛ 3

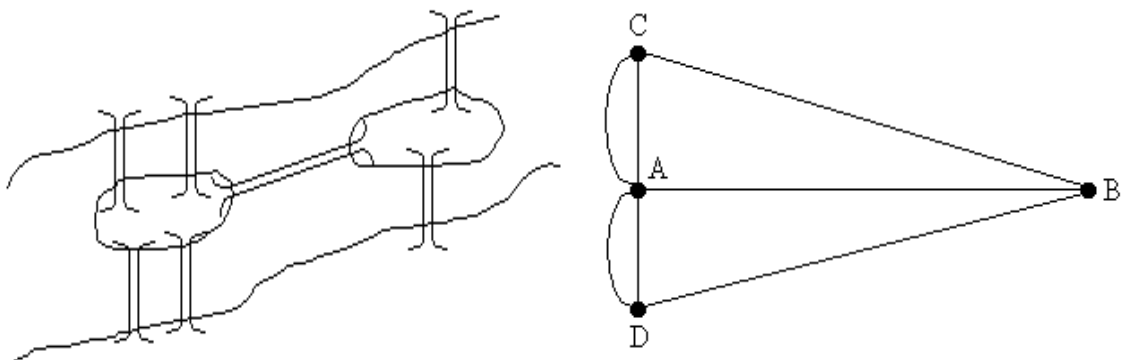
ОСНОВЫ ТЕОРИИ ГРАФОВ

Теория графов – это раздел дискретной математики, имеющий многочисленные приложения в различных областях экономики, социологии, техники, программирования. Это обусловлено тем, что стройная система специальных терминов и обозначений теории графов позволяет просто и доступно описывать различные понятия, иллюстрировать взаимодействия элементов в различных структурах. Еще одной важной особенностью является то, что графы в наглядной графической интерпретации передают изучаемый объект. Например, в теории множеств мы с помощью орграфа показывали бинарные отношения.

Теория графов многократно «открывалась» разными авторами при решении различных прикладных задач.

Исторически считается, что теория графов берет начало с решения задачи о кенигсбергских мостах в 1736 году знаменитым математиком Леонардом Эйлером (1707 – 1783, родился в Швейцарии, жил и работал в России).

Задача о кенигсбергских мостах



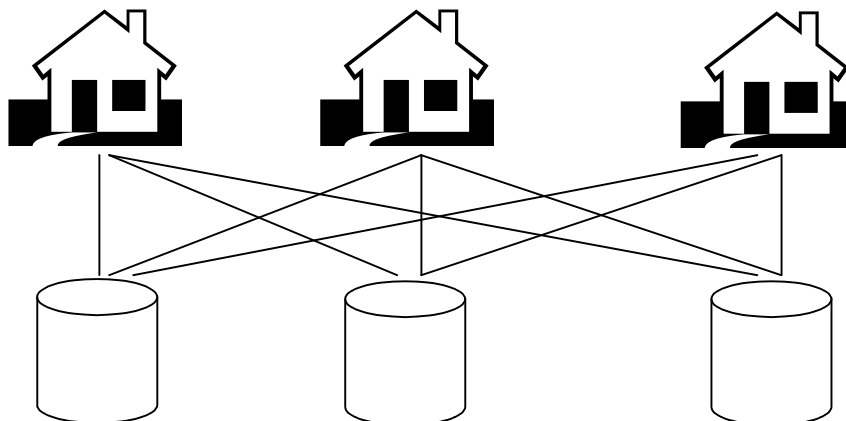
Жителям Кенигсберга нравилось гулять по дорожкам, которые включали семь мостов, соединяющих два острова и берега реки Преголя. Люди интересовались, могут ли они, начав путь с одного участка суши, обойти все мосты, посетив каждый лишь однажды, и вернуться в точку начала пути, не переплывая реку. Для решения задачи Эйлер построил граф:

участки суши изобразил вершинами, а дорожки через мосты – ребрами. Сначала им была сформулирована и доказана теорема. И опираясь на эту теорему, Эйлер показал, что такого маршрута нельзя построить.

Кроме задачи о кенигсбергских мостах, классическими задачами стали: *задача о трех домах и трех колодцах; задача о четырех красках.*

Задача о трех домах и трех колодцах

Имеется три дома и три колодца, каким-то образом расположенные на плоскости. Провести от каждого дома к каждому колодцу тропинку так, чтобы тропинки не пересекались. Эта задача была решена (показано, что решения не существует) К. Куратовским (1896 – 1979) в 1930 году.



Задача о четырех красках

Разбиение плоскости на непересекающиеся области называется *картой*. Области карты называются соседними, если они имеют общую границу. Задача состоит в раскрашивании карты таким образом, чтобы никакие две области не были закрашены одним цветом. С конца XIX века известна гипотеза, что для этого достаточно четырех красок. В 1976 году Appel и Хейкен опубликовали решение задачи о четырех красках, которое базировалось на переборе вариантов с помощью компьютера.

Суть опубликованного решения состоит в том, чтобы перебрать большое, но конечное число (около 2000) типов потенциальных контрпримеров к теореме о четырех красках и показать, что ни один случай контрпримером не является. Этот перебор был выполнен программой примерно

за тысячу часов работы суперкомпьютера. Проверить «вручную» полученное решение невозможно – объем перебора выходит за рамки человеческих возможностей. Тем не менее, многие математики ставят вопрос: можно ли считать такое «программное доказательство» действительным доказательством? Ведь в программе могут быть ошибки. Методы формального доказательства правильности программ не применимы к программам такой сложности, как обсуждаемая. Тестирование не может гарантировать отсутствие ошибок. Таким образом, остается уповать на программистскую квалификацию авторов и верить, что они все сделали правильно.

1. ГРАФЫ. ОРГРАФЫ

1.1. Графы и орграфы. Основные понятия

Определение 1.1. *Графом $G = G(V, E)$ называется совокупность двух множеств – непустого множества V (множества вершин) и множества E двухэлементных подмножеств множества V . Множество E называется множеством ребер.*

Вершины v_i и v_j множества V называются *соединенными* ребром (v_i, v_j) или *инцидентными* к ребру (v_i, v_j) , если $(v_i, v_j) \in E$. Если (v_i, v_j) – ребро, тогда вершины v_i и v_j называются *концами* ребра (v_i, v_j) . Ребро (v_i, v_j) называется также *инцидентным* к вершинам v_i и v_j . Две вершины называются *смежными*, если они инцидентны к одному ребру. Два ребра называются *смежными*, если они инцидентны к общей вершине.

Число вершин графа G обозначим n , а число ребер – m :

$$|V| = n, \quad |E| = m.$$

Определение 1.2. *Ориентированный граф или орграф $G = G(V, E)$ – это такой граф, который состоит из множества V вершин и множества E упорядоченных пар элементов из V . В этом случае элемент множества E называется *дугой*. Если $(v_i, v_j) \in E$, то v_i называется *начальной вершиной* дуги (v_i, v_j) , а v_j – *конечной вершиной*.*

Геометрическое представление графов следующее: вершина графа – точка в пространстве (на плоскости), ребро неориентированного графа – отрезок, дуга ориентированного графа – направленный отрезок.

Ребро, которое соединяет вершину саму с собой, называется *петлей*. Если в графе допускается наличие петель, то он называется *графом с петлями* или *псевдографом*.

Если в графе допускается наличие более одного ребра между двумя вершинами, то он называется *мультиграфом*. Например, граф, построенный для решения задачи о кенигсбергских мостах, является мультиграфом.

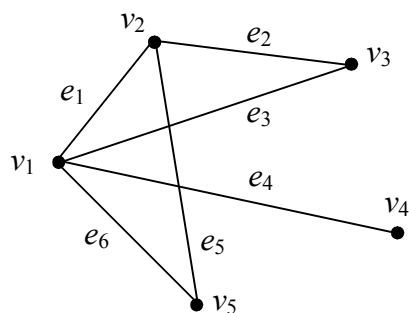
Если каждая вершина графа и (или) ребра помечена, то такой граф называется *помеченным* (или *нагруженным*). В качестве пометок обычно используются буквы или целые числа.

Если орграф содержит более чем одну дугу из одной вершины в другую, то называется *ориентированным мультиграфом*. Если каждая дуга помечена, то будем говорить, что это *помеченный орграф*.

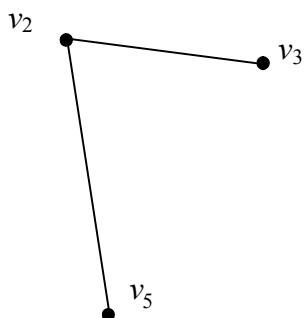
Определение 1.3. Граф $G'(V', E')$ называется *подграфом* (или *частью*) графа $G(V, E)$, если $V' \subseteq V$, $E' \subseteq E$. Если $V' = V$, то G' называется *остовным подграфом* G .

Аналогично, как и для графа, для орграфа вводится понятие ориентированный подграф.

Пример 1.1. Дан неориентированный граф.



$G(V, E)$ – граф
 $V = \{v_1, v_2, v_3, v_4, v_5\}$ – множество вершин
 $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$ – множество ребер
 Например, ребро $e_2 = (v_2, v_3)$: ребро e_2 инцидентно к вершинам v_2 и v_3 .
 Вершины v_2 и v_3 смежные



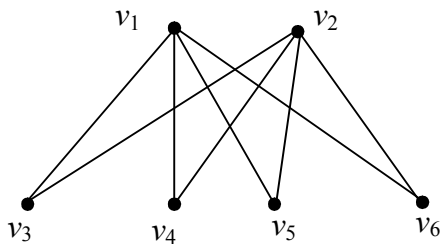
Подграф данного графа

□

Определение 1.4. Граф называется *полным*, если любые две его вершины соединены ребром. Полный граф с n вершинами обозначается через K_n .

Определение 1.5. Граф $G(V, E)$ называется *двудольным*, если V можно представить как объединение непересекающихся множеств $(V = A \cup B)$ так, что каждое ребро имеет вид (v_i, v_j) , где $v_i \in A$ и $v_j \in B$. Двудольный граф называется *полным двудольным графом* $K_{m,n}$, если A содержит m вершин, B содержит n вершин и для каждого $v_i \in A$, $v_j \in B$ имеем $(v_i, v_j) \in E$.

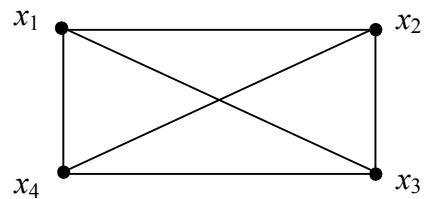
Пример 1.2. Построить полный двудольный граф $K_{2,4}$ и полный граф K_4 .



Полный двудольный граф $K_{2,4}$

$$A = \{v_1, v_2\}$$

$$B = \{v_3, v_4, v_5, v_6\}$$



Полный граф K_4

□

1.2. Связность графов

Пусть $G = G(V, E)$ – граф с вершинами $v_i \in V$ ($i = \overline{1, n}$) и ребрами $e_j \in E$ ($j = \overline{1, m}$).

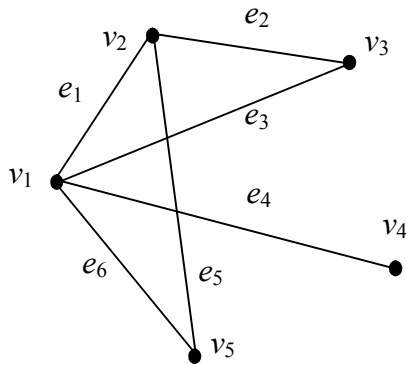
Определение 1.6. *Маршрутом (путем) длины k* из вершины v_1 в вершину v_{k+1} (или между v_1 и v_{k+1}) называется последовательность $v_1 e_1 v_2 e_2 v_3 \dots v_k e_k v_{k+1}$ такая, что $e_k = (v_k, v_{k+1})$.

Таким образом, путь длины k имеет k ребер. По причине избыточности обозначений в этом определении путь для графа в общем случае будет обозначаться как $v_1 v_2 v_3 \dots v_k v_{k+1}$.

Если все ребра различны, то путь называется *цепью*. Если все вершины различны (а значит, и ребра), то путь называется *простой цепью*. Замкнутая цепь называется *циклом*. Замкнутая простая цепь называется *простым циклом*. Граф без циклов называется *ациклическим*.

Аналогично, как и для графа, для орграфа вводятся понятия ориентированный путь, ориентированный цикл.

Пример 1.3. Дан неориентированный граф.



$G(V, E)$ – граф

$v_1 e_1 v_2 e_2 v_3 e_3 v_1 e_1 v_2 e_5 v_5$ или $v_1 v_2 v_3 v_1 v_2 v_5$ – маршрут

$v_1 v_2 v_3 v_1 v_5$ – цепь

$v_1 v_3 v_2 v_5$ – простая цепь

$v_1 v_3 v_2 v_5 v_1$ – простой цикл

□

Определение 1.7. Граф $G = G(V, E)$ называется *связным*, если имеется цепь между любыми двумя его различными вершинами.

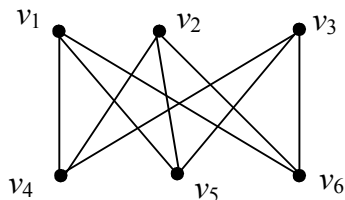
Для заданного ориентированного графа G можно построить неориентированный граф G^S такой, что каждая ориентированная дуга G (исключая петли) станет неориентированным ребром графа G^S . В таком случае граф $G^S(V, E^S)$ называется *соотнесенным графом* орграфа $G(V, E)$.

Определение 1.8. Орграф $G = G(V, E)$ называется *связным*, если его соотнесенный граф является связным. Орграф называется *сильно связным*, если для любой пары вершин $v_i, v_j \in V$ существует ориентированный путь из v_i в v_j .

1.3. Изоморфизм графов

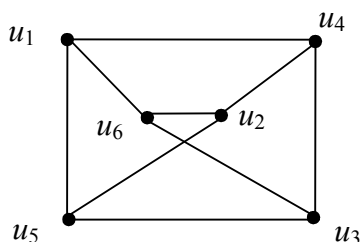
Определение 1.9. Функция $f : G(V, E) \rightarrow G_1(V_1, E_1)$ является *изоморфизмом* (обозначается $G \sim G_1$), если $f : V \rightarrow V_1$ и $f : E \rightarrow E_1$ представляют собой взаимно однозначные соответствия. Если $f : G \rightarrow G_1$ – изоморфизм, то G и G_1 называются *изоморфными*.

Пример 1.4. Дан неориентированный помеченный граф $G_1(V_1, E_1)$. Построить изоморфные ему графы.



$G_1(V_1, E_1)$ – граф

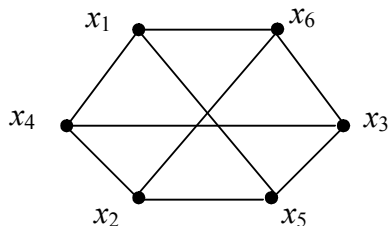
Решение.



$G_2(V_2, E_2)$ – граф

Изоморфизм $f : G_1 \rightarrow G_2$

$$\begin{aligned} f(v_1) &= u_1 & f(v_2) &= u_2 & f(v_3) &= u_3 \\ f(v_4) &= u_4 & f(v_5) &= u_5 & f(v_6) &= u_6 \\ f(v_1, v_2) &\rightarrow (u_1, u_2) \text{ и т.д.} \end{aligned}$$



$G_3(V_3, E_3)$ – граф

Изоморфизм $g : G_1 \rightarrow G_3$

$$\begin{aligned} g(v_1) &= x_1 & g(v_2) &= x_2 & g(v_3) &= x_3 \\ g(v_4) &= x_4 & g(v_5) &= x_5 & g(v_6) &= x_6 \\ g(v_1, v_2) &= (x_1, x_2) \text{ и т.д.} \end{aligned}$$

□

Теорема 1.1. *Изоморфизм графов есть отношение эквивалентности.*

Доказательство.

Действительно, изоморфизм обладает всеми необходимыми свойствами отношения эквивалентности:

– рефлексивность: $G \sim G$, где требуемая биекция есть тождественная функция;

– симметричность: если $G_1 \sim G_2$ с биекцией h , то $G_2 \sim G_1$ с биекцией h^{-1} ;

– транзитивность: если $G_1 \sim G_2$ с биекцией h и $G_2 \sim G_3$ с биекцией g , то $G_1 \sim G_3$ с биекцией $g \circ h$. □

1.4. Степень вершин

Определение 1.10. *Степенью* вершины v для неориентированного графа называется количество ребер, инцидентных этой вершине. Вершина степени 0 называется *изолированной*. Вершина степени 1 называется *висячей*. Обозначается степень вершины: $d(v)$.

Определение 1.11. *Полустепенью исхода* вершины v для орграфа называется количество дуг, для которых v является начальной вершиной, обозначается: $d^-(v)$. *Полустепенью захода* вершины v называется количество дуг, для которых v является конечной вершиной, обозначается: $d^+(v)$. Если $d^+(v) = 0$, то вершина v называется *истоком*. Если $d^-(v) = 0$, то вершина v называется *стоком*.

Теорема 1.2 (Эйлера). *Сумма степеней вершин графа равна удвоенному количеству ребер:*

$$\sum_{v \in V} d(v) = 2m.$$

Доказательство.

При подсчете суммы степеней вершин каждое ребро учитывается два раза: для одного конца ребра и для другого. \square

Следствие 1.1. *Число вершин нечетной степени четно.*

Доказательство.

По теореме Эйлера сумма степеней всех вершин – четное число. Сумма степеней вершин четной степени четна, значит, сумма степеней вершин нечетной степени также четна, следовательно, их четное число. \square

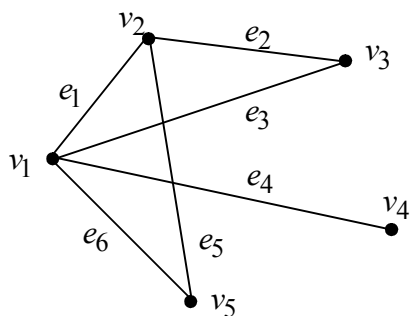
Следствие 1.2. *Сумма полустепеней вершин орграфа равна удвоенному количеству дуг:*

$$\sum_{v \in V} d^-(v) + \sum_{v \in V} d^+(v) = 2m.$$

Доказательство.

Сумма полустепеней вершин орграфа равна сумме степеней вершин графа, полученного из орграфа забыванием ориентации дуг. \square

Пример 1.5. Определить степени вершин данного графа.



$G(V, E)$ – граф

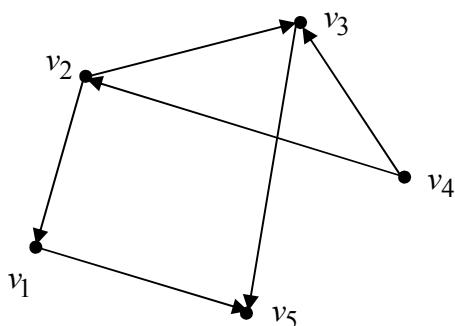
$$d(v_1) = 4 \quad d(v_2) = 3 \quad d(v_3) = 2$$

$$d(v_4) = 1 \quad d(v_5) = 2$$

вершина v_4 – висячая

□

Пример 1.6. Определить полустепени исхода и захода данного орграфа.



$$d^-(v_1) = 1 \quad d^-(v_2) = 2 \quad d^-(v_3) = 1$$

$$d^+(v_1) = 1 \quad d^+(v_2) = 1 \quad d^+(v_3) = 2$$

$$d^-(v_4) = 2 \quad d^-(v_5) = 0$$

$$d^+(v_4) = 0 \quad d^+(v_5) = 2$$

вершина v_4 – исток, вершина v_5 – сток

□

1.5. Представление графов в компьютере

Известны различные способы представления графов в памяти компьютера, которые различаются объемом занимаемой памяти и скоростью выполнения операций над графами. Представление выбирается исходя из потребностей конкретной задачи. В подавляющем большинстве случаев граф задается матрицей. Чаще всего графы представляют либо матрицей смежности, либо матрицей инцидентий.

Определение 1.12. Матрица смежности вершин орграфа G , содержащего n вершин, – это квадратная матрица $A_{n \times n} = [a_{ij}]_{n \times n}$, у которой строки и столбцы матрицы соответствуют вершинам орграфа. Элементы a_{ij} матрицы A равны числу дуг, направленных из i -той вершины в j -тую. Если орграф состоит из однократных дуг, то элементы матрицы равны либо 0, либо 1.

Матрицей смежности вершин неориентированного графа G , содержащего n вершин, называют квадратную матрицу $A_{n \times n} = [a_{ij}]_{n \times n}$, у которой строки и столбцы матрицы соответствуют вершинам неориентированного графа. Элементы a_{ij} матрицы A равны числу ребер, направленных из i -той вершины в j -тую. В случае неориентированного графа G ему вместе с ребром (v_i, v_j) принадлежит и ребро (v_j, v_i) , поэтому матрица смежности вершин $A_{n \times n} = [a_{ij}]_{n \times n}$ будет симметрична относительно главной диагонали.

Определение 1.13. *Матрица смежности дуг орграфа – это квадратная матрица $B_{m \times m} = [b_{ij}]_{m \times m}$, у которой строки и столбцы матрицы соответствуют дугам орграфа. Элементы b_{ij} матрицы B равны 1, если дуга e_i непосредственно предшествует дуге e_j , и 0 в остальных случаях.*

Матрицей смежности ребер неориентированного графа является матрица $B_{m \times m} = [b_{ij}]_{m \times m}$, у которой строки и столбцы матрицы соответствуют ребрам графа. Элементы b_{ij} матрицы B равны 1, если ребра e_i и e_j имеют общую вершину, и 0 в остальных случаях.

Определение 1.14. *Матрицей инциденций (инцидентности) неориентированного помеченного графа с n вершинами и m ребрами называется матрица $C_{n \times m} = [c_{ij}]_{n \times m}$, строки которой соответствуют вершинам, а столбцы – ребрам. Элементы c_{ij} матрицы инциденций неориентированного графа равны 1, если вершина v_i инцидентна ребру e_j , и 0 в противном случае.*

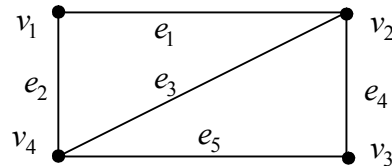
Матрицей инциденций (инцидентности) орграфа с n вершинами и m дугами называется матрица $C_{n \times m} = [c_{ij}]_{n \times m}$, строки которой соответствуют вершинам, а столбцы – дугам орграфа:

$$C_{n \times m} = [c_{ij}]_{n \times m} = \begin{cases} 1, & \text{если вершина } v_i \text{ инцидентна дуге } e_j \\ & \text{и является ее началом,} \\ 0, & \text{если вершина } v_i \text{ и дуга } e_j \text{ не инцидентны,} \\ -1, & \text{если вершина } v_i \text{ инцидентна дуге } e_j \\ & \text{и является ее концом.} \end{cases}$$

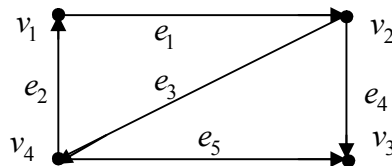
Если каждому ребру графа приписано некоторое положительное число, то такое число называется весом, а сам граф называется взвешенным графом. Простой взвешенный граф (сеть) может быть представлен также своей *матрицей весов* $\Omega = [\omega_{ij}]$, где ω_{ij} – вес ребра (дуги), соединяющего вершины v_i и v_j . Весы несуществующих ребер (дуг) полагают равными нулю или бесконечности в зависимости от приложений.

Пример 1.7.

1) Для заданного неориентированного графа построить матрицы смежностей и матрицу инциденций.



2) Для заданного ориентированного графа построить матрицы смежностей и матрицу инциденций.



Решение.

1) Строим матрицу смежности вершин, которая будет иметь размерность 4×4 . Строим матрицу смежности ребер, которая будет иметь размерность 5×5 .

$$A = \begin{pmatrix} & v_1 & v_2 & v_3 & v_4 \\ v_1 & 0 & 1 & 0 & 1 \\ v_2 & 1 & 0 & 1 & 1 \\ v_3 & 0 & 1 & 0 & 1 \\ v_4 & 1 & 1 & 1 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} & e_1 & e_2 & e_3 & e_4 & e_5 \\ e_1 & 0 & 1 & 1 & 1 & 0 \\ e_2 & 1 & 0 & 1 & 0 & 1 \\ e_3 & 1 & 1 & 0 & 1 & 1 \\ e_4 & 1 & 0 & 1 & 0 & 1 \\ e_5 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

Строим матрицу инцидентий, которая будет иметь размерность 4×5 .

$$C = \left(\begin{array}{c|ccccc} & e_1 & e_2 & e_3 & e_4 & e_5 \\ \hline v_1 & 1 & 1 & 0 & 0 & 0 \\ v_2 & 1 & 0 & 1 & 1 & 0 \\ v_3 & 0 & 0 & 0 & 1 & 1 \\ v_4 & 0 & 1 & 1 & 0 & 1 \end{array} \right).$$

2) Строим матрицу смежности вершин размерности 4×4 . Строим матрицу смежности ребер размерности 5×5 .

$$A = \left(\begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & 0 & 1 & 0 & 0 \\ v_2 & 0 & 0 & 1 & 1 \\ v_3 & 0 & 0 & 0 & 0 \\ v_4 & 1 & 0 & 1 & 0 \end{array} \right), \quad B = \left(\begin{array}{c|ccccc} & e_1 & e_2 & e_3 & e_4 & e_5 \\ \hline e_1 & 0 & 0 & 1 & 1 & 0 \\ e_2 & 1 & 0 & 0 & 0 & 0 \\ e_3 & 0 & 1 & 0 & 0 & 1 \\ e_4 & 0 & 0 & 0 & 0 & 0 \\ e_5 & 0 & 0 & 0 & 0 & 0 \end{array} \right).$$

Строим матрицу инцидентий размерности 4×5 .

$$C = \left(\begin{array}{c|ccccc} & e_1 & e_2 & e_3 & e_4 & e_5 \\ \hline v_1 & 1 & -1 & 0 & 0 & 0 \\ v_2 & -1 & 0 & 1 & 1 & 0 \\ v_3 & 0 & 0 & 0 & -1 & -1 \\ v_4 & 0 & 1 & -1 & 0 & 1 \end{array} \right).$$

□

1.6. Выявление маршрутов с заданным количеством ребер (дуг)

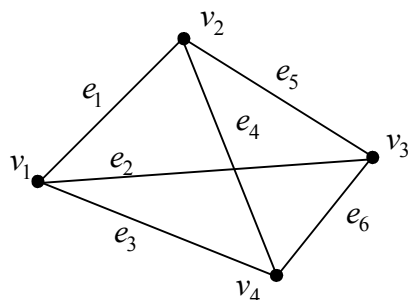
С помощью матрицы смежности вершин можно найти все маршруты, содержащие заданное количество ребер (дуг).

Теорема 1.3*. Для определения маршрутов, состоящих из k ребер (дуг), необходимо возвести в k -тую степень матрицу смежности вершин

А. Тогда элемент $a_{ij}^{(k)}$ матрицы A^k даст количество маршрутов длины k (состоящих из k ребер) из вершины v_i в вершину v_j .

Пример 1.8.

Для неориентированного графа, изображенного на рисунке, найти все маршруты длины 2.



Решение. Составим матрицу смежности вершин A и возведем ее в квадрат. Результат возведения:

$$A^2 = \begin{pmatrix} & v_1 & v_2 & v_3 & v_4 \\ v_1 & 0 & 1 & 1 & 1 \\ v_2 & 1 & 0 & 1 & 1 \\ v_3 & 1 & 1 & 0 & 1 \\ v_4 & 1 & 1 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} & v_1 & v_2 & v_3 & v_4 \\ v_1 & 0 & 1 & 1 & 1 \\ v_2 & 1 & 0 & 1 & 1 \\ v_3 & 1 & 1 & 0 & 1 \\ v_4 & 1 & 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} & v_1 & v_2 & v_3 & v_4 \\ v_1 & 3 & 2 & 2 & 2 \\ v_2 & 2 & 3 & 2 & 2 \\ v_3 & 2 & 2 & 3 & 2 \\ v_4 & 2 & 2 & 2 & 3 \end{pmatrix}.$$

Рассмотрим первую строку. Например, элемент $a_{11}^{(2)} = 3$. Это значит, что существует три маршрута из v_1 в v_1 длиной в два ребра. Действительно, это маршруты $v_1 e_1 v_2 e_1 v_1$, $v_1 e_2 v_3 e_2 v_1$, $v_1 e_3 v_4 e_3 v_1$. Из v_1 в v_2 существует два маршрута: $v_1 e_2 v_3 e_5 v_2$, $v_1 e_3 v_4 e_4 v_2$.

Если использовать числовую матрицу смежности вершин, то для нахождения самих маршрутов необходимо работать с графом. Если воспользоваться модифицированной матрицей смежности, в ячейки которой записаны названия ребер, то можно получить не только количество маршрутов, но и сами маршруты.

Действительно, для данного примера имеем

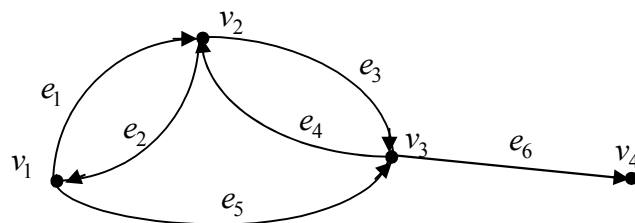
$$A^2 = \begin{pmatrix} & v_1 & v_2 & v_3 & v_4 \\ v_1 & 0 & e_1 & e_2 & e_3 \\ v_2 & e_1 & 0 & e_5 & e_4 \\ v_3 & e_2 & e_5 & 0 & e_6 \\ v_4 & e_3 & e_4 & e_6 & 0 \end{pmatrix} \cdot \begin{pmatrix} & v_1 & v_2 & v_3 & v_4 \\ v_1 & 0 & e_1 & e_2 & e_3 \\ v_2 & e_1 & 0 & e_5 & e_4 \\ v_3 & e_2 & e_5 & 0 & e_6 \\ v_4 & e_3 & e_4 & e_6 & 0 \end{pmatrix} =$$

$$= \begin{pmatrix} & v_1 & v_2 & v_3 & v_4 \\ v_1 & e_1e_1 + e_2e_2 + e_3e_3 & e_2e_5 + e_3e_4 & e_1e_5 + e_3e_6 & e_1e_4 + e_2e_6 \\ v_2 & e_5e_2 + e_4e_3 & e_1e_1 + e_5e_5 + e_4e_4 & e_1e_2 + e_4e_6 & e_1e_3 + e_5e_6 \\ v_3 & e_5e_1 + e_6e_3 & e_2e_1 + e_6e_4 & e_2e_2 + e_5e_5 + e_6e_6 & e_2e_3 + e_5e_4 \\ v_4 & e_4e_1 + e_6e_2 & e_3e_1 + e_6e_5 & e_3e_2 + e_4e_5 & e_3e_3 + e_4e_4 + e_6e_6 \end{pmatrix}$$

□

Пример 1.9.

Для орграфа, изображенного на рисунке, найти все маршруты с тремя дугами.



Решение. Матрица смежности A и результаты ее возведения в квадрат и куб имеют следующий вид:

$$A^2 = \begin{pmatrix} & v_1 & v_2 & v_3 & v_4 \\ v_1 & 0 & 1 & 1 & 0 \\ v_2 & 1 & 0 & 1 & 0 \\ v_3 & 0 & 1 & 0 & 1 \\ v_4 & 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} & v_1 & v_2 & v_3 & v_4 \\ v_1 & 0 & 1 & 1 & 0 \\ v_2 & 1 & 0 & 1 & 0 \\ v_3 & 0 & 1 & 0 & 1 \\ v_4 & 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} & v_1 & v_2 & v_3 & v_4 \\ v_1 & 1 & 1 & 1 & 1 \\ v_2 & 0 & 2 & 1 & 1 \\ v_3 & 1 & 0 & 1 & 0 \\ v_4 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$A^3 = \begin{pmatrix} & v_1 & v_2 & v_3 & v_4 \\ v_1 & 1 & 1 & 1 & 1 \\ v_2 & 0 & 2 & 1 & 1 \\ v_3 & 1 & 0 & 1 & 0 \\ v_4 & 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} & v_1 & v_2 & v_3 & v_3 \\ v_1 & 0 & 1 & 1 & 0 \\ v_2 & 1 & 0 & 1 & 0 \\ v_3 & 0 & 1 & 0 & 1 \\ v_4 & 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} & v_1 & v_2 & v_3 & v_4 \\ v_1 & 1 & 2 & 2 & 1 \\ v_2 & 2 & 1 & 2 & 1 \\ v_3 & 0 & 2 & 1 & 1 \\ v_4 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Рассмотрим, например, после возведения матрицы смежности вершин в квадрат элемент $a_{22}^{(2)} = 2$. Это значит, из вершины v_2 в вершину v_2 есть два маршрута длиной в две дуги. Это маршруты $v_2e_3v_3e_4v_2$ и $v_2e_2v_1e_1v_2$. После возведения матрицы в куб рассмотрим, например, $a_{12}^{(3)} = 2$. Это значит, что есть два маршрута длиной в три дуги из вершины v_1 в вершину v_2 . Это маршруты $v_1e_1v_2e_2v_1e_1v_2$ и $v_1e_1v_2e_3v_3e_4v_2$. \square

Для получения цепей (маршрутов, в которых каждое ребро встречается один раз) нужно в модифицированной матрице A^3 вычеркнуть те слагаемые, в которых какой-либо сомножитель встречается более одного раза.

1.7. Упорядочение вершин и дуг орграфа

Расчеты в задачах, связанных с графами, заметно упрощаются, если их элементы упорядочены. Под *упорядочением вершин* связного орграфа без циклов понимают такое разбиение его вершин на группы, при котором:

- 1) вершины первой группы не имеют предшествующих вершин, а вершины последней группы последующих;
- 2) вершины любой другой группы не имеют предшествующих в следующей группе;
- 3) вершины одной и той же группы дугами не соединяются.

Аналогичным образом вводится понятие *упорядочения дуг*. В результате упорядочения элементов получают орграф, изоморфный исходному. Упорядочение элементов выполняется графическим или матричным способом.

Графический способ упорядочения вершин, дуг орграфа носит название *алгоритма Фалкерсона*.

Алгоритм Фалкерсона для упорядочения вершин

1. Находят вершины графа, в которые не входит ни одна дуга. Они образуют первую группу. Нумеруют вершины группы в натуральном порядке: 1, 2, При этом присвоение номеров вершинам внутри группы может быть сделано не единственным образом, что не имеет значения.

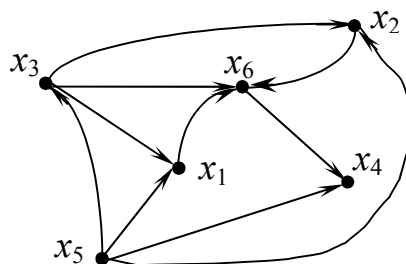
2. Мысленно вычеркиваем все пронумерованные вершины и дуги, из них выходящие. В получившемся графе найдется, по крайней мере, одна вершина, в которую не входит ни одна дуга. Этой вершине, входящей во вторую группу, присваивается очередной номер и т.д. Этот шаг повторяется до тех пор, пока все вершины не будут упорядочены (пронумерованы).

Алгоритм Фалкерсона для упорядочения дуг

1. Найти дуги, не имеющие непосредственно предшествующих (они образуют I группу).

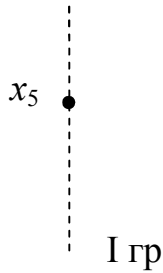
2. Вычеркнуть найденные дуги; после этого появится, по крайней мере, одна новая дуга, не имеющая непосредственно предшествующей (в графе без дуг I группы). Такие дуги составляют II группу. Повторять этот шаг, пока все дуги не будут разбиты на группы. В заключение упорядочения дугам присваивают новые обозначения с индексами 1, 2,

Пример 1.10. Графическим способом упорядочить вершины заданного орграфа.



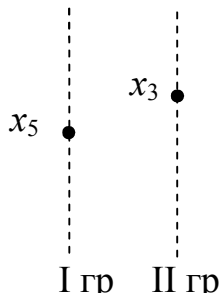
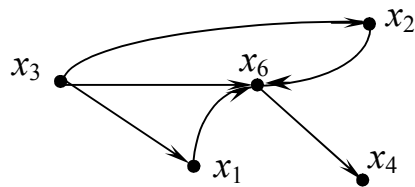
Решение. Используя алгоритм Фалкерсона, упорядочим вершины заданного орграфа.

Сначала находим вершины орграфа, из которых дуги только выходят. В нашем случае – это вершина x_5 .



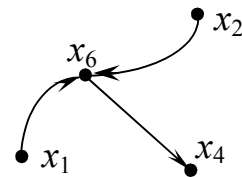
Вершина x_5 войдет в первую группу.
 В заданном орграфе мысленно вычеркнем выбранную вершину и дуги, выходящие из этой вершины. В результате получим следующий орграф.

В новом орграфе опять найдем вершины, из которых дуги только выходят. Этой вершиной будет вершина x_3 , которая войдет во вторую группу.

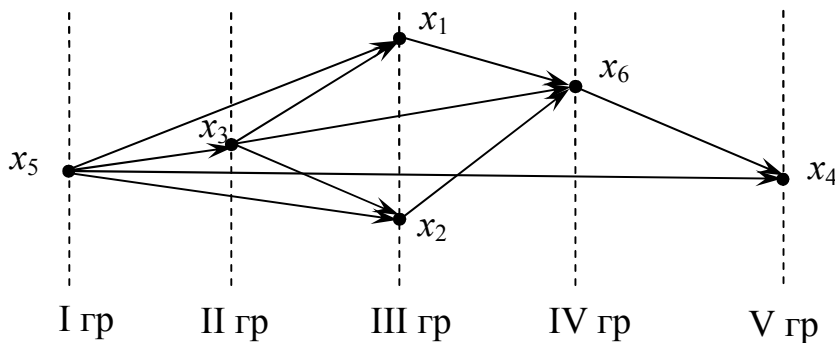


В получившемся орграфе мысленно вычеркнем вершину x_3 и дуги, выходящие из этой вершины. В результате получим следующий орграф.

В получившемся орграфе найдем вершины, из которых дуги выходят, но ни одна дуга не заходит. Этими вершинами будут вершины x_1 и x_2 , которые войдут в третью группу.



Повторив этапы алгоритма Фалкерсона еще два раза, получим следующее: вершина x_6 образует четвертую группу и вершина x_4 – пятую группу. Соединив вершины соответствующими дугами, получаем орграф, изоморфный исходному.



□

2. ДЕРЕВЬЯ. ОСТОВНЫЕ ДЕРЕВЬЯ

2.1. Деревья

Существует один простой и важный тип графов, которому разные авторы дали одинаковое название – *дерево*. Для них имеют место многие утверждения, которые не всегда выполняются для графов в общем случае. Деревья являются самым распространенным классом графов, применяемых в программировании, причем в самых разных ситуациях.

Определение 2.1. *Деревом* называется граф $T(V, E)$ без циклов. *Лес* – это граф, компоненты которого являются деревьями.

Ориентированное дерево T' представляет собой свободный от петель ориентированный граф, соотнесенный граф которого является деревом.

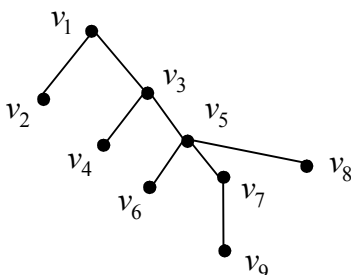
Если дерево имеет хотя бы одно ребро, оно имеет хотя бы две вершины со степенью 1. Вершины степени 1 называются *листьями*. Другие вершины называются *внутренними вершинами*.

Если дерево изображено таким образом, что в верхней части помещена вершина, а остальные находятся ниже его, то вершина в самой верхней части называется *корнем* дерева. Если корень дерева определен, то дерево называется *корневым деревом*.

Если корень выбран, *уровень вершины* v определяется длиной единственной цепи из корня в вершину v .

Определение 2.2. *Высотой* дерева называется длина самой длинной цепи от корня до листа, обозначается: $h(T)$.

Пример 2.1.



v_1 – корень дерева

v_2, v_4, v_6, v_8, v_9 – листья

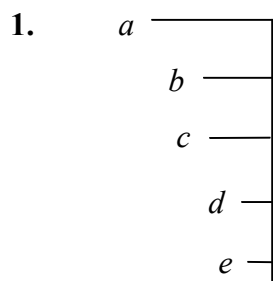
v_3, v_5, v_7 – внутренние вершины

высота дерева $h(T) = 4$

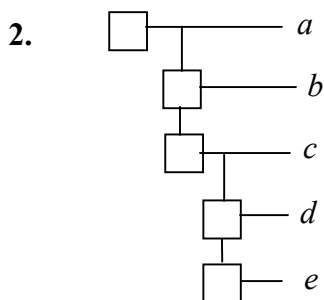
□

Пример 2.2.

Примеры изображения деревьев в программировании.



Для представления иерархической структуры вложенности элементов данных и/или операторов управления используется техника отступов, показанная на рисунке.



Структура вложенности каталогов и файлов в современных операционных системах является упорядоченным ориентированным деревом. Обычно для изображения таких деревьев применяется способ, показанный на рисунке.

□

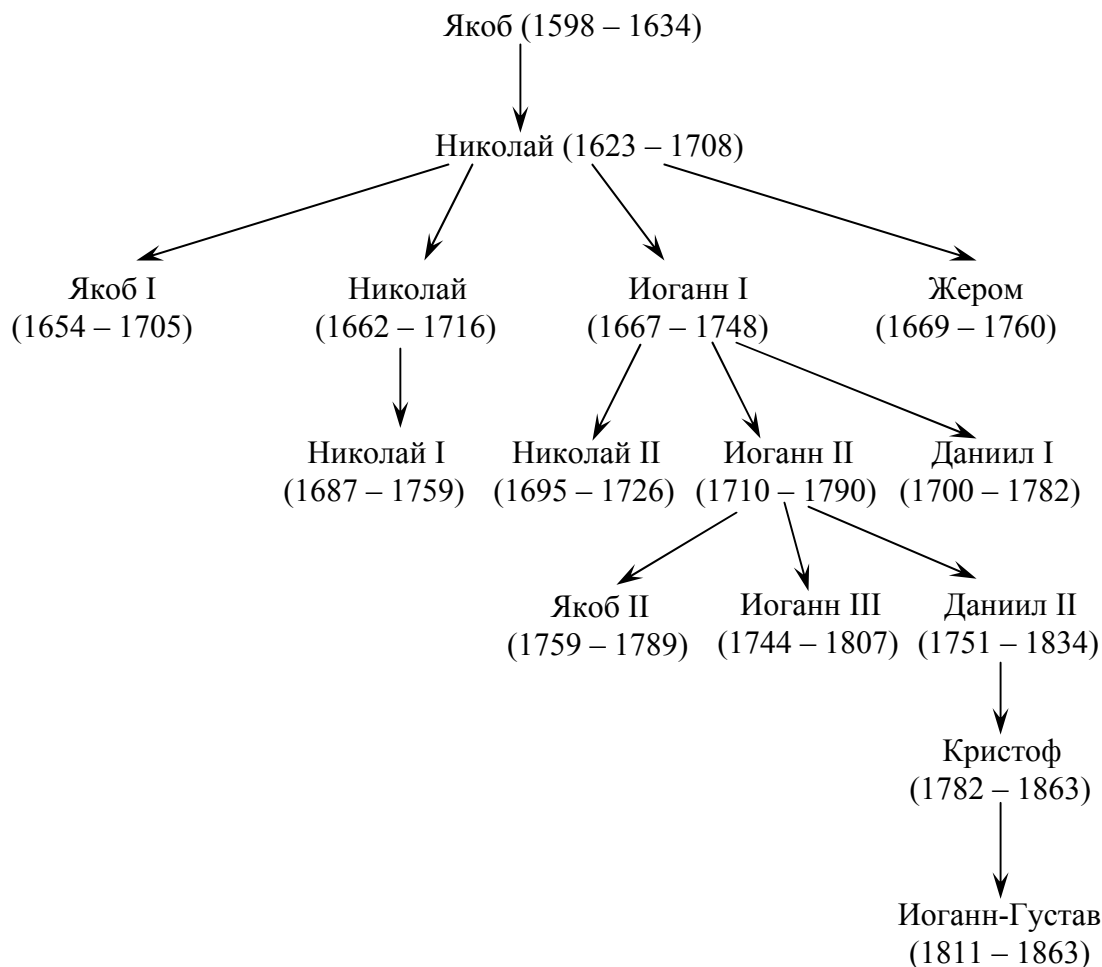
Тот факт, что большинство систем управления файлами использует ориентированные деревья, отражается даже в терминологии – «*корневой каталог диска*».

Пример 2.3 (генеалогическое дерево семейства Бернулли)

Купеческая протестантская семья Бернулли жила в Антверпене (Нидерланды). Свой род она вела из Фландрии, где Бернулли в XV в. носили еще фамилию Бернуйла (Bernuilla). В связи с событиями XVI в. на территории Нидерландов, которые были направлены против протестантов, семья Бернулли вынуждена была уехать во Франкфурт-на-Майне. В это время главой семьи был Якоб Бернулли, который умер в 1583 г.

Среди Бернулли некоторые имена повторялись из поколения в поколение, поэтому их различают, как королей, присоединив к имени соответ-

ствующую цифру. В виде дерева родословная семейства Бернулли имеет следующий вид:



Якоб I (1654 – 1705) – швейцарский математик, по образованию – богослов, с 1687 г. – профессор математики Базельского университета. Ему принадлежат важные заслуги в развитии анализа бесконечно малых, начало которому положила работа Лейбница, опубликованная в 1684 г. Якоб I вычислил площади многих плоских фигур, площади поверхностей и длины линий. Ряд работ относится к алгебре, арифметике, геометрии, теории рядов, теории вероятностей, а также – к физике. В книге «Искусство предположений» Якоб I доказал теорему (названную позже его именем), имеющую важное значение в теории вероятностей и ее приложениях к статистике. Его учениками были младший брат Иоганн I; племянник Николай I; член Петербургской академии наук, механик и математик Я. Герман; отец великого Л. Эйлера – Пауль Эйлер.

Иоганн I (1667 – 1748), брат Якоба I, десятый ребенок в семье Николая. По образованию Иоганн I – врач, с 1695 г. – профессор математики Гронингенского университета (Голландия), с 1705 г. – профессор матема-

тики Базельского университета. Иоганн I являлся почетным членом Петербургской академии наук.

Николай I (1687 – 1759) – швейцарский математик, сын Николая, по образованию – юрист. Николай I был профессором математики в Падуе, а затем – профессором логики и права в Базельском университете. Его исследования посвящены теории вероятностей, интегральному исчислению, демографии.

Николай II (1695 – 1726), сын Иоганна I, по образованию – юрист. Профессор права в Берне, профессор математики в Петербурге.

Даниил I (1700 – 1782), уроженец Гронингена, сын Иоганна I, по образованию – врач. В 1725 – 1733 гг. работал на кафедрах физиологии и механики в Петербургской академии наук, с 1733 г. – профессор кафедры физиологии, с 1750 г. – профессор кафедры механики в Базеле. Почетный член Петербургской академии наук. В математике Даниилу Бернулли принадлежат: метод численного решения алгебраических уравнений с помощью возвратных рядов, работы по обыкновенным дифференциальным уравнениям, по теории вероятностей с приложением к статистике народонаселения и отчасти к астрономии, по теории рядов. В работах, завершенных написанным в Петербурге трудом «Гидродинамика» (1738), вывел основное уравнение стационарного движения идеальной жидкости, носящее его имя. Даниил I разрабатывал кинематические представления о газах.

Иоганн II (1710 – 1790), сын Иоганна I, по образованию – юрист. Профессор элоквенции (красноречия), профессор математики в Базеле.

Иоганн III (1744 – 1807), старший сын Иоганна II. По образованию – юрист. Астроном Берлинской академии наук, там же – директор математического класса.

Даниил II (1751 – 1834), второй сын Иоганна II, по образованию – врач, профессор красноречия в Базеле.

Якоб II (1759 – 1789), третий сын Иоганна II, по образованию – юрист. Математик Петербургской академии наук.

Кристоф (1782 – 1863), сын Даниила II. Профессор технологии в Базеле.

Иоганн-Густав (1811 – 1863), сын Кристофа. Профессор технологии в Базеле.

Представители рода Бернулли живут в Базеле и в настоящее время.

□

Рассмотрим некоторые утверждения, которые относятся к деревьям.

Теорема 2.1. Для любых вершин a и b дерева T существует единственная цепь из a в b .

Доказательство

Доказательство проведем методом «от противного». Предположим, что для некоторых вершин a и b дерева T цепь из a в b не является единственной, и покажем, что в таком случае T не будет деревом.

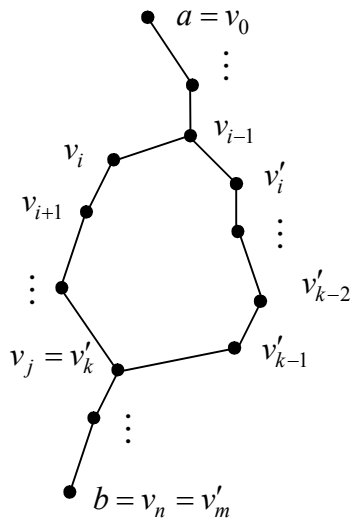


Рис. 2.1

Допустим, что существует две различные цепи: $v_0v_1v_2\dots v_n$ длины n и $v_0v'_1v'_2\dots v'_m$ длины m , где $a = v_0$ и $b = v_n = v'_m$, рис. 2.1. В каждой цепи должна существовать первая вершина, начиная с которой соответствующие вершины не совпадают, скажем $v_i \neq v'_i$, и в каждом из цепей должна существовать точка, начиная с которой вершины опять одни и те же, скажем $v_j = v'_k$. Тогда $v_{i-1}v_iv_{i+1}v_{i+2}\dots v_jv'_kv'_{k-1}v'_{k-2}\dots v'_iv_{i-1}$ является циклом, а значит, граф T не является деревом.

Поэтому наше предположение неверно. То есть для любых вершин a и b дерева T существует единственная цепь из a в b . \square

Верна также и обратная теорема.

Теорема 2.2*. Если для любых двух вершин графа G существует единственная цепь из вершины a в вершину b , тогда G является деревом.

Теорема 2.3. Если у дерева T имеется t ребер и n вершин, то $n = t + 1$.

Доказательство.

Предположим, что имеется дерево T . Любое дерево можно представить как корневое дерево, и это никаким образом не меняет ни числа ребер, ни числа вершин.

Рассмотрим теперь ориентированное дерево T' , порожденное деревом T . У каждой дуги ориентированного дерева одна и только одна конечная вершина. Следовательно, число дуг и вершин одно и то же, исключая корневую. Если учесть корневую вершину, получим, что вершин на одну больше, чем дуг.

Значит, и исходное дерево имеет число вершин на одну больше, чем число ребер. \square

Справедлива также и обратная теорема.

Теорема 2.4*. *Если в связном графе G , содержащем m ребер и n вершин, выполнено равенство $n = m + 1$, то граф G является деревом.*

2.2. Остовные деревья

Определение 2.3. *Дерево T называется *остовным деревом* графа G , если T – подграф графа G и каждая вершина графа G является вершиной в T .*

Теорема 2.5*. *У каждого связного графа существует подграф, который является остовным деревом.*

Для построения остовных деревьев существуют разные методы. Самыми распространенными являются два метода: метод поиска в ширину и метод поиска в глубину.

Согласно методу поиска остовного дерева в ширину произвольную вершину v_0 графа G выбираем в качестве корня дерева T . Для каждой вершины v графа G , смежной с вершиной v_0 , в дерево T добавляется вершина v и ребро (v_0, v) . Это вершины первого уровня. Затем берем каждую вершину v_i первого уровня и для каждой вершины v_j графа G , смежной с вершиной v_i из тех, что еще не выбраны, добавляем в дерево T вершину v_j и ребро (v_i, v_j) . Вершины, добавленные на этом этапе, – это вершины второго уровня. Продолжаем процесс, пока в графе G не останется вершин, которые можно было бы добавить в дерево. По построению граф T является деревом. Если расстояние от v_0 до вершины v графа G равно n , то эта вершина будет добавлена в дерево на уровне n . Следовательно, T является остовным деревом.

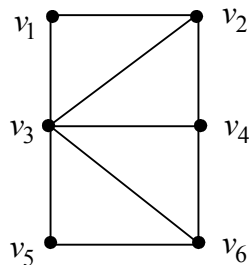
При поиске в ширину, первым делом, отыскиваются все вершины, смежные с заданной вершиной, а потом осуществляется переход на следующий уровень.

При поиске в глубину усилия направлены на построение для дерева как можно более длинного пути.

Метод поиска в глубину начинается с задания вершины графа, которую будем считать корнем. Выбираем вершину v_i , смежную с корнем, и формируем ребро дерева. Затем выбираем вершину v_j , смежную с ранее выбранной вершиной v_i , и формируем новое ребро. По ходу необходимо помечать использованные вершины с той целью, чтобы каждая вершина использовалась один раз. Если, находясь в вершине v , мы выбираем другую вершину w и обнаруживаем, что вершина w уже была добавлена в дерево, то ребро (v, w) между этими вершинами не может быть добавлено в дерево.

Пример 2.4.

Задан граф. Требуется:

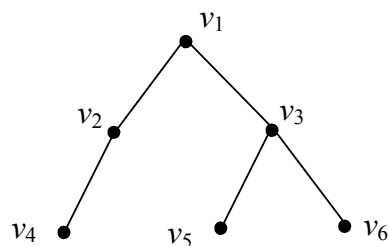


- 1) найти остовное дерево с помощью метода поиска в ширину;
- 2) найти остовное дерево с помощью метода поиска в глубину.

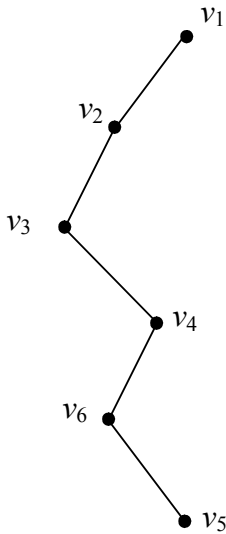
Решение.

1) Выберем в качестве первой вершины, например, вершину v_1 , которая будет корнем остовного дерева. С вершиной v_1 смежными являются две вершины v_2 и v_3 . Поэтому в строящееся дерево добавляем два ребра (v_1, v_2) и (v_1, v_3) . Вершины v_2 и v_3 будут вершинами первого уровня.

Далее находим вершины, смежные с вершинами v_2 и v_3 . С вершиной v_2 смежная вершина v_4 , а с вершиной v_3 – вершины v_5 и v_6 . Добавляем в дерево ребра (v_2, v_4) , (v_3, v_5) и (v_3, v_6) . Вершины v_4 , v_5 и v_6 будут вершинами второго уровня. В результате получаем следующее корневое остовное дерево, которое изображено на рисунке.



2) При использовании метода поиска в глубину, как и в предыдущем методе в качестве первой вершины выберем, например, вершину v_1 . Эта вершина будет корнем остоного дерева.



С вершиной v_1 смежными являются две вершины v_2 и v_3 . Берем, например, вершину v_2 и добавляем в строящееся дерево ребро (v_1, v_2) . С вершиной v_2 смежными являются две вершины v_3 и v_4 . Берем, например, вершину v_3 и добавляем в строящееся дерево ребро (v_2, v_3) . С вершиной v_3 смежными являются три вершины: v_4 , v_5 и v_6 . Берем, например, вершину v_4 и добавляем в строящееся дерево ребро (v_3, v_4) . С вершиной v_4 смежной является одна вершина v_6 , которая еще не вошла в остоное дерево. Добавляем в строящееся дерево ребро (v_4, v_6) . И, наконец, добавляем ребро (v_6, v_5) .

В результате получаем корневое остоное дерево, которое изображено на рисунке.

□

Имея граф, хотелось бы знать ответ на вопрос: сколько можно построить остовных деревьев для помеченного графа с n вершинами?

Теорема 2.6 (Теорема Кэли)*. Число остовных деревьев для K_n , у которого вершины помечены, равно $t_n = n^{n-2}$.

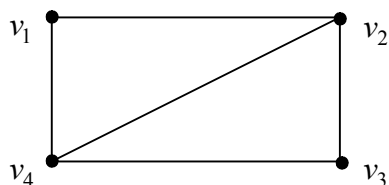
Если рассматривать определенный граф, который отличается от полного, то эта формула не годится для определения числа остовных графов. В этом случае используют теорему Кирхгофа. Прежде введем понятие матрицы Кирхгофа.

Определение 2.4. Матрицей Кирхгофа связного графа G с помеченными вершинами называется матрица $K_{n \times n} = K(G)$, $|V| = n$, которую можно определить следующим образом:

$$K_{n \times n} = K(G) = [k_{ij}] = \begin{cases} -1, & \text{если } v_i \text{ и } v_j \text{ смежны,} \\ 0, & \text{если } v_i \text{ и } v_j \text{ не смежны и } i \neq j, \\ d(v_i), & \text{если } i = j. \end{cases}$$

Теорема 2.7 (Теорема Кирхгофа)*. Число остовных деревьев в связном графе G порядка $n \geq 2$ равно алгебраическому дополнению любого элемента матрицы Кирхгофа $K(G)$.

Пример 2.5. Найти число остовных деревьев данного графа.



Решение.

Сначала составим для данного графа матрицу Кирхгофа.

$$K(G) = \begin{pmatrix} & v_1 & v_2 & v_3 & v_4 \\ v_1 & 2 & -1 & 0 & -1 \\ v_2 & -1 & 3 & -1 & -1 \\ v_3 & 0 & -1 & 2 & -1 \\ v_4 & -1 & -1 & -1 & 3 \end{pmatrix}.$$

Далее найдем алгебраические дополнения. Например, для элемента k_{11} матрицы Кирхгофа получим

$$A_{11} = (-1)^{1+1} \begin{vmatrix} 3 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 3 \end{vmatrix} = 18 - 1 - 1 - 2 - 3 - 3 = 8,$$

для элемента k_{23}

$$A_{23} = (-1)^{2+3} \begin{vmatrix} 2 & -1 & -1 \\ 0 & -1 & -1 \\ -1 & -1 & 3 \end{vmatrix} = -1 \cdot (-6 - 1 + 0 + 1 - 0 - 2) = 8.$$

Таким образом, данный граф имеет 8 остовных деревьев. \square

2.3. Алгоритм Краскала и алгоритм Прима

Определение 2.5. Вес остовного дерева взвешенного графа G равен сумме весов, приписанных ребрам остовного дерева. Обозначение: $\omega(T)$.

Минимальным остовным деревом называется такое остовное дерево графа G , что вес дерева T меньше или равен весу любого другого остовного дерева графа G . Вес минимального остовного дерева будем обозначать $\omega_{\min}(T)$.

Построение остова графа G , имеющего наименьший вес, нашло широкое применение при решении некоторого класса задач прикладного характера. Приведем постановку одной из таких задач.

Пусть, например, граф $G = G(V, E, \Omega)$ служит моделью железнодорожной сети, соединяющей пункты $v_1, v_2, \dots, v_n \in V$, а $\omega(v_i, v_j)$ – расстояние между пунктами v_i и v_j . Требуется проложить сеть телеграфных линий вдоль железнодорожной сети так, чтобы все пункты v_1, v_2, \dots, v_n были связаны между собой телеграфной сетью, протяженность которой была бы наименьшей.

Рассмотрим два способа построения минимального остовного дерева взвешенного графа: алгоритм Краскала и алгоритм Прима.

Идея метода Краскала состоит в том, чтобы формировать дерево $T(V, E)$, выбирая ребра с наименьшим весом так, чтобы не возникал цикл.

Алгоритм Краскала

1. Выбрать в графе G ребро t минимального веса, не принадлежащее множеству E и такое, чтобы его добавление в множество E не создавало в дереве T цикл.
2. Добавить это ребро в множество ребер E .
3. Продолжать действия первого и второго шагов до тех пор, пока имеются ребра, обладающие указанными свойствами.

Принципиальное отличие алгоритма Прима состоит в том, что всегда имеется дерево, к которому ребра добавляют до тех пор, пока не получится остовное дерево.

Алгоритм Прима

1. Выбрать вершину v_1 графа G и ребро с наименьшим весом e_1 , для которого v_1 – одна из вершин, и сформировать дерево T_1 .

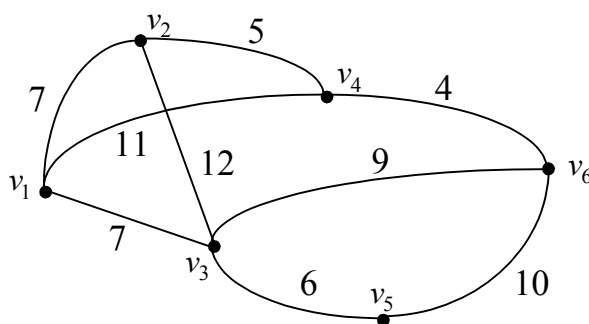
2. Для заданного дерева T_k с ребрами $e_1, e_2, e_3, \dots, e_k$ при наличии вершины, не принадлежащей T_k , следует выбрать ребро с наименьшим весом, смежное с ребром дерева T_k и имеющее вершину вне дерева T_k . Добавить новую вершину в дерево T_k , формируя дерево T_{k+1} .

3. Продолжать, пока имеются вершины графа G , не принадлежащие дереву.

Построение минимального остовного дерева можно проводить двумя способами: 1) остов графа строится непосредственно на самом графе, выделяя ребра утолщенной линией, которые входят в остовное дерево; 2) отдельно строится корневое дерево, которое будет минимальным остовным деревом. Второй случай используется, если требуется определить высоту построенного дерева.

Пример 2.6. Для данного взвешенного графа:

- 1) определить степень каждой вершины;
- 2) построить матрицу смежности вершин, матрицу смежности ребер, матрицу инцидентий;
- 3) найти число остовных деревьев, используя теорему Кирхгофа;
- 4) найти минимальное корневое остовное дерево, используя алгоритм Краскала или алгоритм Прима.



Решение.

- 1) Используя определение степени вершины графа, находим:

$$d(v_1) = 3, \quad d(v_2) = 3, \quad d(v_3) = 4, \quad d(v_4) = 3, \quad d(v_5) = 2, \quad d(v_6) = 3.$$

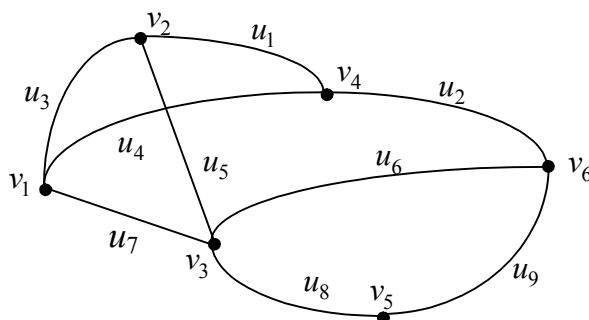
2) Построим матрицу смежности вершин. Данный граф имеет шесть вершин, поэтому матрица смежности вершин A является квадратной матрицей шестого порядка. Для заполнения первой строки мат-

рицы A найдем число ребер, соединяющих первую вершину с каждой из остальных вершин.

Поскольку количество ребер, соединяющих первую вершину с первой равно нулю (в графе нет петель, соответствующих первой вершине), количество ребер, соединяющих первую со второй, равно одному, первую с третьей – одному, первую с четвертой – одному, первую с пятой – нулю, первую с шестой – нулю, то первая строка матрицы A имеет вид $(0 \ 1 \ 1 \ 1 \ 0 \ 0)$. Заполняя аналогично остальные строки, получим матрицу A вида

$$A = \begin{pmatrix} & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \\ v_1 & 0 & 1 & 1 & 1 & 0 & 0 \\ v_2 & 1 & 0 & 1 & 1 & 0 & 0 \\ v_3 & 1 & 1 & 0 & 0 & 1 & 1 \\ v_4 & 1 & 1 & 0 & 0 & 0 & 1 \\ v_5 & 0 & 0 & 1 & 0 & 0 & 1 \\ v_6 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

Для построения матрицы смежности ребер и матрицы инцидентий рассмотрим невзвешенный граф, предварительно занумеровав ребра графа произвольным образом.



Так как граф имеет девять ребер, то матрица смежности ребер B является квадратной матрицей девятого порядка.

Ребро u_1 имеет общую вершину с ребрами u_3 и u_5 , а также общую вершину с ребрами u_2 и u_4 . Поэтому первая строка матрицы B имеет вид $(0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0)$. Заполняя аналогично остальные строки, получим матрицу B вида

$$B = \left(\begin{array}{c|ccccccccc} & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 & u_9 \\ \hline u_1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ u_2 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ u_3 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ u_4 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ u_5 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ u_6 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ u_7 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ u_8 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ u_9 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{array} \right).$$

Граф имеет шесть вершин и девять ребер, поэтому матрица инцидентий C имеет размерность 6×9 . Поскольку вершина v_1 инцидентна ребрам u_3 , u_4 и u_7 и не инцидентна остальным ребрам, то первая строка матрицы C имеет вид $(0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0)$. Остальные строки матрицы заполняются аналогично. Следовательно, матрица C будет иметь вид

$$C = \left(\begin{array}{c|ccccccccc} & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 & u_9 \\ \hline v_1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ v_2 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ v_3 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ v_4 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ v_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ v_6 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{array} \right).$$

3) Для того чтобы найти число остовных деревьев, сначала составим матрицу Кирхгофа.

$$K(G) = \left(\begin{array}{c|cccccc} & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \\ \hline v_1 & 3 & -1 & -1 & -1 & 0 & 0 \\ v_2 & -1 & 3 & -1 & -1 & 0 & 0 \\ v_3 & -1 & -1 & 4 & 0 & -1 & -1 \\ v_4 & -1 & -1 & 0 & 3 & 0 & -1 \\ v_5 & 0 & 0 & -1 & 0 & 2 & -1 \\ v_6 & 0 & 0 & -1 & -1 & -1 & 3 \end{array} \right).$$

Используя теорему Кирхгофа, найдем, например, алгебраическое дополнение элемента k_{11} :

$$A_{11} = (-1)^{1+1} \begin{vmatrix} 3 & -1 & -1 & 0 & 0 \\ -1 & 4 & 0 & -1 & -1 \\ -1 & 0 & 3 & 0 & -1 \\ 0 & -1 & 0 & 2 & -1 \\ 0 & -1 & -1 & -1 & 3 \end{vmatrix} = \begin{vmatrix} \text{третью строку умножим на } (-1) \\ \text{и прибавим ко второй,} \\ \text{третью строку умножим на } 3 \\ \text{и прибавим к первой} \end{vmatrix} =$$

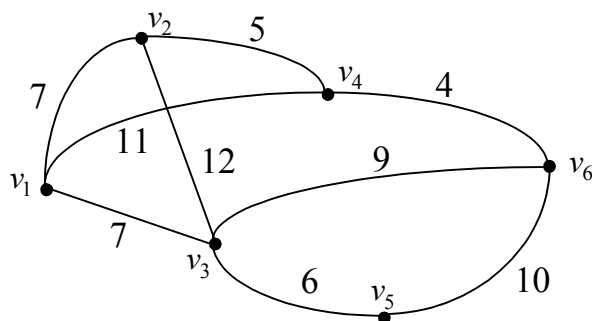
$$= \begin{vmatrix} 0 & -1 & 8 & 0 & -3 \\ 0 & 4 & -3 & -1 & 0 \\ -1 & 0 & 3 & 0 & -1 \\ 0 & -1 & 0 & 2 & -1 \\ 0 & -1 & -1 & -1 & 3 \end{vmatrix} = (-1) \cdot (-1)^{3+1} \begin{vmatrix} -1 & 8 & 0 & -3 \\ 4 & -3 & -1 & 0 \\ -1 & 0 & 2 & -1 \\ -1 & -1 & \boxed{-1} & 3 \end{vmatrix} =$$

$$= (-1) \cdot \begin{vmatrix} -1 & 8 & 0 & -3 \\ 5 & -2 & 0 & -3 \\ -3 & -2 & 0 & 5 \\ -1 & -1 & -1 & 3 \end{vmatrix} =$$

$$= -1 \cdot (-1) \cdot (-1)^{4+3} \cdot \begin{vmatrix} -1 & 8 & -3 \\ 5 & -2 & -3 \\ -3 & -2 & 5 \end{vmatrix} = -1 \cdot (10 + 72 + 30 + 18 + 6 - 200) = 64.$$

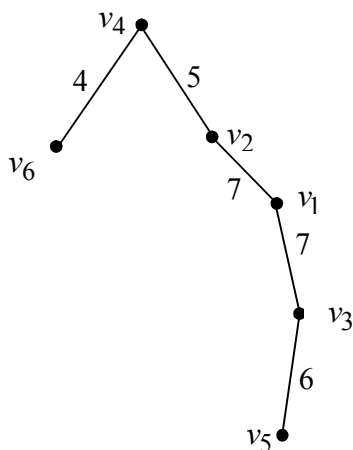
Таким образом, для данного графа можно построить 64 остовных дерева.

4) Для построения минимального остовного (корневого) дерева рассмотрим взвешенный граф, данный в условии задачи.



I способ. Используем для построения остовного дерева алгоритм Краскала. В графе выберем ребро с минимальным весом. В нашем случае это ребро, соединяющее вершины v_4 и v_6 с весом, равным 4. Пусть, например, вершина v_4 будет корнем дерева.

Далее выбираем ребра, инцидентные вершинам v_4 , v_6 и имеющие минимальный вес. Таким будет ребро с весом 5, соединяющее вершины v_4 и v_2 . Включим его в строящееся дерево. Затем к вершине v_2 присоединим ребро с весом 7, соединяющее вершины v_2 и v_1 . К вершине v_1 присоединим ребро с весом 7, соединяющее вершины v_1 и v_3 . И, в заключение, к вершине v_3 присоединим ребро с весом 6, соединяющее вершины v_3 и v_5 . Таким образом, получаем минимальное остовное дерево. Минимальный вес построенного дерева $\omega_{\min}(T) = 4 + 5 + 7 + 7 + 6 = 29$.

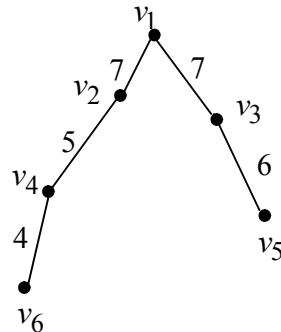


Так как мы в качестве корня дерева выбрали вершину v_4 , то высота дерева будет равна $h(T) = 4$.

Заметим, что если в качестве корня выбрать вершину v_6 , то высота дерева будет равна $h(T) = 5$.

II способ. Рассмотрим построение минимального корневого остовного дерева данного взвешенного графа с помощью алгоритма Прима. Выберем вершину v_1 , которая будет корнем дерева. Из трех ребер, которые инцидентны вершине v_1 , выберем те, что имеют наименьший вес. Два ребра с весом, равным 7, инцидентны вершине v_1 . Присоединим эти ребра к выбранной вершине. К вершине v_1 присоединим ребро с весом 5, соединяющее вершины v_2 и v_4 . К вершине v_4 присоединим ребро с весом 4, соединяющее вершины v_4 и v_6 . К вершине v_3 присоединим ребро с весом 6, со-

единяющее вершины v_3 и v_5 . Таким образом, получаем следующее минимальное корневое дерево с весом, равным $\omega_{\min}(T) = 7 + 7 + 6 + 5 + 4 = 29$. Заметим, что высота построенного дерева равна $h(T) = 3$.



Ответ:

- 1) $d(v_1) = 3, d(v_2) = 3, d(v_3) = 4, d(v_4) = 3, d(v_5) = 2, d(v_6) = 3$;
- 3) число остовных деревьев равно 64;
- 4) минимальный вес дерева равен $\omega_{\min}(T) = 29$.

3. НАХОЖДЕНИЕ КРАТЧАЙШИХ ПУТЕЙ. АЛГОРИТМ ДЕЙКСТРЫ

Остановимся на вопросе перемещения из одной вершины в другую с позиции «наилучшего способа» достижения оптимального значения определенного критерия. Детализируем это требование: это может быть как самый дешевый путь по стоимости, так и самый кратчайший путь по протяженности, как самый безопасный путь, так и наименее энергопотребляющий, и т.д.

Для решения такого класса задач рассматриваются взвешенные графы (орграфы), ребрам (дугам) которых приписан некоторый вес. Общий подход к решению задачи о кратчайшем пути был развит американским математиком Ричардом Беллманом (1920 – 1984), который предложил для этого вида задач название *динамическое программирование*. Задача о кратчайшем пути – это частный случай задачи, которую можно сформулировать следующим образом: *найти в заданном графе пути, соединяющие две заданные вершины и доставляющие минимум или максимум некоторой аддитивной функции, определенной как критерий эффективности множества путей*. Чаще всего эта функция трактуется как длина пути, и задача называется *задачей о кратчайшем пути*.

Алгоритм Дейкстры (Едсгер Дейкстра, нидерландский математик, 1930 – 2002) является одной из реализаций этой задачи. Данный алгоритм часто еще называют алгоритмом расстановки меток.

Пусть $G = G(V, E, \Omega)$ – ориентированный граф с взвешенными дугами. Обозначим через s (s -вершина) – вершину начала пути и через t (t -вершина) – вершину конца пути. В процессе работы алгоритма Дейкстры вершинам орграфа $x_i \in V$ приписываются числа (метки) $d(x_i)$, которые служат оценкой длины (веса) кратчайшего пути от s -вершины к вершине x_i . Если вершина x_i получила на некотором шаге метку $d(x_i)$, это означает, что в графе G существует путь из s -вершины в x_i , имеющий вес $d(x_i)$. Метки могут находиться в двух состояниях – быть временными или постоянными. Превращение временной метки в постоянную означает, что кратчайшее расстояние от вершины s до соответствующей вершины найдено. Алгоритм Дейкстры содержит одно ограничение – веса дуг должны быть положительными. Сам алгоритм состоит из двух этапов. На первом этапе находится длина кратчайшего пути, на втором – строится сам путь от вершины s к вершине t .

Алгоритм Дейкстры

Этап 1. Нахождение длины кратчайшего пути.

Шаг 1. *Присвоение вершинам начальных меток.*

Полагаем $d(s) = 0^*$ и считаем эту метку постоянной (постоянные метки помечаем сверху звездочкой). Для остальных вершин $x_i \in V$, $x_i \neq s$ полагаем $d(x_i) = \infty$ и считаем эти метки временными. Обозначим через \tilde{x} текущую вершину и примем первоначально $\tilde{x} = s$.

Шаг 2. *Изменение меток.*

Для каждой вершины x_i с временной меткой, непосредственно следующей за вершиной \tilde{x} , меняем ее метку в соответствии со следующим правилом:

$$d_{\text{нов}}(x_i) = \min \{ d_{\text{стар}}(x_i); d(\tilde{x}) + \omega(\tilde{x}; x_i) \}.$$

Шаг 3. *Превращение временной метки в постоянную.*

Из всех вершин с временными метками выбираем вершину x_j^* с наименьшим значением метки

$$d(x_j^*) = \min \{d(x_j) \mid x_j \in V\}, \text{ где } d(x_j) - \text{временная метка.}$$

Превращаем эту метку в постоянную и полагаем $\tilde{x} = x_j^*$.

Шаг 4. *Проверка на завершение первого этапа.*

Если $\tilde{x} = t$, то $d(\tilde{x})$ – длина кратчайшего пути от вершины s до t . В противном случае происходит возвращение ко второму шагу.

Этап 2. Построение кратчайшего пути.

Шаг 5. *Последовательный поиск дуг кратчайшего пути.*

Среди вершин, непосредственно предшествующих вершине \tilde{x} с постоянными метками, находим вершину x_i , удовлетворяющую соотношению

$$d(\tilde{x}) = d(x_i) + \omega(x_i; \tilde{x}).$$

Включаем дугу $(x_i; \tilde{x})$ в искомый путь и полагаем $\tilde{x} = x_i$.

Шаг 6. *Проверка на завершение второго этапа.*

Если $\tilde{x} = s$, то кратчайший путь найден – его образует последовательность дуг, полученных на пятом шаге и выстроенных в обратном порядке. В противном случае возвращаемся к пятому шагу.

Пример 3.1. По заданной матрице весов

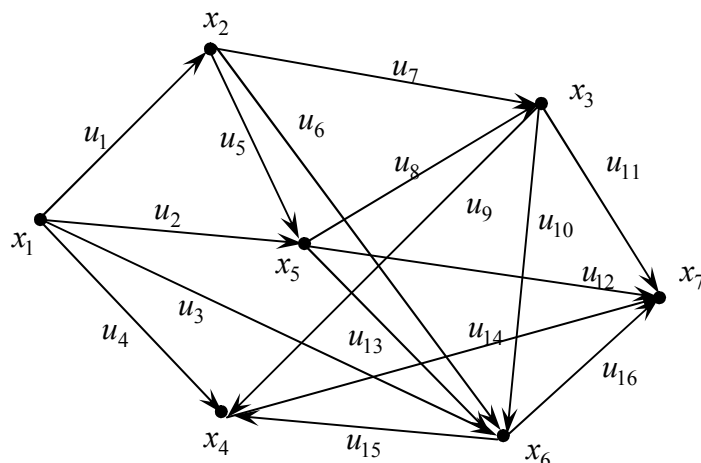
$$\Omega = [\omega_{ij}] = \begin{pmatrix} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ x_1 & - & 5 & \infty & 10 & 8 & 12 & \infty \\ x_2 & \infty & - & 7 & \infty & 4 & 9 & \infty \\ x_3 & \infty & \infty & - & 5 & \infty & 6 & 11 \\ x_4 & \infty & \infty & \infty & - & \infty & \infty & 14 \\ x_5 & \infty & \infty & 6 & \infty & - & 13 & 21 \\ x_6 & \infty & \infty & \infty & 8 & \infty & - & 9 \\ x_7 & \infty & \infty & \infty & \infty & \infty & \infty & - \end{pmatrix}$$

- 1) построить ориентированный граф;
- 2) найти полустепень захода и полустепень исхода каждой вершины;
- 3) построить матрицу смежности вершин, матрицу смежности дуг, матрицу инциденций;

4) найти величину минимального пути и сам путь от вершины $s = x_1$ до вершины $t = x_7$ с помощью алгоритма Дейкстры.

Решение

1) По данной матрице весов изобразим оргграф G , который не будет являться взвешенным.



2) Найдем полустепень исхода и полустепень захода для каждой вершины:

$$d^-(x_1) = 4, \quad d^+(x_1) = 0; \quad d^-(x_2) = 3, \quad d^+(x_2) = 1;$$

$$d^-(x_3) = 3, \quad d^+(x_3) = 2; \quad d^-(x_4) = 1, \quad d^+(x_4) = 3;$$

$$d^-(x_5) = 3, \quad d^+(x_5) = 2; \quad d^-(x_6) = 2, \quad d^+(x_6) = 4;$$

$$d^-(x_7) = 0, \quad d^+(x_7) = 4.$$

Так как вершина x_1 имеет $d^+(x_1) = 0$, то эта вершина является источником. Так как вершина x_7 имеет $d^-(x_7) = 0$, то эта вершина является стоком.

3) Построим матрицу смежности вершин. Данный оргграф имеет семь вершин, поэтому матрица смежности вершин A является квадратной матрицей седьмого порядка. Для заполнения первой строки матрицы A найдем число дуг, соединяющих первую вершину с каждой из остальных вершин.

Поскольку количество дуг, исходящих из первой вершины в первую равно нулю (в орграфе нет петель, соответствующих первой вершине), исходящих из первой во вторую, – единице, из первой в третью – нулю, из

первой в четвертую – единице, из первой в пятую – единице, из первой в шестую – единице, из первой в седьмую – нулю, то первая строка матрицы A имеет вид $(0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0)$. Заполнив аналогично остальные строки, получим матрицу A :

$$A = \left(\begin{array}{c|ccccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ \hline x_1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ x_2 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ x_3 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ x_4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ x_5 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ x_6 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ x_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right).$$

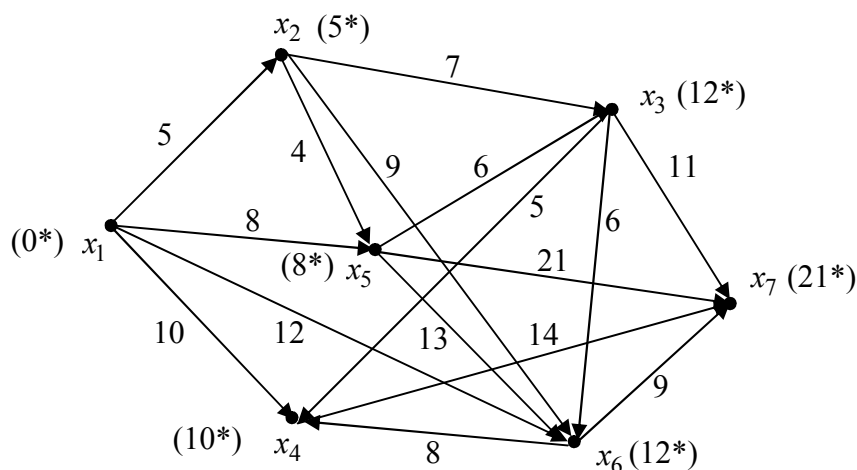
Так как орграф имеет шестнадцать дуг, то матрица смежности дуг B является квадратной матрицей шестнадцатого порядка. Дуга u_1 предшествует дугам u_5 , u_6 и u_7 . Поэтому первая строка матрицы B имеет вид $(0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0)$. Заполняем аналогично остальные строки, получаем матрицу B вида

$$B = \left(\begin{array}{c|cccccccccccccccc} & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 & u_9 & u_{10} & u_{11} & u_{12} & u_{13} & u_{14} & u_{15} & u_{16} \\ \hline u_1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ u_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ u_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ u_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ u_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ u_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ u_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ u_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ u_9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ u_{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ u_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ u_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ u_{13} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ u_{14} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ u_{15} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ u_{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right)$$

Орграф имеет семь вершин и шестнадцать дуг, поэтому матрица инциденций C имеет размерность 7×16 . Поскольку вершина x_1 инцидентна дугам u_1, u_2, u_3 и u_4 , которые выходят из вершины x_1 , и не инцидентна остальным дугам, то первая строка матрицы C имеет вид $(1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0)$. Остальные строки матрицы заполняем аналогично, учитывая то, в каких случаях вершина x_i будет началом, а в каких случаях – концом дуги u_j . Следовательно, матрица C имеет вид

$$C = \begin{pmatrix} & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 & u_9 & u_{10} & u_{11} & u_{12} & u_{13} & u_{14} & u_{15} & u_{16} \\ x_1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x_2 & -1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x_3 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ x_4 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ x_5 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ x_6 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 1 & 1 \\ x_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & -1 & 0 & -1 \end{pmatrix}.$$

4) Чтобы найти величину минимального пути и сам путь от вершины $s = x_1$ до вершины $t = x_7$ с помощью алгоритма Дейкстры, построим орграф, у которого к каждой дуге припишем ее вес.



Этап 1. Шаг 1. Полагаем $d(x_1) = 0^*$,

$$d(x_2) = d(x_3) = d(x_4) = d(x_5) = d(x_6) = d(x_7) = \infty.$$

Шаг 2. За вершиной $\tilde{x} = x_1$ следуют вершины, которые образуют множество $\tilde{s} = \{x_2, x_4, x_5, x_6\}$.

Пересчитаем временные метки:

$$d(x_2) = \min\{\infty; 0^* + 5\} = 5, \quad d(x_4) = \min\{\infty; 0^* + 10\} = 10,$$

$$d(x_5) = \min\{\infty; 0^* + 8\} = 8, \quad d(x_6) = \min\{\infty; 0^* + 12\} = 12.$$

Получаем $d(x_2) < d(x_5) < d(x_4) < d(x_6)$. Следовательно, вершине x_2 присваивается постоянная метка: $d(x_2) = 5^*$, $\tilde{x} = x_2$.

Шаг 3. $\tilde{s} = \{x_4, x_5, x_6, x_3\}$. Пересчитаем временные метки:

$$d(x_4) = \min\{\infty; 0^* + 10\} = 10, \quad d(x_5) = \min\{\infty; \underline{0^* + 8}; 5^* + 4\} = 8,$$

$$d(x_6) = \min\{\infty; \underline{0^* + 12}; 5^* + 9\} = 12, \quad d(x_3) = \min\{\infty; 5^* + 7\} = 12.$$

Получаем $d(x_5) < d(x_4) < d(x_3) = d(x_6)$. Следовательно, вершине x_5 присваивается постоянная метка: $d(x_5) = 8^*$, $\tilde{x} = x_5$.

Шаг 4. $\tilde{s} = \{x_4, x_3, x_6, x_7\}$. Пересчитаем временные метки:

$$d(x_4) = \min\{\infty; 0^* + 10\} = 10, \quad d(x_3) = \min\{\infty; \underline{5^* + 7}; 8^* + 6\} = 12,$$

$$d(x_6) = \min\{\infty; \underline{0^* + 12}; 5^* + 9; 8^* + 13\} = 12, \quad d(x_7) = \min\{\infty; 8^* + 21\} = 29.$$

Получаем $d(x_4) < d(x_3) < d(x_6) = d(x_7)$. Следовательно, вершине x_4 присваивается постоянная метка: $d(x_4) = 10^*$, $\tilde{x} = x_4$.

Шаг 5. $\tilde{s} = \{x_3, x_6, x_7\}$. Пересчитаем временные метки:

$$d(x_3) = \min\{\infty; \underline{5^* + 7}; 8^* + 6\} = 12,$$

$$d(x_6) = \min\{\infty; \underline{0^* + 12}, 5^* + 9; 8^* + 13\} = 12,$$

$$d(x_7) = \min\{8^* + 21; \underline{10^* + 14}\} = 24.$$

Получаем $d(x_3) = d(x_6) < d(x_7)$. Следовательно, вершинам x_3 и x_6 присваиваются постоянные метки: $d(x_3) = 12^*$, $d(x_6) = 12^*$, $\tilde{x} = x_3$, $\tilde{x} = x_6$.

Шаг 6. $d(x_7) = \min\{\infty; 12^* + 11; 10^* + 14; 8^* + 21; \underline{12^* + 9}\} = 21$. Вершине x_7 присваивается постоянная метка $d(x_7) = 21^*$, $\tilde{x} = x_7$.

Этап 2.

Проводим последовательный поиск дуг кратчайшего пути.

Вершине $\tilde{x} = x_7$ предшествуют вершины $\tilde{s} = \{x_3, x_4, x_5, x_6\}$. Кратчайшее расстояние получим при прохождении по дуге $(x_6; x_7)$.

Вершине $\tilde{x} = x_6$ предшествуют вершины $\tilde{s} = \{x_1, x_2, x_3, x_5\}$. Кратчайшее расстояние получим при прохождении по дуге $(x_1; x_6)$.

Таким образом, кратчайший путь от вершины x_1 до вершины x_7 построен. Его длина (вес) составляет 21, т.е. $d_{\min} = 21$, сам путь задает следующую последовательность дуг: $(x_1; x_6) - (x_6; x_7)$.

Ответ: $d_{\min} = 21$, $(x_1; x_6) - (x_6; x_7)$.

4. ЭЙЛЕРОВЫ И ГАМИЛЬТОНОВЫ ЦИКЛЫ

4.1. Пути и циклы Эйлера

Выше были введены понятия «путь» и «цикл» для связного графа. Особый интерес представляют два цикла, имеющие практическое применение – эйлеров цикл и гамильтонов цикл.

Пусть $G = G(V, E)$ – граф.

Определение 4.1. Цикл, который включает все ребра и вершины графа G , называется *эйлеровым циклом*. Граф, в котором существует эйлеров цикл, называется *эйлеровым*.

Эйлеров цикл содержит не только все ребра (каждое ребро включается в цикл только один раз), но и все вершины графа (каждая вершина может быть включена в цикл несколько раз). С эйлеровым циклом непосредственно связана задача о кенигсбергских мостах. Заметим, что эйлеровым может быть только связный граф, однако не в каждом случае.

Теорема 4.1 (Эйлера). *Связный граф является эйлеровым тогда и только тогда, когда степень каждой его вершины четная.*

Доказательство

Необходимость. Пусть G – связный эйлеров граф. Эйлеров цикл этого графа, проходя через каждую его вершину, входит в нее по одному ребру, а выходит по другому. Это означает, что каждое прохождение вершины по циклу добавляет в ее степень слагаемое 2. Поскольку цикл содержит все ребра графа, то степени всех вершин будут четными.

Достаточность. Предположим, что степени всех вершин связного графа G четные. Начнем цепь P_1 из произвольной вершины v_1 и будем продлевать ее, выбирая каждый раз новое ребро. Так как степень каждой из вершин четная, то, попав в некоторую вершину, мы всегда будем иметь в распоряжении еще не пройденное ребро. Таким образом, построение цепи P_1 обязательно закончится в вершине v_1 , и P_1 будет циклом. Если P_1 содержит все ребра графа G , то построен эйлеров цикл.

В противном случае, удалив из графа G ребра цепи P_1 , получим граф G_2 . Так как степень всех вершин графов G и цепи (графа) P_1 были четными, то и граф G_2 будет обладать этим свойством. В силу связности G графы P_1 и G_2 должны иметь хотя бы одну общую вершину v_2 . Теперь, начиная из v_2 , построим в графе G цикл P_2 подобно тому, как построили P_1 .

Объединим циклы P_1 и P_2 следующим образом: пройдем часть P_1 от вершины v_1 до вершины v_2 , затем пройдем цикл P_2 , затем – оставшуюся часть P_1 от v_2 до v_1 , рис. 4.1.

Если объединенный цикл не эйлеров, проделав аналогичные построения, получим еще больший цикл. Поскольку степени вершин во всех графах, составленных из ребер, не попавших в строящийся цикл, четные, и число ребер в этих графах убывает, то процесс закончится построением эйлерова цикла.

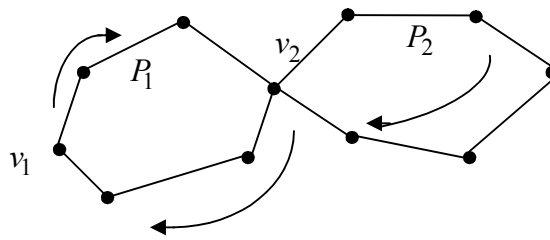


Рис. 4.1

□

Заметим, что данная теорема справедлива также и для мультиграфов, и для псевдографов, исключая тот случай, когда псевдограф имеет только одну вершину.

Данная теорема служит критерием существования эйлерова цикла в графе. Это связано с тем, что эйлеровых графов с числом вершин n среди множества графов с тем же числом вершин почти нет. Можно отметить, что характерной особенностью эйлерова графа является то, что его можно нарисовать, не отрывая карандаша от бумаги и не повторяя линий.

Заметим также, что теорема 4.1 позволяет аргументировать невозможность построения эйлерова цикла в задаче о кенигсбергских мостах: этот граф содержит все вершины нечетной степени.

С другой стороны, в данной задаче о кенигсбергских мостах можно было бы поставить и такой вопрос: «Возможно, ли пройти каждый мост по одному разу, необязательно возвращаясь в исходную точку?» В связи с этим рассмотрим такие понятия, как *эйлеров путь* и *собственный эйлеров путь*.

Пусть $G = G(V, E)$ – граф.

Определение 4.2. Путь, который включает каждое ребро графа G только один раз, называется *эйлеровым путем*. Эйлеров путь, который не является циклом, называется *собственным эйлеровым путем*.

Теорема 4.2*. *Граф (мультиграф или псевдограф) имеет собственный эйлеров путь тогда и только тогда, когда он связный и ровно две его вершины имеют нечетную степень.*

Согласно этой теореме, задача о кенигсбергских мостах также не имеет и эйлерова пути.

Аналогичным образом можно ввести понятие эйлера цикла для ориентированного графа и условие существования эйлера цикла в орграфе.

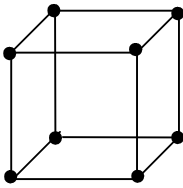
Пусть $G = G(V, E)$ – ориентированный граф.

Определение 4.3. Ориентированный цикл, который включает все ребра и вершины графа G , называется *эйлеровым циклом*.

Теорема 4.3*. *Ориентированный граф имеет эйлеров цикл тогда и только тогда, когда он связный и полустепень захода каждой вершины равна ее полустепени исхода.*

Пример 4.1. Из какого минимального числа кусков проволоки можно спаять каркас куба? Толщина всех ребер каркаса должна быть одинаковой.

Решение.



В этом графе 8 вершин нечетной степени. Используя ниже сформулированную теорему, можно сказать, что минимальное число цепей, на которое можно разбить ребра графа, – 4. Следовательно, наименьшее число кусков проволоки – тоже 4. \square

Теорема 4.4*. *Если связный граф содержит ровно $2n$ вершин нечетной степени, то минимальное число реберно-непересекающихся цепей, на которые можно разбить его ребра, равно n .*

4.2. Пути и циклы Гамильтона. Алгоритм Литтла

В 1857 году математик Уильям Роуэн Гамильтон придумал игру. Существует несколько версий того, как это произошло. По одной из версий он описал игру в письме к другу. Согласно другой, он действительно изобрел игру и продал ее производителю игрушек. В любом случае, данная игра включала додекаэдр, т.е. правильный многогранник, 12 граней которых представляли собой равные правильные пятиугольники. В каждом из 20 углов, или вершин тела, просверливалась дырочка, в которую вставлялся колышек, изображавший город. Используя веревку, требова-

лось найти путь через города, посетив каждый город один раз, и вернуться в исходный город.

В этом случае проблема сводится к нахождению в графе цикла, проходящего через каждую вершину только один раз, исключая начальную. Отсюда любой цикл, обладающий таким свойством, называется гамильтоновым циклом. Этот цикл в некотором смысле противоположен эйлерову циклу, который проходит через все ребра только один раз. До определенного момента оба цикла могут показаться схожими, но на самом деле цикл Гамильтона намного сложнее.

Пусть $G(V, E)$ – граф.

Определение 4.4. *Гамильтонов путь* – это простой путь, который проходит через каждую вершину графа G . *Гамильтонов цикл* – это простой цикл, который проходит через каждую вершину графа G .

До настоящего времени никому не удалось установить необходимые и достаточные условия существования у графа гамильтонова цикла. Тем не менее доказан ряд теорем, в которых приводятся некоторые условия существования у графа гамильтонова цикла.

Рассмотрим одну из задач, в которой требуется отыскать во взвешенном графе гамильтонов цикл минимальной длины. Приведем формулировку задачи и алгоритм ее решения.

Задача коммивояжера (или бродячего торговца)

Постановка задачи: коммивояжер должен так составить свой маршрут движения, чтобы посетить один и только один раз каждый из n городов и вернуться в исходный пункт. Его маршрут должен минимизировать суммарную длину пройденного пути.

После текстовой постановки задачи с помощью математического языка (символов, функций, уравнений или неравенств и т.д.) построим *математическую модель задачи*. Строгое понятие «математическая модель» введем в следующем разделе «Элементы математического моделирования», на данном шаге будем оперировать данным понятием, как некоторым математическим «переводом» текстового условия.

Итак, пусть $C = [c_{ij}]$ – матрица расстояний между городами. Для составления математической модели задачи обозначим через переменные x_{ij}

факт переезда коммивояжером из города i в город j . Поскольку переезд из одного города в другой может осуществляться только один раз, то переменные x_{ij} должны принимать только два значения: 1 или 0, т.е. булевы значения. Таким образом:

$$x_{ij} = \begin{cases} 1, & \text{если коммивояжер из города } i \\ & \text{переезжает непосредственно в город } j; \\ 0, & \text{в противном случае.} \end{cases}$$

Математическая модель задачи имеет следующий вид:

$$Z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min, \quad (4.1)$$

$$\sum_{i=1}^n x_{ij} = 1, \quad (j = \overline{1, n}), \quad (4.2)$$

$$\sum_{j=1}^n x_{ij} = 1, \quad (i = \overline{1, n}), \quad (4.3)$$

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

Система ограничений (4.2) обеспечивает построение маршрута, при котором коммивояжер въезжает в каждый город только один раз, а система (4.3) – маршрута, при котором он выезжает из каждого города только один раз. Устранение подциклов и получение маршрута, образующего полный цикл, включающий все города, достигается при добавлении системы ограничений:

$$u_i - u_j + nx_{ij} \leq n - 1 \quad (i = \overline{1, n}, j = \overline{1, n}, i \neq j), \quad (4.4)$$

где все переменные u_i и u_j могут принимать произвольные действительные значения.

Решение задачи коммивояжера методом ветвей и границ (алгоритм Литтла)

Если считать города вершинами графа, а коммуникации $(i; j)$ – его дугами, то нахождение минимального пути, проходящего один и только

один раз через каждый город с возвращением в исходную точку, можно рассматривать как нахождение на графе гамильтонова цикла минимальной длины.

Рассмотрим алгоритм поиска гамильтонова цикла минимальной длины на графе с n вершинами (алгоритм Литтла). Если между вершинами i и j нет дуги, то ставится символ ∞ . Этот же символ ставится на главной диагонали, что означает запрет на возвращение в вершину, через которую уже проходил цикл. Основная идея метода состоит в том, что сначала строят нижнюю границу длин $\varphi_0(\Omega^0)$ множества гамильтоновых циклов Ω^0 . Затем множество циклов Ω^0 разбивается на два подмножества таким образом, чтобы первое подмножество Ω_{ij}^1 состояло из гамильтоновых циклов, содержащих некоторую дугу (i, j) , а другое подмножество $\tilde{\Omega}_{ij}^1$ не содержало эту дугу. Для каждого из подмножеств определяются нижние границы по тому же правилу, что и для первоначального множества гамильтоновых циклов. Полученные нижние границы подмножеств Ω_{ij}^1 и $\tilde{\Omega}_{ij}^1$ оказываются не меньше нижней границы для всего множества гамильтоновых циклов, то есть

$$\varphi_0(\Omega^0) \leq \varphi(\Omega_{ij}^1) \equiv \varphi_{ij}^1, \quad \varphi_0(\Omega^0) \leq \varphi(\tilde{\Omega}_{ij}^1) \equiv \tilde{\varphi}_{ij}^1.$$

Сравнивая нижние границы φ_{ij}^1 и $\tilde{\varphi}_{ij}^1$ подмножеств Ω_{ij}^1 и $\tilde{\Omega}_{ij}^1$, можно выделить среди них то, которое с большей вероятностью содержит гамильтонов цикл минимальной длины. Затем одно из подмножеств Ω_{ij}^1 или $\tilde{\Omega}_{ij}^1$ по аналогичному правилу разбивается на два новых Ω_{ij}^2 и $\tilde{\Omega}_{ij}^2$. Для них снова определяются нижние границы φ_{ij}^2 и $\tilde{\varphi}_{ij}^2$ и т.д., рис. 4.2.

Процесс ветвления продолжается до тех пор, пока не отыщется единственный гамильтонов цикл. Его называют *первым рекордом*. Затем просматривают оборванные ветви. Если их нижние границы больше длины первого рекорда, то задача решена. Если же найдутся такие, для которых нижние границы меньше, чем длина первого рекорда, то подмножество с наименьшей нижней границей подвергают дальнейшему ветвлению, пока не убеждаются, что оно не содержит лучшего гамильтонова цикла. Если же такой найдется, то анализ оборванных ветвей продолжается относи-

тельно нового значения длины цикла. Его называют *вторым рекордом*. Процесс решения заканчивается тогда, когда будут проанализированы все подмножества.

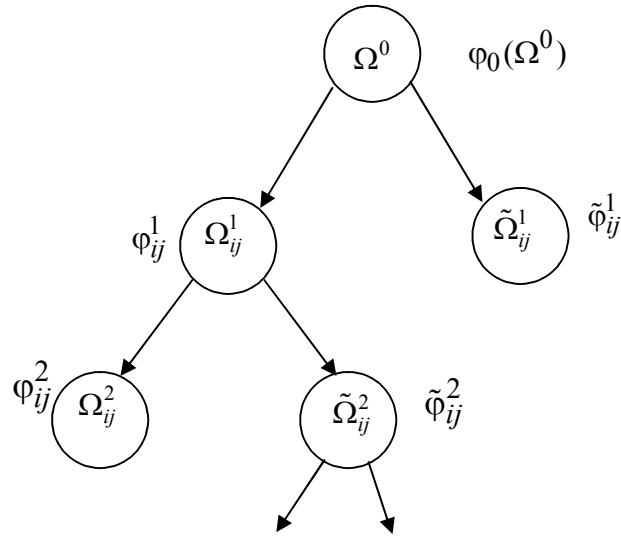


Рис. 4.2

Алгоритм Литтла

Пусть известна матрица весов некоторого орграфа, вершины которого занумеруем числами от 1 до n :

$$\Omega = \left(\begin{array}{c|cccc} & 1 & 2 & \dots & n \\ \hline 1 & \infty & \omega_{12} & \dots & \omega_{1n} \\ 2 & \omega_{21} & \infty & \dots & \omega_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ n & \omega_{n1} & \omega_{n2} & \dots & \infty \end{array} \right),$$

где ω_{ij} – вес дуги, соединяющей вершины i и j , $i = \overline{1, n}$, $j = \overline{1, n}$.

Символ ∞ ставится для блокировки дуг графа, которые явно не включаются в гамильтонов цикл, т.е. в расчет не берутся дуги l_{ii} и $\omega_{ij} = \infty$. Можно показать, что оптимальность решения не изменяется при прибавлении некоторого числа к строке или столбцу матрицы Ω .

Алгоритм Литтла разбивается на несколько шагов.

Шаг 1. Приведение исходной матрицы

В каждом столбце и в каждой строке матрицы Ω нужно получить хотя бы один нуль. Для этого, например, в первом столбце выберем минимальный элемент и вычтем его из всех элементов первого столбца. Аналогично поступим с остальными столбцами. Если при этом в некоторых строках не появляются нули, то для них осуществим ту же процедуру.

После этого вычислим константу приведения φ_0 – сумму минимальных элементов столбцов и строк, использованных для преобразования матрицы Ω . Получим приведенную матрицу Ω_0 с константой приведения φ_0 .

Шаг 2. Определение степеней нулей.

Найдем степени каждого нуля – сумму минимальных элементов строки i_k и столбца j_m , $k = \overline{1, n}$, $m = \overline{1, n}$, в которых стоит этот нуль, без учета самого нуля. Нуль с максимальной степенью определяет дугу $l(i_1; j_1)$, которая вероятнее всего войдет в гамильтонов цикл. Например, если нуль с максимальной степенью находится на пересечении второй строки и третьего столбца, то дуга $l(2; 3)$, вероятнее всего, войдет в гамильтонов цикл.

Шаг 3. Ветвление.

На самом деле, дуга, соответствующая максимальной степени нуля, может как входить в гамильтонов цикл, так и не входить в него. Поэтому следует рассмотреть сразу два случая.

Первый случай. Возможно, дуга $l(i_1; j_1)$ вошла в гамильтонов цикл. Блокируем ее, полагая $\omega_{j_1 i_1} = \infty$. Строку i_1 и столбец j_1 вычеркиваем. Если требуется, приводим полученную матрицу меньшего порядка $\Omega_1(i_1; j_1)$ с константой приведения $\varphi_1 = \varphi_0 + \Delta_1$.

Второй случай. Дуга $l(i_1; j_1)$ не вошла в гамильтонов цикл. Полагаем $\omega_{ij} = \infty$. Если требуется, приводим полученную матрицу $\tilde{\Omega}_1(i_1; j_1)$ с константой приведения $\tilde{\varphi}_1 = \varphi_0 + \tilde{\Delta}_1$.

Если $\varphi_1 < \tilde{\varphi}_1$, то шаги 2, 3 повторяем с матрицей $\Omega_1(i_1; j_1)$.

Если $\varphi_1 > \tilde{\varphi}_1$, то шаги 2, 3 повторяем с матрицей $\tilde{\Omega}_1(i_1; j_1)$. И так до тех пор, пока не дойдем до матрицы второго порядка, содержащей два нуля:

$$\Omega_{n-2}(i_{n-2}; j_{n-2}) = \left(\begin{array}{c|cc} & j_{n-1} & j_n \\ \hline i_{n-1} & 0 & A \\ i_n & B & 0 \end{array} \right),$$

где A и B – некоторые числа или ∞ . Нули соответствуют двум последним дугам гамильтонова цикла: $l(i_{n-1}; j_{n-1})$, $l(i_n; j_n)$, при этом $\varphi = \varphi_{n-2}$.

Если $\varphi_{n-2} < \tilde{\varphi}_k$, где $k = 1, 2, \dots, n-2$, то задача решена. Если же для некоторого k_0 получается $\varphi_{n-2} > \tilde{\varphi}_{k_0}$, то всю процедуру следует провести с матрицей $\tilde{\Omega}_k(i_k; j_k)$.

На последнем этапе получим новое значение функции φ : φ'_{n-2} . Это значение сравниваем с φ_{n-2} , то есть процесс продолжается до тех пор, пока новые значения φ'_k не станут меньше φ_{n-2} .

Пример 4.2. Коммивояжер должен посетить пять городов, заезжая в каждый город по одному разу. Расстояния между городами следующие: между первым городом и вторым – 12 км, между первым и третьим – 25 км, между первым и четвертым – 16 км, между первым и пятым – 31 км, между вторым и третьим – 28 км, между вторым и четвертым – 35 км, между вторым и пятым – 22 км, между третьим и четвертым – 36 км, между третьим и пятым – 20 км, между четвертым и пятым – 19 км. Составленный маршрут должен минимизировать суммарную длину пройденного пути.

Решение.

Сначала получим приведенный вид данной матрицы. Для этого пронумеруем строки и столбцы. В каждом столбце определим минимальный элемент и запишем его в дополнительной нижней строке:

$$A = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 12 & 25 & 16 & 31 \\ 2 & 12 & \infty & 28 & 35 & 22 \\ 3 & 25 & 28 & \infty & 36 & 20 \\ 4 & 16 & 35 & 36 & \infty & 19 \\ 5 & 31 & 22 & 20 & 19 & \infty \\ \hline & 12 & 12 & 20 & 16 & 19 \end{array} \right).$$

Из каждого элемента столбца вычтем соответствующий минимальный элемент. Получим матрицу A_1 :

$$A_1 = \left(\begin{array}{c|ccccc|c} & 1 & 2 & 3 & 4 & 5 & \\ \hline 1 & \infty & 0 & 5 & 0 & 12 & 0 \\ 2 & 0 & \infty & 8 & 19 & 3 & 0 \\ 3 & 13 & 16 & \infty & 20 & 1 & 1 \\ 4 & 4 & 23 & 16 & \infty & 0 & 0 \\ 5 & 19 & 10 & 0 & 3 & \infty & 0 \end{array} \right).$$

Матрица A_1 оказалась неприведенной, поэтому определим минимальный элемент в каждой строке и вычтем его из всех элементов соответствующей строки. В результате получаем приведенную матрицу A_0 :

$$A_0 = \left(\begin{array}{c|ccccc|c} & 1 & 2 & 3 & 4 & 5 & \\ \hline 1 & \infty & 0^{(10)} & 5 & 0^{(3)} & 12 & \\ 2 & 0^{(7)} & \infty & 8 & 19 & 3 & \\ 3 & 12 & 15 & \infty & 19 & 0^{(12)} & \\ 4 & 4 & 23 & 16 & \infty & 0^{(4)} & \\ 5 & 19 & 10 & 0^{(8)} & 3 & \infty & \end{array} \right).$$

Вычислим константу приведения φ_0 :

$$\varphi_0 = 12 + 12 + 20 + 16 + 19 + 0 + 0 + 1 + 0 + 0 = 80.$$

Найдем степени каждого нуля – сумму минимальных элементов строки и столбца, в которых стоит ноль (без учета самого нуля). К каждому нулю припишем сверху его степень. Максимальной степенью является число 12.

Нули с максимальной степенью определяют дуги, которые вероятнее всего войдут в гамильтонов цикл. В нашем случае наиболее вероятной дугой гамильтонова цикла является $l(3;5)$.

Выберем дугу $l(3;5)$.

В связи с этим рассмотрим две матрицы: $A_1(3;5)$ и $\tilde{A}_1(3;5)$. В матрице $A_1(3;5)$ уберем третью строку и пятый столбец, элемент a_{53} заменим на ∞ .

В матрице $\tilde{A}_1(3;5)$ элемент a_{35} заменим на ∞ . Получим:

$$A_1(3;5) = \left(\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & \infty & 0 & 5 & 0 \\ 2 & 0 & \infty & 8 & 19 \\ 4 & 4 & 23 & 16 & \infty \\ 5 & 19 & 10 & \infty & 3 \end{array} \right), \quad \tilde{A}_1(3;5) = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 0 & 5 & 0 & 12 \\ 2 & 0 & \infty & 8 & 19 & 3 \\ 3 & 12 & 15 & \infty & 19 & \infty \\ 4 & 4 & 23 & 16 & \infty & 0 \\ 5 & 19 & 10 & 0 & 3 & \infty \end{array} \right).$$

Обе матрицы не являются приведенными. Для приведения матриц определим минимальные элементы строк и столбцов.

Сначала будем работать с матрицей $A_1(3;5)$. Определим минимальные элементы строк и вычтем их из каждого элемента соответствующих строк:

$$A_1(3;5) = \left(\begin{array}{c|cccc|c} & 1 & 2 & 3 & 4 & \\ \hline 1 & \infty & 0 & 5 & 0 & 0 \\ 2 & 0 & \infty & 8 & 19 & 0 \\ 4 & 4 & 23 & 16 & \infty & 4 \\ 5 & 19 & 10 & \infty & 3 & 3 \end{array} \right) \Rightarrow A_1(3;5) = \left(\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & \infty & 0 & 5 & 0 \\ 2 & 0 & \infty & 8 & 19 \\ 4 & 0 & 19 & 12 & \infty \\ 5 & 16 & 7 & \infty & 0 \end{array} \right).$$

Определим минимальные элементы каждого столбца и вычтем их из каждого элемента соответствующего столбца:

$$A_1(3;5) = \left(\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & \infty & 0 & 5 & 0 \\ 2 & 0 & \infty & 8 & 19 \\ 4 & 0 & 19 & 12 & \infty \\ 5 & 16 & 7 & \infty & 0 \\ \hline & 0 & 0 & 5 & 0 \end{array} \right) \Rightarrow A_1(3;5) = \left(\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & \infty & 0 & 0 & 0 \\ 2 & 0 & \infty & 3 & 19 \\ 4 & 0 & 19 & 7 & \infty \\ 5 & 16 & 7 & \infty & 0 \end{array} \right).$$

Аналогично преобразуем матрицу $\tilde{A}_1(3;5)$.

$$\tilde{A}_1(3;5) = \left(\begin{array}{c|ccccc|c} & 1 & 2 & 3 & 4 & 5 & \\ \hline 1 & \infty & 0 & 5 & 0 & 12 & 0 \\ 2 & 0 & \infty & 8 & 19 & 3 & 0 \\ 3 & 12 & 15 & \infty & 19 & \infty & 12 \\ 4 & 4 & 23 & 16 & \infty & 0 & 0 \\ 5 & 19 & 10 & 0 & 3 & \infty & 0 \\ \hline & 0 & 0 & 0 & 0 & 0 & \end{array} \right) \Rightarrow \tilde{A}_1(3;5) = \left(\begin{array}{c|ccccc|c} & 1 & 2 & 3 & 4 & 5 & \\ \hline 1 & \infty & 0 & 5 & 0 & 12 & \\ 2 & 0 & \infty & 8 & 19 & 3 & \\ 3 & 0 & 3 & \infty & 7 & \infty & \\ 4 & 4 & 23 & 16 & \infty & 0 & \\ 5 & 19 & 10 & 0 & 3 & \infty & \end{array} \right).$$

Таким образом, получены две приведенные матрицы:

$$A_1(3;5) = \left(\begin{array}{c|cccc|c} & 1 & 2 & 3 & 4 & \\ \hline 1 & \infty & 0 & 0 & 0 & \\ 2 & 0 & \infty & 3 & 19 & \\ 4 & 0 & 19 & 7 & \infty & \\ 5 & 16 & 7 & \infty & 0 & \end{array} \right) \quad \text{и} \quad \tilde{A}_1(3;5) = \left(\begin{array}{c|ccccc|c} & 1 & 2 & 3 & 4 & 5 & \\ \hline 1 & \infty & 0 & 5 & 0 & 12 & \\ 2 & 0 & \infty & 8 & 19 & 3 & \\ 3 & 0 & 3 & \infty & 7 & \infty & \\ 4 & 4 & 23 & 16 & \infty & 0 & \\ 5 & 19 & 10 & 0 & 3 & \infty & \end{array} \right).$$

Вычислим константы приведения:

$$\varphi_1(3;5) = \varphi_0 + \Delta_1(3;5) = 80 + 5 + 4 + 3 + 0 = 92,$$

$$\tilde{\varphi}_1(3;5) = \varphi_0 + \tilde{\Delta}_1(3;5) = 80 + 12 + 0 = 92,$$

где $\Delta_1(3;5)$ и $\tilde{\Delta}_1(3;5)$ – суммы минимальных элементов строк и столбцов матриц $A_1(3;5)$ и $\tilde{A}_1(3;5)$.

В данном случае $\varphi_1(3;5) = \tilde{\varphi}_1(3;5)$, но далее рассмотрим матрицу $A_1(3;5)$. Определим степени нулей этой матрицы:

$$A_1(3;5) = \left(\begin{array}{c|cccc|c} & 1 & 2 & 3 & 4 & \\ \hline 1 & \infty & 0^{(7)} & 0^{(3)} & 0^{(0)} & \\ 2 & 0^{(3)} & \infty & 3 & 19 & \\ 4 & 0^{(7)} & 19 & 7 & \infty & \\ 5 & 16 & 7 & \infty & 0^{(7)} & \end{array} \right).$$

Максимальная степень – число 7. Наиболее вероятными дугами гамильтонова цикла являются: $l(1;2)$, $l(4;1)$, $l(5,4)$. Выберем, например, дугу $l(1;2)$.

В связи с этим рассмотрим две матрицы: $A_2(1;2)$ и $\tilde{A}_2(1;2)$. В матрице $A_1(3;5)$ уберем первую строку и второй столбец, элемент a_{21} заменим на ∞ , получим матрицу $A_2(1;2)$. В матрице $\tilde{A}_2(1;2)$ элемент a_{12} заменим на ∞ . Имеем:

$$A_2(1;2) = \left(\begin{array}{c|cccc} & 1 & 3 & 4 & \\ \hline 2 & \infty & 3 & 19 & \\ 4 & 0 & 7 & \infty & \\ 5 & 16 & \infty & 0 & \end{array} \right), \quad \tilde{A}_2(1;2) = \left(\begin{array}{c|cccc} & 1 & 2 & 3 & 4 & \\ \hline 1 & \infty & \infty & 0 & 0 & \\ 2 & 0 & \infty & 3 & 19 & \\ 4 & 0 & 19 & 7 & \infty & \\ 5 & 16 & 7 & \infty & 0 & \end{array} \right).$$

Обе матрицы не являются приведенными. Преобразуем их по той же схеме, как это делали выше:

$$A_2(1;2) = \left(\begin{array}{c|cccc|c} & 1 & 3 & 4 & & \\ \hline 2 & \infty & 3 & 19 & & 3 \\ 4 & 0 & 7 & \infty & & 0 \\ 5 & 16 & \infty & 0 & & 0 \end{array} \right) \Rightarrow A_2(1;2) = \left(\begin{array}{c|cccc} & 1 & 3 & 4 & \\ \hline 2 & \infty & 0 & 16 & \\ 4 & 0 & 7 & \infty & \\ 5 & 16 & \infty & 0 & \end{array} \right),$$

$$\tilde{A}_2(1;2) = \left(\begin{array}{c|cccc} & 1 & 2 & 3 & 4 & \\ \hline 1 & \infty & \infty & 0 & 0 & \\ 2 & 0 & \infty & 3 & 19 & \\ 4 & 0 & 19 & 7 & \infty & \\ 5 & 16 & 7 & \infty & 0 & \\ \hline & 0 & 7 & 0 & 0 & \end{array} \right) \Rightarrow \tilde{A}_2(1;2) = \left(\begin{array}{c|cccc} & 1 & 2 & 3 & 4 & \\ \hline 1 & \infty & \infty & 0 & 0 & \\ 2 & 0 & \infty & 3 & 19 & \\ 4 & 0 & 12 & 7 & \infty & \\ 5 & 16 & 0 & \infty & 0 & \end{array} \right).$$

Таким образом, получены две приведенные матрицы $A_2(1;2)$ и $\tilde{A}_2(1;2)$.

Вычислим константы приведения:

$$\varphi_2(1;2) = \varphi_1(3;5) + \Delta_2(1;2) = 92 + 3 = 95,$$

$$\tilde{\varphi}_2(1;2) = \varphi_1(3;5) + \tilde{\Delta}_2(1;2) = 92 + 7 = 99.$$

Так как $\varphi_2(1;2) < \tilde{\varphi}_2(1;2)$, то далее следует рассмотреть матрицу $A_2(1;2)$. Определим степени каждого нуля.

$$A_2(1;2) = \left(\begin{array}{c|ccc} & 1 & 3 & 4 \\ \hline 2 & \infty & 0^{(23)} & 16 \\ 4 & 0^{(23)} & 7 & \infty \\ 5 & 16 & \infty & 0^{(32)} \end{array} \right).$$

Максимальная степень – число 32. Наиболее вероятной дугой гамильтонова цикла является дуга $l(5;4)$.

В связи с этим рассмотрим две матрицы: $A_3(5;4)$ и $\tilde{A}_3(5;4)$. В матрице $A_2(1;2)$ уберем строку под номером 5 и столбец под номером 4, получим матрицу $A_3(5;4)$. В матрице $\tilde{A}_3(5;4)$ элемент a_{54} заменим на ∞ .

Имеем:

$$A_3(5;4) = \left(\begin{array}{c|cc} & 1 & 3 \\ \hline 2 & \infty & 0 \\ 4 & 0 & 7 \end{array} \right), \quad \tilde{A}_3(5;4) = \left(\begin{array}{c|ccc} & 1 & 3 & 4 \\ \hline 2 & \infty & 0 & 16 \\ 4 & 0 & 7 & \infty \\ 5 & 16 & \infty & \infty \end{array} \right).$$

Матрица $\tilde{A}_3(5;4)$ не является приведенной. Преобразуем ее по той же схеме, как это делали выше.

$$\tilde{A}_3(5;4) = \left(\begin{array}{c|ccc|c} & 1 & 3 & 4 & \\ \hline 2 & \infty & 0 & 16 & 0 \\ 4 & 0 & 7 & \infty & 0 \\ 5 & 16 & \infty & \infty & 16 \\ \hline & 0 & 0 & 16 & \end{array} \right) \Rightarrow \tilde{A}_3(5;4) = \left(\begin{array}{c|ccc} & 1 & 3 & 4 \\ \hline 2 & \infty & 0 & 0 \\ 4 & 0 & 7 & \infty \\ 5 & 0 & \infty & \infty \end{array} \right).$$

Тогда

$$\varphi_3(5;4) = \varphi_2(1;2) + \Delta_3(5;4) = 95 + 0 = 95,$$

$$\tilde{\varphi}_3(5;4) = \varphi_2(1;2) + \tilde{\Delta}_3(5;4) = 95 + 32 = 127.$$

Так как $\varphi_3(5;4) < \tilde{\varphi}_3(5;4)$, то далее рассмотрим матрицу $A_3(5;4)$. Последние две дуги гамильтонова цикла определим из матрицы $A_3(5;4)$: $l(2;3)$ и $l(4;1)$.

Таким образом, получим решение, которое состоит из следующих дуг: $(3;5)$, $(1;2)$, $(5;4)$, $(2;3)$ и $(4;1)$. Гамильтонов цикл имеет вид: $1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 4 \rightarrow 1$, длина цикла составляет 95 км.

Сравним константу приведения $\varphi_3(5;4)$ с константами приведения $\tilde{\varphi}_k$ альтернативных вариантов ($k = \overline{1,3}$):

$$\varphi_3(5;4) > \tilde{\varphi}_1(3;5): 95 > 92; \quad \varphi_3(5;4) < \tilde{\varphi}_2(1;2): 95 < 99;$$

$$\varphi_3(5;4) < \tilde{\varphi}_3(5;4): 95 < 127.$$

Так как $\varphi_3(5;4) > \tilde{\varphi}_1(3;5)$, то вернемся к приведенной матрице $\tilde{A}_1(3;5)$ и от нее начнем строить гамильтонов цикл.

Определим степени нулей этой матрицы:

$$\tilde{A}_1(3;5) = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 0^{(3)} & 5 & 0^{(3)} & 12 \\ 2 & 0^{(3)} & \infty & 8 & 19 & 3 \\ 3 & 0^{(3)} & 3 & \infty & 7 & \infty \\ 4 & 4 & 23 & 16 & \infty & 0^{(7)} \\ 5 & 19 & 10 & 0^{(8)} & 3 & \infty \end{array} \right).$$

Максимальная степень – число 8. Наиболее вероятной дугой гамильтонова цикла является дуга $l(5;3)$.

В связи с этим рассмотрим две матрицы: $A'_1(5;3)$ и $\tilde{A}'_1(5;3)$. В матрице $A'_1(5;3)$ уберем пятую строку и третий столбец, элемент a_{35} уже заменен на ∞ . В матрице $\tilde{A}'_1(5;3)$ элемент a_{53} заменим на ∞ . Получим:

$$A'_1(5;3) = \left(\begin{array}{c|cccc} & 1 & 2 & 4 & 5 \\ \hline 1 & \infty & 0 & 0 & 12 \\ 2 & 0 & \infty & 19 & 3 \\ 3 & 0 & 3 & 7 & \infty \\ 4 & 4 & 23 & \infty & 0 \end{array} \right), \quad \tilde{A}'_1(5;3) = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 0 & 5 & 0 & 12 \\ 2 & 0 & \infty & 8 & 19 & 3 \\ 3 & 0 & 3 & \infty & 7 & \infty \\ 4 & 4 & 23 & 16 & \infty & 0 \\ 5 & 19 & 10 & \infty & 3 & \infty \end{array} \right).$$

Матрица $A'_1(5;3)$ – приведенная, а матрица $\tilde{A}'_1(5;3)$ не является приведенной, поэтому над ней следует выполнить преобразования:

$$\tilde{A}'_1(5;3) = \left(\begin{array}{c|ccccc|c} & 1 & 2 & 3 & 4 & 5 & \\ \hline 1 & \infty & 0 & 5 & 0 & 12 & 0 \\ 2 & 0 & \infty & 8 & 19 & 3 & 0 \\ 3 & 0 & 3 & \infty & 7 & \infty & 0 \\ 4 & 4 & 23 & 16 & \infty & 0 & 0 \\ 5 & 19 & 10 & \infty & 3 & \infty & 3 \\ \hline & 0 & 0 & 5 & 0 & 0 & \end{array} \right) \Rightarrow \tilde{A}'_1(5;3) = \left(\begin{array}{c|ccccc|c} & 1 & 2 & 3 & 4 & 5 & \\ \hline 1 & \infty & 0 & 0 & 0 & 12 & \\ 2 & 0 & \infty & 3 & 19 & 3 & \\ 3 & 0 & 3 & \infty & 7 & \infty & \\ 4 & 4 & 23 & 11 & \infty & 0 & \\ 5 & 16 & 7 & \infty & 0 & \infty & \end{array} \right).$$

Вычислим константы приведения:

$$\varphi'_1(5;3) = \tilde{\varphi}_1(3;5) + \Delta'_1(5;3) = 92 + 0 = 92,$$

$$\tilde{\varphi}'_1(5;3) = \tilde{\varphi}_1(3;5) + \tilde{\Delta}'_1(5;3) = 92 + 3 + 5 + 0 = 100,$$

где $\Delta'_1(5;3)$ и $\tilde{\Delta}'_1(5;3)$ – суммы минимальных элементов строк и столбцов матриц $A'_1(5;3)$ и $\tilde{A}'_1(5;3)$.

Так как $\varphi'_1(5;3) < \tilde{\varphi}'_1(5;3)$, то далее рассмотрим матрицу $A'_1(5;3)$. Определим степени нулей этой матрицы:

$$A'_1(5;3) = \left(\begin{array}{c|cccc} & 1 & 2 & 4 & 5 \\ \hline 1 & \infty & 0^{(3)} & 0^{(7)} & 12 \\ 2 & 0^{(3)} & \infty & 19 & 3 \\ 3 & 0^{(3)} & 3 & 7 & \infty \\ 4 & 4 & 23 & \infty & 0^{(7)} \end{array} \right).$$

Максимальная степень – число 7. Наиболее вероятными дугами гамильтонова цикла являются $l(1;4)$ и $l(4;5)$. Выберем дугу $l(4;5)$.

В связи с этим рассмотрим две матрицы: $A'_2(4;5)$ и $\tilde{A}'_2(4;5)$. В матрице $A'_1(5;3)$ уберем строку под номером 4 и столбец под номером 5, получим матрицу $A'_2(4;5)$. В матрице $\tilde{A}'_2(4;5)$ элемент a_{45} заменим на ∞ . Имеем:

$$A'_2(4;5) = \left(\begin{array}{c|ccc} & 1 & 2 & 4 \\ \hline 1 & \infty & 0 & 0 \\ 2 & 0 & \infty & 19 \\ 3 & 0 & 3 & 7 \end{array} \right), \quad \tilde{A}'_2(4;5) = \left(\begin{array}{c|cccc} & 1 & 2 & 4 & 5 \\ \hline 1 & \infty & 0 & 0 & 12 \\ 2 & 0 & \infty & 19 & 3 \\ 3 & 0 & 3 & 7 & \infty \\ 4 & 4 & 23 & \infty & \infty \end{array} \right).$$

Матрица $A'_2(4;5)$ – приведенная, а матрица $\tilde{A}'_2(4;5)$ не является приведенной. Выполним ее приведение:

$$\tilde{A}'_2(4;5) = \left(\begin{array}{c|cccc|c} & 1 & 2 & 4 & 5 & \\ \hline 1 & \infty & 0 & 0 & 12 & 0 \\ 2 & 0 & \infty & 19 & 3 & 0 \\ 3 & 0 & 3 & 7 & \infty & 0 \\ 4 & 4 & 23 & \infty & \infty & 4 \\ \hline & 0 & 0 & 0 & 3 & \end{array} \right) \Rightarrow \tilde{A}'_2(4;5) = \left(\begin{array}{c|cccc} & 1 & 2 & 4 & 5 \\ \hline 1 & \infty & 0 & 0 & 9 \\ 2 & 0 & \infty & 19 & 0 \\ 3 & 0 & 3 & 7 & \infty \\ 4 & 0 & 19 & \infty & \infty \end{array} \right).$$

Вычисляем константы приведения:

$$\varphi'_2(4;5) = \varphi'_1(5;3) + \Delta'_2(4;5) = 92 + 0 = 92,$$

$$\tilde{\varphi}'_2(4;5) = \varphi'_1(5;3) + \tilde{\Delta}'_2(4;5) = 92 + 3 + 4 = 99.$$

Так как $\varphi'_2(4;5) < \tilde{\varphi}'_2(4;5)$, то далее рассмотрим матрицу $A'_2(4;5)$. Определим степени каждого нуля:

$$A'_2(4;5) = \left(\begin{array}{c|ccc} & 1 & 2 & 4 \\ \hline 1 & \infty & 0^{(3)} & 0^{(7)} \\ 2 & 0^{(19)} & \infty & 19 \\ 3 & 0^{(3)} & 3 & 7 \end{array} \right).$$

Максимальная степень – число 19. Наиболее вероятной дугой гамильтонова цикла является дуга $l(2;1)$.

В связи с этим рассмотрим две матрицы: $A'_3(2;1)$ и $\tilde{A}'_3(2;1)$. В матрице $A'_2(4;5)$ уберем строку под номером 2 и столбец под номером 1, элемент

a_{12} заменим на ∞ и получим матрицу $A'_3(2;1)$. В матрице $\tilde{A}'_3(2;1)$ элемент a_{21} заменим на ∞ . Имеем:

$$A'_3(2;1) = \left(\begin{array}{c|cc} & 2 & 4 \\ \hline 1 & \infty & 0 \\ 3 & 3 & 7 \end{array} \right), \quad \tilde{A}'_3(2;1) = \left(\begin{array}{c|ccc} & 1 & 2 & 4 \\ \hline 1 & \infty & 0 & 0 \\ 2 & \infty & \infty & 19 \\ 3 & 0 & 3 & 7 \end{array} \right).$$

Обе матрицы не являются приведенными. Преобразуем их по той же схеме, как это выполняли выше.

$$A'_3(2;1) = \left(\begin{array}{c|cc|c} & 2 & 4 & \\ \hline 1 & \infty & 0 & 0 \\ 3 & 3 & 7 & 3 \end{array} \right) \Rightarrow A'_3(2;1) = \left(\begin{array}{c|cc} & 2 & 4 \\ \hline 1 & \infty & 0 \\ 3 & 0 & 4 \end{array} \right),$$

$$\tilde{A}'_3(2;1) = \left(\begin{array}{c|ccc|c} & 1 & 2 & 4 & \\ \hline 1 & \infty & 0 & 0 & 0 \\ 2 & \infty & \infty & 19 & 19 \\ 3 & 0 & 3 & 7 & 0 \end{array} \right) \Rightarrow \tilde{A}'_3(2;1) = \left(\begin{array}{c|ccc} & 1 & 2 & 4 \\ \hline 1 & \infty & 0 & 0 \\ 2 & \infty & \infty & 0 \\ 3 & 0 & 3 & 7 \end{array} \right).$$

$$\varphi'_3(2;1) = \varphi'_2(4;5) + \Delta'_3(2;1) = 92 + 3 = 95,$$

$$\tilde{\varphi}'_3(2;1) = \tilde{\varphi}'_2(4;5) + \tilde{\Delta}'_3(2;1) = 92 + 19 = 111.$$

Так как $\varphi'_3(2;1) < \tilde{\varphi}'_3(2;1)$, то далее рассмотрим матрицу $A'_3(2;1)$. Последние две дуги гамильтонова цикла определим из матрицы $A'_3(2;1)$: $l(1;4)$ и $l(3,2)$.

Таким образом, получаем решение, которое состоит из следующих дуг: $(5;3)$, $(4;5)$, $(2;1)$, $(1;4)$, $(3;2)$. Гамильтонов цикл имеет вид $1 \rightarrow 4 \rightarrow 5 \rightarrow 3 \rightarrow 2 \rightarrow 1$, длина цикла также составляет 95 км.

Сравним константу приведения $\varphi'_3(2;1)$ с константами приведения $\tilde{\varphi}'_k$ альтернативных вариантов ($k = \bar{1}, 3$):

$$\varphi'_3(2;1) < \tilde{\varphi}'_1(5;3): 95 < 100,$$

$$\varphi'_3(2;1) < \tilde{\varphi}'_2(4;5): 95 < 99,$$

$$\varphi'_3(2;1) < \tilde{\varphi}'_3(2;1): 95 < 111.$$

Это значит, что получено оптимальное решение.

Весь процесс поиска оптимального плана можно изобразить в виде дерева ветвления, рис. 4.3.

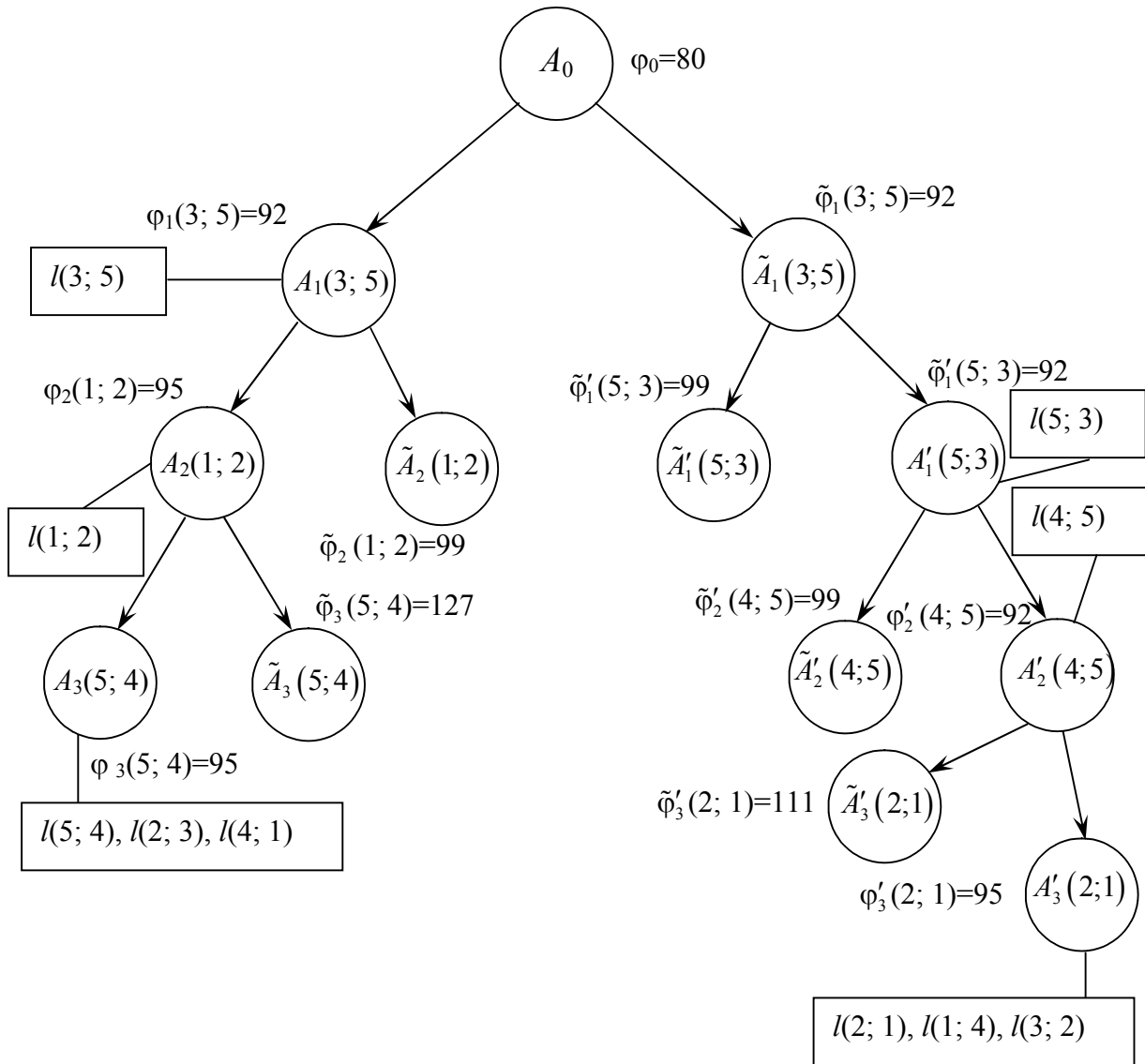
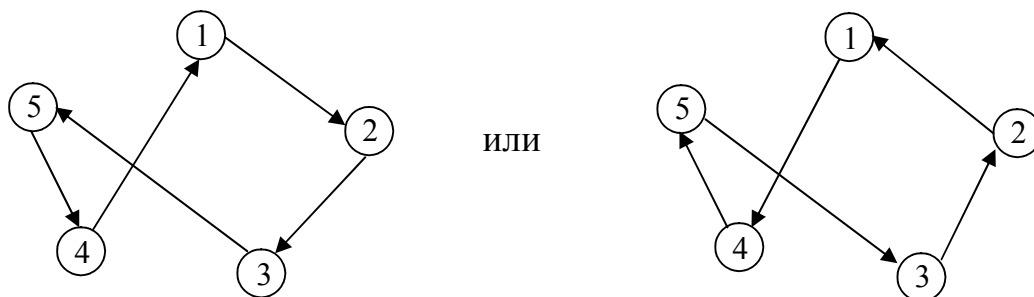


Рис. 4.3

Полученные гамильтоновы циклы $1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 4 \rightarrow 1$ или $1 \rightarrow 4 \rightarrow 5 \rightarrow 3 \rightarrow 2 \rightarrow 1$ можно изобразить в виде орграфов:



В первом случае это означает следующее: коммивояжер сначала посещает город 1, потом – город 2, затем – город 3, после – город 5 и, наконец, город 4, а потом возвращается в город 1. При этом длина всего пути составит 95 км. Второе решение говорит о том, что коммивояжер может ехать в противоположном направлении. □

С помощью метода ветвей и границ, используя современное программное обеспечение ЭВМ, можно решать задачи коммивояжера для $n \leq 40$. В связи с малой эффективностью точных методов получили широкое применение эвристические. В настоящее время существует более ста приближенных методов решения задачи коммивояжера, среди которых наиболее прост *метод ближайшего соседа*. Он реализует требование включения в искомый замкнутый цикл вершины, ближайшей к только что найденной. Алгоритм метода состоит в последовательном добавлении к начальной вершине следующей ближайшей к ней и т.д. Метод очень прост, однако степень приближения к оптимальному решению существенно зависит от выбора начальной вершины. Поэтому алгоритм целесообразно применить к каждой из вершин, как к начальной, и только затем выбрать замкнутый цикл наименьшей длины. Отметим, что если ближайший сосед для некоторой вершины уже вошел в цикл, то берется следующая по близости вершина и т.д. Например, используя данный метод для полного взвешенного графа с n вершинами, потребуется рассмотреть $n!$ гамильтоновых циклов, из которых выбирается цикл наименьшей длины.

Классическая задача коммивояжера, только в несколько иной формулировке, нашла применение и при организации производства в машиностроении – задача о переналадке оборудования при переходе линии на обработку деталей, схожих по конструкции и габаритам.

Пример 4.3. На одной и той же поточной линии предприятие может обрабатывать пять видов деталей. Время наладки при переходе линии на

обработку от одного вида деталей к другому представлена матрицей $A = [a_{ij}]$, где a_{ij} – затраты времени (часы) на наладку переменного-поточной линии при переходе к обработке деталей от i -того вида к деталям j -того вида. С помощью алгоритма Литтла найти последовательность перестройки линии с одной детали на другую, при которой должны быть обеспечены минимальные общие потери рабочего времени на ее переналадку.

$$A = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 10 & 8 & 11 & 10 \\ 2 & 10 & \infty & 13 & 9 & 10 \\ 3 & 8 & 13 & \infty & 9 & 11 \\ 4 & 11 & 9 & 9 & \infty & 12 \\ 5 & 10 & 10 & 11 & 12 & \infty \end{array} \right).$$

Решение. Сначала получим приведенный вид данной матрицы. В каждом столбце определим минимальный элемент и запишем его в нижней строке:

$$A = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 10 & 8 & 11 & 10 \\ 2 & 10 & \infty & 13 & 9 & 10 \\ 3 & 8 & 13 & \infty & 9 & 11 \\ 4 & 11 & 9 & 9 & \infty & 12 \\ 5 & 10 & 10 & 11 & 12 & \infty \\ \hline & 8 & 9 & 8 & 9 & 10 \end{array} \right).$$

Из каждого элемента столбца вычтем соответствующий минимальный элемент и получим матрицу A_0 :

$$A_0 = \left(\begin{array}{c|ccccc|c} & 1 & 2 & 3 & 4 & 5 & \\ \hline 1 & \infty & 1 & 0 & 2 & 0 & 0 \\ 2 & 2 & \infty & 5 & 0 & 0 & 0 \\ 3 & 0 & 4 & \infty & 0 & 1 & 0 \\ 4 & 3 & 0 & 1 & \infty & 2 & 0 \\ 5 & 2 & 1 & 3 & 3 & \infty & 1 \end{array} \right).$$

Матрица A_0 не является приведенной, поэтому определим минимальный элемент в каждой строке и вычтем его из всех элементов соответствующей строки. В результате получим приведенную матрицу A'_0 :

$$A'_0 = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 1 & 0 & 2 & 0 \\ 2 & 2 & \infty & 5 & 0 & 0 \\ 3 & 0 & 4 & \infty & 0 & 1 \\ 4 & 3 & 0 & 1 & \infty & 2 \\ 5 & 1 & 0 & 2 & 2 & \infty \end{array} \right).$$

Вычислим константу приведения φ_0 :

$$\varphi_0 = 8 + 9 + 8 + 9 + 10 + 0 + 0 + 0 + 0 + 1 = 45.$$

Найдем степени каждого нуля – сумму минимальных элементов строки и столбца, в которых стоит ноль. К каждому нулю припишем сверху его степень:

$$A'_0 = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 1 & 0^{(1)} & 2 & 0^{(0)} \\ 2 & 2 & \infty & 5 & 0^{(0)} & 0^{(0)} \\ 3 & 0^{(1)} & 4 & \infty & 0^{(0)} & 1 \\ 4 & 3 & 0^{(1)} & 1 & \infty & 2 \\ 5 & 1 & 0^{(1)} & 2 & 2 & \infty \end{array} \right).$$

Максимальная степень равна 1. Нули с максимальной степенью определяют дуги, которые вероятнее всего войдут в гамильтонов цикл. В нашем случае наиболее вероятными дугами гамильтонова цикла являются $l(1;3)$, $l(3;1)$, $l(4;2)$, $l(5;2)$.

Выберем дугу $l(1;3)$. В связи с этим рассмотрим две матрицы: $A_1(1;3)$ и $\tilde{A}_1(1;3)$. В матрице $A_1(1;3)$ уберем первую строку и третий столбец, элемент a_{31} заменим на ∞ . В матрице $\tilde{A}_1(1;3)$ элемент a_{13} заменим на ∞ . Получим:

$$A_1(1;3) = \left(\begin{array}{c|ccccc} & 1 & 2 & 4 & 5 \\ \hline 2 & 2 & \infty & 0 & 0 \\ 3 & \infty & 4 & 0 & 1 \\ 4 & 3 & 0 & \infty & 2 \\ 5 & 1 & 0 & 2 & \infty \end{array} \right), \quad \tilde{A}_1(1;3) = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 1 & \infty & 2 & 0 \\ 2 & 2 & \infty & 5 & 0 & 0 \\ 3 & 0 & 4 & \infty & 0 & 1 \\ 4 & 3 & 0 & 1 & \infty & 2 \\ 5 & 1 & 0 & 2 & 2 & \infty \end{array} \right).$$

Обе матрицы не являются приведенными. Для их приведения определим минимальные элементы столбцов и выполним их вычитание из элементов соответствующих столбцов:

$$A_1(1;3) = \left(\begin{array}{c|ccccc} & 1 & 2 & 4 & 5 \\ \hline 2 & 2 & \infty & 0 & 0 \\ 3 & \infty & 4 & 0 & 1 \\ 4 & 3 & 0 & \infty & 2 \\ 5 & 1 & 0 & 2 & \infty \\ \hline & 1 & 0 & 0 & 0 \end{array} \right), \quad \tilde{A}_1(1;3) = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 1 & \infty & 2 & 0 \\ 2 & 2 & \infty & 5 & 0 & 0 \\ 3 & 0 & 4 & \infty & 0 & 1 \\ 4 & 3 & 0 & 1 & \infty & 2 \\ 5 & 1 & 0 & 2 & 2 & \infty \\ \hline & 0 & 0 & 1 & 0 & 0 \end{array} \right).$$

В результате получим следующие приведенные матрицы: $A_1(1;3)$ и $\tilde{A}_1(1;3)$.

$$A_1(1;3) = \left(\begin{array}{c|ccccc} & 1 & 2 & 4 & 5 \\ \hline 2 & 1 & \infty & 0 & 0 \\ 3 & \infty & 4 & 0 & 1 \\ 4 & 2 & 0 & \infty & 2 \\ 5 & 0 & 0 & 2 & \infty \end{array} \right), \quad \tilde{A}_1(1;3) = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 1 & \infty & 2 & 0 \\ 2 & 2 & \infty & 4 & 0 & 0 \\ 3 & 0 & 4 & \infty & 0 & 1 \\ 4 & 3 & 0 & 0 & \infty & 2 \\ 5 & 1 & 0 & 1 & 2 & \infty \end{array} \right).$$

Вычислим константы приведения:

$$\varphi_1(1;3) = \varphi_0 + \Delta_1(1;3) = 45 + 1 = 46, \quad \tilde{\varphi}_1(1;3) = \varphi_0 + \tilde{\Delta}_1(1;3) = 45 + 1 = 46,$$

где $\Delta_1(1;3)$ и $\tilde{\Delta}_1(1;3)$ – суммы минимальных элементов строк и столбцов матриц $A_1(1;3)$ и $\tilde{A}_1(1;3)$.

Так как $\varphi_1(1;3) = \tilde{\varphi}_1(1;3)$, то далее будем рассматривать матрицу $A_1(1;3)$, поскольку ее размерность меньше, чем размерность матрицы $\tilde{A}_1(1;3)$. Определим степени нулей матрицы $A_1(1;3)$:

$$A_1(1;3) = \left(\begin{array}{c|cccc} & 1 & 2 & 4 & 5 \\ \hline 2 & 1 & \infty & 0^{(0)} & 0^{(1)} \\ 3 & \infty & 4 & 0^{(1)} & 1 \\ 4 & 2 & 0^{(2)} & \infty & 2 \\ 5 & 0^{(1)} & 0^{(0)} & 2 & \infty \end{array} \right).$$

Максимальная степень – число 2. Наиболее вероятной дугой гамильтонова цикла является дуга $l(4;2)$.

В связи с этим рассмотрим две матрицы: $A_2(4;2)$ и $\tilde{A}_2(4;2)$. В матрице $A_2(4;2)$ уберем строку под номером четыре и столбец под номером два, элемент a_{24} заменим на ∞ . В матрице $\tilde{A}_2(4;2)$ элемент a_{24} заменим на ∞ . Получим:

$$A_2(4;2) = \left(\begin{array}{c|ccc} & 1 & 4 & 5 \\ \hline 2 & 1 & \infty & 0 \\ 3 & \infty & 0 & 1 \\ 5 & 0 & 2 & \infty \end{array} \right), \quad \tilde{A}_2(4;2) = \left(\begin{array}{c|cccc} & 1 & 2 & 4 & 5 \\ \hline 2 & 1 & \infty & 0 & 0 \\ 3 & \infty & 4 & 0 & 1 \\ 4 & 2 & \infty & \infty & 2 \\ 5 & 0 & 0 & 2 & \infty \end{array} \right).$$

Матрица $A_2(4;2)$ – приведенная. Матрица $\tilde{A}_2(4;2)$ не является приведенной. Определим минимальные элементы строк матрицы $\tilde{A}_2(4;2)$ и вычтем их из элементов соответствующих строк:

$$\tilde{A}_2(4;2) = \left(\begin{array}{c|cccc|c} & 1 & 2 & 4 & 5 & \\ \hline 2 & 1 & \infty & 0 & 0 & 0 \\ 3 & \infty & 4 & 0 & 1 & 0 \\ 4 & 2 & \infty & \infty & 2 & 2 \\ 5 & 0 & 0 & 2 & \infty & 0 \end{array} \right) \Rightarrow \tilde{A}_2(4;2) = \left(\begin{array}{c|cccc} & 1 & 2 & 4 & 5 \\ \hline 2 & 1 & \infty & 0 & 0 \\ 3 & \infty & 4 & 0 & 1 \\ 4 & 0 & \infty & \infty & 0 \\ 5 & 0 & 0 & 2 & \infty \end{array} \right).$$

Вычислим константы приведения:

$$\varphi_2(4;2) = \varphi_1(1;3) + \Delta_2(4;2) = 46 + 0 = 46,$$

$$\tilde{\varphi}_2(4;2) = \varphi_1(1;3) + \tilde{\Delta}_2(4;2) = 46 + 2 = 48.$$

Так как $\varphi_2(4;2) < \tilde{\varphi}_2(4;2)$, то далее будем рассматривать матрицу $A_2(4;2)$, для которой определим степени каждого нуля:

$$A_2(4;2) = \left(\begin{array}{c|ccc} & 1 & 4 & 5 \\ \hline 2 & 1 & \infty & 0^{(2)} \\ 3 & \infty & 0^{(3)} & 1 \\ 5 & 0^{(3)} & 2 & \infty \end{array} \right).$$

Максимальная степень – число 3. Наиболее вероятными дугами гамильтонова цикла являются дуги $l(3;4)$ и $l(5;1)$.

Выберем дугу $l(3;4)$. В связи с этим рассмотрим две матрицы: $A_3(3;4)$ и $\tilde{A}_3(3;4)$. В матрице $A_3(3;4)$ уберем строку под номером три и столбец под номером четыре. В матрице $\tilde{A}_3(3;4)$ элемент a_{34} заменим на ∞ . Получим:

$$A_3(3;4) = \left(\begin{array}{c|cc} & 1 & 5 \\ \hline 2 & 1 & 0 \\ 5 & 0 & \infty \end{array} \right), \quad \tilde{A}_3(3;4) = \left(\begin{array}{c|ccc} & 1 & 4 & 5 \\ \hline 2 & 1 & \infty & 0 \\ 3 & \infty & \infty & 1 \\ 5 & 0 & 2 & \infty \end{array} \right).$$

Матрица $A_3(3;4)$ – приведенная. Матрица $\tilde{A}_3(3;4)$ не является приведенной. Определим минимальные элементы строк и столбцов матрицы $\tilde{A}_3(3;4)$ и выполним приведение:

$$\tilde{A}_3(3;4) = \left(\begin{array}{c|ccc|c} & 1 & 4 & 5 & \\ \hline 2 & 1 & \infty & 0 & 0 \\ 3 & \infty & \infty & 1 & 1 \\ 5 & 0 & 2 & \infty & 0 \\ \hline & 0 & 2 & 0 & \end{array} \right) \Rightarrow \tilde{A}_3(3;4) = \left(\begin{array}{c|ccc} & 1 & 4 & 5 \\ \hline 2 & 1 & \infty & 0 \\ 3 & \infty & \infty & 0 \\ 5 & 0 & 0 & \infty \end{array} \right).$$

Вычислим константы приведения:

$$\varphi_3(3;4) = \varphi_2(4;2) + \Delta_3(3;4) = 46 + 0 = 46,$$

$$\tilde{\varphi}_3(3;4) = \varphi_2(4;2) + \tilde{\Delta}_3(3;4) = 46 + 1 + 2 = 49.$$

Так как $\varphi_3(3;4) < \tilde{\varphi}_3(3;4)$, то рассмотрим матрицу $A_3(3;4)$. Последние две дуги гамильтонова цикла определим из этой матрицы: $l(2;5)$ и $l(5;1)$.

Сравним константу приведения $\varphi_3(3;4)$ с константами приведения $\tilde{\varphi}_k$ альтернативных вариантов ($k = \overline{1,3}$):

$$\varphi_3(3;4) = \tilde{\varphi}_1(1;3): \quad 46 = 46;$$

$$\varphi_3(3;4) < \tilde{\varphi}_2(4;2): \quad 46 < 48;$$

$$\varphi_3(3;4) < \tilde{\varphi}_3(3;4): \quad 46 < 49.$$

Так как $\varphi_3(3;4) \leq \tilde{\varphi}_k$ ($k = \overline{1,3}$), то полученное решение, состоящее из дуг $(1;3)$, $(4;2)$, $(3;4)$, $(2;5)$, $(5;1)$, является оптимальным, но не единственным.

Весь процесс поиска оптимального плана можно изобразить в виде дерева ветвления, рис. 4.4.

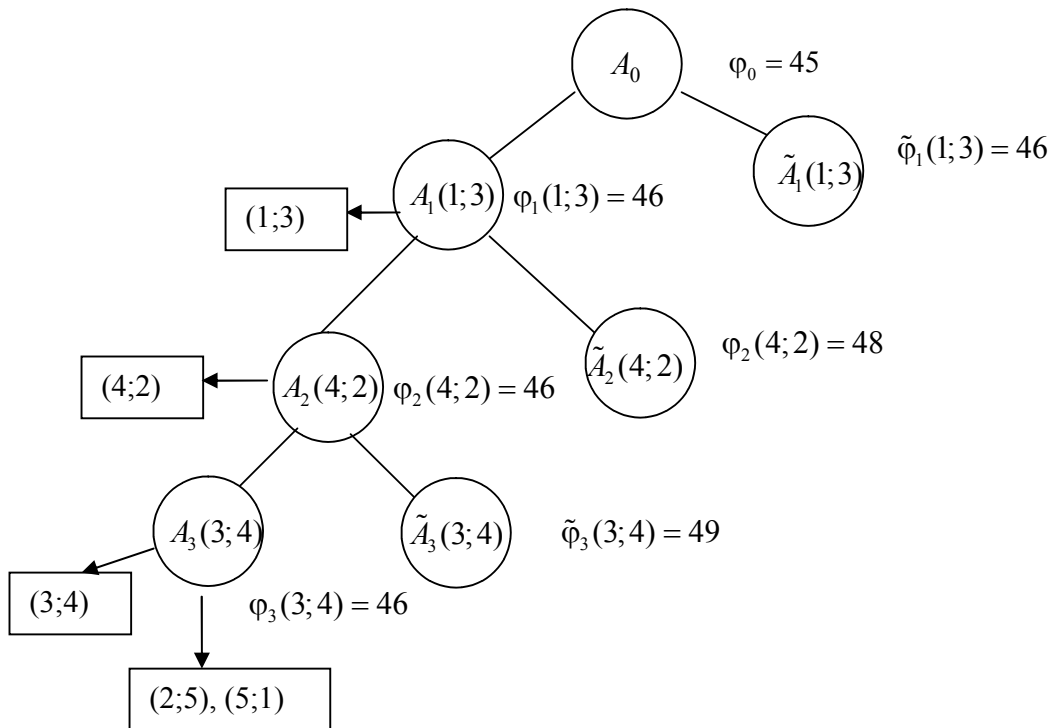
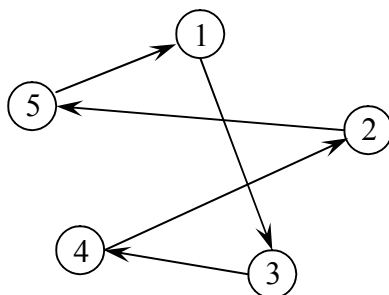


Рис. 4.4

Из полученных дуг составим замкнутый гамильтонов цикл: $1 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 5 \rightarrow 1$, который можно представить в виде орграфа:



Построенный гамильтонов цикл соответствует последовательности перестройки линии с обработки одной детали на другую, что означает следующее: после обработки деталей первого вида следует наладить обработку деталей третьего вида; после третьего – четвертого вида; после четвертого – второго вида; после второго – пятого вида; после пятого – снова вернуться к обработке первого вида деталей. При этом минимальные общие потери рабочего времени составят $\varphi = \varphi_3(3;4) = 46$ часов.

Ответ:

последовательность перестройки линии: $1 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 5 \rightarrow 1$;

минимальные общие потери рабочего времени составят 46 часов. \square

5. ПЛАНАРНЫЕ ГРАФЫ

5.1. Основные определения

Графы, которые рассматривались выше, изображали на плоскости, где вершинами являлись точки плоскости, а ребрами – непрерывные плоские линии. Среди многих прикладных задач из теории графов интерес представляют те, которые можно изобразить на плоскости так, чтобы их ребра не пересекались.

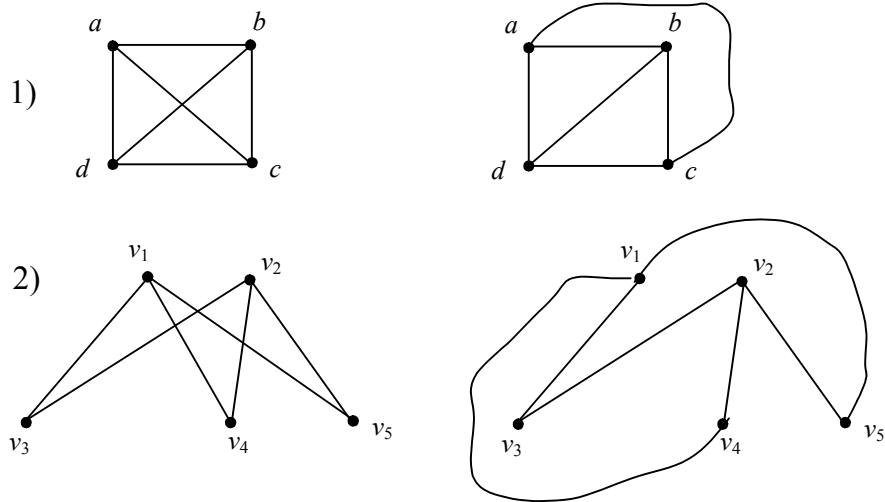
Например, интегральная микросхема состоит из слоев миниатюрных микросхем, впечатанных в пластину. В такой ситуации крайне важно исключить пересечение проводов в местах, не предназначенных для соединений. Если изобразить места указанных соединений вершинами графа, то возникает задача построения графа с непересекающимися ребрами. Аналогичная задача возникает при проектировании железнодорожных и других путей, где нежелательны переезды.

Подчеркнем, что интерес представляет вопрос возможности построения графа с непересекающимися ребрами.

Определение 5.1. *Планарным графом* называется граф, который может быть изображен на плоскости таким образом, что его ребра не пересекаются.

Все планарные графы «укладываются» на плоскости (имеют плоскую укладку).

Пример 5.1. На рисунках изображен граф и его плоская укладка.



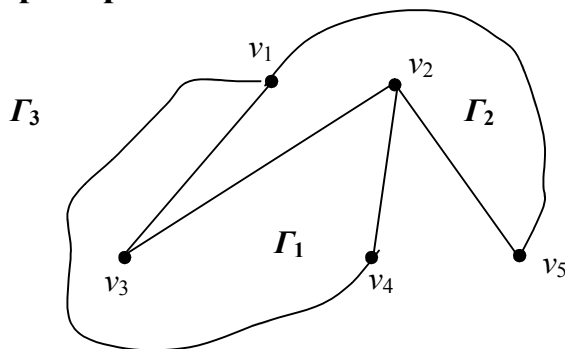
□

Очевидно, что:

- 1) всякий подграф планарного графа планарен.
- 2) граф планарен тогда и только тогда, когда каждая связная компонента этого графа – планарный граф.

Определение 5.2. *Гранью* планарного графа называется множество точек плоскости, любые две из которых могут быть соединены плоской кривой, не пересекающей ребер графа. *Границей* грани называется множество вершин и ребер графа, принадлежащих этой грани.

Пример 5.2.



Γ_1, Γ_2 – внутренние грани
 Γ_3 – внешняя грань

□

Теорема 5.1 (теорема Эйлера). Если G – связный планарный граф, содержащий n вершин, m ребер и f граней, то

$$n - m + f = 2. \quad (5.1)$$

Доказательство.

Пусть G – связный планарный граф, который имеет n вершин. Рассмотрим некоторый остов \tilde{G} этого графа. Остов имеет всего одну внешнюю грань, n вершин и $n-1$ ребер, т.е. $f=1$, $n=n$ и $m=n-1$. Значит, $n - m + f = n - n + 1 + 1 = 2$. Формула (5.1) для остова \tilde{G} выполняется.

Будем поочередно добавлять к остову \tilde{G} недостающие ребра графа G . При каждом добавлении число вершин не изменится, число ребер увеличивается на единицу, так же как и число граней, поскольку при добавлении к остову ребра, связывающего две несмежные вершины, получается цикл, разделяющий текущую грань на две.

Таким образом, формула (5.1) будет верна для всякого графа, получающегося в результате таких операций, а поскольку графом G заканчивается вся эта процедура, то эта формула будет верна и для него. \square

Пример 5.3. Пусть планарный граф имеет вершины со степенями 2, 2, 2, 3, 3, 3, 4, 4, 4 и 5, соответственно. Сколько у него ребер и граней?

Решение. Всего вершин 10, т.е. $n=10$. Найдем сумму степеней вершин графа $\sum_i d(v_i) = 32$. Значит, число ребер равно $m = \frac{32}{2} = 16$ (по теореме Эйлера о сумме степеней вершин графа). Тогда число граней $f = 2 - 10 + 16 = 8$. \square

5.2. Критерии планарности графа

Теорема 5.2. Полный двудольный граф $K_{3,3}$ не является планарным.

Доказательство.

Используем метод «от противного» и предположим, что граф $K_{3,3}$ – планарный.

Если $K_{3,3}$ – планарный граф, и поскольку имеется девять ребер и шесть вершин, то $6 - 9 + f = 2$. Поэтому $f = 5$.

Пусть A и B – непересекающиеся трехэлементные множества вершин, формирующие множество V вершин графа $K_{3,3}$. Если начать путь из одного из непересекающихся множеств, например, A , и не повторять ребра, то можно попасть в вершину из множества B , вернуться в вершину из множества A , вернуться в вершину из множества B и, наконец, вернуться в вершину множества A , прежде чем завершить цикл. Каждый цикл в $K_{3,3}$ представляет собой путь, длина которого, по крайней мере, равна 4. Поэтому каждая грань определена циклом, в котором не менее четырех ребер. Следовательно, сумма ребер всех граней больше, чем $4f$. Но каждое ребро подсчитывается не более двух раз, поскольку оно может служить границей только двух граней. Значит, сумма ребер граней должна быть меньше, чем $2m$. Объединяя эти неравенства, получаем $4f \leq 2m$. Поэтому $4f \leq 18$. Но это противоречит тому, что $f = 5$.

Следовательно, мы пришли к противоречию, и граф $K_{3,3}$ не является планарным. □

Лемма 5.1*. *В произвольном связном планарном графе G с количеством вершин не менее трех имеет место неравенство $3n - m \geq 6$.*

Теорема 5.3. *Полный граф K_5 не является планарным.*

Доказательство.

Граф K_5 имеет пять вершин и десять ребер, поэтому $3n - m = 3 \cdot 5 - 10 = 5$. Согласно лемме 5.1, граф K_5 не является планарным. □

Если граф $G(V, E)$ содержит ребро $e = (v_i, v_j)$ и граф $G'(V', E')$ получен из графа $G(V, E)$ добавлением новой вершины v в множество V и заменой ребра (v_i, v_j) ребрами (v_i, v) и (v, v_j) , то граф $G'(V', E')$ называется *расширением* графа $G(V, E)$. Если графы $G_1, G_2, G_3, \dots, G_n$ таковы, что для любого индекса i граф G_{i+1} является расширением графа G_i , то граф G_n называется *производным* от графа G_1 .

Если граф $G'(V', E')$ – расширение графа $G(V, E)$, то на одном из ребер графа G появляется новая вершина, которая разбивает данное

ребро на два новых ребра, соединяющие вершины, инцидентные исходному ребру.

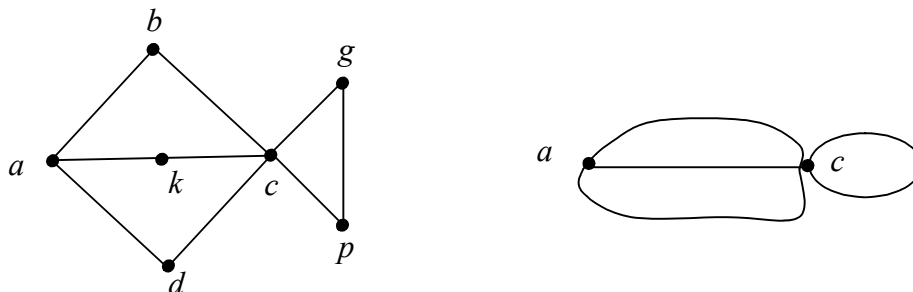
Определение 5.3. Графы G' и G'' называются *гомеоморфными*, если существует граф G такой, что оба графа, G' и G'' , являются производными от графа G .

Пример 5.4. Граф, изображенный слева, является расширением графа, изображенного справа.



□

Пример 5.5. Граф, изображенный слева, является производным от графа, изображенного справа.



□

Теорема 5.4*. *Каждый планарный граф содержит вершину степени 5 или менее.*

Теорема 5.5*. *Если два связных графа гомеоморфные, то они либо оба планарные, либо оба не планарные.*

Теорема 5.6* (Понтрягина – Куратовского).

Граф является планарным тогда и только тогда, когда он не содержит подграф, гомеоморфный $K_{3,3}$ или K_5 .

6. СЕТИ. ПОТОКИ В СЕТЯХ

6.1. Сеть. Поток в сети

Функциональное назначение большинства физически реализованных сетей состоит в том, что они служат носителями систем потоков, т.е. систем, в которых некоторые объекты текут, движутся или транспонируются по системе каналов (дуг сети) ограниченной пропускной способности. Приведем некоторые примеры практических задач.

1. Пусть имеется сеть автомобильных дорог, по которым можно проехать из пункта A в пункт B . Дороги могут пересекаться в промежуточных пунктах. Количество автомобилей, которые могут проехать по каждому отрезку дороги в единицу времени, не безгранично, оно определяется такими факторами, как ширина проезжей части, качество дорожного покрытия, действующие ограничения скорости движения и т.д. (обычно данную характеристику называют «пропускной способностью» дороги). Каково максимальное количество автомобилей, которые могут проехать из пункта A в пункт B без образования пробок на дорогах (эту величину называют «автомобильным потоком»)? Можно также поставить другой вопрос: какие дороги и насколько можно расширить или улучшить, чтобы увеличить максимальный поток на заданную величину?

2. Пусть имеется сеть трубопроводов, соединяющих пункт A (например, нефтепромысел) с пунктом B (нефтеперерабатывающим заводом). Трубопроводы могут соединяться и разветвляться в промежуточных пунктах. Количество нефти, которое может быть перекачано по каждому отрезку трубопровода в единицу времени, также не безгранично и определяется такими факторами, как диаметр трубы, мощность нагнетающего насоса и т.д. (данная характеристика называется «пропускной способностью» или «максимальным расходом» трубопровода). Сколько нефти можно пропускать через такую сеть в единицу времени?

Изучение данных практических задач и многочисленных им подобных приводит к теории потоков в сетях. Данная теория разрабатывает решения общей задачи, которая называется *задача об оптимальном потоке*. Рассмотрим частный случай этой задачи, а именно, *задачу определения максимальной величины потока*.

Определение 6.1. *Сетью* называется связный ориентированный граф $G(V, E)$ без петель с выделенными вершинами I – *истоком* и S – *стоком*, причем каждой дуге поставлено в соответствие некоторое натуральное число $c(v_i, v_j)$ – *пропускная способность дуги*.

Напомним, что истоком называется вершина, у которой $d^+(v) = 0$, т.е. из которой дуги только выходят; а стоком – вершина, у которой $d^-(v) = 0$, т.е. вершина, в которую дуги только входят.

Пропускная способность дуги характеризует максимальное количество вещества, которое может пропустить за единицу времени дуга (v_i, v_j) . На сети пропускную способность дуги будем записывать в круглых скобках.

Поток в сети определяет способ пересылки некоторых объектов из одной вершины графа в другую по направлению дуги. Число объектов (количество вещества) $f(v_i, v_j)$, пересылаемых вдоль дуги (v_i, v_j) , не может превышать пропускной способности $c(v_i, v_j)$ этой дуги:

$$0 \leq f(v_i, v_j) \leq c(v_i, v_j).$$

Будем считать, что если существует дуга из вершины v_i в вершину v_j , то нет дуги из вершины v_j в вершину v_i . Таким образом, рассматривается поток вещества только в одну сторону.

Постановка задачи о максимальном потоке

В сети с заданными пропускными способностями дуг требуется сформировать максимальный по величине поток F_{\max} между ее истоком и стоком. Этот поток обеспечивается назначением в каждой дуге (v_i, v_j) величины $f(v_i, v_j)$ передаваемого ею потока.

Задача о максимальном потоке в сети должна удовлетворять следующим условиям:

– сумма потоков дуг, выходящих из истока сети, должна быть равна сумме потоков дуг, входящих в сток сети:

$$\sum_{(I, v_i) \in E} f(I, v_i) = \sum_{(v_j, S) \in E} f(v_j, S);$$

– для вершины v , не являющейся стоком или истоком, т.е. $v \neq I$, $v \neq S$, количество единиц потока, входящего в вершину, должно быть равно количеству единиц потока, выходящего из нее (т.е. требуется сохранение потока):

$$\sum_{(v_i, v) \in E} f(v_i, v) = \sum_{(v, v_j) \in E} f(v, v_j);$$

– максимальный поток на пути от истока I к стоку S определяется той дугой (v_i, v_j) , которая имеет минимальную пропускную способность из всех дуг, принадлежащих этому пути.

Если пропускная способность $c(v_i, v_j)$ дуги (v_i, v_j) равна идущему через нее потоку $f(v_i, v_j)$ (в сети значение $f(v_i, v_j)$ обозначается числом без скобок), то такая дуга называется *насыщенной*, а любой путь, в который она включена, называется *насыщенным путем*. Поток называется *насыщенным*, если любой путь из истока I в сток S содержит дугу (v_i, v_j) , для которой $f(v_i, v_j) = c(v_i, v_j)$. Первая часть решения задачи о максимальном потоке как раз и состоит в нахождении насыщенного потока. Но насыщенный поток не всегда является максимальным.

По определению, поток в сети будет максимальным, если величина этого потока F_{\max} будет больше величины любого другого потока в этой сети.

6.2. Разрез на сети

Максимальный поток определяется с помощью одного из основных понятий теории сетей – разреза.

Разрез может быть представлен как множество дуг, исключение которых из сети сделало бы оргграф несвязным.

Предположим, что дана некоторая сеть. Разобьем множество вершин V этой сети на два непересекающихся подмножества A и B ($A \cup B = V$, $A \cap B = \emptyset$) так, чтобы исток I попал в подмножество A , а сток

S – в подмножество B , т.е. $I \in A, S \in B$. В этом случае говорят, что на сети произведен разрез, отделяющий исток I от стока S .

Пусть $R(A/B)$ – разрез на сети, представляющий совокупность дуг, которые связывают подмножества вершин A и B . В разрез входят дуги, начальные вершины которых принадлежат подмножеству A , а конечные – подмножеству B , обозначим их через R^+ , т.е. $R^+ = \{(v_i, v_j) \mid v_i \in A, v_j \in B\}$. Также в разрез входят дуги, начальные вершины которых принадлежат подмножеству B , а конечные – подмножеству A , обозначим их через R^- , т.е. $R^- = \{(v_i, v_j) \mid v_i \in B, v_j \in A\}$.

Пропускной способностью или величиной разреза $R(A/B)$ называется величина $C(A/B)$, которая определяется следующей формулой:

$$C(A/B) = C(R^+) - C(R^-) = \sum_{v_i \in A, v_j \in B} c(v_i, v_j) - \sum_{v_i \in B, v_j \in A} c(v_i, v_j). \quad (6.1)$$

Потоком через разрез $R(A/B)$ называется величина $F(A/B)$, которая определяется следующей формулой:

$$F(A/B) = F(R^+) - F(R^-) = \sum_{v_i \in A, v_j \in B} f(v_i, v_j) - \sum_{v_i \in B, v_j \in A} f(v_i, v_j). \quad (6.2)$$

Пример 6.1. Рассмотрим сеть с заданными пропускными способностями дуг, которые записаны в круглых скобках, рис. 6.1.

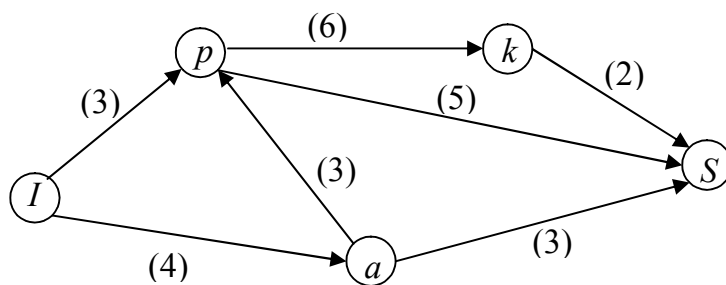


Рис. 6.1

Построим в сети некоторый поток, величину которого по каждой дуге будем записывать без скобок:

путь $I \rightarrow p \rightarrow k \rightarrow S$: $\min\{3; 6; 2\} = 2$, значит, по этому пути пропускаем поток в 2 единицы;

путь $I \rightarrow a \rightarrow S$: $\min\{4; 3\} = 3$, значит, по этому пути пропускаем поток в 3 единицы;

путь $I \rightarrow a \rightarrow p \rightarrow S$: $\min\{4 - 3; 3; 5\} = 1$, значит, по этому пути пропускаем поток в 1 единицу.

Проведем на этой сети разрез $R(A/B)$, при котором вершины разбиты, например, на подмножества $A = \{I; p\}$ и $B = \{k; a; S\}$. Тогда сам разрез, рис. 6.2, состоит из дуг: $(I; a)$, $(a; p)$, $(p; S)$, $(p; k)$, т.е.

$$R(A/B) = \{(I; a), (a; p), (p; S), (p; k)\}.$$

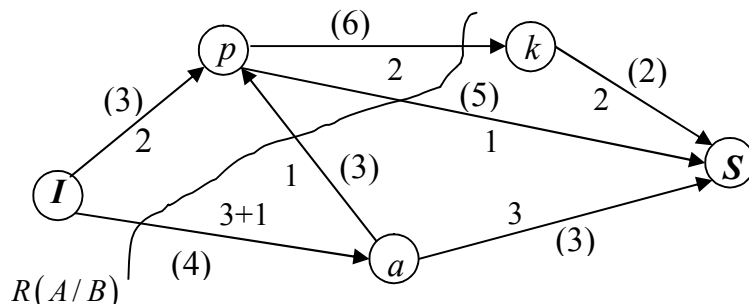


Рис. 6.2

Пропускная способность данного разреза $R(A/B)$ равна

$$C(A/B) = c(I; a) + c(p; S) + c(p; k) - c(a; p) = 4 + 5 + 6 - 3 = 12 \text{ (ед.)},$$

а поток через этот разрез составляет

$$F(A/B) = f(I; a) + f(p; S) + f(p; k) - f(a; p) = 4 + 1 + 2 - 1 = 6 \text{ (ед.)}.$$

□

6.3. Алгоритм нахождения максимального потока в сети

Теорема 6.1 (Форда – Фалкерсона).

Максимальный поток в сети равен минимальной пропускной способности разреза:

$$F_{\max} = C_{\min}(A/B).$$

Доказательство теоремы представляет алгоритм определения максимального потока в сети.

Алгоритм нахождения максимального потока в сети:

Этап 1. *Насыщение потока.*

Шаг 1. Сформировать произвольный начальный поток.

Шаг 2. Найти оставшиеся возможные пути из истока I в сток S , имеющие только ненасыщенные дуги. Если такой путь найден, то переходим к шагу 3. Если путь не найден, то переходим к шагу 4.

Шаг 3. Увеличить поток по найденному пути таким образом, чтобы, по крайней мере, одна из дуг стала насыщенной.

Шаг 4. Получившийся поток насыщен.

Этап 2. *Пометка вершин сети (перераспределение потока).*

Шаг 5. Вершину I пометить $-I$.

Шаг 6. Пусть m – любая из уже помеченных вершин, n – произвольная непомеченная вершина, смежная с вершиной m . Вершину n помечаем $+m$, если данные вершины соединены *ненасыщенной* дугой $m \rightarrow n(+m)$, и помечаем $-m$, если соединены *непустой* дугой $m \leftarrow n(-m)$.

После пометки вершин возможны два случая: вершина S оказалась либо помеченной, либо непомеченной.

Шаг 7. Вершина S оказалась помеченной. Значит, существует последовательность помеченных вершин от истока I к стоку S . В этой последовательности каждая последующая вершина помечена буквой предыдущей вершины. На дугах последовательности определяем новый поток. Увеличиваем на δ единиц поток на дугах, имеющих направление от I к S и уменьшаем на δ единиц поток на дугах, имеющих обратное направление. Число δ равно наименьшей разнице между пропускной способностью и потоком дуг, входящих в последовательность. Заметим, что поток можно увеличивать (уменьшать) на прямых (обратных) дугах до тех пор, пока одна из дуг не станет насыщенной (пустой). Далее вновь переходим к пометке вершин (шаг 5). Перераспределение потока сохраняет все его свойства и увеличивает поток на δ единиц в вершину S .

Этап 3. *Определение максимального потока.*

Шаг 8. Вершина S осталась непомеченной. Пусть A – множество всех помеченных вершин, B – множество всех непомеченных вершин. То-

гда дуги, связывающие два подмножества вершин A и B , определяют разрез $R(A/B)$. Таким образом, найден поток F и разрез $R(A/B)$, для которого выполняется условие $F_{\max} = C_{\min}(A/B)$.

Пример 6.2. В сети из примера 6.1. сформировать максимальный по величине поток, направленный из истока I в сток S . Выписать ребра, образующие на сети разрез минимальной пропускной способности.

Решение.

Этап 1.

Шаг 1. Воспользуемся тем, что в примере 6.1 уже был сформирован некоторый поток на сети, рис. 6.2.

Шаг 2. Путь $I \rightarrow p \rightarrow k \rightarrow S$ содержит насыщенную дугу $(k; S)$. Добавить потока по этому пути больше нельзя. Пути $I \rightarrow a \rightarrow S$, $I \rightarrow a \rightarrow p \rightarrow S$ и $I \rightarrow a \rightarrow p \rightarrow k \rightarrow S$ содержат насыщенные дуги.

Но путь $I \rightarrow p \rightarrow S$ имеет две дуги, которые еще не насыщенные.

Шаг 3. Увеличим поток по найденному пути на величину: $\min\{3-2; 5-1\} = 1$, т.е. на 1 единицу. В результате – дуга $(I; p)$ стала насыщенной, рис. 6.3.

Шаг 4. Таким образом, получаем насыщенный поток, поскольку каждый рассмотренный путь содержит хотя бы одну насыщенную дугу.

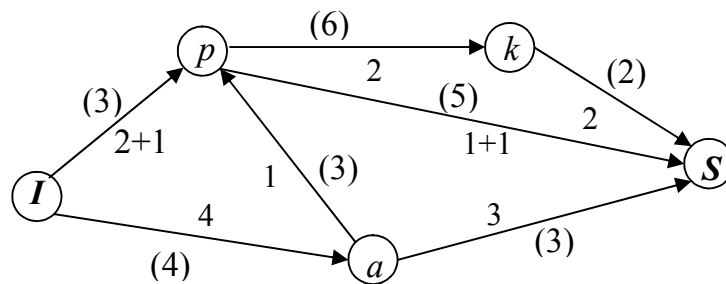


Рис. 6.3

Этап 2.

Выясним, является ли построенный поток максимальным по величине. Строим сеть, на которой отметим все вершины и ненасыщенные дуги. На этой сети разность пропускных способностей дуги и проходящего по ней потока обозначим числом в квадратных скобках.

Шаг 5, 6. Вершину I пометим $-I$. На шаге 6 предусматривается пометка вершин, смежных с вершиной I , соединенных ненасыщенными дугами. Но на построенной сети таких вершин нет.

В результате вершина S оказалась непомеченной, рис. 6.4.

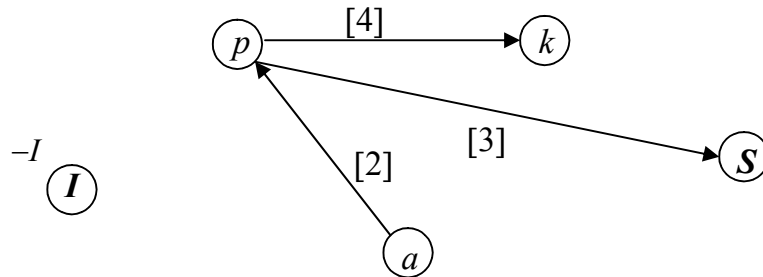


Рис. 6.4

Этап 3.

Шаг 7. Так как вершина S непомеченная, то поток, сформированный в сети, получился максимальным.

Строим разрез на сети. Разбиваем множество вершин на два подмножества: A и B . Так как только одна вершина оказалась помеченной, то множество A состоит из одной вершины – истока I , а остальные вершины образуют множество B :

$$A = \{I\}, B = \{a; p; k; S\}.$$

Проводим разрез $R(A/B)$, который состоит из дуг $(I; p)$ и $(I; a)$, рис. 6.5:

$$R(A/B) = \{(I; p), (I; a)\}.$$

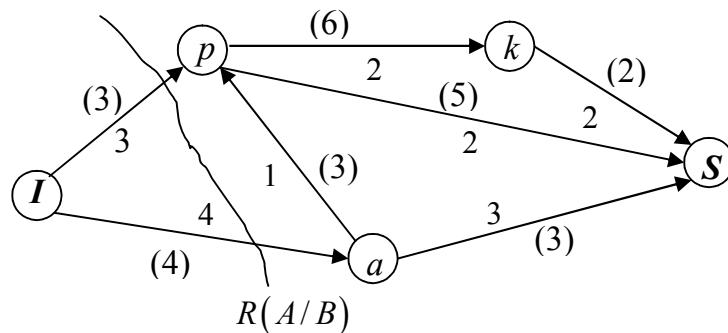


Рис. 6.5

Определим величину максимального потока F_{\max} :

$$F_{\max} = C_{\min}(A/B) = 3 + 4 = 7 \text{ (ед.)}.$$

□

Пример 6.3. На заданной сети в скобках указаны пропускные способности дуг, рис. 6.6. Требуется сформировать в сети максимальный поток, направленный из истока I в сток S , и выписать ребра, образующие на сети разрез минимальной пропускной способности.

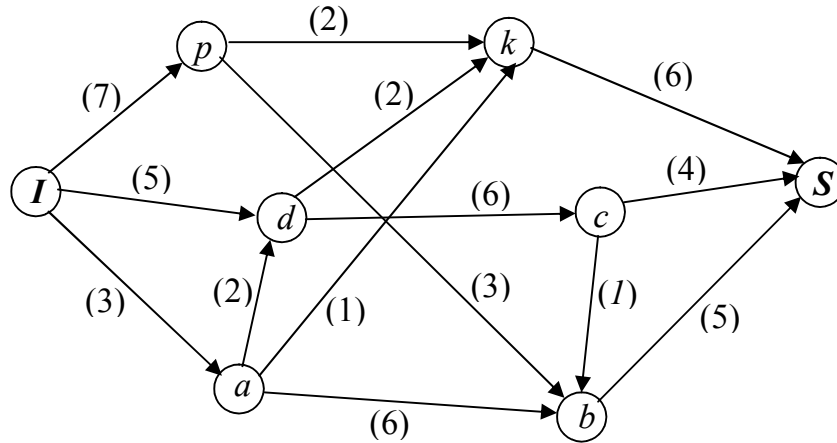


Рис. 6.6

Решение.

Этап 1.

Сформируем в сети начальный поток. Рассмотрим путь $I \rightarrow p \rightarrow k \rightarrow S$. Поскольку $\min\{7; 2; 6\} = 2$, по этому пути пропускаем поток в 2 единицы. В сети значение потока обозначим числами без скобок.

По пути $I \rightarrow d \rightarrow c \rightarrow S$ пропускаем поток в 4 единицы, так как $\min\{5; 6; 4\} = 4$. По пути $I \rightarrow a \rightarrow b \rightarrow S$ пропускаем поток в 3 единицы, так как $\min\{3; 6; 5\} = 3$. Таким образом, начальный поток имеет вид:

$$I \xrightarrow{2} p \xrightarrow{2} k \xrightarrow{2} S,$$

$$I \xrightarrow{4} d \xrightarrow{4} c \xrightarrow{4} S,$$

$$I \xrightarrow{3} a \xrightarrow{3} b \xrightarrow{3} S.$$

Начальный поток изображен на рис. 6.7.

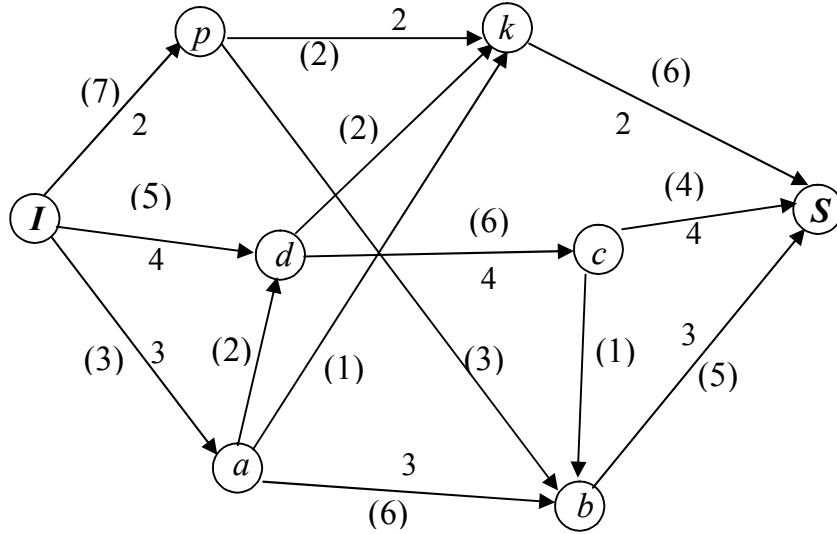


Рис. 6.7

Каждый из рассмотренных путей содержит насыщенную дугу, поэтому эти пути насыщенные. На сети есть еще пути, которые содержат ненасыщенные дуги, а именно: $I \rightarrow p \rightarrow b \rightarrow S$, $I \rightarrow d \rightarrow k \rightarrow S$ и $I \rightarrow d \rightarrow c \rightarrow b \rightarrow S$. Для первого пути дополнительно увеличим поток на 2 единицы, так как $\min\{7-2; 3; 5-3\} = 2$. Второму и третьему пути содержат одну и ту же дугу $(I; d)$ с минимальной для них оставшейся разностью $q(I; d) = 1$. Поэтому для увеличения потока на 1 единицу выберем, например, путь $I \rightarrow d \rightarrow k \rightarrow S$.

Теперь каждый из этих путей содержит насыщенную дугу, следовательно, полученный поток – насыщенный, рис. 6.8.

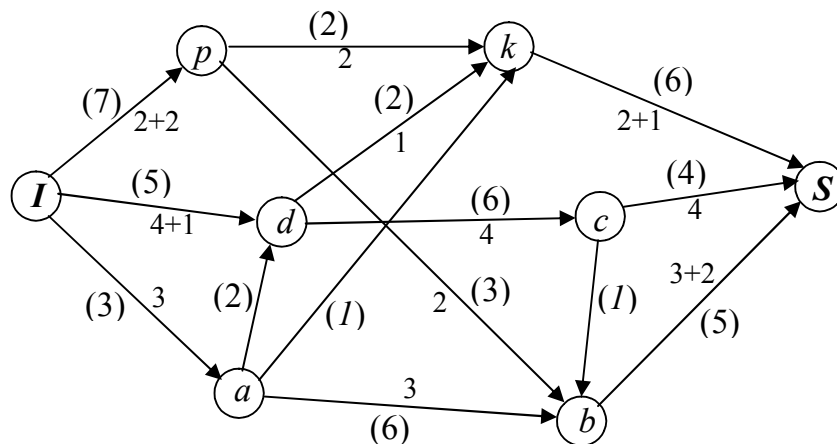


Рис. 6.8

Выясним, является ли построенный поток максимальным. Изобразим сеть, на которой отметим все вершины и ненасыщенные дуги. На этой сети разность пропускной способности дуги и проходящего по ней потока обозначим числом в квадратных скобках. Пропускную способность дуги, по которой поток не проходит, оставим в круглых скобках, рис. 6.9.

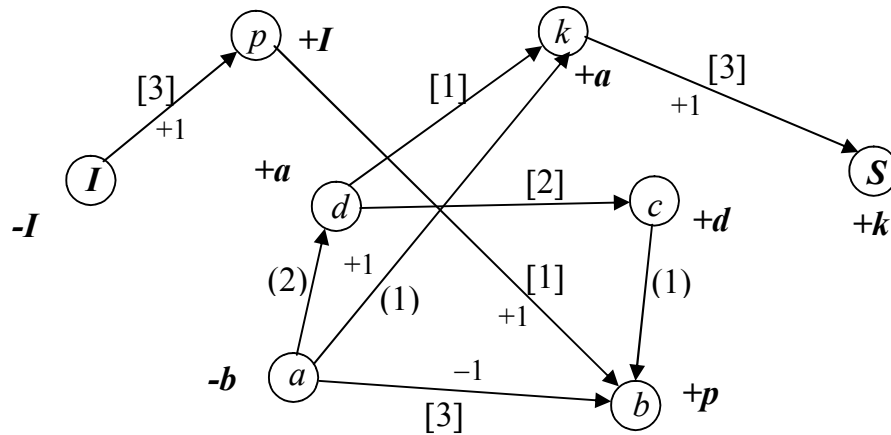


Рис. 6.9

Согласно рис. 6.9, на сети исток I и сток S связаны дугами. Значит, можно добавить какое-то количество потока по ненасыщенным дугам, при этом придется перераспределить поток.

Этап 2.

На построенной сети помечаем вершины. Вершину I помечим $-I$. Смежную ей вершину p помечим $+I$, так как эти вершины соединяет ненасыщенная дуга $I \rightarrow p$. Вершину b помечаем $+p$, так как вершины p и b соединяет ненасыщенная дуга $p \rightarrow b$. Вершину a помечаем $-b$, так как вершины b и a соединяет непустая дуга $b \leftarrow a$. Вершины d и k помечаем $+a$, так как они соединены с вершиной a ненасыщенными дугами $a \rightarrow d$ и $a \rightarrow k$. Вершина c смежна вершине d , и эти вершины соединены ненасыщенной дугой $d \rightarrow c$, поэтому вершину c помечаем $+d$. Вершина S смежна вершине k , и эти вершины соединены ненасыщенной дугой $k \rightarrow S$, поэтому вершину S помечаем $+k$.

Вершина S оказалась помеченной. Значит, существует последовательность помеченных вершин от I к S : $I \rightarrow p \rightarrow b \leftarrow a \rightarrow k \rightarrow S$. В этой последовательности каждая последующая вершина помечена буквой предыдущей вершины. Перераспределим поток на этом пути. Определим число δ : $\delta = \min\{7 - 4; 3 - 2; 6 - 3; 1; 6 - 3\} = \min\{3; 1; 3; 1; 3\} = 1$. Увеличим на 1

единицу поток на дугах, имеющих направление от I к S : $(I;p)$, $(p;b)$, $(a;k)$, $(k;S)$. Уменьшим на 1 единицу поток на дугах, имеющих обратное направление: $(a;b)$. Получим следующую сеть с новым сформированным потоком, который изображен на рисунке 6.10.

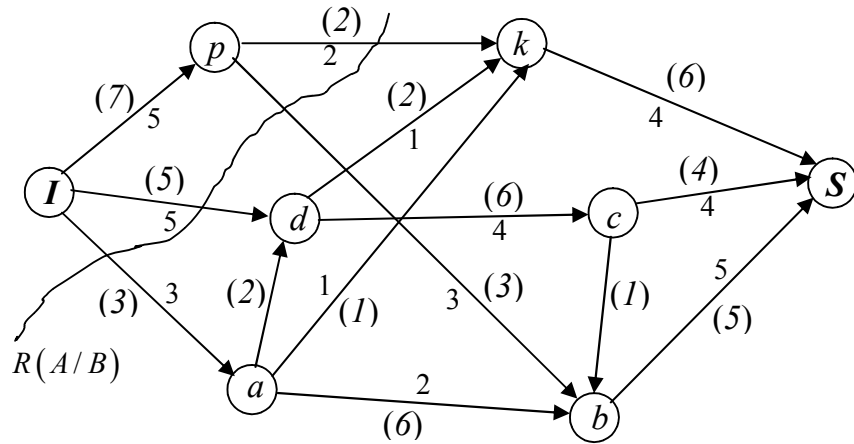


Рис. 6.10

Проверим, будет ли построенный поток максимальным. Изобразим сеть, на которой отметим все вершины и ненасыщенные дуги, рис. 6.11.

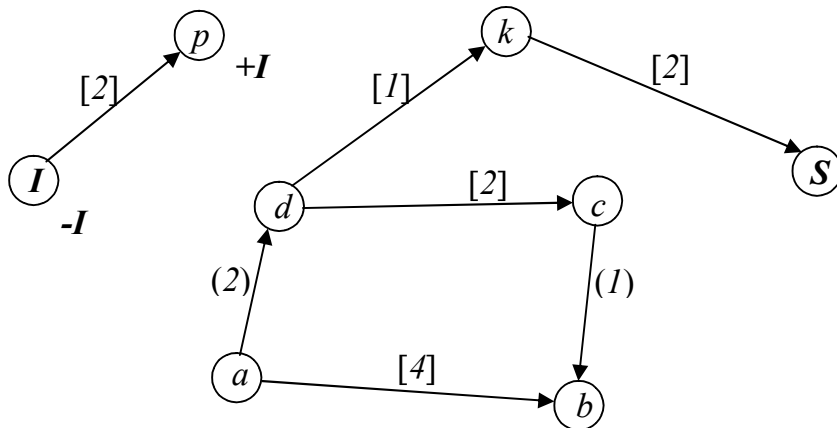


Рис. 6.11

Вновь помечим вершины. Вершину I помечим $-I$. Смежную ей вершину p помечим $+I$, так как эти вершины соединяет ненасыщенная дуга $I \rightarrow p$. Все остальные вершины, в том числе и вершина S , остаются непомяченными. Значит, поток в сети максимальный.

Этап 3.

Согласно рис. 6.11, на последней сети исток I и сток S не связаны дугами. Значит поток, изображенный на рис. 6.10, является максимальным.

Строим разрез на сети. Разбиваем множество вершин на два подмножества: A и B . Помеченные вершины образуют множество $A = \{I; p\}$, непомеченные – множество $B = \{a; d; k; c; b; S\}$.

Проводим разрез $R(A/B)$, который состоит из дуг $(p;k)$, $(I;d)$, $(I;a)$, $(p;b)$:

$$R(A/B) = \{(I; a), (I; d), (p; k), (p; b)\}.$$

Определим величину максимального потока F_{\max} :

$$F_{\max} = C_{\min}(A/B) = 2 + 3 + 5 + 3 = 13 \text{ (ед.)}.$$

Ответ: мощность максимального потока составляет 13 ед., разрез минимальной пропускной способности образуют ребра $(I; a)$, $(I; d)$, $(p; k)$, $(p; b)$.

□

7. ПАРСОЧЕТАНИЕ

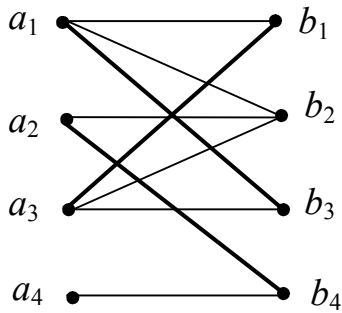
7.1. Основные понятия

Теория потоков в сети имеет непосредственное применение в теории графов, в частности, используется в двудольных графах.

Напомним, что граф $G = G(V, E)$ называется двудольным, если множество вершин V можно представить как объединение непересекающихся множеств, скажем $V = A \cup B$, таким образом, что каждое ребро имеет вид (v_i, v_j) , где $v_i \in A$ и $v_j \in B$. Не существует двух вершин из одного и того же множества (A или B), которые были бы соединены ребром друг с другом.

Определение 7.1. Паросочетанием в неориентированном графе $G = G(V, E)$ называется произвольное множество ребер $M \subseteq E$ такое, что никакие два ребра из M не инцидентны одной вершине.

Пример 7.1. На рисунке изображен двудольный граф.



Выделенные ребра образуют паросочетание.

$M = \{(a_1; b_3), (a_2; b_4), (a_3; b_1)\}$ – паросочетание.

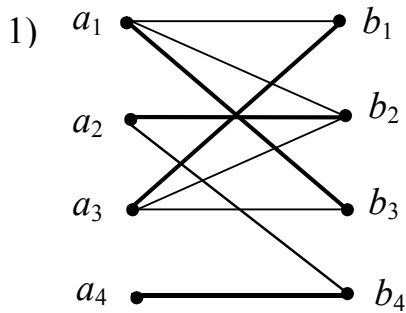
□

С практической точки зрения целесообразно рассматривать паросочетания, которые включают наибольшее число ребер. В связи с этим введем следующие понятия.

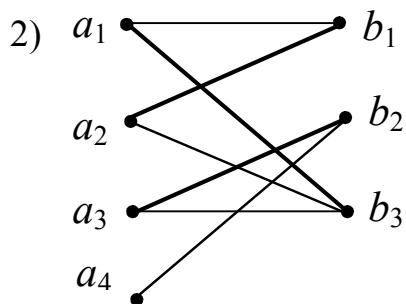
Определение 7.2. Паросочетание M на двудольном графе $G = G(V, E)$ называется *максимальным*, если никакое другое паросочетание на G не содержит ребер больше, чем M .

Определение 7.3. Паросочетание M на двудольном графе $G = G(V, E)$, где $V = A \cup B$, называется *полным*, если для каждой вершины $v_i \in A$ существует вершина $v_j \in B$ такая, что $(v_i, v_j) \in M$.

Пример 7.2. На рисунке изображен двудольный граф. Выделенные ребра образуют паросочетание.



$M = \{(a_1; b_3), (a_2; b_2), (a_3; b_1), (a_4; b_4)\}$
– максимальное и полное паросочетание.



$M = \{(a_1; b_3), (a_2; b_1), (a_3; b_2)\}$
– максимальное, но не полное паросочетание.

□

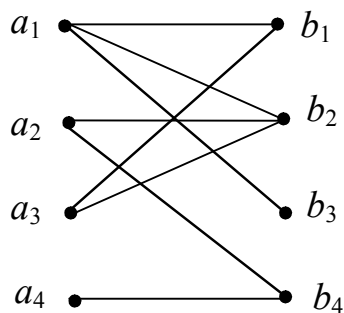
Очевидно, нужен некоторый метод, позволяющий находить максимальное паросочетание в двудольном графе.

Задачу нахождения максимального паросочетания в двудольном графе можно свести к нахождению максимального потока в некоторой сети. Для упрощения заменим рассматриваемый двудольный неориентированный граф ориентированным графом, дуги которого имеют начало в вершинах множества A и заканчиваются в вершинах множества B . Добавим две новые вершины I и S , а также дуги из вершины I в каждую вершину множества A и дуги из каждой вершины множества B в вершину S .

Теперь двудольный граф превращен в сеть. Такая сеть называется *сетью паросочетаний*. Для каждой $v_i \in A$ обозначим через e_i дугу из I в v_i . Положим пропускную способность $c(e_i) = 1$ для каждой дуги e_i . Аналогично, для каждой вершины $v_j \in B$ обозначим через e_j дугу из v_j в S и положим пропускную способность $c(e_j) = 1$ для каждой дуги e_j . Для каждой дуги e_{ij} из v_i в v_j положим $c(e_{ij}) = |A| + 1$ так, чтобы пропускная способность была бы больше, чем количество вершин в множестве A . Так как поток в каждую вершину v_i не может превысить 1, то поток через каждую дугу e_{ij} из v_i в v_j должен быть 1 или 0 независимо от пропускной способности $c(e_{ij})$. Для дуги e_{ij} из v_i в v_j определим поток $f(e_{ij})$ равным 1, если имеется паросочетывающая дуга между v_i и v_j , и равным 0 в противном случае. Величина потока F_{\max} в таком случае равна количеству дуг в паросочетании между A и B . Максимальный поток имеет место, когда паросочетание максимально. Таким образом, задача нахождения максимального паросочетания сведена к задаче построения максимального потока.

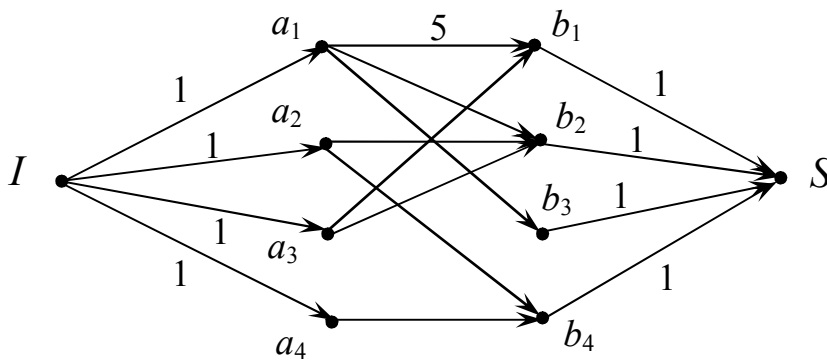
Покажем на примере, как с помощью алгоритма построения максимального потока в сети можно найти максимальное паросочетание в двудольном графе.

Пример 7.3. На рисунке изображен двудольный граф.

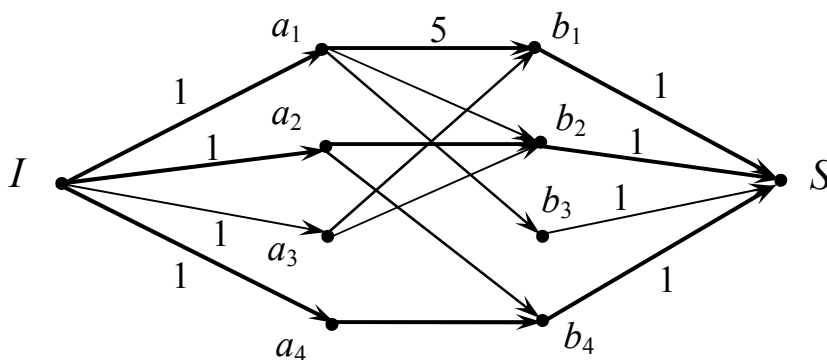


Найти максимальное паросочетание на данном графе, используя алгоритм построения максимального потока в сети.

Решение. Данный двудольный неориентированный граф заменим ориентированным, направив дуги (a_i, b_j) из вершины a_i в вершину b_j , где $a_i \in A (i = \overline{1, 4})$, $b_j \in B (j = \overline{1, 4})$. Дополним двумя вершинами I и S ; дугами (I, a_i) , (b_j, S) , которым припишем пропускные способности $c(I, a_i) = 1$, $c(b_j, S) = 1$. Так как число вершин, входящих в множество A , равно 4, то дугам (a_i, b_j) припишем пропускные способности $c(a_i, b_j) = |A| + 1 = 4 + 1 = 5$. Поскольку каждой дуге (a_i, b_j) приписываем число 5, то на рисунке, из-за избыточности обозначений, припишем это число только дуге (a_1, b_1) . Таким образом, получаем следующую сеть паросочетаний.



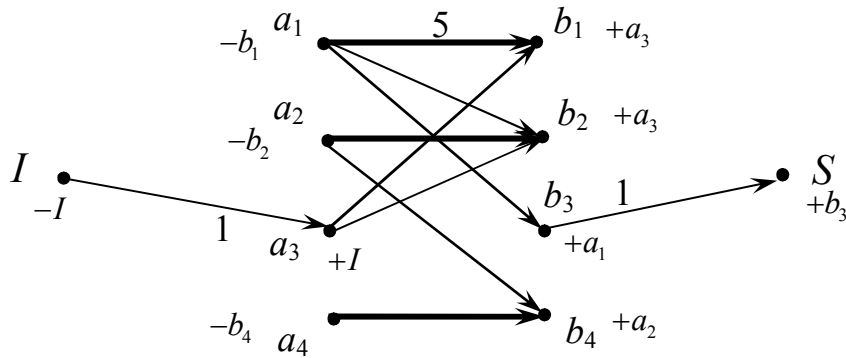
В полученной сети формируем некоторый первоначальный поток. На пути $I \rightarrow a_1 \rightarrow b_1 \rightarrow S$ пропускаем одну единицу. Аналогично по каждому пути $I \rightarrow a_2 \rightarrow b_2 \rightarrow S$ и $I \rightarrow a_4 \rightarrow b_4 \rightarrow S$ пропускаем по одной единице. На сети выделим жирной линией те дуги рассмотренных путей, по которым пустили поток.



Сформированный поток в сети позволяет записать первоначальное паросочетание $M_1 = \{(a_1; b_1), (a_2; b_2), (a_4; b_4)\}$.

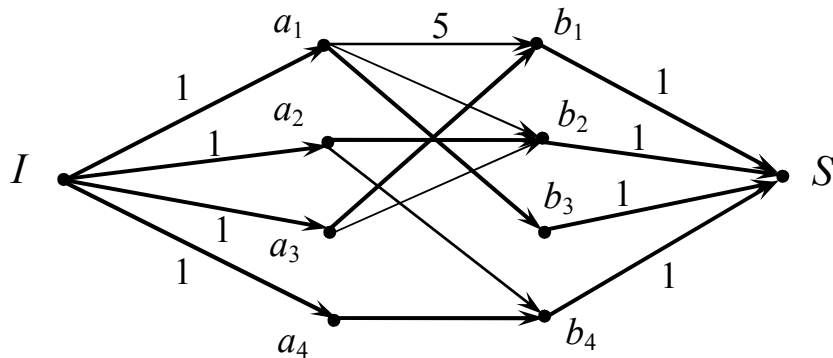
Выясним, является ли это паросочетание максимальным. Изобразим сеть, удалив насыщенные дуги $(I, a_1), (I, a_2), (I, a_4), (b_1, S), (b_2, S), (b_4, S)$. Надо отметить, что по дугам $(a_1, b_1), (a_2, b_2), (a_4, b_4)$ мы пропустили по одной единице потока, и больше единиц потока пускать нельзя. По этим дугам мы можем только вычитать одну единицу потока при необходимом перераспределении.

Вершины помечаем таким же образом, как в алгоритме о формировании максимального потока. В результате получаем следующую сеть.



По данной сети можно определить путь из I в S : $I \rightarrow a_3 \rightarrow b_1 \leftarrow a_1 \rightarrow b_3 \rightarrow S$. Это значит, что поток следует перераспределить.

По дугам $(I, a_3), (a_3, b_1), (a_1, b_3), (b_3, S)$ этого пути мы пропускаем одну единицу, а от потока по дуге (a_1, b_1) вычитаем одну единицу. В результате получаем следующую сеть.



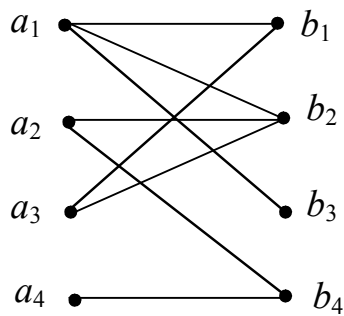
По каждой дуге $(I, a_i), (b_j, S)$ пропустили одну единицу потока. Значит, в сети сформирован поток максимальной мощности $F_{\max} = 4$. Можно сделать вывод, что максимальное паросочетание построено.

Ответ: $M_2 = \{(a_1, b_3), (a_2, b_2), (a_3, b_1), (a_4, b_4)\}$. □

Для нахождения максимального паросочетания можно использовать другой метод – *метод построения чередующейся цепи*. При использовании этого метода нет необходимости заботиться о вершинах I и S . Суть метода в том, что цепь в ориентированном двудольном графе строится, начиная из вершины, принадлежащей множеству A , которая не паросочетается, и движение продолжается назад и вперед, пока цепь не закончится в вершине множества B , у которой нет паросочетанной дуги.

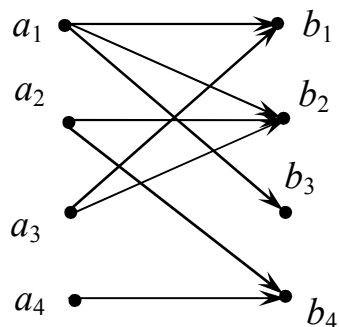
Рассмотрим применение метода построения чередующейся цепи на примере двудольного графа из 7.3.

Пример 7.3 (продолжение). На рисунке изображен двудольный граф.



Найти максимальное паросочетание на данном графе, используя метод чередующейся цепи.

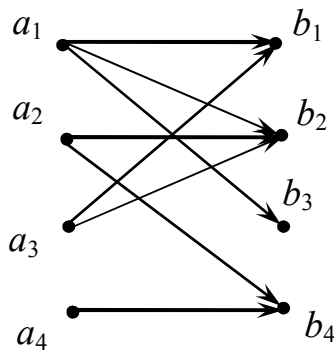
Решение. Данный двудольный неориентированный граф заменим ориентированным, направив дуги (a_i, b_j) из вершины a_i в вершину b_j , где $a_i \in A$ ($i = \overline{1, 4}$), $b_j \in B$ ($j = \overline{1, 4}$). Получаем следующий орграф:



Строим чередующуюся цепь, начиная, например, с вершины a_4 , так как она имеет одну инцидентную дугу:

$$a_4 \rightarrow b_4 \leftarrow a_2 \rightarrow b_2 \leftarrow a_1 \rightarrow b_1 \leftarrow a_3.$$

Движение закончено в вершине a_3 . При дальнейшем движении по дуге (a_3, b_2) попадем в вершину b_2 , которая инцидентна двум дугам (a_2, b_2) и (a_1, b_2) , вошедшим в чередующуюся цепь. В результате получаем ориентированный двудольный граф с выделенными дугами, которые вошли в паросочетание $M_1 = \{(a_1, b_1), (a_2, b_2), (a_4, b_4)\}$.



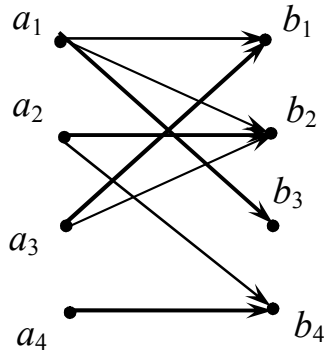
Паросочетание будет максимальным, если чередующаяся цепь закончится в вершине из множества B . Проанализируем построенную цепь. При движении из вершины a_2 по дуге (a_2, b_2) попали в вершину b_2 , после чего движение продолжили в противоположном направлении по дуге (a_1, b_2) . Начиная с вершины b_2 , изменим порядок движения по дугам при построении чередующейся цепи. Вместо дуги (a_1, b_2) включим в цепь дугу (a_3, b_2) , после чего продолжим дальнейшее построение чередующейся цепи. В результате получаем следующую чередующуюся цепь:

$$a_4 \rightarrow b_4 \leftarrow a_2 \rightarrow b_2 \leftarrow a_3 \rightarrow b_1 \leftarrow a_1 \rightarrow b_3.$$

Движение заканчивается в вершине b_3 , т.е. в вершине из множества B . Причем данная чередующаяся цепь позволяет получить следующее паросочетание, которое является максимальным:

$$M_2 = \{(a_1, b_3), (a_2, b_2), (a_3, b_1), (a_4, b_4)\}.$$

Изобразим ориентированный двудольный граф с выделенными дугами, которые вошли в паросочетание M_2 .



□

7.2. Задача о назначениях

Существует ряд прикладных задач, постановка и решение которых сводится к построению максимального паросочетания. Примером такой задачи является *задача о назначениях*.

Постановка задачи о назначениях.

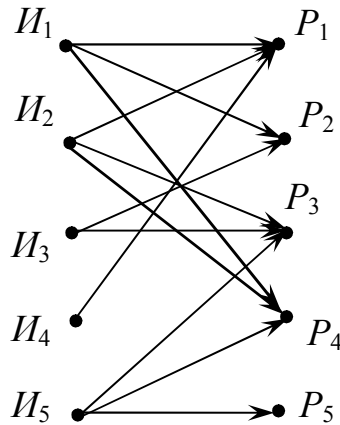
На n видов работ $P = \{P_1, P_2, \dots, P_n\}$ можно назначить n исполнителей $I = \{I_1, I_2, \dots, I_n\}$. Квалификация каждого работника позволяет ему выполнять лишь определенные виды работ. Можно ли распределить работы между исполнителями таким образом, чтобы одновременно выполнялось как можно большее число работ?

Рассмотрим решение данной задачи на примере.

Пример 7.4. Проект включает работы P_1, P_2, P_3, P_4, P_5 , которые могут выполняться независимо друг от друга. Исполнители I_1, I_2, I_3, I_4, I_5 могут выполнять не любые, а лишь вполне определенные работы. Возможности исполнителей характеризуются элементами матрицы $\Omega = [\omega_{ij}]$: если $\omega_{ij} = 1$, то исполнитель I_i может выполнить работу P_j , если же $\omega_{ij} = 0$, то – нет. При этом каждый исполнитель одновременно может выполнять только одну работу, и каждая работа одновременно может выполняться только одним исполнителем. Распределить работы между исполнителями следует так, чтобы одновременно выполнялось как можно большее число работ.

$$\Omega = [\omega_{ij}] = \begin{pmatrix} & P_1 & P_2 & P_3 & P_4 & P_5 \\ I_1 & 1 & 1 & 0 & 1 & 0 \\ I_2 & 1 & 0 & 1 & 1 & 0 \\ I_3 & 0 & 1 & 1 & 0 & 0 \\ I_4 & 1 & 0 & 0 & 0 & 0 \\ I_5 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

Решение. Строим ориентированный двудольный граф, у которого множество вершин разобьем на два множества $I = \{I_1, I_2, I_3, I_4, I_5\}$ и $P = \{P_1, P_2, P_3, P_4, P_5\}$. Вершину I_i соединяем с вершиной P_j дугой (I_i, P_j) , если элемент матрицы $\omega_{ij} = 1$, в противном случае – нет.



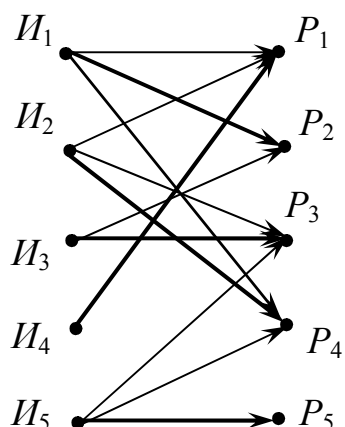
Построение чередующейся цепи лучше начинать с вершины из множества I , которая имеет наименьшее число инцидентных дуг. В нашем случае это вершина I_4 . Строим чередующуюся цепь, начиная с вершины I_4 . Получим

$$I_4 \rightarrow P_1 \leftarrow I_1 \rightarrow P_2 \leftarrow I_3 \rightarrow P_3 \leftarrow I_2 \rightarrow P_4 \leftarrow I_5 \rightarrow P_5.$$

Чередующаяся цепь проходит через каждую вершину множества I и каждую вершину множества P по одному разу. Следовательно, максимальное паросочетание построено:

$$M = \{(I_1, P_2), (I_2, P_4), (I_3, P_3), (I_4, P_1), (I_5, P_5)\}.$$

Данное паросочетание на графе изображено выделенными дугами.



Вывод. Построенное паросочетание свидетельствует о том, что исполнителя I_1 следует назначить на выполнение работы P_2 ; исполнителя I_2 следует назначить на выполнение работы P_4 ; исполнителя I_3 – на выполнение работы P_3 ; исполнителя I_4 – на выполнение работы P_1 ; исполнителя I_5 – на выполнение работы P_5 .

□

8. ЭЛЕМЕНТЫ СЕТЕВОГО ПЛАНИРОВАНИЯ

8.1. Сетевой график. Работа. Событие

Теория графов нашла широкое применение для решения многих классов прикладных задач в различных областях экономики, производства, психологии и т.д.

При планировании сложных комплексов взаимосвязанных и взаимообусловленных работ и управления ходом их выполнения наиболее эффективными оказались *методы сетевого планирования и управления* (СПУ). Доступность и простота этих методов позволяют широко использовать их в практической работе.

Существует несколько методов сетевого планирования. Так, метод *СРМ* (*Critical Path Method* – метод критического пути) используется для контроля выполнения проекта, когда оценки времени операций детерминированы (определены). Таким проектом может быть разработка какого-либо производственного процесса; строительство предприятия, здания или сооружения; ремонт сложного оборудования и т.д.

Метод *PERT* (*Program Evaluation and Review Technique* – метод оценки обзора программ) применяется для контроля сроков выполнения проекта, для которых продолжительность выполнения всех или некоторых работ не удастся определить точно. Он применяется при проектировании и внедрении новых систем, планировании научно-исследовательских и опытно-конструкторских разработок.

В основе методов СПУ лежит графическое представление проекта (комплекса работ для достижения поставленной цели) в виде сетевого графика (орграфа). С математической точки зрения *сетевой график* – это связный орграф без петель и циклов. Будем отождествлять вершины орграфа с событиями, а дуги – с работами.

Работа – это любой процесс, происходящий во времени. Все работы можно разделить на действительные работы, ожидания и фиктивные работы (зависимости). Под действительной работой понимают любой трудовой процесс, сопровождающийся затратами ресурсов и приводящий к определенным результатам. Ожидание – это некоторый процесс, не требующий ресурсов, но имеющий определенную продолжительность (затверждение бетона, сушка штукатурки и т.п.). Фиктивные работы не требуют ресурсов и имеют нулевую продолжительность, используются для обозначения логических зависимостей между действительными работами. На сетевых графиках работы изображают отрезками прямых линий с указанием направления, т.е. дугами, при этом фиктивные работы изображают пунктирными стрелками.

Событие обозначает факт (результат) окончания всех работ, в него входящих, или начала работ, из него выходящих. Событие не имеет продолжительности во времени и не потребляет ресурсов. На каждом сетевом графике можно выделить исходное, промежуточное и завершающее события. Событие, с которого начинается выполнение проекта, является исходным (*I*), оно не имеет предшествующих работ. Событие, констатирующее факт завершения проекта, называется завершающим (*S*), оно не имеет последующих работ. Все прочие события являются промежуточными.

Для изображения события на сетевом графике будем использовать геометрические фигуры (круги, квадраты), которые представляют собой вершины орграфа. Будем обозначать события через x_n или просто нумеровать их в последовательности свершения.

Работы будем обозначать через a_n или (i, j) , где i – начальная вершина дуги (начальное событие данной работы) и j – конечная вершина

(конечное событие данной работы). Рядом со стрелкой в случае необходимости будут указаны числовые характеристики работы: время выполнения работы, расход ресурса, количество исполнителей и т.д.

В каждое событие может входить и выходить из него несколько работ, а каждая работа ограничена двумя событиями. Событие выражает логическую связь между работами, заключающуюся в том, что работы, входящие в данное событие, непосредственно предшествуют работам, выходящим из него; ни одна выходящая из данного события работа не может начинаться до окончания всех работ, входящих в это событие.

Прежде чем представить проект сетевым графиком, необходимо составить перечень работ, оценить продолжительность каждой работы и установить последовательность работ, т.е. точно определить, какие работы обязательно должны быть закончены, чтобы могла начаться любая из работ, входящих в проект. Такой перечень удобно представить в виде структурно-временной таблицы.

Правила, используемые при построении сетевого графика

1. В сетевых графиках не должно быть событий (кроме исходящего), которым не предшествует хотя бы одна работа.

2. В сетевых графиках не должно быть «тупиков», т.е. событий, из которых не выходит ни одна работа (за исключением завершающего события).

3. При построении сетевых графиков нельзя допускать, чтобы два смежных события были связаны двумя или большим числом работ, что чаще всего бывает при изображении параллельно выполняемых работ. Эта ошибка приводит к идентификации этих работ, т.е. две различные работы будут иметь одно и то же обозначение. Чтобы избежать этого, рекомендуется ввести дополнительные события и связать их с последующими фиктивными работами.

4. В сети не должно быть замкнутых циклов, т.е. цепей, соединяющих некоторые события с ними же самими.

5. Если какие-либо сложные работы могут быть начаты до полного окончания непосредственно предшествующей им работы, то последняя изображается как ряд последовательно выполняемых работ, каждая из которых завершается определенным событием.

6. Если для выполнения одной из работ необходимо получение результатов всех работ, входящих в предшествующее для нее событие, а для другой работы достаточно получить результат только одной или несколь-

ких из этих работ, то должно быть дополнительно введено новое событие, а также фиктивная работа, связывающая новое событие с прежним.

Построенный с соблюдением этих правил график является сетевой моделью выполнения проекта. При этом сначала, как правило, составляются частные сетевые графики, охватывающие работы по отдельным, имеющим самостоятельное значение частям общего комплекса работ, а затем путем «сшивания» получается комплексный (сводный) график, охватывающий всю совокупность работ, подлежащих выполнению.

Пример 8.1. Проект включает в себя следующие работы, представленные в таблице 8.1. Построить сетевой график выполнения комплекса работ.

Таблица 8.1

Работа	Предшествующие ей работы	Продолжительность (дни)
a_1	–	2
a_2	–	4
a_3	a_1	3
a_4	a_1, a_2	2
a_5	a_4	5
a_6	a_4	7
a_7	a_3, a_5	3

Решение.

Работам a_1 и a_2 не предшествуют никакие работы, следовательно, на графике они изображаются дугами, выходящими из исходного события (1), которое означает момент начала выполнения проекта. Работе a_3 предшествует работа a_1 , поэтому на графике дуга a_3 непосредственно следует за дугой a_1 . Событие (2) означает момент окончания работы a_1 и начала работ, которым она предшествует. Работе a_4 предшествуют работы a_1 и a_2 . На графике эта зависимость отражается с помощью введения фиктивной работы (2, 3). Моментом свершения события (3) будет момент, к которому будут выполнены работы a_1 и a_2 и может начинаться работа a_4 . Аналогично, с учетом взаимосвязей, изображаются на графике все остальные работы. Завершающее событие (6) означает момент выполнения всего проекта, рис. 8.1.

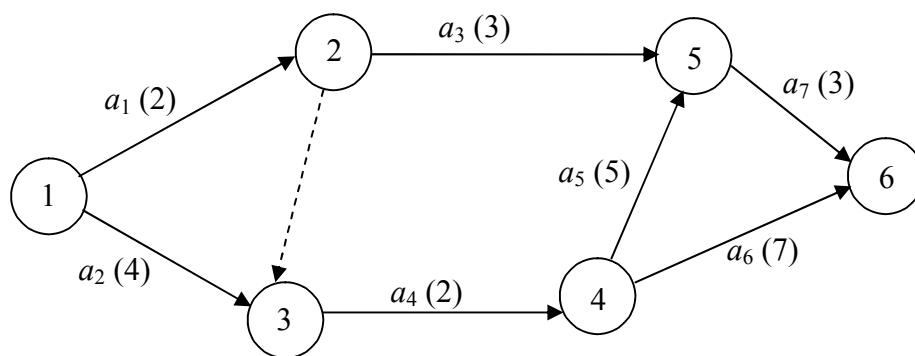


Рис. 8.1

8.2. Основные параметры сетевого графика

К основным параметрам сетевого графика относятся: продолжительность выполнения всего проекта, время свершения событий, сроки выполнения отдельных работ и их резервы времени.

Любая последовательность работ сети, в которой конечное событие каждой работы совпадает с начальным событием следующей за ней работы, называется *путем*. Под *длиной пути* следует понимать продолжительность выполнения всей последовательности работ, составляющих этот путь. Путь, в котором начальная вершина совпадает с исходным событием, а конечная – с завершающим, называется *полным*.

Критическим называется полный путь, имеющий наибольшую продолжительность. Таких путей в сети может быть несколько. Работы и события, принадлежащие критическому пути, называются *критическими*. Суммарная продолжительность работ, принадлежащих критическому пути, равна *критическому времени* $t_{кр}$ выполнения всего комплекса работ. На сетевом графике критический путь, как правило, выделяется двойной или жирной линией.

Рассмотрим основные временные параметры свершения событий – это ранний и поздний сроки свершения событий, резерв времени события, которые находятся по соответствующим формулам.

Ранний срок $t_p(j)$ *свершения события* j – это самый ранний момент времени, к которому завершаются все работы, предшествующие этому событию:

$$t_p(j) = \max_{(i,j) \in e_j^+} \{t_p(i) + t(i,j)\} \quad (j = \overline{2, n}), \quad (8.1)$$

где $t_p(i)$ – ранний срок свершения начального события работы (i, j) , $t(i, j)$ – продолжительность работы (i, j) . Предполагается, что $t_p(1) = t_p(I) = 0$, $t_p(S) = t_{кр}$.

Поздний срок $t_n(i)$ свершения события i – такой предельный момент времени, после которого остается ровно столько времени, сколько необходимо для завершения всех работ, следующих за этим событием:

$$t_n(i) = \min_{(i,j) \in e_j^-} \{t_n(j) - t(i, j)\} \quad (j = \overline{1, n-1}), \quad (8.2)$$

где $t_n(i)$ – поздний срок свершения конечного события работы (i, j) . Для завершающего события S предполагается, что $t_n(S) = t_p(S) = t_{кр}$.

Резерв времени $R(i)$ события i показывает, на какой предельно допустимый срок может задержаться свершение события i без нарушения срока наступления завершающего события:

$$R(i) = t_n(i) - t_p(i). \quad (8.3)$$

Кроме временных параметров свершения событий, рассматриваются временные параметры свершения работ – это ранний срок начала и окончания работы, поздний срок начала и окончания работы, полный резерв времени работы, свободный резерв времени.

Ранний срок начала работы (i, j) равен раннему сроку свершения события i :

$$t_{p.n.}(i, j) = t_p(i). \quad (8.4)$$

Ранний срок окончания работы равен сумме раннего срока свершения начального события работы и ее продолжительности:

$$t_{p.o.}(i, j) = t_{p.n.}(i, j) + t(i, j). \quad (8.5)$$

Поздний срок окончания работы совпадает с поздним сроком свершения ее конечного события:

$$t_{n.o.}(i, j) = t_n(j). \quad (8.6)$$

Поздний срок начала работы равен разности между поздним сроком свершения ее конечного события и продолжительностью:

$$t_{n.n.}(i, j) = t_{n.o.}(i, j) - t(i, j). \quad (8.7)$$

Полный резерв времени $R_n(i, j)$ работы (i, j) – это максимально возможный запас времени, на который можно отсрочить начало работы или увеличить продолжительность ее выполнения при условии, что весь комплекс работ будет завершен в критический срок:

$$R_n(i, j) = t_n(j) - t_p(i) - t(i, j). \quad (8.8)$$

Свободный резерв времени $R_c(i, j)$ работы (i, j) – это максимальный запас времени, на которое можно отсрочить начало работы или (если она началась в свой ранний срок) увеличить ее продолжительность при условии, что не нарушатся ранние сроки начала всех последующих работ:

$$R_c(i, j) = t_p(j) - t_p(i) - t(i, j). \quad (8.9)$$

Критические работы, как и критические события, резервов не имеют.

При расчете временных параметров вручную (если количество событий невелико) удобно проводить вычисления непосредственно на графе, воспользовавшись четырехсекторной схемой, рис. 8.2. В этом случае каждый кружок, обозначающий событие, делят на четыре сектора, в каждом из которых записывается соответствующая информация.

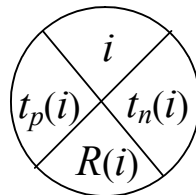


Рис. 8.2

1. В верхних секторах проставляются номера событий (в соответствии с ранжированием).

2. События рассматриваются в порядке возрастания номеров. По входящим в каждое событие работам определяем $t_p(i)$ и записываем в левом секторе. При этом следует иметь в виду, что $t_p(1) = 0$.

3. Начиная с конечного события, для которого $t_n(n) = t_{кр}$ (n – номер конечного события), для каждого события по выходящим из него работам определяем $t_n(i)$ и записываем в правом секторе.

4. В нижнем секторе записываем резерв времени события $R(i)$.

5. Критические события имеют резерв времени, равный 0, они и определяют критические работы и критический путь.

Для небольших проектов удобным дополнением к сетевому графику является *линейный график* (график *Ганта*). На линейном графике каждая работа $(i; j)$ изображается в привязке к оси времени Ot горизонтальным отрезком, длина которого в соответствующем масштабе равна продолжительности работы $t(i; j)$. Начало каждой работы совпадает с ранним сроком свершения ее начального события. Работы изображаются в той же последовательности, что и на сети.

Пример 8.2. Проект включает в себя работы, представленные в таблице 8.1, пример 8.1.

1) Построить сетевой график выполнения комплекса работ и рассчитать непосредственно на сетевом графике ранний и поздний сроки событий, резерв времени, если продолжительность каждой работы задана в таблице 8.2. Определить критический путь и критическое время выполнения всего комплекса работ, ранние и поздние сроки начала и окончания всех работ, а также полные и свободные резервы времени всех работ.

Таблица 8.2

Работа	Предшествующие ей работы	Продолжительность (дни)	Потребность в рабочих (чел.)
a_1	—	2	2
a_2	—	4	8
a_3	a_1	3	5
a_4	a_1, a_2	2	7
a_5	a_4	5	4
a_6	a_4	7	5
a_7	a_3, a_5	3	9

2) По сетевому графику и количеству исполнителей (рабочих), необходимых для выполнения каждой работы, построить линейный график Ганта с учетом занятости рабочих. Максимальное число рабочих в любой момент времени для данного проекта $R_{max} = 10$.

Решение.

1) Построенный сетевой график в примере 8.1 изобразим таким образом, чтобы можно было произвести расчеты временных параметров на сетевом графике, рис. 8.3.

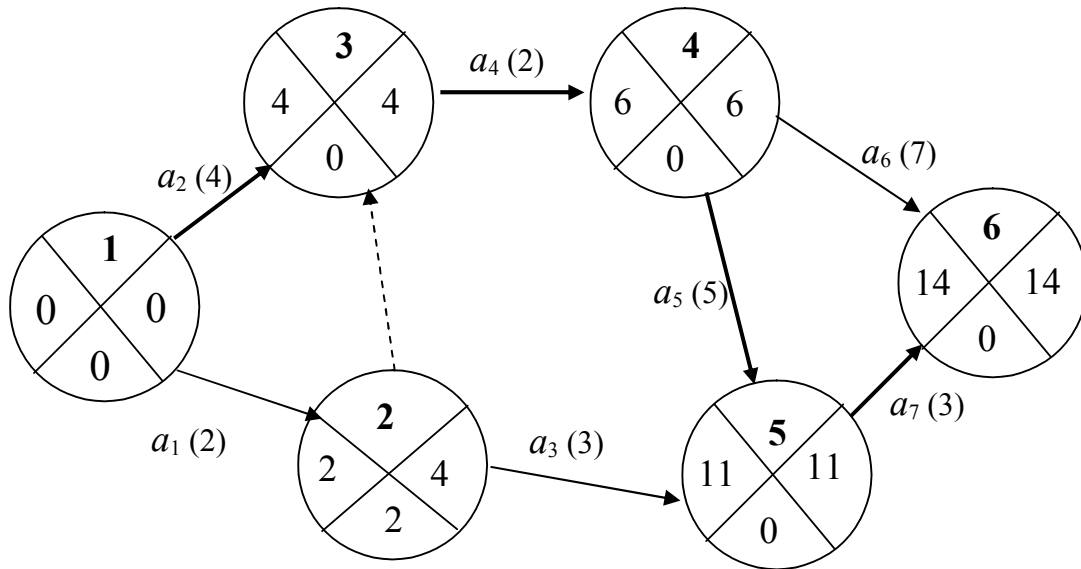


Рис. 8.3

1.1) Вычислим ранний срок свершения событий и запишем значения в левых секторах кругов-вершин:

$$t_p(1) = 0,$$

$$t_p(2) = \max \{t_p(1) + t(1;2)\} = \max \{0 + 2\} = 2,$$

$$t_p(3) = \max \{t_p(1) + t(1;3); t_p(2) + t(2;3)\} = \max \{\underline{0+4}; 2+0\} = 4,$$

$$t_p(4) = \max \{t_p(3) + t(3;4)\} = \max \{4 + 2\} = 6,$$

$$t_p(5) = \max \{t_p(2) + t(2;5); t_p(4) + t(4;5)\} = \max \{2 + 3; \underline{6+5}\} = 11,$$

$$t_p(6) = \max \{t_p(4) + t(4;6); t_p(5) + t(5;6)\} = \max \{6 + 7; \underline{11+3}\} = 14.$$

1.2) Определим критическое время: $t_{кр} = t_p(6) = t_n(6) = 14$.

Теперь можно определить работы, принадлежащие критическому пути, возвращаясь от завершающего события к начальному. Из двух работ, входящих в событие (6), $t_{кр} = 14$ определила работа (5;6). Поэтому эта работа является критической. Момент свершения события (5) определила работа (4;5), поэтому данная работа будет критической. В свою очередь момент свершения события (4) определила работа (3;4), событие (3) – работа (1;3). Все эти работы определили критический путь на сетевом графике,

который запишем следующим образом: $L_{кр} = (1 - 3 - 4 - 5 - 6)$. Критический путь выделим жирной линией.

1.3) Вычислим поздний срок свершения событий и запишем значения в правом секторе кругов-вершин:

$$t_n(6) = t_{кр} = 14,$$

$$t_n(5) = \min \{ t_n(6) - t(5;6) \} = \min \{ 14 - 3 \} = 11,$$

$$t_n(4) = \min \{ t_n(6) - t(4;6); t_n(5) - t(4;5) \} = \min \{ 14 - 7; \underline{11 - 5} \} = 6,$$

$$t_n(3) = \min \{ t_n(4) - t(3;4) \} = \min \{ 6 - 2 \} = 4,$$

$$t_n(2) = \min \{ t_n(3) - t(2;3); t_n(5) - t(2;5) \} = \min \{ 11 - 3; \underline{4 - 0} \} = 4,$$

$$t_n(1) = \min \{ t_n(2) - t(1;2); t_n(3) - t(1;3) \} = \min \{ \underline{4 - 4}; 4 - 2 \} = 0.$$

1.4) Вычислим резервы времени каждого события. Резервы времени всех критических событий равно 0: $R(1) = R(3) = R(4) = R(5) = R(6) = 0$.

$$R(2) = t_n(2) - t_p(2) = 4 - 2 = 2.$$

1.5) Определим ранние сроки начала и окончания работ. При этом фиктивные работы не рассматриваются.

$$t_{p.н.}(1;2) = t_{p.н.}(1;3) = t_p(1) = 0, \quad t_{p.н.}(2;5) = t_p(2) = 2,$$

$$t_{p.н.}(3;4) = t_p(3) = 4, \quad t_{p.н.}(4;5) = t_{p.н.}(4;6) = t_p(4) = 6,$$

$$t_{p.н.}(5;6) = t_p(5) = 11.$$

$$t_{p.o.}(1;2) = t_{p.н.}(1;2) + t(1;2) = 0 + 2 = 2;$$

$$t_{p.o.}(1;3) = t_{p.н.}(1;3) + t(1;3) = 0 + 4 = 4;$$

$$t_{p.o.}(2;5) = t_{p.н.}(2;5) + t(2;5) = 2 + 3 = 5;$$

$$t_{p.o.}(3;4) = t_{p.н.}(3;4) + t(3;4) = 4 + 2 = 6;$$

$$t_{p.o.}(4;5) = t_{p.н.}(4;5) + t(4;5) = 6 + 5 = 11;$$

$$t_{p.o.}(4;6) = t_{p.n.}(4;6) + t(4;6) = 6 + 7 = 13;$$

$$t_{p.o.}(5;6) = t_{p.n.}(5;6) + t(5;6) = 11 + 3 = 14.$$

1.6) Определим поздние сроки начала и окончания работ. При этом фиктивные работы не рассматриваются.

$$t_{n.o.}(1;2) = t_n(2) = 4; \quad t_{n.o.}(1;3) = t_n(3) = 4; \quad t_{n.o.}(2;5) = t_n(5) = 11;$$

$$t_{n.o.}(3;4) = t_n(4) = 6; \quad t_{n.o.}(4;6) = t_n(6) = 14; \quad t_{n.o.}(5;6) = t_n(6) = 14.$$

$$t_{n.n.}(1;2) = t_{n.o.}(1;2) - t(1;2) = 4 - 2 = 2;$$

$$t_{n.n.}(1;3) = t_{n.o.}(1;3) - t(1;3) = 4 - 4 = 0;$$

$$t_{n.n.}(2;5) = t_{n.o.}(2;5) - t(2;5) = 11 - 3 = 8;$$

$$t_{n.n.}(3;4) = t_{n.o.}(3;4) - t(3;4) = 6 - 2 = 4;$$

$$t_{n.n.}(4;5) = t_{n.o.}(4;5) - t(4;5) = 11 - 5 = 6;$$

$$t_{n.n.}(4;6) = t_{n.o.}(4;6) - t(4;6) = 14 - 7 = 7;$$

$$t_{n.n.}(5;6) = t_{n.o.}(5;6) - t(5;6) = 14 - 3 = 11.$$

1.7) Определим полные и свободные резервы работ. Критические работы не рассматриваются, так как они имеют нулевые значения резервов. Некритическими работами являются (1;2), (2;5), (4;6).

$$R_n(1;2) = t_n(2) - t_p(1) - t(1;2) = 4 - 0 - 2 = 2;$$

$$R_n(2;5) = t_n(5) - t_p(2) - t(2;5) = 11 - 2 - 3 = 6;$$

$$R_n(4;6) = t_n(6) - t_p(4) - t(4;6) = 14 - 6 - 7 = 1.$$

$$R_c(1;2) = t_p(2) - t_p(1) - t(1;2) = 2 - 0 - 2 = 0;$$

$$R_c(2;5) = t_p(5) - t_p(2) - t(2;5) = 11 - 2 - 3 = 6;$$

$$R_c(4;6) = t_p(6) - t_p(4) - t(4;6) = 14 - 6 - 7 = 1.$$

2) Построим график Ганта, рис. 8.4.

Каждая работа на линейном графике изображается прямолинейным отрезком в привязке к оси времени Ot , на которую нанесена временная шкала. Длина отрезка в выбранном масштабе равна продолжительности $t(i; j)$ выполнения работы $(i; j)$. На вертикальной оси отмечаем работы $(i; j)$. Над отрезками указываем число рабочих, необходимых для выполнения каждой работы.

В нашем случае комплекс работ начинается работами $(1; 2)$ и $(1; 3)$. Поэтому начала отрезков $1-2$ и $1-3$ расположим на вертикали при $t = 0$ (в произвольных точках), а длины их будут равны соответственно $t(1; 2) = 2$, $t(1; 3) = 4$. После завершения работы $(1; 2)$ выполняются работа $(2; 5)$ и фиктивная работа $(2; 3)$. Поэтому отрезок $2-5$ следует взять на вертикали при $t = 2$, а длина его будет соответственно равна $t(2; 5) = 3$. Фиктивная работа $(2; 3)$ на линейном графике изображается точкой (продолжительность по времени фиктивной работы равна нулю) на вертикали при $t = 2$. Работа $(3; 4)$ следует за работой $(1; 3)$. Поэтому начало отрезка $3-4$ расположим на вертикали при $t = 4$, а длина соответственно составит $t(3; 4) = 2$. За работой $(3; 4)$ следуют работы $(4; 5)$ и $(4; 6)$. Поэтому начала отрезков $4-5$ и $4-6$ расположим на вертикали при $t = 6$, а длины соответственно составят $t(4; 5) = 5$, $t(4; 6) = 7$. Работа $(5; 6)$ непосредственно следует за работами $(2; 5)$ и $(4; 5)$. Но так как работа $(4; 5)$ заканчивается позже, то отрезок $5-6$ берет начало на вертикали при $t = 11$, а длина соответственно будет равна $t(5; 6) = 3$.

По линейному графику можно определить $t_{кр}$ и критический путь. В данном случае $t_{кр} = 14$, критический путь — $L_{кр} = (1-3-4-5-6)$ (выделим более жирной линией).

Руководителю комплекса работ надо заранее знать, как будут использоваться специалисты в ходе выполнения работ комплекса. По линейному графику можно построить шкалу занятости специалистов. С этой целью спроецируем на ось времени Ot начальные и конечные точки всех работ и получим промежутки постоянства занятости: $(0; 2)$, $(2; 4)$, $(4; 5)$ и т.д. Для получения показателей занятости по промежуткам просуммируем интенсивность использования специалистов по отдельным работам, расположенным над каждым промежутком. Так, в промежутке $(0; 2)$ будет занято $2 + 8 = 10$ (чел.), в промежутке $(2; 4)$ — $8 + 5 = 13$ (чел.) и т.д. Ниже оси времени Ot помещена шкала занятости рабочих.

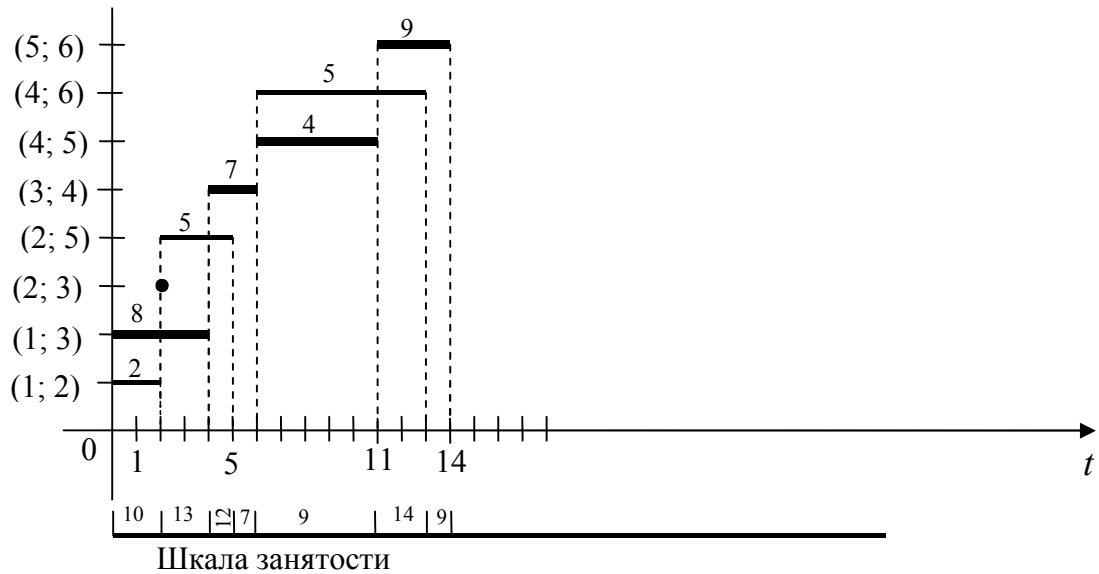


Рис. 8.4

Так как в данном случае уже для промежутка (2;4) наблюдается ситуация избытка специалистов ($13 > 10$), то для установления времени начала и окончания каждой работы дальнейшее решение задачи следует разбить на отдельные шаги, на которых придется пересмотреть сроки выполнения работ и выполнить перенос отдельных работ, сдвигая их во времени. Результатом станет следующий преобразованный линейный график, рис 8.5. В данном случае шкала занятости показывает, что при выполнении работ в любой промежуток времени используется не более 10 рабочих.

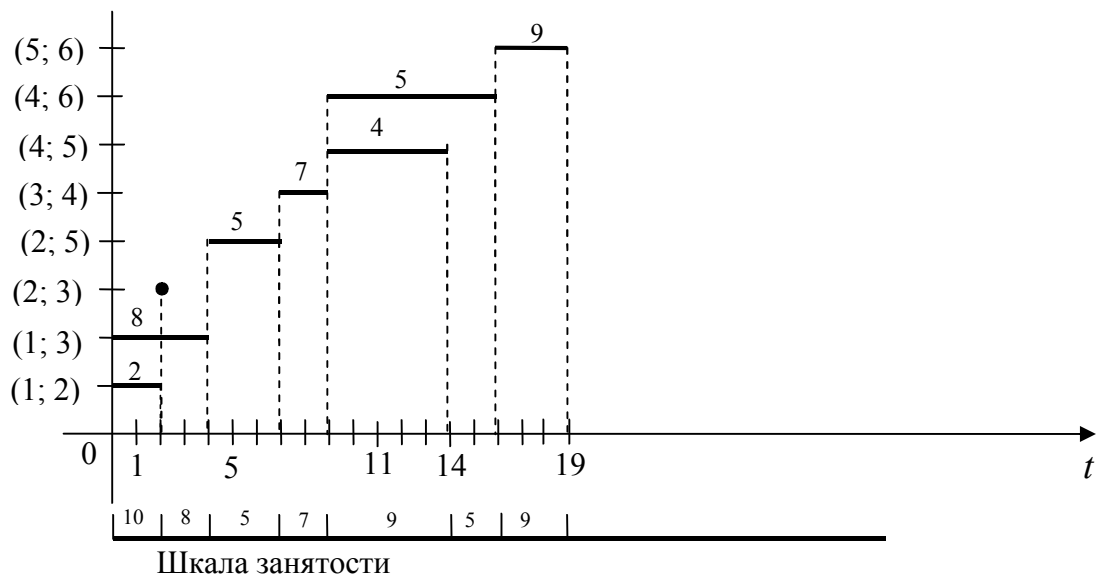


Рис. 8.5

□

ВОПРОСЫ И ЗАДАНИЯ ДЛЯ САМОКОНТРОЛЯ

1. Что называется графом $G = G(V, E)$? Какие вершины называются смежными? Какие ребра называются смежными?
2. Что называется ориентированным графом $G = G(V, E)$?
3. Какой граф называется псевдографом? Какой граф называется мультиграфом? Какой граф называется помеченным?
4. Что называют подграфом графа $G = G(V, E)$?
5. Какой граф называется полным? Как обозначается полный граф?
6. Сформулировать определение двудольного графа? Какой граф называется полным двудольным графом и как он обозначается?
7. Сформулировать определение маршрута (пути) длины k .
8. Какой путь называется цепью; простой цепью; циклом; простым циклом? Какой граф называется ациклическим?
9. Дать определение связного графа.
10. Какой граф называется соотнесенным графом орграфа $G = G(V, E)$?
11. В каком случае орграф называется связным? В каком случае орграф называется сильно связным?
12. Дать определение изоморфизма.
13. Сформулировать и доказать теорему об изоморфизме графов.
14. Сформулировать определение степени вершины неориентированного графа. Какая вершина называется изолированной? Какая вершина называется висячей?
15. Сформулировать определение полустепени исхода и полустепени захода вершины орграфа. Какая вершина называется истоком? Какая вершина называется стоком?
16. Сформулировать и доказать теорему Эйлера о сумме степеней вершин графа.

17. Сформулировать и доказать следствие о числе вершин нечетной степени.
18. Сформулировать и доказать следствие о сумме полустепеней вершин орграфа.
19. Дать определение матрицы смежности вершин неориентированного графа; матрицы смежности ребер неориентированного графа; матрицы инцидентий неориентированного графа.
20. Дать определение матрицы смежности вершин ориентированного графа; матрицы смежности дуг орграфа; матрицы инцидентий орграфа.
21. Какой граф называется взвешенным?
22. Как с помощью матрицы смежности вершин можно найти все маршруты, содержащие заданное количество ребер (дуг)?
23. Что называется упорядочиванием вершин связного орграфа?
24. Какой граф называется деревом? Какой орграф называется ориентированным деревом?
25. Какие вершины дерева называются листьями; внутренними вершинами? Какая вершина называется корнем?
26. Дать определение высоты дерева.
27. Сформулировать и доказать теорему о существовании единственной цепи между двумя вершинами дерева T .
28. Сформулировать и доказать теорему о связи числа вершин и числа ребер дерева T .
29. Сформулировать определение остовного дерева.
30. Какие методы можно использовать для построения остовного дерева.
31. Дать определение матрицы Кирхгофа.
32. Сформулировать теорему, которая используется для определения числа остовных деревьев в связном графе.

33. Что называется весом остовного графа? Какое остовное дерево называется минимальным остовным деревом?
34. Какие алгоритмы используются для построения минимального остовного дерева?
35. Какой алгоритм используется для нахождения кратчайших путей между двумя вершинами ориентированного графа.
36. Сформулировать определение эйлерова цикла. Какой граф называется эйлеровым?
37. Сформулировать и доказать теорему Эйлера об эйлеровом графе.
38. Какой путь называется эйлеровым?
39. Сформулировать критерий существования в графе эйлерова пути.
40. Сформулировать критерий существования в ориентированном графе эйлерова цикла.
41. Какой путь называется гамильтоновым путем? Какой цикл называется гамильтоновым циклом?
42. Сформулировать задачу коммивояжера и построить ее математическую модель.
43. Какой граф называется планарным?
44. Что называют гранью планарного графа? Что называют границей планарного графа?
45. Сформулировать и доказать теорему Эйлера о числе вершин, ребер и граней связного планарного графа.
46. Доказать теорему о том, что полный двудольный граф $K_{3,3}$ не является планарным.
47. Доказать теорему о том, что полный граф K_5 не является планарным.
48. Какой граф называется расширением графа $G = G(V, E)$? Какой граф называется производным от графа G_1 ?
49. Какие графы называются гомеоморфными?

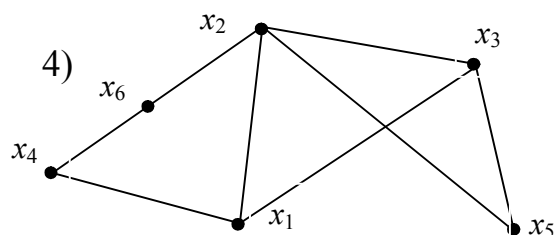
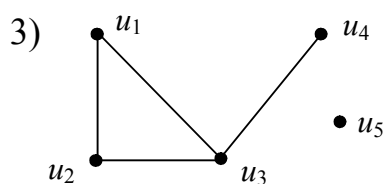
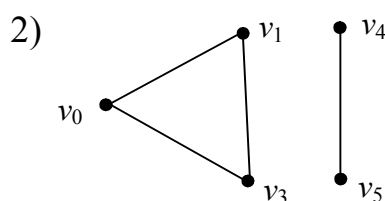
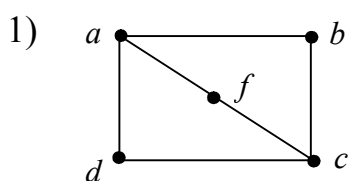
50. Сформулировать теорему Понтрягина – Куратовского.
51. Какой оргграф называется сетью? Что характеризует пропускная способность дуги?
52. Что представляет собой поток в сети?
53. Сформулировать задачу о максимальном потоке. Каким условиям должна удовлетворять эта задача?
54. Какая дуга называется насыщенной? Какой путь называется насыщенным? Какой поток называется насыщенным?
55. Какой поток в сети называется насыщенным?
56. Что представляет собой разрез на сети? Как производится разрез на сети?
57. Сформулировать теорему Форда – Фалкерсона.
58. Что называют паросочетанием в неориентированном графе?
59. Какое паросочетание называется максимальным? Какое паросочетание называется полным?
60. Какие методы используются для построения максимального паросочетания?
61. Что называют сетевым графиком? С чем отождествляются вершины и дуги оргграфа в графическом представлении комплекса работ для достижения поставленной цели?
62. Какими правилами необходимо руководствоваться при построении сетевого графика?
63. Что называют путем в сетевом графике? Что называют длиной пути?
64. Какой путь в сетевом графике называется критическим? Какие работы и события называются критическими? Какое время называется критическим?
65. Перечислить основные временные параметры свершения событий.
66. Перечислить временные параметры свершения работ.
67. Объяснить, как строится линейный график (график Ганта).

ЗАДАНИЯ ДЛЯ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

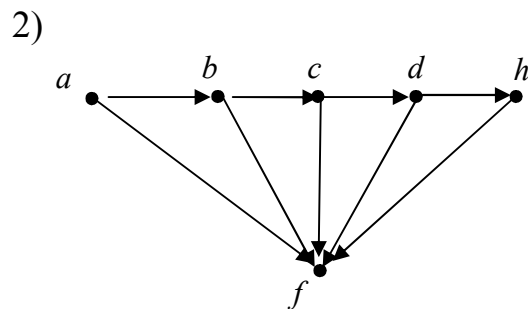
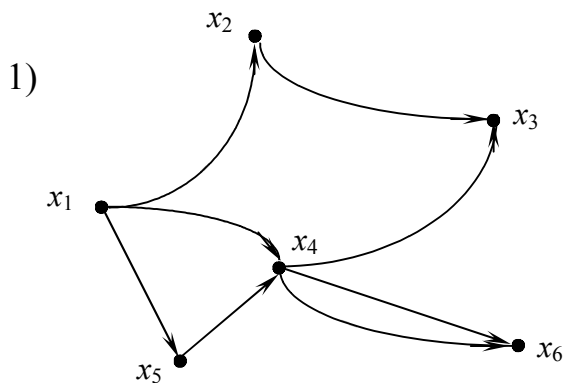
1. Графы, орграфы. Изоморфизм графов.

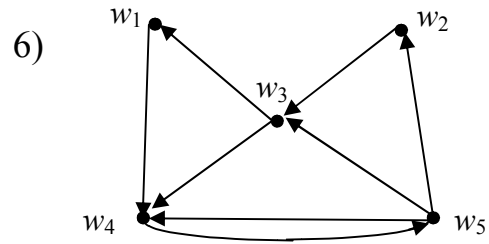
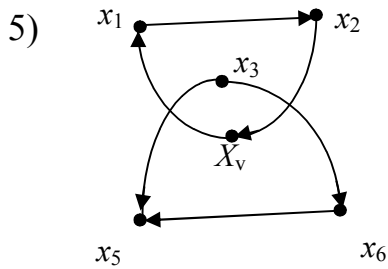
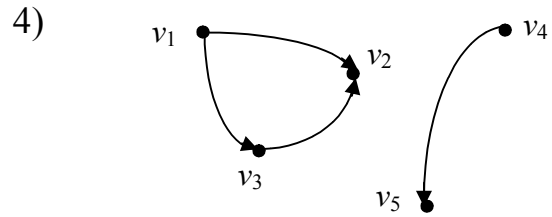
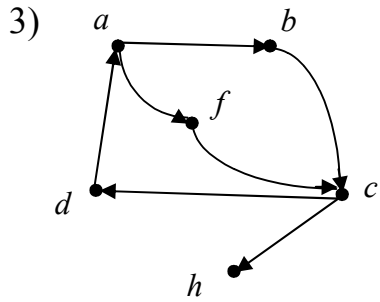
Представление графов в компьютере

1.1. Для указанных графов определить степень каждой вершины. Найти сумму степеней вершин графа. Найти 3 – 4 подграфа указанных графов. Определить, являются ли указанные графы связными? Для указанного графа найти 2 – 3 цикла. Которые из них являются простыми циклами? Построить матрицы смежности вершин и ребер, матрицу инцидентий.



1.2. Определить полустепень исхода и полустепень захода для каждой вершины заданных орграфов. Имеются ли здесь истоки и стоки? Для каждого графа построить три подграфа. Какой из приведенных ниже орграфов является связным? Какой из них является сильно связным? Для каждого графа найдите, если возможно, ориентированный путь длины 2, ориентированный путь длины 3, ориентированный путь длины 4. Построить матрицы смежности вершин и дуг, матрицу инцидентий.

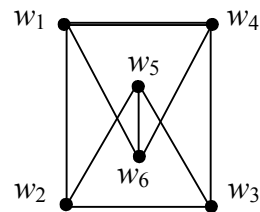
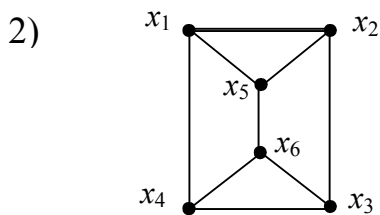
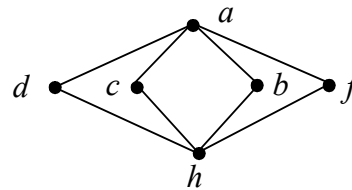
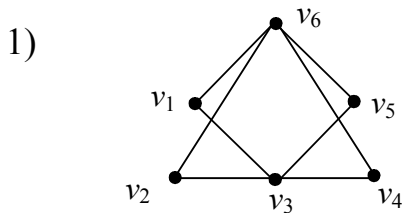




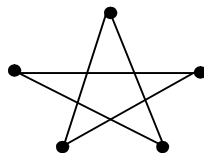
1.3. Изобразить следующие графы:

- 1) K_5 ; 2) K_6 ; 3) $K_{2,3}$; 4) $K_{1;4}$; 5) $K_{3;6}$.

1.4. Для каждой приведенной ниже пары графов описать изоморфизм.



1.5. Для данного графа построить изоморфный граф.



1.6. Изобразить все попарно неизоморфные 4-вершинные графы без петель и кратных ребер.

1.7. Сколько существует попарно неизоморфных 6-вершинных графов без петель и кратных ребер со следующим набором степеней вершин: (2, 2, 3, 3, 3, 5)?

Ответ: 2 графа.

1.8. Построить все попарно неизоморфные орграфы (без петель и кратных дуг), содержащие:

- 1) 3 вершины и 3 дуги;
- 2) 3 вершины и 4 дуги;
- 3) 4 вершины и 3 дуги.

1.9. По матрицам смежности вершин построить наглядные изображения графов:

$$1) \begin{bmatrix} 2 & 1 & 0 & 3 \\ 1 & 0 & 2 & 1 \\ 0 & 2 & 1 & 1 \\ 3 & 1 & 1 & 0 \end{bmatrix}; \quad 2) \begin{bmatrix} 0 & 2 & 1 & 0 & 0 & 2 \\ 2 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 2 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 2 & 1 & 1 & 2 \\ 2 & 0 & 1 & 1 & 2 & 0 \end{bmatrix}; \quad 3) \begin{bmatrix} 0 & 0 & 3 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \\ 1 & 0 & 0 & 1 \end{bmatrix}.$$

1.10. По матрицам смежности дуг построить наглядные изображения орграфов:

$$1) \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad 2) \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

1.11. В шахматном турнире по круговой системе участвуют семь студентов. Известно, что Иван сыграл шесть партий, Анатолий – пять, Алексей и Дмитрий – по три, Семен и Илья – по две, Евгений – одну. С кем сыграл Алексей?

Ответ: Алексей сыграл с Анатолием, Иваном, Дмитрием.

1.12. Чемпионат по футболу проводился по круговой системе. За победу в матче давалось 2 очка, за ничью – 1, за поражение – 0. Если две команды набирали одинаковое количество очков, то место определялось по разности забитых и пропущенных мячей. Чемпион набрал семь очков, второй призер – пять, третий – три. Сколько очков набрала команда, занявшая последнее место?

Ответ: 2 очка.

1.13. Группа, в составе которой Петр совершил туристическую поездку, состояла из пятнадцати человек. Вернувшись из путешествия, Петр рассказал, что каждый участник группы был знаком ровно с пятью другими участниками. Возможно ли это?

Ответ: невозможно.

1.14. Может ли в государстве, в котором из каждого города выходит ровно три дороги, быть ровно 100 дорог между городами?

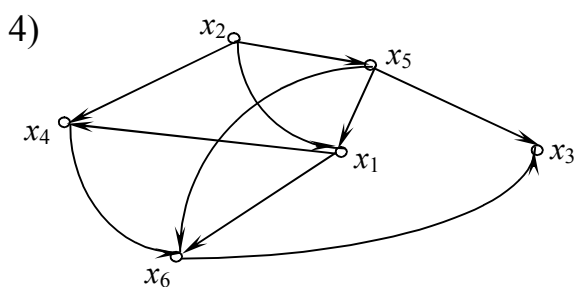
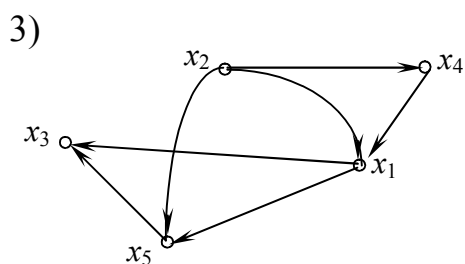
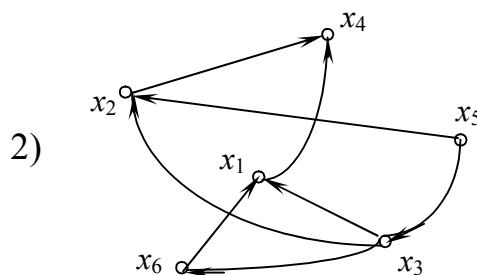
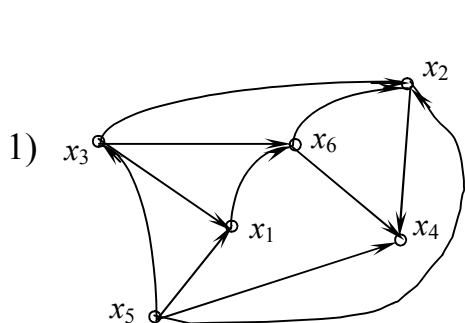
Ответ: невозможно.

1.15. Десять кандидатов готовятся к двум космическим экспедициям на Марс. Поскольку экспедиции будут продолжаться несколько лет, а их участники окажутся в замкнутом пространстве небольшого объема, то важное значение приобретает психологическая совместимость членов экипажа. Путем тестирования были установлены пары кандидатов, присутствие которых в одной и той же экспедиции было бы нежелательным. Результаты тестирования отражены в таблице. (Если на пересечении i -той строки и j -того столбца таблицы находится знак «+», то участие i -того и j -того кандидатов в одной экспедиции нежелательно). Разделите кандидатов на две группы для участия в экспедициях.

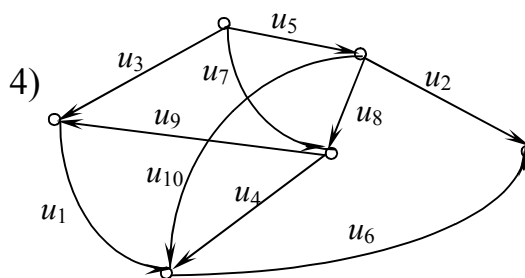
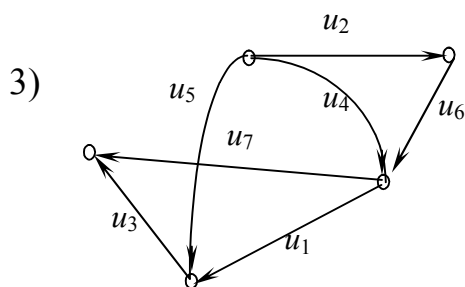
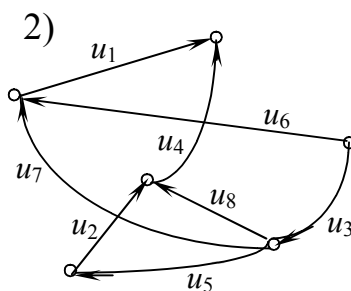
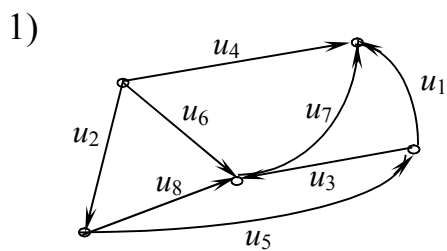
	1	2	3	4	5	6	7	8	9	10
1		+	+	+						
2	+				+					
3	+						+			
4	+					+				
5		+						+		
6				+						+
7			+					+		+
8					+		+		+	
9								+		+
10						+	+		+	

2. Упорядочение вершин и дуг орграфа

2.1. Упорядочить вершины орграфов, изображенных на рисунках, графическим способом. Построить наглядные изображения изоморфных графов.



2.2. Упорядочить дуги орграфов, изображенных на рисунках, графическим способом. Построить наглядные изображения изоморфных графов.



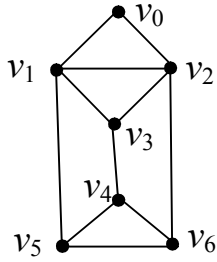
3. Остовные деревья. Алгоритмы Прима и Краскала

3.1. Заданы графы, у которых вершины упорядочены так, как показаны на рисунке.

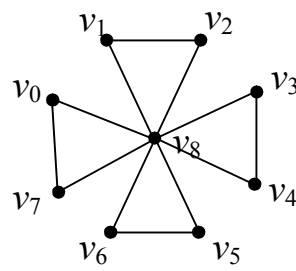
Требуется:

- 1) найти остовное дерево с помощью метода поиска в ширину;
- 2) найти остовное дерево с помощью метода поиска в глубину.

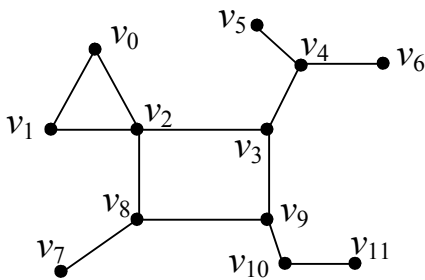
1)



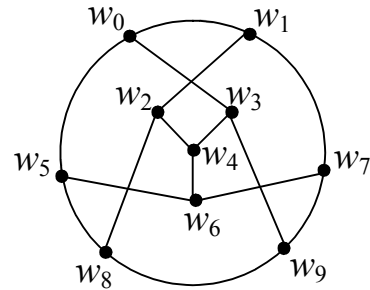
2)



3)



4)

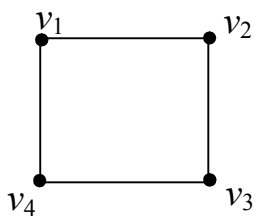


5) K_5 ,

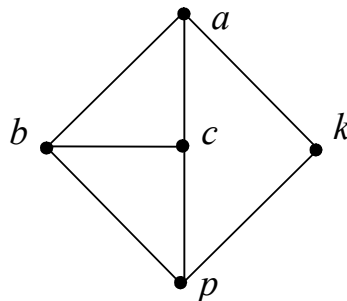
6) $K_{3,4}$.

3.2. Используя матричную формулу Кирхгофа, определить число остовных деревьев для каждого из приведенных ниже графов.

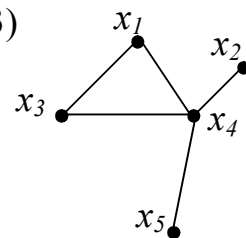
1)



2)

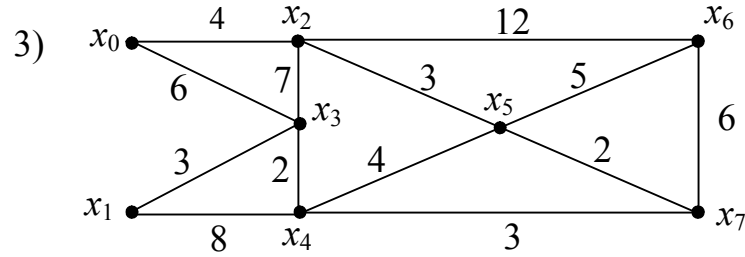
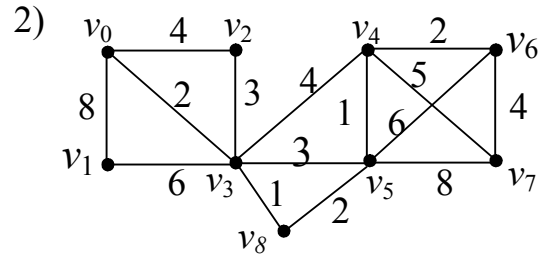
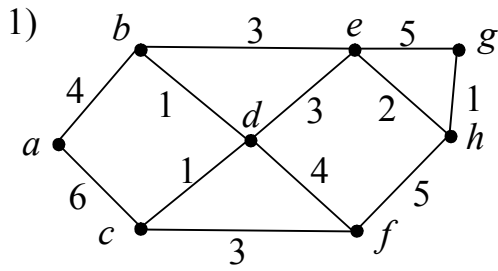


3)



3.3. Заданы взвешенные графы. Найти минимальные остовные деревья, воспользовавшись:

- 1) алгоритмом Краскала;
- 2) алгоритмом Прима.



Ответ. 1) $\omega_{\min}(T) = 15$; 2) $\omega_{\min}(T) = 21$; 3) $\omega_{\min}(T) = 22$.

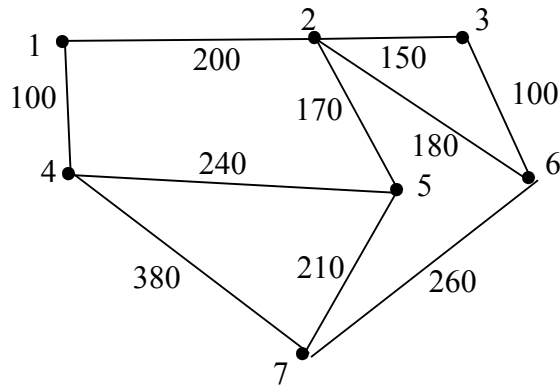
3.4. В таблице приведены расстояния (в милях) между шестью городами Ирландии. Построить сеть дорог минимальной общей длины, связывающую все шесть городов.

	Атлон	Дублин	Голуэй	Лимерик	Слайго	Уэксфорд
Атлон	–	78	56	73	71	114
Дублин	78	–	132	121	135	96
Голуэй	56	132	–	64	85	154
Лимерик	73	121	64	–	144	116
Слайго	71	135	85	144	–	185
Уэксфорд	114	96	154	116	185	–

Ответ: 365.

3.5. На строительном участке нужно создать телефонную сеть, соединяющую все бытовки. Для того чтобы телефонные линии не мешали строительству, их решили проводить вдоль дорог. Схема участка изображена на рисунке, где бытовкам соответствуют вершины графа и указаны длины дорог между ними.

Каким образом провести телефонные линии, чтобы их общая длина была минимальной?



3.6. Граф G задан матрицей весов. Изобразить граф. Построить минимальный по весу остов T и найти его вес $\omega_{\min}(T)$:

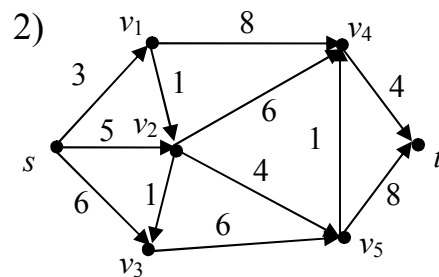
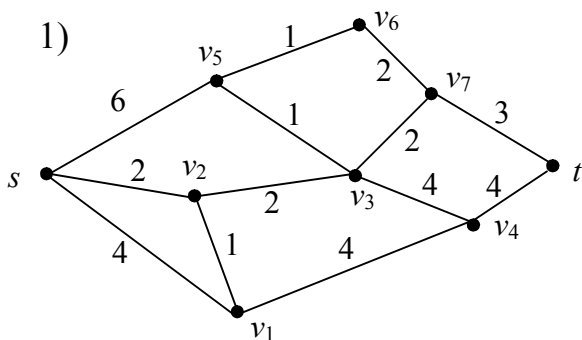
1)
$$\begin{pmatrix} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ x_1 & - & 10 & \infty & 5 & \infty & \infty & 14 \\ x_2 & 10 & - & 6 & 2 & 4 & 8 & \infty \\ x_3 & \infty & 6 & - & 3 & 1 & 1 & \infty \\ x_4 & 5 & 2 & 3 & - & 6 & \infty & 3 \\ x_5 & \infty & 4 & 1 & 6 & - & 5 & \infty \\ x_6 & \infty & 8 & 1 & \infty & 5 & - & 2 \\ x_7 & 14 & \infty & \infty & 3 & \infty & 2 & - \end{pmatrix}$$

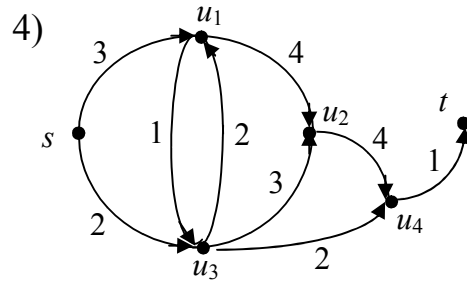
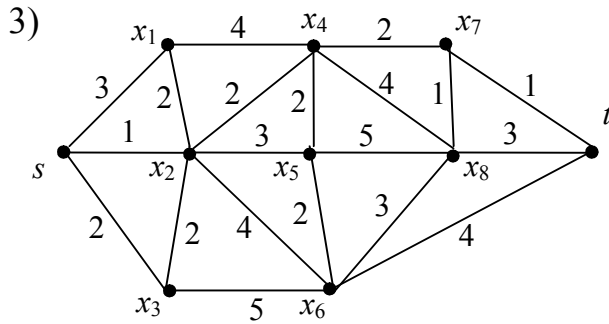
2)
$$\begin{pmatrix} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ x_1 & - & 10 & 11 & \infty & 14 & \infty & 12 \\ x_2 & 10 & - & 10 & 9 & \infty & \infty & 7 \\ x_3 & 11 & 10 & - & 12 & 10 & \infty & 6 \\ x_4 & \infty & 9 & 12 & - & 9 & 12 & \infty \\ x_5 & 14 & \infty & 10 & 9 & - & 11 & 12 \\ x_6 & \infty & \infty & \infty & 12 & 11 & - & \infty \\ x_7 & 12 & 7 & 6 & \infty & 12 & \infty & - \end{pmatrix}$$

Ответ. 1) $\omega_{\min}(T) = 14$; 2) $\omega_{\min}(T) = 52$.

4. Алгоритм Дейкстры

4.1. Использовать алгоритм Дейкстры для нахождения в приведенных ниже графах (орграфах) кратчайших расстояний между вершинами s и t .





Ответ: 1) $d_{\min} = 11$; 2) $d_{\min} = 13$; 3) $d_{\min} = 6$; 4) $d_{\min} = 5$.

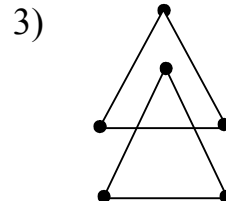
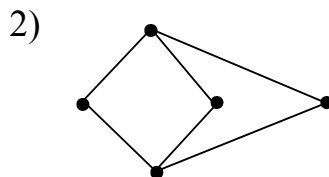
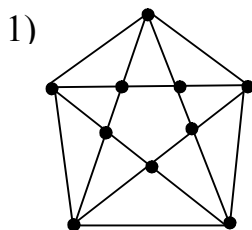
4.2. По заданной матрице весов графа G найти величину минимального пути и сам путь от вершины $s = x_1$ до вершины $t = x_7$ ($t = x_6$) с помощью алгоритма Дейкстры.

1) $\Omega = \begin{pmatrix} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ x_1 & - & 6 & 9 & 13 & 12 & \infty \\ x_2 & \infty & - & 5 & 9 & 6 & \infty \\ x_3 & \infty & \infty & - & 6 & \infty & 15 \\ x_4 & \infty & \infty & \infty & - & 8 & 9 \\ x_5 & \infty & \infty & \infty & \infty & - & 8 \\ x_6 & \infty & \infty & \infty & \infty & \infty & - \end{pmatrix}$; 2) $\Omega = \begin{pmatrix} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ x_1 & - & \infty & 7 & 12 & \infty & 10 & \infty \\ x_2 & \infty & - & \infty & \infty & 6 & 13 & \infty \\ x_3 & \infty & 5 & - & 4 & 5 & 6 & \infty \\ x_4 & \infty & \infty & \infty & - & \infty & \infty & 4 \\ x_5 & \infty & \infty & \infty & 8 & - & 5 & 9 \\ x_6 & \infty & \infty & \infty & \infty & \infty & - & 6 \\ x_7 & \infty & \infty & \infty & \infty & \infty & \infty & - \end{pmatrix}$.

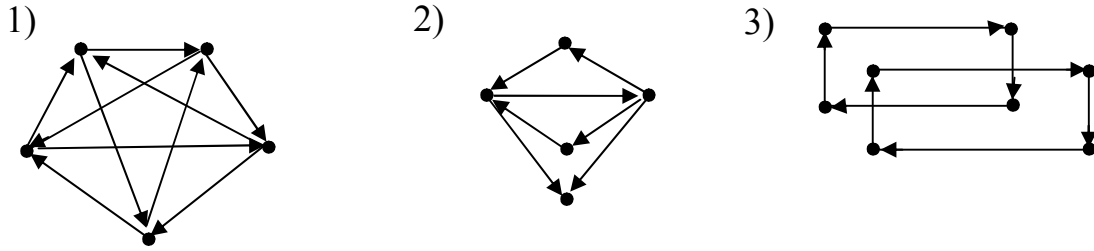
Ответ. 1) $d_{\min} = 20$; 2) $d_{\min} = 15$.

5. Эйлеровы и гамильтоновы циклы

5.1. Среди приведенных ниже графов найти те, которые имеют эйлеров цикл.



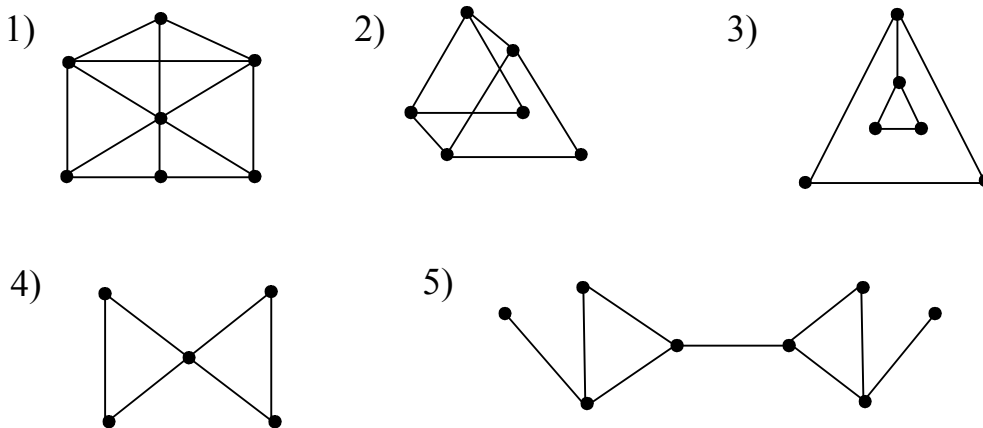
5.2. Какие из следующих орграфов имеют эйлеровы циклы?



5.3. Из какого минимального числа кусков проволоки можно спаять каркас тетраэдра; октаэдра; икосаэдра; додекаэдра? (Толщина всех ребер каркаса должна быть одинаковой).

Указание: Используйте теорему 4.4.

5.4. Найти гамильтонов цикл, если он существует, для каждого из приведенных ниже графов. Найти гамильтонов путь, если он существует, для каждого из приведенных ниже графов.



5.5. Нарисовать граф с шестью вершинами, который имеет гамильтонов цикл, но не имеет эйлерова цикла.

5.6. Нарисовать граф с шестью вершинами, который имеет эйлеров цикл, но не имеет гамильтонова цикла.

5.7. На одной и той же поточной линии предприятие может обрабатывать n видов деталей. Время наладки при переходе линии на обработку от одного вида деталей к другому представлена матрицей $A = [a_{ij}]$, где a_{ij} – затраты времени (часы) на наладку переменного-поточной линии при переходе к обработке деталей от i -того вида к деталям j -того вида. С по-

мощью алгоритма Литтла найти последовательность перестройки линии с одной детали на другую, при которой должны быть обеспечены минимальные общие потери рабочего времени на ее переналадку.

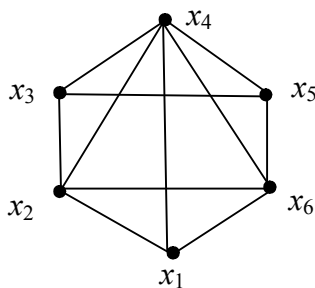
$$1) \begin{pmatrix} \infty & 10 & 8 & 9 & 11 \\ 10 & \infty & 13 & 12 & 12 \\ 8 & 13 & \infty & 6 & 6 \\ 9 & 12 & 6 & \infty & 10 \\ 11 & 12 & 6 & 10 & \infty \end{pmatrix}; 2) \begin{pmatrix} \infty & 10 & 10 & 11 \\ 10 & \infty & 13 & 9 \\ 10 & 13 & \infty & 12 \\ 11 & 9 & 12 & \infty \end{pmatrix};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

- Ответ. 1) $1 \rightarrow 2 \rightarrow 5 \rightarrow 3 \rightarrow 4 \rightarrow 1$ или $1 \rightarrow 4 \rightarrow 3 \rightarrow 5 \rightarrow 2 \rightarrow 1$, $\varphi = 43$;
 2) $1 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 1$, $\varphi = 41$;
 3) $1 \rightarrow 4 \rightarrow 3 \rightarrow 6 \rightarrow 5 \rightarrow 2 \rightarrow 1$, $\varphi = 21$;

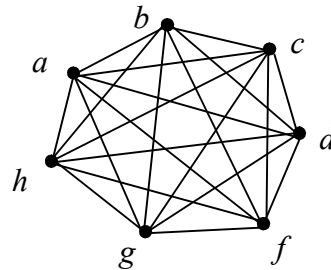
6. Планарные графы

6.1. Каждый из приведенных графов проверить на планарность. Если граф планарен, то изобразить его плоскую укладку. Для каждого планарного графа проверить, что $n - m + f = 2$.

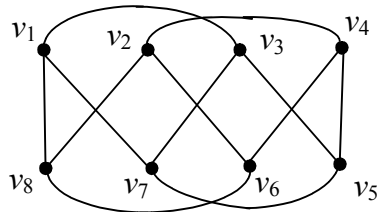
1)



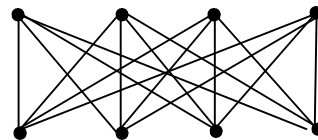
2)



3)



4)



6.2. Проверить, выполняется ли теорема Эйлера для правильных многогранников.

6.3. Если планарный граф содержит 12 вершин со степенью 3, то сколько у него ребер и граней?

Ответ. 18 ребер и 8 граней.

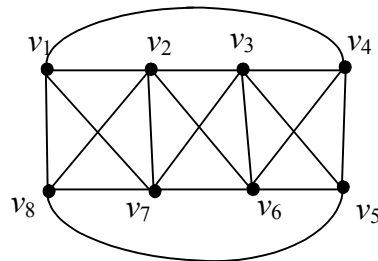
6.4. Если планарный граф имеет вершины со степенями 2, 2, 2, 3, 3, 3, 4, 4, 4 и 5, соответственно, то сколько у него ребер и граней?

Ответ. 16 ребер и 8 граней.

6.5. Доказать, что произвольный граф с четырьмя вершинами является планарным графом.

6.6. Один инженер придумал схему печатной платы, которая состоит из 12 приборов и 32 проводников, соединяющих их. Можно ли изготовить такую плату, что все проводники будут расположены на одной ее стороне?

6.7. Выяснить, какое наименьшее число вершин нужно удалить из графа, изображенного на рисунке, чтобы получить планарный граф?



Ответ. 2 вершины.

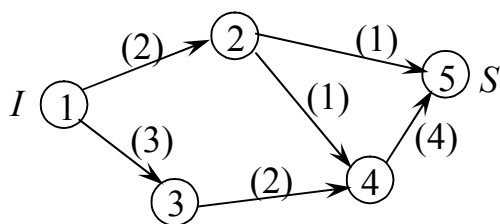
6.8. Выяснить, какое наименьшее число ребер надо удалить из графа K_6 , чтобы получить планарный граф?

Ответ. 3 ребра.

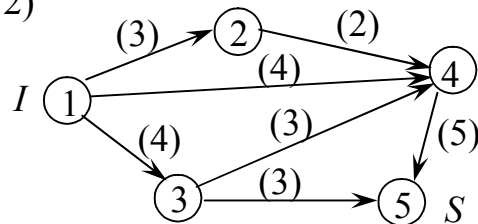
7. Сети. Потоки в сетях

7.1. На заданной сети сформировать максимальный по величине поток, направленный из истока I в сток S . Выписать ребра, образующие на сети разрез минимальной пропускной способности.

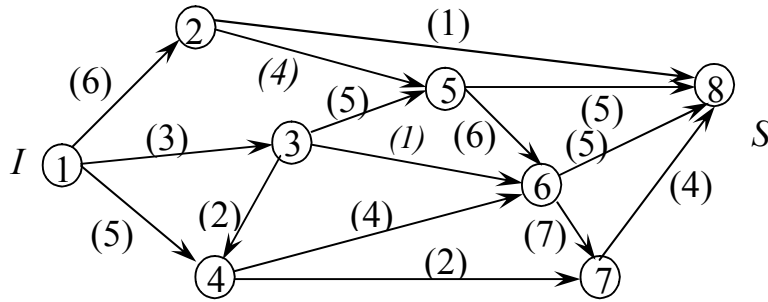
1)



2)



3)



8. Паросочетания

8.1. Найти оптимальное распределение работ между исполнителями с учетом их возможностей, оцениваемых элементами данной матрицы $\Omega = [\omega_{ij}]$: если $\omega_{ij} = 1$, то исполнитель I_i может выполнить работу P_j , если же $\omega_{ij} = 0$, то нет. При этом каждый исполнитель одновременно может выполнять только одну работу, и каждая работа одновременно может выполняться только одним исполнителем.

Распределить работы между исполнителями так, чтобы одновременно выполнялось, возможно, большее число работ.

1)

$$\Omega = [\omega_{ij}] = \left(\begin{array}{c|cccc} & P_1 & P_2 & P_3 & P_4 \\ \hline I_1 & 1 & 1 & 0 & 0 \\ I_2 & 1 & 0 & 1 & 1 \\ I_3 & 0 & 1 & 1 & 0 \\ I_4 & 1 & 0 & 0 & 0 \end{array} \right);$$

2)

$$\Omega = [\omega_{ij}] = \left(\begin{array}{c|cccc} & P_1 & P_2 & P_3 & P_4 \\ \hline I_1 & 1 & 0 & 1 & 0 \\ I_2 & 0 & 1 & 0 & 0 \\ I_3 & 0 & 0 & 1 & 0 \\ I_4 & 1 & 1 & 0 & 0 \\ I_5 & 1 & 0 & 0 & 1 \end{array} \right).$$

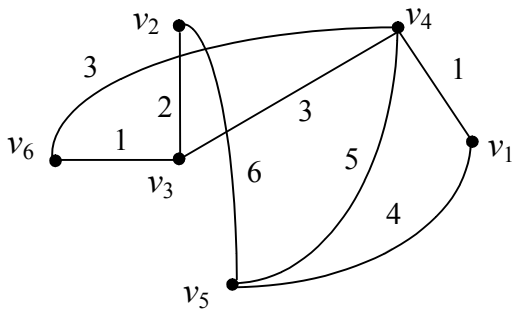
ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

Задание 1

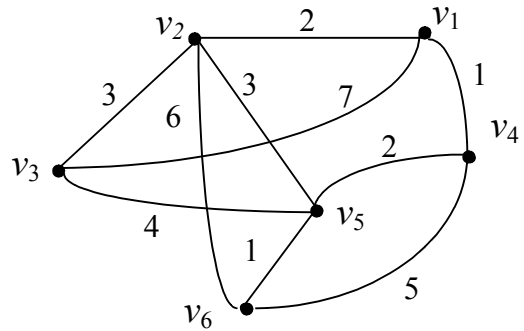
Для данного взвешенного графа:

- 1) определить степень каждой вершины;
- 2) построить матрицу смежности вершин, матрицу смежности ребер, матрицу инциденций;
- 3) используя теорему Кирхгофа, найти число остовных деревьев;
- 4) найти минимальное остовное (корневое) дерево, используя алгоритм Краскала или алгоритм Прима. Определить высоту построенного корневого дерева.

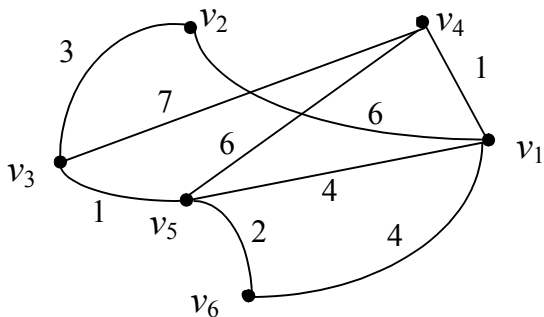
1)



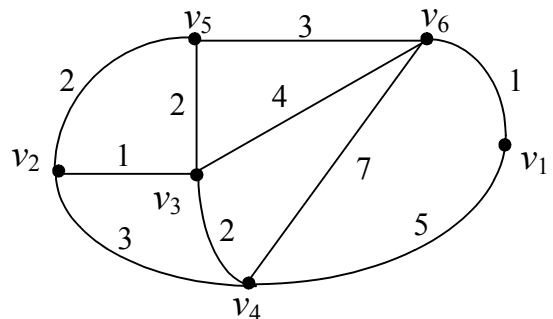
2)

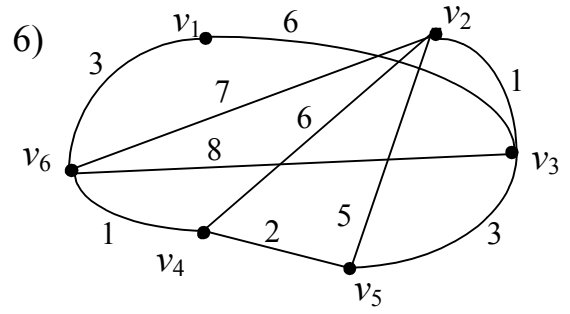
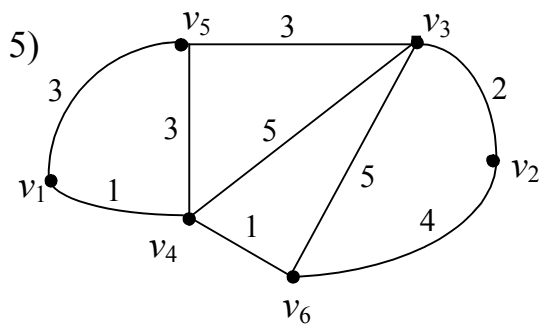


3)

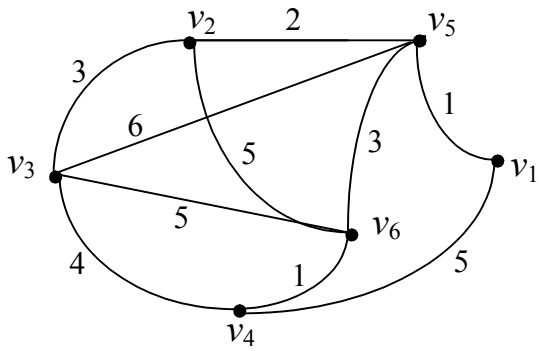


4)

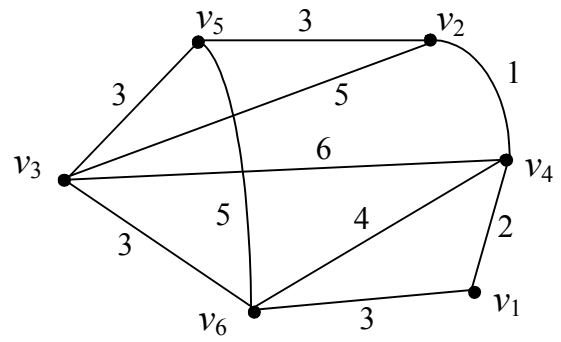




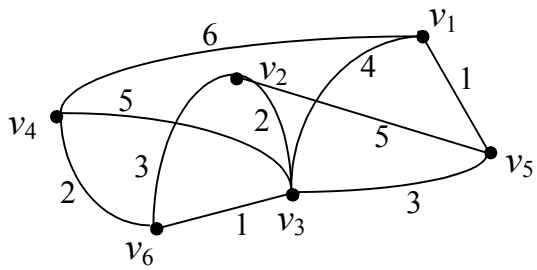
7)



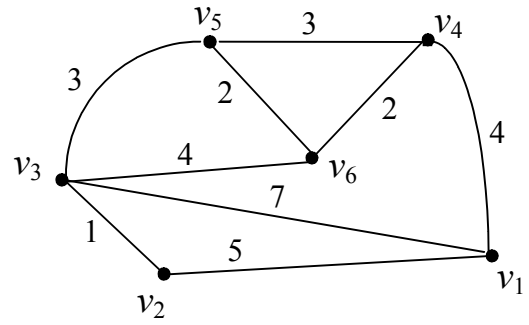
8)



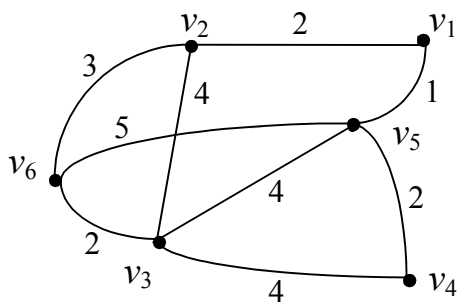
9)



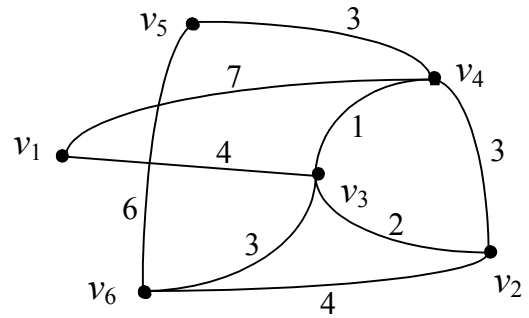
10)



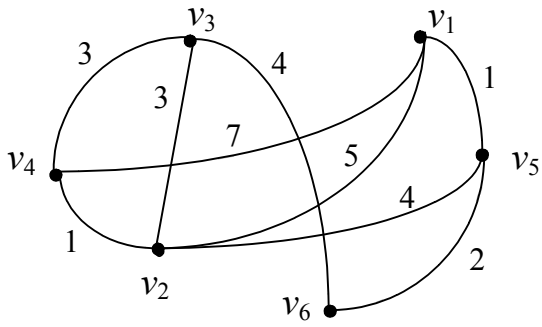
11)



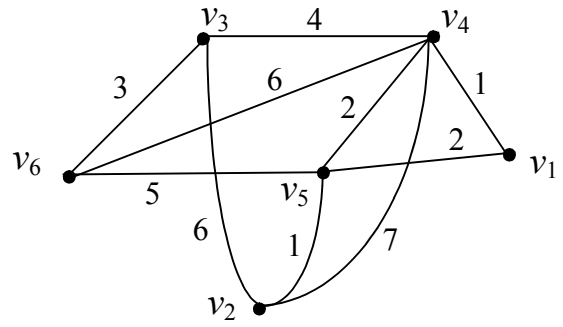
12)



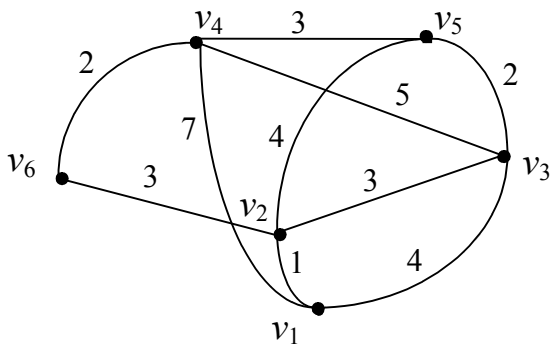
13)



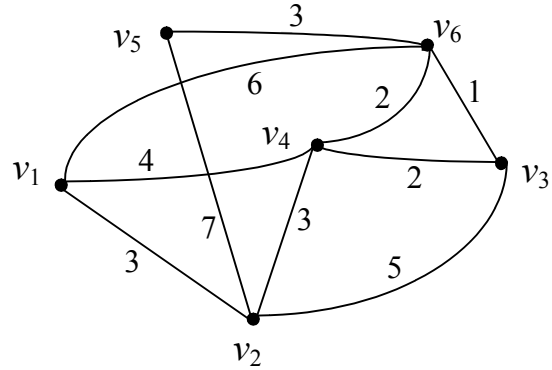
14)



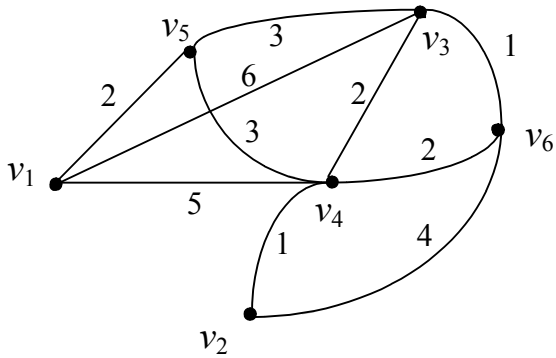
15)



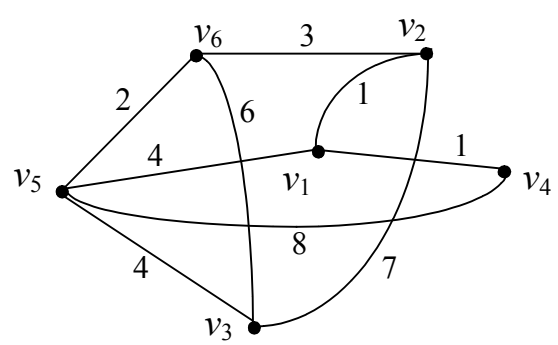
16)



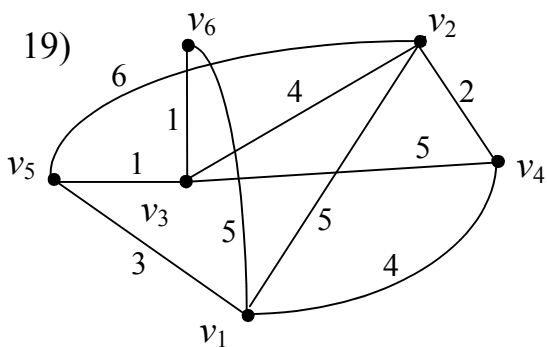
17)



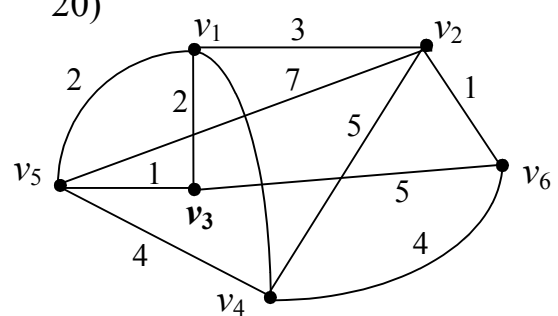
18)

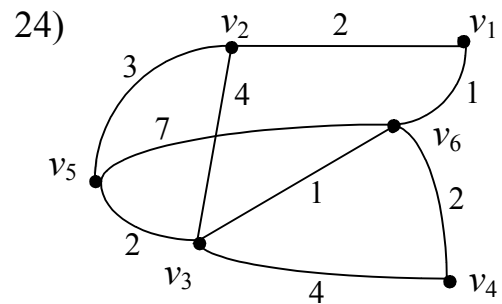
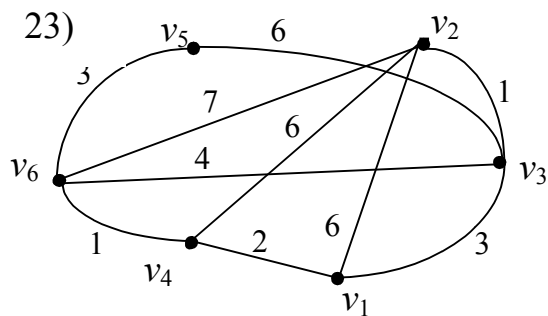
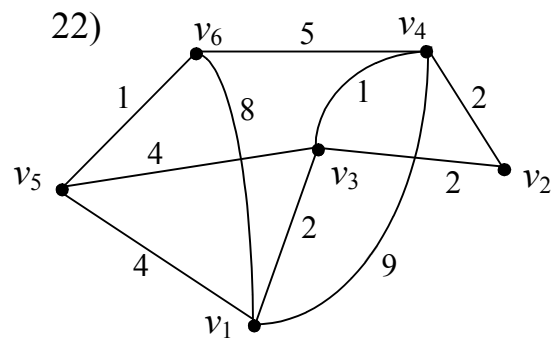
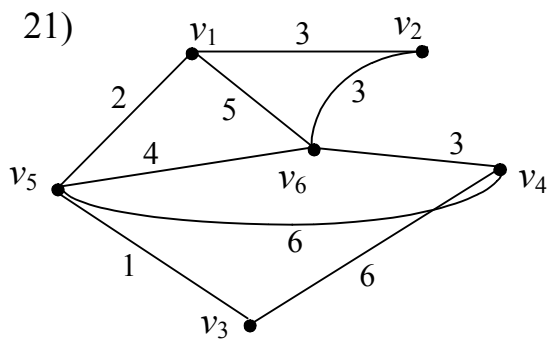


19)



20)





Задание 2

По заданной матрице весов $\Omega = [\omega_{ij}]$:

- 1) построить ориентированный граф;
- 2) найти полустепень захода и полустепень исхода каждой вершины;
- 3) построить матрицу смежности вершин, матрицу смежности дуг, матрицу инцидентий;
- 4) найти величину минимального пути и сам путь от вершины $s = x_1$ до вершины $t = x_6$ с помощью алгоритма Дейкстры.

1)
$$\Omega = \left(\begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & - & 11 & \infty & 14 & 15 & \infty \\ x_2 & \infty & - & 13 & \infty & \infty & \infty \\ x_3 & \infty & \infty & - & \infty & \infty & 13 \\ x_4 & \infty & 7 & 11 & - & 9 & \infty \\ x_5 & \infty & 11 & 10 & \infty & - & 14 \\ x_6 & \infty & \infty & \infty & \infty & \infty & - \end{array} \right)$$

2)
$$\Omega = \left(\begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & - & 5 & 8 & 7 & 18 & \infty \\ x_2 & \infty & - & 11 & \infty & \infty & \infty \\ x_3 & \infty & \infty & - & \infty & \infty & 17 \\ x_4 & \infty & 10 & 12 & - & 6 & \infty \\ x_5 & \infty & 7 & 8 & \infty & - & 11 \\ x_6 & \infty & \infty & \infty & \infty & \infty & - \end{array} \right)$$

$$3) \left(\begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & - & 6 & 8 & 11 & 10 & \infty \\ x_2 & \infty & - & \infty & 9 & 7 & 15 \\ x_3 & \infty & 8 & - & 7 & 4 & 11 \\ x_4 & \infty & \infty & \infty & - & 6 & 7 \\ x_5 & \infty & \infty & \infty & \infty & - & 9 \\ x_6 & \infty & \infty & \infty & \infty & \infty & - \end{array} \right)$$

$$4) \left(\begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & - & 5 & 6 & 9 & \infty & \infty \\ x_2 & \infty & - & \infty & 3 & \infty & 14 \\ x_3 & \infty & 3 & - & 3 & 4 & 16 \\ x_4 & \infty & \infty & \infty & - & \infty & 4 \\ x_5 & \infty & \infty & \infty & 3 & - & 8 \\ x_6 & \infty & \infty & \infty & \infty & \infty & - \end{array} \right)$$

$$5) \left(\begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & - & 7 & 9 & \infty & 11 & \infty \\ x_2 & \infty & - & \infty & 6 & \infty & 13 \\ x_3 & \infty & 6 & - & 5 & 6 & \infty \\ x_4 & \infty & \infty & \infty & - & \infty & 7 \\ x_5 & \infty & 4 & \infty & 6 & - & 8 \\ x_6 & \infty & \infty & \infty & \infty & \infty & - \end{array} \right)$$

$$6) \left(\begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & - & 7 & 15 & \infty & 14 & \infty \\ x_2 & \infty & - & 7 & 16 & \infty & \infty \\ x_3 & \infty & \infty & - & 19 & \infty & 21 \\ x_4 & \infty & \infty & \infty & - & \infty & 17 \\ x_5 & \infty & 13 & 14 & 15 & - & 18 \\ x_6 & \infty & \infty & \infty & \infty & \infty & - \end{array} \right)$$

$$7) \left(\begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & - & 10 & 12 & \infty & \infty & \infty \\ x_2 & \infty & - & 11 & 9 & \infty & 19 \\ x_3 & \infty & \infty & - & \infty & 10 & \infty \\ x_4 & \infty & \infty & 13 & - & 11 & 10 \\ x_5 & \infty & \infty & \infty & \infty & - & 6 \\ x_6 & \infty & \infty & \infty & \infty & \infty & - \end{array} \right)$$

$$8) \left(\begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & - & 7 & 2 & \infty & 13 & \infty \\ x_2 & \infty & - & \infty & \infty & 6 & \infty \\ x_3 & \infty & 2 & - & 1 & 3 & 11 \\ x_4 & \infty & \infty & \infty & - & \infty & 5 \\ x_5 & \infty & \infty & \infty & 3 & - & 5 \\ x_6 & \infty & \infty & \infty & \infty & \infty & - \end{array} \right)$$

$$9) \left(\begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & - & 6 & \infty & 9 & 12 & \infty \\ x_2 & \infty & - & 6 & \infty & \infty & \infty \\ x_3 & \infty & \infty & - & \infty & \infty & 6 \\ x_4 & \infty & 4 & 8 & - & 6 & 14 \\ x_5 & \infty & 7 & 5 & \infty & - & 10 \\ x_6 & \infty & \infty & \infty & \infty & \infty & - \end{array} \right)$$

$$10) \left(\begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & - & 4 & 9 & 8 & \infty & \infty \\ x_2 & \infty & - & 2 & \infty & \infty & \infty \\ x_3 & \infty & \infty & - & \infty & \infty & 3 \\ x_4 & \infty & 2 & 4 & - & 6 & \infty \\ x_5 & \infty & 2 & \infty & \infty & - & 3 \\ x_6 & \infty & \infty & \infty & \infty & \infty & - \end{array} \right)$$

$$11) \left(\begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & - & 11 & 14 & \infty & \infty & \infty \\ x_2 & \infty & - & 8 & 10 & 15 & \infty \\ x_3 & \infty & \infty & - & 11 & 16 & 20 \\ x_4 & \infty & \infty & \infty & - & \infty & 12 \\ x_5 & \infty & \infty & \infty & 11 & - & 14 \\ x_6 & \infty & \infty & \infty & \infty & \infty & - \end{array} \right)$$

$$12) \left(\begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & - & 9 & 7 & 13 & \infty & \infty \\ x_2 & \infty & - & \infty & \infty & 15 & \infty \\ x_3 & \infty & 5 & - & \infty & \infty & \infty \\ x_4 & \infty & 6 & 7 & - & 8 & 10 \\ x_5 & \infty & \infty & \infty & \infty & - & 12 \\ x_6 & \infty & \infty & \infty & \infty & \infty & - \end{array} \right)$$

$$13) \left(\begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & - & 4 & 8 & \infty & \infty & \infty \\ x_2 & \infty & - & \infty & 3 & \infty & 10 \\ x_3 & \infty & 3 & - & 4 & 3 & \infty \\ x_4 & \infty & \infty & \infty & - & \infty & 4 \\ x_5 & \infty & 2 & \infty & 5 & - & 7 \\ x_6 & \infty & \infty & \infty & \infty & \infty & - \end{array} \right)$$

$$14) \left(\begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & - & 12 & 10 & \infty & 11 & \infty \\ x_2 & \infty & - & \infty & 10 & 7 & 15 \\ x_3 & \infty & 8 & - & 7 & 10 & \infty \\ x_4 & \infty & \infty & \infty & - & \infty & 11 \\ x_5 & \infty & \infty & \infty & 6 & - & 12 \\ x_6 & \infty & \infty & \infty & \infty & \infty & - \end{array} \right)$$

$$15) \left(\begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & - & 15 & 10 & \infty & \infty & \infty \\ x_2 & \infty & - & \infty & \infty & 12 & 18 \\ x_3 & \infty & 10 & - & 9 & 12 & 19 \\ x_4 & \infty & \infty & \infty & - & \infty & 13 \\ x_5 & \infty & \infty & \infty & 11 & - & 14 \\ x_6 & \infty & \infty & \infty & \infty & \infty & - \end{array} \right)$$

$$16) \left(\begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & - & 5 & 4 & \infty & 10 & \infty \\ x_2 & \infty & - & \infty & 8 & \infty & 13 \\ x_3 & \infty & 6 & - & 5 & 8 & \infty \\ x_4 & \infty & \infty & \infty & - & \infty & 8 \\ x_5 & \infty & \infty & \infty & 4 & - & 10 \\ x_6 & \infty & \infty & \infty & \infty & \infty & - \end{array} \right)$$

$$17) \left(\begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & - & 7 & 8 & 13 & \infty & \infty \\ x_2 & \infty & - & 9 & 7 & 12 & \infty \\ x_3 & \infty & \infty & - & 6 & 7 & 8 \\ x_4 & \infty & \infty & \infty & - & 9 & 17 \\ x_5 & \infty & \infty & \infty & \infty & - & 10 \\ x_6 & \infty & \infty & \infty & \infty & \infty & - \end{array} \right)$$

$$18) \left(\begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & - & 8 & \infty & 5 & 10 & \infty \\ x_2 & \infty & - & 7 & \infty & \infty & \infty \\ x_3 & \infty & \infty & - & \infty & \infty & 8 \\ x_4 & \infty & 4 & 6 & - & 5 & 11 \\ x_5 & \infty & 5 & 5 & \infty & - & 3 \\ x_6 & \infty & \infty & \infty & \infty & \infty & - \end{array} \right)$$

$$19) \left(\begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & - & 5 & 10 & 13 & \infty & \infty \\ x_2 & \infty & - & 8 & 9 & 13 & \infty \\ x_3 & \infty & \infty & - & 5 & 3 & 6 \\ x_4 & \infty & \infty & \infty & - & 8 & 10 \\ x_5 & \infty & \infty & \infty & \infty & - & 9 \\ x_6 & \infty & \infty & \infty & \infty & \infty & - \end{array} \right)$$

$$20) \left(\begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & - & \infty & 3 & \infty & \infty & \infty \\ x_2 & \infty & - & \infty & 4 & 6 & 8 \\ x_3 & \infty & 8 & - & 5 & 6 & 12 \\ x_4 & \infty & \infty & \infty & - & 5 & 7 \\ x_5 & \infty & \infty & \infty & \infty & - & 3 \\ x_6 & \infty & \infty & \infty & \infty & \infty & - \end{array} \right)$$

$$21) \left(\begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & - & 10 & 11 & 6 & \infty & \infty \\ x_2 & \infty & - & 13 & 8 & 11 & 17 \\ x_3 & \infty & \infty & - & 5 & 6 & 15 \\ x_4 & \infty & \infty & \infty & - & 7 & \infty \\ x_5 & \infty & \infty & \infty & \infty & - & 9 \\ x_6 & \infty & \infty & \infty & \infty & \infty & - \end{array} \right)$$

$$22) \left(\begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & - & 8 & 10 & \infty & \infty & \infty \\ x_2 & \infty & - & 10 & 9 & 12 & \infty \\ x_3 & \infty & \infty & - & 10 & 12 & 7 \\ x_4 & \infty & \infty & \infty & - & 9 & 13 \\ x_5 & \infty & \infty & \infty & \infty & - & 11 \\ x_6 & \infty & \infty & \infty & \infty & \infty & - \end{array} \right)$$

$$23) \left(\begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & - & 6 & \infty & 9 & 11 & \infty \\ x_2 & \infty & - & 6 & \infty & \infty & \infty \\ x_3 & \infty & \infty & - & \infty & \infty & 6 \\ x_4 & \infty & 5 & 8 & - & 6 & 14 \\ x_5 & \infty & 7 & 5 & \infty & - & 10 \\ x_6 & \infty & \infty & \infty & \infty & \infty & - \end{array} \right)$$

$$24) \left(\begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & - & 4 & 11 & \infty & \infty & \infty \\ x_2 & \infty & - & \infty & 3 & \infty & 10 \\ x_3 & \infty & 4 & - & 4 & 3 & \infty \\ x_4 & \infty & \infty & \infty & - & \infty & 4 \\ x_5 & \infty & 2 & \infty & 5 & - & 7 \\ x_6 & \infty & \infty & \infty & \infty & \infty & - \end{array} \right)$$

Задание 3

На одной и той же поточной линии предприятие может обрабатывать пять видов деталей. Время наладки при переходе линии на обработку с одного вида деталей к другому представлена матрицей $\Omega = [\omega_{ij}]$, где ω_{ij} – затраты времени на наладку переменного-поточной линии при переходе к обработке деталей от i -того вида к j -тому, время дано в часах. С помо-

щью алгоритма Литтла найти последовательность перестройки линии с одной детали на другую, при которой должны быть обеспечены минимальные общие потери рабочего времени на ее переналадку.

$$1) \quad \Omega = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 9 & 10 & 11 & 12 \\ 2 & 9 & \infty & 8 & 9 & 10 \\ 3 & 10 & 8 & \infty & 13 & 13 \\ 4 & 11 & 9 & 13 & \infty & 7 \\ 5 & 12 & 10 & 13 & 7 & \infty \end{array} \right)$$

$$2) \quad \Omega = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 11 & 11 & 10 & 9 \\ 2 & 11 & \infty & 10 & 12 & 11 \\ 3 & 11 & 10 & \infty & 8 & 10 \\ 4 & 10 & 12 & 8 & \infty & 8 \\ 5 & 9 & 11 & 10 & 8 & \infty \end{array} \right)$$

$$3) \quad \Omega = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 9 & 10 & 9 & 12 \\ 2 & 9 & \infty & 11 & 10 & 8 \\ 3 & 10 & 11 & \infty & 12 & 8 \\ 4 & 9 & 10 & 12 & \infty & 11 \\ 5 & 12 & 8 & 8 & 11 & \infty \end{array} \right)$$

$$4) \quad \Omega = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 10 & 11 & 11 & 12 \\ 2 & 10 & \infty & 9 & 9 & 10 \\ 3 & 11 & 9 & \infty & 12 & 8 \\ 4 & 11 & 9 & 12 & \infty & 7 \\ 5 & 12 & 10 & 8 & 7 & \infty \end{array} \right)$$

$$5) \quad \Omega = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 9 & 8 & 11 & 10 \\ 2 & 9 & \infty & 11 & 9 & 13 \\ 3 & 8 & 11 & \infty & 10 & 8 \\ 4 & 11 & 9 & 10 & \infty & 7 \\ 5 & 10 & 13 & 8 & 7 & \infty \end{array} \right)$$

$$6) \quad \Omega = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 11 & 10 & 9 & 12 \\ 2 & 11 & \infty & 14 & 12 & 10 \\ 3 & 10 & 14 & \infty & 10 & 13 \\ 4 & 9 & 12 & 10 & \infty & 9 \\ 5 & 12 & 10 & 13 & 9 & \infty \end{array} \right)$$

$$7) \quad \Omega = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 10 & 13 & 8 & 12 \\ 2 & 10 & \infty & 10 & 9 & 10 \\ 3 & 13 & 10 & \infty & 11 & 13 \\ 4 & 8 & 9 & 11 & \infty & 6 \\ 5 & 12 & 10 & 13 & 6 & \infty \end{array} \right)$$

$$8) \quad \Omega = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 9 & 10 & 11 & 10 \\ 2 & 9 & \infty & 10 & 7 & 10 \\ 3 & 10 & 10 & \infty & 12 & 8 \\ 4 & 11 & 7 & 12 & \infty & 7 \\ 5 & 10 & 10 & 8 & 7 & \infty \end{array} \right)$$

$$9) \quad \Omega = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 11 & 10 & 11 & 14 \\ 2 & 11 & \infty & 9 & 9 & 10 \\ 3 & 10 & 9 & \infty & 12 & 8 \\ 4 & 11 & 9 & 12 & \infty & 7 \\ 5 & 14 & 10 & 8 & 7 & \infty \end{array} \right)$$

$$10) \quad \Omega = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 11 & 10 & 12 & 8 \\ 2 & 11 & \infty & 9 & 11 & 10 \\ 3 & 10 & 9 & \infty & 13 & 9 \\ 4 & 12 & 11 & 13 & \infty & 6 \\ 5 & 8 & 10 & 9 & 6 & \infty \end{array} \right)$$

$$11) \quad \Omega = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 12 & 8 & 11 & 9 \\ 2 & 12 & \infty & 8 & 10 & 10 \\ 3 & 8 & 8 & \infty & 13 & 12 \\ 4 & 11 & 10 & 13 & \infty & 10 \\ 5 & 9 & 10 & 12 & 10 & \infty \end{array} \right)$$

$$12) \quad \Omega = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 11 & 15 & 11 & 9 \\ 2 & 11 & \infty & 11 & 7 & 10 \\ 3 & 15 & 11 & \infty & 9 & 13 \\ 4 & 11 & 7 & 9 & \infty & 10 \\ 5 & 9 & 10 & 13 & 10 & \infty \end{array} \right)$$

$$13) \quad \Omega = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 9 & 12 & 9 & 11 \\ 2 & 9 & \infty & 10 & 9 & 10 \\ 3 & 12 & 10 & \infty & 7 & 13 \\ 4 & 9 & 9 & 7 & \infty & 10 \\ 5 & 11 & 10 & 13 & 10 & \infty \end{array} \right)$$

$$14) \quad \Omega = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 11 & 10 & 6 & 12 \\ 2 & 11 & \infty & 8 & 16 & 9 \\ 3 & 10 & 8 & \infty & 12 & 7 \\ 4 & 6 & 16 & 12 & \infty & 10 \\ 5 & 12 & 9 & 7 & 10 & \infty \end{array} \right)$$

$$15) \quad \Omega = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 9 & 10 & 11 & 10 \\ 2 & 9 & \infty & 6 & 9 & 11 \\ 3 & 10 & 6 & \infty & 8 & 13 \\ 4 & 11 & 9 & 8 & \infty & 11 \\ 5 & 10 & 11 & 13 & 11 & \infty \end{array} \right)$$

$$16) \quad \Omega = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 12 & 10 & 7 & 12 \\ 2 & 12 & \infty & 9 & 9 & 10 \\ 3 & 10 & 9 & \infty & 10 & 13 \\ 4 & 7 & 9 & 10 & \infty & 11 \\ 5 & 12 & 10 & 13 & 11 & \infty \end{array} \right)$$

$$17) \quad \Omega = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 10 & 9 & 11 & 11 \\ 2 & 10 & \infty & 8 & 11 & 10 \\ 3 & 9 & 8 & \infty & 9 & 13 \\ 4 & 11 & 11 & 9 & \infty & 10 \\ 5 & 11 & 10 & 13 & 10 & \infty \end{array} \right)$$

$$18) \quad \Omega = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 11 & 10 & 11 & 9 \\ 2 & 11 & \infty & 9 & 10 & 10 \\ 3 & 10 & 9 & \infty & 6 & 13 \\ 4 & 11 & 10 & 6 & \infty & 8 \\ 5 & 9 & 10 & 13 & 8 & \infty \end{array} \right)$$

$$19) \quad \Omega = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 10 & 11 & 11 & 12 \\ 2 & 10 & \infty & 14 & 12 & 9 \\ 3 & 11 & 14 & \infty & 13 & 9 \\ 4 & 11 & 12 & 13 & \infty & 12 \\ 5 & 12 & 9 & 12 & 12 & \infty \end{array} \right)$$

$$20) \quad \Omega = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 10 & 12 & 9 & 10 \\ 2 & 10 & \infty & 14 & 11 & 10 \\ 3 & 12 & 14 & \infty & 9 & 13 \\ 4 & 9 & 11 & 9 & \infty & 15 \\ 5 & 10 & 10 & 13 & 15 & \infty \end{array} \right)$$

$$21) \quad \Omega = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 9 & 13 & 10 & 11 \\ 2 & 9 & \infty & 12 & 12 & 13 \\ 3 & 13 & 12 & \infty & 9 & 11 \\ 4 & 10 & 12 & 9 & \infty & 12 \\ 5 & 11 & 13 & 11 & 12 & \infty \end{array} \right)$$

$$22) \quad \Omega = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 10 & 13 & 11 & 10 \\ 2 & 10 & \infty & 10 & 9 & 10 \\ 3 & 13 & 10 & \infty & 9 & 11 \\ 4 & 11 & 9 & 9 & \infty & 17 \\ 5 & 10 & 10 & 11 & 17 & \infty \end{array} \right)$$

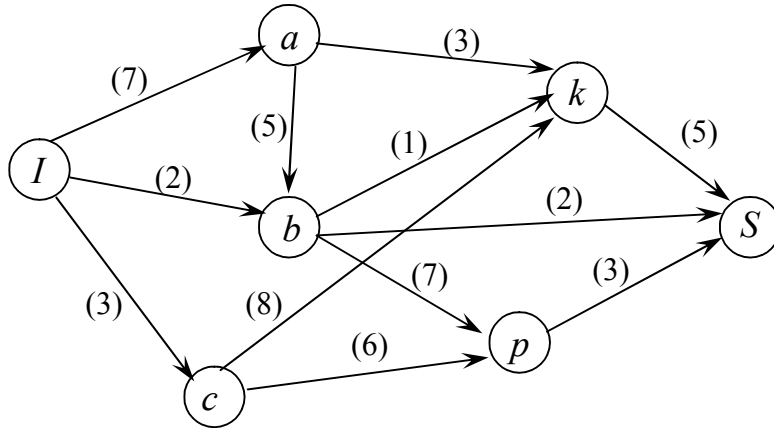
$$23) \quad \Omega = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 9 & 9 & 11 & 10 \\ 2 & 9 & \infty & 10 & 7 & 10 \\ 3 & 9 & 10 & \infty & 12 & 8 \\ 4 & 11 & 7 & 12 & \infty & 7 \\ 5 & 10 & 10 & 8 & 7 & \infty \end{array} \right)$$

$$24) \quad \Omega = \left(\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & \infty & 11 & 8 & 11 & 9 \\ 2 & 11 & \infty & 8 & 10 & 10 \\ 3 & 8 & 8 & \infty & 13 & 12 \\ 4 & 11 & 10 & 13 & \infty & 10 \\ 5 & 9 & 10 & 12 & 10 & \infty \end{array} \right)$$

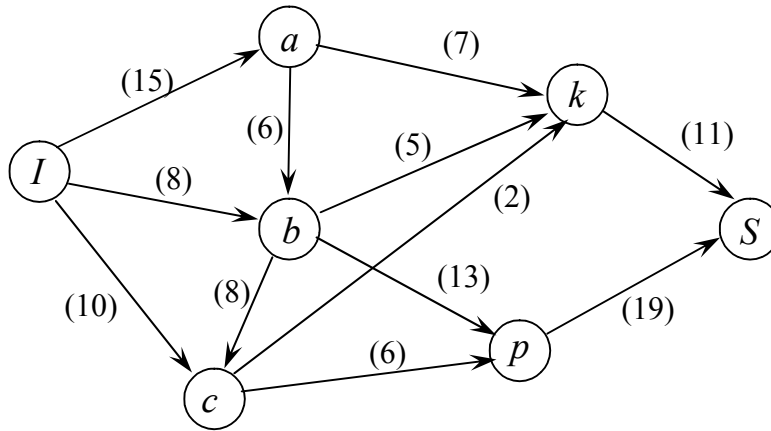
Задание 4

На данной сети в скобках указаны пропускные способности дуг. Требуется сформировать в сети поток максимальной мощности, направленный из истока I в сток S , и выписать дуги, образующие на сети разрез минимальной пропускной способности.

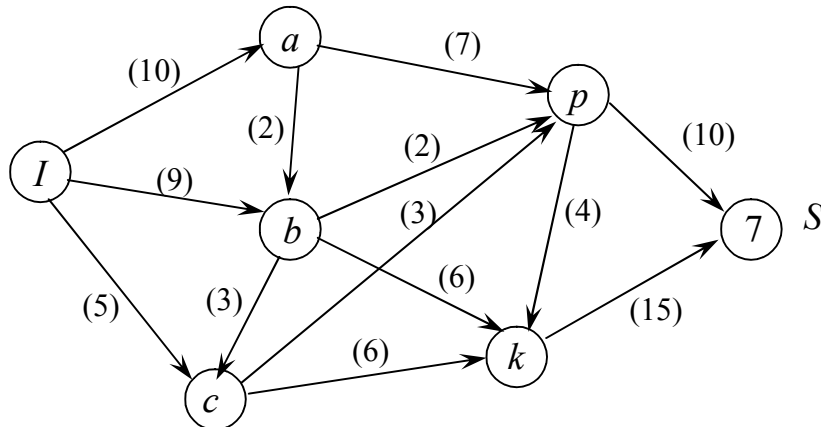
1)

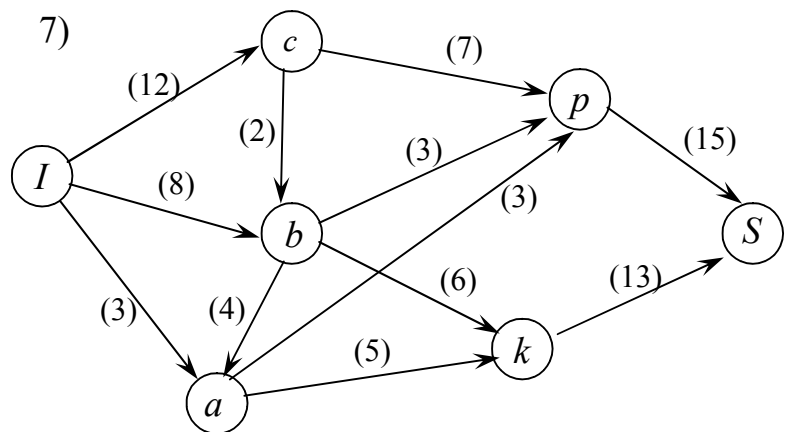
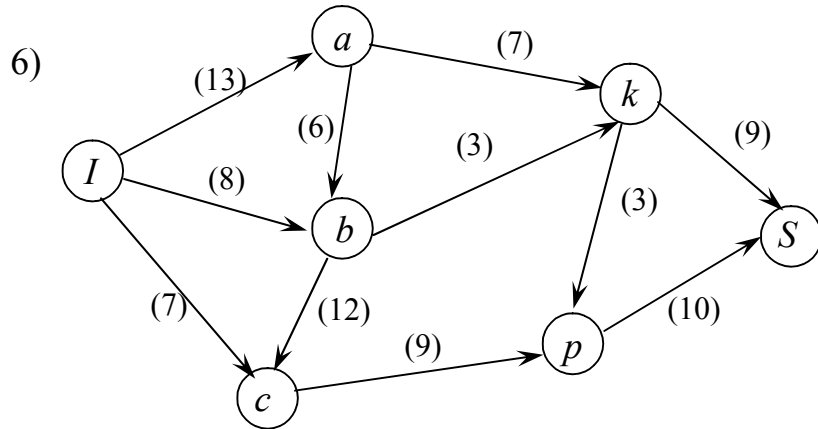
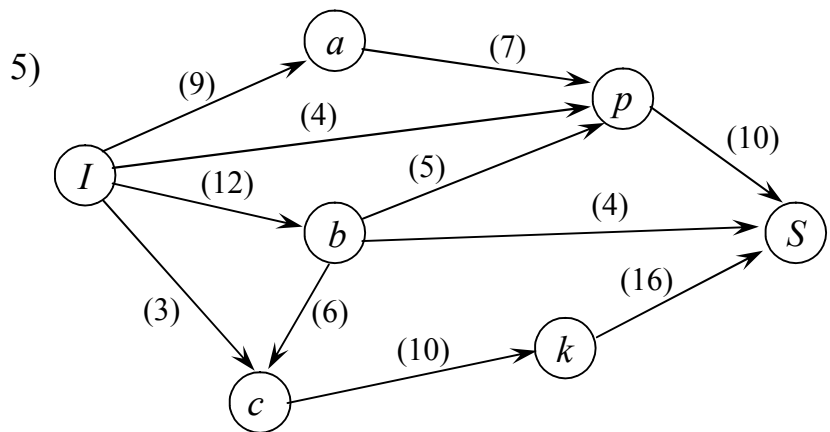
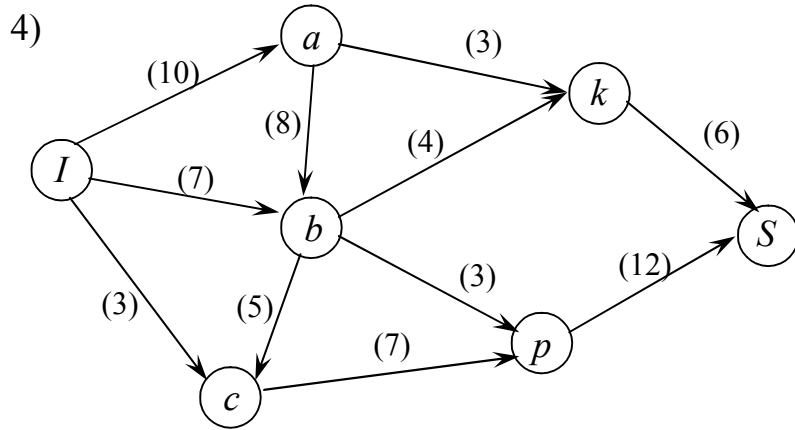


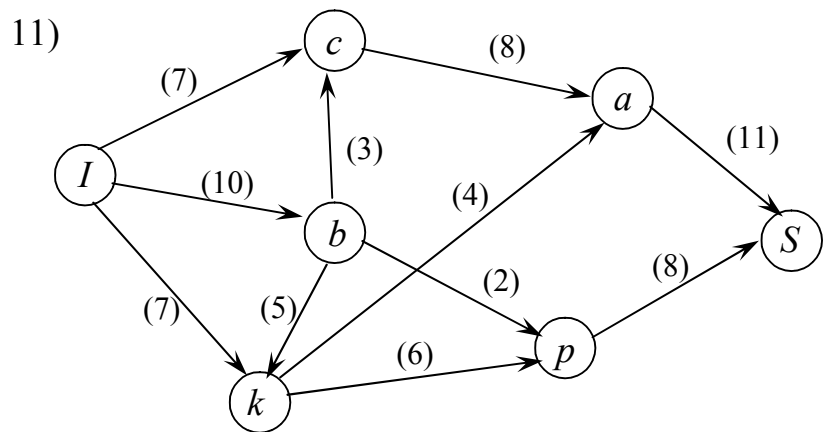
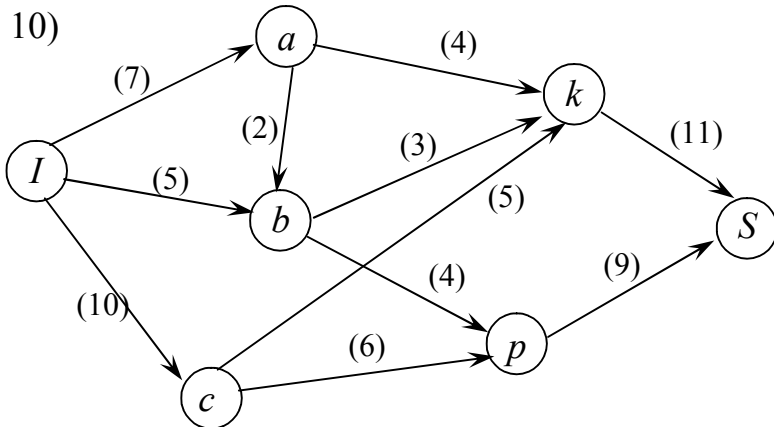
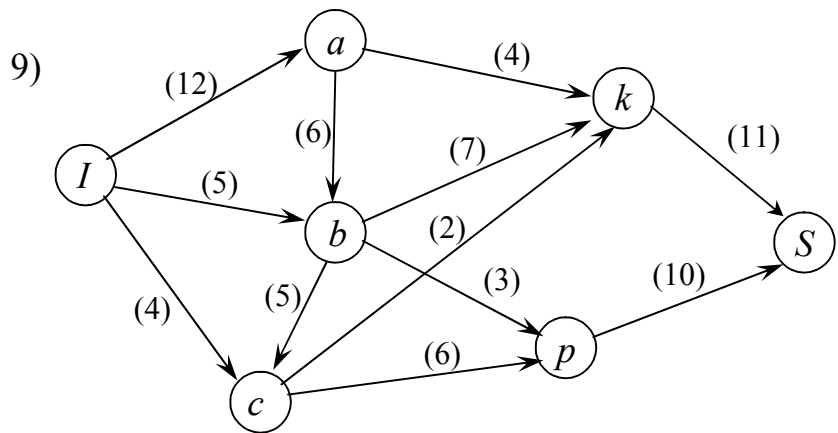
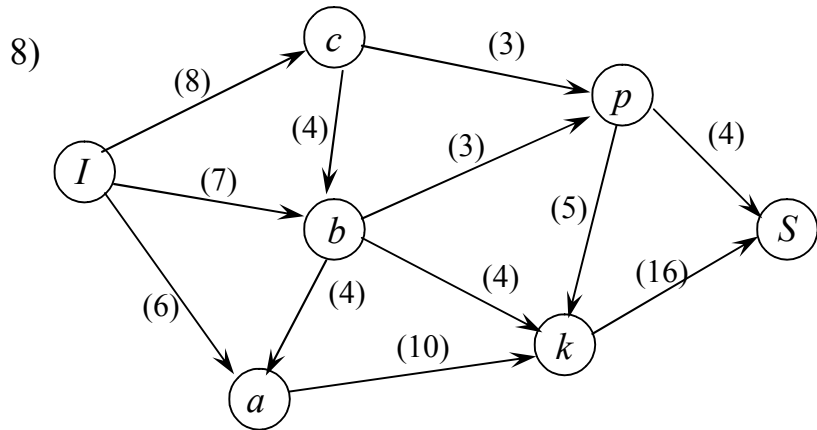
2)

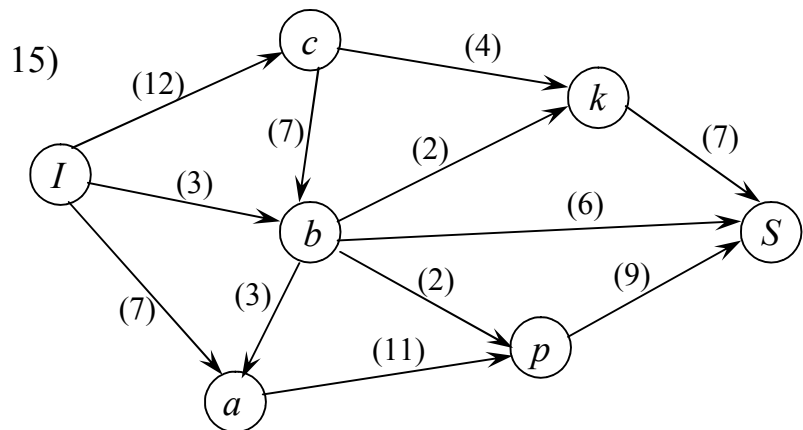
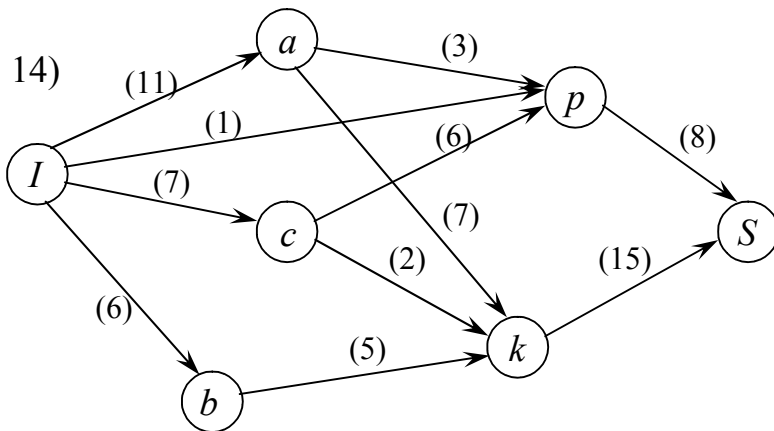
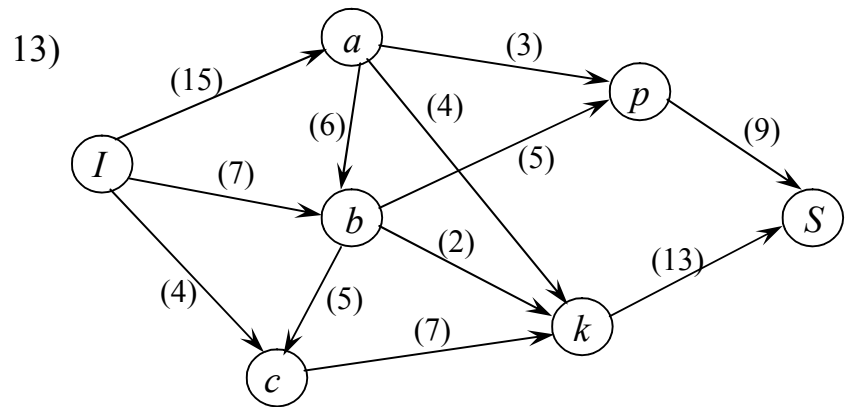
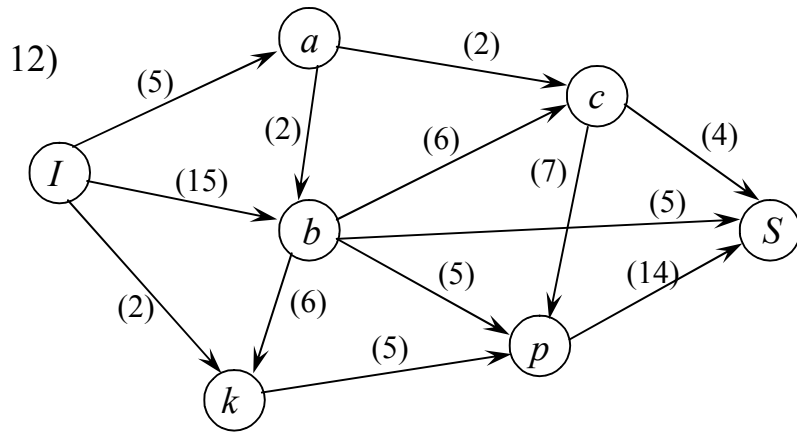


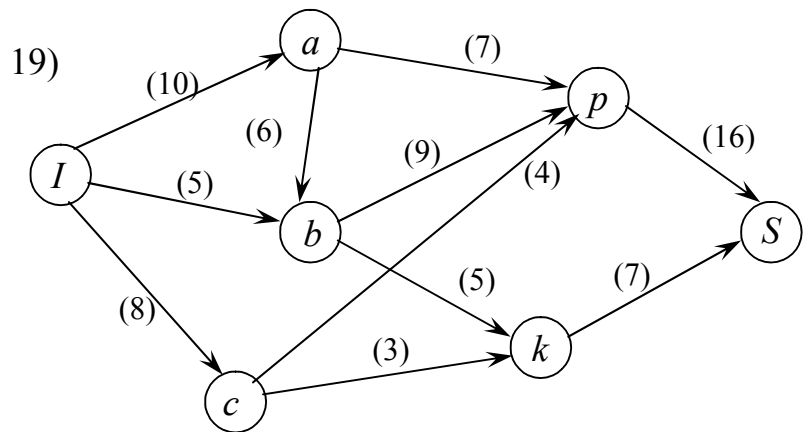
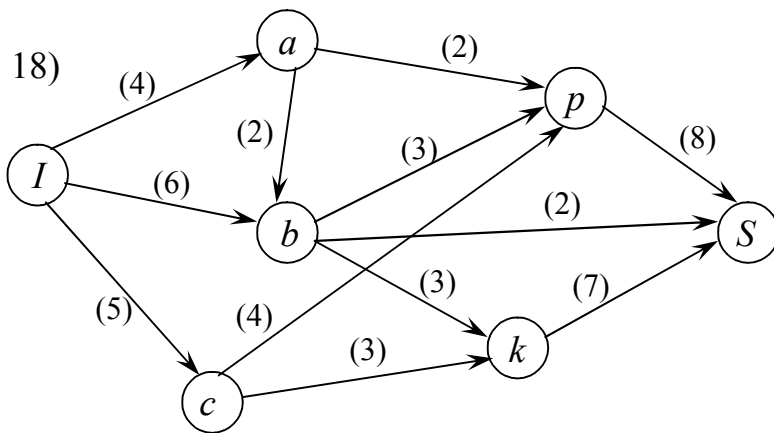
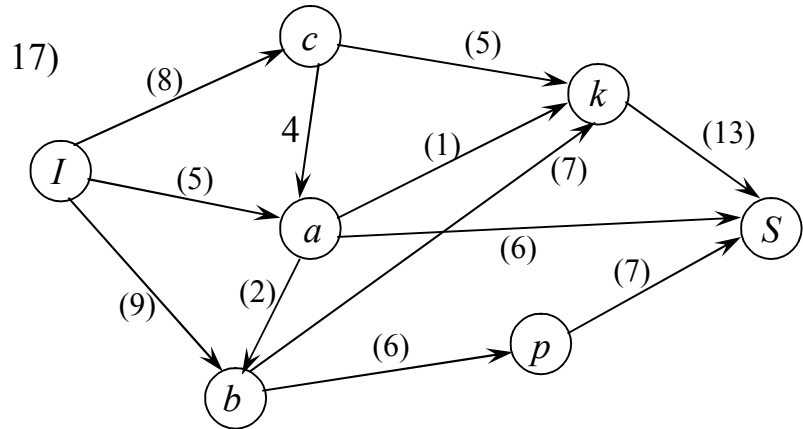
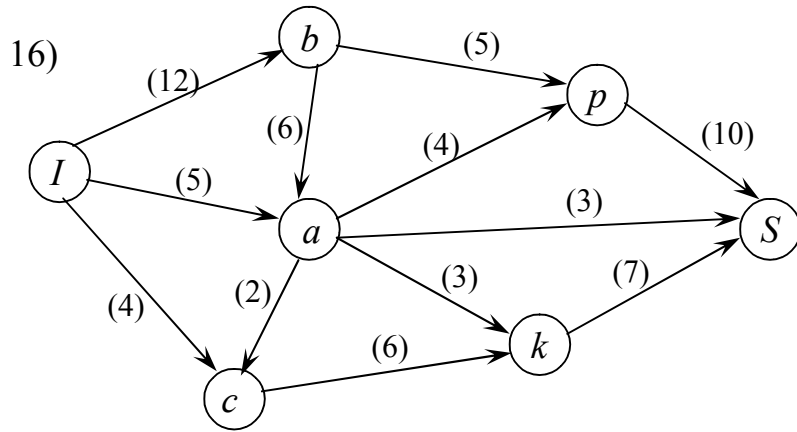
3)



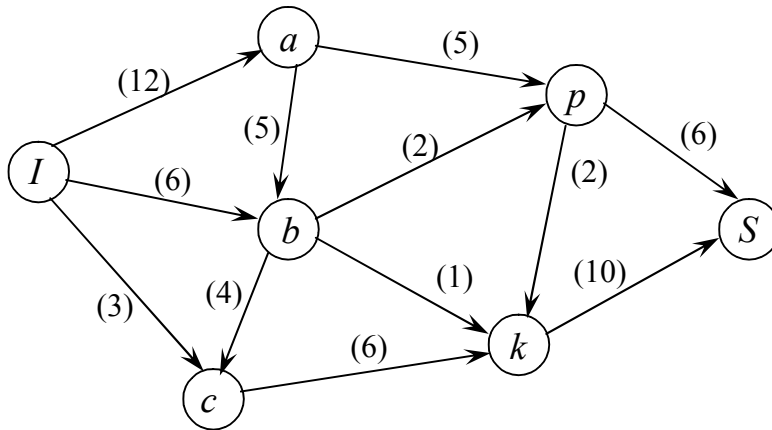




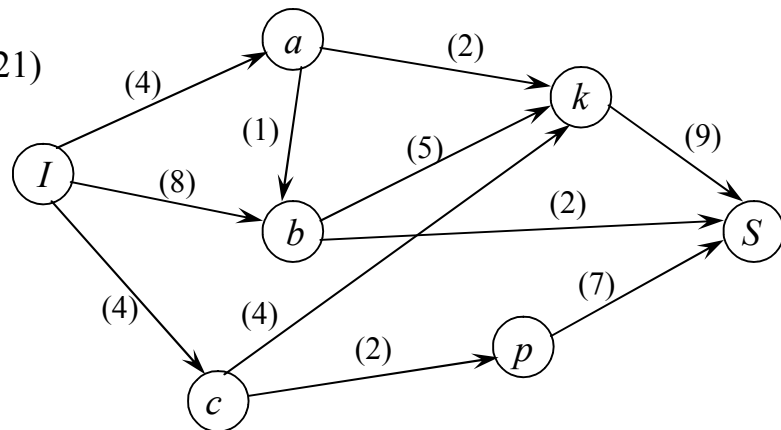




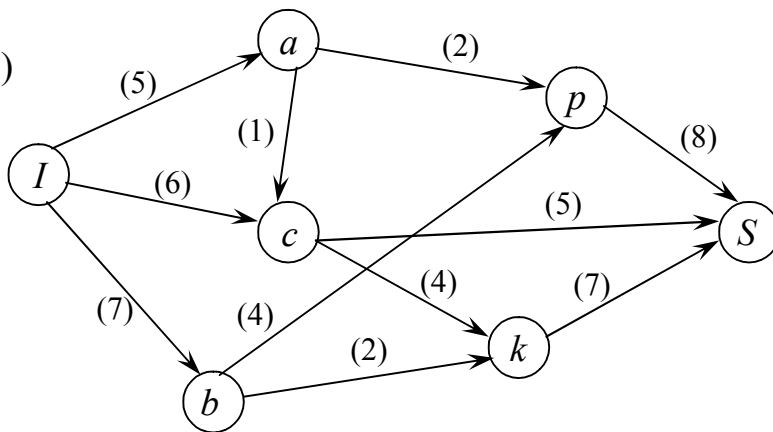
20)



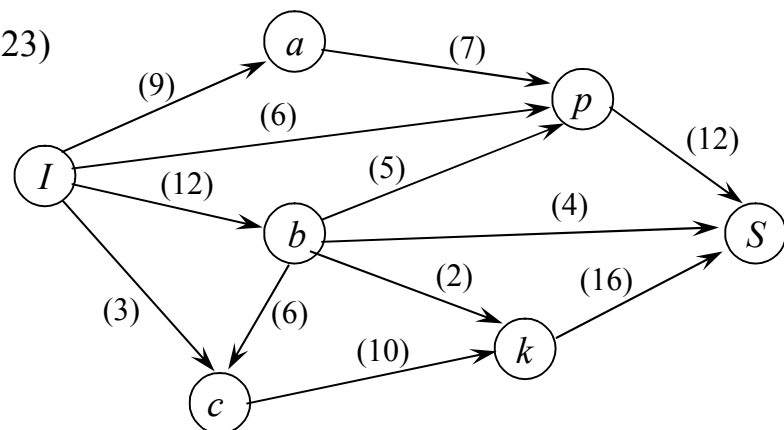
21)



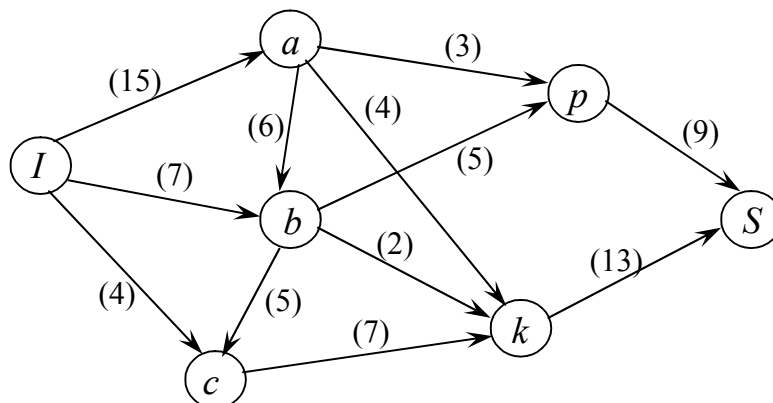
22)



23)



24)



Задание 5

По приведенному перечню работ и их взаимной последовательности построить сетевой график, определить критический срок, ранние и поздние сроки свершения событий, резервы времени событий, ранние и поздние сроки начала и окончания всех работ, а также полные и свободные резервы времени всех работ.

По сетевому графику и объему потребления ресурса для каждой работы построить линейный график с учетом ресурсных ограничений. Максимальный расход ресурса в любой момент времени составляет Res_{max} .

$$Res_{max} = 100$$

Таблица варианта 1

Основные работы	Работы, предшествующие основной	Длительность основных работ	Интенсивность потребления ресурса
a_1	—	5	20
a_2	—	8	40
a_3	—	11	50
a_4	a_1	6	30
a_5	a_1, a_2	12	30
a_6	a_1, a_2, a_3	18	70
a_7	a_4, a_5, a_6	7	80

$$Res_{max} = 100$$

Таблица варианта 2

Основные работы	Работы, предшествующие основной	Длительность основных работ	Интенсивность потребления ресурса
a_1	–	9	70
a_2	–	5	30
a_3	a_1	11	50
a_4	a_1, a_2	7	30
a_5	a_1, a_2	4	30
a_6	a_3, a_4	13	40
a_7	a_1, a_5	15	50

$$Res_{max} = 100$$

Таблица варианта 3

Основные работы	Работы, предшествующие основной	Длительность основных работ	Интенсивность потребления ресурса
a_1	–	7	30
a_2	–	11	40
a_3	–	5	60
a_4	a_1	10	20
a_5	a_1, a_2, a_3	4	40
a_6	a_3	13	30
a_7	a_3, a_4, a_5	12	70
a_8	a_4	8	20

$$Res_{max} = 50$$

Таблица варианта 4

Основные работы	Работы, предшествующие основной	Длительность основных работ	Интенсивность потребления ресурса
a_1	–	3	10
a_2	–	6	30
a_3	–	4	40
a_4	a_1	6	20
a_5	a_1, a_2, a_3	4	20
a_6	a_3	7	30
a_7	a_3, a_4, a_5	5	40

$$Res_{max} = 50$$

Таблица варианта 5

Основные работы	Работы, предшествующие основной	Длительность основных работ	Интенсивность потребления ресурса
a_1	–	10	30
a_2	–	12	20
a_3	–	9	30
a_4	a_1	6	10
a_5	a_1, a_3	7	20
a_6	a_2, a_4	9	30
a_7	a_1, a_2, a_4, a_5	5	10

$$Res_{max} = 100$$

Таблица варианта 6

Основные работы	Работы, предшествующие основной	Длительность основных работ	Интенсивность потребления ресурса
a_1	–	5	60
a_2	–	6	40
a_3	a_1	9	50
a_4	a_1, a_2	11	20
a_5	a_1, a_2	4	60
a_6	a_3	6	20
a_7	a_3	10	30
a_8	a_1, a_4, a_6	8	40
a_9	a_1, a_4, a_5, a_6	12	80

$$Res_{max} = 100$$

Таблица варианта 7

Основные работы	Работы, предшествующие основной	Длительность основных работ	Интенсивность потребления ресурса
a_1	–	8	20
a_2	a_1	4	30
a_3	–	10	20
a_4	–	13	60
a_5	a_1, a_3	3	10
a_6	a_2, a_4	6	40
a_7	a_1, a_4, a_5	5	50

$$Res_{max} = 50$$

Таблица варианта 8

Основные работы	Работы, предшествующие основной	Длительность основных работ	Интенсивность потребления ресурса
a_1	–	4	30
a_2	–	6	30
a_3	a_1, a_2	12	20
a_4	a_1, a_2	4	30
a_5	a_2	5	20
a_6	a_3, a_4, a_5	10	50
a_7	a_3	7	10

$$Res_{max} = 50$$

Таблица варианта 9

Основные работы	Работы, предшествующие основной	Длительность основных работ	Интенсивность потребления ресурса
a_1	–	5	30
a_2	–	7	20
a_3	–	10	30
a_4	a_1, a_2	5	10
a_5	a_2, a_3	7	50
a_6	a_3	10	20
a_7	a_3, a_4, a_5	5	30

$$Res_{max} = 50$$

Таблица варианта 10

Основные работы	Работы, предшествующие основной	Длительность основных работ	Интенсивность потребления ресурса
a_1	–	5	10
a_2	–	6	40
a_3	a_1	8	20
a_4	a_1, a_2	10	50
a_5	a_2, a_3	7	30
a_6	a_1, a_3, a_4	5	20
a_7	a_4	7	30

$$Res_{max} = 100$$

Таблица варианта 11

Основные работы	Работы, предшествующие основной	Длительность основных работ	Интенсивность потребления ресурса
a_1	–	5	30
a_2	–	7	20
a_3	–	8	70
a_4	a_3	5	50
a_5	a_2, a_4	10	30
a_6	a_1, a_2, a_3	12	40
a_7	a_2, a_5	4	80

$$Res_{max} = 50$$

Таблица варианта 12

Основные работы	Работы, предшествующие основной	Длительность основных работ	Интенсивность потребления ресурса
a_1	–	8	10
a_2	–	10	40
a_3	a_1	8	10
a_4	a_1, a_2	12	30
a_5	a_2	6	40
a_6	a_3, a_4, a_5	9	30
a_7	a_3, a_4	7	10

$$Res_{max} = 100$$

Таблица варианта 13

Основные работы	Работы, предшествующие основной	Длительность основных работ	Интенсивность потребления ресурса
a_1	–	4	40
a_2	–	4	50
a_3	a_2	6	100
a_4	a_1, a_2	7	20
a_5	a_1, a_3	10	40
a_6	a_2, a_4, a_5	5	80
a_7	a_2	8	60

$$Res_{max} = 100$$

Таблица варианта 14

Основные работы	Работы, предшествующие основной	Длительность основных работ	Интенсивность потребления ресурса
a_1	–	8	60
a_2	–	5	40
a_3	–	6	30
a_4	a_1, a_2	10	50
a_5	a_2	10	30
a_6	a_2, a_3	8	50
a_7	a_3, a_4, a_6	5	70

$$Res_{max} = 50$$

Таблица варианта 15

Основные работы	Работы, предшествующие основной	Длительность основных работ	Интенсивность потребления ресурса
a_1	–	7	20
a_2	–	9	40
a_3	a_1, a_2	11	30
a_4	a_2	6	20
a_5	a_3, a_4	8	20
a_6	a_2	10	10
a_7	a_3, a_4, a_6	12	20
a_8	a_2, a_5	13	30

$$Res_{max} = 50$$

Таблица варианта 16

Основные работы	Работы, предшествующие основной	Длительность основных работ	Интенсивность потребления ресурса
a_1	–	6	20
a_2	–	10	40
a_3	a_2	10	20
a_4	a_1, a_2	8	20
a_5	a_3	6	30
a_6	a_2, a_4, a_5	4	50
a_7	a_2	3	30

$$Res_{max} = 50$$

Таблица варианта 17

Основные работы	Работы, предшествующие основной	Длительность основных работ	Интенсивность потребления ресурса
a_1	–	5	20
a_2	–	7	30
a_3	–	8	20
a_4	a_3	8	40
a_5	a_2, a_4	10	40
a_6	a_1, a_2, a_3	5	20
a_7	a_2, a_5	7	30

$$Res_{max} = 50$$

Таблица варианта 18

Основные работы	Работы, предшествующие основной	Длительность основных работ	Интенсивность потребления ресурса
a_1	–	5	40
a_2	–	4	20
a_3	a_1	6	50
a_4	a_1, a_2	5	10
a_5	a_2, a_3	7	30
a_6	a_2, a_3, a_4	9	20
a_7	a_4	4	20
a_8	a_5, a_6	7	30

$$Res_{max} = 50$$

Таблица варианта 19

Основные работы	Работы, предшествующие основной	Длительность основных работ	Интенсивность потребления ресурса
a_1	–	6	40
a_2	–	8	50
a_3	a_2	12	10
a_4	a_1, a_2	5	10
a_5	a_1, a_3	7	40
a_6	a_2, a_4, a_5	7	50
a_7	a_2	7	40

$$Res_{max} = 50$$

Таблица варианта 20

Основные работы	Работы, предшествующие основной	Длительность основных работ	Интенсивность потребления ресурса
a_1	–	7	30
a_2	–	5	20
a_3	–	10	50
a_4	a_3	14	40
a_5	a_2, a_4	8	10
a_6	a_1, a_2, a_3	10	40
a_7	a_2, a_5	6	40

$$Res_{max} = 50$$

Таблица варианта 21

Основные работы	Работы, предшествующие основной	Длительность основных работ	Интенсивность потребления ресурса
a_1	–	6	10
a_2	–	9	20
a_3	–	8	30
a_4	a_1	5	20
a_5	a_3	3	10
a_6	a_3	7	30
a_7	a_3, a_4	9	30
a_8	a_1, a_2, a_5, a_6	11	30
a_9	a_6	10	10

$$Res_{max} = 50$$

Таблица варианта 22

Основные работы	Работы, предшествующие основной	Длительность основных работ	Интенсивность потребления ресурса
a_1	–	3	10
a_2	–	6	30
a_3	–	4	40
a_4	a_1	6	20
a_5	a_1, a_2, a_3	5	30
a_6	a_3	7	30
a_7	a_3, a_4, a_5	5	40

$$Res_{max} = 100$$

Таблица варианта 23

Основные работы	Работы, предшествующие основной	Длительность основных работ	Интенсивность потребления ресурса
a_1	–	11	20
a_2	–	9	30
a_3	–	7	60
a_4	a_2	5	50
a_5	a_1	6	40
a_6	a_3, a_4	8	30
a_7	a_2, a_3, a_4, a_5	10	60
a_8	a_6	13	60
a_9	a_1, a_7, a_8	15	90

$$Res_{max} = 100$$

Таблица варианта 24

Основные работы	Работы, предшествующие основной	Длительность основных работ	Интенсивность потребления ресурса
a_1	–	8	60
a_2	–	5	40
a_3	–	6	30
a_4	a_1, a_2	9	40
a_5	a_2	10	30
a_6	a_2, a_3	8	50
a_7	a_3, a_4, a_6	5	70

РАЗДЕЛ 4

ЭЛЕМЕНТЫ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ

1. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ

Среди многочисленных проблем современного производства и социальной сферы самой актуальной является, пожалуй, проблема управления. Промышленные и сельскохозяйственные предприятия, опытно-конструкторские, научно-исследовательские организации, научно-производственные объединения, предприятия сферы обслуживания и организации производственной и непроизводственной сфер являются сложными человеко-машинными системами, эффективность работы которых во многом зависит от качественного уровня управления ими. Чтобы обеспечить высокий уровень управления, руководителю теперь недостаточно интуиции, личного опыта и хороших организаторских способностей. При принятии стратегических и тактических решений ему приходится учитывать многочисленные факторы, влияющие на эффективность достижения конечной цели, и опираться на сложные и порой противоречивые критерии. Неоценимую помощь руководителю при выработке и принятии наилучших управленческих решений наряду с другими подходами и методами оказывает такой раздел математики, как *исследование операций*.

Исследование операций, как самостоятельный раздел, появилось сравнительно недавно и в настоящее время представляет собой одну из быстро развивающихся отраслей науки, находящую все более широкие области применения.

Для уяснения понятия «исследование операций» воспользуемся двумя определениями.

1. *Исследованием операций* называется теория математических моделей принятия оптимальных решений и практика их использования.

2. *Исследование операций* представляет собой искусство давать плохие ответы на те практические вопросы, на которые даются еще худшие ответы другими способами.

Первое определение отражает математическую сторону вопроса, поэтому иначе может быть сформулировано следующим образом: под *исследованием операций* понимается применение математических, количествен-

ных методов для обоснования решений, принимаемых в тех областях целенаправленной человеческой деятельности, в которых существуют различные пути, ведущие к достижению цели.

Суть второго определения, данного в шутку специалистом в области исследования операций Т.Л. Саати, заключается в том, что практические ситуации, в которых приходится принимать решения, бывают настолько сложными и важными, что даже незначительная «помощь» математических методов является весьма существенной.

Подводя итог сказанному, можно дать следующую трактовку понятию «исследование операций».

Исследование операций может быть охарактеризовано как применение научных принципов, методов и средств к задачам, связанным с функционированием организационных систем, с целью предоставить тем, кто управляет этими системами, оптимальные решения.

В настоящее время под операциями понимают совокупность действий, мероприятий, объединенных общим замыслом и направленных на достижение цели. Примеры операций: система мероприятий, направленных на выполнение производственного плана промышленным предприятием; организация поиска неисправности технического устройства; разработка маршрутов движения транспорта и т.д.

Некоторые классы таких задач уже встречались ранее при изучении раздела «Элементы теории графов»: задача коммивояжера (метод ветвей и границ); задача о формировании в сети максимального потока; задача сетевого планирования.

Из всех задач, рассматриваемых в исследовании операций, особое внимание уделяется тем, в которых активно участвуют ограничения на область применения переменных. Такие задачи были выделены в особую область математики, которая получила название *математическое программирование*.

Математическое программирование – область математики, разрабатывающая теорию и численные методы решения многомерных экстремальных задач с ограничениями, т.е. задач на экстремум функции многих переменных с ограничениями на область изменения этих переменных.

В разделе «Элементы математического моделирования» будут рассмотрены некоторые задачи, которые нашли применение в организации производства в машиностроении.

1.1. Математическая модель

Для применения численных методов решения экстремальных задач с ограничениями необходимо построить *математическую модель* исходной задачи.

Математическая модель – это символическая запись, отражающая определенные особенности изучаемых явлений или процессов. Формализованная на том или ином языке (в виде дифференциальных или разностных уравнений, неравенств, функций и т.д.) математическая модель отражает определенные свойства реальных процессов, протекающих в окружающем мире или в различных областях деятельности человека.

Требования, предъявляемые к создаваемым моделям, весьма противоречивы. С одной стороны, желательно, чтобы модель была достаточно простой (это значительно упрощает процесс ее исследования). С другой стороны, модель должна быть достаточно полной, т.е. в ней должны быть учтены все существенные факторы, оказывающие наибольшее влияние на течение и исход действий.

Модель задачи математического программирования включает:

1) *план задачи* (вектор управления, решение, стратегия, поведение и др.) – совокупность неизвестных величин $X = \bar{x} = (x_1; x_2; \dots; x_j; \dots; x_n)$, действуя на которые, систему можно совершенствовать;

2) *целевую функцию* (показатель эффективности, критерий оптимальности, функционал задачи и др.) – функция, экстремальное значение которой нужно найти в условиях экономических возможностей. Целевую функцию обозначим $Z = Z(X)$. Это может быть прибыль, объем выпуска или реализации, затраты производства, уровень обслуживания, число комплектов, отходы и т.д.;

3) *систему ограничений* (условий, налагаемых на неизвестные величины. Ограничениями являются материальные, финансовые и трудовые ресурсы, возможности технического, технологического и вообще научного потенциала. Математически ограничения выражаются в виде уравнений и неравенств. Их совокупность образует *область допустимых решений*. Объединение всех условий (ограничений), налагаемых на неизвестные (искомые) величины x_j задачи, обозначим буквой Ω ($X \in \Omega$).

В развернутом виде математическую модель можно представить следующим образом:

Найти план $X = (x_1; x_2; \dots; x_j; \dots; x_n)$, доставляющий экстремальное значение целевой функции Z , т.е.

$$Z = Z(x_1; x_2; \dots; x_j; \dots; x_n) \rightarrow \max (\min)$$

при ограничениях

$$\varphi_i(x_1; x_2; \dots; x_j; \dots; x_n) \{ \leq, =, \geq \} b_i \quad (i = \overline{1, m}).$$

Из экономических или физических соображений на план задачи или некоторые его компоненты (координаты), как правило, налагаются условия неотрицательности

$$x_j \geq 0,$$

иногда – целочисленности.

План X , удовлетворяющий системе ограничений задачи, называется *допустимым* ($X \in \Omega$). Допустимый план, доставляющий функции цели экстремальное значение, называется *оптимальным*. Оптимальный план будем обозначать X^* , экстремальное значение функции цели – $Z^* = Z(X^*)$. Оптимальное решение, вообще говоря, не обязательно единственное, возможны случаи, когда оно не существует, имеется конечное или бесчисленное множество оптимальных решений.

1.2. Классификация методов математического программирования

В зависимости от особенностей целевой функции $Z = Z(X)$ и функций, задающих ограничения $\varphi_i(X)$ ($i = \overline{1, m}$), задачи математического программирования делятся на ряд следующих типов:

1) *задачи линейного программирования* (ЗЛП), если целевая функция и функции, входящие в систему ограничений, линейны (первой степени) относительно входящих в задачу неизвестных x_j . Методы и модели линейного программирования широко применяются при оптимизации процессов во всех отраслях народного хозяйства: при разработке производственной программы предприятия, распределения ее по исполнителям, при размещении заказов между исполнителями по временным интервалам; при планировании грузопотоков; в задачах развития и размещения производительных сил, баз и складов систем обращения материальных ресурсов и т.д.

2) *задачи нелинейного программирования* (НЛП), если в задаче математического программирования целевая функция и (или) хотя бы одна из функций системы ограничений нелинейны. Методы НЛП получили широкое применение при расчете экономически выгодных партий запуска деталей в производство, при определении поставочного комплекта, размеров запаса, размещении производительных сил и т.д.

3) *задачи целочисленного (ЦП) или дискретного программирования*, если на все или некоторые переменные x_j наложено условие дискретности, например, целочисленности. Методами ЦП решается широкий круг задач оптимизации с неделимостями, комбинаторного типа, с логическими условиями, с разрывной целевой функцией и т.д. В частности, задача выбора (о назначениях), о контейнерных перевозках (о рюкзаке), о маршрутизации (задача коммивояжера), теории расписаний и т.д.

4) *задачи динамического программирования* (ДП), если параметры целевой функции и (или) системы ограничений изменяются во времени

или целевая функция имеет аддитивный $Z(X) = \sum_{j=1}^n Z_j(x_j)$ либо мульти-

пликативный $Z(X) = \prod_{j=1}^n Z_j(x_j)$ вид, или сам процесс решения имеет мно-

гошаговый характер. Методами ДП могут решаться задачи перспективного и текущего планирования, управления производством, поставками и запасами в условиях изменяющегося спроса, замены оборудования и т.д.

В перечисленных выше разделах математического программирования предполагается, что вся информация о протекании процессов известна и достоверна. Такие методы оптимизации называются *детерминированными* или методами обоснования решений в условиях определенности.

5) *задачи стохастического программирования* (СП), если параметры, входящие в функцию цели, или ограничения задачи являются случайными, недостоверными величинами или если приходится принимать решения в условиях риска, неполной или недостоверной информации. Сюда следует отнести методы и модели выработки решений в условиях конфликтных ситуаций (математическая теория игр), при неполной информации (экспертные оценки), в условиях риска (статистические решения) и др.

Позднее появились иные типы задач, учитывающих специфику целевой функции и системы ограничений, в связи с чем возникли параметрическое, дробно-линейное, блочное, сетевое (потокосное), многоиндексное, булевское, комбинаторное и другие типы программирования. В случае нели-

нейностей специфика задач породила квадратичное, биквадратичное, сепарабельное, выпуклое и другие типы программирования. Появились численные методы отыскания оптимальных решений: градиентные, штрафных и барьерных функций, возможных направлений, линейной аппроксимации, случайного поиска и др.

К математическому программированию относятся также методы решения экстремальных задач с бесконечным числом переменных – *бесконечномерное программирование*.

И, наконец, отметим, что задачи математического программирования с одной целевой функцией решаются методами скалярной оптимизации. Однако реальные ситуации настолько сложны, что нередко приходится одновременно учитывать несколько целевых функций, которые должны принимать экстремальные значения. Задачи, где находят решение по нескольким целевым функциям, относятся к векторной оптимизации – это так называемые *задачи многокритериального подхода*.

2. ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

2.1. Виды задач линейного программирования

Линейное программирование – раздел математического программирования, применяемый при разработке методов отыскания экстремума линейной функции нескольких переменных при линейных дополнительных ограничениях, налагаемых на переменные. По типу решаемых задач его методы разделяются на универсальные и специальные. С помощью универсальных методов могут решаться любые задачи линейного программирования. Специальные методы учитывают особенности модели задачи, ее целевой функции и системы ограничений.

Особенностью ЗЛП является то, что экстремума целевая функция достигает на границе области допустимых решений.

Задача о наилучшем использовании ресурсов

Пусть некоторая производственная единица (цех, завод, объединение и т.д.), исходя из конъюнктуры рынка, технических или технологических возможностей и имеющихся ресурсов, может выпускать n различных видов продукции (товаров), известных под номерами, обозначенными индек-

сом j ($j = \overline{1, n}$), где каждый j -тый вид продукции обозначим Π_j . Предприятие при производстве этих видов продукции должно ограничиваться имеющимися видами ресурсов, технологий, других производственных факторов (сырья, полуфабрикатов, рабочей силы, оборудования, электроэнергии и т.д.). Все виды этих ограничивающих факторов будем называть *ресурсами*. Пусть их число равно m ; поставим с соответствие им индекс i ($i = \overline{1, m}$). Ресурсы ограничены, и их количества равны соответственно $b_1, b_2, \dots, b_i, \dots, b_m$ условных единиц. Таким образом, $\bar{b} = (b_1, b_2, \dots, b_i, \dots, b_m)$ – вектор ресурсов.

Известна экономическая выгода (мера полезности) производства продукции каждого вида, исчисляемая, скажем, по отпускной цене товара, его прибыльности, издержкам производства и т.д. Примем в качестве такой меры, например, цену реализации c_j ($j = \overline{1, n}$), т.е. $\bar{c} = (c_1, c_2, \dots, c_j, \dots, c_n)$ – вектор цен.

Известны также технологические коэффициенты a_{ij} , которые указывают, сколько единиц i -того ресурса требуется для производства единицы продукции j -того вида. Матрицу коэффициентов a_{ij} называют *технологической* и обозначают буквой A , т.е. $A = [a_{ij}]_{m \times n}$.

Обозначим через $X = (x_1; x_2; \dots; x_j; \dots; x_n)$ план выпуска продукции видов $\Pi_1, \Pi_2, \dots, \Pi_j, \dots, \Pi_n$, которые обеспечивают предприятию максимум объема реализации при имеющихся объемах ресурсов.

Получаем следующую *математическую модель*: найти план выпуска $X = (x_1; x_2; \dots; x_j; \dots; x_n)$ продукции видов $\Pi_1, \Pi_2, \dots, \Pi_j, \dots, \Pi_n$, обеспечивающий максимум объема реализации в стоимостном выражении

$$Z = \sum_{j=1}^n c_j x_j \rightarrow \max, \quad (2.1)$$

при ограничениях на лимитируемые ресурсы

$$\sum_{j=1}^n a_{ij} x_j \leq b_i \quad (i = \overline{1, m}) \quad (2.2)$$

и при условиях неотрицательности

$$x_j \geq 0 \quad (j = \overline{1, n}), \quad (2.3)$$

где $\bar{c} = (c_1, c_2, \dots, c_j, \dots, c_n)$ – вектор цен, т.е. цена реализации единицы каждого вида продукции;

a_{ij} – технологический коэффициент, указывающий, сколько единиц i -того ресурса требуется для производства единицы продукции j -того вида;

$\bar{b} = (b_1, b_2, \dots, b_i, \dots, b_m)$ – вектор ресурсов, где i -тая компонента вектора соответствует количеству ресурса i -того вида.

Аналогичная математическая модель составляется для задачи о выборе оптимальных технологий.

Задача о раскрое материалов

Суть задачи об оптимальном раскрое состоит в разработке таких технологически допустимых планов раскроя, при которых получается необходимый комплект заготовок, а отходы (по длине, площади, объему, массе или стоимости) сводятся к минимуму. Рассмотрим простейшую модель раскроя по одному измерению. Более сложные постановки ведут к задачам целочисленного программирования.

Постановка задачи раскроя по одному измерению длиномерных материалов (прутков, труб, профильного проката и др.) следующая. Пусть имеется N штук исходного материала, длина каждой из которых равна L . Нужны заготовки m видов, длины которых равны l_i ($i = \overline{1, m}$). Известна потребность в заготовках каждого вида, которая составляет b_i . Изучение вопроса раскроя (построение технологической карты раскроя) показывает, что можно выделить n приемлемых вариантов раскроя исходного материала длиной L на заготовки длиной l_i . Обозначим через a_{ij} количество заготовок i -того вида, получаемое при раскрое единицы исходного материала по j -тому ($j = \overline{1, n}$) варианту, c_j – отходы при раскрое единицы исходного материала по j -тому варианту.

План задачи $X = (x_1; x_2; \dots; x_j; \dots; x_n)$, где x_j – количество единиц исходного материала, планируемое к раскрою по j -тому варианту.

Функция цели – минимум отходов, получаемых при раскрое:

$$Z = \sum_{j=1}^n c_j x_j \rightarrow \min, \quad (2.4)$$

при ограничениях на число единиц исходного материала

$$\sum_{j=1}^n x_j \leq N, \quad (2.5)$$

на удовлетворение ассортиментного спроса потребителей

$$\sum_{j=1}^n a_{ij} x_j \geq b_i \quad (i = \overline{1, m}), \quad (2.6)$$

и при условиях неотрицательности

$$x_j \geq 0 \quad (j = \overline{1, n}), \quad (2.7)$$

где c_j – отходы при раскросе единицы исходного материала по j -тому варианту;

a_{ij} – количество заготовок i -того вида, получаемое при раскросе единицы исходного материала по j -тому ($j = \overline{1, n}$) варианту;

N – количество штук исходного материала, длина каждой из которых равна L ;

b_i – потребность в заготовках каждого вида.

Сходные математические модели строятся для задачи о смесях, задачи о диете.

К задачам линейного программирования относится также и транспортная задача, формулировка и математическая модель которой будут приведены позже.

Рассмотрим примеры задач линейного программирования.

Пример 2.1. При изготовлении изделий I_1 и I_2 используются токарные и фрезерные станки, а также сталь и цветные металлы. По технологическим нормам на производство единицы изделия I_1 требуется 300 и 200 единиц соответственно токарного и фрезерного оборудования (в станко-часах), 10 и 20 единиц стали и цветных металлов (в килограммах). Для производства единицы изделия I_2 требуется 400, 100, 70 и 50 соответствующих единиц тех же ресурсов. Цех располагает 12400 и 6800 станко-ч оборудования, 640 и 840 кг материалов. Прибыль от реализации единицы изделия I_1 составляет 6 тыс. ден. ед., I_2 – 16 тыс. ден. ед. Требуется:

- 1) свести исходные данные в таблицу, удобную для построения модели;
- 2) составить математическую модель задачи (показатель эффективности – прибыль).

Решение.

1) Обозначим через x_1 число изделий I_1 , через x_2 – изделий I_2 , через Z – суммарную прибыль от реализации производственных изделий. Исходные данные удобно представить в виде таблицы.

Ресурсы	Затраты на единицу изделия		Объем ресурса	Вид ограничений
	I_1	I_2		
Станки, станко-ч токарные	300	400	12400	\leq
фрезерные	200	100	6800	\leq
Сталь, кг	10	70	640	\leq
Цветные металлы, кг	20	50	840	\leq
Прибыль, тыс. ден. ед.	6	16	} Z	
План выпуска, шт.	x_1	x_2		

2) Так как каждое изделие I_1 дает прибыль 6 тыс. ден. ед., а таких изделий изготавливается x_1 ед., то все изделия I_1 дадут прибыль $6x_1$; аналогично изделия I_2 обеспечат прибыль $16x_2$. Суммарную прибыль можно записать в виде целевой функции, которая максимизирует прибыль:

$$Z = 6x_1 + 16x_2 \rightarrow \max .$$

Токарного оборудования на выпуск одного изделия I_1 требуется 300 станко-часов, на изделие I_2 – 400 станко-часов. Тогда для изготовления x_1 изделий I_1 и x_2 изделий I_2 потребуется токарного оборудования $300x_1 + 400x_2$ (станко-часов). Так как общий фонд рабочего времени токарных станков не может превышать 12400 станко-ч, должно выполняться неравенство

$$300x_1 + 400x_2 \leq 12400 .$$

Аналогично можно записать условия, налагаемые на фонд рабочего времени фрезерных станков:

$$200x_1 + 100x_2 \leq 6800 ,$$

и лимитирующие материалы: по стали

$$10x_1 + 70x_2 \leq 640 ,$$

по цветным металлам

$$20x_1 + 50x_2 \leq 840 .$$

Итак, искомый план задачи $X = (x_1; x_2)$. Тогда математическая модель задачи определяется следующим образом.

Целевая функция

$$Z = 6x_1 + 16x_2 \rightarrow \max.$$

Система ограничений

$$\begin{cases} 300x_1 + 400x_2 \leq 12400, \\ 200x_1 + 100x_2 \leq 6800, \\ 10x_1 + 70x_2 \leq 640, \\ 20x_1 + 50x_2 \leq 840. \end{cases}$$

Переменные x_1 и x_2 не могут быть выражены отрицательными числами, поэтому

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

□

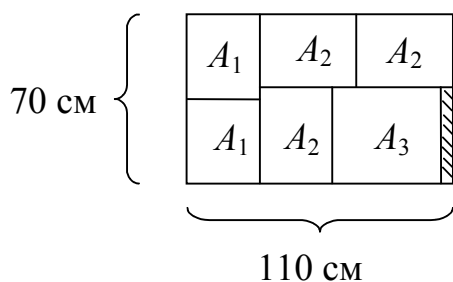
Пример 2.2. На предприятии имеется листовой материал, представляющий прямоугольники размера 70×110 см². Требуется выкроить прямоугольные заготовки типов A_1, A_2, A_3 , число которых соответственно равно 220, 120 и 80. Размеры заготовок следующие: $A_1 : 35 \times 30$; $A_2 : 30 \times 40$; $A_3 : 45 \times 40$.

Требуется:

- 1) составить возможные варианты раскроя;
- 2) построить математическую модель раскроя материала, минимизирующую отходы и удовлетворяющую потребность в заготовках типов $A_i, i = \overline{1, 3}$.

Решение.

- 1) Составим хотя бы четыре возможных варианта раскроя листового материала. Заштрихованные места – это отходы, полученные при разрезе листов.



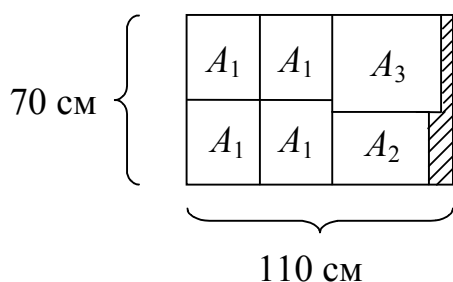
Способ 1 (M_1)

$$A_1 - 2$$

$$A_2 - 3$$

$$A_3 - 1$$

Отходы: 200 см²



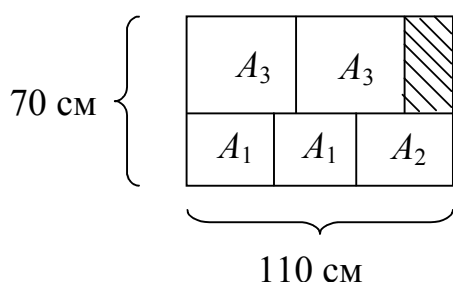
Способ 2 (M_2)

$$A_1 - 4$$

$$A_2 - 1$$

$$A_3 - 1$$

Отходы: 500 см^2



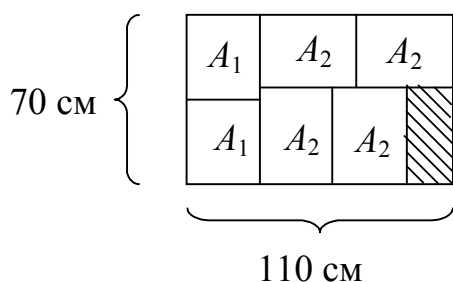
Способ 3 (M_3)

$$A_1 - 2$$

$$A_2 - 1$$

$$A_3 - 2$$

Отходы: 800 см^2



Способ 4 (M_4)

$$A_1 - 2$$

$$A_2 - 4$$

$$A_3 - 0$$

Отходы: 800 см^2

2) Для построения математической модели раскроя листового материала полученные варианты заносим в таблицу.

Вид заготовки	Размеры заготовки, см^2	Способы раскроя				Потребность в заготовках, шт.
		M_1	M_2	M_3	M_4	
A_1	35×30	2	4	2	2	220
A_2	30×40	3	1	1	4	120
A_3	45×40	1	1	2	0	80
Отходы, см^2		200	500	800	800	

Пусть $X = (x_1; x_2; x_3; x_4)$ – план задачи, где x_j ($j = \overline{1, 4}$) – число листов, раскраиваемых соответствующими способами M_1 , M_2 , M_3 и M_4 . Получаем следующую математическую модель задачи:

$$Z = 200 \cdot x_1 + 500 \cdot x_2 + 800 \cdot x_3 + 800 \cdot x_4 \rightarrow \min ,$$

$$\begin{cases} 2x_1 + 4x_2 + 2x_3 + 2x_4 = 220, \\ 3x_1 + x_2 + x_3 + 4x_4 = 120, \\ x_1 + x_2 + 3x_3 = 80. \end{cases}$$

$$x_i \geq 0, \quad i = \overline{1, 4}$$

□

2.2. Формы записи задач линейного программирования

Модель задачи линейного программирования может быть записана в одной из приведенных ниже форм.

1. *Общая, или произвольная, форма записи (ОЗЛП):*

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

при ограничениях:

$$\begin{cases} \sum_{j=1}^n a_{ij} x_j \leq b_i & (i = \overline{1, m_1}), \\ \sum_{j=1}^n a_{ij} x_j = b_i & (i = \overline{m_1 + 1, m_2}), \\ \sum_{j=1}^n a_{ij} x_j \geq b_i & (i = \overline{m_2 + 1, m}), \end{cases}$$

$$x_j \geq 0 \quad (j = \overline{1, n_1})$$

x_j – произвольная ($j = \overline{n_1 + 1, n}$).

2. *Симметричная, или стандартная, форма записи (СФЗЛП):*

$$\begin{aligned} \max Z &= \sum_{j=1}^n c_j x_j, \\ \sum_{j=1}^n a_{ij} x_j &\leq b_i \quad (i = \overline{1, m}), \\ x_j &\geq 0 \quad (j = \overline{1, n}) \end{aligned}$$

$$\begin{aligned} \min Z &= \sum_{j=1}^n c_j x_j, \\ \sum_{j=1}^n a_{ij} x_j &\geq b_i \quad (i = \overline{1, m}), \\ x_j &\geq 0 \quad (j = \overline{1, n}) \end{aligned}$$

3. Каноническая, или основная, форма записи (КФЗЛП)

$$\begin{aligned} \max Z &= \sum_{j=1}^n c_j x_j, \\ \sum_{j=1}^n a_{ij} x_j &= b_i \quad (i = \overline{1, m}), \\ x_j &\geq 0 \quad (j = \overline{1, n}). \end{aligned}$$

Указанные выше три формы записи ЗЛП эквивалентны в том смысле, что каждая из них с помощью несложных преобразований может быть сведена к другой форме, т.е. если имеется способ нахождения оптимального решения задачи в одной из указанных форм, то тем самым может быть определен оптимальный план задачи в любой другой форме (говорят о стратегической эквивалентности задачи в любой из форм).

Так, при необходимости задачу минимизации можно заменить задачей максимизации, и наоборот. Для функции одной переменной это утверждение очевидно. В самом деле, если x^* – точка минимума функции $y = f(x)$, то для функции $y = -f(x)$ она является точкой максимума, так как графики функций $f(x)$ и $-f(x)$ симметричны относительно оси абсцисс (см. рисунок 2.1).

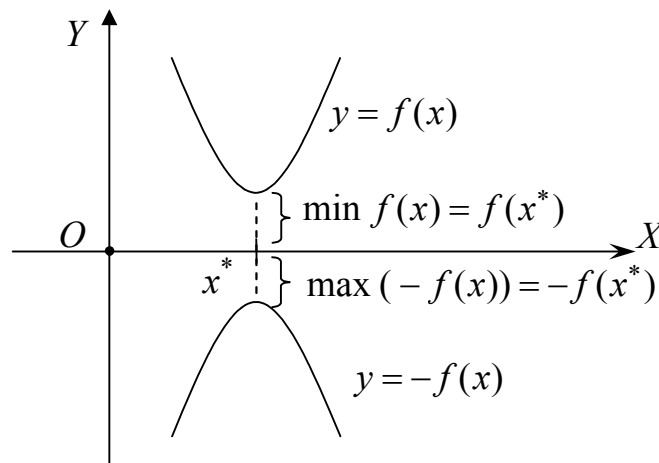


Рис. 2.1

Итак,

$$\min f(x^*) = -\max(-f(x^*)).$$

Данное равенство верно и в случае функции n переменных:

$$\min Z(x_1^*, x_2^*, \dots, x_n^*) = -\max(-Z(x_1^*, x_2^*, \dots, x_n^*)).$$

Неравенство типа « \geq » путем умножения левых и правых частей на -1 можно превратить в неравенство типа « \leq », и наоборот. Ограничения-неравенства $a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \begin{cases} \leq \\ \geq \end{cases} b_i$ преобразуются в ограничения-равенства путем прибавления (вычитания) к левым частям дополнительных (балансовых) неотрицательных переменных $x_{n+i} \geq 0$:

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \begin{cases} + \\ - \end{cases} x_{n+i} = b_i.$$

В случае необходимости ограничение-равенство

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n = b_i$$

можно записать в виде системы неравенств

$$\begin{cases} a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i, \\ a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \geq b_i. \end{cases}$$

Если в ЗЛП какая-то переменная x_k не подчиняется условию неотрицательности, ее заменяют разностью двух других неотрицательных переменных $x'_k \geq 0$ и $x''_k \geq 0$:

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

Вводимые дополнительные переменные имеют определенный экономический смысл, прямо связанный с содержанием задачи. Так, в задачах об использовании ресурсов они показывают величину неиспользованного ресурса, в задачах о смесях – потребление соответствующего компонента сверх нормы.

Рассмотрим на примерах, как можно выполнять переход от одной формы записи ЗЛП к другой.

Пример 2.3. Привести к канонической форме записи ЗЛП:

$$Z = 6x_1 + 5x_2 \rightarrow \min,$$

$$\begin{cases} 5x_1 + 11x_2 \geq 55, \\ 11x_1 - 3x_2 \geq 32, \\ 16x_1 + 13x_2 \leq 21, \\ -17x_1 + 12x_2 = 25, \end{cases}$$

$$x_1 \geq 0, \quad x_2 - \text{любого знака}.$$

Решение. Заменяем функцию Z на $Z' = -Z$. Из левых частей ограничений типа « \geq » вычтем неотрицательные переменные x_3, x_4 , а к левой части ограничения типа « \leq » прибавим неотрицательную переменную x_5 . Переменную x_2 , которая может быть произвольного знака, заменим разностью двух неотрицательных переменных:

$$x_2 = x'_2 - x''_2.$$

В результате получаем модель задачи в каноническом виде:

$$Z' = -6x_1 - 5(x'_2 - x''_2) \rightarrow \max,$$

$$\begin{cases} 5x_1 + 11x'_2 - 11x''_2 - x_3 & = 55, \\ 11x_1 - 3x'_2 + 3x''_2 - x_4 & = 32, \\ 16x_1 + 13x'_2 - 13x''_2 + x_5 & = 21, \\ -17x_1 + 12x'_2 - 12x''_2 & = 25, \end{cases}$$

$$x_1 \geq 0, \quad x'_2 \geq 0, \quad x''_2 \geq 0.$$

□

Пример 2.4. Привести к симметрической форме записи задачу, заданную в виде:

$$Z = -2x_1 - x_2 - x_3 + 2x_4 + x_5 \rightarrow \max,$$

$$\begin{cases} -2x_3 - x_4 + x_5 = 4, \\ -x_2 + 4x_3 + 2x_4 = 8, \\ x_1 + x_3 + x_4 = 6, \end{cases}$$

$$x_j \geq 0 \quad (j = \overline{1, 5}).$$

Решение. Так как целевая функция, по условию, максимизируется, то все ограничения в канонической форме записи должны иметь вид « \leq ». Поскольку в систему ограничений входят три неравенства, то исключим из системы любые три переменные. В данном случае удобно исключить из первого ограничения x_5 , из второго — x_2 и из третьего — x_1 . Учитывая неотрицательность переменных, получаем:

$$\begin{cases} x_5 = 4 + 2x_3 + x_4 & \geq 0, \\ x_2 = -8 + 4x_3 + 2x_4 & \geq 0, \\ x_1 = 6 - x_3 - x_4 & \geq 0. \end{cases}$$

Подставив x_5, x_2, x_1 в целевую функцию, получаем

$$Z = -2(6 - x_3 - x_4) - (-8 + 4x_3 + 2x_4) - x_3 + 2x_4 + 4 + 2x_3 + x_4 = -x_3 + 3x_4.$$

Исключив x_5, x_2, x_1 , приходим к эквивалентным неравенствам. В результате получаем следующую ЗЛП в симметрической форме:

$$Z = -x_3 + 3x_4 \rightarrow \max,$$

$$\begin{cases} -2x_3 - x_4 \leq 4, \\ -4x_3 - 2x_4 \leq -8, \\ x_3 + x_4 \leq 6. \end{cases}$$

$$x_3 \geq 0, \quad x_4 \geq 0.$$

□

2.3. Геометрическая интерпретация ЗЛП

Геометрическая интерпретация оптимизационных задач дает возможность наглядно представить их структуру, выявить особенности и открывает пути исследования более сложных свойств. Задачу линейного программирования с двумя переменными всегда можно решить графически. Однако уже в трехмерном пространстве такое решение усложняется, а в пространстве, размерность которого больше трех, графическое решение, вообще говоря, невозможно.

Случай двух переменных не имеет особого практического значения, однако его рассмотрение проясняет свойства ОЗЛП, приводит к идее решения, делает геометрически наглядными способы решения и пути их практической реализации.

Пусть дана задача: найти $X = (x_1; x_2)$ – план задачи, если

$$Z = c_1x_1 + c_2x_2 \rightarrow \max, \tag{2.8}$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 \leq b_1, \\ a_{21}x_1 + a_{22}x_2 \leq b_2, \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 \leq b_m, \end{cases} \tag{2.9}$$

$$x_1 \geq 0, \quad x_2 \geq 0. \tag{2.10}$$

Дадим геометрическую интерпретацию элементов этой задачи. Каждое из ограничений (2.9) и (2.10) задает на плоскости Ox_1x_2 некоторую полуплоскость. Полуплоскость – выпуклое множество. Пересечение любого числа выпуклых множеств является выпуклым множеством. Отсюда следует, что область допустимых решений задачи (2.8) – (2.10) есть выпуклое множество.

На рисунке представлены возможные случаи области допустимых решений задачи линейного программирования.

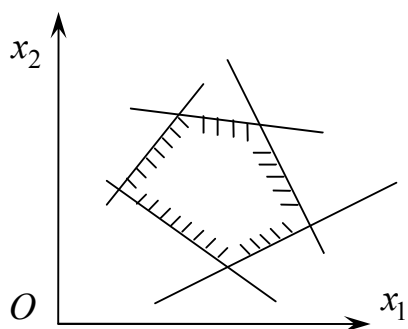


Рис. 2.2. Выпуклый многоугольник

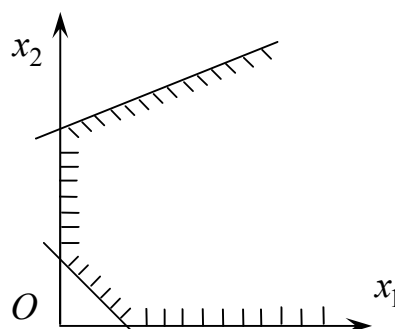


Рис. 2.3. Неограниченная выпуклая область

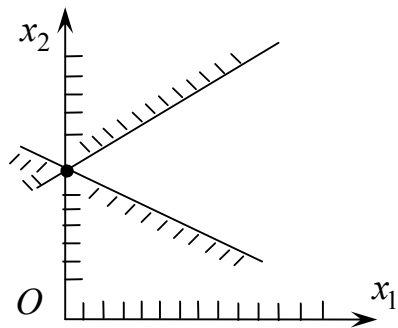


Рис. 2.4. Единственная Точка

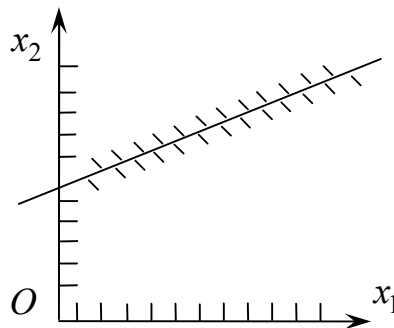


Рис. 2.5. Луч

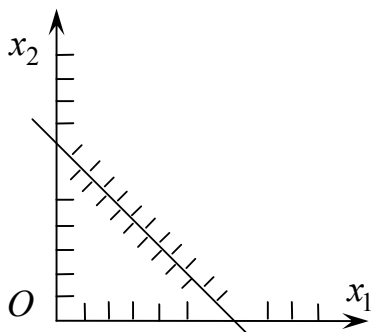


Рис. 2.6. Отрезок

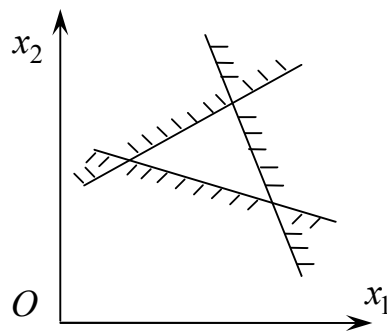


Рис. 2.7. Пустое множество

Перейдем к геометрической интерпретации целевой функции. Пусть область допустимых решений ЗЛП – непустое множество, например, многоугольник $A_1A_2A_3A_4A_5A_6$, рис. 2.8.

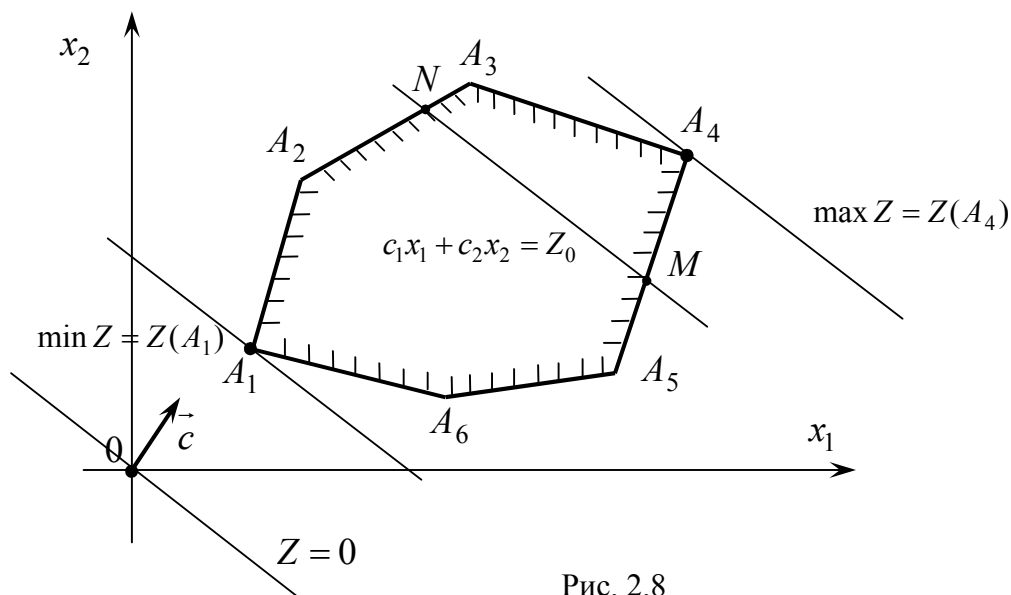


Рис. 2.8

Выберем произвольное значение целевой функции $Z = Z_0$. Получаем $c_1x_1 + c_2x_2 = Z_0$. Это уравнение прямой линии. В точках прямой MN целевая функция сохраняет одно и то же постоянное значение Z_0 . Считая в равенстве (2.8) Z параметром, получим уравнение семейства параллельных прямых, называемых *линиями уровня целевой функции* (*линиями постоянного значения*).

Возникает вопрос: как установить направление возрастания (убывания) целевой функции?

Найдем частные производные целевой функции по x_1 и x_2 :

$$\frac{\partial Z}{\partial x_1} = c_1, \quad (2.11)$$

$$\frac{\partial Z}{\partial x_2} = c_2. \quad (2.12)$$

Частная производная (2.11) или (2.12) функции показывает скорость ее возрастания вдоль данной оси. Следовательно, c_1 и c_2 – скорости возрастания Z соответственно вдоль осей Ox_1 и Ox_2 . Вектор $\vec{c} = \overline{\text{grad } Z} = (c_1; c_2)$ называется *градиентом функции*. Он показывает направление наискорейшего возрастания целевой функции: $\vec{c} = \left(\frac{\partial Z}{\partial x_1}; \frac{\partial Z}{\partial x_2} \right)$.

Вектор $(-\vec{c})$ указывает направление наискорейшего убывания целевой функции. Его называют *антиградиентом*.

Вектор $\vec{c} = (c_1; c_2)$ перпендикулярен к прямой $Z = \text{const}$ семейства $c_1x_1 + c_2x_2 = Z$.

Из геометрической интерпретации элементов задачи линейного программирования следует порядок ее графического решения.

Графический метод решения ЗЛП

1. С учетом системы ограничений строим область допустимых решений Ω .

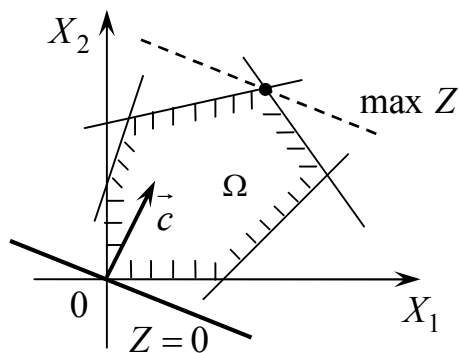
2. Строим вектор $\vec{c} = (c_1; c_2)$ наискорейшего возрастания целевой функции – вектор градиентного направления.

3. Проводим произвольную линию уровня $Z = Z_0$ (проще всего провести $Z = 0$, перпендикулярную к вектору \vec{c}).

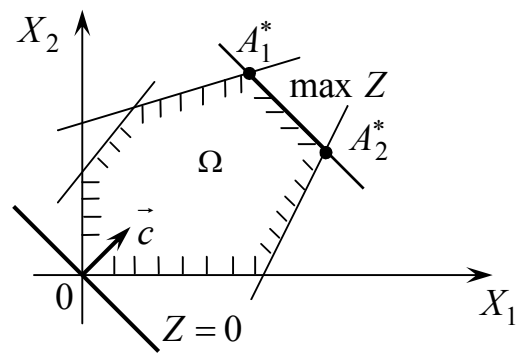
4. При решении задачи на максимум перемещаем линию уровня $Z = Z_0$ в направлении вектора \vec{c} так, чтобы она касалась области допустимых решений в ее крайнем положении (крайней точке). На рисунке выше – это точка A_4 . В случае решения задачи на минимум линию уровня $Z = Z_0$ перемещаем в антиградиентном направлении. На рисунке выше – это точка A_1 .

5. Определяем оптимальный план $X^* = (x_1^*; x_2^*)$ и экстремальное значение целевой функции $Z^* = Z(X^*)$.

В результате решения задачи линейного программирования возможны следующие случаи оптимального плана, которые представлены на рисунке 2.9.



(1)



(2)

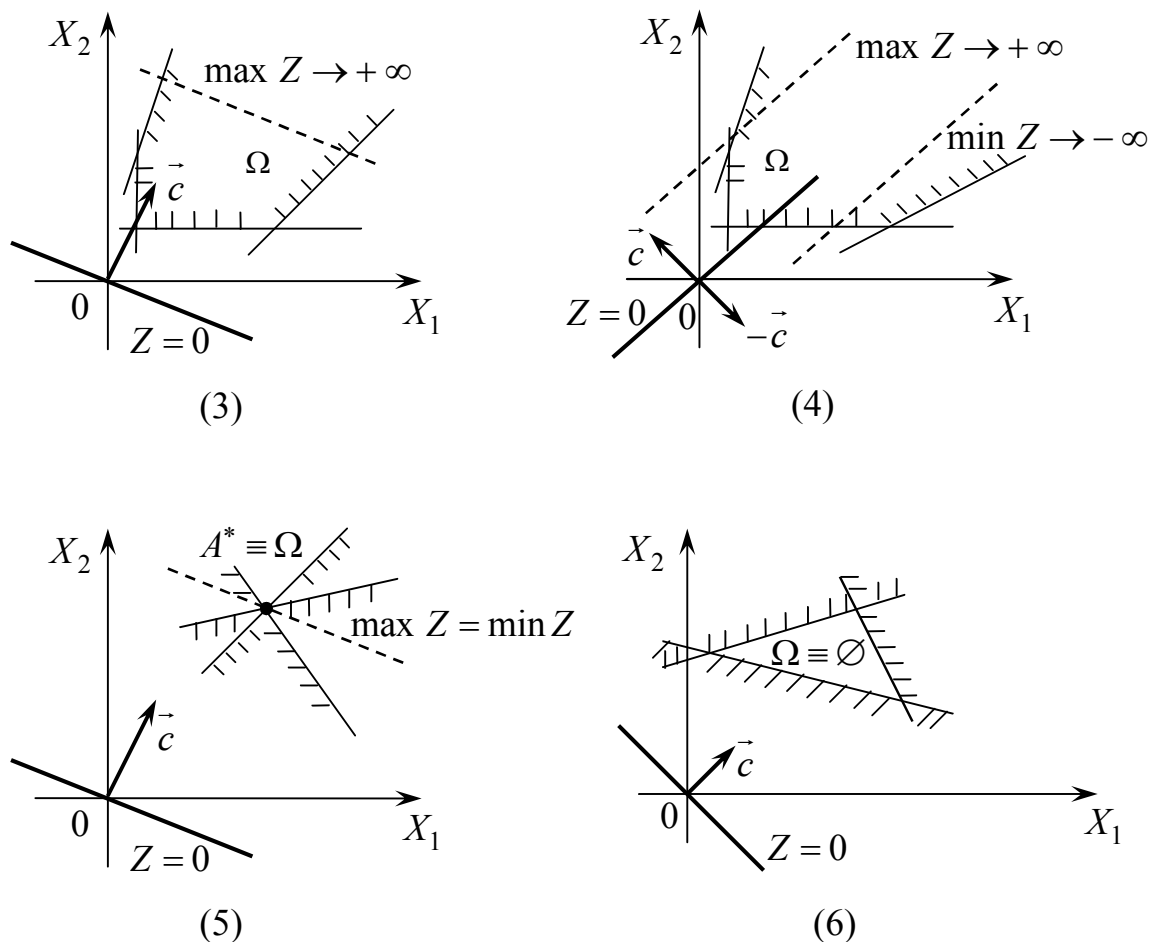


Рис. 2.9

(1) – оптимальный план единственный: линия уровня и область допустимых решений Ω в разрешающем положении имеют одну общую точку;

(2) – оптимальных планов бесчисленное множество: в разрешающем положении линия уровня проходит через сторону области допустимых решений;

(3), (4) – целевая функция не ограничена: линия уровня, сколько бы ее не перемещали, не может занять разрешающего положения; только в случае (3) возможно решение задачи на минимизацию целевой функции;

(5) – область допустимых решений состоит из единственной точки, где целевая функция достигает одновременно и максимального, и минимального значений;

(6) – задача не имеет решения: область допустимых решений – пустое множество, т.е. система ограничений задачи несовместна.

Пример 2.5. Решить ЗЛП

$$Z = 2x_1 - 3x_2 \rightarrow \min (\max),$$

$$\begin{cases} x_1 + x_2 \leq 6, \\ x_1 + 4x_2 \geq 4, \\ 2x_1 - x_2 \geq 0, \end{cases}$$

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

Решение. Для построения области допустимых решений строим в системе Ox_1x_2 соответствующие данным ограничениям-неравенствам граничные прямые:

$$(1): x_1 + x_2 = 6; \quad (2): x_1 + 4x_2 = 4; \quad (3): 2x_1 - x_2 = 0.$$

Находим полуплоскости, в которых выполняются данные неравенства. Для этого вследствие выпуклости любой полуплоскости достаточно взять произвольную точку, через которую не проходит соответствующая граничная прямая, и проверить, удовлетворяет ли эта пробная точка ограничению-неравенству. Если удовлетворяет, то данное неравенство выполняется в полуплоскости, содержащей пробную точку. В противном случае берется полуплоскость, не содержащая пробной точки. В качестве пробной точки часто удобно брать начало координат $O(0; 0)$. Для нашего примера область допустимых решений – множество точек четырехугольника $ABCD$, рис 2.10.

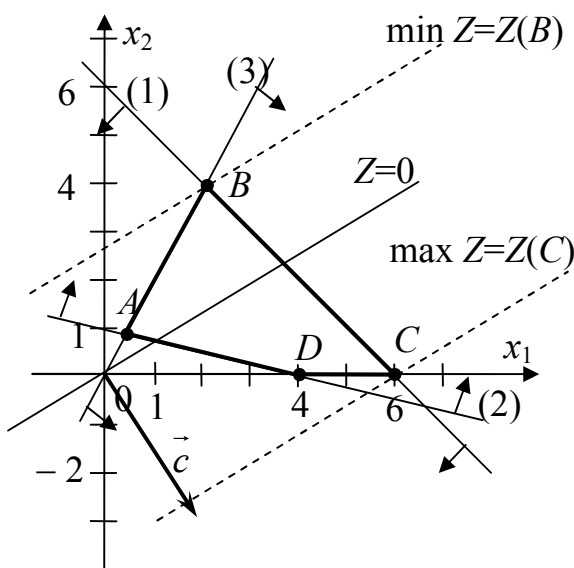


Рис. 2.10

Строим вектор $\vec{c} = (c_1; c_2) = (2; -3)$. Перпендикулярно вектору \vec{c} проводим линию уровня $Z=0$. Параллельным перемещением прямой $Z=0$ находим крайнюю точку B , в которой целевая функция достигает минимум, и точку C , в которой целевая функция достигает максимум. Координаты точки B определяются системой

$$\begin{cases} x_1 + x_2 = 6, & (1) \\ 2x_1 - x_2 = 0, & (3) \end{cases}$$

откуда $B(2; 4)$.

Координаты точки C можно найти из решения системы

$$\begin{cases} x_2 = 0, & (\text{ось } Ox_1) \\ x_1 + x_2 = 6, & (1) \end{cases}$$

откуда $C(6; 0)$.

Таким образом, $Z_{\min} = Z(2; 4) = -8$, $Z_{\max} = Z(6; 0) = 12$.

Ответ: $Z_{\min} = Z(2; 4) = -8$, $Z_{\max} = Z(6; 0) = 12$. □

Пример 2.6. При производстве продукции Π_1 и Π_2 используют четыре группы оборудования: A , B , C и D . На выпуск единицы продукции Π_1 расходуется 1; 0,5; 2 и 0 ед. времени оборудования A , B , C и D соответственно, а на выпуск продукции Π_2 – 1; 1; 0 и 2 ед. времени оборудования. Фонд рабочего времени оборудования группы A составляет 18 ед. времени; B – 12 ед.; C – 24 ед. и D – 18 ед. Предприятие реализует единицу продукции Π_1 по цене 40 ден. ед., Π_2 – 60 ден. ед. Найти план выпуска продукции, при котором выручка предприятия будет максимальной.

Решение.

1) Запишем условие задачи в виде таблицы.

Вид оборудования	Продукция		Фонд рабочего времени, ед.
	Π_1	Π_2	
A	1	1	18
B	0,5	1	12
C	2	0	24
D	0	2	18
Цена единицы продукции, ден. ед.	40	60	

Пусть x_1 – объем выпуска продукции вида Π_1 , x_2 – объем выпуска продукции Π_2 , т.е. $X = (x_1; x_2)$ – план выпуска продукции предприятием. Предприятие стремится максимизировать прибыль, которую получает от реализации продукции, тогда целевая функция имеет вид

$$Z = 40x_1 + 60x_2 \rightarrow \max.$$

На производство двух видов продукции будет затрачено $x_1 + x_2$ единиц времени оборудования группы A ; $0,5x_1 + x_2$ – оборудования группы B ; $2x_1$ – оборудования C ; $2x_2$ – оборудования D . При этом, учитывая

фонд рабочего времени каждого вида оборудования, получаем систему ограничений:

$$\begin{cases} x_1 + x_2 \leq 18, \\ 0,5x_1 + x_2 \leq 12, \\ 2x_1 \leq 24, \\ 2x_2 \leq 18. \end{cases}$$

По содержанию задачи переменные $x_1 \geq 0$, $x_2 \geq 0$.

Таким образом, получаем следующую математическую модель, которая является задачей линейного программирования.

$$Z = 40x_1 + 60x_2 \rightarrow \max,$$

$$\begin{cases} x_1 + x_2 \leq 18, \\ 0,5x_1 + x_2 \leq 12, \\ 2x_1 \leq 24, \\ 2x_2 \leq 18, \end{cases}$$

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

2) Так как план выпуска продукции содержит две переменные, то ЗЛП можно решить графическим способом.

Для построения области допустимых решений строим в системе координат Ox_1x_2 соответствующие данным ограничениям-неравенствам граничные прямые, рис. 2.11:

$$(1): x_1 + x_2 = 18; \quad (2): 0,5x_1 + x_2 = 12; \quad (3): 2x_1 = 24;$$

$$(4): 2x_2 = 18; \quad (5): x_1 = 0; \quad (6): x_2 = 0.$$

Область допустимых решений – множество точек многоугольника $OABCD$.

Строим вектор $\vec{c}_1 = \frac{1}{10}\vec{c}$ и линию уровня $Z = 0$. Параллельным перемещением прямой $Z = 0$ в направлении вектора \vec{c}_1 находим крайнюю точку C , в которой целевая функция достигает максимум. Координаты точки C определяются из решения системы

$$\begin{cases} x_1 + x_2 = 18, & (1) \\ 0,5x_1 + x_2 = 12, & (2) \end{cases}, \text{ откуда } C(12; 6).$$

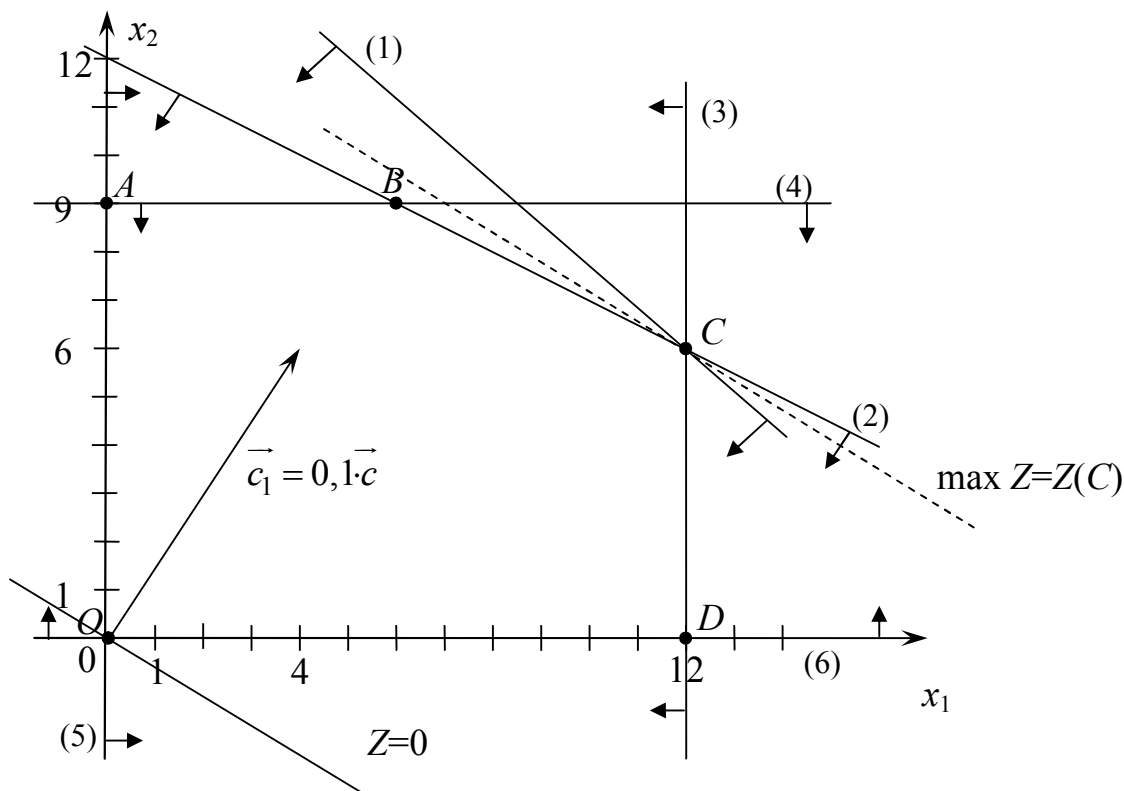


Рис. 2.11

Таким образом, $X = (12; 6)$ – план выпуска продукции предприятием. Тогда максимальная прибыль

$$Z_{\max} = Z(12; 6) = 40 \cdot 12 + 60 \cdot 6 = 840 \text{ (ден. ед.)}.$$

Ответ: следует выпускать 12 ед. продукции вида Π_1 и 6 ед. продукции вида Π_2 , при этом прибыль составит 840 ден. ед.

□

Графическим методом можно решить ЗЛП с $n > 2$ переменными, если в ее канонической записи число неизвестных n и число линейно независимых уравнений m связаны соотношением $n - m \leq 2$. В этом случае каноническую форму задачи преобразовывают в симметричную, которая будет содержать не более двух переменных.

Решая эту задачу графически, находят два компонента оптимального плана. Подставляя их в ограничения задачи, определяют и остальные компоненты.

Анализируя решение приведенных выше примеров, заметим, что для ЗЛП с двумя переменными экстремум достигается в вершине области допустимых решений.

В связи с этим следует отметить, что, независимо от числа переменных, экстремум целевой функции достигается в вершине многогранника планов решений ЗЛП или, если эта вершина не является единственной, в точках, являющихся выпуклой линейной комбинацией вершин, где достигается экстремум. В результате возникает способ решения ЗЛП любой размерности, который заключается в том, чтобы найти каким-нибудь образом все крайние точки многогранника планов, их не больше, чем $C_n^r = \frac{n!}{r!(n-r)!}$, и сравнить в них значения целевой функции. Однако такой путь решения даже с относительно небольшим числом переменных и ограничений практически неосуществим, так как процесс отыскания крайних точек сравним по трудности с решением исходной задачи, к тому же число крайних точек многогранника планов может оказаться весьма большим.

В связи с этим возник метод рационального перебора крайних точек. Суть его в следующем. Если известны какая-нибудь крайняя точка и значение в ней целевой функции, то все крайние точки, в которых целевая функция принимает худшее значение, заведомо не нужны. Тогда естественно стремление найти способ перехода от данной крайней точки к лучшей, которая смежная с данной по ребру, от нее к еще лучшей (не худшей) и т.д. Для этого нужно иметь признак того, что лучших крайних точек, чем данная крайняя точка вообще нет. В этом и состоит идея наиболее широко применяемого в настоящее время *симплексного метода* (*метода последовательного улучшения плана*) для решения ЗЛП.

На базе этого метода составлен ряд компьютерных программ по оптимизации ЗЛП.

3. ТРАНСПОРТНАЯ ЗАДАЧА

3.1. Постановка транспортной задачи по критерию стоимости в матричной форме

Рассмотрим простейший вариант модели транспортной задачи (ТЗ). Пусть речь идет о рациональной перевозке некоторого однородного продукта от производителя к потребителям, при этом имеется баланс между суммарным спросом потребителей и возможностями поставщи-

ков по их удовлетворению. Потребителям безразлично, из каких пунктов будет поступать продукция, лишь бы их заявки были полностью удовлетворены. Так как от схемы закрепления потребителей за поставщиками существенно зависит объем транспортной работы, возникает задача о наиболее рациональном закреплении, правильном направлении перевозок грузов, при котором потребности полностью удовлетворяются, вся продукция от поставщиков вывозится, а затраты на транспортировку минимальны.

Транспортную задачу можно сформулировать следующим образом:

В m пунктах отправления A_1, \dots, A_m сосредоточен однородный груз в количествах соответственно a_1, \dots, a_m единиц. Имеющийся груз необходимо доставить потребителям B_1, \dots, B_n , спрос которых выражается величинами b_1, \dots, b_n единиц. Известна стоимость c_{ij} перевозки единицы груза из i -того ($i = \overline{1, m}$) пункта отправления в j -тый ($j = \overline{1, n}$) пункт назначения. Удельные транспортные издержки (расходы) записывают в форме матрицы $C = [c_{ij}]_{m \times n}$, которую называют *матрицей тарифов*. Требуется спланировать перевозки, т.е. указать, сколько единиц груза должно быть отправлено от i -того поставщика j -тому потребителю, так, чтобы максимально удовлетворить спрос потребителей и чтобы суммарные транспортные затраты на перевозки были при этом минимальными. Рассмотрим простейший случай, когда суммарные запасы поставщиков равны суммарным потребностям $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$, т.е. наблюдается баланс между спросом и потреблением.

Для составления математической модели задачи введем переменные x_{ij} ($i = \overline{1, m}; j = \overline{1, n}$), обозначающие количество единиц груза, которые необходимо доставить из i -того пункта отправления в j -тый пункт назначения. Все эти переменные можно записать в виде матрицы $X = [x_{ij}]_{m \times n}$, которая будет называться *матрицей перевозок*:

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix}.$$

В связи с этим ТЗ можно представить в виде таблицы, которую называют *распределительной*, табл. 3.1. Распределительную таблицу называют иногда *табличной* или *матричной моделью ТЗ*.

Таблица 3.1

Поставщики	Потребители				Запас груза, a_i
	B_1	B_2	...	B_n	
A_1	c_{11} x_{11}	c_{12} x_{12}	...	c_{1n} x_{1n}	a_1
A_2	c_{12} x_{12}	c_{22} x_{22}	...	c_{2n} x_{2n}	a_2
...
A_m	c_{m1} x_{m1}	c_{m2} x_{m2}	...	c_{mn} x_{mn}	a_m
Потребность в грузе, b_j	b_1	b_2	...	b_n	

Цель транспортной задачи – минимизировать общие затраты на реализацию плана перевозок, которые можно представить целевой функцией:

$$Z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} = c_{11}x_{11} + c_{12}x_{12} + \dots + c_{ij}x_{ij} + \dots + c_{mn}x_{mn} \rightarrow \min . \quad (3.1)$$

Переменные должны удовлетворять следующим условиям:

1) ограничения по запасам:

$$x_{i1} + x_{i2} + \dots + x_{in} = \sum_{j=1}^n x_{ij} = a_i \quad (i = \overline{1, m}); \quad (3.2)$$

2) ограничения по потребностям:

$$x_{1j} + x_{2j} + \dots + x_{mj} = \sum_{i=1}^m x_{ij} = b_j \quad (j = \overline{1, n}); \quad (3.3)$$

3) условия неотрицательности:

$$x_{ij} \geq 0 \quad (i = \overline{1, m}; j = \overline{1, n}). \quad (3.4)$$

где c_{ij} – стоимость перевозки единицы груза из i -того ($i = \overline{1, m}$) пункта отправления в j -тый ($j = \overline{1, n}$) пункт назначения;

a_i – количество груза, сосредоточенного в пункте A_i ;

b_j – количество груза, необходимое для доставки потребителю B_j .

Если план перевозок $X = [x_{ij}]_{m \times n}$ удовлетворяет ограничениям (3.2) – (3.4), то такой план называется *допустимым*. Допустимый план перевозок, составляющий минимум целевой функции, называется *оптимальным*.

Критерий существования допустимого плана можно представить в виде теоремы.

Теорема 3.1 (о существовании допустимого плана)*.

Для того чтобы транспортная задача имела допустимые планы, необходимо и достаточно выполнение равенства

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j. \quad (3.5)$$

Модель ТЗ называется *закрытой*, если суммарный объем груза, имеющегося у поставщиков, равен суммарному спросу потребителей, т.е. выполняется равенство

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

Если для ТЗ выполняется одно из условий:

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

то модель задачи называется *открытой*.

Для разрешимости ТЗ с открытой моделью необходимо преобразовать ее в закрытую.

Так, при выполнении условия $\sum_{i=1}^m a_i > \sum_{j=1}^n b_j$ необходимо ввести фиктивный $(n+1)$ -й пункт назначения B_{n+1} , т.е. в матрице задачи предусматривается дополнительный столбец. Спрос фиктивного потребителя полагают равным небалансу, т.е. $b_{n+1} = \sum_{i=1}^m a_i - \sum_{j=1}^n b_j$, а все тарифы – одинаковыми, чаще всего равными нулю $c_{i, n+1} = 0$ ($i = \overline{1, m}$).

Аналогично, при выполнении условия $\sum_{i=1}^m a_i < \sum_{j=1}^n b_j$ вводится фиктивный поставщик A_{m+1} , у которого запас груза равен $a_{m+1} = \sum_{j=1}^n b_j - \sum_{i=1}^m a_i$,

а тарифы дополнительной строки распределительной таблицы равны нулю, т.е. $c_{m+1, j} = 0$ ($j = \overline{1, n}$).

При преобразовании открытой задачи в закрытую целевая функция не изменяется, так как все слагаемые, соответствующие дополнительным перевозкам, равны нулю.

3.2. Построение исходного опорного плана

Построение опорных планов, а также их преобразование проводят непосредственно в распределительной таблице. Если в плане перевозок переменная x_{ik} равна некоторому числу $a \neq 0$, то это число записывают в соответствующую клетку $(i; k)$ и считают ее занятой (или базисной), если же $x_{ik} = 0$, то клетку $(i; k)$ оставляют свободной. При этом число занятых клеток должно быть равно $m + n - 1$, а остальные $mn - (m + n - 1) = (m - 1)(n - 1)$ клеток будут свободными.

Для построения начального опорного плана в распределительной таблице можно использовать *правило «северно-западного угла»*, *правило «минимального элемента»*, *метод Фогеля*.

Рассмотрим *правило «минимального элемента»*. Суть его состоит в следующем. Просматривая тарифы распределительной таблицы, в первую очередь заполняют клетку с минимальным значением тарифа. При этом в клетку записывается максимально возможное значение поставки. Затем из рассмотрения исключают строку, соответствующую поставщику, запасы которого полностью израсходованы, или столбец, соответствующий потребителю, спрос которого полностью удовлетворен. После этого из оставшихся клеток таблицы снова выбирают клетку с наименьшим тарифом. Процесс распределения заканчивается, когда все запасы поставщиков исчерпаны, а спрос потребителей полностью удовлетворен. В результате получаем опорный план, который должен содержать $m + n - 1$ загруженных клеток. В процессе заполнения таблицы могут быть одновременно исключены строка и столбец. Так бывает, когда полностью исчерпывается запас груза и полностью удовлетворяется спрос (вырожденная задача). В этом случае в свободные клетки надо записать число 0 – «*нуль-загрузку*», условно считая такую клетку занятой. Число 0 может быть записано в те свободные клетки, которые не образуют циклов с ранее занятыми клетками.

3.3. Метод потенциалов

После построения начального опорного плана ТЗ возникает вопрос: как определить, будет ли построенный план оптимальным? Чтобы ответить на этот вопрос, необходимо знать признак оптимальности. Для этого разработан *метод потенциалов*, который опирается на теорему о потенциалах.

Каждому поставщику (ограничению по запасам) поставим в соответствие число u_i ($i = \overline{1, m}$), а каждому потребителю (ограничению по спросу) – число v_j ($j = \overline{1, n}$). Числа u_i и v_j называются *потенциалами* соответственно i -того поставщика и j -того потребителя.

Теорема 3.2 (о потенциалах)*

Если план $X^ = [x_{ij}^*]_{m \times n}$ транспортной задачи является оптимальным, то ему соответствует система из $m+n$ чисел u_i^*, v_j^* , удовлетворяющих условиям $u_i^* + v_j^* = c_{ij}$ для $x_{ij}^* > 0$ и $u_i^* + v_j^* \leq c_{ij}$ для $x_{ij}^* = 0$.*

Из теоремы следует, что для оптимального плана ТЗ необходимо выполнение условий:

- 1) каждой занятой клетке в распределительной таблице соответствует сумма потенциалов, равная тарифу этой клетки, т.е. $u_i + v_j = c_{ij}$;
- 2) каждой свободной клетке соответствует сумма потенциалов, не превышающая тарифа этой клетки, т.е. $u_i + v_j \leq c_{ij}$.

Согласно теореме о потенциалах, каждой занятой клетке будет соответствовать уравнение $u_i + v_j = c_{ij}$. Так как всех занятых клеток должно быть $m+n-1$, т.е. на единицу меньше, то для определения чисел u_i и v_j необходимо решить систему из $m+n-1$ уравнений $u_i + v_j = c_{ij}$ с $m+n$ неизвестными. Система неопределенная, и чтобы найти частные решения, одному из потенциалов придается произвольное числовое значение, тогда остальные потенциалы определяются однозначно. Для облегчения расчетов одному из потенциалов придается обычно значение, равное нулю.

Для исследования плана на оптимальность по каждой свободной клетке проверяется условие $u_i + v_j \leq c_{ij}$. Если хотя бы одна свободная клетка не удовлетворяет данному условию, то опорный план не является оптимальным, его можно улучшить за счет загрузки этой клетки. Если таких клеток несколько, то наиболее перспективной для загрузки является клетка, для которой разность (оценка) между тарифом клетки и суммой потенциалов наименьшая, т.е.

$$s_{ij} = c_{ij} - (u_i + v_j) < 0.$$

Если для всех свободных клеток оценки $s_{ij} \geq 0$, то опорный план перевозок является оптимальным. Если для опорного плана перевозок указанное условие оптимальности не выполняется, то за счет загрузки свободной клетки с отрицательной оценкой план перевозок улучшается. Для наиболее перспективной свободной клетки строится замкнутый цикл с вершинами в загруженных клетках. Вершинам этого цикла условно приписываются знаки: свободной клетке – знак «+», следующей по часовой или против часовой стрелки занятой клетке – знак «-», следующей – снова «+» и т.д. Из поставок в клетках цикла с «отрицательными» вершинами выбирается наименьшее количество λ груза, которое и перемещается по клеткам этого цикла: прибавляется к поставкам в положительных вершинах и вычитается из поставок в отрицательных, в результате чего баланс цикла не нарушается.

В общем случае цикл представляет собой замкнутую ломаную линию, состоящую из звеньев, пересекающихся под прямым углом. Каждое звено соединяет две клетки строки (столбца). Цикл включает одну свободную клетку, остальные клетки цикла заняты. В цикле всегда четное число клеток. Для свободной клетки всегда можно построить единственный цикл. Если из занятых клеток образуется цикл, то план перевозок не является опорным. Цикл строится лишь для свободных клеток.

Алгоритм решения ТЗ методом потенциалов

1. Построить опорный план.
2. Вычислить потенциалы поставщиков и потребителей u_i ($i = \overline{1, m}$) и v_j ($j = \overline{1, n}$), решив систему уравнений вида $u_i + v_j = c_{ij}$.
3. Вычислить оценки s_{ij} для всех свободных клеток по формуле $s_{ij} = c_{ij} - (u_i + v_j)$. Если $s_{ij} \geq 0$, то полученный план является оптимальным.

При этом если все $s_{ij} > 0$, то полученный оптимальный план единственный. В случае если в оптимальном плане хотя бы одна оценка $s_{ij} = 0$, имеем бесчисленное множество оптимальных планов с одним и тем же значением целевой функции. Если же найдется оценка $s_{ij} < 0$ для некоторой клетки (i, j) , то следует выполнить перераспределение плана перевозок, загрузив данную свободную клетку.

Пример 3.1. В пунктах A_i ($i = \overline{1, 3}$) производится однородная продукция в количествах $a_1 = 360, a_2 = 150, a_3 = 380$ единиц. Готовая продукция поставляется в пункты B_j ($j = \overline{1, 4}$), потребности которых составляют $b_1 = 270, b_2 = 190, b_3 = 340, b_4 = 200$ единиц. Стоимость c_{ij} перевозки единицы продукции из пункта A_i в пункт B_j заданы матрицей $C = [c_{ij}]_{3 \times 4}$:

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

Требуется:

- 1) методом потенциалов найти план перевозок продукции, при котором минимизируются суммарные затраты по ее доставке потребителям;
- 2) вычислить суммарные затраты f_{\min} .

Решение.

- 1) Определим суммарные запасы продукции и суммарные потребности:

$$a_1 + a_2 + a_3 = 360 + 150 + 380 = 890,$$

$$b_1 + b_2 + b_3 + b_4 = 270 + 190 + 340 + 200 = 1000.$$

Поскольку $\sum_{i=1}^3 a_i \neq \sum_{j=1}^4 b_j$, то есть отсутствует баланс между произво-

димой продукцией и потребностями, то данная задача относится к задачам открытого типа. Вводим фиктивного поставщика A_4 , который имеет запас

продукции в объеме $a_4 = \sum_{j=1}^4 b_j - \sum_{i=1}^3 a_i = 1000 - 890 = 110$ (ед.). Все тарифы

на доставку продукции фиктивного поставщика равны нулю, т.е. $c_{41} = c_{42} = c_{43} = c_{44} = 0$.

Данные задачи занесем в таблицу 3.2.

Таблица 3.2

Поставщики	Потребители				Запас груза, a_i
	B_1	B_2	B_3	B_4	
A_1	7	3	6	1	360
		160		200	
A_2	5	2	4	8	150
		30	120		
A_3	3	5	7	9	380
	160		220		
A_4	0	0	0	0	110
	110				
Потребность в грузе, b_j	270	190	340	200	1000

Начальный опорный план получим по правилу «минимального элемента».

Загрузку начинаем с клетки, которой соответствует наименьший тариф $c_{ij} \neq 0$ всей матрицы тарифов. В нашем случае минимальным тарифом является $c_{14} = 1$, который соответствует клетке (1;4). В эту клетку вписываем $x_{14} = \min(360; 200) = 200$. Исключаем четвертый столбец. У поставщика A_1 осталось еще 160 единиц продукции, которые поместим в клетку (1;2). Исключаем первую строку. Но потребителю B_2 необходимо еще 30 ед. продукции, которые берем у поставщика A_2 и помещаем в клетку (2;2). Исключаем второй столбец. Оставшиеся у поставщика A_2 120 ед. продукции помещаем в клетку (2;3). Исключаем вторую строку. Необходимые потребителю B_3 220 ед. продукции берем у поставщика A_3 и помещаем в клетку (3;3). Исключаем третий столбец. В пункте A_3 оставшиеся 160 ед. продукции помещаем в клетку (3;1). Исключаем третий столбец. Оставшиеся 110 единиц продукции помещаем в клетку (4;1).

Полученный нами план является невырожденным, так как выполняется условие базисных клеток: $m + n - 1 = 4 + 4 - 1 = 7$, и число заполненных клеток тоже 7. А значит, получено опорное решение, которое запишем матрицей

$$X_1 = \begin{pmatrix} 0 & 160 & 0 & 200 \\ 0 & 30 & 120 & 0 \\ 160 & 0 & 220 & 0 \end{pmatrix}.$$

Методом потенциалов проверим, является ли полученное решение оптимальным. Для этого каждому поставщику и потребителю поставим в соответствие число, называемое потенциалом. Потенциалы выбираем так, чтобы их сумма для каждой загруженной клетки была равна тарифу перевозки единицы груза, т.е.

$$u_i + v_j = c_{ij},$$

где u_i – потенциал i -того поставщика; v_j – потенциал j -того потребителя; c_{ij} – тариф заполненной клетки.

В нашем случае получаем следующие уравнения:

$$u_1 + v_2 = 3,$$

$$u_1 + v_4 = 1,$$

$$u_2 + v_2 = 2,$$

$$u_2 + v_3 = 4,$$

$$u_3 + v_1 = 3,$$

$$u_3 + v_3 = 7,$$

$$u_4 + v_1 = 0.$$

Полагая $v_4 = 0$, получаем следующие значения потенциалов:

$$\begin{array}{llll} u_1 = 1, & u_3 = 3, & v_1 = 0, & v_3 = 4, \\ u_2 = 0, & u_4 = 0, & v_2 = 2, & v_4 = 0. \end{array}$$

Все полученные данные заносим в таблицу 3.3.

Таблица 3.3

Поставщики	Потребители				u_i
	B_1 (270)	B_2 (190)	B_3 (340)	B_4 (200)	
A_1 (360)	7	3	6	1	1
A_2 (150)	5	2	4	8	0
A_3 (380)	3	5	7	9	3
A_4 (110)	0	0	0	0	0
v_j	0	2	4	0	

Далее вычисляем оценки свободных клеток по формуле

$$s_{ij} = c_{ij} - (u_i + v_j).$$

$$s_{11} = c_{11} - (u_1 + v_1) = 7 - (1 + 0) = 6 \geq 0, \quad s_{13} = c_{13} - (u_1 + v_3) = 6 - (1 + 4) = 1 \geq 0,$$

$$s_{21} = c_{21} - (u_2 + v_1) = 5 - (0 + 0) = 5 \geq 0, \quad s_{24} = c_{24} - (u_2 + v_4) = 8 - (0 + 0) = 8 \geq 0,$$

$$s_{32} = c_{32} - (u_3 + v_2) = 5 - (3 + 2) = 0, \quad s_{34} = c_{34} - (u_3 + v_4) = 9 - (3 + 0) = 6 \geq 0,$$

$$s_{42} = c_{42} - (u_4 + v_2) = 0 - (0 + 2) = -2, \quad s_{43} = c_{43} - (u_4 + v_3) = 0 - (0 + 4) = -4,$$

$$s_{44} = c_{44} - (u_4 + v_4) = 0 - (0 + 0) = 0.$$

Так как $s_{42} < 0$ и $s_{43} < 0$, то полученный план не является оптимальным. Наиболее потенциальной клеткой является клетка (4;3). Поэтому строим для клетки (4;3) замкнутый цикл, который в таблице 3.3 выделяем пунктирной линией:

$$(4;3) \rightarrow (3;3) \rightarrow (3;1) \rightarrow (4;1).$$

Свободной клетке условно приписываем знак «+», тогда следующей клетке по ходу часовой стрелки – знак «-» и т.д., знаки чередуются.

В отрицательных вершинах цикла определяем наименьшую загрузку клетки, т.е.

$$\min(x_{33}, x_{41}) = \min(220, 110) = 110.$$

Количество продукции, равное 110 ед., прибавляем к поставкам в клетках со знаком «+» и вычитаем из поставок в клетках со знаком «-».

Получаем новый план, который заносим в таблицу 3.4.

Таблица 3.4

Поставщики	Потребители				u_i
	B_1 (270)	B_2 (190)	B_3 (340)	B_4 (200)	
A_1 (360)	7	3	6	1	5
A_2 (150)	5	2	4	8	4
A_3 (380)	3	5	7	9	7
A_4 (110)	0	0	0	0	0
v_j	-4	-2	0	-4	

Проверим, является ли полученный план оптимальным. Для этого каждому поставщику и потребителю поставим в соответствие потенциал. Вычислим потенциалы так же, как это делали выше:

$$\begin{aligned}u_1 + v_2 &= 3, \\u_1 + v_4 &= 1, \\u_2 + v_2 &= 2, \\u_2 + v_3 &= 4, \\u_3 + v_1 &= 3, \\u_3 + v_3 &= 7, \\u_4 + v_3 &= 0.\end{aligned}$$

Полагая $v_3 = 0$, получим следующие значения потенциалов:

$$\begin{array}{cccc}u_1 = 5, & u_3 = 7, & v_1 = -4, & v_3 = 0, \\u_2 = 4, & u_4 = 0, & v_2 = -2, & v_4 = -4.\end{array}$$

Вычислим оценки свободных клеток:

$$\begin{aligned}s_{11} &= 7 - (5 - 4) = 6 \geq 0, & s_{34} &= 9 - (7 - 4) = 6 \geq 0, \\s_{13} &= 6 - (5 + 0) = 1 \geq 0, & s_{41} &= 0 - (0 - 4) = 4 \geq 0, \\s_{21} &= 5 - (4 - 4) = 5 \geq 0, & s_{42} &= 0 - (0 - 2) = 2 \geq 0, \\s_{24} &= 8 - (4 - 4) = 8 \geq 0, & s_{44} &= 0 - (0 - 4) = 4 \geq 0, \\s_{32} &= 5 - (7 - 2) = 0.\end{aligned}$$

Поскольку все $s_{ij} > 0$, то полученный план является оптимальным, который можно представить в виде матрицы

$$X^* = \begin{pmatrix} 0 & 160 & 0 & 200 \\ 0 & 30 & 120 & 0 \\ 270 & 0 & 110 & 0 \end{pmatrix}.$$

2) Суммарные затраты f_{\min} при этом будут составлять:

$$\begin{aligned}f_{\min} &= 3 \cdot 160 + 1 \cdot 200 + 2 \cdot 30 + 4 \cdot 120 + 3 \cdot 270 + 7 \cdot 110 = \\&= 480 + 200 + 60 + 480 + 810 + 770 = 2800 \text{ (ден.ед.)}\end{aligned}$$

Потребитель B_3 не дополучил 110 единиц продукции.

Ответ: оптимальный план перевозок определяется матрицей

$$X^* = \begin{pmatrix} 0 & 160 & 0 & 200 \\ 0 & 30 & 120 & 0 \\ 270 & 0 & 110 & 0 \end{pmatrix},$$

суммарные затраты при этом составят 2800 ден.ед.

□

Пример 3.2. В пунктах A_i ($i = \overline{1, 3}$) производится однородная продукция в количествах $a_1 = 500, a_2 = 900, a_3 = 100$ единиц. Готовая продукция поставляется в пункты B_j ($j = \overline{1, 4}$), потребности которых составляют $b_1 = 200, b_2 = 650, b_3 = 150, b_4 = 300$ единиц. Стоимость c_{ij} перевозки единицы продукции из пункта A_i в пункт B_j заданы матрицей $C = [c_{ij}]_{3 \times 4}$:

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

Требуется:

- 1) методом потенциалов найти план перевозок продукции, при котором минимизируются суммарные затраты по ее доставке потребителям;
- 2) вычислить суммарные затраты f_{\min} .

Решение.

- 1) Определим суммарные запасы продукции и суммарные потребности

$$a_1 + a_2 + a_3 = 500 + 900 + 100 = 1500,$$

$$b_1 + b_2 + b_3 + b_4 = 200 + 650 + 150 + 300 = 1300.$$

Поскольку $\sum_{i=1}^3 a_i > \sum_{j=1}^4 b_j$, то есть отсутствует баланс между производимой продукцией и потребностями, то данная задача относится к задачам открытого типа. Введем фиктивного потребителя B_5 , который нуждается в продукции в количестве $b_5 = \sum_{i=1}^3 a_i - \sum_{j=1}^4 b_j = 1500 - 1300 = 200$ (ед.). Все та-

рифы на доставку продукции фиктивному потребителю равны нулю, т.е. $c_{15} = c_{25} = c_{35} = 0$.

Данные задачи занесем в таблицу 3.5.

Таблица 3.5

Поставщики	Потребители					u_i
	B_1 (200)	B_2 (650)	B_3 (150)	B_4 (300)	B_5 (200)	
A_1 (500)	7	7	8	4	0	0
A_2 (900)	6	1	2	7	0	0
A_3 (100)	4	7	5	6	0	-2
v_j	6	1	2	4	0	

Начальный опорный план получим по правилу «минимального элемента».

Загрузку начнем с клетки, которой соответствует наименьший тариф $c_{ij} \neq 0$ всей матрицы тарифов. В нашем случае минимальным тарифом является $c_{22} = 1$, который соответствует клетке (2;2). В эту клетку вписываем $x_{22} = \min(650, 900) = 650$. Исключаем второй столбец. У поставщика A_2 осталось еще 250 единиц продукции, которые располагаем следующим образом: 100 единиц в клетку (2;1) и 150 единиц в клетку (2;3). Исключаем вторую строку и третий столбец. Но потребителю B_1 необходимо еще 100 ед. продукции, которые берем у поставщика A_3 и помещаем в клетку (3;1). Исключаем первый столбец и третью строку. 500 единиц продукции у поставщика A_1 размещаем следующим образом: 300 единиц в клетку (1;4) и 200 единиц – в клетку (1;5).

Полученный нами план является вырожденным, так как не выполняется условие базисных клеток: $m + n - 1 = 3 + 5 - 1 = 7$, а число заполненных клеток равно 6.

В одну из базисных клеток, например, в клетку (2;5) таблицы 3.5 вписываем число нуль – «нуль-загрузку», и эту клетку считаем базисной.

Заметим, что нельзя вписывать 0 в клетки (3;2) и (3;3), поскольку образуется цикл. Если вписывать 0 в другие незанятые клетки, то произой-

дет перераспределение продукции, количество которой равно 0. И в итоге заполнится клетка (2;5).

Таким образом, будет выполнено условие занятости базисных клеток: $m + n - 1 = 3 + 5 - 1 = 7$. В результате получено опорное решение, которое запишем матрицей

$$X_1 = \begin{pmatrix} 0 & 0 & 0 & 300 & 200 \\ 100 & 650 & 150 & 0 & 0 \\ 100 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Методом потенциалов проверим, является ли полученное решение оптимальным. Для этого каждому поставщику и потребителю придается число, называемое потенциалом. Потенциалы выбираем так, чтобы их сумма для каждой загруженной клетки была равна тарифу перевозки единицы груза, т.е.

$$u_i + v_j = c_{ij},$$

где u_i – потенциал i -того поставщика; v_j – потенциал j -того потребителя; c_{ij} – тариф заполненной клетки.

В нашем случае получаем следующие уравнения:

$$u_1 + v_4 = 4,$$

$$u_1 + v_5 = 0,$$

$$u_2 + v_1 = 6,$$

$$u_2 + v_2 = 1,$$

$$u_2 + v_3 = 2,$$

$$u_2 + v_5 = 0,$$

$$u_3 + v_1 = 4.$$

Полагая $v_4 = 0$, получаем следующие значения потенциалов:

$$u_1 = 0, \quad v_1 = 6, \quad v_4 = 4,$$

$$u_2 = 0, \quad v_2 = 1, \quad v_5 = 0,$$

$$u_3 = -2, \quad v_3 = 2$$

Все полученные данные заносим в таблицу 3.5.

Далее вычисляем оценки свободных клеток по формуле

$$s_{ij} = c_{ij} - (u_i + v_j).$$

$$s_{11} = c_{11} - (u_1 + v_1) = 7 - (0 + 6) = 1 \geq 0, \quad s_{12} = c_{12} - (u_1 + v_2) = 7 - (0 + 1) = 6 \geq 0,$$

$$s_{13} = c_{13} - (u_1 + v_3) = 8 - (0 + 2) = 6 \geq 0, \quad s_{24} = c_{24} - (u_2 + v_4) = 7 - (0 + 4) = 3 \geq 0,$$

$$s_{32} = c_{32} - (u_3 + v_2) = 7 - (-2 + 1) = 8 \geq 0, \quad s_{33} = c_{33} - (u_3 + v_3) = 5 - (-2 + 2) = 5 \geq 0,$$

$$s_{34} = c_{34} - (u_3 + v_4) = 6 - (-2 + 4) = 4 \geq 0, \quad s_{35} = c_{35} - (u_3 + v_5) = 0 - (-2 + 0) = 2 \geq 0.$$

Поскольку все $s_{ij} > 0$, то полученный план является оптимальным и единственным. Представим его в виде матрицы

$$X^* = \begin{pmatrix} 0 & 0 & 0 & 300 \\ 100 & 650 & 150 & 0 \\ 100 & 0 & 0 & 0 \end{pmatrix}.$$

2) Суммарные затраты f_{\min} при этом будут составлять:

$$f_{\min} = 4 \cdot 300 + 6 \cdot 100 + 1 \cdot 650 + 2 \cdot 150 + 4 \cdot 100 =$$

$$= 1200 + 600 + 650 + 300 + 400 = 3150 \text{ (ден.ед.)}.$$

В пункте A_1 нераспределенными остались 200 единиц продукции.

Ответ: оптимальный план перевозок определяется матрицей

$$X^* = \begin{pmatrix} 0 & 0 & 0 & 300 \\ 100 & 650 & 150 & 0 \\ 100 & 0 & 0 & 0 \end{pmatrix}, \quad \text{суммарные затраты при этом составят}$$

3150 ден. ед.

□

4. ДИСКРЕТНОЕ ПРОГРАММИРОВАНИЕ

4.1. Классические задачи целочисленного программирования

Задачи, сформулированные в терминах линейного программирования и содержащие требование «все или некоторые x_j – целые числа или принадлежат конечному множеству», играют важную роль в исследованиях различных прикладных проблем. Такие задачи были объединены в один раздел математического программирования, который называется *дискретным программированием*.

Дискретное программирование – раздел математического программирования, изучающий экстремальные задачи, в которых на искомые переменные налагается условие дискретности, а область допустимых решений конечна.

Существует огромное количество задач с дискретной природой. Прежде всего, это задачи с физической неделимостью многих факторов и объектов расчета. Дискретными являются задачи с логическими переменными, принимающие только два значения – нуль или единица (вариант отвергается или принимается).

Иногда дискретное программирование называется *целочисленным*. Этот термин некоторые математики считают неправильным, так как, строго говоря, дискретное – не обязательно целочисленное. Например, ряд вместимостей (в м³) 1,3; 1,6; 1,9; ... – дискретный, но не целочисленный. Таким образом, целочисленное программирование правильнее считать частным случаем дискретного.

В формальном плане область определения задач целочисленного программирования (ЗЦП) представляет собой совокупность конечного числа точек (если все x_j целочисленные) или множество непересекающихся линий, плоскостей, гиперплоскостей (если целочисленные лишь отдельные x_j). Соответствующие коррективы вносятся и в методологию поиска оптимальных решений, причем серьезные трудности вызывает разработка теоретически обоснованных методов получения экстремальных плана задачи X^* и значения целевой функции $Z^*(X^*)$.

Укажем некоторые модели задач дискретного программирования.

Общая задача целочисленного программирования

Математическая модель задачи имеет такой же вид, как модель ЗЛП, только на переменные накладывается условие целочисленности, т.е. $x_j \in \mathbb{Z}$.

Задача о контейнерных перевозках (задача о рюкзаке)

Постановка задачи: контейнер оборудован m отсеками вместимостью b_i ($i = \overline{1, m}$) для перевозки n видов продукции P_j ($j = \overline{1, n}$). Виды продукции характеризуются свойством неделимости, т.е. их можно брать в

количестве 0, 1, 2, ... единиц. Пусть a_{ij} – расход i -того отсека для перевозки единицы j -той продукции. Обозначим через c_j полезность единицы j -той продукции (это может быть цена реализации, прибыль и др.). Требуется найти план $X = (x_1; x_2; \dots, x_n)$ (x_j – количество единиц j -той продукции, погруженной в контейнер) перевозки, при котором максимизируется общая полезность перевозки. Модель задачи примет вид

$$Z = \sum_{j=1}^n c_j x_j \rightarrow \max, \quad (4.1)$$

при ограничениях на вместимость отсеков

$$\sum_{j=1}^n a_{ij} x_j \leq b_i \quad (i = \overline{1, m}), \quad (4.2)$$

условиях неотрицательности

$$x_j \geq 0 \quad (j = \overline{1, n}), \quad (4.3)$$

условиях целочисленности.

$$x_j - \text{целые} \quad (j = \overline{1, n}). \quad (4.4)$$

Задача о рюкзаке является частным случаем задачи о контейнерных перевозках. В задаче о рюкзаке предусмотрены перевозки с одним отсеком, и каждый вид продукции (предмет) может быть взят или нет, т.е. $x_j \in \{0; 1\}$: $x_j = 1$, если предмет j -того вида берется, и $x_j = 0$ в противном случае.

Задача о назначении (проблема выбора, задача о женихах и невестах)

Один из видов этих задач рассматривался в разделе «Элементы теории графов», параграф «Паросочетания».

Постановка задачи: имеется n исполнителей, которые могут выполнять n различных работ. Известна полезность c_{ij} , связанная с выполнением i -тым исполнителем j -той работы ($i = \overline{1, n}, j = \overline{1, n}$). Необходимо назначить исполнителей на работы, чтобы добиться максимальной полезности при условии, что каждый исполнитель может быть назначен только на одну работу и за каждой работой должен быть закреплен только один исполнитель.

Для составления математической модели задачи обозначим через x_{ij} факт назначения или неназначения i -того исполнителя на j -тую работу.

Так как количество исполнителей равно количеству работ, и каждый из них может быть назначен только на одну работу, то x_{ij} должны принимать только два значения: 1 или 0.

$$x_{ij} = \begin{cases} 1, & \text{если } i\text{-тый исполнитель назначается на } j\text{-тую работу,} \\ 0, & \text{в противном случае} \end{cases}$$

Получаем следующую математическую модель задачи: найти план назначения x_{ij} , который максимизирует суммарную полезность назначений

$$Z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \max, \quad (4.5)$$

при следующих ограничениях:

каждый исполнитель назначается только на одну работу

$$\sum_{j=1}^n x_{ij} = 1 \quad (i = \overline{1, n}), \quad (4.6)$$

на каждую работу назначается только один исполнитель

$$\sum_{i=1}^n x_{ij} = 1 \quad (j = \overline{1, n}), \quad (4.7)$$

условия неотрицательности и целочисленности (булевости)

$$x_{ij} \geq 0, \quad x_{ij} \in \{0; 1\}. \quad (4.8)$$

Заметим, что задача о назначении, – частный случай транспортной задачи при $a_i = 1$, $b_j = 1$. Однако с учетом специфики задачи для ее решения разработаны специальные, более эффективные, алгоритмы.

Задача коммивояжера (бродячего торговца)

Математическая модель этой задачи была введена в разделе «Элементы теории графов», «Гамильтонов цикл».

4.2. Суть методов дискретной оптимизации

Во всех случаях решение ЗЦП, казалось бы, может быть найдено обычными методами с отброшенными условиями целочисленности и последующим округлением нецелых переменных в ответе. Однако такое ок-

ругление может привести к решению, далекому от оптимального. Рассмотрим, например, геометрическую интерпретацию задачи дискретного линейного программирования.

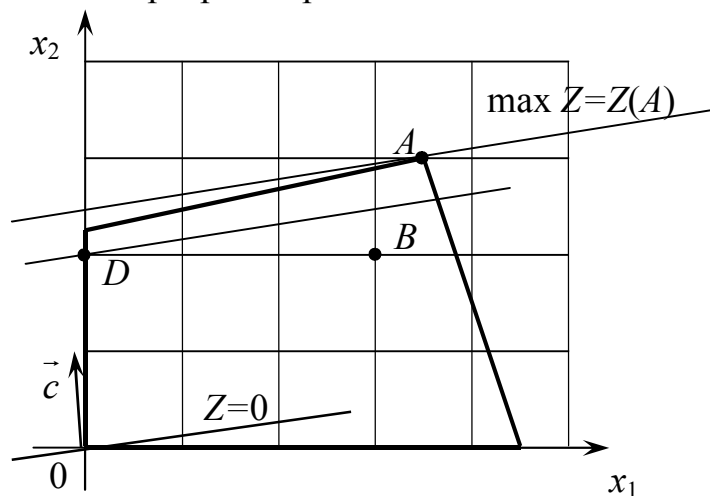


Рис. 4.1

Оптимальным решением нецелочисленной задачи служит точка A . Как следует из рисунка 4.1, точка B ближе всего к точке A в смысле округления. Но целевой точкой, находящейся ближе всего к разрешающей линии уровня $\max Z$, является точка D .

Таким образом, попытка решить задачу с отброшенным условием целочисленности и последующим округлением полученного оптимального плана до ближайших целых значений не всегда состоятельна.

С практической точки зрения подобный подход допустим в тех случаях, когда значения переменных, образующих оптимальное решение исходной задачи, достаточно велики и погрешностями округления можно пренебречь.

В первом приближении методы целочисленной оптимизации можно разделить на две основные группы: точные и приближенные. К точным относятся методы отсечения и комбинаторные методы (метод ветвей и границ). Это универсальные методы дискретной оптимизации. Кроме универсальных, имеется много специальных точных методов, учитывающих специфику задачи. Однако точные методы имеют слабую сходимость. Многие экспериментальные и прикладные задачи не удалось решить точными методами за десятки и сотни тысяч итераций, хотя их конечность теоретически доказана. Трудности машинной реализации точных методов привели к появлению различного рода приближенных методов, построенных на использовании особенностей конкретной задачи. Среди приближенных методов наметились два направления: 1) разработка детерминированных эвристических алгоритмов, учитывающих специфику задачи; 2) использование случайного поиска в сочетании с локальной оптимизацией.

Общая идея решения задачи дискретного программирования методами отсечения состоит в следующем. Исходная задача решается сначала

без учета ограничений целочисленности. Если полученный оптимальный план удовлетворяет условиям целочисленности, то задача решена. В противном случае к ограничениям исходной задачи добавляется новое, обладающее следующими свойствами: 1) полученный нецелочисленный план нарушает это ограничение; 2) любой целочисленный допустимый план исходной задачи заведомо удовлетворяет и новому ограничению. Затем задача решается с учетом нового ограничения. В случае необходимости добавляется еще одно ограничение и т.д. Геометрически каждому новому ограничению соответствует поверхность, которая отсекает от области допустимых решений некоторую его часть с оптимальной точкой с нецелыми координатами, но не затрагивает ни одной из целочисленных точек этого многогранника.

На основе этой идеи американский математик Р. Гомори предложил ряд сходящихся алгоритмов решения задач дискретного линейного программирования. Ему удалось обосновать правила построения дополнительных ограничений и доказать конечность алгоритмов.

Для решения задач дискретного (особенно нелинейного) программирования получили широкое распространение комбинаторные методы направленного частичного перебора допустимых планов. Из них наиболее универсален *метод ветвей и границ*.

4.3. Метод ветвей и границ

Метод ветвей и границ относится к группе комбинаторных методов. Комбинаторные методы, исходя из конечности числа допустимых планов задачи, заменяют полный перебор всех планов их частичным направленным перебором. Комбинаторные методы в значительно меньшей степени подвержены в процессе вычислений влиянию ошибок округления, поэтому являются более предпочтительными по сравнению с методами отсечения. Метод ветвей и границ – один из наиболее эффективных методов решения задач комбинаторного типа.

Перейдем к изложению идеи метода ветвей и границ. Для этого рассмотрим общую задачу дискретного программирования

$$Z = f(X) \rightarrow \max ,$$

$$X \in \Omega ,$$

где Ω – конечное множество допустимых планов. Метод ветвей и границ предполагает:

1) нахождение верхней границы (оценки) функции $f(X)$, $X \in \Omega$, т.е. такого числа $\varphi_0(\Omega)$, что для любых $X \in \Omega$

$$f(X) \leq \varphi_0(\Omega).$$

Если при этом удастся найти такой план X_0 задачи, для которого выполняется равенство

$$f(X_0) = \varphi_0(\Omega),$$

то X_0 – оптимальный план задачи.

2) если оптимальный план не найден, то некоторым способом выполняется разбиение множества Ω на конечное число непересекающихся подмножеств Ω_r^1 :

$$\Omega = \bigcup_{r=1}^{r_1} \Omega_r^1, \quad \bigcap_{r=1}^{r_1} \Omega_r^1 = \emptyset,$$

и для каждого из этих подмножеств устанавливается верхняя граница $\varphi_r^1(\Omega_r^1)$ ($r = \overline{1, r_1}$). Если при этом удастся найти такой план $X_r^1 \in \Omega_r^1$ ($r = \overline{1, r_1}$), что выполняется соотношение

$$f(X_r^1) = \varphi_r^1(\Omega_r^1) = \varphi_0(\Omega) \quad (r = \overline{1, r_1}),$$

то X_r^1 – оптимальный план задачи. Если же такой план не найден, то выбирается подмножество Ω_r^1 с наибольшей верхней границей (перспективное подмножество), которое разбиваем на несколько непересекающихся подмножеств Ω_s^2 ($s = \overline{1, s_1}$). Для каждого нового подмножества находится верхняя граница $\varphi_s^2(\Omega_s^2)$ ($s = \overline{1, s_1}$). Если будет найден такой план X_s^2 , что

$$f(X_s^2) = \varphi_s^2(\Omega_s^2) = \varphi_r^1(\Omega_r^1),$$

то X_s^2 – оптимальный план задачи. Если оптимальный план не найден, то дальнейшему ветвлению подвергается подмножество с наибольшей верхней границей, и т.д. Процесс продолжается до получения оптимального плана. Способы ветвления и нахождения верхних границ выбираются для каждой конкретной задачи дискретного программирования. Процесс сопровождается построением дерева ветвления, рис. 4.2.

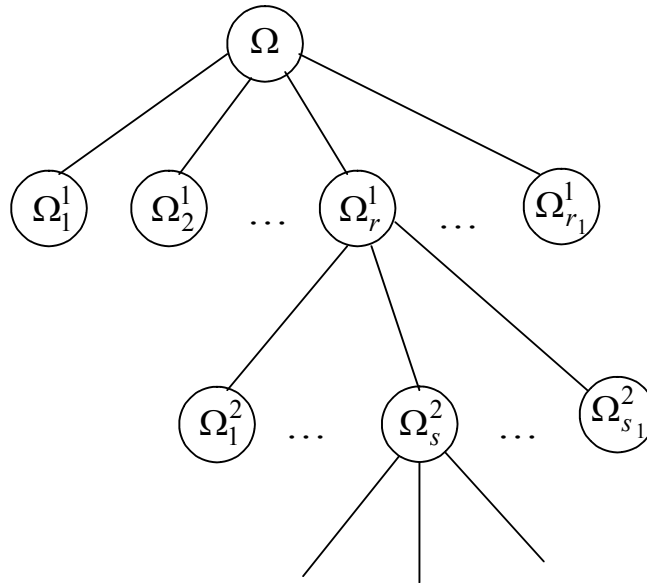


Рис. 4.2

4.4. Решение общей задачи целочисленного программирования

Математическая модель общей задачи целочисленного программирования (ЦЗЛП) имеет такой же вид, как модель ЗЛП, только на переменные накладывается условие целочисленности. Метод ветвей и границ позволяет найти точное оптимальное решение исходной задачи, удовлетворяющее требованиям целочисленности (если оно существует).

Алгоритм метода ветвей и границ решения ЦЗЛП

1. Ищется оптимальное решение ЗЛП без учета целочисленности неизвестных. Если при этом получится решение, удовлетворяющее всем требованиям целочисленности, то решение исходной задачи найдено. В противном случае множество допустимых планов задачи разбивается (ветвится) на непересекающиеся подмножества следующим образом.

Пусть неизвестная x_j должна быть целой, но в полученном решении она приняла нецелое значение x_j^* . Тогда во множестве допустимых планов существует целый «слой», задаваемый неравенством

$$[x_j^*] < x_j < [x_j^*] + 1,$$

все планы из которого также не удовлетворяют данному условию целочисленности. Этот «слой» следует исключить из дальнейшего рассмотрения.

Таким образом, множество допустимых планов исходной задачи разбивается на два подмножества:

– первое подмножество удовлетворяет дополнительному неравенству

$$x_j \leq [x_j^*], \quad (4.9)$$

– второе подмножество – дополнительному неравенству

$$x_j \geq [x_j^*] + 1. \quad (4.10)$$

2. Производится поиск решений ЗЛП отдельно в каждом подмножестве допустимых планов с учетом дополнительного ограничения (4.1) или (4.2) соответственно. Если при этом на каком-либо подмножестве получится оптимальное решение, удовлетворяющее всем требованиям целочисленности, то оно фиксируется как наилучшее целочисленное для данного подмножества (поиск в этом подмножестве больше не производится). Если впоследствии на другом подмножестве будет найдено целочисленное решение с лучшим, чем уже зафиксированное, значением целевой функции, то оно фиксируется вместо предыдущего. Все подмножества, на которых значение целевой функции хуже зафиксированного, исключаются из дальнейшего рассмотрения как неперспективные, исключаются также подмножества, на которых решение ЗЛП не существует.

3. Из всех оставшихся подмножеств допустимых планов выбирается наиболее перспективное (то, на котором целевая функция достигла наилучшего значения). Это подмножество, в свою очередь, разбивается на два непересекающихся подмножества способом, описанным на первом шаге алгоритма, и над ними производятся действия второго шага. При этом все дополнительные ограничения вида (4.1) или (4.2), установленные для разбиваемого подмножества при предыдущих ветвлениях, сохраняются для обоих вновь полученных подмножеств.

Затем для нового набора подмножеств (неисключенные плюс добавленные) выполняются действия данного шага и т.д. В результате будет получен один из следующих двух исходов: либо все подмножества будут исключены (исходная задача не имеет решений), либо последнее зафиксированное целочисленное решение будет наилучшим по сравнению с возможным для остальных подмножеств (данное решение – оптимальное целочисленное решение исходной задачи).

Пример 4.1. Станкостроительный завод выпускает две модели станков с ЧПУ. Изготовление частей станков и их сборка производится последовательно в трех цехах. Мощности цехов составляют 300, 170 и

200 чел.×дней в декаду. В первом цехе для изготовления частей одного станка первой модели требуется 6 чел.×дней в декаду, для второй модели – 4 чел.×дней в декаду; во втором цехе на сборку станка первой модели требуется 3 чел.×дней в декаду, для второй модели – 4 чел.×дней в декаду; в третьем цехе – соответственно 2 и 6 чел.×дней в декаду. Прибыль, получаемую заводом от продажи одного станка с ЧПУ каждой модели, составляют соответственно 15 и 13 млн. ден. ед. Известно, что все выпущенные станки будут проданы. Каким должен быть оптимальный выпуск станков в течение декады?

Решение. Для построения математической модели составим таблицу, в которую внесем все исходные данные задачи.

Таблица 4.1

Цех завода	Вид (модель) станка		Ограничения (фонд) ресурсов
	M_1	M_2	
I	6	4	300
II	3	4	170
III	2	6	200
Прибыль, млн. ден. ед.	15	13	

Пусть $X = (x_1, x_2)$ – план выпуска станков с ЧПУ в течение декады. Тогда математическая модель задачи принимает следующий вид:

– целевая функция

$$Z = 15x_1 + 13x_2 \rightarrow \max,$$

– система ограничений

$$\begin{cases} 6x_1 + 4x_2 \leq 300, \\ 3x_1 + 4x_2 \leq 170, \\ 2x_1 + 6x_2 \leq 200, \end{cases}$$

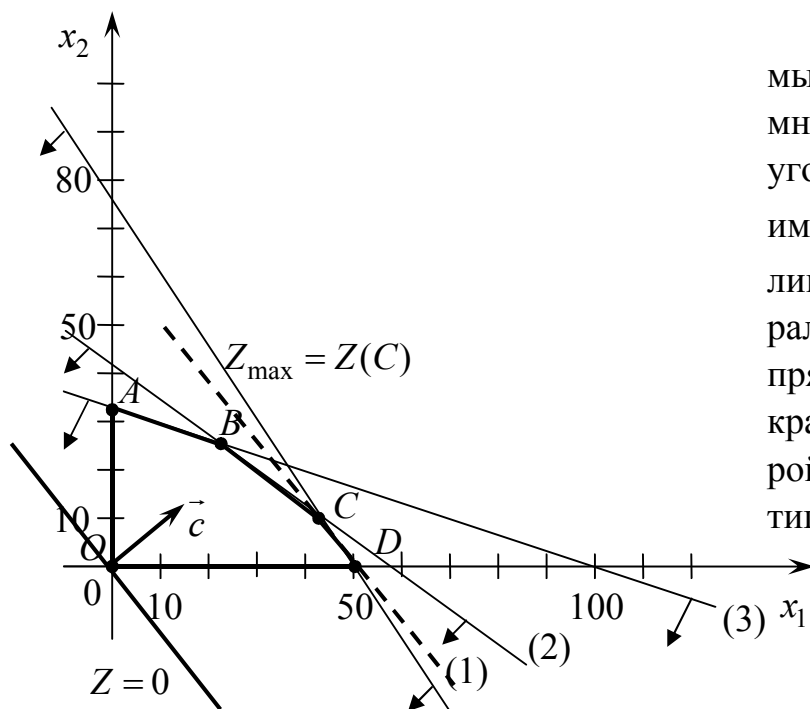
$$x_1 \geq 0, x_2 \geq 0,$$

$$x_1, x_2 - \text{целые.}$$

Сначала решим задачу графическим способом, не учитывая условия целочисленности переменных x_1, x_2 .

Построим множество допустимых планов задачи Ω . На плоскости Ox_1x_2 строим соответствующие данным ограничениям-неравенствам граничные прямые:

$$(1) \begin{cases} 6x_1 + 4x_2 = 300, \\ 3x_1 + 2x_2 = 150, \end{cases} \quad (2) \begin{cases} 3x_1 + 4x_2 = 170, \\ x_1 + 3x_2 = 100, \end{cases} \quad (3) \begin{cases} 2x_1 + 6x_2 = 200, \\ x_1 + 3x_2 = 100, \end{cases} \quad (4) \begin{cases} x_1 = 0, \\ x_2 = 0. \end{cases}$$



Множество допустимых планов задачи Ω – множество точек многоугольника $OABCD$. Строим вектор $\vec{c} = (15; 13)$ и линию уровня $Z = 0$. Параллельным переносом прямой $Z = 0$ находим крайнюю точку C , в которой целевая функция достигает максимума.

Координаты точки C определяются из решения системы

$$\begin{cases} 3x_1 + 2x_2 = 150, & (1) \\ 3x_1 + 4x_2 = 170, & (2) \end{cases}$$

откуда $C(43\frac{1}{3}; 10)$. Следовательно,

$$Z_{\max} = Z(43\frac{1}{3}; 10) = 15 \cdot \frac{130}{3} + 13 \cdot 10 = 650 + 130 = 780 \text{ (ден. ед.)}.$$

Так как по условию задачи x_1 и x_2 – целые числа, то план $X_0 = (43\frac{1}{3}; 10)$ не является оптимальным.

Продолжим решение задачи. Выберем любую нецелочисленную переменную. В нашем случае – это одна переменная $x_1 = 43\frac{1}{3}$. Разобьем множество Ω планов задачи на два непересекающихся подмножества:

$$\Omega = \Omega_1^1 \cup \Omega_2^1, \quad \Omega_1^1 \cap \Omega_2^1 = \emptyset.$$

$$\Omega_1^1 = \left\{ X \mid X \in \Omega \text{ и } x_1 \leq \left[43\frac{1}{3} \right] \right\},$$

$$\Omega_2^1 = \left\{ X \mid X \in \Omega \text{ и } x_1 \geq \left[43\frac{1}{3} \right] + 1 \right\}, \text{ где } \left[43\frac{1}{3} \right] = 43.$$

Составим математические модели для каждого подмножества Ω_1^1 и Ω_2^1 планов задачи.

Для Ω_1^1 :

целевая функция

$$Z = 15x_1 + 13x_2 \rightarrow \max,$$

система ограничений

$$\begin{cases} 6x_1 + 4x_2 \leq 300, \\ 3x_1 + 4x_2 \leq 170, \\ 2x_1 + 6x_2 \leq 200, \\ x_1 \leq 43, \\ x_1 \geq 0, x_2 \geq 0, \\ x_1, x_2 - \text{целые} \end{cases}$$

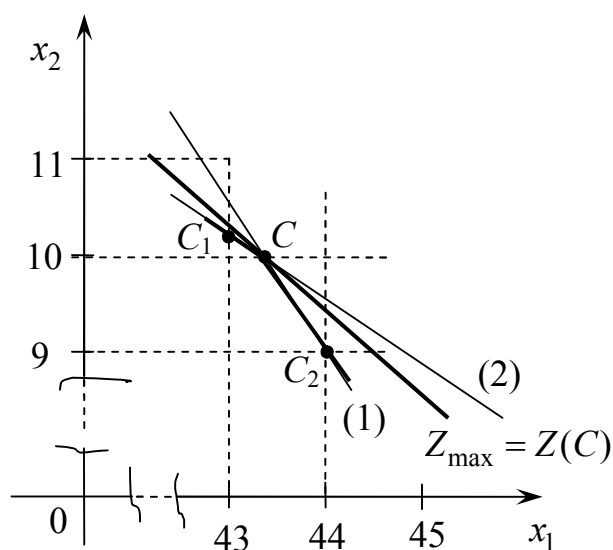
Для Ω_2^1 :

целевая функция

$$Z = 15x_1 + 13x_2 \rightarrow \max .$$

система ограничений

$$\begin{cases} 6x_1 + 4x_2 \leq 300, \\ 3x_1 + 4x_2 \leq 170, \\ 2x_1 + 6x_2 \leq 200, \\ x_1 \geq 44, \\ x_1 \geq 0, x_2 \geq 0, \\ x_1, x_2 - \text{целые} \end{cases}$$



С помощью рисунка найдем план задачи для каждого подмножества Ω_1^1 и Ω_2^1 .

Для Ω_1^1 максимум достигается в точке C_1 , координаты которой определяются из решения системы

$$\begin{cases} x_1 = 43, \\ 3x_1 + 4x_2 = 170. \end{cases} \quad (2)$$

Откуда $C_1(43; 10,25)$.

Следовательно,

$$Z_{\max} = Z(43; 10,25) = 15 \cdot 43 + 13 \cdot 10,25 = 645 + 133,25 = 778,25 \text{ (ден. ед.)}.$$

Для Ω_2^1 максимум достигается в точке C_2 , координаты которой определяются из решения системы

$$\begin{cases} 3x_1 + 2x_2 = 150, \\ x_1 = 44. \end{cases} \quad (1)$$

Откуда $C_2(44; 9)$. Следовательно,

$$Z_{\max} = Z(44; 9) = 15 \cdot 44 + 13 \cdot 9 = 660 + 117 = 777 \text{ (ден. ед.)}.$$

Итак, для Ω_1^1 план задачи $X_1^1 = (43; 10, 25)$, который не является целочисленным, а для $\Omega_2^1 - X_2^1 = (44; 9)$, который является целочисленным.

То подмножество, для которого получили нецелочисленный план задачи, подлежит разбиению на два непересекающихся подмножества.

В нашем случае множество Ω_1^1 разбиваем на два непересекающихся подмножества Ω_1^2 и Ω_2^2 :

$$\Omega_1^1 = \Omega_1^2 \cup \Omega_2^2, \quad \Omega_1^2 \cap \Omega_2^2 = \emptyset.$$

$$\Omega_1^2 = \{X \mid X \in \Omega \text{ и } x_1 \leq 43, x_2 \leq [10, 25]\},$$

$$\Omega_2^2 = \{X \mid X \in \Omega \text{ и } x_1 \leq 43, x_2 \geq [10, 25] + 1\}, \text{ где } [10, 25] = 10.$$

Составим математические модели для каждого подмножества Ω_1^2 и Ω_2^2 планов задачи.

Для Ω_1^2 :

целевая функция

$$Z = 15x_1 + 13x_2 \rightarrow \max,$$

система ограничений

$$\begin{cases} 6x_1 + 4x_2 \leq 300, \\ 3x_1 + 4x_2 \leq 170, \\ 2x_1 + 6x_2 \leq 200, \\ x_1 \leq 43, \\ x_2 \leq 10, \\ x_1 \geq 0, x_2 \geq 0, \\ x_1, x_2 - \text{целые} \end{cases}$$

Для Ω_2^2 :

целевая функция

$$Z = 15x_1 + 13x_2 \rightarrow \max,$$

система ограничений

$$\begin{cases} 6x_1 + 4x_2 \leq 300, \\ 3x_1 + 4x_2 \leq 170, \\ 2x_1 + 6x_2 \leq 200, \\ x_1 \leq 43, \\ x_2 \geq 11, \\ x_1 \geq 0, x_2 \geq 0, \\ x_1, x_2 - \text{целые} \end{cases}$$

Найдем план задачи для каждого подмножества Ω_1^2 и Ω_2^2 . Для Ω_1^2 план задачи $X_1^2 = (43; 10)$, значение целевой функции равно

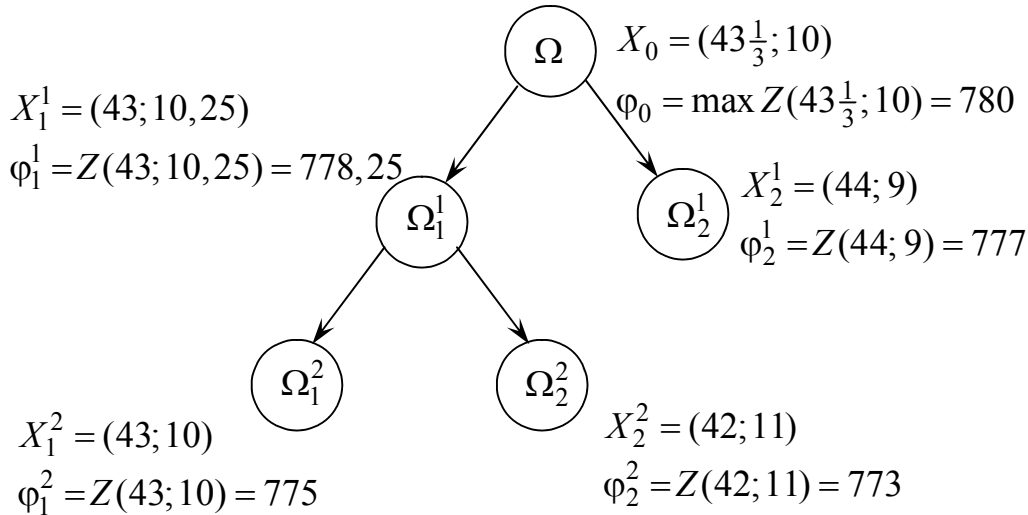
$$Z_{\max} = Z(43; 10) = 15 \cdot 43 + 13 \cdot 10 = 645 + 130 = 775 \text{ (ден. ед.)}.$$

Для $\Omega_2^2 - X_2^2 = (42; 11)$, значение целевой функции равно

$$Z_{\max} = Z(42; 11) = 15 \cdot 42 + 13 \cdot 11 = 630 + 143 = 773 \text{ (ден. ед.)}.$$

Планы X_2^1 , X_1^2 и X_2^2 являются целочисленными. Задача решена.

Весь ход решения данной задачи можно изобразить в виде следующего дерева ветвления:



Оптимальным планом является план $X_2^1 = (44; 9)$, при котором прибыль составит 777 млн. ден. ед. Согласно этому плану предприятие должно выпускать станки с ЧПУ первой модели в количестве 44 единиц, а второй модели – в количестве 9 единиц в декаду.

Ответ: предприятие должно выпускать станки с ЧПУ первой модели в количестве 44 единиц, второй модели – в количестве 9 единиц в декаду, что обеспечит прибыль в 777 млн. ден. ед. \square

4.5. Задача о назначении

Математическая модель задачи о назначении, в которой рассматривается целевая функция, максимизирующая суммарную полезность назначений, была введена в п. 4.1, Но в группе задач о назначении есть задачи, в которых рассматривается целевая функция, минимизирующая некоторую суммарную полезность. В этом случае, задачу о назначении, которая ставится при условии получения максимального эффекта, можно свести к задаче на минимум.

Пусть $C = [c_{ij}]_{n \times n}$ – матрица эффективности.

В каждом столбце найдем максимальный элемент $l_j = \max_i c_{ij}$.

Построим матрицу $C' = [c'_{ij}] = [l_j - c_{ij}]$.

Задача (4.5) – (4.8) будет эквивалентна задаче с этими же ограничениями и целевой функцией

$$f' = \sum_{i=1}^n \sum_{j=1}^n (l_j - c_{ij})x_{ij} \rightarrow \min. \quad (4.11)$$

Для решения задачи о назначении используется *алгоритм венгерского метода* (Е. Эгервари (1932 г.) – венгерский математик).

Основной принцип венгерского метода заключается в том, что оптимальность назначения не нарушается при уменьшении или увеличении элементов строки (столбца) матрицы эффективности на одно и то же число. Уменьшением элементов строк и столбцов с сохранением условия $c_{ij} \geq 0$ ($i = \overline{1, n}, j = \overline{1, n}$) стараются достичь того, чтобы сумма

$$S = \sum_{i=1}^n \sum_{j=1}^n c'_{ij}x_{ij} = 0, \text{ где } c'_{ij} - \text{преобразованные элементы матрицы эффектив-}$$

тивности, т.е. если $x_{ij} = 1$, то $c'_{ij} = 0$. Очевидно, что при этом условии сумма S будет минимальной.

Алгоритм венгерского метода

Этап 1. *Приведение матрицы C .*

Пусть исходная задача имеет вид (4.5) – (4.8)

Найдем $l_j = \max_i \{c_{ij}\}$.

Заменим задачу максимизации задачей минимизации

$$f' = \sum_{i=1}^n \sum_{j=1}^n c'_{ij}x_{ij} \rightarrow \min,$$

где $c'_{ij} = l_j - c_{ij}$.

В каждой строке матрицы $C' = [c'_{ij}]$ найдем минимальный элемент, который вычтем из всех элементов соответствующей строки матрицы. Перейдем к матрице

$$C'' = [c'_{ij} - \min_j c'_{ij}] = [c''_{ij}].$$

Заметим, что данное преобразование не изменит решения исходной задачи. В результате преобразования в каждой строке матрицы C'' появляется хоть бы один нуль.

Этап 2. Поиск назначения.

Выбираем один из нулей, помечаем его, например, точкой или звездочкой (или еще каким-нибудь способом), а остальные нули строки и столбца, в которых стоит выбранный нуль, перечеркиваем. Далее переходим к следующему нулю. И так до тех пор, пока каждый нуль будет либо помечен, либо перечеркнут. Помеченные нули составят назначения. Если число помеченных нулей равно n , то назначение будет полным. Полагая соответствующие им переменные x_{ij} равными единице, а все остальные – равными нулю, получим оптимальный план X^* назначения. Задача решена.

Если назначение неполное, переходим к следующему этапу.

Замечание. Пометку рекомендуется начинать со строк и столбцов, содержащих минимальное количество нулей.

Этап 3. Выделение в матрице C подматрицы, не содержащей нулей.

3(1) помечаем (точкой, звездочкой) строки, не содержащие помеченных нулей;

3(2) помечаем столбцы, содержащие перечеркнутый нуль в одной из помеченных строк;

3(3) помечаем строки, содержащие помеченный нуль хотя бы в одном из помеченных столбцов. От 3(3) опять переходим к 3(2).

Пункты 3(2) и 3(3) повторяются до тех пор, пока есть что пометать.

После этого вычеркиваем непомеченные строки и помеченные столбцы. Незачеркнутые элементы составят подматрицу, не содержащую нулей.

Этап 4. Перемещение нулей.

Среди незачеркнутых элементов матрицы выбираем минимальный, вычитаем его из каждого невычеркнутого столбца и прибавляем к каждой вычеркнутой строке. Получим хотя бы один дополнительный нуль.

Переходим на 2-й этап. Пометку следует начинать со вновь полученного нуля. И так до тех пор, пока не получим полное назначение.

Пример 4.2. Решить задачу о назначении исполнителей на работы, если матрица эффективностей имеет вид

$$C = \begin{pmatrix} 10 & 7 & 6 & 9 \\ 3 & 7 & 5 & 1 \\ 5 & 6 & 8 & 3 \\ 6 & 11 & 7 & 5 \end{pmatrix}.$$

Решение.

Этап 1.

Найдем максимальный элемент в каждом из столбцов матрицы и вычтем каждый элемент матрицы C из максимального элемента соответствующего столбца: $l_1 = 10$; $l_2 = 11$; $l_3 = 8$; $l_4 = 9$.

Тогда

$$C = \begin{pmatrix} 10 & 7 & 6 & 9 \\ 3 & 7 & 5 & 1 \\ 5 & 6 & 8 & 3 \\ 6 & 11 & 7 & 5 \\ \hline 10 & 11 & 8 & 9 \end{pmatrix} \rightarrow C' = \begin{pmatrix} 0 & 4 & 2 & 0 & | & 0 \\ 7 & 4 & 3 & 8 & | & 3 \\ 5 & 5 & 0 & 6 & | & 0 \\ 4 & 0 & 1 & 4 & | & 0 \end{pmatrix}.$$

В полученной матрице C' найдем минимальный элемент в каждой из строк (они записаны за вертикальной чертой в матрице C') и вычтем из соответствующих элементов матрицы C' :

В результате получится матрица C'' :

$$C'' = \begin{pmatrix} 0 & 4 & 2 & 0 \\ 4 & 1 & 0 & 5 \\ 5 & 5 & 0 & 6 \\ 4 & 0 & 1 & 4 \end{pmatrix}.$$

Этап 2.

$$C'' = \begin{pmatrix} 0^* & 4 & 2 & \cancel{0} \\ 4 & 1 & 0^* & 5 \\ 5 & 5 & \cancel{0} & 6 \\ 4 & 0^* & 1 & 4 \end{pmatrix}$$

Помечаем звездочкой нуль, стоящий в первой строке и первом столбце, а нуль, не помеченный звездочкой и стоящий в первой строке, перечеркиваем. Помечаем нуль, стоящий во второй строке, а нуль, не помеченный звездочкой и стоящий в третьем столбце, перечеркиваем. Помечаем нуль, стоящий в четвертой строке.

Помеченные нули составляют назначение. Так как число помеченных нулей меньше $n = 4$, то назначение неполное.

Переходим к следующему этапу.

Этап 3.

$$\left(\begin{array}{cccc|c} 0^* & 4 & 2 & \cancel{0} & \\ 4 & 1 & 0^* & 5 & * \\ 5 & 5 & \cancel{0} & 6 & * \\ 4 & 0^* & 1 & 4 & \\ \hline & & * & & \end{array} \right)$$

Помечаем звездочкой третью строку, не содержащую помеченный нуль. В третьей строке перечеркнутый нуль стоит в третьем столбце, поэтому помечаем этот столбец звездочкой. В третьем столбце помеченный нуль стоит во второй строке, поэтому помечаем вторую строку. Поскольку во второй строке перечеркнутых нулей нет, то пометку строк и столбцов прекращаем.

После этого мысленно вычеркиваем непомеченные строки и помеченный столбец. Незачеркнутые элементы составят подматрицу, которая не содержит нулей.

Этап 4.

Среди незачеркнутых элементов матрицы выбираем минимальный. Это элемент 1.

Вычитаем 1 из каждого элемента незачеркнутого столбца и прибавляем 1 к каждому элементу вычеркнутой строки:

$$\left(\begin{array}{cccc} 0-1+1 & 4-1+1 & 2+1 & 0-1+1 \\ 4-1 & 1-1 & 0 & 5-1 \\ 5-1 & 5-1 & 0 & 6-1 \\ 4-1+1 & 0-1+1 & 1+1 & 4-1+1 \end{array} \right)$$

В результате получаем следующую матрицу:

$$\left(\begin{array}{cccc} 0 & 4 & 3 & 0 \\ 3 & 0 & 0 & 4 \\ 4 & 4 & 0 & 5 \\ 4 & 0 & 2 & 4 \end{array} \right)$$

Переходим на второй этап и повторяем все операции.

Проводим пометку и вычеркивание нулей:

$$\left(\begin{array}{cccc} 0^* & 4 & 3 & \cancel{0} \\ 3 & 0^* & \cancel{0} & 4 \\ 4 & 4 & 0^* & 5 \\ 4 & \cancel{0} & 2 & 4 \end{array} \right)$$

Так как помечено нулей меньше $n = 4$, то назначение неполное.

Далее переходим к 3-му и 4-му этапам.

Производим пометку строк и столбцов.

$$\left(\begin{array}{cccc|c} 0^* & 4 & 3 & \cancel{0} & \\ 3 & 0^* & \cancel{0} & 4 & * \\ 4 & 4 & 0^* & 5 & * \\ 4 & \cancel{0} & 2 & 4 & * \\ \hline & * & * & & \end{array} \right)$$

Мысленно вычеркиваем непомеченные строки и помеченные столбцы. В полученной подматрице определяем минимальный элемент – это 3. Вычитаем его из каждого незачеркнутого столбца и прибавляем к каждой вычеркнутой строке.

Далее

$$\left(\begin{array}{cccc} 0-3+3 & 4+3 & 3+3 & 0-3+3 \\ 3-3 & 0 & 0 & 4-3 \\ 4-3 & 4 & 0 & 5-3 \\ 4-3 & 0 & 2 & 4-3 \end{array} \right)$$

В результате получаем матрицу

$$\left(\begin{array}{cccc} 0 & 7 & 6 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 4 & 0 & 2 \\ 1 & 0 & 2 & 1 \end{array} \right)$$

Снова переходим ко второму этапу и проводим пометку и вычеркивание нулей.

$$\left(\begin{array}{cccc} \cancel{0} & 7 & 6 & 0^* \\ 0^* & \cancel{0} & \cancel{0} & 1 \\ 1 & 4 & 0^* & 2 \\ 1 & 0^* & 2 & 1 \end{array} \right)$$

В этот раз начнем помечать звездочкой нуль, стоящий в четвертой строке и втором столбце. Нуль, стоящий во втором столбце и второй строке вычеркиваем. Помечаем звездочкой нуль, стоящий в третьей строке; нуль, стоящий в третьем столбце и во второй строке вычеркиваем. Помечаем звездочкой нуль, стоящий во

второй строке и первом столбце. Вычеркиваем нуль, стоящий в первой строке и первом столбце. Помечаем звездочкой нуль, стоящий в первой строке и четвертом столбце.

Поскольку число помеченных нулей равно 4, то задача решена. Оптимальный план задачи представим в виде матрицы, в которой номер строки соответствует номеру исполнителя, а номер столбца – номеру выполняемой работы:

$$X^* = \left(\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 0 & 0 & 1 \\ 2 & 1 & 0 & 0 & 0 \\ 3 & 0 & 0 & 1 & 0 \\ 4 & 0 & 1 & 0 & 0 \end{array} \right).$$

По данной матрице можно сделать следующий вывод: 1-й исполнитель назначается на 4-ю работу; 2-й исполнитель – на 1-ю работу; 3-й исполнитель – на 3-ю работу; 4-й исполнитель – на 2-ю работу. При таком назначении эффективность будет составлять

$$f(X^*) = 9 + 3 + 8 + 11 = 31.$$

Ответ: 1-го исполнителя следует назначить на 4-ю работу; 2-го – на 1-ю работу; 3-го – на 3-ю работу; 4-го – на 2-ю работу; эффективность назначения составит 31 ед. \square

4.6. Задача о рюкзаке

Математическая модель задачи о контейнерных перевозках была введена выше. Задача о рюкзаке является частным случаем задачи о контейнерных перевозках. В зависимости от того, что берется в качестве целевой функции, математическая модель задачи о рюкзаке может быть сформулирована в одном из двух вариантов.

Вариант 1. Пусть имеется n неделимых предметов с номерами i , $i = \overline{1, n}$. Вес i -того предмета равен p_i , его ценность c_i . Требуется выбрать совокупность предметов с минимальным общим весом при условии, что общая ценность груза не меньше заданной величины c .

Введем переменные x_i , $i = \overline{1, n}$, которые принимают лишь два значения: $x_i = 1$, если данный предмет укладывается в рюкзак, $x_i = 0$ – в противном случае. Тогда математическая модель задачи имеет вид:

$$Z(X) = \sum_{i=1}^n p_i x_i \rightarrow \min, \quad (4.12)$$

$$\sum_{i=1}^n c_i x_i \geq c, \quad (4.13)$$

$$x_i \in \{0; 1\}, \quad i = \overline{1, n}. \quad (4.14)$$

Вариант 2. Имеется n предметов. Предмет i , $i = \overline{1, n}$ имеет вес p_i и обладает полезностью c_i . Пусть b – общий максимальный допустимый вес предметов, которые можно положить в рюкзак. Требуется выбрать предметы таким образом, чтобы их общий вес не превышал максимально допустимый и при этом суммарная полезность (ценность) содержимого рюкзака была максимальной.

Пусть $x_i = 1$, если предмет положен в рюкзак, и $x_i = 0$ в противном случае. Математическая модель задачи имеет вид:

$$F(X) = \sum_{i=1}^n c_i x_i \rightarrow \max, \quad (4.15)$$

$$\sum_{i=1}^n p_i x_i \leq b, \quad (4.16)$$

$$x_i \in \{0, 1\}, \quad i = \overline{1, n}. \quad (4.17)$$

Процесс решения этой задачи опирается на *метод ветвей и границ*. Пусть Ω – множество планов задачи (4.12) – (4.14). Расширим его до «непрерывного» $\Omega_0 = \{X \mid X \in \Omega, 0 \leq x_i \leq 1\}$, $i = \overline{1, n}$, т.е. будем считать, что все предметы допускают произвольную делимость без потери *относительной ценности* (т.е. ценности на единицу веса). В качестве оценки множества Ω рассмотрим следующую величину:

$$\varphi(X) = \min Z(X), \quad X \in \Omega_0.$$

Очевидно, что $\varphi(X) \leq \min Z(X^*)$, $X^* \in \Omega$.

Алгоритм решения задачи о рюкзаке

Этап 1. *Определение нижней оценки начального опорного плана.*

Подсчитываем относительные веса предметов, т.е. вычисляем величины $\frac{p_i}{c_i}$ – отношение веса i -того предмета к единице ценности. Выбираем предмет с номером i_1 с наименьшим относительным весом и загружаем его (точнее «засыпаем» в раздробленном виде) в рюкзак до тех пор, пока не будет достигнута заданная ценность c или же не будет «засыпан» весь предмет. Возможны следующие случаи.

1(1). Если $c_{i_1} < c$, то $x_{i_1} = 1$. Из оставшихся предметов выбираем

предмет с номером i_2 с наименьшим относительным весом. Если $c_{i_2} < c - c_{i_1}$, то $x_{i_2} = 1$ и т.д.

1(2). Если для какого-либо номера i_k получаем

$$c_{i_k} \geq c - (c_{i_1} + c_{i_2} + \dots + c_{i_{k-1}}), \text{ то } x_{i_k} = \frac{c - \sum_{m=1}^{k-1} c_{i_m}}{c_{i_k}}. \text{ Остальные } x_{i_{k+1}} = 0.$$

В результате получаем начальный опорный план $X_0 = (x_1; x_2; \dots; x_n)$, для которого оценкой будет величина $\varphi_0(X_0) = \min Z(X_0)$.

1(3). Если $\sum_{i=1}^n c_i < c$, то план $X = \emptyset$, а $\varphi(X) = \infty$. Это значит, задача не имеет решения из-за несовместности ограничений.

Этап 2. Построение оптимального плана.

Если начальный опорный план X_0 – целочисленный, т.е. $x_i \in \{0; 1\}$, $i = \overline{1, n}$, то этот план является оптимальным. Задача решена.

Если план X_0 не является целочисленным, то разбиваем множество Ω_0 – множество всех планов задачи на два подмножества:

$$\Omega_1 = \{X \mid X \in \Omega_0, x_1 = 0\} \text{ и } \tilde{\Omega}_1 = \{X \mid X \in \Omega_0, x_1 = 1\}.$$

В качестве оценок данных множеств будут следующие величины:

$$(1.1) \text{ для } \Omega_1: \varphi_1(X_1) = \min \sum_{i=2}^n p_i x_i,$$

$$\sum_{i=2}^n c_i x_i \geq c,$$

$$0 \leq x_i \leq 1, \quad i = \overline{2, n}.$$

$$(1.2) \text{ для } \tilde{\Omega}_1: \tilde{\varphi}_1(\tilde{X}_1) = \min \left(p_1 + \sum_{i=2}^n p_i x_i \right),$$

$$\sum_{i=2}^n c_i x_i \geq c - c_1,$$

$$0 \leq x_i \leq 1, \quad i = \overline{2, n}.$$

В результате получаем два плана X_1 и \tilde{X}_1 , для которых находим $\varphi_1(X_1)$ и $\tilde{\varphi}_1(\tilde{X}_1)$. Сравниваем $\varphi_1(X_1)$ и $\tilde{\varphi}_1(\tilde{X}_1)$.

Если $\varphi_1(X_1) < \tilde{\varphi}_1(\tilde{X}_1)$, то дальнейшему разбиению подлежит множество Ω_1 , которое разбиваем на два подмножества $\Omega_2 = \{X \mid X \in \Omega_1, x_2 = 0\}$ и $\tilde{\Omega}_2 = \{X \mid X \in \Omega_1, x_2 = 1\}$.

$$(1.1.1) \text{ для } \Omega_2: \varphi_2(X_2) = \min \sum_{i=3}^n p_i x_i, \\ \sum_{i=3}^n c_i x_i \geq c, \\ 0 \leq x_i \leq 1, \quad i = \overline{3, n}.$$

$$(1.1.2) \text{ для } \tilde{\Omega}_2: \tilde{\varphi}_2(\tilde{X}_2) = \min \left(p_2 + \sum_{i=3}^n p_i x_i \right), \\ \sum_{i=3}^n c_i x_i \geq c - c_2, \\ 0 \leq x_i \leq 1, \quad i = \overline{3, n}.$$

Если $\tilde{\varphi}_1(\tilde{X}_1) < \varphi_1(X_1)$, то разбиению подлежит множество $\tilde{\Omega}_1$ на два подмножества $\Omega_2 = \{X \mid X \in \tilde{\Omega}_1, x_2 = 0\}$ и $\tilde{\Omega}_2 = \{X \mid X \in \tilde{\Omega}_1, x_2 = 1\}$.

$$(1.2.1) \text{ для } \Omega_2: \varphi_2(X_2) = \min \left(p_1 + \sum_{i=3}^n p_i x_i \right), \\ \sum_{i=3}^n c_i x_i \geq c - c_1, \\ 0 \leq x_i \leq 1, \quad i = \overline{3, n}.$$

$$(1.2.2) \text{ для } \tilde{\Omega}_2: \tilde{\varphi}_2(\tilde{X}_2) = \min \left(p_1 + p_2 + \sum_{i=3}^n p_i x_i \right), \\ \sum_{i=3}^n c_i x_i \geq c - c_1 - c_2, \\ 0 \leq x_i \leq 1, \quad i = \overline{3, n}.$$

Далее находим оценки $\varphi_2(X_2)$ и $\tilde{\varphi}_2(\tilde{X}_2)$ по правилу первого этапа. Сравниваем полученные оценки.

Множество с меньшей оценкой подлежит разбиению на два подмножества. Первый и второй этапы продолжаются, пока не получим планы задачи X_{n-1} и \tilde{X}_{n-1} .

Этап 3. *Получение оптимального плана.*

Определяем оценки опорных целочисленных планов X_{n-1} и \tilde{X}_{n-1} , т.е. $\varphi_{n-1}(X_{n-1})$ и $\tilde{\varphi}_{n-1}(\tilde{X}_{n-1})$.

Из двух оценок выбираем наименьшую. Полученную оценку сравниваем с оценками отброшенных планов. Если она окажется меньше или равной оценкам отброшенных планов, то эта оценка соответствует оптимальному плану задачи. Если она окажется больше некоторой оценки отброшенного плана, то построенный целочисленный план не является оптимальным. Далее возвращаемся к отброшенному плану с меньшей оценкой, и весь процесс решения задачи повторяется, пока не будет найден оптимальный план.

Пример 4.3. Решить задачу о рюкзаке с общей ценностью груза $c = 50$ и данными, представленными таблицей

i	1	2	3	4	5	6
c_i	12	15	10	16	8	5
p_i	4	6	10	5	4	1

Решение. Составляем математическую модель задачи:

$$Z(X) = 4x_1 + 6x_2 + 10x_3 + 5x_4 + 4x_5 + x_6 \rightarrow \min ,$$

$$12x_1 + 15x_2 + 10x_3 + 16x_4 + 8x_5 + 5x_6 \geq 50 ,$$

$$x_i \in \{ 0; 1 \}, \quad i = \overline{1, 6} .$$

Составим оценку множества Ω_0 всех планов задачи:

$$\varphi(X) = 4x_1 + 6x_2 + 10x_3 + 5x_4 + 4x_5 + x_6 \rightarrow \min ,$$

$$12x_1 + 15x_2 + 10x_3 + 16x_4 + 8x_5 + 5x_6 \geq 50 ,$$

$$0 \leq x \leq 1 \quad (i = \overline{1, 6}) .$$

Этап 1. Установим последовательность загрузки предметов, для чего подсчитаем относительные веса по формуле $\frac{p_i}{c_i}$, $i = \overline{1, 6}$. Полученные величины, а так же последовательность загрузки занесем в таблицу.

i	1	2	3	4	5	6
$\frac{p_i}{c_i}$	$\frac{1}{3} = 0,33$	$\frac{2}{5} = 0,4$	1	$\frac{5}{16} = 0,31$	$\frac{1}{2} = 0,5$	$\frac{1}{5} = 0,2$
Последовательность загрузки	III	IV	VI	II	V	I

Согласно полученным данным, первым загружается шестой предмет. Поскольку $c_6 < c$ ($5 < 50$), то полагаем $x_6 = 1$.

Вторым загружается четвертый предмет. Поскольку $c_4 < c - c_6$ ($16 < 45$), то $x_4 = 1$.

Так как $c_1 < c - (c_6 + c_4)$ ($12 < 29$), то $x_1 = 1$. Третьим загружается первый предмет.

$c_2 < c - (c_6 + c_4 + c_1)$ ($15 < 17$), то $x_2 = 1$. Четвертым загружается второй предмет.

$$c_5 \geq c - (c_6 + c_4 + c_1 + c_2) \quad (8 \geq 2), \quad \text{то} \quad x_5 = \frac{c - (c_6 + c_4 + c_1 + c_2)}{c_5} = \frac{2}{8} = \frac{1}{4}.$$

Пятым по счету загружается пятый предмет, но не целым.

Заданная ценность достигнута. Значит, $x_3 = 0$.

Получаем следующий начальный план $X_0 = \left(1; 1; 0; 1; \frac{1}{4}; 1\right)$. Оценка множества всех планов (нижняя граница)

$$\varphi_0(X_0) = \left(4 + 6 + 0 + 5 + 4 \cdot \frac{1}{4} + 1\right)_{\min} = 17.$$

Этап 2. План X_0 не является оптимальным, поскольку не является целочисленным, т.к. x_5 – дробная. Разбиваем множество планов Ω на два подмножества: $\Omega_1 = \{X \mid X \in \Omega_0, x_1 = 0\}$ и $\tilde{\Omega}_1 = \{X \mid X \in \Omega_0, x_1 = 1\}$.

$$\text{Для } \Omega_1: \quad \varphi_1(X_1) = (6x_2 + 10x_3 + 5x_4 + 4x_5 + x_6)_{\min},$$

$$15x_2 + 10x_3 + 16x_4 + 8x_5 + 5x_6 \geq 50,$$

$$0 \leq x_i \leq 1 \quad (i = \overline{2, 6}).$$

Для $\tilde{\Omega}_1$: $\varphi_1(X_1) = (4 + 6x_2 + 10x_3 + 5x_4 + 4x_5 + x_6)_{\min}$,

$$15x_2 + 10x_3 + 16x_4 + 8x_5 + 5x_6 \geq 50 - 12 = 38,$$

$$0 \leq x_i \leq 1 \quad (i = \overline{2, 6}).$$

(1.1) для Ω_1 : $x_1 = 0$.

$$c_6 < c \quad (5 < 50) \Rightarrow x_6 = 1;$$

$$c_4 < c - c_6 \quad (16 < 45) \Rightarrow x_4 = 1;$$

$$c_2 < c - (c_6 + c_4) \quad (15 < 29) \Rightarrow x_2 = 1;$$

$$c_5 < c - (c_6 + c_4 + c_2) \quad (8 < 14) \Rightarrow x_5 = 1;$$

$$c_3 \geq c - (c_6 + c_4 + c_2 + c_5) \quad (10 \geq 6) \Rightarrow x_3 = \frac{6}{10} = \frac{3}{5}.$$

Значит, опорный план задачи $X_1 = \left(0; 1; \frac{3}{5}; 1; 1; 1\right)$,

$$\varphi_1(X_1) = \left(6 + 10 \cdot \frac{3}{5} + 5 + 4 + 1\right)_{\min} = 22.$$

(1.2) для $\tilde{\Omega}_1$: $x_1 = 1$.

$$c_6 < c - c_1 \quad (5 < 38) \Rightarrow x_6 = 1;$$

$$c_4 < c - (c_1 + c_6) \quad (16 < 33) \Rightarrow x_4 = 1;$$

$$c_2 < c - (c_1 + c_6 + c_4) \quad (15 < 17) \Rightarrow x_2 = 1;$$

$$c_5 \geq c - (c_1 + c_6 + c_4 + c_2) \quad (8 \geq 2) \Rightarrow x_5 = \frac{1}{4};$$

$$x_3 = 0.$$

Значит, $\tilde{X}_1 = \left(1; 1; 0; 1; \frac{1}{4}; 1\right) = X_0$, $\tilde{\varphi}_1(\tilde{X}_1) = \varphi_0(X_0) = 17$.

Так как $\tilde{\varphi}_1(\tilde{X}_1) < \varphi_1(X_1)$ и план \tilde{X}_1 не является целочисленным, то далее разбиваем множество $\tilde{\Omega}_1$ на два подмножества

$$\Omega_2 = \{X \mid X \in \tilde{\Omega}_1, x_2 = 0\} \text{ и } \tilde{\Omega}_2 = \{X \mid X \in \tilde{\Omega}_1, x_2 = 1\}.$$

(1.2.1) для Ω_2 : $\varphi_2(X_2) = 4 + 10x_3 + 5x_4 + 4x_5 + x_6 \rightarrow \min$,

$$10x_3 + 16x_4 + 8x_5 + 5x_6 \geq 38,$$

$$0 \leq x_i \leq 1 \quad (i = \overline{3, 6}).$$

$$x_1 = 1, x_2 = 0;$$

$$c_6 < c - c_1 \quad (5 < 38) \Rightarrow x_6 = 1;$$

$$c_4 < c - (c_1 + c_6) \quad (16 < 33) \Rightarrow x_4 = 1;$$

$$c_5 < c - (c_1 + c_6 + c_4) \quad (8 < 17) \Rightarrow x_5 = 1;$$

$$c_3 \geq c - (c_1 + c_6 + c_4 + c_5) \quad (10 \geq 9) \Rightarrow x_3 = \frac{9}{10}.$$

$$\text{Значит, } X_2 = \left(1; 0; \frac{9}{10}; 1; 1; 1\right), \varphi_2(X_2) = \left(4 + 10 \cdot \frac{9}{10} + 5 + 4 + 1\right)_{\min} = 23.$$

(1.2.2) для $\tilde{\Omega}_2$: $\tilde{\varphi}_2(\tilde{X}_2) = 10 + 10x_3 + 5x_4 + 4x_5 + x_6 \rightarrow \min$,

$$10x_3 + 16x_4 + 8x_5 + 5x_6 \geq 23,$$

$$0 \leq x_i \leq 1 \quad (i = \overline{3, 6}).$$

$$x_1 = 1, x_2 = 1;$$

$$c_6 < c - (c_1 + c_2) \quad (5 < 23) \Rightarrow x_6 = 1;$$

$$c_4 < c - (c_1 + c_2 + c_6) \quad (16 < 18) \Rightarrow x_4 = 1;$$

$$c_5 < c - (c_1 + c_2 + c_6 + c_4) \quad (8 \geq 2) \Rightarrow x_5 = \frac{1}{4};$$

$$x_3 = 0.$$

$$\text{Значит, } \tilde{X}_2 = \left(1; 1; 0; 1; \frac{1}{4}; 1\right), \tilde{\varphi}_2(\tilde{X}_2) = 17.$$

Так как $\tilde{\varphi}_2(\tilde{X}_2) < \varphi_2(X_2)$ и план \tilde{X}_2 не является целочисленным, то далее разбиваем множество $\tilde{\Omega}_2$ на два подмножества

$$\Omega_3 = \{X \mid X \in \tilde{\Omega}_2, x_3 = 0\} \text{ и } \tilde{\Omega}_3 = \{X \mid X \in \tilde{\Omega}_2, x_3 = 1\}.$$

(1.2.2.1) для Ω_3 : $\varphi_3(X_3) = 10 + 5x_4 + 4x_5 + x_6 \rightarrow \min$,

$$16x_4 + 8x_5 + 5x_6 \geq 23,$$

$$0 \leq x_i \leq 1 \quad (i = \overline{4, 6}).$$

$$x_1 = 1, x_2 = 1, x_3 = 0;$$

$$c_6 < c - (c_1 + c_2) \quad (5 < 23) \Rightarrow x_6 = 1;$$

$$c_4 < c - (c_1 + c_2 + c_6) \quad (16 < 18) \Rightarrow x_4 = 1;$$

$$c_5 < c - (c_1 + c_2 + c_6 + c_4) \quad (8 \geq 2) \Rightarrow x_5 = \frac{1}{4}.$$

$$\text{Значит, } X_3 = \left(1; 1; 0; 1; \frac{1}{4}; 1 \right), \quad \varphi_3(X_3) = 17.$$

$$(1.2.2.2) \text{ для } \tilde{\Omega}_3: \tilde{\varphi}_3(\tilde{X}_3) = 20 + 5x_4 + 4x_5 + x_6 \rightarrow \min,$$

$$16x_4 + 8x_5 + 5x_6 \geq 13,$$

$$0 \leq x_i \leq 1 \quad (i = \overline{4, 6}).$$

$$x_1 = 1, \quad x_2 = 1, \quad x_3 = 1;$$

$$c_6 < c - (c_1 + c_2 + c_3) \quad (5 < 13) \Rightarrow x_6 = 1;$$

$$c_4 < c - (c_1 + c_2 + c_3 + c_6) \quad (16 \geq 8) \Rightarrow x_4 = \frac{1}{2};$$

$$x_5 = 0.$$

$$\text{Значит, } \tilde{X}_3 = \left(1; 1; 1; \frac{1}{2}; 0; 1 \right), \quad \tilde{\varphi}_3(\tilde{X}_3) = 23,5.$$

Так как $\varphi_3(X_3) < \tilde{\varphi}_3(\tilde{X}_3)$ и план X_3 не является целочисленным, то далее разбиваем множество Ω_3 на два подмножества

$$\Omega_4 = \{X \mid X \in \Omega_3, x_4 = 0\} \text{ и } \tilde{\Omega}_4 = \{X \mid X \in \Omega_3, x_3 = 1\}.$$

$$(1.2.2.1.1) \text{ для } \Omega_4: \varphi_4(X_4) = (10 + 4x_5 + x_6) \rightarrow \min,$$

$$8x_5 + 5x_6 \geq 23,$$

$$0 \leq x_i \leq 1 \quad (i = \overline{5, 6}).$$

$$x_1 = 1, \quad x_2 = 1, \quad x_3 = 0, \quad x_4 = 0;$$

$$c_6 < c - (c_1 + c_2) \quad (5 < 23) \Rightarrow x_6 = 1;$$

$$c_5 < c - (c_1 + c_2 + c_6) \quad (8 < 18).$$

Получаем, что $c_6 + c_5 < 23$, это значит, план $X_4 = \emptyset$, а $\varphi_4(X_4) = \infty$.

$$(1.2.2.1.2) \text{ для } \tilde{\Omega}_4: \tilde{\varphi}_4(\tilde{X}_4) = (15 + 4x_5 + x_6) \rightarrow \min,$$

$$8x_5 + 5x_6 \geq 7,$$

$$0 \leq x_i \leq 1 \quad (i = 5, 6).$$

$$x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 1;$$

$$c_6 < c - (c_1 + c_2 + c_4) \quad (5 < 7) \Rightarrow x_6 = 1;$$

$$c_5 < c - (c_1 + c_2 + c_4 + c_6) \quad (8 \geq 2) \Rightarrow x_5 = \frac{1}{4};$$

$$\text{Значит, } \tilde{X}_4 = \left(1; 1; 0; 1; \frac{1}{4}; 1\right), \tilde{\varphi}_4(\tilde{X}_4) = 17.$$

План \tilde{X}_4 не является целочисленным, поэтому далее разбиваем множество $\tilde{\Omega}_4$ на два подмножества $\Omega_5 = \{X \mid X \in \tilde{\Omega}_4, x_5 = 0\}$ и $\tilde{\Omega}_5 = \{X \mid X \in \tilde{\Omega}_4, x_5 = 1\}$.

$$(1.2.1.2.1) \text{ для } \Omega_5: \varphi_5(X_5) = (15 + x_6) \rightarrow \min,$$

$$5x_6 \geq 7,$$

$$0 \leq x_6 \leq 1.$$

$$x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 1, x_5 = 0.$$

Получаем, что $c_6 < 7$, это значит, план $X_5 = \emptyset$, а $\varphi_5(X_5) = \infty$.

$$(1.2.2.1.2.1) \text{ для } \tilde{\Omega}_5: \tilde{\varphi}_5(\tilde{X}_5) = (19 + x_6) \rightarrow \min,$$

$$5x_6 \geq -1,$$

$$0 \leq x_6 \leq 1.$$

$$x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 1, x_5 = 1;$$

$$c_6 < c - (c_1 + c_2 + c_3 + c_4 + c_5) \quad (5 \geq -1) \Rightarrow x_6 = 0.$$

$$\text{Значит, } \tilde{X}_5 = (1; 1; 0; 1; 1; 0), \tilde{\varphi}_5(\tilde{X}_5) = 19.$$

Этап 3. Итак, план $\tilde{X}_5 = (1; 1; 0; 1; 1; 0)$ целочисленный. Его оценка равна $\tilde{\varphi}_5(\tilde{X}_5) = 19$. Планы $X_4 = \emptyset$ и $X_5 = \emptyset$ исключаем и дальше не рассматриваем. Сравниваем оценки планов X_1, X_2, \tilde{X}_3 с оценкой плана \tilde{X}_5 . Так как

$$\tilde{\varphi}_5(\tilde{X}_5) < \varphi_1(X_1) \quad (19 < 22);$$

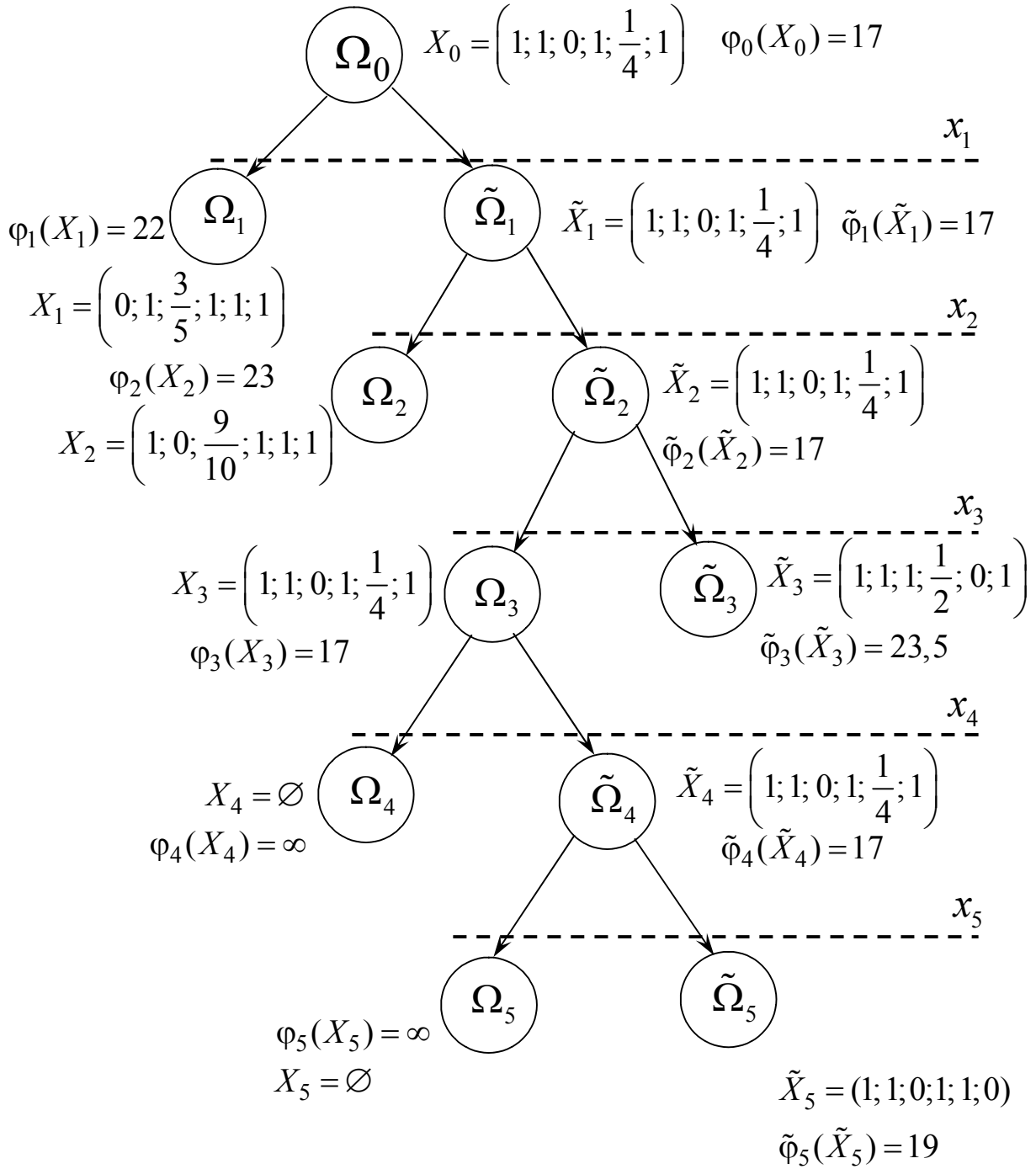
$$\tilde{\varphi}_5(\tilde{X}_5) < \varphi_2(X_2) \quad (19 < 23);$$

$$\tilde{\varphi}_5(\tilde{X}_5) < \tilde{\varphi}_3(\tilde{X}_3) \quad (19 < 23,5),$$

то план \tilde{X}_5 является оптимальным.

Значит, вес рюкзака равен 19 и в него должны быть загружены первый, второй, четвертый и пятый предметы.

Весь процесс решения задачи можно изобразить в виде дерева ветвления.



□

5. ЭЛЕМЕНТЫ ТЕОРИИ РАСПИСАНИЙ

5.1. Предмет теории расписаний и классификация задач

Развитие календарного планирования производства стало началом формирования *теории расписаний*. Поэтому в отечественной науке общую задачу теории расписаний часто называют *задачей календарного планирования*.

Возникновение теории расписаний относится к середине XX в. Но лишь с появлением ЭВМ и общим развитием методов дискретной оптимизации эта теория получила широкое развитие. Математиками в то время был сформулирован ряд задач, которые и составили основу названной науки.

Являясь разделом исследования операций, теория расписаний занимается изучением специфических моделей, имеющих место в оперативно-календарном планировании, и разработкой математических методов построения календарных планов, наилучших в смысле заданного критерия эффективности. В большинстве случаев модели, изучаемые в теории расписаний, могут быть отнесены к классу детерминированных. Это значит, что обстановка, в которой приходится выработать наилучшие решения, характеризуется полной определенностью, т.е. четко известны цель операции и все значения неконтролируемых факторов. Но в последнее время все больше внимания уделяется исследованию операций, в которых приходится принимать решения в условиях риска.

С учетом характерных особенностей задачи теории расписаний могут быть условно разделены на три группы: *задачи согласования* (задачи сетевого планирования и управления); *задачи распределения* (задачи балансирования сборочной линии, задачи о назначениях и др.); *задачи упорядочения*. В настоящее время большинство авторов относят к теории расписаний только задачи упорядочивания. Поэтому в дальнейшем мы ограничимся рассмотрением задач именно этого класса.

В качестве примера задачи упорядочения рассмотрим одну из задач оперативно-календарного планирования работы производственного участка, обеспечивающего выпуск некоторого количества деталей различных типов. Для деталей каждого типа предполагаются известными технологическая последовательность их обработки на станках и время обработки каждой детали на каждом из станков. Требуется принять решения, направ-

ленные на эффективную организацию работы участка, т.е. определить такой порядок запуска деталей в производство, при котором общее время пребывания их на обработке (участке) было бы минимальным.

Покажем, что общее время пребывания деталей на обработке зависит от порядка их запуска в производство.

Пример 5.1. Пусть на участке обработка деталей происходит на двух станках (№ 1 и № 2). По плану должны быть изготовлены три различные детали, каждая из которых обрабатывается сначала на станке № 1, затем на станке № 2. Время обработки деталей на каждом из станков задано в таблице.

Станок	Деталь		
	D_1	D_2	D_3
№ 1	3	1	2
№ 2	2	4	1

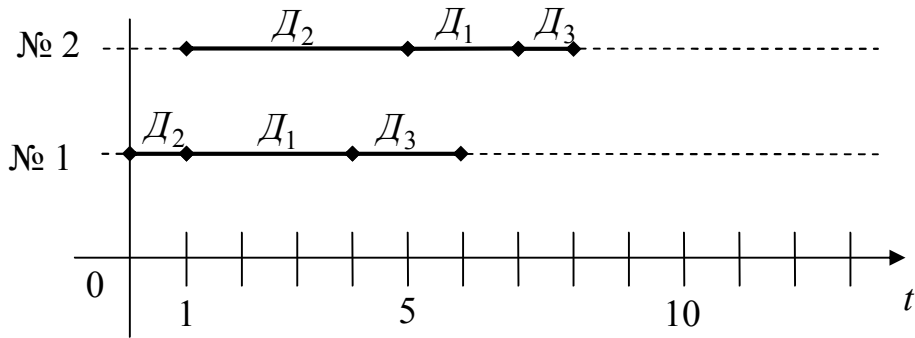
Существует шесть ($P_3 = 3! = 6$) различных способов запуска деталей в производство. Рассмотрим и сравним два из них: способ (D_2, D_1, D_3) и способ (D_3, D_1, D_2). Способ (D_2, D_1, D_3) означает следующее: сначала запускается в производство деталь D_2 , потом D_1 , а затем D_3 . График Ганта обработки деталей имеет вид, представленный на первом графике для первого способа обработки и на втором графике для второго способа обработки, рис. 5.1.

Из рисунка видно, что при первом способе запуска деталей в производство для выполнения всего плана потребуется 8 ед. времени ($T_{ок} = 8$), в то время как при втором – 11 ед. ($T_{ок} = 11$). Это значит, что первый способ организации работы является более выгодным, выигрыш во времени получается за счет сокращения простоев станка № 2 (время простоя станка в первом случае составляет $T_{пр} = 1$, а во втором – $T_{пр} = 4$).

Пример 5.2. На двух станках обрабатывают пять деталей. Время обработки деталей на каждом из станков задано в таблице

i	t_{ij}	t_{i1}	t_{i2}
1		3	2
2		4	1
3		2	3
4		3	5
5		1	4

Первый способ: (D_2, D_1, D_3) .



Второй способ: (D_3, D_1, D_2) .

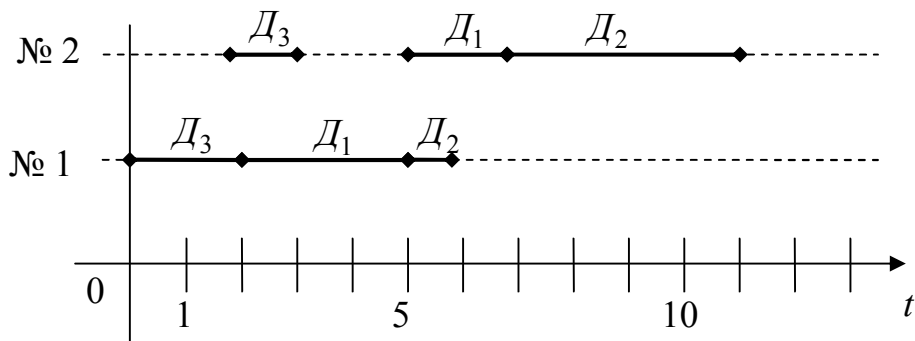


Рис. 5.1

□

Деталь обрабатывают сначала на одном станке, а затем на другом. Всего существует $P_5 = 5! = 120$ вариантов обработки.

Представим графики Ганта обработки деталей для двух последовательностей:

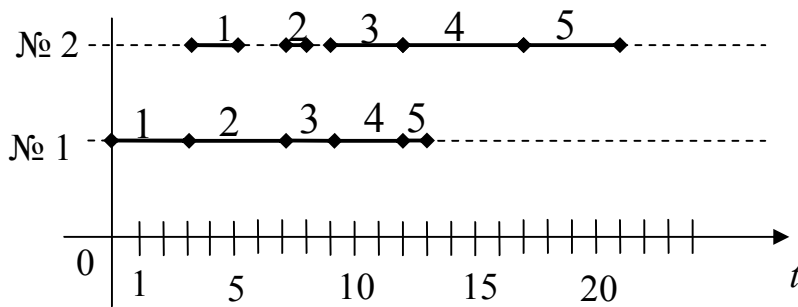
$$(1 - 2 - 3 - 4 - 5) \text{ и } (5 - 3 - 4 - 2 - 1)$$

и сравним результаты, рис. 5.2.

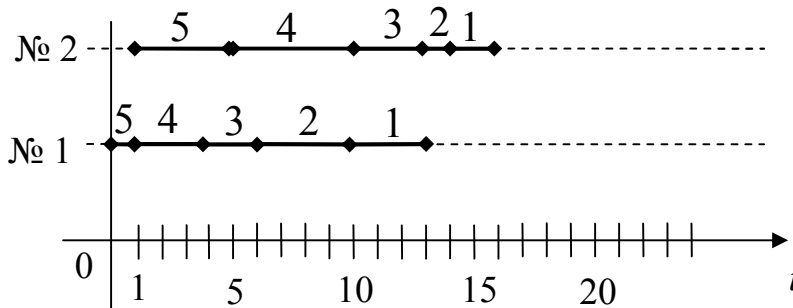
При $(1 - 2 - 3 - 4 - 5)$ $T_{np} = 6$, $T_{ок} = 21$;

при $(5 - 3 - 4 - 2 - 1)$ $T_{np} = 1$, $T_{ок} = 16$.

Первый способ: (1–2–3–4–5).



Второй способ: (5–4–3–2–1).



□

Эта же задача упорядочения может быть сформулирована следующим образом: определить порядок запуска деталей в производство, минимизирующий суммарное время выполнения плана.

В общем случае для задачи упорядочения должны быть определены:

- 1) подлежащие выполнению работы и операции (каждая работа может состоять из ряда отдельных операций);
- 2) количество и типы машин, на которых будут выполняться работы;
- 3) трудоемкость или время и порядок выполнения операций;
- 4) критерий эффективности расписания.

Задачи упорядочения различаются числом выполняемых работ и операций, из которых состоят работы, характером поступления работ в систему (одновременно или в некоторые фиксированные моменты времени), последовательностью участия машин в выполнении конкретных работ и т.д.

5.2. Формы представления расписаний

Формы представления расписаний могут быть разделены на три группы: *графические, аналитические, табличные*.

Там, где требуется наглядность, часто используются графические способы представления расписаний: графики Ганта, планировочные графики, учетные графики и т.д.).

График Ганта представляет собой хронограмму занятости устройств, механизмов, рабочих мест и т.д. Каждая операция на графике изображается масштабированным отрезком прямой, соответствующим длительности выполнения операции.

Планировочный график – это схематическое изображение процесса, состоящего из ряда работ. При построении планировочного графика каждой работе отводится отдельная строка, в которой указывается название работы, и проводится линия, соответствующая продолжительности работы в выбранном масштабе времени.

Ленточный график служит для изображения последовательности выполнения операций, составляющих одну работу. Так как операции во времени не пересекаются, на графике они изображаются последовательно с учетом времени, необходимого для их выполнения.

Развитие электронно-вычислительной техники и ее широкое применение для управления производственными процессами на различных уровнях позволяют легко получать расписания, выдаваемые в виде таблиц.

Среди аналитических способов задания расписания можно выделить способ задания с помощью *вектор-функции* времени. Так, например, для задания расписания, изображенного в примере 5.1., достаточно задать вектор-функцию

$$s(t) = (s_1(t); s_2(t)),$$

где $s_1(t)$ описывает загрузку станка № 1, а $s_2(t)$ – станка № 2.

Если $s_i(t) = 0$, $i = 1, 2$, это значит, что станок i в момент времени t простаивает. Если же $s_i(t) = k$, это значит, что в момент времени t на станке i выполняется операция k . Так, расписание представленное в примере 5.1 первым способом, может быть записано в виде вектор-функции следующим образом:

$$s(t) = \left(\begin{array}{l} s_1(t) = \begin{cases} 2 & \text{при } 0 \leq t < 1; \\ 1 & \text{при } 1 \leq t < 4; \\ 3 & \text{при } 4 \leq t < 6; \\ 0 & \text{– в остальных случаях;} \end{cases} & s_2(t) = \begin{cases} 2 & \text{при } 1 \leq t < 5; \\ 1 & \text{при } 5 \leq t < 7; \\ 3 & \text{при } 7 \leq t < 8; \\ 0 & \text{– в остальных случаях} \end{cases} \end{array} \right).$$

Для определения оптимальных решений в задачах теории расписаний широко используются различные методы исследования операций: комбинаторные, математического программирования (в особенности динамического и дискретного программирования) и т.д. Среди приближенных методов особое место занимают эвристические процедуры, которые позволяют отыскивать оптимальные или близкие к оптимальным решения, используя опыт, накопленный при исследовании моделей определенных классов.

Для построения расписаний в любой форме (табличной или графической) могут широко использоваться различные пакеты прикладных программ, в частности Ms Excel.

5.3. Задачи теории расписаний с одним обслуживающим устройством

В теории расписаний есть ряд задач, которые имеют большое практическое значение и решаются сравнительно просто. К их числу может быть отнесена задача с одним обслуживающим устройством. Дадим постановку этой задачи в терминах «машина – работа (операция)». Здесь термины «работа» и «операция» обозначают одно и то же, поскольку каждая работа состоит из одной операции. Эта терминология будет использована и в дальнейшем.

Пусть для выполнения одной машиной одновременно поступает множество $N = \{1, 2, \dots, n\}$ работ (случай, когда работы поступают не одновременно, а в некоторые фиксированные моменты времени, может быть легко сведен к рассматриваемому). Предположим также, что продолжительность выполнения каждой работы на машине известна, и обозначим ее через t_i , $i \in N$. Задача построения расписания выполнения множества работ одной машиной состоит в том, чтобы определить такой порядок выполнения работ, при котором заданный критерий эффективности будет принимать оптимальное значение.

Метод решения задачи существенно зависит от критерия эффективности. Прежде чем сформулировать различные критерии эффективности, которые используются для оценки качества расписаний, введем следующие обозначения: \underline{t}_i – время начала выполнения работы $i \in N$; \bar{t}_i – время окончания выполнения работы $i \in N$; d_i – директивное время, в течение которого должно быть завершено выполнение работы $i \in N$; α_i – штраф за ожидание работы i в единицу времени (или некоторая стоимость, связанная с выполняемой работой) до начала ее выполнения.

Время начала и окончания выполнения работ связано следующей зависимостью: $\bar{t}_i = \underline{t}_i + t_i$. Здесь и в дальнейшем предполагается, что перерывы в выполнении работ не допускаются. Это значит, что если машина приступила к выполнению некоторой работы, то она продолжает выполнять ее до тех пор, пока не закончит. Время T , необходимое для выполнения всех работ множества N , не зависит от порядка выполнения работ и равно сумме выполнения всех работ:

$$T = \sum_{i \in N} t_i. \quad (5.1)$$

Учитывая принятые обозначения, можно записать, что задержка z_i (превышение директивного срока пребывания в системе) работы i состоит

$$z_i = \max(0; \bar{t}_i - d_i), \quad i \in N. \quad (5.2)$$

Таким образом, можно ввести следующие критерии эффективности, которые, как правило, требуется обратить в минимум:

1) определяющий величину суммарных издержек, связанных с опозданием выполнения работ к заданным срокам, будет иметь следующий вид:

$$Z_1 = \sum_{i \in N} \alpha_i z_i \rightarrow \min; \quad (5.3)$$

2) критерий, позволяющий вычислить максимальный штраф, связанный также с опозданием выполнения работ, имеет вид

$$Z_2 = \max_{i \in N} \alpha_i z_i. \quad (5.4)$$

Этот критерий приводит к такому расписанию, в котором издержки, связанные с опозданием выполнения работ, распределены наиболее равномерно.

Часто встречается критерий

$$Z_3 = \sum_{i=1}^n \alpha_i \underline{t}_i \rightarrow \min, \quad (5.5)$$

представляющий собой сумму штрафов, связанных с ожиданием работ в системе.

Рассмотрим задачу построения оптимального расписания по критерию (5.5). Для вывода правила воспользуемся следующими рассуждениями: пусть имеется оптимальное расписание, в котором порядок выполнения работ совпадает с их номерами. Тогда этому расписанию будет соответствовать минимальное значение критерия, т.е. $Z = \sum_{i=1}^n \alpha_i t_i \rightarrow \min$. Вся-

кая перестановка в порядке выполнения двух или более работ уже не может привести к уменьшению критерия эффективности, поэтому переставив метами работы с номерами k и $k+1$, можно записать, что $Z_{\min} \leq Z'$, где Z_{\min} – соответственно минимальное значение критерия эффективности, Z' – критерий эффективности, полученный после перестановки. Тогда с учетом критерия (5.5) данное неравенство может быть записано в виде

$$\sum_{i=1}^{k-1} \alpha_i t_i + \alpha_k t_k + \alpha_{k+1} t_{k+1} + \sum_{i=k+2}^n \alpha_i t_i \leq \sum_{i=1}^{k-1} \alpha_i t_i + \alpha_{k+1} t'_{k+1} + \alpha_k t'_k + \sum_{i=k+2}^n \alpha_i t_i,$$

где t'_{k+1} и t'_k – время начала выполнения соответственно $(k+1)$ -й и k -тый работ после перестановки. Заметим, что как в первом (оптимальном), так и во втором (с перестановкой) расписании k -тая и соответственно $(k+1)$ -я работы начнут выполняться после того, как будут выполнены предшествующие $k-1$ работы, поэтому $t_k = t'_{k+1}$, а $t_{k+1} = t_k + t_k$, $t'_k = t'_{k+1} + t_{k+1} = t_k + t_{k+1}$. В результате можно записать:

$$\alpha_k t_k + \alpha_{k+1} (t_k + t_k) \leq \alpha_{k+1} t_k + \alpha_k (t_k + t_{k+1}).$$

Выполнив простейшие преобразования, получим $\alpha_{k+1} t_k \leq \alpha_k t_{k+1}$, откуда

$$\frac{t_k}{\alpha_k} \leq \frac{t_{k+1}}{\alpha_{k+1}}.$$

На основании последнего соотношения может быть сформулировано следующее правило построения расписания, оптимального по критерию (5.5):

- 1) для всех работ вычислить отношение $\frac{t_i}{\alpha_i}$;
- 2) упорядочить работы по возрастанию этого отношения.

Пример 5.3. Определить оптимальный порядок обработки деталей, минимизирующий сумму штрафов, связанных с пролеживанием

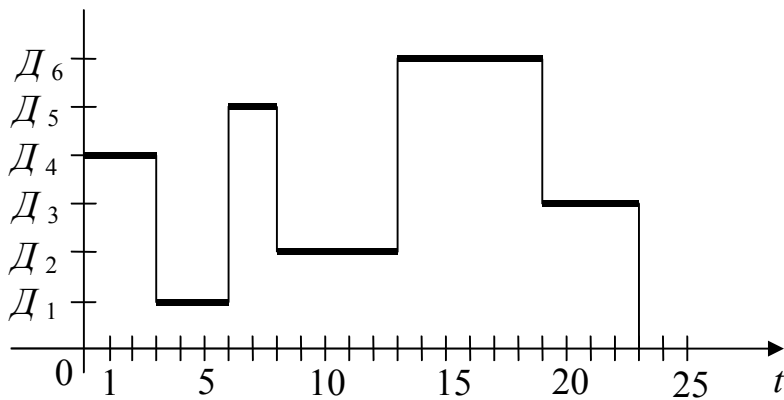
деталей в ожидании обслуживания, для исходных данных, приведенных в таблице. Построить оптимальное расписание в виде кусочно-непрерывной функции.

Характеристики	Номер детали					
	D_1	D_2	D_3	D_4	D_5	D_6
t_i	3	5	4	3	2	6
α_i	2	2	1	3	1	2

Решение. Для построения оптимального расписания в соответствии с правилом вычислим отношение $\frac{t_i}{\alpha_i}$ и пронумеруем детали в порядке возрастания этого отношения. Результаты вычислений представлены в таблице.

Параметры	Номер детали					
	D_1	D_2	D_3	D_4	D_5	D_6
t_i/α_i	1,5	2,5	4	1	2	3
Оптимальный порядок	2	4	6	1	3	5

Расписание в виде кусочно-непрерывной функции показан на рисунке.



В данном случае критерий эффективности

$$Z_{\min} = \sum_{i=1}^6 \alpha_i t_i = 3 \cdot 0 + 2 \cdot 3 + 1 \cdot 6 + 2 \cdot 8 + 2 \cdot 13 + 1 \cdot 19 = 73.$$

□

5.4. Задачи теории расписаний с двумя последовательными обслуживающими устройствами

Одной из наиболее простых в смысле постановки и наиболее трудных в отношении получения решения в общем виде задач упорядочения

является *задача Джонсона*. Прежде чем сформулировать задачу для случая двух обслуживающих устройств, дадим ее формулировку в общем виде.

Имеется множество $N = \{1, 2, \dots, n\}$ работ, которые должны быть выполнены на m машинах. Время выполнения работы i на машине j обозначим через t_{ij} , $i = \overline{1, n}$, $j = \overline{1, m}$, предполагая его заранее известным. Порядок выполнения операций, составляющих работу, может быть как одним и тем же, так и различным для разных работ. Задача построения расписания состоит в указании порядка, в котором должны выполняться работы, чтобы суммарное время простоя всех машин было минимальным.

При построении любого расписания, в том числе и оптимального, должны учитываться следующие условия:

- 1) в любой момент времени на машине не может быть выполняться более одной работы;
- 2) одна работа в фиксированный момент времени может выполняться только одной машиной.

В общем случае нет правила, позволяющего построить сразу же оптимальное расписание, поэтому сначала рассмотрим случай, когда число машин равно двум (M_1 и M_2). Каждая работа состоит из двух операций, которые выполняются сначала на первой машине, затем – на второй. Время выполнения работы i на первой, а затем на второй машине равно t_{i1} и t_{i2} соответственно.

Прежде чем перейти к построению оптимального расписания для двух машин, надо решить вопрос о порядке выполнения операций на первой и второй машинах. Должен ли он в оптимальном расписании совпадать для первой и второй машин? Здесь может быть доказано следующее утверждение: порядок выполнения операций на любой машине может быть изменен так, что он будет совпадать для первой и второй машин, при этом время выполнения всех работ не изменится. Считая, что порядок выполнения операций на первой и второй машинах один и тот же, приведем алгоритм построения оптимального расписания, который называется *алгоритмом Джонсона*.

Алгоритм Джонсона

Предварительный этап. Записываем матрицу времени выполнения операций:

$$T = [t_{ij}], \quad i = \overline{1, n}, \quad j = 1, 2.$$

Этап 1. Выбираем в матрице $T = [t_{ij}]$ минимальный элемент. Если он находится в первой строке (соответствующей первой машине), то данную работу выполняем первой, если во второй строке – последней.

Этап 2. Исключаем из рассмотрения время выполнения операций, относящееся к упорядоченной работе. Если множество элементов матрицы $T = [t_{ij}]$ уже пусто, то задача решена, если нет – переходим к первому этапу.

Таким образом, для построения оптимального расписания, минимизирующего суммарное время простоя первой и второй машин, первый и второй этапы должны быть повторены n раз. Если же случится, что $t_{i1} = t_{i2}$, то эта работа может быть упорядочена как по t_{i1} так и по t_{i2} .

Пример 5.4. Имеется пять работ, состоящих каждая из двух операций, которые выполняются сначала на первой машине, затем на второй. Время выполнения операций задано в таблице.

M	P				
	P_1	P_2	P_3	P_4	P_5
M_1	5	3	4	1	2
M_2	2	1	3	2	3

Определить оптимальный порядок выполнения работ, чтобы минимизировать время простоя машин.

Решение.

1) Выбираем минимальный элемент матрицы. В нашем случае – это 1, которая есть как в первой, так и во второй строках. В первой строке элемент 1 соответствует P_4 , значит, эта работа будет выполняться первой. Во второй строке элемент 1 соответствует P_2 , значит, эта работа будет выполняться последней.

2) Исключаем из рассмотрения работы P_2 и P_4 .

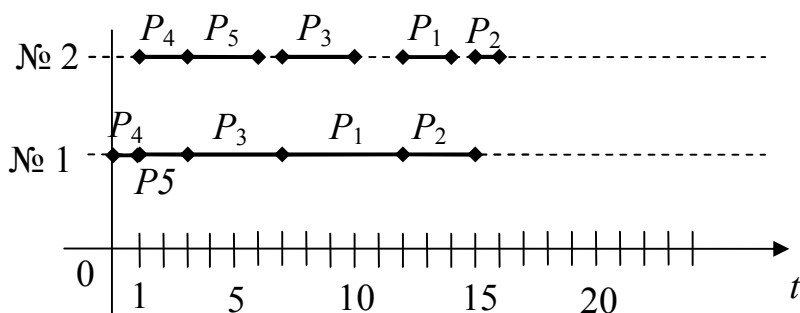
В первой строке минимальным будет теперь элемент 2, что соответствует работе P_5 , которая будет выполняться после работы P_4 .

Во второй строке из оставшихся элементов минимальным будет элемент 2, который соответствует работе P_1 . Поэтому эта работа будет выполняться предпоследней, т.е. перед работой P_2 .

3) Работа P_3 будет выполняться после работы P_5 .

Таким образом, получаем следующий оптимальный порядок выполнения работ:

$$(P_4; P_5; P_3; P_1; P_2).$$



$$T_{ок} = 16, \quad \min T_{np} = 1 + 1 + 2 + 1 = 5.$$

□

ВОПРОСЫ САМОКОНТРОЛЯ

1. Что представляет собой модель задачи математического программирования?
2. Какой план называется допустимым? Какой допустимый план называется оптимальным?
3. На какие типы делятся задачи математического программирования в зависимости от целевой функции и системы ограничений?
4. Какой раздел математического программирования называется линейным программированием?
5. Сформулировать задачу о наилучшем использовании ресурсов и записать ее математическую модель
6. Сформулировать задачу о раскрое материала и записать ее математическую модель.
7. Перечислить формы записи ЗЛП. Записать математическую модель для каждой формы записи ЗЛП.
8. Какой вид математической модели ЗЛП для двух переменных?
9. Дать геометрическую интерпретацию системе ограничений математической модели ЗЛП для двух переменных. Дать геометрическую интерпретацию целевой функции математической модели ЗЛП для двух переменных.
10. Что показывают частные производные целевой функции?
11. Сформулировать правила решения ЗЛП графическим методом.
12. Какие возможны случаи построения оптимального плана в результате решения ЗЛП?
13. Сформулировать транспортную задачу (ТЗ). Какой вид имеет математическая модель ТЗ? Что представляет собой матричная модель ТЗ?

14. Сформулировать критерий существования допустимого плана ТЗ.
15. Какая модель ТЗ называется открытой? Какие преобразования выполняются для перехода от открытой модели ТЗ к закрытой?
16. Раскрыть суть построения исходного опорного плана.
17. Раскрыть суть метода потенциалов. Сформулировать теорему о потенциалах.
18. Какие преобразования проводятся в случае невыполнения условия оптимальности?
19. Изложить алгоритм решения ТЗ методом потенциалов.
20. Какой раздел математического программирования называется дискретным?
21. Сформулировать задачу о контейнерных перевозках и записать ее математическую модель.
22. Сформулировать задачу о назначении и записать ее математическую модель.
23. В чем заключается суть методов дискретной оптимизации?
24. Изложить идею метода ветвей и границ.
25. Изложить алгоритм метода ветвей и границ для решения задач целочисленного программирования.
26. Изложить алгоритм венгерского метода для решения задачи о назначении.
27. Какой вид математической модели имеет задача о рюкзаке в зависимости от того, что берется в качестве целевой функции?
28. Какой метод используется для решения задачи о рюкзаке?
30. Что должно быть определено в общем случае для задачи упорядочения?
31. Перечислить формы представления расписаний.
32. Каким условиям должно удовлетворять время, необходимое для выполнения одной работы и для выполнения всех работ?
33. Сформулировать правило построения расписания с одним обслуживающим устройством по критерию минимизации суммы штрафов, связанных с ожиданием работ в системе.
34. Сформулировать в общем виде задачу построения расписания.
35. Какие условия должны учитываться при построении любого расписания?
36. Изложить алгоритм Джонсона для построения расписания с двумя последовательными обслуживающими устройствами.

ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

Задание 1 (1)

Для производства двух видов изделий A_1 и A_2 используется токарное (Т), фрезерное (Ф) и шлифовальное (Ш) оборудование. Нормы затрат времени на изготовление одного изделия вида A_1 каждого из типов оборудования следующие: Т – t_1 ст.-ч; Ф – h_1 ст.-ч; Ш – w_1 ст.-ч. Нормы затрат времени на изготовление одного изделия вида A_2 каждого из типов оборудования следующие: Т – t_2 ст.-ч; Ф – h_2 ст.-ч; Ш – w_2 ст.-ч. Общий фонд рабочего времени каждого оборудования: токарного – t ст.-ч; фрезерного – h ; шлифовального – w . Прибыль от реализации одного изделия каждого вида следующая: для A_1 – a_1 ден. ед.; для A_2 – a_2 ден. ед.

Найти план выпуска изделий A_1 и A_2 , обеспечивающий максимальную прибыль от их реализации.

Вариант Показатель	1	3	5	7	9	11	13	15	17	19	21	23
t_1	6	6	5	2	8	6	6	5	2	8	6	6
t_2	3	4	11	1	1	3	4	11	1	1	3	6
t	120	372	550	162	320	120	372	550	162	320	120	378
h_1	9	3	6	2	3	9	3	6	2	3	9	3
h_2	9	4	8	4	1	9	4	8	4	1	9	4
h	225	240	504	240	150	225	240	504	240	150	315	240
w_1	1	1	15	1	1	1	2	15	1	1	1	2
w_2	6	4	5	7	2	6	4	5	7	2	6	4
w	90	160	945	350	200	96	160	940	350	200	90	160
a_1	50	100	60	40	150	250	300	450	300	80	90	100
a_2	100	80	30	120	200	100	100	300	250	160	200	200

Задание 1 (2)

Для изготовления изделий двух моделей M_1 и M_2 на фабрике используется два вида сырья и рабочие ресурсы. На изготовление одного изделия модели M_1 затрачивается рабочего времени t_1 чел.-ч; сырья 1 с – m_{11} м²; сырья 2 с – m_{21} м². На изготовление одного изделия модели M_2 затрачивается рабочего времени t_2 чел.-ч; сырья 1 с – m_{12} м²; сырья 2 с – m_{22} м². Общий фонд производственных факторов таков: рабочего времени – t чел.-ч; сырья 1 с – m' м²; сырья 2 с – m'' м². Прибыль от реализации единицы изделия модели M_1 равна a_1 ден. ед., единицы изделия модели M_2 – a_2 ден. ед. Найти план выпуска продукции по ассортименту, который максимизирует прибыль.

Вариант Показатель	2	4	6	8	10	12	14	16	18	20	22	24
t_1	1	2	3	3	4	1	2	3	3	4	1	2
t_2	2	3	2	4	3	2	3	2	4	3	2	3
t	1000	1200	1800	1200	900	1000	1200	1800	1200	900	1400	1200
m_{11}	2	2	1	3	4	2	2	1	3	2	2	0
m_{21}	1	1	2	2	2	1	1	2	2	2	1	4
m'	800	600	1400	900	800	800	600	1400	600	800	800	600
m_{12}	2	1	3	0	0	2	1	0	1	6	2	1
m_{22}	0	0	0	2	3	4	5	3	2	3	0	3
m''	600	250	1200	540	600	600	250	1200	64	600	600	240
a_1	40	50	40	70	50	60	80	90	120	140	90	110
a_2	60	80	60	120	100	30	50	60	80	100	30	90

Задание 2

В экономическом районе имеются четыре предприятия, выпускающие однородную продукцию, объемы которой соответственно равны a_1, a_2, a_3 , и удовлетворяющие потребности пяти потребителей в объемах,

которые соответственно равны b_1, b_2, b_3, b_4 . Стоимости перевозки единицы продукции от предприятия к потребителю заданы матрицей $C = (c_{ij})$. Найти план перевозки продукции, который минимизирует суммарные расходы на доставку продукции.

1)

$$\begin{aligned} a_1 &= 20; & b_1 &= 60; \\ a_2 &= 60; & b_2 &= 30; \\ a_3 &= 30; & b_3 &= 40; \\ & & b_4 &= 10; \end{aligned}$$

$$C = (c_{ij}) = \begin{pmatrix} 50 & 40 & 35 & 60 \\ 80 & 30 & 20 & 10 \\ 75 & 10 & 15 & 10 \end{pmatrix}.$$

2)

$$\begin{aligned} a_1 &= 150; & b_1 &= 200; \\ a_2 &= 200; & b_2 &= 140; \\ a_3 &= 300; & b_3 &= 160; \\ & & b_4 &= 80; \end{aligned}$$

$$C = (c_{ij}) = \begin{pmatrix} 15 & 17 & 30 & 20 \\ 18 & 20 & 25 & 27 \\ 32 & 18 & 14 & 14 \end{pmatrix}.$$

3)

$$\begin{aligned} a_1 &= 50; & b_1 &= 30; \\ a_2 &= 90; & b_2 &= 50; \\ a_3 &= 40; & b_3 &= 70; \\ & & b_4 &= 40; \end{aligned}$$

$$C = (c_{ij}) = \begin{pmatrix} 80 & 90 & 30 & 50 \\ 30 & 60 & 10 & 15 \\ 85 & 10 & 25 & 35 \end{pmatrix}.$$

4)

$$\begin{aligned} a_1 &= 200; & b_1 &= 140; \\ a_2 &= 150; & b_2 &= 100; \\ a_3 &= 120; & b_3 &= 90; \\ & & b_4 &= 200; \end{aligned}$$

$$C = (c_{ij}) = \begin{pmatrix} 14 & 17 & 28 & 20 \\ 10 & 20 & 15 & 17 \\ 15 & 18 & 14 & 25 \end{pmatrix}.$$

5)

$$\begin{aligned} a_1 &= 70; & b_1 &= 40; \\ a_2 &= 80; & b_2 &= 60; \\ a_3 &= 80; & b_3 &= 40; \\ & & b_4 &= 60; \end{aligned}$$

$$C = (c_{ij}) = \begin{pmatrix} 40 & 10 & 25 & 30 \\ 15 & 25 & 40 & 10 \\ 10 & 60 & 15 & 20 \end{pmatrix}.$$

6)

$$\begin{aligned} a_1 &= 150; & b_1 &= 200; \\ a_2 &= 200; & b_2 &= 250; \\ a_3 &= 170; & b_3 &= 100; \\ & & b_4 &= 60; \end{aligned}$$

$$C = (c_{ij}) = \begin{pmatrix} 33 & 12 & 32 & 20 \\ 18 & 27 & 25 & 12 \\ 12 & 18 & 25 & 24 \end{pmatrix}.$$

7)

$$a_1 = 80; \quad b_1 = 50;$$

$$a_2 = 50; \quad b_2 = 60;$$

$$a_3 = 40; \quad b_3 = 40;$$

$$b_4 = 40;$$

$$C = (c_{ij}) = \begin{pmatrix} 40 & 80 & 65 & 10 \\ 20 & 10 & 20 & 35 \\ 45 & 50 & 55 & 20 \end{pmatrix}.$$

8)

$$a_1 = 90; \quad b_1 = 200;$$

$$a_2 = 200; \quad b_2 = 80;$$

$$a_3 = 450; \quad b_3 = 100;$$

$$b_4 = 180;$$

$$C = (c_{ij}) = \begin{pmatrix} 10 & 17 & 50 & 20 \\ 15 & 20 & 15 & 26 \\ 22 & 18 & 10 & 34 \end{pmatrix}.$$

9)

$$a_1 = 80; \quad b_1 = 40;$$

$$a_2 = 90; \quad b_2 = 70;$$

$$a_3 = 60; \quad b_3 = 80;$$

$$b_4 = 30;$$

$$C = (c_{ij}) = \begin{pmatrix} 10 & 25 & 45 & 30 \\ 20 & 30 & 10 & 40 \\ 65 & 15 & 15 & 20 \end{pmatrix}.$$

10)

$$a_1 = 350; \quad b_1 = 200;$$

$$a_2 = 100; \quad b_2 = 160;$$

$$a_3 = 200; \quad b_3 = 260;$$

$$b_4 = 80;$$

$$C = (c_{ij}) = \begin{pmatrix} 25 & 27 & 32 & 20 \\ 10 & 20 & 26 & 21 \\ 12 & 16 & 16 & 24 \end{pmatrix}.$$

11)

$$a_1 = 20; \quad b_1 = 60;$$

$$a_2 = 70; \quad b_2 = 30;$$

$$a_3 = 40; \quad b_3 = 20;$$

$$b_4 = 50;$$

$$C = (c_{ij}) = \begin{pmatrix} 35 & 40 & 30 & 40 \\ 70 & 25 & 10 & 10 \\ 45 & 15 & 35 & 20 \end{pmatrix}.$$

12)

$$a_1 = 250; \quad b_1 = 120;$$

$$a_2 = 300; \quad b_2 = 240;$$

$$a_3 = 500; \quad b_3 = 60;$$

$$b_4 = 180;$$

$$C = (c_{ij}) = \begin{pmatrix} 45 & 17 & 32 & 22 \\ 24 & 10 & 22 & 20 \\ 12 & 14 & 34 & 36 \end{pmatrix}.$$

13)

$$a_1 = 50; \quad b_1 = 20;$$

$$a_2 = 80; \quad b_2 = 50;$$

$$a_3 = 60; \quad b_3 = 50;$$

$$b_4 = 40;$$

$$C = (c_{ij}) = \begin{pmatrix} 80 & 50 & 10 & 50 \\ 10 & 60 & 25 & 15 \\ 75 & 45 & 20 & 35 \end{pmatrix}.$$

14)

$$a_1 = 320; \quad b_1 = 140;$$

$$a_2 = 150; \quad b_2 = 150;$$

$$a_3 = 260; \quad b_3 = 90;$$

$$b_4 = 100;$$

$$C = (c_{ij}) = \begin{pmatrix} 14 & 17 & 28 & 20 \\ 10 & 20 & 15 & 17 \\ 15 & 18 & 14 & 25 \end{pmatrix}.$$

15)

$$\begin{aligned} a_1 &= 60; & b_1 &= 40; \\ a_2 &= 70; & b_2 &= 60; \\ a_3 &= 50; & b_3 &= 20; \\ & & b_4 &= 50; \end{aligned}$$

$$C = (c_{ij}) = \begin{pmatrix} 30 & 20 & 25 & 30 \\ 25 & 20 & 45 & 15 \\ 10 & 10 & 30 & 25 \end{pmatrix}.$$

16)

$$\begin{aligned} a_1 &= 150; & b_1 &= 200; \\ a_2 &= 200; & b_2 &= 250; \\ a_3 &= 170; & b_3 &= 100; \\ & & b_4 &= 60; \end{aligned}$$

$$C = (c_{ij}) = \begin{pmatrix} 33 & 12 & 32 & 20 \\ 18 & 27 & 25 & 12 \\ 12 & 18 & 25 & 24 \end{pmatrix}.$$

17)

$$\begin{aligned} a_1 &= 80; & b_1 &= 90; \\ a_2 &= 70; & b_2 &= 30; \\ a_3 &= 20; & b_3 &= 50; \\ & & b_4 &= 60; \end{aligned}$$

$$C = (c_{ij}) = \begin{pmatrix} 20 & 10 & 45 & 20 \\ 45 & 15 & 10 & 35 \\ 40 & 50 & 55 & 25 \end{pmatrix}.$$

18)

$$\begin{aligned} a_1 &= 190; & b_1 &= 200; \\ a_2 &= 350; & b_2 &= 480; \\ a_3 &= 400; & b_3 &= 120; \\ & & b_4 &= 70; \end{aligned}$$

$$C = (c_{ij}) = \begin{pmatrix} 14 & 10 & 50 & 30 \\ 23 & 20 & 18 & 26 \\ 20 & 18 & 10 & 24 \end{pmatrix}.$$

19)

$$\begin{aligned} a_1 &= 80; & b_1 &= 40; \\ a_2 &= 70; & b_2 &= 70; \\ a_3 &= 50; & b_3 &= 90; \\ & & b_4 &= 50; \end{aligned}$$

$$C = (c_{ij}) = \begin{pmatrix} 35 & 25 & 45 & 50 \\ 10 & 30 & 15 & 40 \\ 55 & 40 & 15 & 10 \end{pmatrix}.$$

20)

$$\begin{aligned} a_1 &= 350; & b_1 &= 90; \\ a_2 &= 500; & b_2 &= 160; \\ a_3 &= 160; & b_3 &= 360; \\ & & b_4 &= 480; \end{aligned}$$

$$C = (c_{ij}) = \begin{pmatrix} 22 & 27 & 42 & 20 \\ 18 & 10 & 29 & 21 \\ 32 & 36 & 10 & 44 \end{pmatrix}.$$

21)

$$\begin{aligned} a_1 &= 60; & b_1 &= 70; \\ a_2 &= 70; & b_2 &= 70; \\ a_3 &= 50; & b_3 &= 30; \\ & & b_4 &= 50; \end{aligned}$$

$$C = (c_{ij}) = \begin{pmatrix} 30 & 10 & 20 & 35 \\ 20 & 20 & 55 & 15 \\ 10 & 15 & 35 & 25 \end{pmatrix}.$$

22)

$$\begin{aligned} a_1 &= 150; & b_1 &= 200; \\ a_2 &= 200; & b_2 &= 170; \\ a_3 &= 370; & b_3 &= 430; \\ & & b_4 &= 90; \end{aligned}$$

$$C = (c_{ij}) = \begin{pmatrix} 23 & 12 & 34 & 20 \\ 28 & 10 & 25 & 12 \\ 16 & 35 & 10 & 24 \end{pmatrix}.$$

23)

$$a_1 = 60; \quad b_1 = 90;$$

$$a_2 = 90; \quad b_2 = 20;$$

$$a_3 = 30; \quad b_3 = 50;$$

$$b_4 = 40;$$

$$C = (c_{ij}) = \begin{pmatrix} 25 & 40 & 45 & 20 \\ 10 & 15 & 20 & 35 \\ 40 & 50 & 55 & 10 \end{pmatrix}.$$

24)

$$a_1 = 190; \quad b_1 = 360;$$

$$a_2 = 370; \quad b_2 = 280;$$

$$a_3 = 430; \quad b_3 = 120;$$

$$b_4 = 170;$$

$$C = (c_{ij}) = \begin{pmatrix} 44 & 30 & 52 & 32 \\ 23 & 20 & 18 & 36 \\ 10 & 16 & 23 & 24 \end{pmatrix}.$$

Задание 3

Имеется 5 человек, которые могут выполнять 5 различных работ. Известна производительность i -того исполнителя при выполнении j -той работы, которая задана матрицей $A = (a_{ij})$. С помощью алгоритма венгерского метода определить, кого и на какую работу следует назначить, чтобы добиться максимальной суммарной производительности при условии, что каждый исполнитель может быть назначен только на одну работу.

$$1) \quad A = (a_{ij}) = \begin{pmatrix} 10 & 8 & 5 & 9 & 16 \\ 4 & 3 & 4 & 11 & 12 \\ 5 & 10 & 29 & 7 & 6 \\ 9 & 2 & 4 & 1 & 3 \\ 7 & 10 & 5 & 2 & 6 \end{pmatrix}$$

$$2) \quad A = (a_{ij}) = \begin{pmatrix} 10 & 7 & 2 & 4 & 5 \\ 8 & 4 & 3 & 7 & 3 \\ 2 & 4 & 10 & 11 & 8 \\ 8 & 12 & 9 & 7 & 6 \\ 9 & 10 & 7 & 5 & 4 \end{pmatrix}$$

$$3) \quad A = (a_{ij}) = \begin{pmatrix} 3 & 9 & 11 & 8 & 11 \\ 6 & 2 & 3 & 6 & 5 \\ 10 & 4 & 13 & 8 & 10 \\ 3 & 8 & 7 & 5 & 7 \\ 2 & 9 & 6 & 3 & 12 \end{pmatrix}$$

$$4) \quad A = (a_{ij}) = \begin{pmatrix} 10 & 10 & 5 & 1 & 5 \\ 4 & 10 & 2 & 10 & 2 \\ 10 & 8 & 10 & 5 & 2 \\ 4 & 2 & 10 & 10 & 8 \\ 5 & 7 & 9 & 8 & 5 \end{pmatrix}$$

$$5) \quad A = (a_{ij}) = \begin{pmatrix} 6 & 2 & 10 & 4 & 10 \\ 2 & 1 & 1 & 9 & 10 \\ 5 & 2 & 10 & 10 & 7 \\ 10 & 5 & 10 & 4 & 8 \\ 6 & 9 & 4 & 7 & 11 \end{pmatrix}$$

$$6) \quad A = (a_{ij}) = \begin{pmatrix} 8 & 10 & 8 & 10 & 4 \\ 3 & 10 & 1 & 6 & 1 \\ 10 & 9 & 10 & 8 & 3 \\ 1 & 12 & 10 & 1 & 10 \\ 5 & 10 & 15 & 5 & 9 \end{pmatrix}$$

$$7) \quad A = (a_{ij}) = \begin{pmatrix} 3 & 4 & 10 & 10 & 3 \\ 9 & 10 & 7 & 7 & 10 \\ 9 & 8 & 10 & 8 & 10 \\ 2 & 10 & 8 & 10 & 7 \\ 3 & 9 & 7 & 5 & 12 \end{pmatrix}$$

$$8) \quad A = (a_{ij}) = \begin{pmatrix} 10 & 1 & 3 & 10 & 2 \\ 10 & 8 & 3 & 10 & 2 \\ 10 & 2 & 16 & 10 & 8 \\ 7 & 10 & 10 & 5 & 3 \\ 5 & 9 & 8 & 7 & 5 \end{pmatrix}$$

$$9) \quad A = (a_{ij}) = \begin{pmatrix} 10 & 6 & 10 & 7 & 3 \\ 2 & 10 & 1 & 9 & 10 \\ 2 & 3 & 15 & 10 & 10 \\ 10 & 5 & 10 & 6 & 5 \\ 5 & 7 & 3 & 5 & 8 \end{pmatrix}$$

$$10) \quad A = (a_{ij}) = \begin{pmatrix} 3 & 9 & 10 & 10 & 2 \\ 7 & 10 & 1 & 3 & 3 \\ 9 & 10 & 10 & 1 & 3 \\ 10 & 6 & 1 & 10 & 6 \\ 12 & 5 & 4 & 6 & 8 \end{pmatrix}$$

$$11) \quad A = (a_{ij}) = \begin{pmatrix} 3 & 4 & 10 & 2 & 10 \\ 3 & 6 & 4 & 10 & 10 \\ 10 & 7 & 8 & 10 & 4 \\ 10 & 4 & 1 & 2 & 10 \\ 5 & 8 & 5 & 4 & 7 \end{pmatrix}$$

$$12) \quad A = (a_{ij}) = \begin{pmatrix} 10 & 5 & 10 & 9 & 6 \\ 1 & 10 & 2 & 2 & 10 \\ 10 & 5 & 12 & 10 & 1 \\ 10 & 9 & 3 & 3 & 10 \\ 5 & 7 & 5 & 4 & 5 \end{pmatrix}$$

$$13) \quad A = (a_{ij}) = \begin{pmatrix} 10 & 10 & 1 & 2 & 9 \\ 10 & 6 & 1 & 5 & 3 \\ 10 & 8 & 14 & 5 & 10 \\ 9 & 10 & 2 & 6 & 10 \\ 5 & 5 & 7 & 10 & 12 \end{pmatrix}$$

$$14) \quad A = (a_{ij}) = \begin{pmatrix} 10 & 4 & 8 & 6 & 10 \\ 10 & 2 & 3 & 10 & 3 \\ 7 & 10 & 10 & 2 & 3 \\ 10 & 3 & 2 & 10 & 1 \\ 5 & 4 & 6 & 5 & 7 \end{pmatrix}$$

$$15) \quad A = (a_{ij}) = \begin{pmatrix} 4 & 6 & 8 & 10 & 2 \\ 3 & 4 & 3 & 9 & 5 \\ 4 & 6 & 13 & 2 & 1 \\ 2 & 3 & 4 & 10 & 9 \\ 5 & 7 & 9 & 5 & 8 \end{pmatrix}$$

$$16) \quad A = (a_{ij}) = \begin{pmatrix} 8 & 5 & 8 & 2 & 7 \\ 9 & 2 & 3 & 10 & 9 \\ 16 & 12 & 12 & 8 & 14 \\ 10 & 6 & 3 & 6 & 10 \\ 5 & 4 & 5 & 7 & 12 \end{pmatrix}$$

$$17) \quad A = (a_{ij}) = \begin{pmatrix} 9 & 10 & 16 & 4 & 12 \\ 13 & 1 & 2 & 1 & 9 \\ 2 & 20 & 12 & 8 & 4 \\ 13 & 10 & 10 & 14 & 18 \\ 8 & 7 & 5 & 3 & 5 \end{pmatrix}$$

$$18) \quad A = (a_{ij}) = \begin{pmatrix} 3 & 10 & 6 & 10 & 10 \\ 10 & 2 & 1 & 8 & 12 \\ 16 & 10 & 15 & 10 & 10 \\ 9 & 14 & 10 & 10 & 10 \\ 5 & 12 & 5 & 6 & 7 \end{pmatrix}$$

$$19) \quad A = (a_{ij}) = \begin{pmatrix} 10 & 8 & 3 & 8 & 5 \\ 10 & 5 & 2 & 1 & 14 \\ 10 & 12 & 20 & 13 & 20 \\ 6 & 2 & 16 & 3 & 18 \\ 4 & 3 & 5 & 5 & 7 \end{pmatrix}$$

$$20) \quad A = (a_{ij}) = \begin{pmatrix} 10 & 6 & 14 & 16 & 7 \\ 8 & 3 & 3 & 8 & 5 \\ 12 & 20 & 14 & 20 & 10 \\ 10 & 5 & 18 & 16 & 4 \\ 6 & 4 & 5 & 8 & 9 \end{pmatrix}$$

$$21) \quad A = (a_{ij}) = \begin{pmatrix} 1 & 3 & 4 & 7 & 5 \\ 10 & 8 & 1 & 2 & 3 \\ 8 & 10 & 14 & 6 & 8 \\ 7 & 5 & 9 & 12 & 11 \\ 6 & 8 & 5 & 4 & 7 \end{pmatrix}$$

$$22) \quad A = (a_{ij}) = \begin{pmatrix} 3 & 5 & 11 & 8 & 10 \\ 8 & 7 & 1 & 4 & 3 \\ 10 & 11 & 12 & 9 & 7 \\ 6 & 4 & 7 & 8 & 10 \\ 5 & 6 & 5 & 10 & 9 \end{pmatrix}$$

$$23) \quad A = (a_{ij}) = \begin{pmatrix} 4 & 5 & 8 & 7 & 5 \\ 5 & 10 & 1 & 11 & 8 \\ 3 & 4 & 8 & 7 & 1 \\ 2 & 10 & 11 & 10 & 4 \\ 6 & 4 & 9 & 5 & 6 \end{pmatrix}$$

$$24) \quad A = (a_{ij}) = \begin{pmatrix} 3 & 5 & 7 & 5 & 4 \\ 10 & 12 & 1 & 4 & 8 \\ 10 & 11 & 10 & 8 & 7 \\ 10 & 8 & 9 & 7 & 8 \\ 7 & 5 & 6 & 3 & 10 \end{pmatrix}$$

Задание 4

Решить задачу о рюкзаке с общей ценностью груза c и данными, представленными таблицей

1) $c = 45$

i	1	2	3	4	5
c_i	20	24	5	20	9
p_i	4	14	2	7	3

2) $c = 45$

i	1	2	3	4	5
c_i	11	19	12	21	9
p_i	3	9	4	10	6

3) $c = 35$

i	1	2	3	4	5
c_i	10	10	5	15	9
p_i	5	4	3	7	3

4) $c = 50$

i	1	2	3	4	5
c_i	10	20	12	15	11
p_i	2	6	6	10	2

5) $c = 26$

i	1	2	3	4	5
c_i	10	6	11	15	12
p_i	6	3	8	10	5

6) $c = 40$

i	1	2	3	4	5
c_i	9	13	12	8	13
p_i	3	8	5	4	7

7) $c = 45$

i	1	2	3	4	5
c_i	15	9	10	8	22
p_i	11	3	4	3	12

8) $c = 55$

i	1	2	3	4	5
c_i	18	35	11	27	12
p_i	6	14	4	4	3

9) $c = 44$

i	1	2	3	4	5
c_i	10	6	15	24	18
p_i	4	2	3	13	8

10) $c = 38$

i	1	2	3	4	5
c_i	8	12	10	16	20
p_i	5	4	3	8	9

11) $c = 32$

i	1	2	3	4	5
c_i	16	10	12	8	6
p_i	7	6	5	4	2

12) $c = 48$

i	1	2	3	4	5
c_i	15	10	12	18	20
p_i	3	5	8	4	6

13) $c = 24$

i	1	2	3	4	5
c_i	10	4	5	10	12
p_i	7	2	3	4	8

14) $c = 38$

i	1	2	3	4	5
c_i	12	10	8	16	4
p_i	7	4	5	9	2

15) $c = 24$

i	1	2	3	4	5
c_i	15	8	9	10	4
p_i	3	6	3	2	2

16) $c = 42$

i	1	2	3	4	5
c_i	15	8	9	12	11
p_i	2	6	3	8	8

17) $c = 33$

i	1	2	3	4	5
c_i	10	25	12	16	6
p_i	4	5	8	6	3

18) $c = 34$

i	1	2	3	4	5
c_i	9	10	9	12	8
p_i	3	4	6	3	6

19) $c = 28$

i	1	2	3	4	5
c_i	8	5	9	10	3
p_i	2	3	3	3	2

20) $c = 50$

i	1	2	3	4	5
c_i	20	8	24	9	12
p_i	8	3	12	6	3

21) $c = 38$

i	1	2	3	4	5
c_i	16	14	8	9	10
p_i	6	7	2	2	4

22) $c = 36$

i	1	2	3	4	5
c_i	6	8	14	12	7
p_i	2	3	10	5	3

23) $c = 39$

i	1	2	3	4	5
c_i	15	10	8	9	10
p_i	5	3	4	2	4

24) $c = 49$

i	1	2	3	4	5
c_i	7	15	20	14	8
p_i	2	10	8	7	6

Задание 5

Имеется пять работ (P_j), состоящих каждая из двух операций, которые выполняются сначала на первой машине (M_1), затем на второй (M_2). Время выполнения операций задано в таблице. Определить оптимальный порядок выполнения работ, чтобы минимизировать время простоя машин.

1)

$P \backslash M$	P_1	P_2	P_3	P_4	P_5
M_1	2	7	17	8	6
M_2	7	13	14	12	12

2)

$P \backslash M$	P_1	P_2	P_3	P_4	P_5
M_1	5	3	13	15	2
M_2	12	10	8	5	9

3)

$P \backslash M$	P_1	P_2	P_3	P_4	P_5
M_1	5	10	7	3	6
M_2	14	3	12	6	9

4)

$P \backslash M$	P_1	P_2	P_3	P_4	P_5
M_1	5	4	10	15	3
M_2	2	11	8	15	9

5)

$P \backslash M$	P_1	P_2	P_3	P_4	P_5
M_1	6	6	17	14	8
M_2	7	13	9	12	12

6)

$P \backslash M$	P_1	P_2	P_3	P_4	P_5
M_1	16	3	5	15	12
M_2	11	7	8	14	5

7)

$P \backslash M$	P_1	P_2	P_3	P_4	P_5
M_1	14	6	14	12	8
M_2	7	10	6	11	12

8)

$P \backslash M$	P_1	P_2	P_3	P_4	P_5
M_1	16	3	4	13	12
M_2	4	7	12	3	5

9)

$P \backslash M$	P_1	P_2	P_3	P_4	P_5
M_1	2	15	17	8	6
M_2	13	3	14	2	10

10)

$P \backslash M$	P_1	P_2	P_3	P_4	P_5
M_1	5	3	13	10	12
M_2	4	10	6	5	3

11)

$P \backslash M$	P_1	P_2	P_3	P_4	P_5
M_1	5	10	7	3	6
M_2	14	3	12	6	9

12)

$P \backslash M$	P_1	P_2	P_3	P_4	P_5
M_1	6	4	2	15	3
M_2	2	11	8	10	11

13)

$P \backslash M$	P_1	P_2	P_3	P_4	P_5
M_1	11	4	17	15	8
M_2	7	13	4	14	12

14)

$P \backslash M$	P_1	P_2	P_3	P_4	P_5
M_1	15	3	5	15	12
M_2	8	7	6	3	5

15)

$P \backslash M$	P_1	P_2	P_3	P_4	P_5
M_1	4	6	14	15	8
M_2	8	4	6	9	12

16)

$P \backslash M$	P_1	P_2	P_3	P_4	P_5
M_1	13	3	4	6	12
M_2	6	7	12	13	15

17)

$P \backslash M$	P_1	P_2	P_3	P_4	P_5
M_1	3	7	17	4	6
M_2	9	3	14	10	12

18)

$P \backslash M$	P_1	P_2	P_3	P_4	P_5
M_1	5	12	13	14	8
M_2	12	10	11	5	9

19)

$P \backslash M$	P_1	P_2	P_3	P_4	P_5
M_1	12	10	8	16	7
M_2	8	13	12	16	9

20)

$P \backslash M$	P_1	P_2	P_3	P_4	P_5
M_1	5	4	2	15	13
M_2	2	11	14	5	9

21)

$P \backslash M$	P_1	P_2	P_3	P_4	P_5
M_1	6	8	17	14	8
M_2	14	13	10	9	6

22)

$P \backslash M$	P_1	P_2	P_3	P_4	P_5
M_1	16	6	15	5	12
M_2	11	17	8	3	6

23)

$P \backslash M$	P_1	P_2	P_3	P_4	P_5
M_1	12	6	4	12	9
M_2	4	15	7	11	10

24)

$P \backslash M$	P_1	P_2	P_3	P_4	P_5
M_1	6	3	14	13	3
M_2	4	8	12	5	15

ЛИТЕРАТУРА

1. Альсевич, В.В. Методы оптимизации: упражнения и задания: учеб. пособие / В.В. Альсевич, В.В. Крахотко. – Мн.: БГУ, 2005. – 405 с.
2. Андерсон, Д.А. Дискретная математика и комбинаторика / Д.А. Андерсон. – пер. с англ. – М.: Издательский дом «Вильямс», 2004. – 960 с.
3. Афанасьев, М.Ю. Прикладные задачи исследования операций: учеб. пособие / М.Ю. Афанасьев К.А. Багриновский, В.М. Матюшок. – М.: ИНФРА-М, 2006. – 352 с.
4. Гаврилов, Г.П. Задачи и упражнения по дискретной математике: учеб. пособие / Г.П. Гаврилов, А.А. Сапоженко. – 3-е изд. – М.: ФИЗМАТЛИТ, 2005. – 416 с.
5. Горбатов, В.А., Дискретная математика: учеб. для студентов втузов / В.А. Горбатов, А.В. Горбатов, М.В. Горбатова. – М.: АСТ: Астрель, 2006. – 447 с.
6. Дегтярев, Ю.И. Исследование операций: учеб. для вузов по спец. АСУ / Ю.И. Дегтярев. – М.: Выш. шк., 1986. – 320 с.
7. Емеличев, В.А. Лекции по теории графов: учеб. пособие / В.А. Емеличев [и др.]. – изд. 2-е. – М.: Книжный дом «ЛИБРОКОМ», 2009. – 392 с.
8. Иванов, Б.Н. Дискретная математика. Алгоритмы и программы. Полный курс / Б.Н. Иванов. – М.: ФИЗМАТЛИТ, 2007. – 408 с.
9. Костевич, Л.С. Исследование операций. Теория игр: учеб. пособие / Л.С. Костевич, А.А. Лапко. – 2-е изд. – Мн.: Выш. шк., 2008. – 368 с.
10. Кузнецов А.В. Высшая математика: Мат. программир.: учеб. – 2-е изд. / А.В. Кузнецов, В.А. Сакович, Н.И. Холод; под общ. ред. А.В. Кузнецова. – Мн.: Выш. шк., 2001. – 531 с.
11. Лавров, И.А. Задачи по теории множеств, математической логике и теории алгоритмов / И.А. Лавров, Л.Л. Максимова. – 5-е изд. – М.: ФИЗМАТЛИТ, 2004. – 256 с.
12. Мельников, О.И. Занимательные задачи по теории графов: учеб.-метод. пособие / О.И. Мельников. – 2-е изд. – Мн.: «ТетраСистем», 2001. – 144 с.
13. Новиков, Ф.А. Дискретная математика для программистов: учеб. для вузов. – 2-е изд. – СПб: Питер, 2006. – 364 с.

14. Оре, О. Теория графов / О. Оре; под ред. Н.Н. Воробьева; пер. с англ. – изд. 2-е. – М.: Книжный дом «ЛИБРОКОМ», 2009. – 352 с.
15. Осипова, В.А. Основы дискретной математики: учеб. пособие / В.А. Осипова. – М.: ФОРУМ: ИНФРА-М, 2006. – 160 с.
16. Плотников, А.Д. Дискретная математика: учеб. пособие / А.Д. Плотников. – 3-е изд. – Мн.: Новое знание, 2008. – 320 с.
17. Сачко, Н.С. Организация и оперативное управление машиностроительным производством: учеб. / Н.С. Сачко. – 2-е изд. – Мн.: Новое знание, 2006. – 636 с.
18. Сборник задач и упражнений по высшей математике: Мат. программирование: учеб. пособие / А.В. Кузнецов [и др.]; под общ. ред. А.В. Кузнецова. – 2-е изд. – Мн.: Выш. шк., 2002. – 447 с.
19. Соболева, Т.С. Дискретная математика: учеб. для студ. вузов / Т.С. Соболева, А.В. Чечкин; под ред. А.В. Чечкина. – М.: Издательский центр «Академия», 2006. – 256 с.
20. Сороговец, И.Б. Экономико-математические методы и модели: Методические указания к выполнению контрольных работ для студентов экономических специальностей заочного факультета. / И.Б. Сороговец, О.А. Дробинина. – Новополоцк, 2003. – 44 с.
21. Хаггарти, Р. Дискретная математика для программистов / Р. Хаггарти. – М.: Техносфера, 2005. – 400 с.
22. Шапоров, С.Д. Дискретная математика. Курс лекций и практических занятий / С.Д. Шапоров. – СПб.: БХВ-Петербург, 2006. – 400 с.
23. Шнеперман, Л.Б. Сборник задач по алгебре и теории чисел / Л.Б. Шнеперман. – Мн.: Дизайн ПРО, 2000. – 240 с.
24. Экономико-математические методы и модели: учеб. пособие / Н.И. Холод [и др.]; под общ. ред. А.В. Кузнецова. – 2-е изд. – Мн.: БГЭУ, 2000. – 412 с.
25. Яблонский, С.В. Введение в дискретную математику: учеб. пособие для вузов / С.В. Яблонский; под ред. В.А. Садовниченко. – 4-е изд. – М.: Выш. шк., 2003. – 384 с.

СОДЕРЖАНИЕ

Раздел 3. Основы теории графов	3
1. Графы. Орграфы	5
1.1. Графы и орграфы. Основные понятия	5
1.2. Связность графов	7
1.3. Изоморфизм графов	8
1.4. Степень вершин	10
1.5. Представление графов в компьютере	11
1.6. Выявление маршрутов с заданным количеством ребер (дуг)	14
1.7. Упорядочение вершин и дуг орграфов	17
2. Деревья. Остовные деревья	20
2.1. Деревья	20
2.2. Остовные деревья	25
2.3. Алгоритм Краскала и алгоритм Прима	28
3. Нахождение кратчайших путей. Алгоритм Дейкстры	35
4. Эйлеровы и гамильтоновы циклы	42
4.1. Пути и циклы Эйлера	42
4.2. Пути и циклы Гамильтона. Алгоритм Литтла	45
5. Планарные графы	69
5.1. Основные определения	69
5.2. Критерии планарности графа	71
6. Сети. Потоки в сетях	74
6.1. Сеть. Поток в сети	74
6.2. Разрез в сети	76
6.3. Алгоритм нахождения максимального потока в сети	78
7. Паросочетание	86
7.1. Паросочетание	86
7.2. Задача о назначениях	93
8. Элементы сетевого планирования	95
8.1. Сетевой график. Работа. Событие	95
8.2. Основные параметры сетевого графика	99
Вопросы и задания для самоконтроля	108
Задания для практических занятий	112
1. Графы и орграфы. Изоморфизм графов. Представление графов в компьютере	112
2. Упорядочение вершин и дуг орграфа	116
3. Остовные деревья. Алгоритмы прима и Краскала	117
4. Алгоритм Дейкстры	119
5. Эйлеровы и гамильтоновы циклы	120
6. Планарные графы	122
7. Сети. Потоки в сетях	123
8. Паросочетания	124
Индивидуальные задания	125
Раздел 4. Элементы математического моделирования	150
1. Математическая модель	150
1.1. Математическая модель	152
1.2. Классификация методов математического моделирования	153
2. Линейное программирование	155
2.1. Виды задач линейного программирования	155
2.2. Формы записи задач линейного программирования	162

2.3. Геометрическая интерпретация ЗЛП.....	166
3. Транспортная задача	175
3.1. Постановка транспортной задачи по критерию стоимости в матричной форме.....	175
3.2. Построение исходного опорного плана	179
3.3. Метод потенциалов	180
4. Дискретное программирование	190
4.1. Классические задачи целочисленного программирования	190
4.2. Суть методов дискретной оптимизации	193
4.3. Метод ветвей и границ.....	195
4.4. Решение общей задачи целочисленного программирования.....	197
4.5. Задача о назначении	203
4.6. Задача о рюкзаке	209
5. Элементы теории расписаний	220
5.1. Предмет теории расписаний и классификация задач	220
5.2. Формы представления расписаний.....	224
5.3. Задачи теории расписаний с одним обслуживающим устройством	225
5.4. Задачи теории расписаний с двумя последовательными обслуживающими устройствами	228
Вопросы для самоконтроля	231
Индивидуальные задания	233
Литература	245

Учебное издание

КАПУСТО Анна Владимировна
КУНЦЕВИЧ Ирина Павловна

ДИСКРЕТНАЯ МАТЕМАТИКА

Учебно-методический комплекс
для студентов машиностроительных специальностей

В двух частях

Часть 2

Редактор *А. Э. Цибульская*

Дизайн обложки *В. А. Виноградовой*

Подписано в печать 03.05.10. Формат 60x84¹/₁₆. Бумага офсетная. Гарнитура Таймс.
Ризография. Усл. печ. л. 14,62. Уч.-изд. л. 13,43. Тираж 175 экз. Заказ 840.

Издатель и полиграфическое исполнение:
учреждение образования «Полоцкий государственный университет»

ЛИ № 02330/0548568 от 26.06.2009

ЛП № 02330/0494256 от 27.05.2009

211440 г. Новополоцк, ул. Блохина, 29