Учебный центр «Резольвента»

Доктор физико-математических наук, профессор

К. Л. САМАРОВ

МАТЕМАТИКА

Учебно-методическое пособие по разделу

ЭЛЕМЕНТЫ ТЕОРИИ ГРАФОВ. ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ. СЕТЕВОЕ ПЛАНИРОВАНИЕ

СОДЕРЖАНИЕ

1. ЭЛЕМЕНТЫ ТЕОРИИ ГРАФОВ……………………………………….4

1.1. Основные понятия, определения и термины……………………………4

1.2. Задача о построении минимального остовного дерева………………8

2. ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ……………………………10

2.1. Общая схема метода динамического программирования …………10

2.2. Задача о распределении средств……………………………………12

3. СЕТЕВОЕ ПЛАНИРОВАНИЕ. ПРИМЕНЕНИЕ АЛГОРИТМОВ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ …………………………15

3.1. Понятие сети………………………………………………………....15

3.2. Построение сетевого графика технологического комплекса………..16

3.3.Постановка задачи о нахождении наименьшего времени выполнения технологического комплекса ………………………………17

3.4. Описание алгоритма динамического программирования для решения задачи о наименьшем времени выполнения технологического комплекса ……………………………………………18

3.4.1. Построение сетевого графика, упорядоченного по этапам……18

3.4.2. Расчет времени завершения узлов ……………………………19

3.4.3. Построение критического пути и нахождение критического времени завершения комплекса работ ……………………………20

3.4.4. Нахождение свободных резервов времени на некритических операциях……………………………………………………..20

3.4.5. Применение алгоритма динамического программирования для решения задачи о наименьшем времени выполнения технологического комплекса ……………………………………21

3.5. Постановка задачи о поиске в сети кратчайшего пути ………………22

3.6. Применение алгоритма динамического программирования для решения задачи о поиске в сети кратчайшего пути…………………23

ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ ……………………………………… 24

ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ……………………………26
ЛИТЕРАТУРА …………………………………………………………………27

1. ЭЛЕМЕНТЫ ТЕОРИИ ГРАФОВ

1.1. Основные понятия, определения и термины

Теория *графов* изучает математические объекты, которые можно изображать в виде рисунков, содержащих кружки и соединяющие их линии.

- Кружки называют *вершинами* (*узлами*) графа, а соединяющие линии – *ребрами* графа.

В качестве примера рассмотрим граф, изображенный на рис. 1. Этот граф состоит из 7 вершин $V_1,...,V_7$ и 8 ребер $\left[V_1,\, V_2\right]$, …, $\left[V_6,\, V_7\right]$.

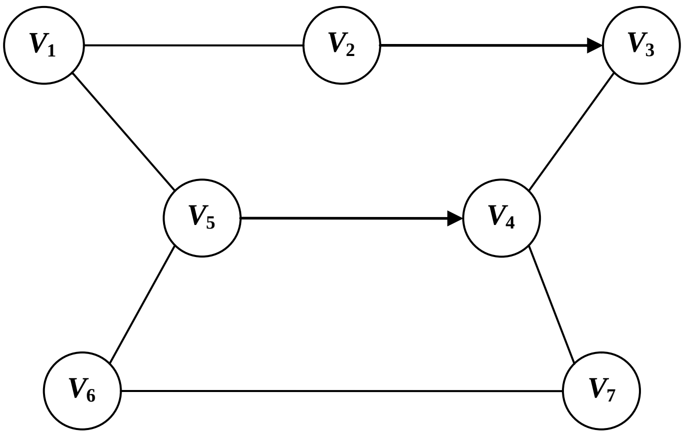


Рис.1.

- Две вершины графа, соединенные ребром, называют *смежными вершинами.*

Воспользовавшись определением смежных вершин, для каждого графа можно построить *матрицу смежности*. У графа, изображенного на рис. 1, матрица смежности имеет следующий вид:

$$C=\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} \qquad (1.1)$$

Если граф имеет n вершин $V_1, V_2, ..., V_n$, то его матрица смежности $C = c_{ij}$ является симметричной квадратной матрицей n - го порядка, состоящей из нулей и единиц. Элемент матрицы $c_{ij} = 1$, если вершины V_i и V_j являются смежными, в противном случае $c_{ij} = 0$.

- Две смежные вершины называют *граничными вершинами* соединяющего их ребра.

- Ребро графа называют *дугой*, если на нем *при помощи стрелки указано направление*.

На рис. 1 дугами являются ребра $\left[V_2, V_3\right]$ и $\left[V_5, V_4\right]$.

- Граничную вершину дуги, из которой стрелка выходит, называют *началом* дуги, а вершина, в которую стрелка входит, – *концом* дуги.

- Граф, в котором *каждое* ребро имеет направление (т.е. является дугой), называют *ориентированным графом.* В противном случае граф называют *неориентированным*.

- Ребро, соединяющее две смежные вершины, называют ребром, *инцидентным* этим вершинам.

Воспользовавшись определением инцидентности, для каждого графа можно построить *таблицу инцидентности*. Для графа, изображенного на рис. 1, таблица инцидентности имеет следующий вид:

$$B = \begin{pmatrix} & [V_1,V_2] & [V_1,V_5] & [V_2,V_3] & [V_3,V_4] & [V_4,V_5] & [V_4,V_7] & [V_5,V_6] & [V_6,V_7] \\ V_1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ V_2 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ V_3 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ V_4 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ V_5 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ V_6 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ V_7 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

В первом столбце таблицы инцидентности записаны все вершины графа, а первой строке – все ребра графа. Если вершина инцидентна ребру, то соответствующий элемент таблицы равен 1, в противном случае элемент таблицы равен 0.

- Число ребер, инцидентных вершине V_i графа, называют *степенью вершины* V_i и обозначают символом $\deg V_i$.

У графа, изображенного на рис.1,

$$\deg V_i = 2 \quad i = 1,2,3,6,7 \ , \ \deg V_4 = \deg V_5 = 3 \ .$$

- Граф, содержащий n вершин и m ребер, называют графом *типа* n, m .

Теорема Эйлера. Сумма степеней всех вершин графа типа n, m равна удвоенному числу его ребер, т.е.

$$\sum_{i=1}^{n} \deg V_i = 2m.$$

Следствие. В любом графе число вершин нечетной степени четно.

<u>Замечание.</u> Теорему Эйлера и следствие из нее легко проиллюстрировать на примере графа, изображенного на рис.1.

- Последовательность смежных вершин графа и связывающих их ребер, называют *цепью* (*путём, контуром*).

Каждая цепь состоит из *звеньев*.

- *Звеном* называют две последовательные вершины цепи и связывающее их ребро.

Если V_i и V_j две последовательные вершины цепи, то соответствующее им звено обозначают символом $V_i \to V_j$.

- *Простой цепью* называют цепь, все вершины которой различны.
- *Связным* называют граф, у которого две любые вершины связаны цепью.
- Замкнутую цепь называют *циклом*.

- Простую замкнутую цепь называют *простым циклом.*

В графе, изображенном на рис. 1, циклом является, в частности, замкнутая цепь, состоящая из следующих звеньев:

$$V_6 \to V_5 \to V_4 \to V_7 \to V_6 .$$

- Связный неориентированный граф, в котором существует простой цикл, содержащий все *ребра* графа, называют *эйлеровым графом*.

Теорема Эйлера. Связный неориентированный граф является эйлеровым графом тогда и только тогда, когда степени всех его вершин четны.

Поскольку у графа, изображенного на рис. 1, степени двух вершин нечетны, то этот граф не является эйлеровым графом.

Пример эйлерова графа можно получить, нарисовав на плоскости произвольный выпуклый многоугольник (рис. 2):

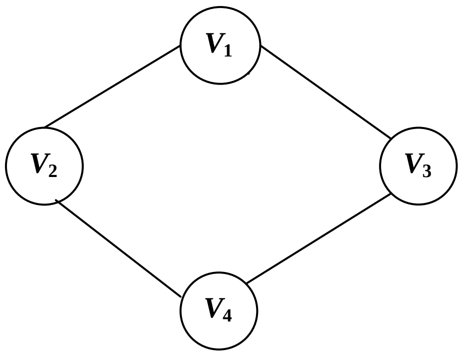


Рис. 2

- Связный неориентированный граф, в котором существует простой цикл, содержащий все *вершины* графа, называют *гамильтоновым графом.*

Теорема Оре. Если в графе не менее трех вершин, и для любых двух его несмежных вершин V_i и V_j выполнено неравенство

$$\deg V_i + \deg V_j \geq n ,$$

то граф является гамильтоновым графом.

<u>Замечание.</u> Граф, изображенный на рис. 2, является как эйлеровым, так и гамильтоновым графом.

- Связный неориентированный граф без циклов называют *деревом.*

Утверждение 1. Граф является деревом тогда и только тогда, когда две его любые вершины связаны *единственной цепью.*

Утверждение 2. Граф является деревом тогда и только тогда, когда, во-первых, он является связным, а, во-вторых, число его ребер на 1 меньше числа его вершин.

Примером дерева служит граф, изображенный на рис. 3.

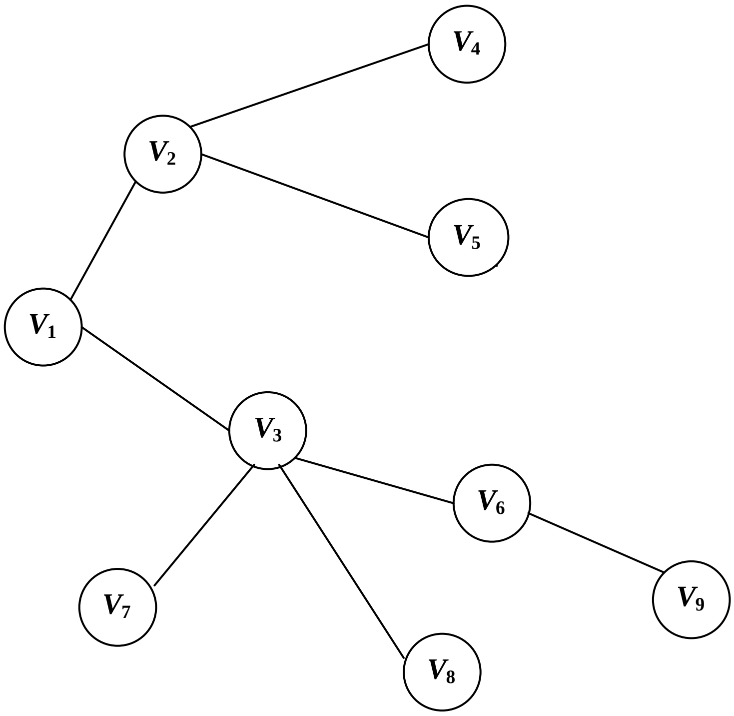


Рис. 3

- *Остовным деревом* графа называют дерево, все вершины которого совпадают с вершинами графа, а ребра являются ребрами графа.

У связного графа остовное дерево всегда существует.

- *Минимальным остовным деревом* связного графа с заданными длинами ребер называют остовное дерево, сумма длин ребер которого минимальна.

1.2. Задача о построении минимального остовного дерева

Следующая задача решается с помощью построения остовного дерева, имеющего наименьшую сумму длин ребер.

Задача 1.2. На рис. 4 изображены населенные пункты (вершины графа) и связывающие их грунтовые дороги (ребра графа):

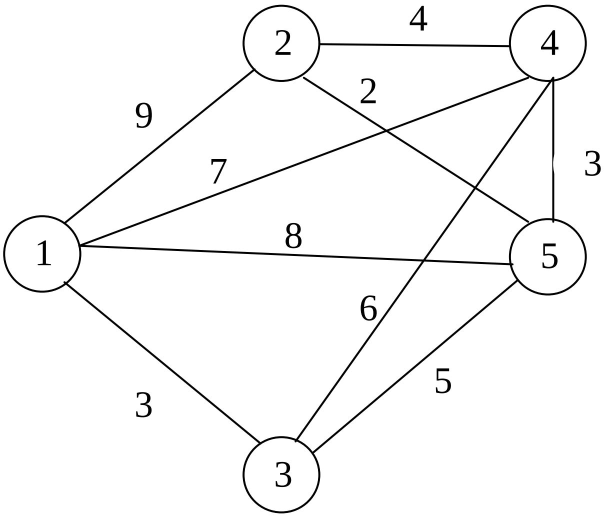


Рис. 4.

На рисунке также отмечены расстояния между населенными пунктами, выраженные в условных единицах. Требуется спланировать наиболее экономичную сеть дорог с твердым покрытием, заменяющих часть грунтовых дорог и связывающую все населенные пункты.

Решение. Рассмотрим какую-нибудь из вершин графа, изображенного на рис. 4, например, вершину № 3. Из всех ребер, соединяющих вершину № 3 с остальными вершинами графа, *самым коротким* является ребро, соединяющее вершины № 3 и № 1.

Теперь рассмотрим множество всех ребер, соединяющих вершины № 3 и № 1 с остальными вершинами графа. *Самым коротким* из них является ребро, соединяющее вершины № 3 и № 5.

Действуя по аналогии, рассмотрим множество всех ребер, соединяющих вершины № 3, № 1 и № 5 с остальными вершинами графа. *Самым коротким* из них является ребро, соединяющее вершины № 5 и № 2.

Наконец, рассмотрим множество всех ребер, соединяющих вершины № 3, № 1, № 5 и № 2 с остальными вершинами графа. *Самым коротким* из них является ребро, соединяющее вершины № 5 и № 4.

В результате мы получаем граф, изображенный на рис. 5:

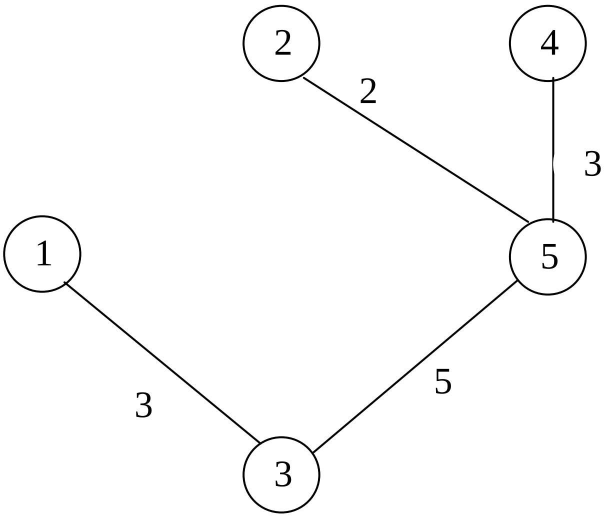


Рис. 5

Этот граф является остовным деревом для графа, изображенного на рис. 4, причем таким остовным деревом, которое обладает наименьшей суммой длин ребер, а дорожная сеть, изображенная на рис. 5, является решением рассматриваемой задачи. Длина этой дорожной сети равна 13 условным единицам.

Замечание. Если в качестве первого шага расчетного алгоритма, использованного при решении данной задачи, избрать не вершину с № 3, а любую другую вершину графа, то полученное в результате работы алгоритма минимальное остовное дерево будет тем же самым.

Решение задачи 1.2. завершено.

2. ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

2.1. Общая схема метода динамического программирования

Метод динамического программирования дает возможность находить последовательные оптимальные решения в задачах, разделенных на этапы.

Изложим схему применения этого метода на следующей модели. Рассмотрим некоторую управляемую систему, которая может находиться в одном из нескольких состояний. На каждом этапе в результате применения управляющего воздействия (*управления*) система может изменить свое состояние или остаться в прежнем состоянии. Эффективность процесса управления характеризуется *целевой функцией прибыли*, зависящей от состояния системы и применяемого управления.

0 этап. В начальный момент времени система находится в исходном состоянии x_0.

1 этап. В результате применения управления y_1 система переходит из состояния x_0 в состояние

$$x_1 = g_1(x_0, y_1),$$

при этом получается прибыль

$$h_1(x_0, y_1).$$

2 этап. В результате применения управления y_2 система переходит из состояния x_1 в состояние

$$x_2 = g_2(x_1, y_2),$$

при этом получается прибыль

$$h_2(x_1, y_2),$$

и так далее.

За N этапов получается последовательность состояний $x_0, x_1, x_2, \ldots x_N$ и последовательность управлений $y_1, y_2, \ldots y_N$, где

$$x_{n+1} = g_n(x_n, y_{n+1}), \quad n = 0, 1, \ldots, N,$$

а общая прибыль на каждом этапе вычисляется по формуле

$$J_n(x_0, y_1, \ldots y_n) = h_1(x_0, y_1) + h_2(x_1, y_2) + \ldots + h_n(x_{n-1}, y_n), \quad n = 1, 2, \ldots, N,$$

Нашей целью является отыскание такой последовательности оптимальных управлений $y_1^*, y_2^*, \ldots, y_N^*$, чтобы функция прибыли J_N достигала максимума

$$J_N(x_0, y_1^*, \ldots y_N^*) = \max_{y_1, \ldots y_N} J_N(x_0, y_1, \ldots y_N).$$

Принцип оптимальности Беллмана утверждает, что на последовательности оптимальных управлений $y_1^*, y_2^*, \ldots, y_N^*$ должна достигать максимума каждая из функций

$$f_n(x_{n-1}, y_n, y_{n+1}, \ldots y_N) = h_n(x_{n-1}, y_n) + h_{n+1}(x_n, y_{n+1}) + \ldots + h_N(x_{N-1}, y_N), n = 1, 2, \ldots, N.$$

Если ввести обозначения

$$\varphi_n(x_{n-1}) = \max_{y_n, \ldots y_N} f_n(x_{n-1}, y_n, \ldots y_N), n = 1, 2, \ldots, N,$$

то из принципа оптимальности Беллмана вытекает, что функции $\varphi_n(x_{n-1})$ должны удовлетворять следующим *функциональным уравнениям Беллмана:*

$$\varphi_n(x_{n-1}) = \max_{y_n} \left[\varphi_{n+1}(g_n(x_{n-1}, y_n)) + h_n(x_{n-1}, y_n) \right], n = 1, 2, \ldots, N. \qquad (2.1)$$

Решение уравнений Беллмана позволяет найти последовательность оптимальных управлений и оптимальное значение функции прибыли.

Замечание. Для того, чтобы методом динамического программирования находить не максимум функции прибыли, как было изложено выше, а *минимум функции* затрат достаточно лишь заменить во всех соответствующих формулах настоящего параграфа максимум на *минимум*.

2.2. Задача о распределении средств

Используем метод динамического программирования для решения следующей задачи.

Задача 2.2. Составить план распределения суммы в 4 миллиона долларов между тремя предприятиями $П_1$, $П_2$, $П_3$, приносящий наибольшую прибыль, если в каждое из предприятий может быть вложено 1, 2, 3 или 4 миллиона долларов, а прибыль каждого из предприятий задана в Таблице 2.2.1:

Таблица 2.2.1

Размер вложенных средств (млн. долл.)	**Прибыль предприятий (%)**		
	П$_1$	**П$_2$**	**П$_3$**
0	0	0	0
1	20	22	25
2	20	18	23
3	20	17	15
4	18	16	19

Решение. Пусть h_1, h_2, h_3 – прибыли (млн. долл.) предприятий П$_1$, П$_2$, П$_3$, соответственно. Тогда, пересчитывая данные из Таблицы 2.2.1, получим следующую Таблицу 2.2.2:

Т а б л и ц а 2.2.2

Размер вложенных средств (млн. долл.)	**Прибыль предприятий (млн. долл.)**		
	h_1	h_2	h_3
0	0	0	0
1	0,2	0,22	0,25
2	0,4	0,36	0,46
3	0,6	0,51	0,45
4	0,72	0,64	0,76

Разобьем процесс выделения средств предприятиям на 3 этапа: на первом этапе выделяется y_1 средств предприятию П$_1$, на втором – y_2 средств предприятию П$_2$, на третьем – y_3 средств предприятию П$_3$.

Будем считать состоянием системы x_i $i = 0,1,2,3$ ту сумму средств, которая осталась нераспределенной после i-го этапа. Поскольку необходимо распределить все 4 миллиона долларов, то $x_0 = 4$. Тогда

$$x_n = x_{n-1} - y_n, \quad n = 1,2,3.$$

Заметим, что на третьем этапе выделения средств весь остаток x_2 вкладывается в предприятие П$_3$, поэтому

$$y_3 = x_2.$$

Воспользуемся уравнениями Беллмана для $N = 3$. Тогда уравнения (2.1) примут следующий вид:

$$\varphi_3(x_2) = \max_{y_3=0,1,2,3,4} h_3(y_3),\ x_2 = 0,1,2,3,4, \qquad (2.2)$$

$$\varphi_2(x_1) = \max_{y_2=0,1,2,3,4} \left[h_2(y_2) + \varphi_3(x_1 - y_2)\right],\ x_1 = 0,1,2,3,4, \qquad (2.3)$$

$$\varphi_1(x_0) = \max_{y_1=0,1,2,3,4} \left[h_1(y_1) + \varphi_2(x_0 - y_1)\right],\ x_0 = 4. \qquad (2.4)$$

Обозначим значения управлений y_1, y_2, y_3, на которых достигается максимум в соотношениях (2.2), (2.3) и (2.4), символами y_1^*, y_2^*, y_3^*, соответственно, и, воспользовавшись Таблицей 2.2.2, заполним по формулам (2.2) Таблицу 2.2.3:

Таблица 2.2.3

x_2	y_3					$\varphi_3(x_2)$	y_3^*
	0	**1**	**2**	**3**	**4**		
0	0	–	–	–	–	0	0
1	–	0,25	–	–	–	0,25	1
2	–	–	0,46	–	–	0,46	2
3	–	–	–	0,45	–	0,45	3
4	–	–	–	–	0,76	0,76	4

Воспользовавшись Таблицей 2.2.2 и формулами (2.3), заполним Таблицу 2.2.4:

Таблица 2.2.4

x_1	y_2					$\varphi_2(x_1)$	y_2^*
	0	**1**	**2**	**3**	**4**		
0	0+0	–	–	–	–	0	0
1	0+0,25	0,22+0	–	–	–	0,25	0
2	0+0,46	0,22+0,25	0,36+0	–	–	0,47	1
3	0+0,45	0,22+0,46	0,36+0,25	0,51+0	–	0,68	1
4	0+0,76	0,22+0,45	0,36+0,46	0,51+0,25	0,64+0	0,82	2

Воспользовавшись Таблицей 2.2.2 и формулой (2.4), заполним Таблицу 2.2.5:

Таблица 2.2.5

x_0	y_1					$\varphi_1(x_0)$	y_1^*
	0	**1**	**2**	**3**	**4**		
4	0+0,82	0,2+0,68	0,4+0,47	0,6+0,25	0,72+0	0,88	1

Из Таблицы 2.2.5 вытекает, что оптимальным управлением будет $y_1^* = 1$, при этом оптимальная прибыль равна 0,88. Далее получаем

$$x_1 = x_0 - y_1^* = 4 - 1 = 3, \ \varphi_2(x_1) = \varphi_2(3) = 0{,}68, \ y_2^* = 1;$$
$$x_2 = x_1 - y_2^* = 3 - 1 = 2, \ \varphi_3(x_2) = \varphi_3(2) = 0{,}46, \ y_3^* = 2.$$

Таким образом, наиболее оптимальным является вложение в предприятия $П_1$, $П_2$, $П_3$ денежных средств в размере 1, 1 и 2 миллионов долларов, соответственно. В этом случае прибыль будет максимальной и составит 0,88 миллиона долларов. Решение задачи 2.2 завершено.

3. СЕТЕВОЕ ПЛАНИРОВАНИЕ. ПРИМЕНЕНИЕ АЛГОРИТМОВ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ

3.1. Понятие сети

Ориентированным циклом называют цикл, который можно обойти, следуя направлениям дуг.

Примером такого цикла служит цикл, изображенный на рис. 6.

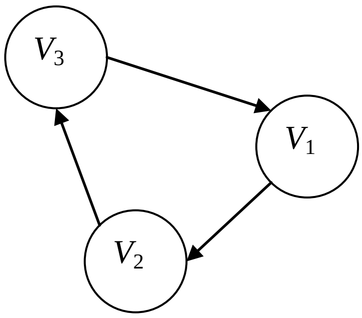


Рис. 6

Сетью называют связный ориентированный граф без ориентированных циклов, удовлетворяющий следующим условиям:

1. Граф имеет единственную вершину, из которой стрелки *выходят.* Эту вершину называют *источником.*
2. Граф имеет единственную вершину, в которую стрелки *входят*. Эту вершину называют *стоком.*

3. Все остальные вершины имеют как входящие, так и выходящие стрелки.
4. Две смежные вершины соединяет только одна стрелка.

Примером сети является граф, изображенный на рис. 7.

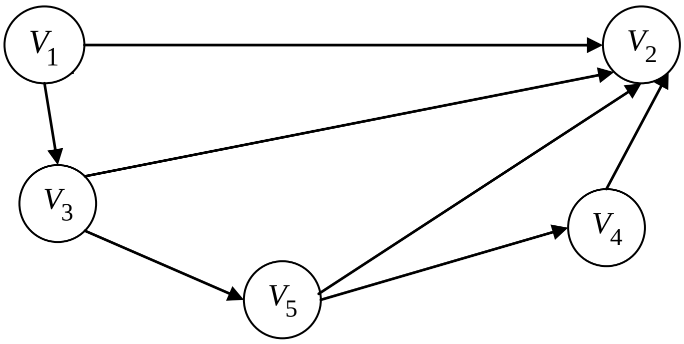


Рис. 7

3.2. Построение сетевого графика технологического комплекса

Понятие сети, введенное в предыдущем параграфе, используется для построения *сетевых графиков* технологических комплексов. Чтобы построить сетевой график, рассмотрим технологический комплекс, включающий нескольких технологических операций, и будем считать, что каждая технологическая операция характеризуется *временем начала и временем выполнения*. Будем также считать, что последовательность выполнения технологических операций известна. Если назвать начало и конец операции *узлами* технологического комплекса, то каждую операцию можно представлять как переход от одного узла к другому. Пронумеруем теперь узлы натуральными числами от 1 до n и введем следующие обозначения:

- Операцию перехода от i - го узла к j - му узлу обозначим символом $i \to j$. Символ $i \to j$ назовем *шифром* операции;
- Время выполнения операции $i \to j$ обозначим символом t_{ij};
- Время начала операции $i \to j$ обозначим символом t_i;
- Время окончания операции $i \to j$ обозначим символом t_j.

Изобразим теперь на рисунке каждый узел кружком, внутри которого указан номер узла. Начальный узел технологического комплекса (источник) изобразим кружком с номером 1. Конечный узел технологического комплекса (сток) изобразим кружком с номером n. Каждую операцию изобразим звеном, состоящим из двух кружков (узлов), соединенных дугой (стрелкой). Начало стрелки соответствует начальному узлу операции, конец стрелки – завершающему. Продолжительность t_{ij} операции укажем над стрелкой (рис. 8).

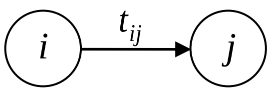


Рис. 8

В результате получим сеть, которую называют *сетевым графиком* рассматриваемого технологического комплекса.

3.3. Постановка задачи о нахождении наименьшего времени выполнения технологического комплекса

Рассмотрим следующую задачу, относящуюся к задачам сетевого планирования.

Задача 3.3. Технологический комплекс производства продукции состоит из 10 узлов. Последовательность выполнения операций и их продолжительность в часах заданы следующей Таблицей 3.3

Т а б л и ц а 3.3

№ п. п.	Шифр операции	Продолжительность операции
1	$1 \longrightarrow 2$	2
2	$1 \longrightarrow 3$	6
3	$1 \longrightarrow 4$	3
4	$2 \longrightarrow 3$	4
5	$2 \longrightarrow 5$	8
6	$3 \longrightarrow 5$	6
7	$3 \longrightarrow 6$	5

8	$3 \longrightarrow 7$	4
9	$4 \longrightarrow 3$	4
10	$4 \longrightarrow 6$	7
11	$4 \longrightarrow 7$	6
12	$5 \longrightarrow 8$	5
13	$6 \longrightarrow 8$	6
14	$6 \longrightarrow 9$	3
15	$6 \longrightarrow 10$	13
16	$7 \longrightarrow 9$	5
17	$8 \longrightarrow 10$	10
18	$9 \longrightarrow 10$	11

Требуется найти *наименьшее время* $T_{кр}$, необходимое для того, чтобы выполнить весь технологический комплекс (*критическое время*).

3.4. Описание алгоритма динамического программирования для решения задачи о наименьшем времени выполнения технологического комплекса

3.4.1. Построение сетевого графика, упорядоченного по этапам

Первым шагом алгоритма динамического программирования для решения указанной задачи сетевого планирования является построение *упорядоченного по этапам* эскиза сетевого графика технологического комплекса. Схема построения эскиза упорядоченного по этапам сетевого графика заключается в следующем:

- Кружок, соответствующий источнику, располагаем левее остальных кружков;
- Источник считаем этапом с номером 0;
- Каждый последующий этап изображаем правее предыдущего;
- Узлы одного этапа изображаем на одной вертикали;

- Каждый последующий этап формируем из узлов, которые не будут иметь входящих стрелок при мысленном удалении узлов с выходящими из них стрелками всех предыдущих этапов;
- Кружки добавляем на эскиз сетевого графика последовательно по этапам и без повторений пронумеровываем числами от 1 до n;
- Кружки соединяем стрелками в соответствии с заданной в таблице последовательностью выполнения технологических операций;
- Над стрелками, изображающими операции $i \to j$, проставляем времена t_{ij};
- Сток считаем конечным этапом;
- Проверяем выполнимость требований, которым должна удовлетворять сеть.

3.4.2. Расчет времени завершения узлов

- Считаем, что начало всех работ технологического комплекса, происходит в момент времени $t_1 = 0$. Число 0 проставляем над кружком (узлом) с номером 1;
- Время завершения каждого j - го узла из 1-го этапа рассчитываем по формуле $t_j = t_1 + t_{1j} = t_{1j}$;
- Время завершения каждого j - го узла из остальных этапов рассчитываем по формуле

$$t_j = \max_i (t_i + t_{ij}), \qquad (*)$$

где максимум берется по всем входящим в j - й *узел стрелкам*, а индексы i - номера предшествующих узлов, из которых выходят входящие в j - й узел стрелки;

- Время t_j проставляем над кружком, соответствующим j - у узлу.

<u>Замечание</u>. Соотношение $(*)$ и является для рассматриваемой задачи функциональным уравнением Р. Беллмана.

3.4.3. Построение критического пути и нахождение критического времени завершения комплекса работ

- Время t_n завершения последнего n - го узла назовем *критическим временем* завершения всего комплекса работ. Критическое время обозначим символом $T_{\text{кр}}$;

- *Критическим путем* назовем путь, который строится обратным ходом, начиная от последнего n - го узла, и достигает первого узла при помощи выделения стрелок, реализующих критическое время;

- Операции, составляющие критический путь, называются *критическими*.

Замечание. Критический путь может быть неединственным.

3.4.4. Нахождение свободных резервов времени на некритических операциях

- Операции, не являющиеся критическими, назовем *некритическими*;

- Путь, начальный и конечный узлы которого лежат на критическом пути, а составляющие его операции являются некритическими, назовем *некритическим*;

- *Свободный резерв времени* P_{ij}^{c} *на некритической операции* $i \to j$ находим по формуле

$$P_{ij}^{\text{c}} = t_j - t_i - t_{ij}$$

- Значение P_{ij}^{c} проставляем над стрелкой $i \to j$ в скобках справа от символа t_{ij} (рис. 9).

$$\textcircled{i} \xrightarrow{\;t_{ij}\,(P_{ij}^{c})\;} \textcircled{j}$$

Рис. 9

Замечание. На критических операциях резерв свободного времени равен нулю. Следовательно, для сокращения времени завершения всего комплекса работ в первую очередь необходимо *сокращать продолжительность критических операций.*

3.4.5. Применение алгоритма динамического программирования для решения задачи о наименьшем времени выполнения технологического комплекса

Проанализировав данные из Таблицы 3.3, составим первоначальный эскиз сетевого графика (рис. 10).

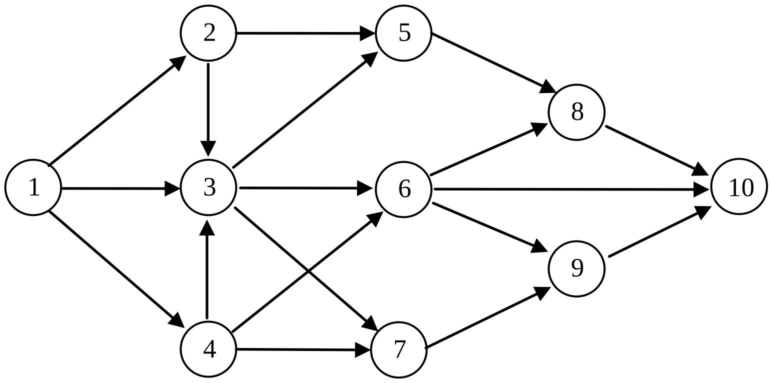


Рис. 10

Упорядочив первоначальный эскиз сетевого графика по этапам, и, произведя необходимые расчеты времен окончания операций, получим рис. 11.

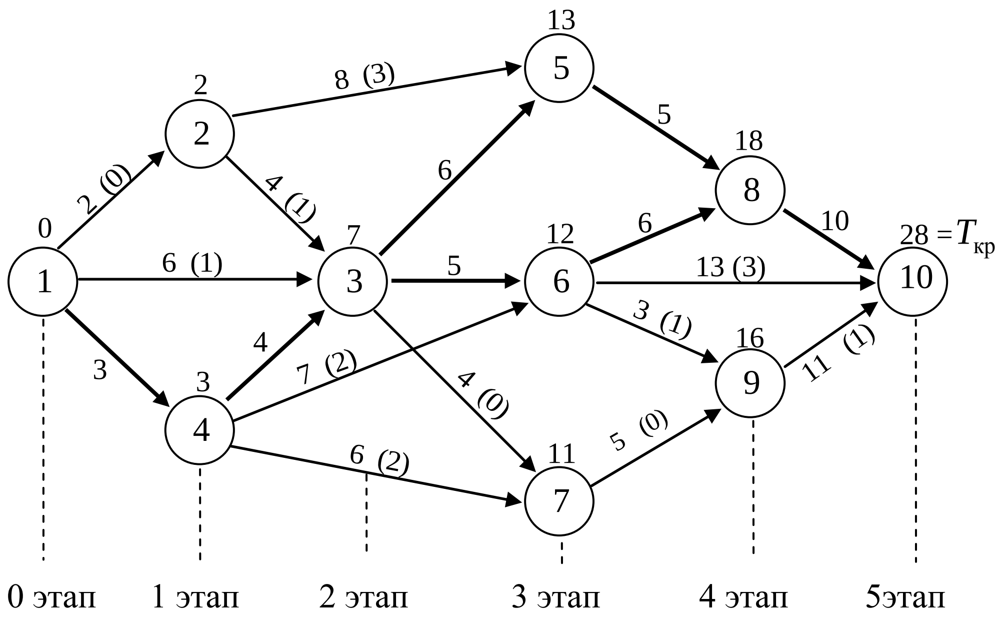


Рис. 11

Из рисунка 11 вытекает, что сетевой график содержит 5 этапов, $T_{кр} = 28$ часов. Заметим также, что сетевой график имеет два критических пути, которые на рис. 11 изображены жирными линиями и состоят из следующих узлов:

$$1 \to 4 \to 3 \to 5 \to 8 \to 10 \text{ и } 1 \to 4 \to 3 \to 6 \to 8 \to 10.$$

На каждом из критических путей критическое время одно и тоже – 28 часов. Свободные резервы времени на некритических операциях $P_{ij}^{с}$ на рисунке проставлены в скобках. Решение задачи 2.3. завершено.

3.5. Постановка задачи о поиске в сети кратчайшего пути

В разделе 3.4. мы рассмотрели задачу о поиске в сети *критического* пути. Целью данного раздела является постановка задачи о поиске в сети *кратчайшего* пути.

Задача 3.5. Рассмотрим сеть, изображенную на рис. 12. На этом рисунке над каждой дугой записано число, называемое *длиной дуги*. Требуется найти *кратчайший путь* (*путь минимальной длины*), ведущий из источника в сток.

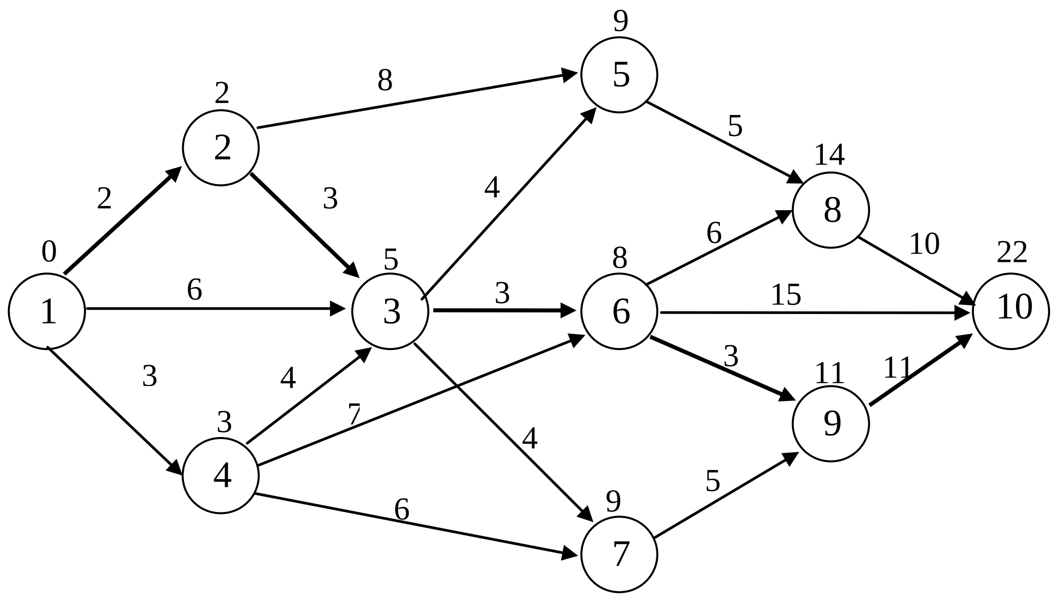


Рис. 12

3.6. Применение алгоритма динамического программирования для решения задачи о поиске в сети кратчайшего пути

Кратчайший путь, ведущий из источника сети в сток, изображен на рис. 12 при помощи последовательности жирных стрелок. Чтобы найти этот путь, требуется чуть-чуть изменить алгоритм *нахождения критического пути*, проставляя над каждым этапом *не максимум,* а *минимум* расстояний по всем входящим в рассматриваемый узел стрелкам. Совершив описанную операцию, находим, что кратчайший путь имеет вид

$$1 \to 2 \to 3 \to 6 \to 9 \to 10,$$

а длина его равна 22. Решение задачи 3.5. завершено.

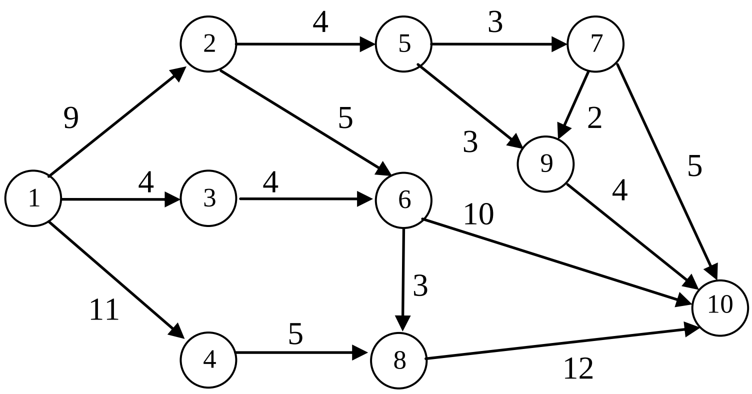
ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ

1. Как изображаются графы в виде рисунков?
2. Какие вершины графа называются смежными?
3. Что называется дугой графа?
4. Какой граф называется ориентированным?
5. Как построить матрицу смежности графа?
6. Как построить таблицу инцидентности графа?
7. Что называется степенью вершины графа?
8. Как определяется тип графа?
9. Как формулируется теорема Эйлера о степенях вершин графа?
10. Что называется цепью в графе?
11. Что называется звеном цепи графа?
12. Что называется простой цепью в графе?
13. Какой граф называется связным?
14. Что называется циклом в графе?
15. Какой граф называется эйлеровым графом?
16. Какой граф называется гамильтоновым графом?
17. Как формулируется теорема Эйлера об эйлеровых графах?
18. Как формулируется теорема Оре о гамильтоновых графах?
19. Какой граф называется деревом?
20. Что называется остовным деревом графа?
21. В чем состоит схема алгоритма для построения минимального остовного дерева графа?
22. В чем состоит общая схема метода динамического программирования?
23. Какие уравнения называются функциональными уравнениями Беллмана?
24. Что называется ориентированным циклом в графе?
25. Какой граф называется сетью?
26. Что называется источником и стоком сети?

27. Как строится сетевой график технологического комплекса?
28. Что называется критическим путем в сетевом графике?
29. Что называется критическим временем для сетевого графика?
30. Какие операции называются критическими в сетевом графике?
31. Какой путь называется некритическим путем в сетевом графике?
32. Что называется свободным резервом времени на некритической операции?
33. Может ли критический путь в сети быть неединственным?

ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

А. Для следующего графа



требуется:

1. Определить тип.
2. Составить матрицу смежности.
3. Составить таблицу инцидентности.
4. Найти сумму степеней всех вершин.
5. Построить минимальное остовное дерево.
6. Построить критический путь.
7. Найти критическое время.
8. Найти кратчайший путь, ведущий из источника в сток.
9. Найти длину кратчайшего пути, ведущего из источника в сток.

Б. Найти наименьшее значение целевой функции

$$Z=\sqrt{x^2+y^2}$$

при наличии ограничения

$$(x-5)^2+(y-12)^2\le 4,$$

и определить точку (x_0, y_0), в которой наименьшее значение целевой функции достигается.

ЛИТЕРАТУРА

Основная:

1. Вентцель Е.С. Исследование операций: Задачи, принципы, методология. Учебное пособие. – М.: Дрофа, 2004.
2. Колемаев В.А. Математическая экономика. Учебник для вузов. - М.: ЮНИТИ-ДАНА, 2005.
3. Кремер Н.Ш. Исследование операций в экономике. – М.: ЮНИТИ, 2006.
4. Орехов Н.А., Левин А.Г., Горбунов Е.А. Математические методы и модели в экономике. Учебное пособие для вузов / Под ред. проф. Н.А. Орехова – М.: ЮНИТИ-ДАНА, 2004.

Дополнительная:

5. Экономико-математическое моделирование. Учебник для вузов / Под общ. ред. И.Н. Дрогобыцкого. – М.: Изд. «Экзамен», 2004.
6. Макоха А.Н., Сахнюк П.А., Червяков Н.И. Дискретная математика: Учебное пособие – М.: Физматлит, 2005.
7. Малыхин В.И. Математика в экономике: Учебное пособие. – М.: ИНФРА-М, 2002.
8. Самаров К.Л., Шапкин А.С. Задачи с решениями по высшей математике и математическим методам в экономике: Учебное пособие – М.: Издательско-торговая корпорация «Дашков и К^o», 2007.
9. Таха Х.А. Введение в исследование операций. – М.: ВИЛЬЯМС, 2007.