ЛАБОРАТОРНАЯ РАБОТА № 1

МАРШРУТИЗАЦИЯ НА СЕТЯХ СВЯЗИ: АЛГОРИТМ ДЕЙКСТРЫ

1.1 Цель работы

Изучить алгоритм определения кратчайшего пути на сетях связи.

1.2 Кратчайшие пути в сетях связи

Распределение информации в сетях связи производится с учетом оптимальности пути. При этом очевидно, что информацию целесообразно передавать по наиболее «коротким» путям или по кратчайшим путям.

Кратчайшим путем передачи информации называется путь, для которого критерий длины пути имеет наименьшее значение из всех возможных путей.

Для оценки длины пути могут быть использованы различные критерии: число транзитных узлов в пути, протяженность пути, качество тракта, вероятность установления соединения, надежность передачи информации и т.п.

В дальнейшем будем решать задачу поиска наилучшего маршрута в смысле кратчайшего расстояния. Эта задача моделируется с помощью сети связи $G=(A,B)$, в которой каждому ребру B приписан положительный целый вес, равный длине ребра. Длина пути между заданными узлами A равна сумме длин ребер, составляющих путь. В терминах сетей связи задача сводится к отысканию кратчайшего пути между заданными узлами.

1.3 Алгоритм Дейкстры

Алгоритм позволяет находить в сети кратчайший путь между двумя выделенными узлами сети i и j. Для этого сеть связи представим матрицей расстояний L, элементы которой L_{kl} означают длину ребра между узлами k и l и равны:

$l_{kl}=\infty$, если между узлами k и l нет ребра;
$l_{kl}=0$ для всех $k=l$; $k,l=1,2,\dots M$;
l_{kl} – длине ребра между узлами k и l,

где M – количество узлов на сети.

Метод Дейкстры состоит из выполнения следующих шагов:

1. Начинаем с непосредственных расстояний, с длины в одно ребро от заданного узла i до всех остальных узлов.

2. Затем выбираем наименьшее из них в качестве «постоянного» наименьшего расстояния, фиксируя узел, до которого наименьшее расстояние, в качестве нового узла.

3. Далее добавляем это наименьшее расстояние к длинам ребер от нового узла до всех остальных узлов.

4. Сравниваем эту сумму с предыдущим расстоянием от узла до остальных узлов и заменяем прежнее расстояние, если новое меньше.

5. Затем новый узел удаляем из списка узлов, до которых еще не определены кратчайшие расстояния, и ему присваиваем «постоянную» метку.

Затем шаги 1...5 повторяем, присоединяя новое кратчайшее расстояние к списку «постоянных» узлов и т.д., пока конечный узел j не окажется соединенным с узлом i путем из выделенных ребер.

Теперь можно сформулировать алгоритм Дейкстры.

Алгоритм Дейкстры служит для определения кратчайшего расстояния L_{kl} от заданного начального узла i до конечного узла j в связной сети связи G, имеющей M узлов и N ребер и представленной матрицей расстояний L**.**

Шаг 0**.** Отмечаем метками все узлы, для этого припишем узлу i «постоянную» метку, а остальным узлам сети «временные» метки.

Шаг 1. Присвоим длину l_{kl} всем ребрам сети между узлами, имеющими непосредственную связь; если между узлами k и l нет ребра, то $l_{kl} = \infty$; $l_{kl} = 0$ для всех $k = l, k, l = 1,2, \dots M$. Присвоим узлу i вес, равный нулю, т.е.$d_i = 0$, остальным узлам присвоим веса, равные бесконечности, т.е. $d_k = \infty, k \neq 1$. Черта над индексом означает, что метка d_i– постоянная.

Шаг 2. Если узел j не включен в список узлов с «постоянной» меткой, то идти к шагу 3, в противном случае задача решена.

Шаг 3. Для каждого узла k с «временной» меткой определим меньшее расстояние по формуле

$$d_k = min[d_k\,, d_m + l_{mk}],\ k = 2,3,4,5,6, m = 1$$

где d_m– вес узла, который включен в список с «постоянной» меткой последним.

Шаг 4. Пусть k – узел, из числа узлов с «временными» метками, до которого расстояние d_k– наименьшее среди всех узлов с «временными» метками; припишем узлу k «постоянную» метку и присвоим ему постоянный вес, равный d_k.

Шаги 2, 3, 4 повторять до тех пор, пока узел j не будет включен в список узлов с «постоянной» меткой.

Продемонстрируем работу алгоритма Дейкстры на примере.

Пример. Для заданной структуры сети (рис.1.1) определить кратчайший путь между узлами 1 и 6. Цифры возле ребер обозначают длину каждого ребра.

Шаг 0. Припишем узлу 1 «постоянную» метку, а остальным узлам – «временные» метки, т.е.

$$C = \{1\}, \qquad C = \{2,3,4,5,6\}$$

Шаг 1. Присвоим длину всем ребрам, т.е. составим матрицу расстояний L:

$$\begin{pmatrix} 0 & 4 & 3 & 7 & \infty & \infty \\ \infty & 0 & \infty & 3 & 2 & \infty \\ \infty & \infty & 0 & \infty & 3 & \infty \\ \infty & \infty & \infty & 0 & \infty & 2 \\ \infty & \infty & \infty & \infty & 0 & 2 \\ \infty & \infty & \infty & \infty & \infty & 0 \end{pmatrix}$$

Присвоим узлу 1 постоянный вес $d_1 = 0$, а остальным узлам временные веса $d_k = \infty,\ k = 2,3,4,5,6$. Следовательно, $m = 1$.

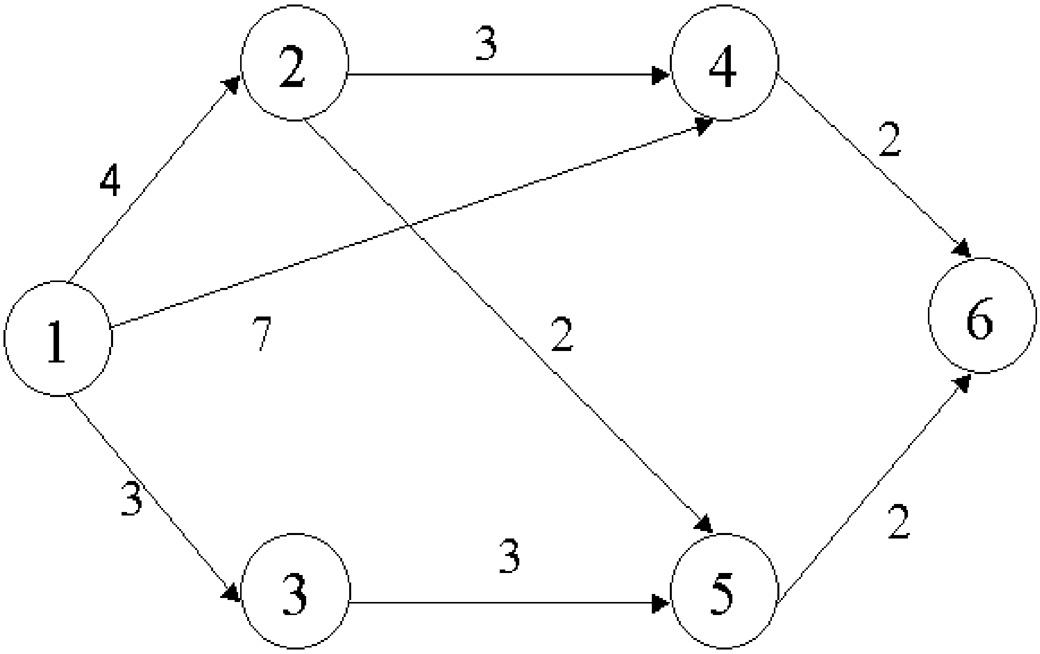


Рисунок 1.1

И т е р а ц и я 1

Шаг 2. Так как узел 6 не включен в список узлов с "постоянной" меткой, то идем к шагу 3.

Шаг 3. Для всех узлов с "временными" метками определим веса по формуле

$$d_k = min[d_k, d_m + l_{mk}], \text{ где } k = 2,3,4,5,6;\ m = 1.$$

Подставляя поочередно k в последнюю формулу, получим

$$d_2 = min[d_2, d_1 + l_{12}] = min[\infty, 0 + 4] = 4,$$
$$d_3 = min[d_3, d_1 + l_{13}] = min[\infty, 0 + 3] = 3,$$
$$d_4 = min[d_4, d_1 + l_{14}] = min[\infty, 0 + 7] = 7,$$
$$d_5 = min[d_5, d_1 + l_{15}] = min[\infty, 0 + \infty] = \infty,$$

$$d_6 = min[d_6, d_1 + l_{16}] = min[\infty, 0 + \infty] = \infty.$$

Шаг 4. Определим наименьший вес из полученных на третьем шаге по формуле

$$min[d_k] = min[4, 3, 7, \infty, \infty] = 3$$

Следовательно, узлу 3 припишем «постоянную» метку и присвоим постоянный вес, равный 3, т.е.

$$C = \{1,3\}, C = \{2,4,5,6\},\ d_1 = 0, d_3 = 3, d_2 = 4,\ d_4 = 7, d_5 = \infty, d_6 = \infty$$

Итерация 2

Шаг 2. Так как узел 6 не включен в список C, идти к шагу 3.

Шаг 3. Для всех узлов с «временными» метками определим веса по формуле

$d_k = min[d_k, d_m + l_{mk}]$, где $k = 2,4,5,6, m = 3$ (так как узел **3** в список с «постоянными» метками включен последним).

Получим

$$d_2 = min[d_2, d_3 + l_{32}] = min[4, 3 + \infty] = 4,$$
$$d_4 = min[d_4, d_3 + l_{34}] = min[7, 3 + \infty] = 7,$$
$$d_5 = min[d_5, d_3 + l_{35}] = min[\infty, 3 + 3] = 6,$$
$$d_6 = min[d_6, d_3 + l_{36}] = min[\infty, 3 + \infty] = \infty.$$

Шаг 4. Определим наименьший вес

$min[d_k] = min[4, 7, 6, \infty] = 4$

Следовательно, $C = \{1,3,2\},\ C = \{4,5,6\},\ d_1 = 0,\ d_3 = 3,\ d_2 = 4,\ d_4 = 7,\ d_5 = 6,\ d_6 = \infty.$

Итерация 3

Шаг 2. Идти к шагу 3.

Шаг 3. Определим веса при $k = 4,5,6, m = 2.$

$$d_4 = min[d_4, d_2 + l_{24}] = min[7, 4 + 3] = 7,$$
$$d_5 = min[d_5, d_2 + l_{25}] = min[6, 4 + 2] = 6,$$
$$d_6 = min[d_6, d_2 + l_{26}] = min[\infty, 4 + \infty] = \infty.$$

Шаг 4.

$$min[d_k] = min[7, 6, \infty] = 6,$$
$$C = \{1,3,2,5\},\ C = \{4,6\},$$
$$d_1 = 0,\ d_3 = 3,\ d_2 = 4,\ d_5 = 6,\ d_4 = 7,\ d_6 = \infty.$$

Итерация 4

Шаг 2. Идти к шагу 3.

Шаг 3. Определим веса при $k = 4{,}6$; $m = 5$:

$$d_4 = min[d_4, d_5 + l_{54}] = min[7, 6 + \infty] = 7,$$
$$d_6 = min[d_6, d_5 + l_{56}] = min[\infty, 6 + 2] = 8.$$

Шаг 4.

$$min[d_k] = min[7{,}8] = 7$$
$$C = \{1{,}2{,}3{,}4{,}5\},\ C = \{6\},$$
$d_1 = 0, d_2 = 4, d_3 = 3, d_4 = 7, d_5 = 6, d_6 = 8.$

Итерация 5

Шаг 2. Идти к шагу 3.

Шаг 3. Определим веса при $k = 6$; $m = 4$:

$$d_6 = min[d_6, d_4 + l_{46}] = min[8, 7 + 2] = 8$$
$$C = \{1{,}2{,}3{,}4{,}5{,}6\},\ C = \{0\},$$
$$d_1 = 0, d_2 = 4, d_3 = 3, d_4 = 7, d_5 = 6, d_6 = 8.$$

Итерация 6

Шаг 2. Так как узел 6 включен в список узлов с «постоянной» меткой, то задача решена.

Промежуточные результаты, полученные при решении данной задачи, приведены в таблице 1.

Как видно из таблицы 1.1, узлу **6** приписывается постоянная метка $d_6 = 8$. Следовательно, длина кратчайшего пути из узла 1 в узел 6 равна 8. Этот путь состоит из ребер, для каждого из которых разность между значениями постоянных меток ее концевых узлов равна длине этого ребра. Иными словами, если d_i и d_j – постоянные метки узлов i и j, соответственно, то условие, при выполнении которого эти узлы принадлежат кратчайшему пути, может быть записано следующим образом:

$$d_j = d_i + l_{ij}$$

Таблица 1.1

Итерация	Узлы					
	1	2	3	4	5	6
0	0	∞	∞	∞	∞	∞
1	0	4	3	7	∞	∞
2	0	4	3	7	6	∞
3	0	4	3	7	6	∞
4	0	4	3	7	6	8
5	0	4	3	7	6	8

Последнее соотношение можно использовать рекурсивно, двигаясь от узла j к узлу i. Определив узел, непосредственно предшествующий j в кратчайшей цепи, будем повторять данную процедуру до тех пор, пока не достигнем узла i. Покажем, как это делать, обратившись к нашему примеру.

1. Определим $d_6 = d_i + l_{i6}$ для $i = 1,2,3,4,5$:

$$d_6 = d_1 + l_{16} = 0 + \infty = \infty,$$

$$d_6 = d_2 + l_{26} = 4 + \infty = \infty,$$

$$d_6 = d_3 + l_{36} = 3 + \infty = \infty,$$

$$d_6 = d_4 + l_{46} = 7 + 2 = 9,$$

$$d_6 = d_5 + l_{56} = 6 + 2 = 8.$$

Так как постоянный вес d_6 равен 8, то из анализа последних выражений видно, что на кратчайшем пути из узла 1 в узел 6 находится узел 5.

2. Далее определим $d_5 = d_i + l_{i5}$ для $i = 1,2,3,4,6$:

$$d_5 = d_1 + l_{15} = 0 + \infty = \infty,$$

$$d_5 = d_2 + l_{25} = 4 + 2 = 6,$$

$$d_5 = d_3 + l_{35} = 3 + 3 = 6,$$

$$d_5 = d_4 + l_{45} = 7 + \infty = \infty,$$

$$d_5 = d_6 + l_{65} = 8 + \infty = \infty.$$

Так как постоянный вес $d_5 = 6$, то на кратчайшем пути из узла 1 в узел 6 находится как узел 2, так и узел 3, выбираем любой из них, например, узел 2.

3. Определим $d_2 = d_i + l_{i2}$ для $i = 1,3,4,5,6$:

$$d_2 = d_1 + l_{12} = 0 + 4 = 4,$$

$$d_2 = d_3 + l_{32} = 3 + \infty = \infty,$$

$$d_2 = d_4 + l_{42} = 7 + \infty = \infty,$$

$$d_2 = d_5 + l_{52} = 6 + \infty = \infty,$$

$$d_2 = d_6 + l_{62} = 8 + \infty = \infty.$$

Так как постоянный вес $d_2 = 4$, узла 1 в узел 6 находится узел 1.

Следовательно, мы показали, что кратчайший путь в рассмотренной нами сети связи образуется последовательностью узлов 1→ 2→ 5→ 6 либо

1→ 3→ 5→ 6, так как здесь есть альтернативное решение при переходе из узла 5.

1.4 Исходные данные к работе

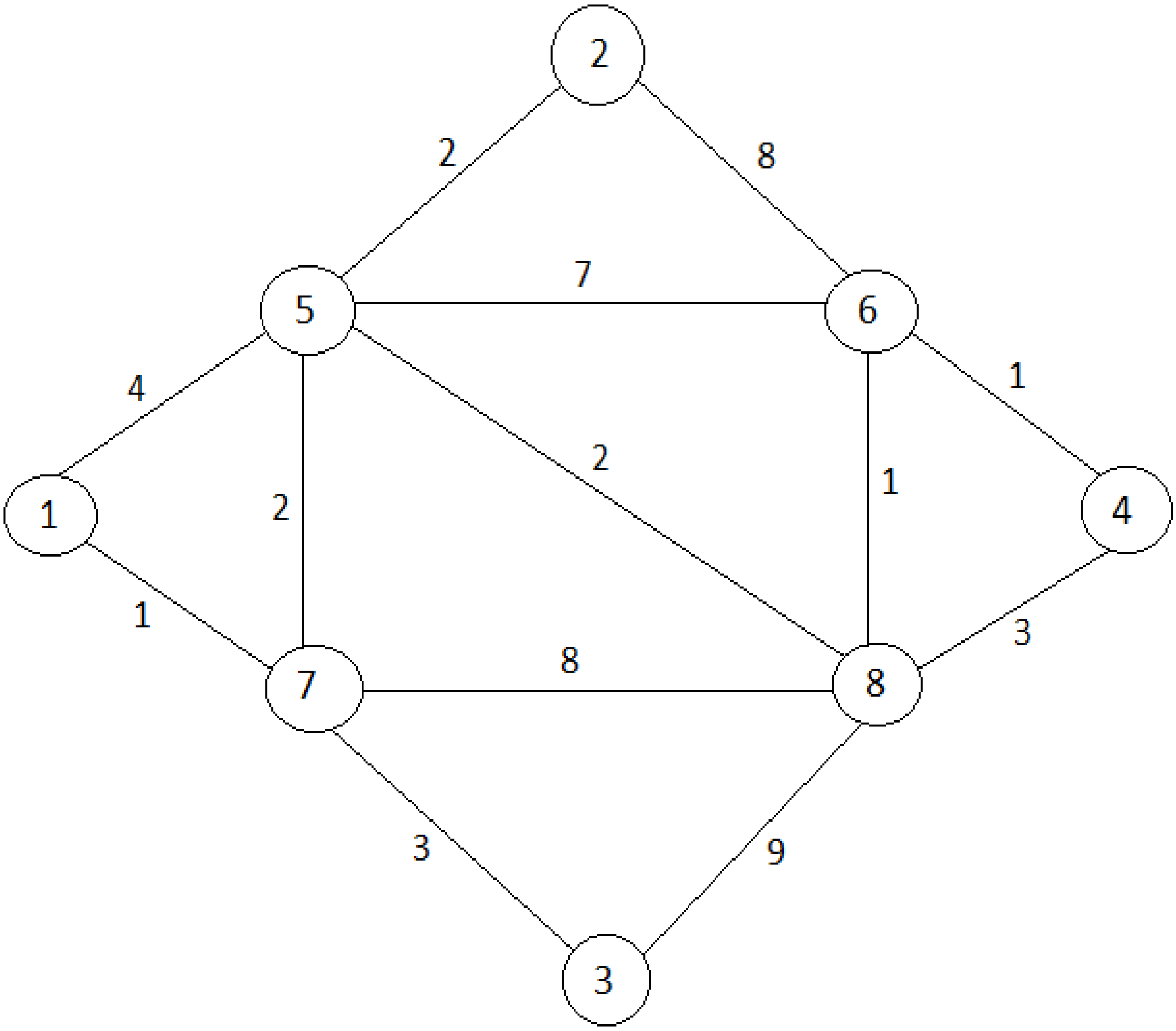


Рисунок 1.2

Вариант	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Направление	1→4	4→1	3→2	2→3	7→6	6→7	6→1	1→6	7→4	4→7
Вариант	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Направление	1→3	3→1	3→5	5→3	7→2	2→7	6→3	3→6	7→6	4→1
Вариант	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
Направление	1→8	8→1	3→4	4→3	7→1	6→7	6→1	7→6	5→4	4→5

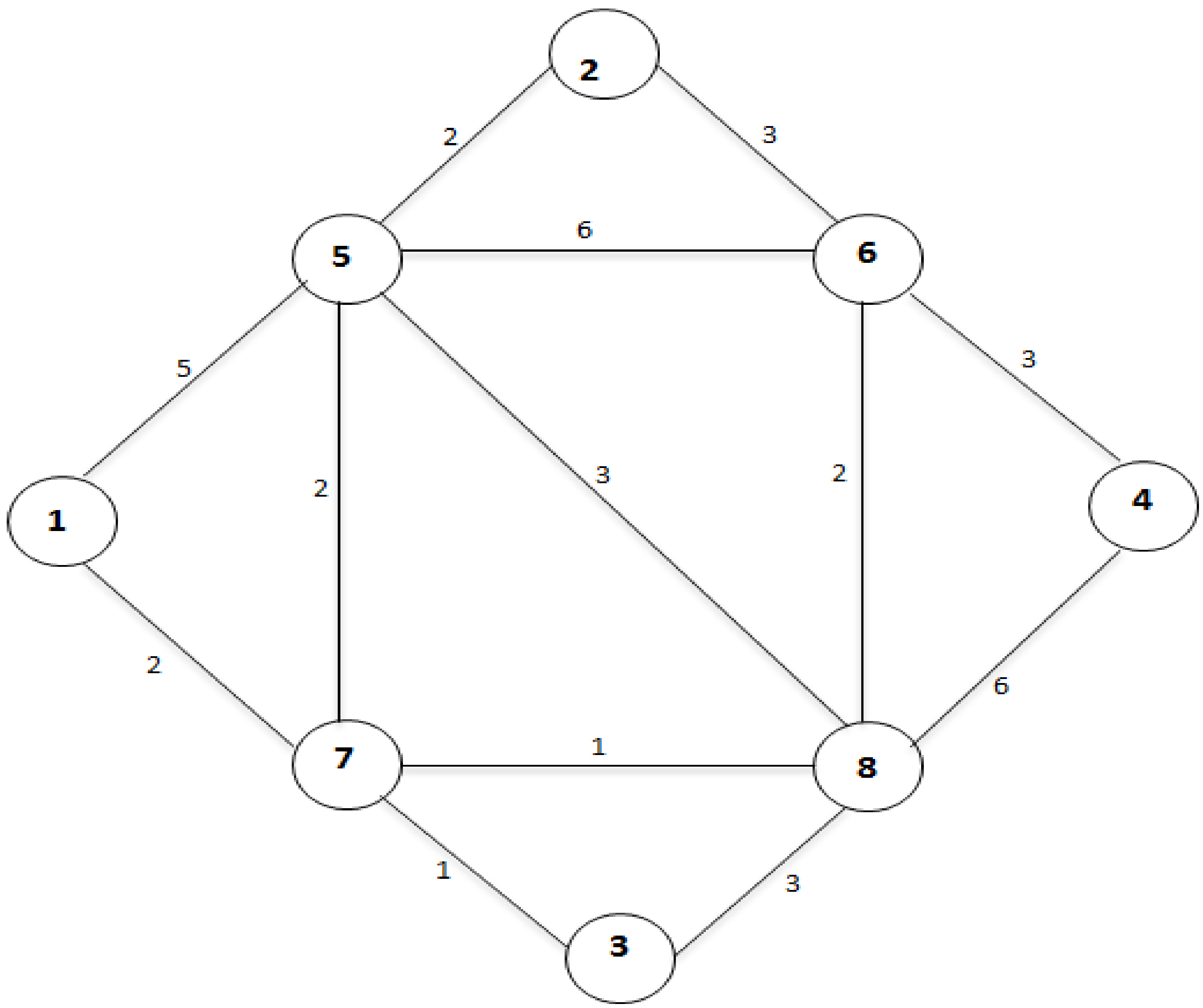


Рисунок 1.3

Вариант	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Направление	1→4	4→1	3→2	2→3	7→6	6→7	6→1	1→6	7→4	4→7
Вариант	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Направление	1→3	3→1	3→5	5→3	7→2	2→7	6→3	3→6	7→6	4→1
Вариант	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
Направление	1→8	8→1	3→4	4→3	7→1	6→7	6→1	7→6	5→4	4→5

1.5 Порядок выполнения работы:

1. Получить исходные данные к работе.
2. Построить сеть связи.
3. Решить задачу на ЭВМ.
4. Используя метод Дейкстры, определить кратчайшие расстояния в сети.

1.6 Содержание отчета:

1. Цель работы.
2. Ручной расчет сети.
3. Блок-схема алгоритма, расчет на ЭВМ.

Лабораторная работа №2

НАХОЖДЕНИЕ КРАТЧАЙШЕГО РАССТОЯНИЯ НА СЕТИ С ПОМОЩЬЮ АЛГОРИТМА ФЛОЙДА

Алгоритм Флойда является одним из популярных алгоритмов маршрутизации, позволяющий находить кратчайший путь на сети связи, заменяя прямой маршрут между двумя узлами на маршрут с одним промежуточным пунктом. Рассмотрим как это реализуется на практике. Пусть есть три узла i, j и k и заданы расстояния между ними (рис. 1). Если выполняется неравенство $d_{ij} + d_{jk} < d_{ik}$, то целесообразно заменить путь $i \rightarrow k$ путем $i \rightarrow j \rightarrow k$. Такая замена (далее ее будем условно называть треугольным оператором) выполняется систематически в процессе выполнения алгоритма Флойда. Алгоритм Флойда требует выполнения нескольких операций.

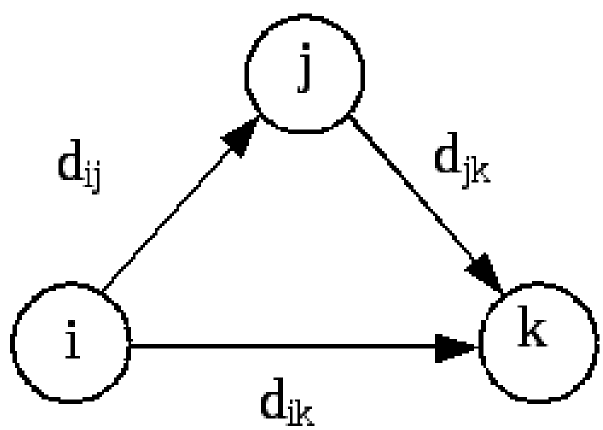


Рис. 1. Треугольный оператор Флойда.

Шаг 0. На этом шаге определим начальную матрицу расстояний D_0 и матрицу последовательности узлов S_0. Диагональные элементы обеих матриц помечаются знаком "-", показывающим, что эти элементы в вычислениях не участвуют. Полагаем $k = 1$, тогда начальная ситуация будет выглядеть следующим образом:

$D_0=$

	1	2	…	j	…	n
1	-	d_{12}	…	d_{1j}	…	d_{1n}
2	d_{21}	-	…	d_{2j}	…	d_{2n}
⋮	⋮	⋮	⋮	⋮	⋮	⋮
i	d_{i1}	d_{i2}	…	-	…	d_{in}
⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	d_{n1}	d_{n2}		d_{nj}		-

$S_0=$

	1	2	…	j	…	n
1	-	2	…	j	…	n
2	1	-	…	j	…	n
⋮	⋮	⋮	⋮	⋮	⋮	⋮
i	1	2	…	-	…	n
⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	1	2		j		-

Основной шаг k. Задаем строку k и столбец k как ведущую строку и ведущий столбец. Рассматриваем возможность применения треугольного оператора ко всем элементам d_{ij} матрицы D_{k-1}. Если выполняется неравенство $d_{ik} + d_{kj} < d_{ij}$, тогда выполняем следующие действия:

- создаем матрицу D_k путем замены в матрице *Dk-1* элемента *dij* на сумму *dik + dkj*,

- создаем матрицу S_k путем замены в матрице S_{k-1} элемента s_{ij} на k. Полагаем $k = k + 1$ и повторяем шаг k.

Поясним действия, выполняемые на k-м шаге алгоритма, представив матрицу D_{k-1} так, как она показана на рисунке 2. На этом рисунке строка k и столбец k являются ведущими. Строка i - любая строка с номером от 1 до $k - 1$, а строка p - произвольная строка с номером от $k + 1$ до n. Аналогично столбец j представляет любой столбец с номером от 1 до $k - 1$, столбец q - произвольный столбец с номером от $k + 1$ до n. Треугольный оператор выполняется следующим образом. Если сумма элементов ведущих строки и столбца (показанных в квадратах) меньше элементов, находящихся в пересечении столбца и строки (показанных в кружках), соответствующих рассматриваемым ведущим элементам, то расстояние (элемент в кружке) заменяется на сумму расстояний, представленных ведущими элементами:

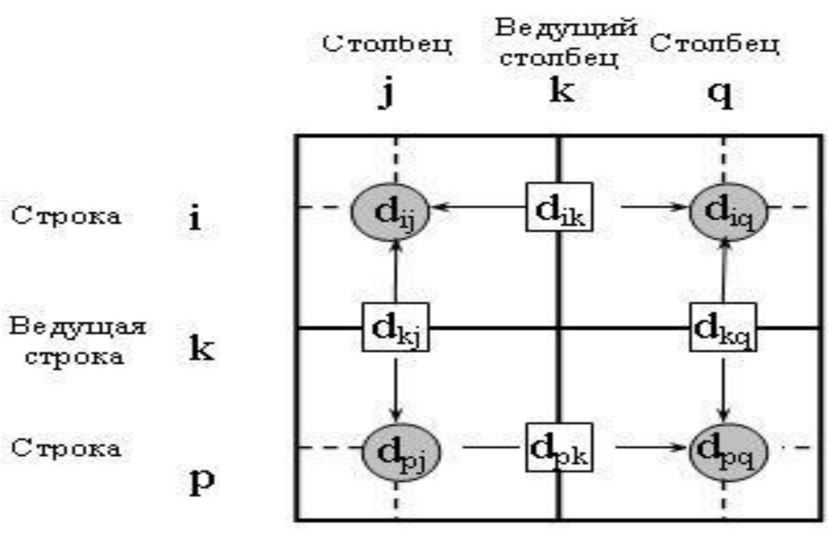


Рис. 2. Иллюстрация алгоритма Флойда

После реализации n шагов алгоритма определение по матрицам D_n и S_n кратчайшего пути между узлами i и j выполняется по следующим правилам.

1. Расстояние между узлами i и j равно элементу d_{ij} в матрице D_n.
2. Промежуточные узлы пути от узла i к узлу j определяем по матрице S_n. Пусть $s_{ij} = k$, тогда имеем путь $i \rightarrow k \rightarrow j$. Если далее $s_{ik} = k$ и $s_{kj} = j$, тогда считаем, что весь путь определен, так как найдены все промежуточные узлы. В противном случае повторяем описанную процедуру для путей от узла i к узлу k и от узла k к узлу j.

Теперь рассмотрим реализацию алгоритма на примере.

Пусть для сети, показанной на рисунке 3, требуется найти кратчайшие пути между любыми двумя узлами. Расстояние между узлами этой сети проставлены на рисунке возле соответствующих ребер. Ребро *(3, 5)* ориентированно, поэтому не допускается движение от узла *5* к узлу *3*. Все остальные ребра позволяют движение в обе стороны:

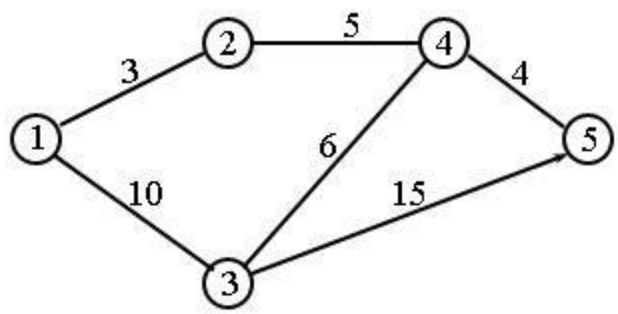


Рис. 3. Пример сети

<u>*Шаг 0*</u>. Начальные матрицы *D0* и *S0* строятся непосредственно по заданной схеме сети. Матрица *D0* симметрична, за исключением пары элементов *d35* и *d53*, где *d53* равно бесконечности, поскольку невозможен переход от узла *5* к узлу *3*:

D0

	1	2	3	4	5
1	—	3	10	∞	∞
2	3	—	∞	5	∞
3	10	∞	—	6	15
4	∞	5	6	—	4
5	∞	∞	∞	4	—

S0

	1	2	3	4	5
1	—	2	3	4	5
2	1	—	3	4	5
3	1	2	—	4	5
4	1	2	3	—	5
5	1	2	3	4	—

Рис. 4. Начальное состояние

<u>*Шаг 1*</u>. В матрице D_0 выделены ведущие строка и столбец $(k = 1)$. Двойной рамкой представлены элементы d_{23} и d_{32}, единственные среди элементов матрицы D_0, значения которых можно улучшить с помощью треугольного оператора. Таким образом, чтобы на основе матриц D_0 и S_0 получить матрицы D_1 и S_1, выполняем следующие действия:

- Заменяем d_{23} на $d_{21} + d_{13} = 3 + 10 = 13$ и устанавливаем $s_{23} = 1$.
- Заменяем d_{32} на $d_{31} + d_{12} = 10 + 3 = 13$ и устанавливаем $s_{32} = 1$.

Матрицы D_1 и S_1 имеют следующий вид:

D1

	1	2	3	4	5
1	—	3	10	∞	∞
2	3	—	**13**	5	∞
3	10	**13**	—	6	15
4	∞	5	6	—	4
5	∞	∞	∞	4	—

S1

	1	2	3	4	5
1	—	2	3	4	5
2	1	—	**1**	4	5
3	1	**1**	—	4	5
4	1	2	3	—	5
5	1	2	3	4	—

Рис. 5. Матрицы D_1 и S_1

<u>*Шаг 2.*</u> Полагаем $k = 2$; в матрице D_1 выделены ведущие строка и столбец. Треугольный оператор применяется к элементам матрицы D_1 и S_1, выделенным двойной рамкой. В результате получаем матрицы D_2 и S_2:

D2

	1	2	3	4	5
1	—	3	10	**8**	∞
2	3	—	13	5	∞
3	10	13	—	6	15
4	**8**	5	6	—	4
5	∞	∞	∞	4	—

S2

	1	2	3	4	5
1	—	2	3	**2**	5
2	1	—	1	4	5
3	1	1	—	4	5
4	**2**	2	3	—	5
5	1	2	3	4	—

Рис. 6. Матрицы D_2 и S_2

<u>*Шаг 3.*</u> Полагаем $k = 3$; в матрице D_2 выделены ведущие строка и столбец. Треугольный оператор применяется к элементам матрицы D_2 и S_2, выделенным двойной рамкой. В результате получаем матрицы D_3 и S_3:

D3

	1	2	3	4	5
1	—	3	10	8	**25**
2	3	—	13	5	**28**
3	10	13	—	6	15
4	8	5	6	—	4
5	∞	∞	∞	4	—

S3

	1	2	3	4	5
1	—	2	3	2	**3**
2	1	—	1	4	**3**
3	1	1	—	4	5
4	2	2	3	—	5
5	1	2	3	4	—

Рис. 7. Матрицы D_3 и S_3

<u>*Шаг 4.*</u> Полагаем $k = 4$, ведущие строка и столбец в матрице D_3 выделены. Получаем новые матрицы D_4 и S_4:

D4

	1	2	3	4	5
1	—	3	10	8	**12**
2	3	—	**11**	5	**9**
3	10	**11**	—	6	**10**
4	8	5	6	—	4
5	**12**	**9**	**10**	4	—

S4

	1	2	3	4	5
1	—	2	3	2	**4**
2	1	—	**4**	4	**4**
3	1	**4**	—	4	**4**
4	**2**	2	3	—	5
5	**4**	**4**	**4**	4	—

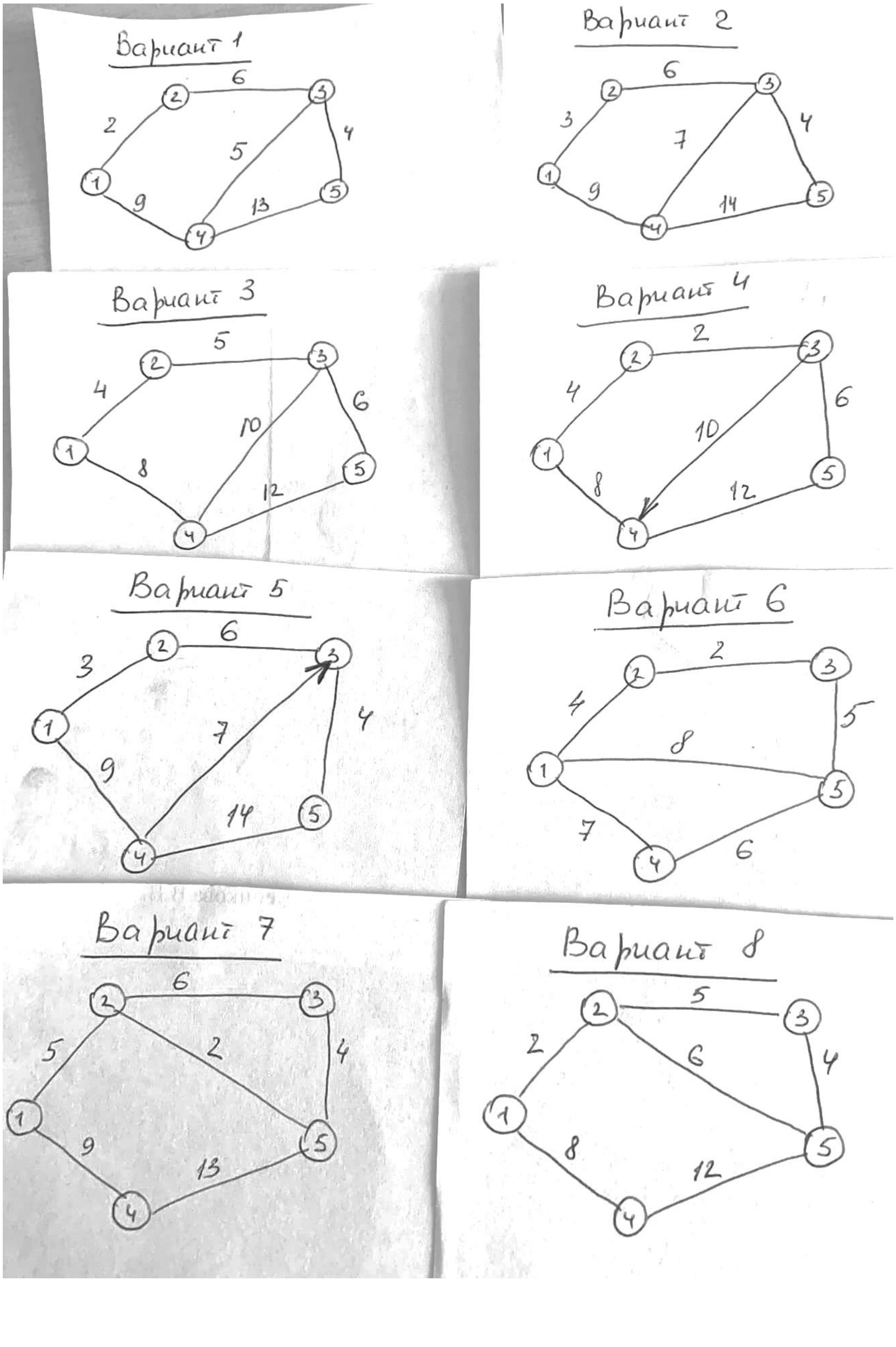
Рис. 8. Матрицы D_4 и S_4

Шаг 5. Полагаем $k = 5$, ведущие строка и столбец в матрице D_4 выделены. Никаких действий на этом шаге не выполняем; вычисления закончены.

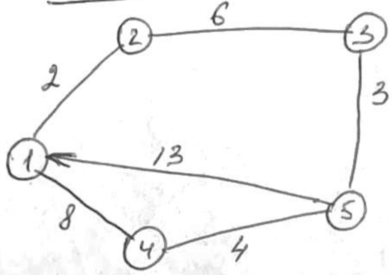
Конечные матрицы D_4 и S_4 содержат всю информацию, необходимую для определения кратчайших путей между любыми двумя узлами сети. Например, кратчайшее расстояние между узлами *1* и *5* равно $d_{15} = 12$.

Для нахождения соответствующих маршрутов напомним, что сегмент маршрута (i, j) состоит из ребра (i, j) только в том случае, когда $s_{ij} = j$. В противном случае узлы i и j связаны, по крайней мере, через один промежуточный узел. Например, поскольку $s_{15} = 4$ и $s_{45} = 5$, сначала кратчайший маршрут между узлами *1* и *5* будет иметь вид $1 \to 4 \to 5$. Но так как s_{14} не равно *4*, узлы *1* и *4* в определенном пути не связаны одним ребром (но в исходной сети они могут быть связаны непосредственно). Далее следует определить промежуточный узел (узлы) между первым и четвертым узлами. Имеем $s_{14} = 2$ и $s_{24} = 4$, поэтому маршрут $1 \to 4$ заменяем $1 \to 2 \to 4$. Поскольку $s_{12} = 2$ и $s_{24} = 4$, других промежуточных узлов нет. Комбинируя определенные сегменты маршрута, окончательно получаем следующий кратчайший путь от узла *1* до узла *5*: $1 \to 2 \to 4 \to 5$. Длина этого пути равна *12* километрам.

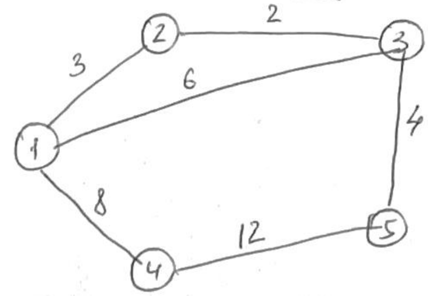
Исходные данные к работе



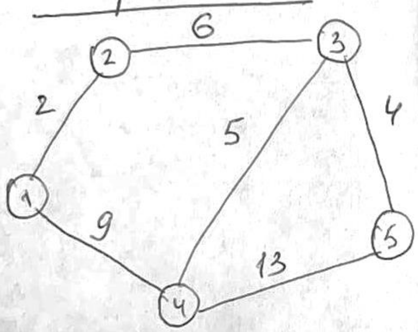
Вариант 9



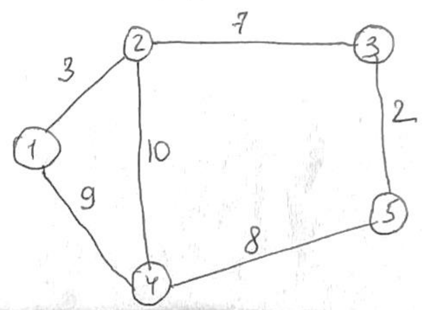
Вариант 10



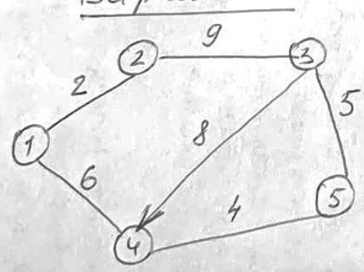
Вариант 11



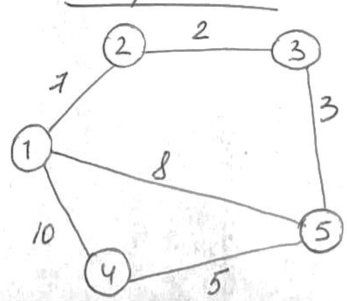
Вариант 12



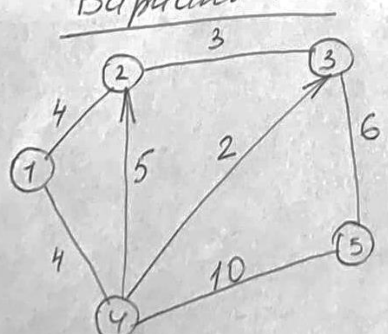
Вариант 13



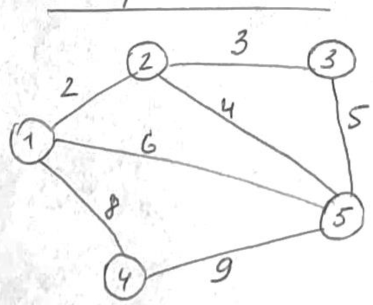
Вариант 14



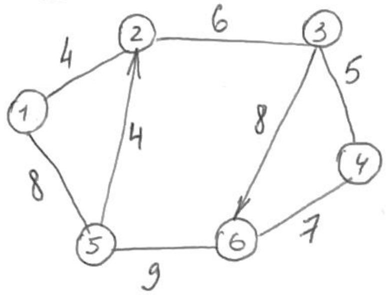
Вариант 15



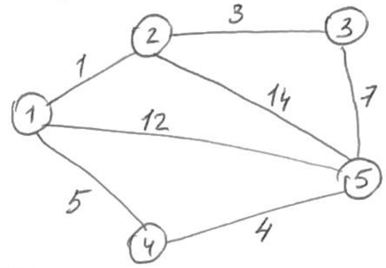
Вариант 16



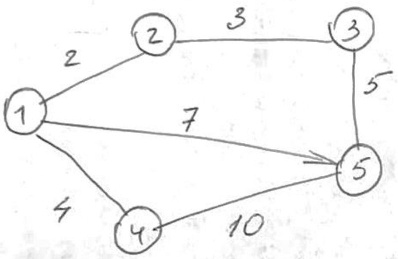
Вариант 17



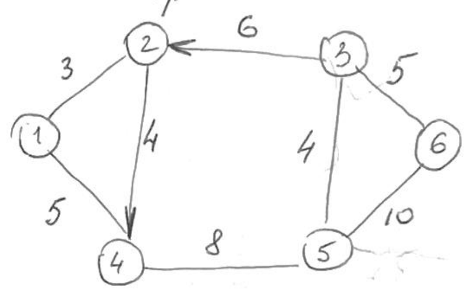
Вариант 18



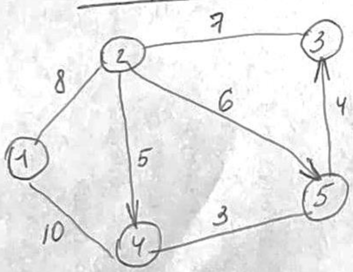
Вариант 19



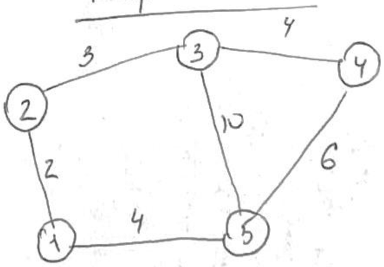
Вариант 20



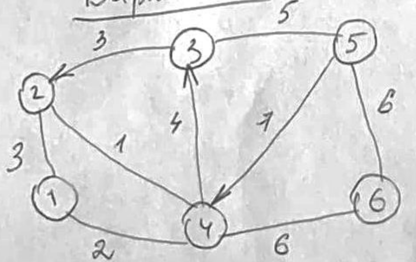
Вариант 22



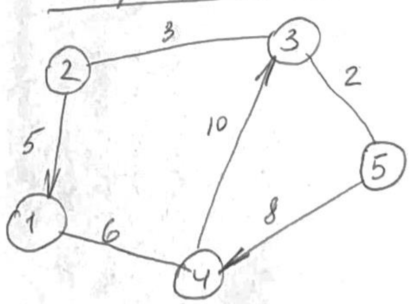
Вариант 23

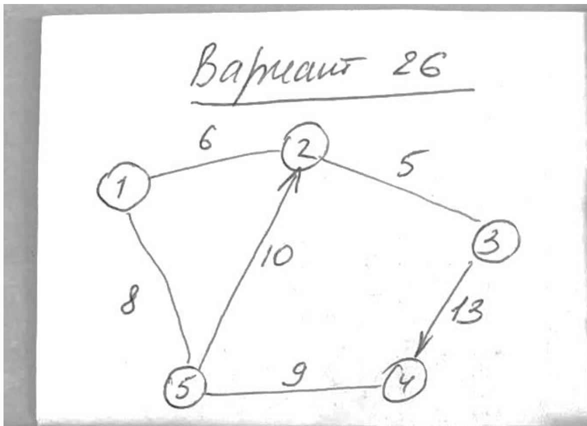


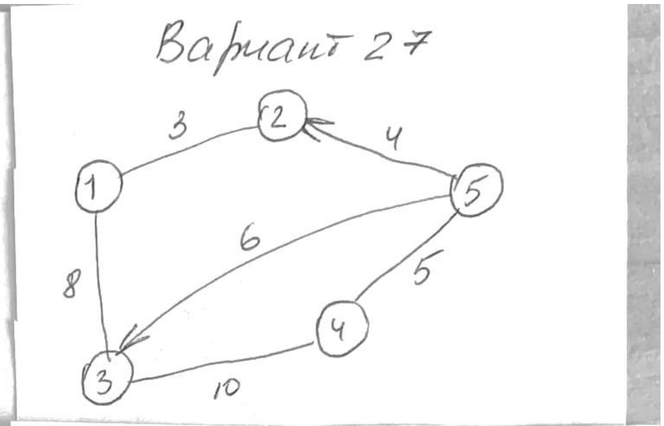
Вариант 24

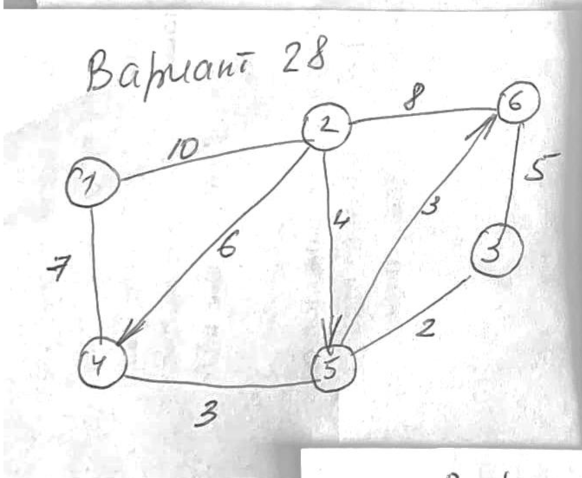


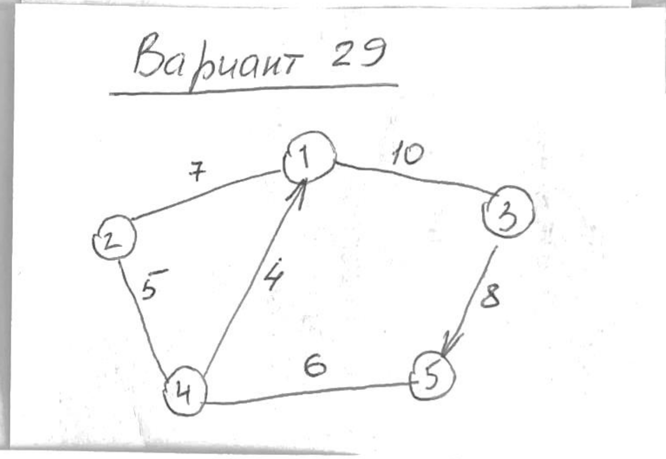
Вариант 25

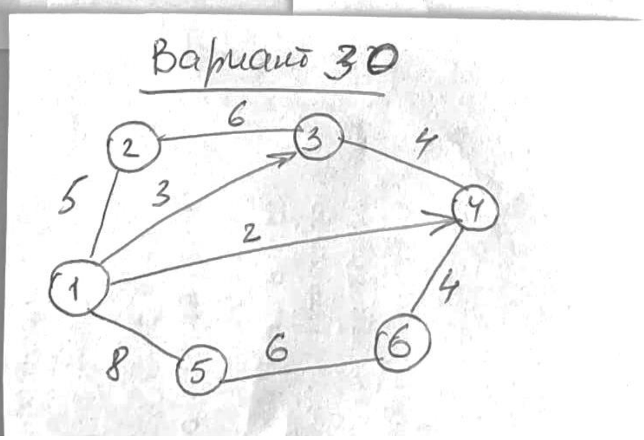












Порядок выполнения работы:

1. Получить исходные данные к работе.
2. Построить сеть связи.
3. Используя метод Флойда, определить кратчайшие расстояния в сети.

Содержание отчета:

1. Цель работы.
2. Ручной расчет сети.
3. Блок-схема алгоритма, расчет на ЭВМ.

ЛАБОРАТОРНАЯ РАБОТА №3

МАКСИМИЗАЦИЯ ПОТОКА В СЕТЯХ СВЯЗИ

3.1 Цель работы

Изучить алгоритм максимизации потока на сети связи

В данной работе рассматривается задача определения максимального потока между двумя выделенными узлами связной сети. Каждая дуга сети обладает пропускными способностями в обоих направлениях, которые определяют максимальное количество потока, проходящего по данной дуге. Ориентированная (односторонняя) дуга соответствует нулевой пропускной способности в запрещенном направлении.

Пропускные способности c_{ij} сети можно представить в матричной форме. Для определения максимального потока из источника s в сток t применяются следующие шаги.

Шаг 1. Найти цепь, соединяющую s с t, по которой поток принимает положительное значение в направлении $s \to t$. Если такой цепи не существует, перейти к шагу 3. В противном случае перейти к шагу 2.

Шаг 2. Пусть c_{ij}^{-} (c_{ij}^{+}) — пропускные способности дуг цепи (s,t) в направлении $s \to t$ $(t \to s)$ и

$$\theta = \min\left\{c_{ij}^{-}\right\} > 0 .$$

Матрицу пропускных способностей $\left(c_{ij}\right)$ изменить следующим образом:

а) вычесть θ из всех c_{ij}^{-};

б) прибавить θ ко всем c_{ij}^{+}.

Заменить текущую c_{ij}^{-} матрицу на вновь полученную и перейти к шагу 1.

Операция «а» дает возможность использовать остатки пропускных способностей дуг выбранной цепи в направлении $s \to t$. Операция «б» восстанавливает исходные пропускные способности сети, поскольку уменьшение пропускной способности дуги в одном направлении можно рассматривать как увеличение ее пропускной способности в противоположном направлении.

Шаг 3. Найти максимальный поток в сети. Пусть $C = \left\| c_{ij} \right\|$ — *исходная* матрица пропускных способностей, и пусть $C^* = \left\| c_{ij}{}^* \right\|$ — *последняя* матрица, получившаяся в результате модификации исходной матрицы (шаги 1 и 2). Оптимальный поток $X = \left\| x_{ij} \right\|$ в дугах задается как

$$x_{ij} = \begin{cases} c_{ij} - c_{ij}{}^*, & c_{ij} > c_{ij}{}^* \\ 0, & c_{ij} \le c_{ij}{}^* \end{cases}.$$

Максимальный поток из s в t равен

$$z = \sum_i x_{si} = \sum_j x_{tj} .$$

Заметим, что z есть сумма всех положительных 0, определенных на шаге 2. Таким образом, можно объяснить, почему используются положительные элементы матрицы $C - C^*$ для определения результирующего потока в направлении $s \to t$.

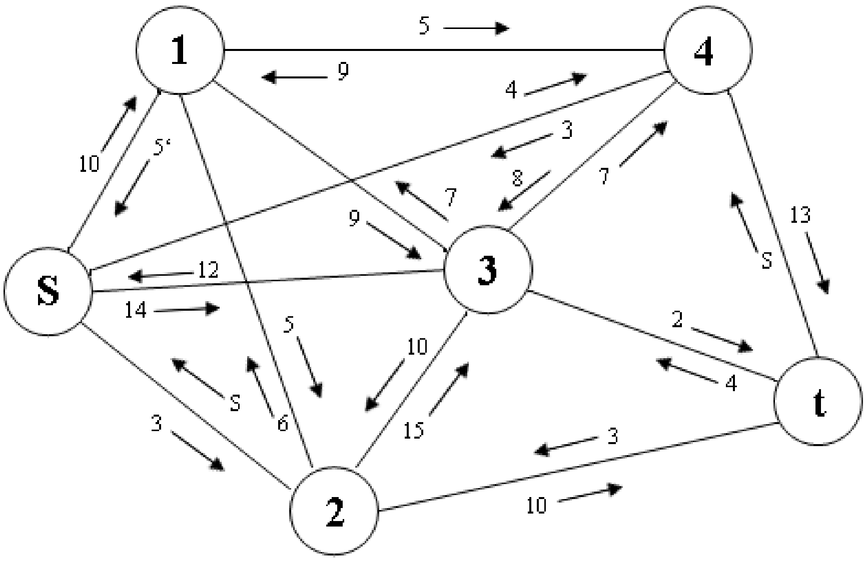


Рисунок 3.1

Пример. Рассмотрим сеть на рисунке 3.1 с данными пропускными способностями. Соответствующая матрица пропускных способностей C приведена в таблице 3.1.

Таблица 3. 1

C=

	s	1	2	3	4	t
S		10-	3	14	4	
1	5+		5	9	5-	
2	5	6		15		10
3	12	7	10		7	2
4	3	9+		8		13-
T			3	4	5+	

Цепь $[s \to 1 \to 4 \to t]$ $\quad \theta = \min\{10,5,13\}$.

В качестве исходной цепи можно выбрать $s \to 1 \to 4 \to t$. Таким образом, ячейки $(s,1)$, $(1,4)$ и $(4,t)$ помечаются знаком (–), ячейки $(1,s)$, $(4,1)$ и $(t,4)$ – знаком (+). Для данной цепи максимальный поток определяется как

$$\theta = \min\left\{c_{s1}, c_{14}, c_{4t}\right\} = \min\{10, 5, 13\} = 5.$$

Заметим, что можно выбирать различные исходные цепи. Очевидно, что хороший выбор (вначале и на каждой итерации) должен давать наибольшее значение θ. Однако при этом, возможно, понадобится перебрать несколько вариантов, что в конечном итоге оказывается малоэффективным. При программировании алгоритма цепь удобно определять непосредственно из матрицы C, начиная с первой строки (*s*-строки) и выбирая следующий узел среди тех, которые соединены с s положительным потоком. Далее рассматривается строка, соответствующая выбранному узлу, и выбирается следующий узел, соединенный с предыдущим положительной дугой. Процесс продолжается до тех пор, пока не будет достигнут узел t.

Матрица C в таблице 3.1 корректируется путем вычитания $\theta = 5$ из всех элементов, помеченных знаком (–), и сложения со всеми элементами, имеющими знак (+). Результаты приведены в таблице 3.2.

Таблица 3.2

	S	1	2	3	4	t
S		5	3	14-	4	
1	10		5	9	0	
2	5	6		15+		10-
3	12+	7	10-		7	2
4	3	14		8		8
T			3+	4	10	

Цепь $[s \to 3 \to 2 \to t]$ $\qquad \theta = \min\{14,10,10\} = 10$.

Результаты последующих итераций приведены в таблицах 3.3—3.6.

Таблица 3.3

	S	1	2	3	4	T
S		5-	3	4	4	
1	10+		5	9-	0	
2	5	6		25		0
3	22	7+	0		7-	2
4	3	14		8+		8-
T			13	4	10+	

Цепь $\left[s \to 1 \to 3 \to 4 \to t\right]$ $\theta = \min\{5,9,7,8\} = 5$.

Таблица 3.4

	S	1	2	3	4	T
S		0	3	4	4-	
1	15		5	4	0	
2	5	6		25		0
3	22	7+	0		2	2-
4	3+	14		13		3-
T			13	4	15+	

Цепь $\left[s \to 4 \to t\right]$ $\theta = \min\{4,3\} = 3$.

Таблица 3.5

	S	1	2	3	4	T
S		0	3	4-	1	
1	15		5	4	0	
2	5	6		25		0
3	22+	12	0		2	2-
4	6	14		13		0
T			13	4	18+	

Цепь $\left[s \to 3 \to t\right]$ $\theta = \min\{4,2\} = 2$.

Таблица 3.6

	s	1	2	3	4	T
S		0	3	2	1	
1	15		5	4	0	
2	5	6		25		0
3	24	12	0		2	0
4	6	14		13		0
T			13	4	20	

(Между s и t нельзя построить цепь)

Из таблицы 3.6 следует, что между s и t нельзя построить цепей с положительным потоком, поскольку все элементы в столбце t равны нулю. Таким образом, таблица 3.6 дает матрицу C^*.

В таблицах 3.1 (матрица C) и 3.6 (матрица C^*) приведены данные, характеризующие оптимальный поток, которые получаются вычислением $X = C - C^*$ и заменой отрицательных величин нулями. Таблица 3.7 дает матрицу X.

Таблица 3.7

X=

	s	1	2	3	4	T
s		10		12	3	
1				5	5	
2						10
3			10		5	2
4						13
t						

Из таблицы 3.7 видно, что

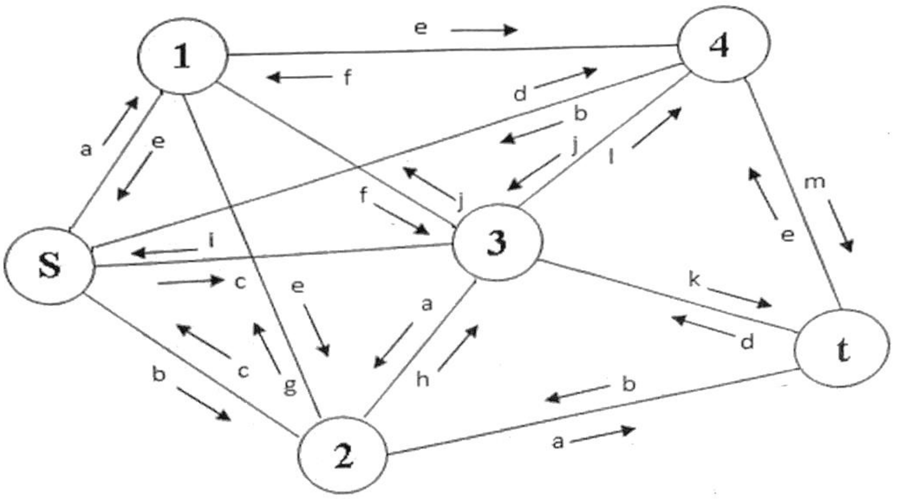
$$z = \sum x_{si} = 10 + 12 + 3 = \sum x_{ti} = 10 + 2 + 13 = 25.$$

Сумма всех $\theta = (5 + 10 + 5 + 3 + 2 = 25)$ также дает максимальный поток. Графически решение представлено на рисунке 3.2. Здесь уместно ввести понятие **минимального разреза. Разрез** в связанной сети представляет собой такое множество дуг, которое определяет нулевой поток из s в t, если пропускные способности этих дуг полагаются равными нулю. Пропускная способность разреза равна сумме пропускных способностей его дуг. В сети на рисунке 3. 2 можно выделить следующие разрезы:

Разрез	*Пропускная способность*
(s, 1), (s, 2), (s, 3), (s, 4) (4, t), (3, t), (2, t) (1, 4), (s, 4), (3, 4), (3, t), (2, t)	10 + 3+14 + 4 = 31 13 + 2+10 = 25 5+4 + 7+2+10 = 28

Интуитивно очевидно, что максимальный поток можно найти, перебирая *все* разрезы сети. Разрез минимальной пропускной способности даст решение. Это интуитивное соображение на самом деле можно доказать, используя **теорему о максимальном потоке — минимальном разрезе,** согласно которой максимальный поток в сети равен пропускной способности минимального разреза.

3.2 Исходные данные к работе



Вариант	a	b	c	d	e	f	g	h	i	j	k	l	m
1	6	1	12	4	10	12	9	18	9	4	5	11	10
2	8	1	12	3	8	11	8	17	10	5	4	10	11
3	9	2	13	3	6	10	7	16	11	6	3	9	12
4	10	3	14	4	5	9	6	15	12	7	2	8	13
5	11	4	15	5	4	8	5	14	13	8	1	7	14
6	2	5	17	6	3	7	4	13	14	9	15	6	8
7	3	6	19	7	2	6	3	12	15	10	29	5	2
8	4	7	47	8	2	5	2	11	16	11	43	4	4
9	5	8	75	9	2	4	1	10	17	12	57	3	6
10	6	9	103	10	2	3	0	9	18	13	71	2	8
11	7	10	131	11	2	2	1	8	19	14	85	1	10
12	8	11	159	12	2	1	2	7	20	15	99	0	12
13	9	12	187	13	2	0	3	6	21	16	113	1	14
14	10	13	215	14	2	1	4	5	22	17	127	2	16
15	11	14	243	15	2	2	5	4	23	18	141	3	18
16	12	15	271	16	2	3	6	3	24	19	155	4	20
17	13	16	299	17	2	4	7	2	25	20	169	5	22
18	14	17	327	18	2	5	8	1	26	21	183	6	24
19	15	18	355	19	2	6	9	2	27	22	197	7	26
20	16	19	383	20	2	7	10	3	28	23	211	8	28
21	17	20	411	21	2	8	11	4	29	24	225	9	30
22	18	21	439	22	2	9	12	5	30	25	239	10	32
23	19	22	467	23	2	10	13	6	31	26	253	11	34
24	20	23	495	24	2	11	14	7	32	27	267	12	36
25	21	24	523	25	2	12	15	8	33	28	281	13	38
26	22	25	551	26	2	13	16	9	34	29	295	14	40
27	23	26	579	27	2	14	17	10	35	30	309	15	42
28	24	27	607	28	2	15	18	11	36	31	323	16	44
29	25	28	635	29	2	16	19	12	37	32	337	17	46
30	26	29	663	30	2	17	20	13	38	33	351	18	48

3.3 Порядок выполнения работы:

1.Получить исходные данные к работе.

2.Построить сеть связи.

3. Определить максимальный поток на сети между узлами $s \to t$.

3.4 Содержание отчета:

1. Цель работы.
2. Ручной расчет сети.
3. Блок-схема алгоритма, расчет на ЭВМ.

ЛАБОРАТОРНАЯ РАБОТА №4

РЕЗЕРВИРОВАНИЕ НА СЕТЯХ СВЯЗИ

4.1 Цель работы

Изучить методы резервирования на сетях связи.

4.2 Общие сведения

Надежность – это свойство системы (элемента) выполнить заданные функции при определенных условиях эксплуатации. Для реализации системой (элементом) своих функций с требуемым качеством необходимо, чтобы их основные параметры не выходили за установленные пределы. К основным параметрам относятся те количественные показатели, которые определяют выполнение рабочих функций. Иногда для большей определенности и конкретизации различают следующие разновидности надежности: эксплуатационную и техническую (номинальную).

Под эксплуатационной надежностью понимается надежность, определяемая в реальных условиях эксплуатации с учетом комплексного воздействия внешних и внутренних факторов, связанных с климатическими и географическими особенностями эксплуатации, реальными режимами работы системы и условиями ее обслуживания.

Под технической (номинальной) надежностью понимается надежность, определяемая путем испытания в заводских условиях при работе аппаратуры в соответствии с типовыми режимами, оговоренными в технических условиях.

Количественные характеристики надежности описываются показателями. Показатель надежности – это мера, посредством которой производится количественная оценка. Численное значение какого-либо показателя для конкретной системы иногда называют параметром надежности. К параметрам надежности предъявляются следующие основные требования:

– максимальный учет факторов, определяющих надежность аппаратуры;

– возможность использования показателей при инженерных расчетах надежности;

– возможность задания показателей надежности в качестве технических параметров проектируемой аппаратуры;

– удобство и быстрота практической проверки показателей в процессе эксплуатации или специальных испытаний.

Для полной количественной характеристики основных сторон надежности используются различные показатели, которые удобно разделить на несколько групп.

К показателям безотказности относятся: вероятность безотказной работы; частота отказов; интенсивность отказов; среднее время безотказной работы; наработка на отказ (среднее время работы между отказами).

Первые четыре показателя используются главным образом для оценки надежности невосстанавливаемых изделий. Однако они могут применяться и при оценке надежности восстанавливаемых изделий до появления первого отказа. Пятый показатель имеет смысл только по отношению к восстанавливаемым изделиям.

Показателями восстанавливаемости являются: вероятность восстановления; среднее время восстановления; интенсивность восстановления.

Показателями технического обслуживания являются: вероятность обслуживания; среднее время обслуживания.

К эксплуатационным коэффициентам надежности относятся: коэффициент использования или коэффициент исправного действия $К_И$; коэффициенты готовности $К_Г$ и оперативной готовности $К_{ОГ}$, а также коэффициенты простоя и стоимости эксплуатации.

Рассмотрим наиболее важные показатели надежности и выясним связь между ними.

Одним из распространенных количественных показателей надежности является вероятность безотказной работы элемента $p(t)$ или системы $P(t)$ за определенный промежуток времени. Вероятность безотказной работы – это вероятность того, что за заданный интервал времени не произойдет ни одного отказа.

Вероятность безотказной работы элемента можно представить, как вероятность того, что время исправной работы будет больше некоторого заданного времени:

$$p(t) = P\{T > t\}$$

Практическая вероятность безотказной работы за некоторый промежуток времени может быть найдена по результатам испытаний элементов на надежность как отношение числа элементов, оставшихся исправными в конце рассматриваемого интервала времени t_i к начальному числу элементов, поставленных на испытание:

$$P_i^* = (N - n_i)/N ,$$

где N - начальное число испытываемых элементов;
n_i - число отказавших элементов за время t_i.

При значительном числе испытываемых элементов статистическая вероятность P_i^* сходится по вероятности к $p(t)$.

Вероятность отказа элемента $q(t)$ связана с $p(t)$ соотношением

$$q(t) = 1 - p(t) = P\{T \leq t\}.$$

Статистическое значение вероятности отказа равно отношению числа отказавших элементов за рассматриваемый промежуток времени к начальному числу испытываемых элементов:

$$q_i^* = 1 - p_i^* = n_i/N.$$

Для системы, состоящей из ряда последовательно соединенных элементов, вероятность безотказной работы может быть представлена в виде произведения вероятностей безотказной работы всех элементов:

$$P(t) = p_1(t)p_2(t) \dots p_N(t) = \prod_{i=1}^{N} p_i(t)$$

Под вероятностью отказа системы $\theta(t)$ понимается вероятность того, что за заданный интервал произойдет отказ, т. е. время исправной работы системы будет меньше заданного. Следовательно, по аналогии с вероятностью отказа элемента $\theta(t)$ является функцией распределения, или интегральным законом распределения времени исправной работы системы.

Так как система может находиться либо в исправном состоянии, либо в состоянии отказа, то сумма вероятностей безотказной работы $P(t)$ и повреждения $\theta(t)$ всегда равна единице. На основании этого вероятность отказа системы:

$$\theta(t) = 1 - P(t) = 1 - p_1(t)p_2(t) \dots p_N(t).$$

Выражая $\theta(t)$ через вероятность отказа элементов, получаем:

$$\theta(t) = 1 - [1 - q_1(t)][1 - q_2(t)] \dots [1 - q_N(t)].$$

Под частотой отказов понимают число отказов в единицу времени, отнесенное к первоначальному числу поставленных на испытание элементов. Если в процессе испытаний на надежность N элементов фиксировать число отказов Δn_i, происшедших в определенные, интервалы Δt_i, то частота отказов в данный промежуток времени определится как:

$$f_i^* = \Delta n_i / N \Delta t_i.$$

Показателем, наиболее полно характеризующим надежность невосстанавливаемых элементов, является интенсивность отказов $\lambda(t)$. В отличие от частоты отказов $f(t)$ этот показатель характеризует степень надежности элемента в каждый данный момент времени, т. е. его локальную надежность. Введение этого показателя надежности оказалось целесообразным также и по соображениям удобства расчета надежности систем по известным значениям интенсивностей отказов элементов, так как получаемые при этом расчетные соотношения являются сравнительно простыми и удобными для инженерной практики.

Под интенсивностью отказов понимают число отказов в единицу времени, отнесенных к числу элементов, оставшихся исправными к началу рассматриваемого промежутка времени. Как и частота отказов, эта характеристика надежности может быть получена из опытных данных и рассчитывается по формуле:

$$\lambda_i^* = \Delta n_i/(N - n_i \Delta t_i),$$

где $\Delta n_i -$ число отказов за промежуток Δt_i;

N — начальное число элементов;

$n_i -$ общее число отказавших элементов к началу рассматриваемого промежутка времени.

Зависимость λ_i^* от t представляет собой функцию интенсивности отказов $\lambda(t)$.

Интенсивность отказов связана однозначной зависимостью с частотой отказов и вероятностью безотказной работы:

$$\lambda(t)= f(t)\ p(t).$$

Под интенсивностью отказов восстанавливаемой системы, состоящей из разнородных по надежности элементов, будем понимать число отказов системы в единицу времени:

$$\Lambda(t)=\mathrm{n}/\Delta t.$$

Надежность однотипных систем и элементов с точки зрения продолжительности их работы до первого отказа можно оценивать средним временем безотказной работы, под которым понимается математическое ожидание времени исправной работы. Среднее время безотказной работы однотипных элементов определяется по данным испытаний элементов на надежность по формуле:

$$T_{\text{ср}}^* = \sum_{i=1}^{N} t_i/N,$$

где t_i - время исправной работы i-го элемента;

N - общее число испытываемых элементов.

Наработка на отказ – это среднее число часов работы между двумя соседними отказами. Таким образом, если аппаратура определенного типа проработала суммарное время T_p часов за определенный календарный срок и имела при этом n отказов в работе, то наработка на отказ рассчитывается по формуле:

$$T_0^* = T_p/n$$

Под восстанавливаемостью принято понимать свойство системы восстанавливать свою работоспособность после возникновения отказа с учетом качества обслуживания.

Количественную оценку восстанавливаемости можно оценить по следующим критериям: вероятности восстановления $v(t)$, среднему времени восстановления T_B и интенсивности восстановления $\mu(t)$, которые в математическом смысле аналогичны рассмотренным критериям надежности системы: $\theta(t)$, T_0 и $\Lambda(t)$.

Под вероятностью восстановления понимается вероятность того, что система будет восстановлена после отказа за заданное время и в определенных условиях ремонта. По аналогии с вероятностью отказа этот критерий можно представить как вероятность того, что случайное время восстановления системы T будет не больше заданного:

$$v(t) = P\{t \leq T\}.$$

Отсюда следует, что $v(t)$ – функция распределения, или интегральный закон распределения времени восстановления.

Среднее время восстановления T_B – это математическое ожидание случайной величины – времени восстановления. Если за определенный период эксплуатации аппаратуры произошло n отказов, то, просуммировав промежутки времени τ_i, затраченного на обнаружение и устранение отказов, и разделив эту сумму на число восстановлений, равное числу отказов, получим величину среднего времени восстановления:

$$T_B^* = \sum_{i=1}^{n} \tau_i / n$$

Под интенсивностью восстановлений системы понимается число восстановлений, произведенное в единицу времени. В случае экспоненциального закона распределения интенсивность восстановления статистически может быть определена как отношение числа восстановлений системы за некоторый период времени к суммарному времени восстановления за тот же период:

$$\mu^* = n / \sum_{i=1}^{n} \tau_i$$

Если известны структура оборудования системы передачи (СП), принципы его функционирования и восстановления работоспособности, то, задавшись определенными критериями отказа, все состояния оборудования можно разделить на два класса: работоспособное – использование только по назначению и неработоспособное, т. е. отказ или нахождение на плановом (неплановом) техническом обслуживании или ремонте и т. д. Если известны интенсивности отказов (λ), поступления оборудования на техническое обслуживание или ремонт (v_{TO}) и восстановления (μ_B), то нахождение оборудования в одном из состояний может быть охарактеризовано рядом показателей, основными из которых являются:

1) коэффициент готовности

$$К_Г = \frac{Т_0}{(Т_0+Т_В)} = \mu/(\mu+\lambda),$$

характеризующий вероятность исправного состояния оборудования в установившемся режиме эксплуатации;

2) коэффициент простоя

$$К_П = 1 - К_г = \frac{Т_В}{(Т_0+Т_В)} = \lambda/(\mu+\lambda);$$

3) коэффициент исправного действия

$$К_К = (Т_К - Т_П)/Т_К\,,$$

где $Т_К$ — календарный цикл эксплуатации оборудования;

$Т_П$ — суммарное время простоя оборудования СП, каналов и трактов за рассматриваемый период эксплуатации.

Вероятность пребывания оборудования СП, каналов и трактов в работоспособном состоянии, т. е. готовность его использования по функциональному назначению, определяется коэффициентом технического использования, под которым понимается отношение вида:

$$К_{ТИ} = \mu \cdot \mu_{ТО}/(\lambda \cdot \mu_{ТО} + \mu \cdot \mu_{ТО} + \mu \cdot \nu_{ТО}).$$

С учетом того, что $\mu = 1/T_B$, $\lambda = 1/T_0$, $\mu_{ТО} = 1/Т_{ТО}$, где $Т_{ТО}$ – период технического обслуживания или ремонта, $\nu_{ТО} = 1/\tau_0$, где τ_0 – время технического обслуживания или ремонта, приводится к виду:

$$К_{ТИ} = Т_0/[Т_0 + Т_В + Т_0(Т_0/\tau_{ТО})].$$

Сложные связные устройства, такие как линии связи, станции, комплексы, стойки, всегда можно представить в виде более мелких структурных подразделений (элементов): линии – в виде отдельных участков, станции – в виде отдельных стоек, стойки – в виде отдельных блоков и т.д.

В каждом объекте связи можно выделить элементы, работоспособность которых необходима для работоспособности объекта в целом. В общем случае элементы обладают различными надежностями и стоимостями. Очень часто надежность элементов такова, что надежность объекта в целом оказывается недостаточной. В этом случае применяется резервирование объекта в целом (общее резервирование) или отдельных элементов (поэлементное резервирование). При этом возникает задача нахождения количества и вида резервных элементов. Часто применяется стопроцентное резервирование всех элементов. Это означает, что резервируются и мало-, и высоконадежные , и дорогие, и дешевые элементы системы. Ясно, что средства, затрачиваемые на такое резервирование, будут использоваться не оптимально.

Целесообразнее в большем объеме резервировать малонадежные и дешевые элементы. Такое резервирование и реализуется при оптимизации структуры резерва, в результате которой обеспечивается требуемая надежность при минимальных затратах на резервные элементы ли наибольшая надежность при заданной величине затрат.

При решении задачи оптимизации резервирования введем следующие предположения:

1) Основной и заменяющий его резервный элемент однотипны, имеют одинаковую стоимость и надежность.
2) Переход на резерв осуществляется практически мгновенно и переключающие устройства абсолютно надежны или их надежность учтена в ненадежности самих элементов.

Надежность объектов характеризуется следующими показателями:

$r(t)$ – вероятность безотказной работы объекта (вероятность того, что объект, бывший работоспособным и начавший работать в момент $t = 0$, проработает безотказно до момента t);

$q(t)$ – вероятность отказов (вероятность того, что объект, работоспособный в момент $t = 0$, до момента времени t, откажет). Поскольку пребывание в состояниях отказа и работоспособности – события противоположные, то $r(t) = 1 - q(t)$.

Для оптимизации структуры резерва объект связи представляется схемой надежности (рис.1), в которой каждый из n блоков должен быть работоспособным для работоспособности всего объекта связи, причем предполагается, что для i-го блока известны его показатели надежности $r_i(t)$ или $q_i(t)$ и стоимость C_i.

Надежность нерезервированного объекта связи равна

$$R = \prod_{i=1}^{n} r_i(t) = \prod_{i=1}^{n} \big(1 - q_i(t)\big)$$

Если блоки объекта связи обладают очень высокой надежностью (что почти всегда выполняется), то $q_i \ll 1$ и надежность всего объекта R можно представить более простым выражением

$$R = \prod_{i=1}^{n} \big(1 - q_i(t)\big) \approx 1 - \sum_{i=1}^{n} q_i = 1 - Q\,, \qquad Q = \sum_{i=1}^{n} q_i\,,$$

где Q – ненадежность объекта связи.

Затраты C на весь объект равны сумме затрат на отдельные блоки

$$C = \sum_{i=1}^{n} C_i$$

Схема резервированного объекта связи состоит из групп: первый блок имеет ×1 резервных блоков, второй - × 2 и т.д.

i-ый рабочий блок и его × i резервных блоков образуют i-ю подсистему с надежностью $R_i(x_i)$ и стоимостью резерва $C_i(x_i)$.

Для работоспособности системы (объекта связи) необходимо, чтобы были работоспособны одновременно все подсистемы, т.е. последовательное соединение подсистем. Следовательно, надежность R системы

$$R = \prod_{i=1}^{n} R_i(x_i) \,,$$

а стоимость C резерва (без учета стоимости рабочих блоков)

$$C = \sum_{i=1}^{n} C_i(x_i) = \sum_{i=1}^{n} C_i * x_i$$

Для работоспособности i-ой подсистемы необходимо, чтобы работал хотя бы один из $(x_i + 1)$ блоков этой подсистемы, что обозначается в схеме параллельным соединением блоков. Ненадежность $Q_i(x_i)$ i-ой подсистемы при этом оказывается равной

$$Q_{i(xi)} = q_i^{x_i+1} = (1 - r_i)^{x_i+1} \,,$$

а надежность

$$R_i(x_i) = 1 - Q_i(x_i) = 1 - (1 - r_i)^{x_i+1}$$

Задачи оптимизации структуры заключается в определении оптимального состава резерва, т.е. в нахождении совокупности неотрицательных чисел ×10, ×20, ×30, …, ×n_0, образующих оптимальный вектор X_0 состава резерва и характеризующих оптимальное количество резервных элементов для каждого рабочего блока.

При этом возможна прямая и обратная задачи оптимизации.

В первом случае задача заключается в том, чтобы найти такой состав X_0 резерва, при котором стоимость резерва минимальна и $R(X_0) \geq R_0$, где R_0 – требуемая величина надежности.

Обратная задача оптимизации заключается в нахождении такого состава резерва, при котором надежность системы $R(X_0)$ максимальна, а стоимость резерва равна заданной величине C_0.

Рассмотренные задачи оптимизации структуры резерва могут решаться различными способами. Наиболее простым методом оптимизации является метод перебора. Метод заключается в переборе всех возможных значений состава резерва и нахождении при этом соответствующих значений надежности системы и стоимости резерва. Этот метод прост по своей сути, дает точное решение, но чрезвычайно трудоемок и применим для решения только самых простых задач.

В настоящее время большее развитие и применение нашли методы оптимизации, основанные на применении ЭВМ, в частности градиентный метод.

Процесс оптимизации с использованием ЭВМ заключается в последовательном изменении количества резервных элементов в соответствии с определенным правилом, приводящим к получению требуемого оптимального состава резерва, причем используемое правило определяет направление или порядок перехода от одного резервируемого элемента к другому.

При градиентном методе движение производится в направлении градиента целевой функции, который характеризует эффективность каждого движения.

При любом методе оптимизация структуры резерва является процессом многошаговым. Шаг – это увеличение на единицу числа резервных элементов того или иного вида. В градиентном методе эффективность шага E оценивается удельным приращением надежности, т.е. отношением приращения надежности ΔR к приращению затрат ΔC:

$$F = \frac{\Delta R}{\Delta C}$$

Каждый раз делается такой шаг и при этом меняется число таких элементов, которые дают наибольшее значение отношения приращения надежности ΔR к приращению затрат ΔC:

$$F = \frac{\Delta R}{\Delta C} \to \max$$

Какой именно элемент дает наибольшее удельное приращение надежности, можно выяснить только в результате пробных шагов.

После каждого шага контролируется величина надежности системы R и стоимости резерва C. При достижении требуемого значения $R = R_0$ или величины лимита $C = C_0$ поиск заканчивается.

Найдем эффективность произвольного $(N+1)$-го шага при том или ином его направлении, т.е. при увеличении на единицу резерва того или иного вида. После N шагов вектор $X^{(N)}$ состава резерва будет иметь вид $X^{(N)} = \left(X_1^{(N)}, X_2^{(N)}, \dots, X_i^{(N)}, \dots, X_n^{(N)}\right)$, где $X_i^{(N)}$ – количество резервных элементов -го вида, $i = 1,2,\dots,n$, после N шагов.

Представим вектор $X^{(N)}$ состава резерва в виде $X^{(N)} = (x_i^{(N)}, X_i^{(N)})$, где выделена -ая составляющая $x_i^{(N)}$, а $X_i^{(N)} = \left(x_1^{(N)}, \dots, x_{i-1}^{(N)}, x_{i+1}^{(N)}, \dots, x_n^{(N)}\right)$.

Показатель надежности системы после i-го шага

$$R^{(N)} = R\left(X^{(N)}\right) = R_i\left(x_i^{(N)}\right) \cdot R\left(X_i^{(N)}\right) = R_i^{(N)} \cdot R\left(X_i^{(N)}\right),$$

а стоимость резерва

$$C^{(N)} = C\left(X^{(N)}\right) = \sum_{i=1}^{n} C_i \cdot x_i^{(N)}.$$

Если при поиске $(N+1)$-го шага включить один резервный блок в i-ой подсистеме, то надежность системы $R^{(N+1)}$ станет равной

$$R^{(N+1)} = R\left(X_i^{(N)}\right) \cdot R_i\left(x_i^{(N)} + 1\right),$$

приращение надежности

$$\Delta R = R^{(N+1)} - R^{(N)} = R\left(X_i^{(N)}\right) \cdot \left(R_i\left(x_i^{(N)} + 1\right) - R_i\left(x_i^{(N)}\right)\right),$$

или, умножив и разделив на $R\left(X_i^{(N)}\right)$, получим

$$\Delta R = R\left(X^{(N)}\right) \cdot \frac{R_i\left(x_i^{(N)} + 1\right) - R_i\left(x_i^{(N)}\right)}{R_i\left(x_i^{(N)}\right)}$$

Стоимость резерва увеличивается на стоимость -го элемента, а показатель эффективности пробного шага

$$\frac{\Delta R}{\Delta C} = R\left(X^{(N)}\right) \cdot \frac{R_i\left(x_i^{(N)} + 1\right) - R_i\left(x_i^{(N)}\right)}{C_i \cdot R_i\left(x_i^{(N)}\right)}$$

Поскольку множитель $R\left(X^{(N)}\right)$ при различных вариантах шага остается неизменным и не влияет на нахождение наиболее эффективного шага, его можно отбросить, а номер подсистемы, в которой следует увеличить резерв на $(N+1)$-м шаге, выбрать из условия

$$F^{(N+1)} = maxFi^{(N)},$$

$$1 \leq i \leq n,$$

где

$$F_i{}^{(N)} = \frac{R_i\left(x_i{}^{(N)} + 1\right) - R_i\left(x_i{}^{(N)}\right)}{C_i R_i(x_i{}^{(N)})}$$

Полученные выражения лежат в основе алгоритма поиска оптимальной структуры резерва.

Если блоки комплекса имеют высокую надежность, т.е. в $q_i \ll 1$, то выражение (2) можно упростить, учитывая, что

$$R_i\left(x_i^{(N)}\right) = R_i^{(N)} = 1 - Q_i^{(N)} = 1 - Q_i\left(x_i^{(N)}\right) = 1 - q_i^{\left(x_i^{(N)}+1\right)},$$

$$F_i^{(N)} = \frac{1}{C_i}\left[\frac{R_i\left(x_i^{(N)}+1\right)}{R_i(x_i^{(N)})} - 1\right] = \frac{1}{C_i}\left[\frac{1 - Q_i^{(N)} \cdot q_i}{1 - Q_i^{(N)}} - 1\right] = \frac{Q_i^{(N)}}{C_i}(1 - q_i) \approx \frac{Q_i^{(N)}}{C_i}.$$

Здесь $Q_i^{(N)}$ означает величину вероятности отказа -ой подсистемы в степени, равной общему количеству блоков i-го вида (основного и резервных) после того, как сделано N шагов.

Из описанного алгоритма следует, что об эффективности пробного шага для всей системы можно судить по его эффективности для той подсистемы, в которой он сделан. Это существенно сокращает объем вычислений. Кроме того, при последующих ш оценки шагах сохраняют силу оценки эффективности пробных шагов во всех подсистемах, в которых не произошло изменений. Дополнительно на каждом шаге приходится определять эффективность только для подсистемы, в которой перед этим был увеличен объем резерва. Это дополнительно резко уменьшает количество необходимых расчетов.

При решении прямой задачи оптимизации контролируется надежность системы $R\left(x^{(N)}\right)$ и процесс поиска заканчивается на таком шаге N, при котором выполняется условие:

$$R\left(x^{(N-1)}\right) \leq R_0 \leq R\left(x^{(N)}\right).$$

При решении обратной задачи контролируется стоимость системы $\mathrm{C}\left(x^{(N)}\right)$ и оптимизация заканчивается на шаге N, когда

$$\mathrm{C}\left(x^{(N-1)}\right) \leq \mathrm{C}_0 \leq \mathrm{C}\left(x^{(N)}\right).$$

В первом случае искомым решением является вектор состава резерва $X^{(N)}$, во втором случае - $X^{(N-1)}$.

4.3 Исходные данные к работе

Таблица 4.1

Номер варианта	q_1	q_2	q_3	q_4	C_1	C_2	C_3	C_4	R
1	0.07	0.055	0.03	0.045	800	40000	12000	8000	0.9996
2	0.04	0.035	0.06	0.08	7000	10000	35000	900	0.9997
3	0.065	0.085	0.035	0.05	30000	1000	11000	7000	0.9995
4	0.05	0.06	0.02	0.07	750	50000	10000	500	0.9998
5	0.45	0.055	0.03	0.07	900	40000	18000	6000	0.9998
6	0.065	0.08	0.04	0.08	8000	15000	9000	900	0.9996
7	0.05	0.01	0.08	0.06	700	20000	10000	1500	0.9994
8	0.055	0.03	0.02	0.01	500	10000	100	18000	0.9997
9	0.03	0.04	0.03	0.055	6000	4500	1500	35000	0.9996
10	0.06	0.065	0.04	0.035	900	6000	8000	11000	0.9997
11	0.035	0.05	0.08	0.085	1500	900	900	10000	0.9998
12	0.02	0.45	0.02	0.06	8000	1500	7000	18000	0.9995
13	0.03	0.065	0.04	0.055	900	35000	500	9000	0.9996
14	0.04	0.05	0.055	0.04	7000	11000	6000	10000	0.9994
15	0.08	0.055	0.035	0.065	500	10000	900	100	0.9998
16	0.02	0.03	0.085	0.05	6000	18000	1500	800	0.9992
17	0.07	0.06	0.06	0.45	900	9000	18000	7000	0.9993
18	0.07	0.035	0.055	0.065	1500	10000	7500	30000	0.9994
19	0.08	0.02	0.04	0.05	18000	100	8000	750	0.9995
20	0.055	0.03	0.065	0.055	6000	6000	35000	900	0.9998
21	0.035	0.04	0.05	0.08	900	900	11000	8000	0.9997
22	0.085	0.08	0.45	0.05	1500	1500	10000	700	0.9999
23	0.06	0.02	0.065	0.07	35000	8000	18000	6000	0.9996
24	0.055	0.04	0.05	0.07	11000	900	9000	900	0.9994
25	0.08	0.065	0.055	0.08	10000	7000	10000	1500	0.9998
26	0.05	0.05	0.055	0.06	18000	500	100	900	0.9995
27	0.07	0.45	0.035	0.035	9000	6000	800	1500	0.9997
28	0.07	0.065	0.085	0.02	10000	900	7000	35000	0.9995
29	0.08	0.05	0.06	0.03	100	1500	30000	11000	0.9994
30	0.06	0.055	0.055	0.04	600	18000	750	10000	0.9993

Требуется обеспечить заданную надежность комплекса при условии, что стоимость резерва минимальна.

4.4 Порядок выполнения работы:

1.Определить вероятность отказов подсистемы комплекса при различном количестве резервных блоков.

Результаты свести в таблицу 4.2.

Таблица4.2

Кол-во резервных блоков	Вероятность отказа i-ой подсистемы			
	$i = 1$	$i = 2$	$i = 3$	$i = 4$
0				
1				
2				
3				
n				

2. Определить показатель эффективности $F_i^{(N)}$ изменения объема резерва в каждой подсистеме при различном количестве уже имеющихся резервных блоков. Результаты свести в таблицу 4.3.

Таблица 4.3

Кол-во резервных блоков после i-го шага	Показатель эффективности $F_i^{(N)}$ изменения объема резерва в каждой подсистеме			
	$i = 1$	$i = 2$	$i = 3$	$i = 4$
0				
1				
2				
3				
n				

3. Провести анализ данных таблицы 4.3 и выбрать оптимальную структуру резерва.

4. Провести расчет надежности комплекса $R^{(N)}$ и стоимости его резерва $C^{(N)}$ после каждого шага оптимизации и результаты расчета свести в таблицу 4.4.

5. Изобразить графически оптимальную структуру резерва

Таблица 4.4

Номер шага	Вероятность отказа системы $Q^{(N)}$	Вероятность безотказной работы $R^{(N)}$	Стоимость резерва, $C^{(N)}$,руб
0			
1			
2			
3			
4			
5			
. . .			

Примечание: вероятность отказа системы $Q^{(N)}$ определяется по формуле:

$$Q^{(N)} = Q_1^{(N)} + Q_2^{(N)} + Q_3^{(N)} + \cdots = \sum_1^n Q_i^{(N)}.$$

Вероятность безотказной работы системы $R^{(N)}$ можно определить по формуле:

$$R^{(N)} = \prod_{i=1}^{n} R_i{}^{(N)} = \prod_{i=1}^{n} (1 - Q^{(N)}) \approx 1 - \left(Q_1{}^{(N)} + Q_2{}^{(N)} + Q_3{}^{(N)} + \cdots\right) =$$

$$= 1 - \sum_{1}^{n} Q_i{}^{(N)}$$

6. Составить блок-схему программы оптимизации структуры резерва градиентным методом.

7. Написать программу оптимизации структуры резерва для ЭВМ .

8. Решить на ЭВМ задачу оптимизации структуры резерва для исходных данных, совпадающих с исходными данными ручного счета.

4.6 Содержание отчета:

1. Цель работы
2. Исходные данные для расчета (показателя надежности в стоимость отдельных блоков объекта связи, а также заданная надежность всего объекта).
3. Вероятности отказов отдельных подсистем объекта связи при Различном количестве резервных блоков (таблица 4.2)
4. Показатели эффективности увеличения на единицу количества резервных блоков каждой подсистемы при различном количестве уже имеющихся резервных блоков (таблица 4.3) с отмеченными шагами оптимизации.
5. Результаты расчета надежности объекта связи и стоимости его резервных блоков после каждого шага оптимизации (таблица 4.3).
6. Графическое изображение оптимальной структуры резерва с указанием стоимости резерва.
7. Блок-схема алгоритма.
8. Программа оптимизации для ЭВМ.