Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Кафедра систем телекоммуникаций

# УПРАВЛЯЮЩИЕ СИСТЕМЫ В ТЕЛЕКОММУНИКАЦИЯХ

Лабораторный практикум
для студентов специальностей
I-45 01 01 «Многоканальные системы телекоммуникаций»
и I-45 01 02 «Системы радиосвязи, радиовещания и телевидения»
всех форм обучения



Минск 2007

УДК 621.395
ББК 32.882
У 67

Р е ц е н з е н т
доцент кафедры сетей и устройств телекоммуникаций БГУИР,
канд. техн. наук А. И. Королев

С о с т а в и т е л ь
О. А. Хацкевич



У 67 **Управляющие** системы в телекоммуникациях : лаб. практикум для студ. спец. I-45 01 01 «Многоканальные системы телекоммуникаций» и I-45 01 02 «Системы радиосвязи, радиовещания и телевидения» всех форм обуч. / сост. О. А. Хацкевич. – Минск : БГУИР, 2007. – 29 с.
ISBN 978-985-488-175-1

Рассмотрены основные алгоритмы управления трафиком на сетях связи: алгоритм Дейкстры, алгоритм Форда, алгоритм оптимизации потока на сети, управления пучками каналов, максимизации потока на сетях связи.

**УДК 621.395**
**ББК 32.882**

**ISBN 978-985-488-175-1**

Содержание

Лабораторная работа №1

# МАРШРУТИЗАЦИЯ В СЕТЯХ СВЯЗИ

## 1. ЦЕЛЬ РАБОТЫ

Изучить методы определения кратчайших путей с целью использования их при решении задач распределения каналов и потоков информации в сетях связи.

## 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.

## 3. МЕТОД ДЕЙКСТРЫ

Метод 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 $k=l$; $k,l=1,2,\dots M$;

$l_{kl}$ – длина 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 $k = l$, $k, l = 1, 2, \ldots M$. 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

$$d_k = \min\left[d_k, d_m + l_{mk}\right],$$

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.

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

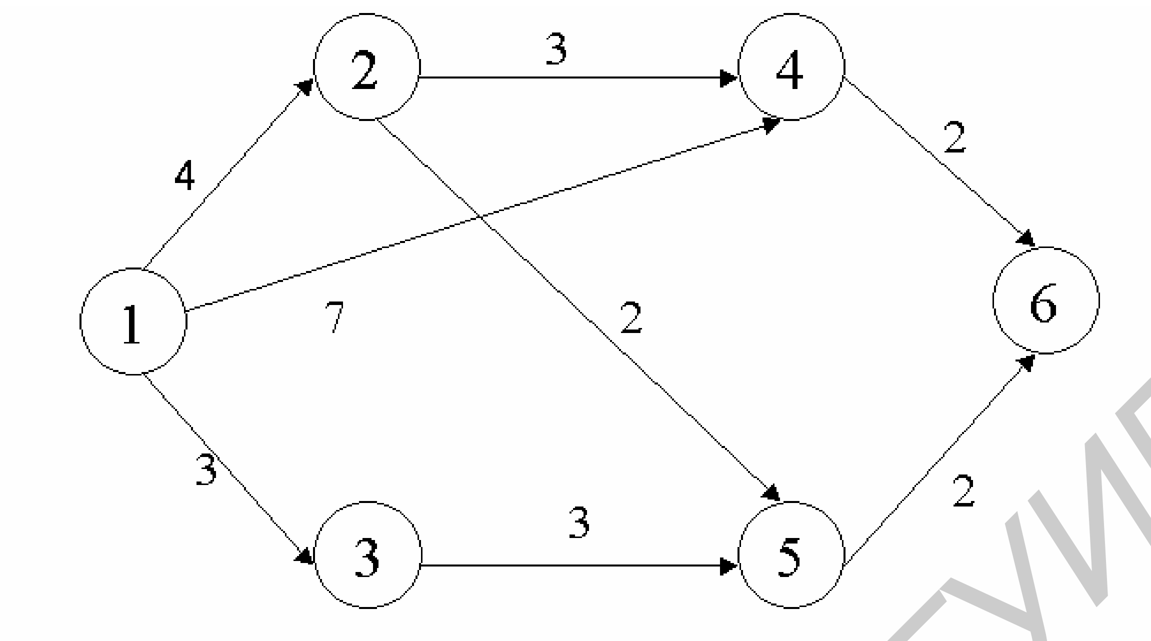


Рис. 1.1

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

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

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, $m=1$.

Итерация 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

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

Подставляя поочередно $k$ в последнюю формулу, получим

$d_2=\min\left[d_2, d_1+l_{12}\right], =\min\left[\infty, 0+4\right]=4$,

$d_3=\min\left[d_3, d_1+l_{13}\right], =\min\left[\infty, 0+3\right]=3$,

$d_4=\min\left[d_4, d_1+l_{14}\right], =\min\left[\infty, 0+7\right]=7$,

$d_5=\min\left[d_5, d_1+l_{15}\right], =\min\left[\infty, 0+\infty\right]=\infty$,

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

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

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

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 С, идти к шагу 3.

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

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

Получим

$d_2 = \min\left[d_2, d_3 + l_{32}\right], = \min\left[4, 3 + \infty\right] = 4$,

$d_4 = \min\left[d_4, d_{10} + l_{34}\right], = \min\left[7, 3 + \infty\right] = 7$,

$d_5 = \min\left[d_5, d_3 + l_{35}\right], = \min\left[\infty, 3 + 3\right] = 6$,

$d_6 = \min\left[d_6, d_1 + l_{36}\right], = \min\left[\infty, 0 + \infty\right] = \infty$.

Шаг 4. Определим наименьший вес

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

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 $k = 4, 5, 6$, $m = 2$.

$d_4 = \min\left[d_4, d_2 + l_{24}\right], = \min\left[7, 4 + 3\right] = 7$,

$d_5 = \min\left[d_5, d_2 + l_{25}\right], = \min\left[6, 4 + 2\right] = 6$,

$d_6 = \min\left[d_6, d_2 + l_{26}\right], = \min\left[\infty, 4 + \infty\right] = \infty$.

Шаг 4.

$\min\left[d_k\right] = \min\left[7, 6, \infty\right] = 6$,

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

$d_1 = 0$, $d_3 = 3$, $d_2 = 4$, $d_5 = 6$, $d_4 = 7$, $d_6 = \infty$.

Итерация 4

Шаг 2. Идти к шагу 3.

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

$d_4 = \min\left[d_4, d_5 + l_{54}\right], = \min\left[7, 6 + \infty\right] = 7$,

$d_6 = \min\left[d_6, d_5 + l_{56}\right], = \min\left[\infty, 6+2\right] = 8$.

Шаг 4

$\min\left[d_k\right] = \min\left[7,8\right] = 7$.

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

$d_1 = 0, d_2 = 4$, $d_3 = 3$, $d_4 = 7$, $d_5 = 6$, $d_6 = 8$.

И т е р а ц и я 5

Шаг 2. Идти к шагу 3.

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

$d_6 = \min\left[d_6, d_4 + l_{46}\right], = \min\left[8, 7+2\right] = 8$

$C = \{1, 2, 3, 4, 5, 6\}$, $C = \{0\}$,

$d_1 = 0, d_2 = 4$, $d_3 = 3$, $d_4 = 7$, $d_5 = 6$, $d_6 = 8$.

И т е р а ц и я 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 табл. 1.1.

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 |

Как видно из табл. 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:

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

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 $i = 1, 2, 3, 4, 5$:

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

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

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

$d_6 = d_4 + l_{49} = 7 + 2 = 9$,

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

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.

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.

Содержание отчета

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

## Лабораторная работа №2
## ОПТИМИЗАЦИЯ СЕТИ МЕТОДОМ ФОРДА – ФАЛКЕРСОНА

### 1. ЦЕЛЬ РАБОТЫ

Изучить алгоритм оптимизации сетей Форда – Фалкерсона.

Данный метод предназначен для распределенной системы динамического управления потоками информации в сетях связи, позволяет найти кратчайшие пути от всех узлов сети к одному общему входящему узлу, который обозначим как узел $I$.

Для сохранения информации о кратчайшем пути будем обозначать каждый узел $k$ парой чисел $\left(n, L_{kl}\right)$, где $n$ – номер следующего, соседнего с узлом $k$ в данном кратчайшем пути, $L_{kl}$ – на текущей итерации кратчайшее расстояние от узла $k$ до узла $I$.

Алгоритм нахождения кратчайших путей по методу Форда – Фалкерсона состоит в следующем

Шаг 1. В исходном состоянии принимается, что для узла $I$ $L_{11}=0$, а все остальные узлы обозначают парой чисел $\left(n, L_{kl}\right)$, при этом вместо номера узла ставится точка, что означает, что еще ни один из соседних узлов сети не выбран в качестве транзитного в кратчайшем пути, а длина пути принимается равной бесконечности, т.е. пара, характеризующая все узлы, кроме узла $I$, записывается в виде $(\cdot, \infty)$.

Шаг 2. На последующих шагах обновляется кратчайшее расстояние для каждого транзитного узла на основе выражения

$$L_{kl} = \min_{i}\left[l_{ki} + L_{il}\right]$$

где $L_{il}$ – кратчайшее расстояние от узла $i$, соседнего с узлом $k$, до узла $I$;

$l_{ki}$ – длина ребра между узлами $k$ и $i$.

При этом берется минимальное значение среди всех значений $\left[l_{ki} + L_{il}\right]$ при связи узла $k$ с $I$ через все возможные узлы $i$.

В обновленном весе узла $k$ указывается соответствующий номер соседнего узла $i$, через который проходит кратчайший путь, выбранный при вычислении нового значения $L_{kl}$. Этот шаг повторяется до тех пор, пока не прекратятся изменения весов в каждом узле.

Проиллюстрируем применение алгоритма Форда – Фалкерсона на примере.

П р и м е р. Для заданной структуры сети (рис. 2.1) определить кратчайшее расстояние между всеми узлами сети и узлом $I$. Цифры рядом с ребрами указывают длину ребра. Порядок выполнения шагов по этому методу указан в табл. 2.1.

Шаг 0. В исходном состоянии ни один из узлов не используется в качестве транзитного в кратчайшем пути от узлов 2...6 к узлу $I$, следовательно, в табл. 2.1 пишем пару чисел $(\cdot,\infty)$.

Т а б л и ц а 2.1

| Шаг итерации | Узел 2 | Узел 3 | Узел 4 | Узел 5 | Узел 6 |
|---|---|---|---|---|---|
| 0 | $(\cdot,\infty)$ | $(\cdot,\infty)$ | $(\cdot,\infty)$ | $(\cdot,\infty)$ | $(\cdot,\infty)$ |
| 1 | (1, 2) | (1, 5) | (1, 1) | $(\cdot,\infty)$ | $(\cdot,\infty)$ |
| 2 | (1, 2) | (4, 4) | (1, 1) | (4, 2) | (3, 10) |
| 3 | (1, 2) | (5, 3) | (1, 1) | (4, 2) | (5, 4) |
| 4 | (1, 2) | (5, 3) | (1, 1) | (4, 2) | (5, 4) |

Шаг 1. В связи с тем, что $L_{11}=0$, на первом шаге для каждого из соседних узлов будет указан узел $I$ и расстояние до него, равное длине соединяющего их ребра, т.е.

$L_{21}=\left[I_{21}+L_{11}\right]=[2+0]=2$,

$L_{31}=\left[I_{31}+L_{11}\right]=[5+0]=5$,

$L_{41}=\left[I_{41}+L_{11}\right]=[1+0]=1$,

$L_{51}=\left[I_{51}+L_{11}\right]=[\infty+0]=\infty$,

$L_{61}=\left[I_{61}+L_{11}\right]=[\infty+0]=\infty$.

Поэтому в табл. 2.1 на первом шаге будут заполнены клетки с номером узла 1.

Шаг 2. На втором шаге уже можно определить пути через один транзитный узел. Определим кратчайшее расстояние от узла 2 к узлу 1 через транзитные узлы 3 и 4:

$L'_{21}=\left[I_{10}+L_{31}\right]=[3+5]=8$.

При этом значение $L_{31}$ берем из результатов, полученных на предыдущем шаге:

$L''_{21}=\left[I_{24}+L_{41}\right]=[2+1]=3$.

На первом шаге кратчайшее расстояние между узлами 2 и 1 было равно 2, следовательно, столбец с узлом 2 не изменится.

Определим кратчайшее расстояние от узла 3 к узлу 1 через узлы 2 и 4:

$L'_{31}=\left[L_{10}+L_{21}\right]=[3+2]=5$,

$L''_{31}=\left[I_{34}+L_{41}\right]=[3+1]=4$.

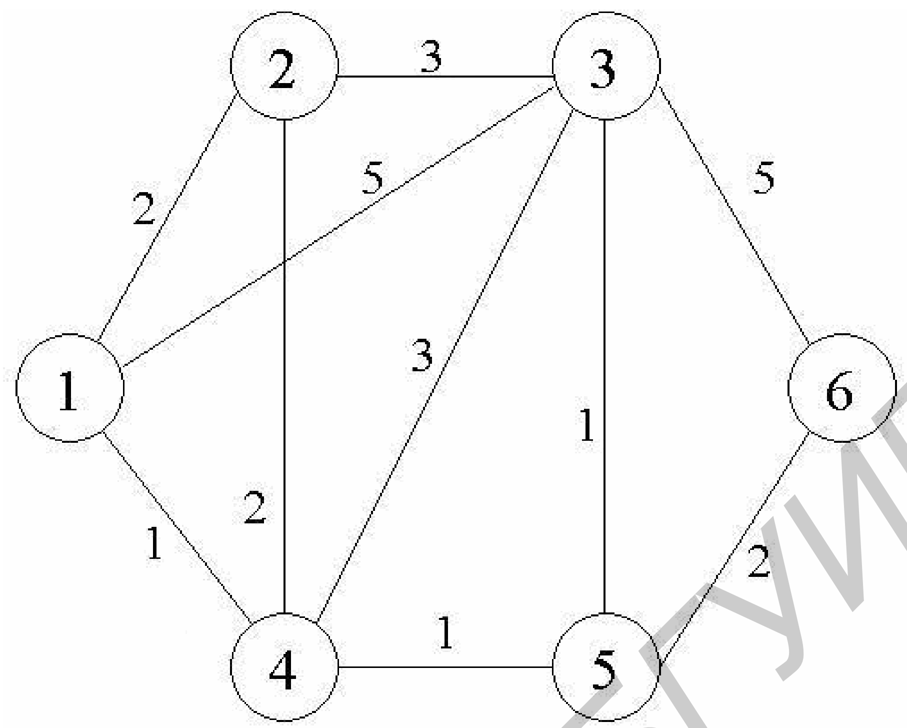


Рис. 2.1

На первом шаге $L_{31} = 5$, следовательно, принимаем в качестве $L_{31}$ новое значение, равное 4, и в таблицу записываем новую пару чисел $(4,4)$.

Аналогично определяем кратчайшее расстояние от узла 4 до узла 1:

$L'_{41} = [I_{43} + L_{31}] = [3+5] = 8$,

$L''_{41} = [I_{42} + L_{21}] = [2+2] = 4$.

В таблице значение данной пары не изменится.

Кратчайшее расстояние между узлами 5 и 1 будет равно наименьшему из двух чисел, определяемых по формулам

$L'_{51} = [I_{53} + L_{31}] = [1+5] = 6$,

$L''_{51} = [I_{52} + L_{41}] = [1+1] = 2$.

Следовательно, $L_{51} = \min[6, 2] = 2$, а в таблицу записываем пару чисел $(4,2)$.

Определим $L_{61}$ через транзитный узел 3:

$L_{61} = [I_{63} + L_{31}] = [5+5] = 10$,

в таблицу запишем пару чисел (3, 10).

Шаг 3. На этом шаге можно определять кратчайшие расстояния от узлов 2...6 к узлу 1.

Как видно из табл. 2.1, для данного примера потребовалось выполнить четыре итеративных шага. При этом на втором и третьем шагах в столбце для узла 3 изменились пары чисел. Это означает, что кратчайшим путем между узлами 3 и 1 является не непосредственная связь, а через транзитные узлы 5 и 4.

Для иллюстрации результатов, приведенных в табл. 2.1, определим кратчайший путь между узлом 6 и 1. Из последней строки табл. 2.1, в столбце для узла 6, из пары чисел (5, 4) определяем, что длина этого пути равна 4, а первым промежуточным узлом в кратчайшем пути является узел 5. Из столбца, принадлежащего узлу 5, определим следующий транзитный узел, т.е. узел 4, расстояние от которого до узла 1 равно 2. Из столбца, относящегося к узлу 4, видно, что следующим в пути будет требуемый узел 1. Таким образом, кратчайший путь из узла 6 в узел 1 определяется последовательность узлов 6, 5, 4, 1, а длина этого пути равна 4.

Порядок выполнения работы:

1. Получить исходные данные к работе.
2. Построить сеть связи.
3. Используя метод Форда–Фалкерсона, определить кратчайшие расстояния на сети.

Лабораторная работа №3
# МАКСИМИЗАЦИЯ ПОТОКА В СЕТЯХ СВЯЗИ

## 1. ЦЕЛЬ РАБОТЫ

Изучить алгоритм максимизации потока в сетях 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

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

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

а) вычесть $q$ из всех $c_{ij}^{-}$;

б) прибавить $q$ 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 $s \to t$.

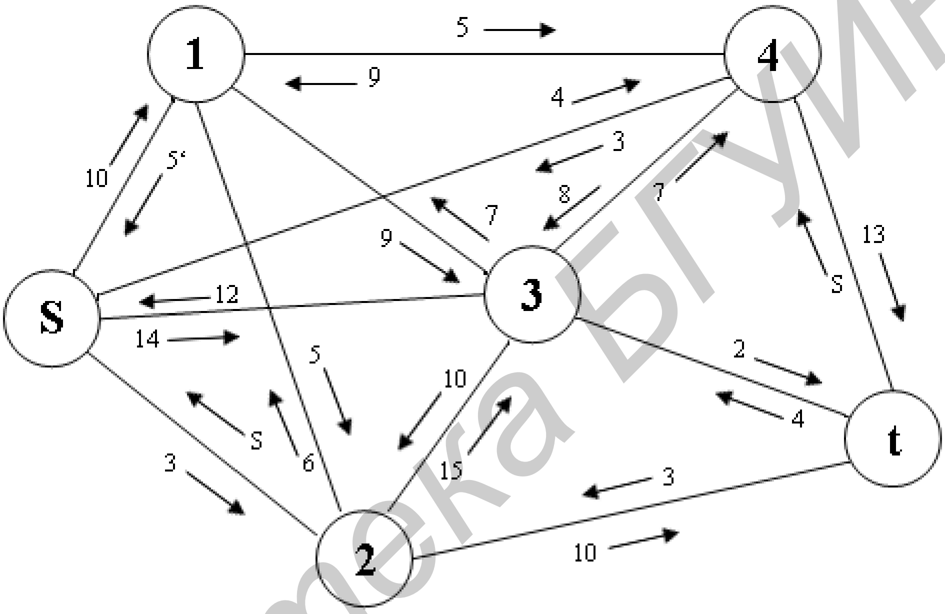


Рис. 3.1

Пример. Рассмотрим сеть на рис. 3.1 с данными пропускными способностями. Соответствующая матрица пропускных способностей $C$ приведена в табл. 3.1.

Таблица 3.1

| | | s | 1 | 2 | 3 | 4 | t |
|---|---|---|---|---|---|---|---|
| | s | | 10+ | 3 | 14 | 4 | |
| | 1 | 5+ | | 5 | 9 | 5- | |
| | 2 | 5 | 6 | | 15 | | 10 |
| C= | 3 | 12 | 7 | 10 | | 7 | 2 |
| | 4 | 3 | 9+ | | 8 | | 13- |
| | t | | | 3 | 4 | 5+ | |

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

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

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

Заметим, что можно выбирать различные исходные цепи. Очевидно, что хороший выбор (вначале и на каждой итерации) должен давать наибольшее значение $q$. 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 табл. 3.1 корректируется путем вычитания $q = 5$ из всех элементов, помеченных знаком (–), и сложения со всеми элементами, имеющими знак (+). Результаты приведены в табл. 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 $[s \to 3 \to 2 \to t]$  $q = \min\{14, 10, 10\} = 10$.

Результаты последующих итераций приведены в табл. 3.3–3.7. Из табл. 3.2 следует, что между $s$ и $t$ нельзя построить цепей с положительным потоком, поскольку все элементы в столбце $t$ равны нулю. Таким образом, табл. 3.7 дает матрицу $C^*$.

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 $[s \to 1 \to 3 \to 4 \to t]$ $\quad q = \min\{5, 9, 7, 8\} = 5$.

В табл. 3.1 (матрица $C$) и 3.3 (матрица $C^*$) приведены данные, характеризующие оптимальный поток, которые получаются вычислением $X = C - C^*$ и заменой отрицательных величин нулями. Табл. 8 дает матрицу $X$ .

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 $[s \to 4 \to t]$ $\quad q = \min\{4, 3\} = 3$.

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 $[s \to 3 \to t]$ $\quad q = \min\{4, 2\} = 2$.

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.)

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 табл. 3. 7 видно, что

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

Сумма всех $q\,(5+10+5+3+2=25)$ также дает максимальный поток. Графически решение представлено на рис. 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 рис. 3.1 можно выделить следующие разрезы:

| Разрез | Пропускная способность |
|---|---|
| ($s$, 1), ($s$, 2), ($s$, 3), ($s$, 4)<br>(4, $t$), (3, $t$), (2, $t$)<br>(1, 4), ($s$, 4), (3, 4), (3, $t$), (2, $t$) | 10 + 3+14 + 4 = 31<br>13 + 2+10 = 25<br>5+4 + 7+2+10 = 28 |

Интуитивно ясно, 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.

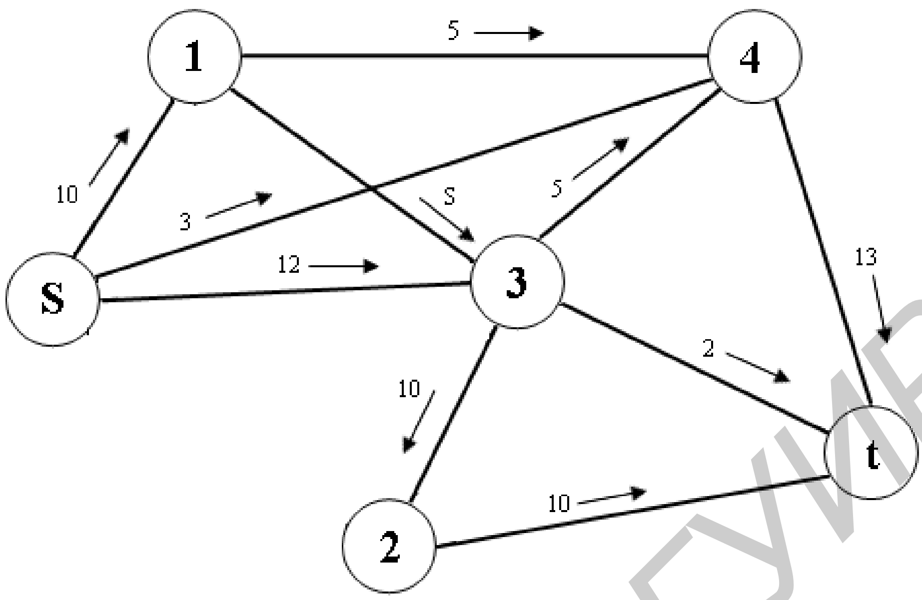


Рис. 3.2

Задачи о кратчайшем пути и максимальном потоке можно сформулировать как задачи линейного программирования. Следует, однако, подчеркнуть, что решение сетевых задач симплекс-методом едва ли целесообразно. С другой стороны, изучение формулировок сетевых задач как задач ЛП помогает идентифицировать модели ЛП, которые на первый взгляд не являются сетевыми, но которые либо непосредственно, либо с некоторыми модификациями можно свести к сетевым. Очевидное преимущество такого подхода состоит в том, что при использовании сетевых постановок эффективность вычислений может значительно увеличиться.

Модель линейного программирования для задачи о кратчайшем пути строится следующим образом.

1. Каждая переменная соответствует дуге.
2. Каждое ограничение соответствует узлу.

Пусть $x_{ij}$ представляет величину потока по дуге $(i, j)$. Тогда задача о кратчайшем пути в сети с $n$ узлами формулируется так:

минимизировать $z = \sum\sum_{(i,j)} d_{ij} x_{ij}$

при ограничениях

$$\sum_{(1,j)} x_{1j} = 1 \text{ (исходный пункт)},$$

$$\sum_{(i,k)} x_{ik} = \sum_{(k,j)} x_{kj} \text{ (для всех } k \neq 1 \text{ или } n),$$

$$\sum_{(i,n)} x_{in} = 1 \text{ (пункт назначения)},$$

$$x_{ij} \geq 0 \text{ для всех } i \text{ и } j.$$

Ограничения модели линейного программирования соответствуют формулировке задачи о кратчайшем пути как транспортной задачи с промежуточными пунктами. Единица потока доставляется из узла 1 в узел $n$. Первым и последним ограничениями устанавливается, что суммарный поток (сумма переменных), выходящий из узла 1, равен 1, как и суммарный поток, поступающий в узел $n$. В любом промежуточном узле суммарный входящий поток равен суммарному выходящему потоку. Целевая функция требует, чтобы общее расстояние, пройденное единицей потока, было минимальным.

Следует подчеркнуть, что данная постановка имеет реальный смысл, если $x_{ij} = 0$ или 1, т. е. дуга $(i, j)$ принадлежит кратчайшему пути, только если $x_{ij} = 1$. Если $x_{ij} = 0$, то $(i, j)$ не входит в кратчайший путь. Несмотря на то, что условия $x_{ij} = 0$ или 1 не отражены в модели в явном виде, ее специальная структура всегда приводит к оптимальному решению, которое удовлетворяет этим требованиям. Действительно, модель обладает свойством абсолютной унимодулярности, согласно которому в решении задачи линейного программирования всегда $x_{ij} = 0$ или 1.

Таким же образом к задаче линейного программирования можно свести задачу о максимальном потоке. Пусть $y$ – поток из источника 1 в сток $n$. Обозначив поток в дуге $(i, j)$ через $x_{ij}$, получим следующую модель линейного программирования:

максимизировать $z = y$

при ограничениях

$$\sum_{(1,j)} x_{1j} = y \text{ (источник)},$$

$$\sum_{(i,k)} x_{ik} = \sum_{(k,j)} x_{kj} \text{ (для всех } k \neq 1 \text{ или } n),$$

$$\sum_{(i,n)} x_{in} = y \text{ (сток)},$$

$$0 \le x_{ij} \le c_{ij} \text{ для всех } i \text{ и } j,$$

где $c_{ij}$ обозначает пропускную способность дуги $(i, j)$. Заметим, что ограничения строятся по той же схеме, которая использовалась для построения модели ЛП задачи о кратчайшем пути.

## ЛАБОРАТОРНАЯ РАБОТА №4

## ОПРЕДЕЛЕНИЕ ЕМКОСТИ ПУЧКОВ КАНАЛОВ

### 1. ЦЕЛЬ РАБОТЫ

Изучить способы определения путей между узлами связи, методику оптимизации емкости пучков каналов между узлами сети, решить задачу оптимизации на ЭВМ симплекс-методом.

### 2. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Основу сети связи составляют узлы (станции и линии связи). Для обеспечения связи любой пары абонентов на сети не всегда целесообразно использовать линии связи, уплотненные аппаратурой связи. Более эффективное использование каналов достигается в том случае, когда такие линии связи прокладываются на основных направлениях, а создание структуры вторичной сети осуществляется на узлах при помощи устройств долговременной коммутации.

Для образования оптимальной структуры сети при проектировании необходимо определить возможные пути организации пучков каналов, произвести оценку целесообразности организации пучков по каждому из путей и выбрать емкости пучков таким образом, чтобы обеспечить передачу заданных объемов информации и наиболее эффективно использовать каналы и оборудование сети.

Аналитически описать происходящие процессы в сетях трудно, а иногда и невозможно. Для успешного решения таких задач необходимо применять современные методы оптимизации и широко использовать вычислительную технику. Важным моментом при решении таких задач является корректное построение математической модели сети. Существует несколько способов описания сети связи. Одним из наиболее простых и наглядных способов является построение дерева путей [1].

Дерево путей для узла 2 сети, изображенной в виде графа на рис. 4.1, приведено на рис. 4.2. Дерево путей строится по матрице связности:

$$B = \begin{bmatrix} - & 1 & 0 & 0 & 1 \\ 1 & - & 1 & 1 & 0 \\ 0 & 1 & - & 1 & 1 \\ 0 & 1 & 1 & - & 1 \\ 1 & 0 & 1 & 1 & - \end{bmatrix}.$$

Перечень путей между узлами сети можно записывать и в виде таблицы путей $B$ размером $k \times m$, где $k$ – число путей, а $m$ – число ветвей сети.

Задачу выбора оптимальной емкости пучков каналов между узлами сети можно сформулировать в виде задачи линейного программирования [2].

Пусть $M = \{m_i\}$ – совокупность путей, которые могут быть использованы для передачи потоков информации между узлами сети; $f_i$ – емкость пути $m_i$, а $c_i$ – вес этого пути, который определяет относительную целесообразность использования пути. Выбор $c_i$ зависит от критерия, по которому производится оптимизация. Он может быть выбран обратно пропорционально длине пути, стоимости и т.д. [1].

Тогда для оптимальной организации пучков каналов между узлами сети необходимо максимизировать сумму вида

$$F = \sum_{m_i \in M} c_i f_i .$$

При этом нужно учесть следующие ограничения:

1) емкости путей не могут быть отрицательными;

2) суммарная емкость путей между произвольной парой узлов сети должна быть равна (если это возможно) числу каналов, которое необходимо организовать в пучке между этой парой узлов сети, т.е.

$$\sum f_i \le j_{i,j} ,$$

где $j_{i,j}$ – требуемое число каналов между узлами $i$ и $j$;

3) для любой ветви $b_{xy}$ сети суммарная емкость всех путей, содержащих эту ветвь, не может быть больше емкости ветви, т.е.

$$\sum_{m_i \in b_{xy}} f_i \le b_{xy} .$$

Приведенный перечень ограничений является полным.

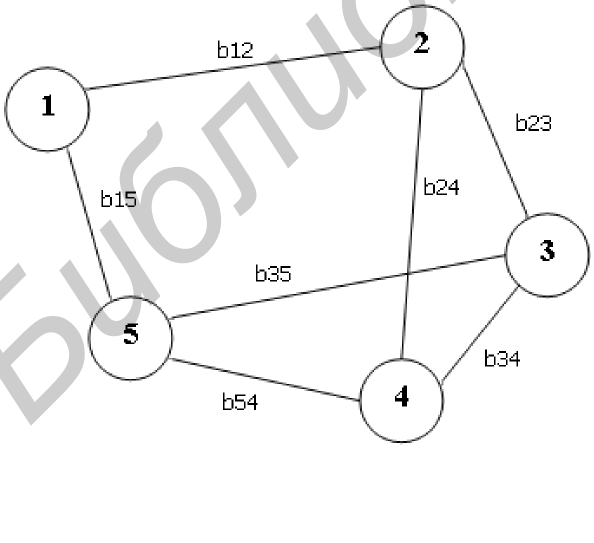


Рис. 4.1

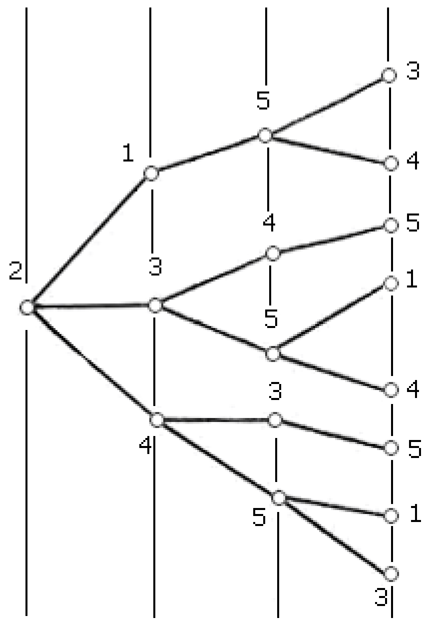


Рис. 4.2

В качестве примера рассмотрим сеть связи, изображенную на рис. 4.1. Емкость ветвей определяется матрицей

$$B = \begin{bmatrix} - & b_{12} & 0 & 0 & \\ b_{21} & - & b_{23} & b_{32} & 0 \\ 0 & b_{32} & - & b_{34} & b_{35} \\ 0 & b_{42} & b_{43} & - & b_{45} \\ b_{51} & 0 & b_{53} & b_{54} & - \end{bmatrix}.$$

Пусть в такой сети требуется организовать три пучка каналов между узлами 2 и 5; 1 и 4; 1 и 3 емкостью $j_1$, $j_2$, $j_3$.

Для организации пучков допускается использовать пути, записанные в табл. 4.1.

Таблица 4.1

| $m_i$ \ $b_{xy}$ | | $b_{12}$ | $b_{15}$ | $b_{23}$ | $b_{24}$ | $b_{34}$ | $b_{35}$ | $b_{45}$ |
|---|---|---|---|---|---|---|---|---|
| $m_1$ | 2…5 | 1 | 1 | | | | | |
| $m_2$ | 2…5 | | | 1 | | | 1 | |
| $m_3$ | 2…5 | | | | 1 | | | 1 |
| $m_4$ | 2…5 | | | 1 | | 1 | | 1 |
| $m_5$ | 2…5 | | | | 1 | 1 | 1 | 1 |
| $m_6$ | 1…4 | 1 | | | 1 | | | 1 |
| $m_7$ | 1…4 | | 1 | | | | | 1 |
| $m_8$ | 1…4 | 1 | | 1 | | 1 | | |
| $m_9$ | 1…4 | | 1 | | | 1 | 1 | |
| $m_{10}$ | 1…3 | 1 | | 1 | | | | |
| $m_{11}$ | 1…3 | | 1 | | | | 1 | |
| $m_{12}$ | 1…3 | 1 | | | 1 | 1 | | |
| $m_{13}$ | 1…3 | | 1 | | | | 1 | 1 |

Предположим, что пути 1, 2, 3, 6, 7 имеют вес 3, а пути 4, 5, 8, 9 – вес 2.

Тогда задачу в терминах линейного программирования можно записать в следующем виде:

$$\max F = \sum_{i=1}^{15} c_i f_i ,$$

при условии, что

$$f_i > 0 ,$$

$$f_1 + f_2 + f_3 + f_4 + f_5 \le j_{25} ,$$

$$f_6 + f_7 + f_8 + f_9 \le j_{14} ,$$

$$f_{10} + f_{11} + f_{12} + f_{13} \le j_{13} ,$$

$$f_1 + f_6 + f_8 + f_{10} + f_{12} \le b_{12},$$
$$f_1 + f_7 + f_8 + f_9 + f_{11} + f_{13} \le b_{15},$$
$$f_2 + f_4 + f_8 + f_{10} \le b_{23},$$
$$f_3 + f_5 + f_6 + f_{12} \le b_{14},$$
$$f_4 + f_5 + f_8 + f_9 + f_{12} \le b_{34},$$
$$f_2 + f_5 + f_{10} + f_{11} + f_{13} \le b_{35},$$
$$f_3 + f_4 + f_5 + f_6 + f_7 + f_{13} \le b_{45}.$$

Для приведения системы ограничений к каноническому виду в каждое из ограничений необходимо ввести по одной свободной переменной $x_1, x_2, ..., x_{10}$. Исходные данные приведены в табл. 4.2. Сформулированную таким образом задачу можно решить либо вручную, либо с помощью ЭВМ.

Если предположить, что $b_{23} = 20$, $b_{24} = b_{34} = 10$, а остальные емкости равны 15, то на седьмом шаге оптимизации можно получить оптимальное решение:

2…5: $m_2 = \{b_{23}, b_{35}\}$, $f_2 = 10$,

$m_3 = \{b_{24}, b_{45}\}$, $f_3 = 5$,

1…4: $m_6 = \{b_{12}, b_{24}\}$, $f_6 = 5$,

$m_7 = \{b_{15}, b_{45}\}$, $f_7 = 10$,

1…3: $m_{10} = \{b_{12}, b_{23}\}$, $f_{10} = 5$,

$m_{11} = \{b_{15}, b_{35}\}$, $f_{11} = 10$.

Приведенный выше метод решения задачи распределения каналов сформулирован для однородных сетей без учета категоричности абонентов. Методы линейного программирования можно использовать и для неоднородных сетей, и с учетом категорийности, однако при этом алгоритм будет более сложным [1].

Таблица 4.2

| $f_i$ | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ | $f_9$ | $f_{10}$ | $f_{11}$ | $f_{12}$ | $f_{13}$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $x_8$ | $x_9$ | $x_{10}$ | $b$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 1 | 1 | 1 | 1 | | | | | | | | | 1 | | | | | | | | | | $j_{25}$ |
| | | | | | | 1 | 1 | 1 | 1 | | | | | | 1 | | | | | | | | | $j_{14}$ |
| | | | | | | | | | | 1 | 1 | 1 | 1 | | | 1 | | | | | | | | $j_{13}$ |
| | 1 | | | | | 1 | | 1 | | 1 | | 1 | | | | | 1 | | | | | | | $b_{12}$ |
| | 1 | | | | | | 1 | 1 | 1 | | 1 | | 1 | | | | | 1 | | | | | | $b_{15}$ |
| | | 1 | | 1 | | | | 1 | | 1 | | | | | | | | | 1 | | | | | $b_{23}$ |
| | | | 1 | | 1 | 1 | | | | | | 1 | | | | | | | | 1 | | | | $b_{24}$ |
| | | | | 1 | 1 | | | 1 | 1 | | | | 1 | | | | | | | | 1 | | | $b_{34}$ |
| | | 1 | | | 1 | | | | 1 | | 1 | | 1 | | | | | | | | | 1 | | $b_{35}$ |
| | | | 1 | 1 | 1 | 1 | 1 | | | | | | 1 | | | | | | | | | | 1 | $b_{45}$ |
| | 3 | 3 | 2 | 2 | 3 | 3 | 3 | 2 | 2 | 3 | 3 | 2 | 2 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

2. Исходные данные к работе

Вариант 1

$$B = \begin{vmatrix} \infty & 20 & 0 & 0 & 20 & 0 \\ 20 & \infty & 25 & 15 & 0 & 0 \\ 0 & 25 & \infty & 15 & 20 & 0 \\ 0 & 15 & 15 & \infty & 20 & 15 \\ 20 & 0 & 20 & 20 & \infty & 20 \\ 0 & 0 & 0 & 15 & 20 & \infty \end{vmatrix}.$$

Между узлами: 1…4: 20 каналов;
2…6: 20 каналов;
2…6: 15 каналов.

Вариант 2

$$B = \begin{vmatrix} \infty & 15 & 0 & 0 & 15 & 0 \\ 15 & \infty & 20 & 10 & 0 & 0 \\ 0 & 20 & \infty & 10 & 15 & 0 \\ 0 & 15 & 10 & \infty & 15 & 10 \\ 15 & 0 & 15 & 15 & \infty & 15 \\ 0 & 0 & 0 & 10 & 15 & \infty \end{vmatrix}.$$

Между узлами: 2...6: 15 каналов;
3...6: 15 каналов;
1...6: 20 каналов.

Вариант 3

$$B = \begin{vmatrix} \infty & 20 & 0 & 0 & 20 & 0 \\ 20 & \infty & 25 & 15 & 0 & 0 \\ 0 & 25 & \infty & 15 & 20 & 0 \\ 0 & 15 & 15 & \infty & 20 & 15 \\ 20 & 0 & 20 & 20 & \infty & 20 \\ 0 & 0 & 0 & 15 & 20 & \infty \end{vmatrix}.$$

Между узлами: 1...6: 20 каналов;
2...5: 20 каналов;
5...6: 10 каналов.

Вариант 4

$$B = \begin{vmatrix} \infty & 15 & 0 & 0 & 15 & 0 \\ 15 & \infty & 20 & 10 & 0 & 0 \\ 0 & 20 & \infty & 10 & 15 & 10 \\ 0 & 10 & 10 & \infty & 15 & 15 \\ 15 & 0 & 15 & 15 & \infty & 0 \\ 0 & 0 & 10 & 15 & 0 & \infty \end{vmatrix}.$$

Между узлами: 2...6: 15 каналов;
1...6: 15 каналов;
1...3: 15 каналов.

Вариант 5

$$B = \begin{vmatrix} \infty & 15 & 0 & 0 & 15 & 0 \\ 15 & \infty & 20 & 10 & 0 & 10 \\ 0 & 20 & \infty & 10 & 15 & 15 \\ 0 & 10 & 10 & \infty & 15 & 0 \\ 15 & 0 & 15 & 15 & \infty & 0 \\ 0 & 10 & 15 & 0 & 0 & \infty \end{vmatrix}.$$

Между узлами: 5...6: 15 каналов;
1...4: 15 каналов;
4...6: 15 каналов.

## 3. СОДЕРЖАНИЕ ОТЧЕТА

Отчёт должен содержать
1. Исходные данные задачи.
2. Математическую формулировку задачи.
3. Симплекс-таблицу.
4. Результаты расчёта.
5. Схему организации каналов.

Литература

1. Введение в исследование операций : В 2 кн. Кн. 1 / Х. Таха; пер. с англ. – М., 1985.– 480 с.

2. Лазарев, В. Г. Динамическое управление потоками информации в сетях связи / В. Г. Лазарев, Ю. В. Лазарев. – М. : Радио и связь, 1983. – 216 с.

3. Майника, Э. Алгоритмы оптимизации на сетях и графах / Э. Майника. – М. : Мир, 1981. – 324 с.

4. Филлипс, Д. Методы анализа сетей / Д. Филлипс, А. Гарсия-Диас; пер. с англ. – М. : Мир, 1984. – 496 с.

Св. план 2007, поз. 129

Учебное издание

# УПРАВЛЯЮЩИЕ СИСТЕМЫ В ТЕЛЕКОММУНИКАЦИЯХ

Лабораторный практикум
для студентов специальностей
I-45 01 01 «Многоканальные системы телекоммуникаций»
и I-45 01 02 «Системы радиосвязи, радиовещания и телевидения»
всех форм обучения

Составитель
**Хацкевич** Олег Александрович

Редактор Е. Н. Батурчик
Корректор М. В. Тезина

| | | |
|---|---|---|
| Подписано в печать | Формат 60x84 1/16. | Бумага офсетная. |
| Гарнитура «Таймс». | Печать ризографическая. | Усл. печ. л. |
| Уч.-изд. л. 1,6. | Тираж 100 экз. | Заказ 13. |

Издатель и полиграфическое исполнение: Учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»
ЛИ №02330/0056964 от 01.04.2004. ЛП №02330/0131666 от 30.04.2004.
220013, Минск, П. Бровки, 6