

Пути на графе.
Транспортные сети.

Наиболее эффективный алгоритм решения задачи о кратчайшем ($s-t$)-пути первоначально дал Дейкстра [10]. В общем случае этот метод основан на приписывании вершинам временных пометок, причем пометка вершины дает верхнюю границу длины пути от s к этой вершине. Эти пометки (их величины) постепенно уменьшаются с помощью некоторой итерационной процедуры, и на каждом шаге итерации точно одна из временных пометок становится постоянной. Последнее указывает на то, что пометка уже не является верхней границей, а дает точную длину кратчайшего пути от s к рассматриваемой вершине. Опишем этот метод подробно.

Алгоритм Дейкстры. Этот алгоритм позволяет найти кратчайший путь между некоторой вершиной s и остальными вершинами при условии, что в графе нет дуг (ребер) с отрицательным весом. Рассмотрим основную идею алгоритма на примере графа G матрицей весов¹² C , представленных на рис. 4.7, взяв в качестве s вершину v_5 .

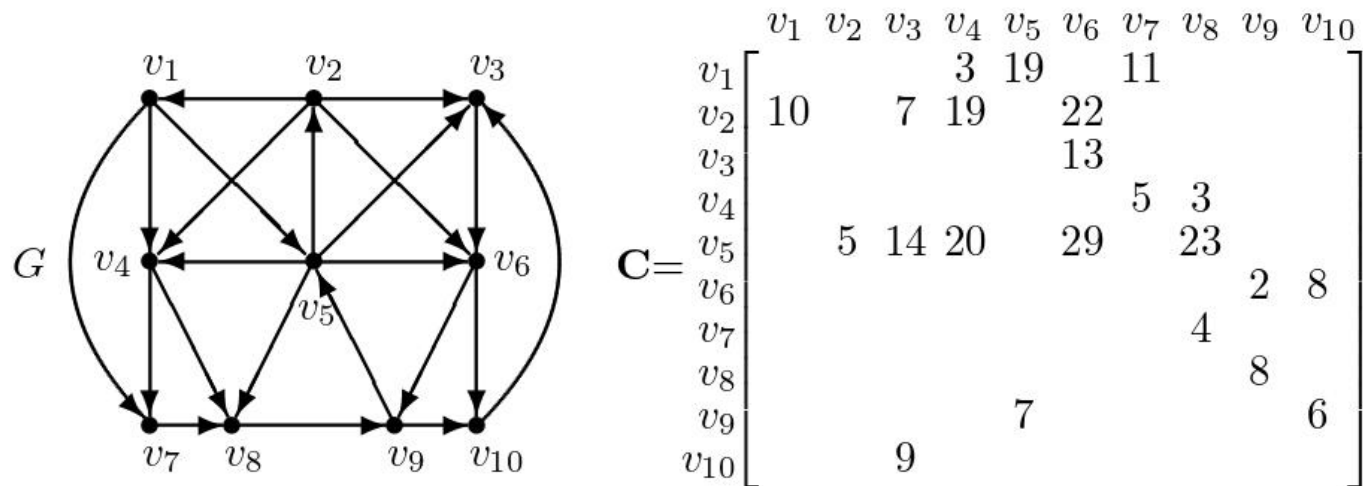


Рис. 4.7

Выделим все вершины $v_i \in \Gamma(v_2) = \{v_1, v_3, v_4, v_6\}$ и рассмотрим дуги (v_2, v_i) (изображены на рис. 4.9 тонкими линиями). Поскольку $l(v_2) + c(v_2, v_3) < l(v_3)$, "обходной" путь (v_5, v_2, v_3) короче, чем "прямой" (v_2, v_3) , поэтому за новое значение метки $l(v_3)$ примем сумму $l(v_2) + c(v_2, v_3) = 12$. Аналогичной проверкой устанавливаем, что прямой путь (v_5, v_4) короче, чем обходной (v_5, v_2, v_4) , следовательно, значение метки $l(v_4)$ сохраняется прежним. Для вершины v_6 , как и для v_3 , метка корректируется и принимает значение 27. Кроме вершин v_3, v_4 и v_6 , достижимых из v_5 непосредственно, есть еще одна вершина v_1 , единственный путь к которой лежит через постоянно помеченную v_2 . Поэтому в качестве значения метки для нее естественно взять сумму $l(v_2) + c(v_2, v_1)$. Будем

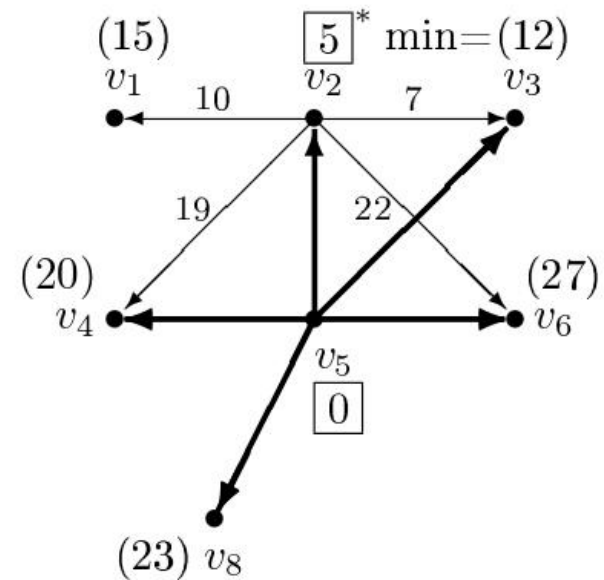


Рис. 4.9

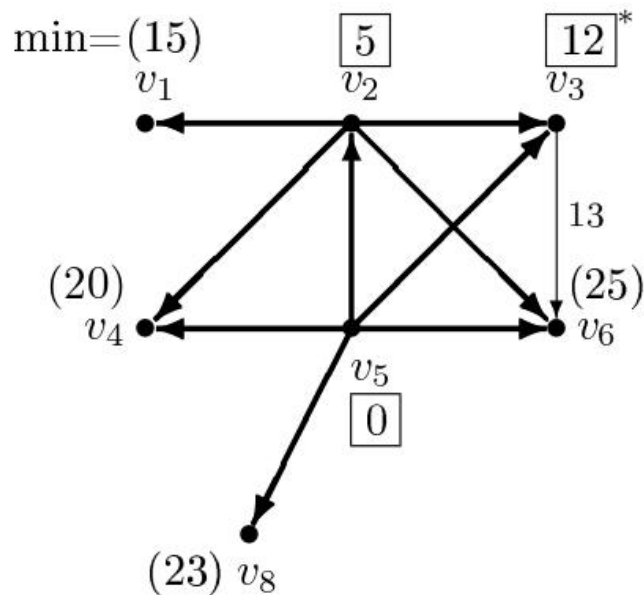


Рис. 4.10

единственная дуга, выходящая из v_3 . Наличие этой дуги требует пересчета метки $l(v_6)$. Так как $l(v_3) + c(v_3, v_6) < 27$, за новое значение $l(v_6)$ примем сумму $l(v_3) + c(v_3, v_6) = 25$. Минимальной среди временных меток является $l(v_1) = 15$. Это зна-

говорить, что теперь метки $l(v_1)$, $l(v_3)$ и $l(v_6)$ получены из вершины v_2 . Из пяти временных меток минимальной является $l(v_3) = 12$, которая и определяет длину кратчайшего пути до v_3 . Так как метка получена из v_2 , путь имеет вид (v_5, v_2, v_3) . Таким образом, задача решена и для v_3 , которая становится постоянно помеченной. Этот факт отражен на рис. 4.10. Кроме уже рассмотренных дуг, изображена и

чит, что выявлен кратчайший путь (v_5, v_2, v_1) , а метка вершины v_1 становится постоянной. Последующие действия выполняются аналогично. Для v_1 , как последней постоянно помеченной, находим множество вершин $\Gamma(v_1) = \{v_4, v_5, v_7\}$ и анали-

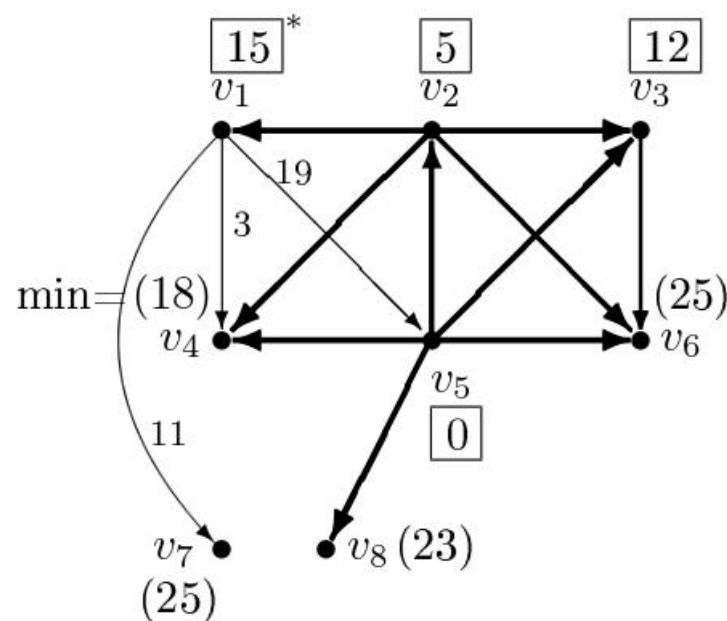


Рис. 4.11

зируем их метки, принимая во внимание длины дуг, связывающих v_1 с этими вершинами. Так как $l(v_1) + c(v_1, v_4) < l(v_4)$, метка вершины v_4 корректируется, становясь равной 18. У вершины v_5 метка постоянная и поэтому не меняется. Наконец, v_7 , до этого не рассматривавшаяся, получает начальную метку, равную $l(v_1) + c(v_1, v_7) = 25$. Граф, отражающий ситуацию, представлен на рис. 4.11. Здесь минимальной временной меткой оказывается $d(v_4) = 18$. Теперь вершина v_4 становится постоянно помеченной. Так как свою метку она получила из v_1 , кратчайший путь к ней состоит из кратчайшего пути до v_1 и дуги (v_1, v_4) и выглядит как (v_5, v_2, v_1, v_4) , а его длина равна 18.

Так как $l(v_1) + c(v_1, v_4) < l(v_4)$, метка вершины v_4 корректируется, становясь равной 18. У вершины v_5 метка постоянная и поэтому не меняется. Наконец, v_7 , до этого не рассматривавшаяся, получает начальную метку, равную $l(v_1) + c(v_1, v_7) = 25$. Граф, отражающий ситуацию, представлен на рис. 4.11. Здесь мини-

Итерационный процесс продолжается до тех пор, пока конечная вершина искомого кратчайшего пути не станет постоянно помеченной. Если же требуется найти кратчайшие пути до всех вершин, итерации выполняются, пока все вершины не получат постоянные метки.

Обработку данных в соответствии с алгоритмом целесообразно оформлять в виде таблицы (см. табл. 4.2), строки которой соответствуют итерациям, а столбцы — вершинам графа и отражают процесс изменения их временных меток вплоть до момента, пока временная метка не станет постоянной. Два дополнительных столбца таблицы p и $l(p)$ нужны, чтобы зафиксировать вершину, которая становится постоянно помеченной на очередной итерации, и длину кратчайшего пути до этой вершины.

В табл. 4.2 содержатся результаты расчетов для разобранного выше примера. Покажем, как она заполняется.

Таблица 4.2

Ите ра ции	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	p	$l(p)$
1	∞	∞	∞	∞	0	∞	∞	∞	∞	∞	v_5	0
2	∞	5	14	20		29	∞	23	∞	∞	v_2	5
3	15		12	20		27	∞	23	∞	∞	v_3	12
4	15			20		25	∞	23	∞	∞	v_1	15
5				18		25	25	23	∞	∞	v_4	18
6						25	23	21	∞	∞	v_8	21
7						25	23		29	∞	v_7	23
8						25			29	∞	v_6	25
9									27	33	v_9	27
10										33	v_{10}	33
<i>pred</i>	v_2	v_5	v_2	v_1	v_5	v_3	v_4	v_4	v_6	v_6		

На первой итерации полагаем временные метки для всех вершин, кроме начальной, равными ∞ , а для начальной – 0. Последней постоянно помеченной считаем вершину v_5 , а ее метку равной 0.

На второй итерации, используя матрицу C , находим все вершины множества $\Gamma(v_5)$ и пересчитываем их временные метки по формуле $l(v_i) = \min\{l(v_i), l(p) + c(p, v_i)\}$. Среди временных меток ищем минимальную. Это метка вершины v_2 , теперь ее считаем постоянно помеченной.

На третьей итерации находим вершины множества $\Gamma(v_2)$ и пересчитываем их временные метки. Среди всех временных вновь ищем минимальную метку. На этой итерации постоянно помеченной становится вершина v_3 , и т. д.

Процесс заканчивается на десятой итерации, когда постоянную метку получает v_{10} . Результаты — длины кратчайших путей — содержатся в двух последних столбцах таблицы.

Чтобы находить сами пути, сформируем еще одну строку таблицы — *pred*, в которой для каждой вершины графа указана ее предшественница на кратчайшем пути. С этой целью в каждом столбце отыскиваем самую нижнюю строку с предпоследним значением метки. Например, в первом столбце это вторая строка, в восьмом — пятая. Искомая вершина-предшественница содержится в столбце p найденной строки. Для v_1 это вершина v_2 , а для v_8 — вершина v_4 . Теперь для определения самого пути достаточно записать цепочку вершин-предшественниц, начиная с конечной вершины. Так, для v_{10} имеем: $pred(v_{10}) = v_6$, $pred(v_6) = v_3$, $pred(v_3) = v_2$ и $pred(v_2) = v_5$. Это значит, что кратчайший путь между v_5 и v_{10} имеет вид $(v_5, v_2, v_3, v_6, v_{10})$.

На графе с ребрами единичной длины рисунка 3.5 найти кратчайшие пути из вершины x_0 в вершины x_{14} , x_{17} и x_{18} .

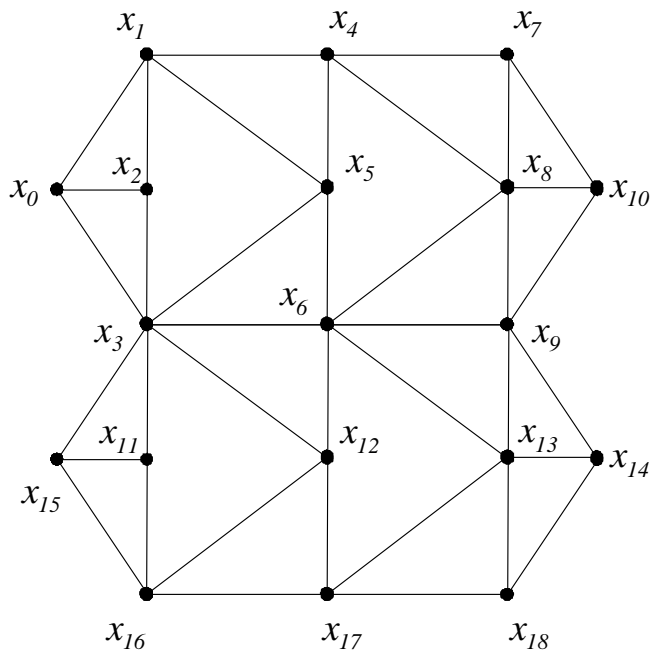


Рисунок 3.5 – Исходный граф к задаче 3.6

Транспортная сеть – особый вид графа, для которого характерны следующие свойства:

- 1) транспортная сеть – конечный граф без петель;
- 2) у графа G существует единственная вершина x_0 , такая, что $\Gamma^{-1}(x_0) = \emptyset$, т.е. в эту вершину не входит ни одна дуга;
- 3) у графа G существует единственная вершина z , такая, что $\Gamma(z) = \emptyset$, т.е. из этой вершины не исходит ни одна дуга;
- 4) с каждой дугой сопоставлено положительное целое число, называемое пропускной способностью дуги.

Постановка задачи о поиске максимального потока.

Дана транспортная сеть, на которой обозначены пропускные способности дуг. (рисунок 2.4). Поток, проходящий по дугам, можно сравнить с жидкостью, протекающей по системе труб, имеющих разный диаметр (разную пропускную способность). При этом вещество не накапливается в узлах сети; количество входящего в узел вещества равно количеству вещества, исходящего из узла.

Дуга сети называется **насыщенной**, если поток через нее совпадает с пропускной способностью дуги. Поток в сети называется **полным**, если в нем есть хотя бы одна насыщенная дуга. Задача состоит в нахождении максимально возможного потока для данной транспортной сети.

Задачу поиска максимального потока на транспортной сети удобно разделить на две подзадачи:

- а) нахождение полного потока (в том случае, если начальный поток через сеть нулевой);
- б) коррекцию полного потока до максимального в соответствии с алгоритмом Форда-Фалкерсона.

Алгоритм Форда-Фалкерсона состоит из двух этапов:

а) этапа разметки. Начальной вершине графа присваиваем метку 0. Далее вершине x_i присваиваем метку « + » и номер вершины $\Gamma^{-1}(x_i)$, т.е. номер той вершины, откуда мы пришли к вершине x_i , если по дуге $(\Gamma^{-1}(x_i), x_i)$ поток можно увеличить; или метку « - » и номер вершины $\Gamma^{-1}(x_i)$, т.е. номер той вершины, откуда мы пришли к вершине x_i , если по дуге $(\Gamma^{-1}(x_i), x_i)$ поток можно уменьшить (т.е. движение происходит против потока в дуге). Иначе вершина меток не получает. Получаем таким образом все возможные пути увеличения потока, рассматривая на каждом шаге новую вершину из множества непомеченных вершин.

б) Если вершина z , т.е. выход транспортной сети, оказалась помеченной, то необходимо построить путь, двигаясь по последовательности индексов-меток. На этом пути необходимо скорректировать поток, двигаясь от входа сети к выходу, увеличивая поток на 1, если дуга имеет знак «+», и уменьшая его, если дуга имеет знак « - ». Далее вновь переходим к этапу разметки, убирая метки от предыдущего этапа.

Если же при разметке вершина z не получила метку, то найденный поток является максимальным.

Ответом на решение задачи служит величина потока ϕ_{max} и пути его прохождения.

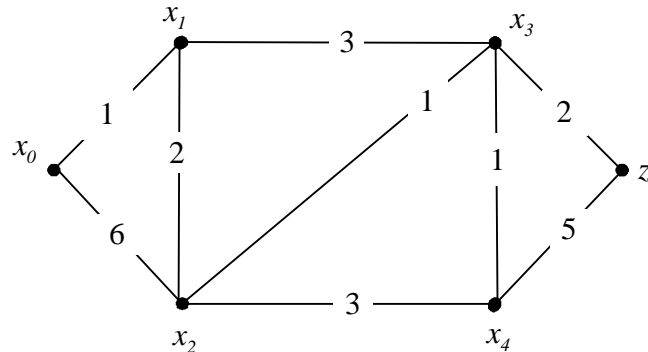


Рисунок 2.4 – Исходные данные к задаче 2.3

Решение:

1. Последовательно увеличиваем поток на транспортной сети, находя для этого соответствующие пути. Насыщенные дуги выделяем жирным, величина потока указывается над дугой.

$\mu_1 = (x_0, x_1, x_3, z)$, $\phi(\mu_1) = 1$ - рисунок 2.5.

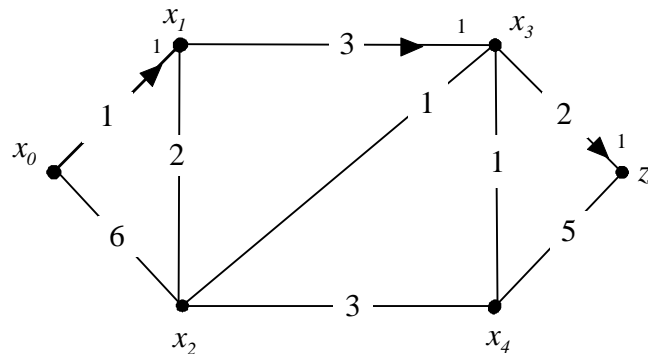


Рисунок 2.5 – Нахождение полного потока

2. Рассматриваем путь $\mu_2(x_0, x_2, x_4, x_3, z)$, $\phi(\mu_2) = 1$, $\phi = 2$ - рисунок 2.6.

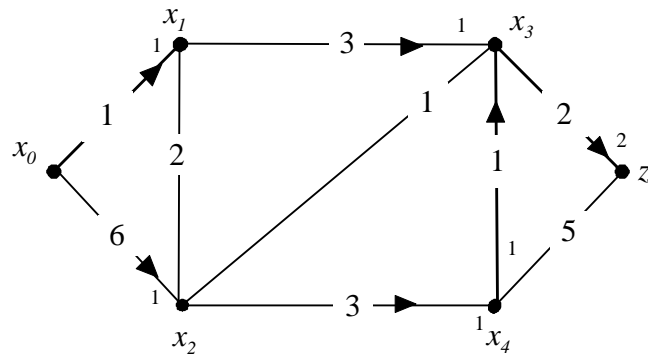


Рисунок 2.6 – Нахождение полного потока

3. Рассматриваем путь $\mu_3(x_0, x_2, x_4, z)$, $\phi(\mu_3) = 2$, $\phi = 4$ - рисунок 2.7.

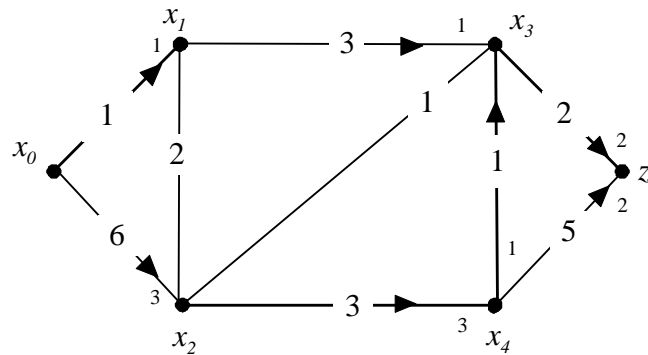


Рисунок 2.7 – Нахождение полного потока

4. На полученном на рисунке 2.7 графе производим разметку.

Сначала вершина x_0 получает метку 0. Смежными с ней являются вершины x_1 и x_2 . Вершина x_1 метку получить не может, так как дуга (x_0, x_1) является насыщенной, поток по ней увеличен быть не может, и развернуть поток невозможно. Вершина x_2 получает метку $+0$, так как по дуге (x_0, x_2) поток можно увеличить. (рисунок 2.8 а).

Переходим к вершине x_2 . Смежными с ней являются вершины x_1 , x_3 и x_4 . Вершина x_1 получает метку $+2$, так как по дуге (x_2, x_1) поток можно увеличить; вершина x_3 получает метку $+2$, так как по дуге (x_2, x_3) поток можно увеличить; вершина x_4 метки не получает, так как дуга (x_2, x_4) является насыщенной, и развернуть поток невозможно (рисунок 2.8 б).

Переходим к вершине x_1 . Все, смежные с ней, вершины имеют метки. Переходим к вершине x_3 . Смежными с ней являются вершины z и x_4 . Вершина z не

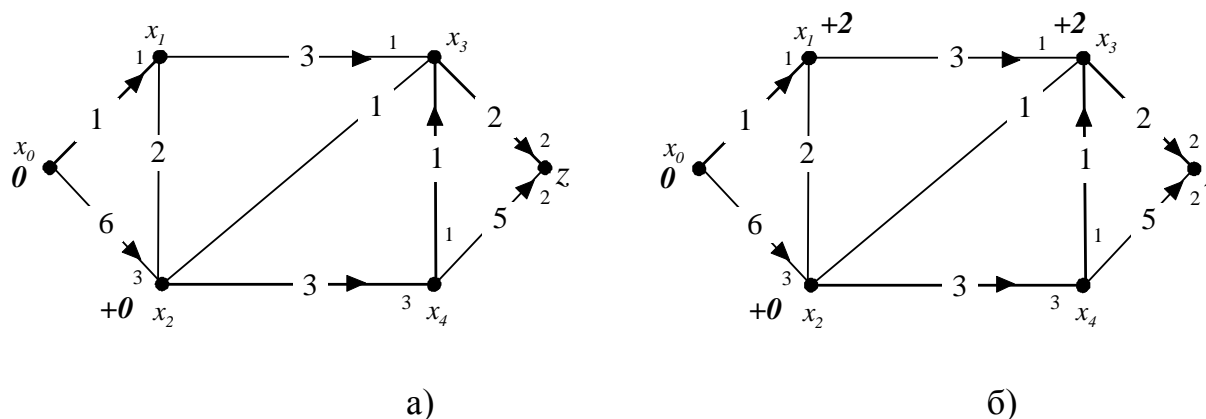


Рисунок 2.8 – Процесс разметки вершин графа

может быть помечена, так как дуга (x_3, z) является насыщенной, и развернуть поток невозможно. Вершина x_4 получает метку -3 , так как возможно идти против потока – рисунок 2.9.

Переходим к вершине x_4 . Смежной с ней непомеченной вершиной является вершина z , она может быть отмечена, так как по дуге (x_4, z) можно увеличить поток.

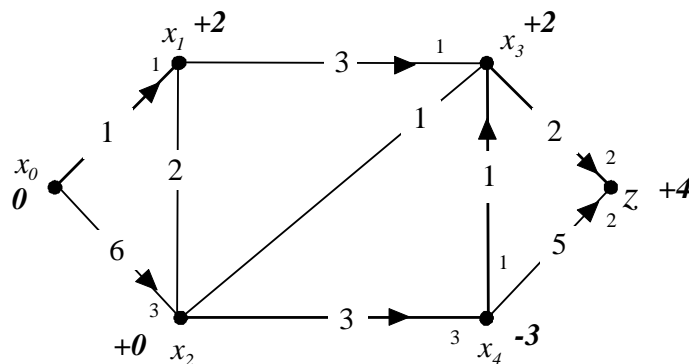


Рисунок 2.9 – Процесс разметки вершин графа

Вершина z оказалась отмеченной. Двигаясь по последовательности индекс-пометок $(+4, -3, +2, +0)$ получаем путь $\mu_4 = (x_0, x_2, x_3, x_4, z)$. Последовательно двигаясь по нему получаем:

- 1) в дуге (x_0, x_2) поток увеличиваем на 1, т.к. дуга имеет знак «+»;
- 2) в дуге (x_2, x_3) поток увеличиваем на 1, т.к. дуга имеет знак «+»;
- 3) в дуге (x_3, x_4) поток уменьшаем на 1, т.к. дуга имеет знак «-»;
- 4) в дуге (x_4, z) поток увеличиваем на 1, т.к. дуга имеет знак «+».

Суммарный поток получился равным 5, граф показан на рисунке 2.10, метки после коррекции потока стираются. Дуга (x_2, x_3) становится насыщенной.

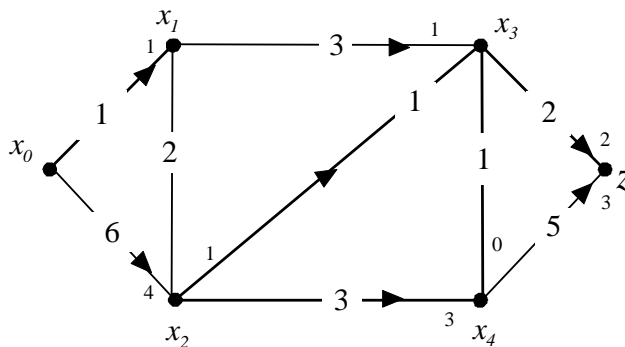


Рисунок 2.10 - Граф после коррекции потока

Вновь начинаем этап разметки.

Начальной вершине присваиваем метку 0; смежной с ней вершине x_2 присваиваем метку +0, так как по дуге (x_0, x_2) поток может быть увеличен.

Рассматриваем вершину x_2 . Смежными с ней являются вершины x_1 , x_3 и x_4 . Вершины x_3 и x_4 помечены быть не могут, так как соответствующие дуги являются насыщенными, а вершина x_1 получает метку +2. Продолжая разметку, получаем граф рисунка 2.11.

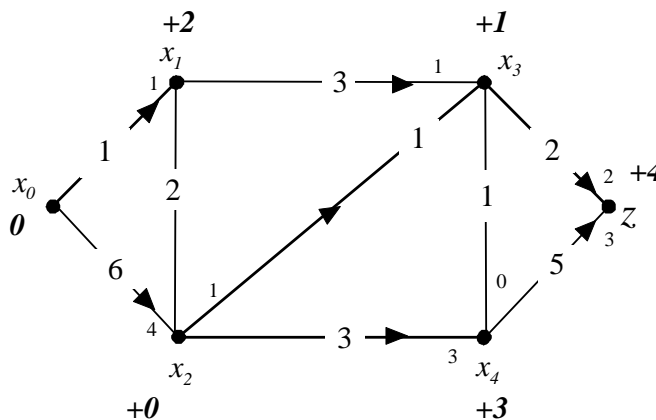


Рисунок 2.11 - Граф после разметки вершин

Вершина z получила метку, значит, поток по транспортной сети можно увеличить.

Двигаясь по последовательности индексов-пометок $(+4, -3, +1, +2, +0)$ получаем путь $\mu_5 = (x_0, x_2, x_1, x_3, x_4, z)$. Последовательно двигаясь по нему получаем:

- 1) в дуге (x_0, x_2) поток увеличиваем на 1, т.к. дуга имеет знак «+»;
- 2) в дуге (x_2, x_1) поток увеличиваем на 1, т.к. дуга имеет знак «+»;
- 3) в дуге (x_1, x_3) поток увеличиваем на 1, т.к. дуга имеет знак «+»;
- 4) в дуге (x_3, x_4) поток увеличиваем на 1, т.к. дуга имеет знак «+»;
- 5) в дуге (x_4, z) поток увеличиваем на 1, т.к. дуга имеет знак «+».

Суммарный поток получился равным 6, граф показан на рисунке 2.12, метки после коррекции потока стираются. Дуга (x_3, x_4) становится насыщенной.

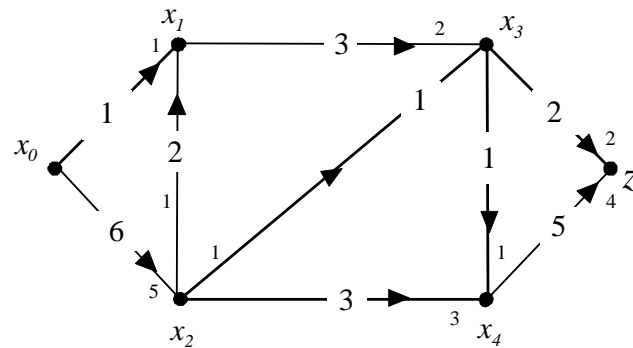


Рисунок 2.12 - Граф после коррекции потока

Возвращаемся к этапу разметки. Граф после выполнения разметки показан на рисунке 2.13.

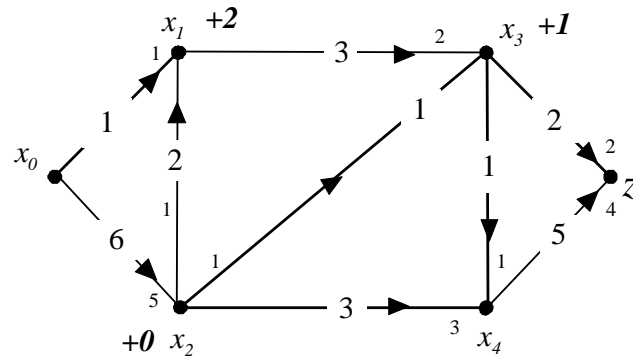
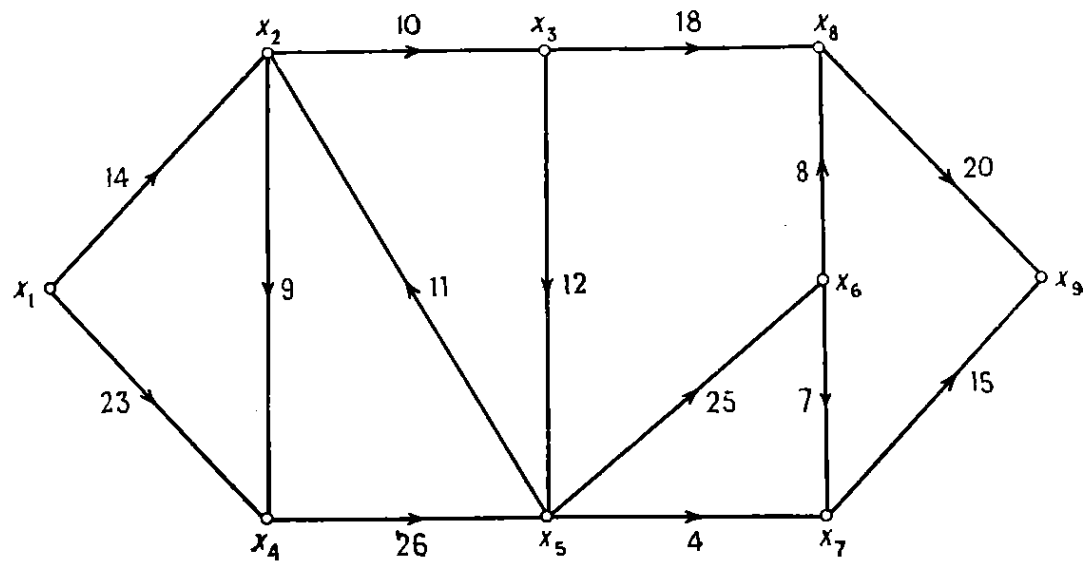


Рисунок 2.13 - Граф после этапа разметки

На рисунке 2.13 после разметки вершина z не получила пометки. Значит, полученный поток является максимальным - $\phi_{max} = 6$.



3.7. На транспортных сетях рисунка 3.6 найти максимальные потоки.

