**Содержание:**

Введение 2

1. Основные понятия теории графов. 3

2. Матричные способы задания графов. 4

3. Упорядочение элементов орграфа. 6

4. Постановка задачи о максимальном потоке. Основные определения. 8

5. Разрез на сети. 11

6. Алгоритм решения задачи о максимальном потоке. 13

Заключение. 20

Список использованной литературы 21

# Введение

В последнее время в различных областях знаний широко применяется теория графов. С помощью теории графов хорошо описываются задачи экономической и планово-производственной практики, как, например, календарное и сетевое планирование и управление, автоматизация управления производством, рационализация схем перевозок и грузопотоков, оптимальное размещение производства т.п.

**1. Основные понятия теории графов**

Основным объектом этой теории является граф. Наглядно граф можно представить как некоторое множество точек плоскости или пространства и множество отрезков кривых или прямых линий, соединяющих все или некоторые из этих точек. Формально граф G определяется заданием двух множеств X и U и обозначается G=(XU). Элементы xB1B, xB2B, …, xBn B множества X называются вершинами и изображаются точками. Элементами uB1B, uB2B, …, uBm множества U являются пары связанных между собой элементов множества X и изображаются отрезками. Взаимное расположение, форма и длины отрезков значения не имеют. Если в паре вершин xBi Bи B BxBj Bуказано направление связи, то есть указано, какая вершина является первой, то отрезок UB1 Bназывается дугой, а если ориентация не указана, - ребром.

Если в графе G все элементы множества U изображаются дугами, то граф называется ориентированным (орграф), если ребрами, - неориентированным.

На практике используются графы, в которых множества X и U состоят из конечного числа элементов, то есть конечные графы. На рисунке располагать вершины графа можно произвольно. Также произвольно выбирается и форма соединяющих их линий. Поэтому один и тот же граф (граф с одной и той же информацией) может быть представлен в различных видах. Такие графы называются изоморфными.

Смежными называют вершины графа, если они различны и существует дуга (ребро), которая соединяет эти вершины. Если две дуги (ребра) имеют общую концевую вершину, то такие дуги (ребра) смежные.

Последовательность дуг, в которой конец каждой предыдущей дуги совпадает с началом следующей, называется путем в орграфе. Путь, проходящий через все вершины и при этом только один раз, называется гамильтоновым. Путь, включающий все дуги графа, и при этом только по одному разу, называется эйлеровым. Полным путем называют любую непрерывную последовательность вершин и дуг, идущих от начальной вершины к конечной. Конечный путь, у которого начальная вершина совпадает с конечной, называется контуром.

Граф называется связным, если для любых двух его вершин существует путь, их соединяющий; в противном случае граф называется несвязным.

В экономике чаще всего используются два вида графов: дерево и сеть.

Дерево представляет собой связный граф без циклов, имеющий исходную вершину (корень) и крайние вершины; пути от исходной вершины к крайним вершинам называются ветвями.

Если граф, вообще говоря, не связный, не содержит циклов, то каждая связная его часть будет деревом. Такой граф называется лесом.

В приложении теории графов к практическим задачам дугам (ребрам) графа сопоставляют какие-то числовые характеристики. Например, пропускная способность транспортной магистрали, запас какого-либо ресурса, продолжительность выполнения какой-либо работы и т.д. Таким образом, дугам графа приписаны определенные веса и граф G называется взвешенным.

# 2. Матричные способы задания графов

Граф может быть представлен как в виде рисунка, так и в виде матрицы. Это бывает необходимо при большом числе вершин и дуг (ребер), когда рисунок теряет свою наглядность. Матричное представление графов используется при исследовании его на ЭВМ.

Пусть имеется граф G, не содержащий петель, имеющий n вершин и m дуг (ребер). Графу G можно сопоставить матрицу инциденций графа G. Строки m этой матрицы соответствуют вершинам, столбцы n – дугам (ребрам) графа. Если граф ориентированный, то элементы aBij Bматрицы инциденций равны: (-1), если дуга u исходит из i-й вершины; (+1), если дуга заходит в i-ю вершину. Если граф неориентированный, то элементами матрицы будут 1 и 0. На рис. 1.2 показаны орграф и его матрица инциденций, на рис. 1.3. - неориентированный граф и матрица инциденций.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | uB1B | uB2B | uB3B | uB4B | uB5B | uB6B | uB7B |
| *x*B*1*B | -1 | 0 | 0 | 0 | -1 | -1 | 0 |
| *x*B*2*B | 1 | -1 | 0 | 0 | 0 | 0 | 0 |
| *x*B*3*B | 0 | 0 | 0 | 1 | 1 | 0 | -1 |
| *x*B*4*B | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| *x*B*5*BBB | 0 | 0 | -1 | 0 | 0 | 1 | 1 |

Рис.1.2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | uB1B | uB2B | uB3B | uB4B | uB5B | uB6B | uB7B |
| *x*B*1*B | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| *x*B*2*B | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| *x*B*3*B | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| *x*B*4*B | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| *x*B*5*B | 0 | 0 | 1 | 0 | 0 | 1 | 1 |

Рис.1.3

Для ориентированных и неориентированных графов можно сформировать матрицу смежности вершин. Пусть орграф G содержит n вершин. Его матрица смежности представляет собой квадратную матрицу n-го порядка. Строки и столбцы этой матрицы соответствуют вершинам орграфа G. Элементы uBij Bесть число дуг, выходящих из i-й вершины и входящий в j-ю. В орграфе, не содержащем параллельных дуг, элементами матрицы будут 1 и 0. На рис. 1.4. представлен орграф и его матрица смежности вершин. Как видно из рис. 1.5, у неориентированного графа матрица смежности вершин будет симметрической. По матрице смежности вершин определяется степень вершины, т.е. число дуг, пересекающихся в этой вершине. Число входящих в i-ю вершину дуг равно сумме элементов i- го столбца; число дуг, исходящих из данной вершины, равно сумме элементов i-й строки.



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | *x*B*1*B | *x*B*2*B | *x*B*3*B | *x*B*4*B | *x*B*5*B | *x*B*6*B | *x*B*7*B |
| *x*B*1*B | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| *x*B*2*B | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| *x*B*3*B | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| *x*B*4*B | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| *x*B*5*B | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| *x*B*6*B | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| *x*B*7*B | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

Рис.1.4



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *x*B*1*B | *x*B*2*B | *x*B*3*B | *x*B*4*B |
| *x*B*1*B | 1 | 1 | 1 | 0 |
| *x*B*2*B | 1 | 0 | 1 | 1 |
| *x*B*3*B | 1 | 1 | 0 | 1 |
| *x*B*4*B | 0 | 1 | 1 | 1 |

Рис.1.5

Граф G также может быть задан матрицей смежности дуг (ребер). Предположим, что некоторый граф имеет m дуг (ребер). Его матрица смежности дуг (ребер) представляет собой квадратную матрицу m-го порядка. Строки и столбцы матриц соответствуют дугам (ребрам) графа. Элементами pBij Bматрицы для ориентированного графа буду 1 и 0. Если дуга uB1 Bпрямо предшествует дуге BuBjB,B Bто pBij B= 1 и 0 – в остальных случаях. Матрица смежности ребер неориентированного графа своими элементами также имеет 1 и 0. Элемент pBijB = 1, если ребра uBi Bи uBj Bсмежные, и 0 – в остальных случаях.

**3. Упорядочение элементов орграфа**

Существенно повысить наглядность графа и упростить расчет его числовых характеристик позволяет процедура упорядочения его элементов. При упорядочении устанавливается порядок следования вершин или дуг. Вершина xBiB предшествует вершине xBj B, если существует путь из xBi Bв xBjB. Другими словами, вершина xBi B- предшествующая («предок»), а вершина xBjB – последующая («потомок»). Необходимо выявить группы вершин по следующим правилам:

1) вершины первой группы не имеют предшествующих, а вершины последней группы не имеют последующих;

2) вершины любой промежуточной группы не имеют предков;

3) вершины одной и той же группы дугами не соединяются.

Упорядочение дуг происходит по тем же правилами. Полученный в результате упорядочения граф является изоморфным исходному. Упорядочение элементов графа можно выполнить графическим или матричным способом. Графический способ упорядочения ведется по следующему алгоритму (алгоритм Фалкерсона).

Мысленно вычеркивают все вершины и дуги, из них выходящие. В получившемся графе найдется хотя бы одна вершина, в которую не входит ни одна дуга. Этой вершине присваивается следующий по порядку номер. Сама вершина (или несколько вершин) образует вторую группу. Описанную процедуру повторяют до тех пор, пока все вершины не будут упорядочены и разобраны по группам.

Дуги орграфа упорядочивают аналогичным образом. Сначала находят дуги, не имеющие предков, и включают их в первую группу. Затем мысленно вычеркивают дуги первой группы. Затем вновь выделяют дуги, не имеющие предшествующих, и включают их во вторую группу. И так далее, пока все дуги не будут разбиты на группы. Упорядоченным дугам присваивают новые обозначения, например, цифрами натурального ряда.

Пример. Упорядочить вершины графа и построить граф, изоморфный данному.

Решение. Просматривая граф, замечаем, что вершина xB6B не имеет предшествующих, так как в нее не входит ни одна дуга. Эта вершина относится к первой группе. Подобных вершин в данном графе больше нет. Исключим из рассмотрения эту вершину и дуги, из них выходящие. На рис. 1.6 пометим их одной черточкой. В оставшемся графе снова ищем вершины, в которые не входит ни ода дуга. Таковыми будут xB4 Bи xB7B. Эти вершины образуют вторую группу. Обозначим второе вычеркивание двумя черточками. И так далее. На рис. 1.7 представлен упорядоченный граф. Вертикальными линиями показаны группы разбиения и на них отложены вершины, принадлежащие соответствующим группам. Далее эти вершины соединены, как на исходном графе.

Рис.1.6 Рис.1.7

**4. Постановка задачи о максимальном потоке. Основные определения**

К задаче о максимальном потоке сводятся многие важные оптимизационные задачи, например: задача об оптимальном назначении; различные задачи организации снабжения; задачи строительства энергетических сетей, нефте- и газопроводов, железных и шоссейных дорог и много других прикладных задач. В таких задачах схема доставки груза, или схема сообщения, представляется в виде графа, по ребрам которого проходят заданные потоки. Основным в теории потоков является понятие сети.

Сеть — это взвешенный граф без циклов и петель, ориентированный в одном направлении от вершины, называемой источником, к вершине, называемой стоком.

На рис. 1.8 представлена сеть. Истоком I является вершина 1, стоком S –вершина 6. Представим, что по ребрам (i,j) сети направляется некоторое вещество (ресурс, информация и т.п.). В скобках указаны пропускные способности ребер. Первое число- это пропускная способность в прямом направлении (от вершины i к вершине j) и второе – в противоположном направлении.

Пропускная способность - максимальное количество вещества rBijB, которое можно пропустить по ребру (i,j). Количество xBij Bвещества, проходящего по ребру (i,j) в единицу времени, называется потоком по ребру. Считается, что если есть поток из вершины i в вершину j, то

xBijB = - xBji B (1.1)

Если поток по ребру (i,j) меньше его пропускной способности, т.е. xBij B< rBijB, то ребро называют ненасыщенным, если xBij B= rBijB, - насыщенным. Совокупность X = {xBijB} потоков по всем ребрам называют потоком по сети или просто потоком.

Поток по ребру не может превысить его пропускной способности, т.е.

 (1.2)

где n – количество вершин сети.

Приведем основное положение в теории потоков, которое называют условием сохранения потока: для любой вершины, кроме истока I и стока S, количество поступающего вещества в эту вершину, равно количеству вещества, вытекающего из нее. В промежуточных вершинах потоки не создаются и не исчезают.

Математическая запись этого ограничения с учетом формулы (1.1) следующая:

 (1.3)

Отсюда вытекает, что общее количество вещества, вытекающего из истока I, совпадает с общим количеством вещества, поступающего в сток S, т.е.

 (1.4)

где j – конечные вершины ребер, исходящих из I;

i – начальные вершины ребер, входящих в S.

Линейную функцию f называют мощностью потока на сети. Сформулируем задачу о максимальном потоке: найти множество XP\* P= {xBijPB\*P} потоков xBijPB \*P по всем ребрам (i,j) сети, которое удовлетворяет условиям (1.1) - (1.3) и доставляет линейной функции (1.4) максимальное значение.

Как видно из формулировки, это типичная задача линейного программирования.

Обратимся к рис. 1.8, где изображена сеть. Для этой сети пропускную способность представим в виде матрицы, используя цифры в скобках. Здесь n = 6 и числа образуют квадратную матрицу шестого порядка (табл. 1.1). Если вершины k и l не соединены, то rBkl B= rB lkB = 0. Сформировать матрицу чисел xBijB – это значит задать поток на сети, т.е. найти nP2 Pчисел, удовлетворяющих условиям (1.1) – (1.3). Рассмотрим полный путь 1 – 2 – 5 – 6.

Пропускная способность этого пути не больше 1 ед. и ограничивается ребром (2, 5), которое лежит на этом пути. Поток по этому пути мощностью в 1 ед. будет допустимым, и условии (1.1) – (1.3) должны выполняться для всех вершин и ребер этого пути.

Таблица 1.1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| i/j | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 0 | 3 | 6 | 2 | 0 | 0 |
| 2 | 5 | 0 | 0 | 0 | 1 | 0 |
| 3 | 6 | 0 | 0 | 3 | 0 | 4 |
| 4 | 7 | 0 | 9 | 0 | 4 | 0 |
| 5 | 0 | 1 | 0 | 2 | 0 | 5 |
| 6 | 0 | 0 | 1 | 0 | 8 | 0 |

Рис. 1.8

Например, возьмем вершину 2 и проверим условие (1.3):

xB21B + xB25B = (-xB21B) + xB25B = (-1) + 1 = 0.

**5. Разрез на сети**

Представим некоторую сеть. Разобьем множество вершин сети на два непересекающихся подмножества A и B так, чтобы исток I попал в подмножество A, а сток S попал в подмножество B. В результате такого разбиения появляются ребра (i, j), конечные точки которых оказываются в разных подмножествах.

Совокупность ребер (i, j), начальные точки которых принадлежат подмножеству A, а конечные точки – подмножеству B, называют разрезом сети и обозначают A/B.

На рис. 1.9 изображена некоторая сеть. Стрелки указывают положительное направление потока. На сети произведено два разреза: I и II. При разрезе I образовалось два подмножества вершин сети: подмножество A = {1, 2} и B = {3, 4, 5}, а ребрами, образующим разрез, стали (1, 3), (1, 4), (2, 4). При разрезе II образовались подмножества A = {1, 2, 3, 4} и B = {5} с образующими разрез ребрами (3, 5) и (4, 5).

Рис. 1.9

Величина , представляющая сумму пропускных способностей всех ребер разреза, называется пропускной способностью разреза.

Если на сети задан поток X = {xBijB} и разрез (A/B), то величина , представляющая сумму потоков по всем ребрам разреза, называется потоком через разрез.

Для разреза I R(I) = rB13B + rB14 B+ rB24B = 6 + 2 + 1 = 9. X(I) = xB13B + xB14B + xB24B = 4 + 2 + 0 = 6. Для разреза II – R(II) = 9, X(II) = 6.

В общем случае, если на сети задан поток X = {xij} и произведен разрез (A/B), то хотя бы одно ребро любого полного пути, идущего из истока в сток, будет обязательно принадлежать разрезу (A/B). Напомним, что величина потока по любому полному пути не превышает пропускную способность каждого его ребра, а потому величина X суммарного потока, стремящегося из истока в сток, не может повысить пропускную способность любого разреза сети, т.е.

 (1.5)

В теории потоков утверждается, что если удастся построить на сети поток XP\* P= {xBijPB\*P}, величина которого равна пропускной способности некоторого разреза (A/B), то этот поток будет максимальным, а разрез обладать минимальной пропускной способностью. Ниже приводится теорема о максимальном потоке, имеющая большое прикладное значение.

Теорема Форда - Фалкерсона. На любой сети сети максимльная величина потока из истока I в исток S равна максимальной пропускной способности разреза, отделяющего I от S.

**6. Алгоритм решения задачи о максимальном потоке**

В разделе 4 был проведен расчет мощности потока, но ничего не было сказано о том, будет ли этот поток максимальным. Чтобы ответить на этот вопрос, необходимо исследовать этот поток.

Пусть задан некоторый поток X = {xij}. Разобьем сеть таким образом, чтобы к подмножеству A отошли исток I и все те вершины i, которые достигаются из истока I хотя бы по одному пути, состоящему из ненасыщенных ребер; к подмножеству B отнесем вершины, которые нельзя достичь из истока по ненасыщенным ребрам. При таком разбиении возникают две ситуации:

1) сток ;

2) сток .

Рассмотрим случай 1. Если , то

Построенное разбиение является разрезом A/B. По условию разбиения для любой вершины существует путь из истока в i, состоящий из ненасыщенных ребер, а для любой вершины такого пути нет. Отсюда следует, что любое ребро (i,j) разреза A/B будет насыщенным (иначе j принадлежало бы A), т.е. xij = rij. Просуммируем все эти равенства по всем и всем и получим

 (1.6)

В этом равенстве слева – величина X потока через разрез, справа – пропускная способность R разреза A/B. Из равенства (1.6) по теореме Форда - Фалкерсона следует, что поток X = {xij} является максимальным.

Рассмотрим случай 2. Если то существует путь из ненасыщенных ребер, ведущий из истока в сток. По ребрам этого пути можно пропустить дополнительный поток величиной , где минимум берется по всем ребрам, входящими в этот путь. Потоки xBijB по всем остальным ребрам остаются без изменения. В результате мощность суммарного потока возрастет на величину и это будет новый поток X = {xBijPB1P}.

При объединении двух рассмотренных случаев просматривается следующий алгоритм построения максимального потока:

1. Построить некоторый начальный поток XP0P = {xBijPB0P}.
2. Найти подмножество A вершин, достижимых из истока I по ненасыщенным ребрам. Если в этом процессе сток S не попадет в подмножество A, то построенный поток максимальный и задача решена. Если же сток S попал в A, то перейти к п. 3 алгоритма.
3. Выделить путь из истока I в сток S, состоящий из ненасыщенных ребер, и увеличить поток xBij Bпо каждому ребру этого пути на величину , где минимум берется по ребрам (i,j) упомянутого пути. Это означает, что будет построен новый поток XP1P = {xBijPB1P}. После этого надо вернуться к п. 2 данного алгоритма.

Замечание: при выполнении п. 3 алгоритма на каждом шаге по крайней мере одно из ненасыщенных раннее ребер становится насыщенным. Поскольку число ребер в сети конечно, то через конечное число шагов максимальный поток будет построен.

Разберем на примере предложенный алгоритм.

Рис. 1.10

На рис. 1.10 изображена сеть с истоком I в вершине (1) и стоком в вершине (6). В скобках проставлена пропускная способность ребер в прямом и обратном направлении. В табл. 1.2 показана матрица пропускных способностей сети.

В соответствии с п. 1 алгоритма сформируем на сети первоначальный поток XP0P. В этом потоке по пути 1 – 3 – 5 – 6 перемещаются 2 ед., поскольку ограничивает величину потока ребро (3, 5); по пути 1 – 2 – 5 – 6 перемещается 1 ед., лимитирующим является ребро (2, 5); по пути 1 – 4 – 6 – 2 – 2 ед., и ребро (1, 4) становится насыщенным. Матрица потока представлена на табл. 1.3.

Мощность потока XP0 Pрассчитаем по формуле (1.4).

f = xB12B + xB13 B+ xB14B = xB46B + xB56B = 1 + 2 + 2 = 2 + 3 = 5.

Таблица 1.2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| i/j | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 0 | 7 | 4 | 2 | 0 | 0 |
| 2 | 3 | 0 | 8 | 4 | 1 | 0 |
| 3 | 6 | 8 | 0 | 0 | 2 | 0 |
| 4 | 5 | 9 | 0 | 0 | 8 | 4 |
| 5 | 0 | 5 | 2 | 3 | 0 | 5 |
| 6 | 0 | 0 | 0 | 6 | 7 | 0 |

Таблица 1.3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| i/j | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 0 | 1 | 2 | 2 | 0 | 0 |
| 2 | -1 | 0 | 0 | 0 | 1 | 0 |
| 3 | -2 | 0 | 0 | 0 | 2 | 0 |
| 4 | -2 | 0 | 9 | 0 | 4 | 0 |
| 5 | 0 | -1 | -2 | 0 | 0 | 3 |
| 6 | 0 | 0 | 0 | -2 | -3 | 0 |

Для выполнения п. 2 алгоритма рассчитаем матрицу R – XP0P, приведенную в табл. 1.4. Элементы (rij – xij0) этой матрицы позволяют судить о насыщенности ребер сети. Нулевые элементы соответствуют насыщенным ребрам, ненулевые – ненасыщенным. С помощью матрицы R – XP0P можно сформировать подмножество A вершин, в которое можно попасть из истока I, двигаясь только по ненасыщенным путям, выделить их (если поток XP0 P не максимален) и с их помощью увеличить мощность потока.

Таблица 1.4

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| i/j | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 0 | 3 | 2 | 2 | 0 | 0 |
| 2 | -3 | 0 | 0 | 2 | 1 | 0 |
| 3 | -2 | 0 | 0 | 0 | 2 | 0 |
| 4 | -2 | -2 | 0 | 0 | 0 | 4 |
| 5 | 0 | -1 | -2 | 0 | 0 | 3 |
| 6 | 0 | 0 | 0 | -4 | -3 | 0 |

Таблица 1.5

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| i/j | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 0 | 6 | 2 | 0 | 0 | 0 |
| 2 | 4 | 0 | 8 | 4 | 0 | 0 |
| 3 | 8 | 8 | 0 | 0 | 0 | 0 |
| 4 | 7 | 9 | 0 | 0 | 8 | 2 |
| 5 | 0 | 6 | 4 | 3 | 0 | 2 |
| 6 | 0 | 0 | 0 | 8 | 10 | 0 |

Вершины подмножества A выделяют постепенно, начиная с истока I. Для этого просматривают первую строку матрицы R – XP0 P( в общем случае строку I) и выписывают номера вершин iB1B, iB2B, …, iBkB, соответствующих ненулевым элементам стоки. Это и будут вершины, в которые можно попасть из истока, перемещаясь по ненасыщенным ребрам. Запишем выявленные вершины в виде списка

Далее рассматривают каждую из вершин iBk Bполученного списка и составляют для нее свой список аналогичным образом. При этом вершины, встречающиеся в прежних списках, повторно не выписываются.

Если в этом процессе сток S не встретится, то поток максимален и задача решена; если же, напротив, при составлении очередного списка в нем появится сток S, то поток не максимален и его мощность можно увеличить. Для этого выделяют ненасыщенный путь из истока в сток. Построение ненасыщенного пути из I в S начинают с последнего ребра этого пути. Этим ребром будет (i Bn-1B, S), где i Bn-1 B– вершина, в список которой попал сток S. Далее находят ребро (i Bn-2B, i Bn-1B), где i Bn-2B - вершина, в список которой попала вершина i Bn-1B, и т.д., пока не встретится ребро (I, iB1B), которым начинается искомый ненасыщенный путь.

В данном примере I = 1, S = 6. Построим подмножество A, начиная с вершины. В табл. 1.4 в первой строке ненулевые элементы стоят во втором и третьем столбцах. Следовательно, запишем .

Последовательно составляем списки вершин 2 и 3. Во второй строке матрицы три элемента отличны от нуля: 4, 8, 4. Цифры 4 и 8 соответствуют вершинам 7 и 3, которые уже вошли в подмножество A, поэтому эти вершины повторно в списки не включаем. В четвертом столбце цифра 4 встречается впервые, поэтому включаем ее в список вершины . Переходим к вершине 3. В третьей строке матрицы R – XP0 Pненулевым элементам соответствуют вершины 1 и 2, которые уже встречались в списках. Значит, список вершины 3 будет пустым: … Аналогичным образом составляется список вершины . Повторим полученный набор списков:

(1.7)

Просматривая списки, замечаем что сток (вершина 6) попал в подмножество A. Значит поток XP0 не максимален и существует путь из истока в сток, состоящий из ненасыщенных ребер.

Перейдем к п. 3 алгоритма – выделению ненасыщенных ребер пути из истока в сток и преобразованию имеющегося потока в новый поток большей мощности. В списке (1.7) последним ребром (i n-1, S) является (4, 6), ребром (i Bn-2B, i Bn-1B) – ребро (2, 4), ребром (I, iB1B) – ребро (1, 2). Найденный путь по ненасыщенным ребрам из истока 1 в сток 6 пройдет через вершины 1, 2, 4 и 6.

Поскольку ненасыщенный путь найден, то следующим шагом является определение с помощью матрицы R – XP0 величины , на которую можно увеличить поток по каждому ребру (i, j) выделенного пути, чтобы получить новый поток X1 мощности, большей на единиц. Как видно из табл. 1.4., по ребру (1, 2) можно дополнительно пропустить 6 ед., по ребру (2, 4) – 4 ед., по ребру (4, 6) – только 2 ед., так что

Для построения матрицы нового потока X1 к соответствующим элементам xij0 матрицы XP0 прибавляется найденное значение , после чего возвращаются к п. 2 алгоритма, и т.д. до получения максимального потока.

Прибавим величину к элементам x120 = 1, x240 = 0 и x460 = 2, обозначающим ненасыщенный путь (по всем остальным ребрам величина потока не изменится). Приходим к матрице нового потока X1 (см. табл. 1.5), мощность которого равна 7 ед. Этот поток вновь надо исследовать на оптимальность, т. е. вернуться к п. 2 алгоритма. Вновь составляем матрицу R – X1 (см. табл. 1.5), а по ней – списки вершин, достижимых из истока I по ненасыщенным путям:

Из этих списков видно, что сток 6 снова в подмножестве A, а путь, ведущий в него, состоит из ненасыщенных ребер (1, 2), (2, 4), (4, 5), (5, 6). Новый поток X2, (его матрица представлена в табл. 1.6, получается преобразованием потока X1, если увеличить на потоки по указанным ребрам найденного ненасыщенного пути. Мощность нового потока X2 составляет 9 ед. Для исследования этого потока составляется матрица R – X2 (табл. 1.8), а по ней – списки.

 (1.8)

Таблица 1.6

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| i/j | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 0 | 5 | 2 | 2 | 0 | 0 |
| 2 | -5 | 0 | 0 | 4 | 1 | 0 |
| 3 | -2 | 0 | 0 | 0 | 2 | 0 |
| 4 | -2 | -4 | 0 | 0 | 2 | 4 |
| 5 | 0 | -1 | -2 | -2 | 0 | 5 |
| 6 | 0 | 0 | 0 | -4 | -5 | 0 |

Таблица 1.7

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| i/j | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 0 | 4 | 2 | 0 | 0 | 0 |
| 2 | 6 | 0 | 8 | 2 | 0 | 0 |
| 3 | 8 | 8 | 0 | 0 | 0 | 0 |
| 4 | 7 | 11 | 0 | 0 | 8 | 0 |
| 5 | 0 | 6 | 4 | 3 | 0 | 2 |
| 6 | 0 | 0 | 0 | 10 | 10 | 0 |

По спискам (1.8) видно, что сток 6 не попал в подмножество A вершин, достижимых из истока по ненасыщенным ребрам. Значит поток X2 максимален. Нанесем его на сеть с указанием направления потоков по отдельным ребрам (см. рис. 1.11).

Таблица 1.8

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| i/j | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 0 | 4 | 2 | 0 | 0 | 0 |
| 2 | 6 | 0 | 8 | 2 | 0 | 0 |
| 3 | 8 | 8 | 0 | 0 | 0 | 0 |
| 4 | 7 | 11 | 0 | 0 | 8 | 0 |
| 5 | 0 | 6 | 4 | 3 | 0 | 2 |
| 6 | 0 | 0 | 0 | 10 | 10 | 0 |

Рис. 1.11

Можно было бы и не строить матрицу R – X2, если своевременно заметить, что потоки по ребрам (4, 6) и (5, 6) равны их пропускным способностям, т.е эти ребра насыщены (см. табл. 1.7 и 1.8).

Используя списки (табл. 1.8) выделим подмножества A и B, на которые оказалось разбитым множество всех вершин: A = {1, 2, 3}, B = {4, 5, 6}. Теперь можно выписать ребра, образующие разрез A / B минимальной пропускной способности: (1, 4), (2, 4), (2, 5), (3, 5).

**Заключение**

Как говорилось выше, теория графов, теория сетей имеют широкое и разнообразное применение. К задаче о максимальном потоке можно, например, свести задачу об оптимальном назначении, хотя такая задача относится к задаче целочисленного программирования и может быть решена соответствующими методами. К задаче о максимальном потоке можно свести транспортную задачу на минимизацию времени перевозок.

**Список использованной литературы**

1. Акулич Н.А. Математическое программирование в примерах и задачах. – М.: Высшая школа, 2003.
2. Иозайтис В.С., Львов Ю.А. Экономико-математическое моделирование производственных систем. – М.: Высшая школа, 2000.
3. Миненко С.Н., Гамазина Г.И. Экономико-математическое моделирование производственных систем. – Учебное пособие, МГИУ