***Тема:*** Выделение связных компонентов. Нахождение максимального потока и минимального разреза

***Задачи:***

1. Закрепить знания основных понятий теории графов.

2. Приобрести практические умения использования специального программного обеспечения для моделирования.

3. Использовать математический аппарат теории графов

***Теоретическая часть:***

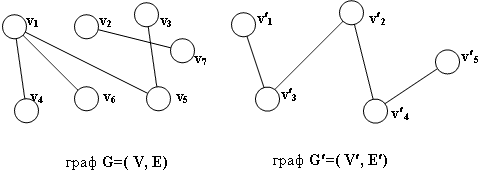
**Компоненты связного и несвязного графа**

В теории графов, понятие связности графа является ключевым при решении многих прикладных задач.

**Определение связного и несвязного графа.**

Граф G(V,E) называется связным, если для любой пары различных вершин этого графа существует цепь, соединяющая эти вершины. Если для графа G(V,E) можно указать пару различных вершин, которые не соединяются цепью, то граф называется несвязным.

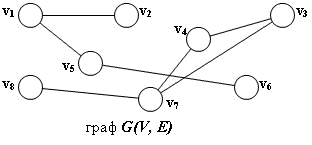
Пример связного и несвязного графов.



**Компонента связности графа** — некоторое множество вершин [графа](http://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) такое, что для любых двух вершин из этого множества существует путь из одной в другую, и не существует пути из вершины этого множества в вершину не из этого множества.

Для ориентированных графов определено понятие сильной компоненты связности

Пример компонент связности графа.



Рассмотрим пример двусвязного графа G(V, E). Всего имеется два множества связных между собой вершин V1 = {v1, v2, v5, v6} и V2 = {v3, v4, v7, v8}.

**Свойства компонент связности.**

1. Каждая вершина графа входит лишь в одну компоненту связности.

2. Любой конечный граф имеет конечное число компонент связности.

3. Граф, состоящий из единственной компоненты связности, является связным.

4. Каждая компонента связности графа является его подграфом.

**Теорема.** Если в конечном графе G ровно две вершины u и v имеют нечетную степень, то они связаны.

**Задание 1. Компоненты сильной связности ориентированного графа.**

*С помощью матрицы смежности найти компоненты сильной связности ориентированного графа D.*

Cоставляем матрицу смежности *A*(*D*) размерности  (*n*− количество вершин) для данного ориентированного графа: она состоит из нулей и единиц, номера строк – индексы вершин , из которых исходят дуги, номера столбцов – индексы вершин, в которые дуги входят (если есть дуга, исходящая из вершины *vi* и входящая в *vj*, то элемент матрицы смежности, стоящий на пересечении *i*-той строки и *j*-того столбца равен 1, иначе – 0.).

Для того, чтобы выделить компоненты сильной связности, необходимо сначала найти матрицу достижимости *T*(*D*)

*Матрица достижимости* ориентированного графа *D* − квадратная матрица *T*(*D*)=[*tij*] порядка *n*, элементы которой равны



ориентированного графа по (1) формуле (T(D)=sign[E+A+A2+A3+… An-1]), затем находим матрицу сильной связности *S*(*D*) ориентированного графа (она должна быть симметрической) по (2) формуле (*S*(*D*)=*T*(*D*)&*TT*(*D*) (*TT*-транспонированная матрица, &- поэлементное умножение)).

***Алгоритм выделения компонент сильной связности***

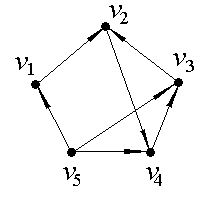
1. Присваиваем *p*=1 (*p* − количество компонент связности), .

2. Включаем в множество вершин *Vp* компоненты сильной связности *Dp* вершины, соответствующие единицам первой строки матрицы *Sp.* В качестве матрицы *A*(*Dp*) возьмем подматрицу матрицы *A*(*D*), состоящую из элементов матрицы *A*, находящихся на пересечении строк и столбцов, соответствующих вершинам из *Vp*.

3. Вычеркиваем из *Sp* строки и столбцы, соответствующие вершинам из *Vp*. Если не остается ни одной строки (и столбца), то *p*- количество компонент сильной связности. В противном случае обозначим оставшуюся после вычеркивания срок и столбцов матрицу как *Sp+1*, присваиваем *p=p*+1 и переходим к п. 2.

***Пример выполнения задания 1***

Выделим компоненты связности ориентированного графа, изображенного на рисунке 1. В данной задаче количество вершин *n=*5.



Рисунке 1.

Значит, для данного ориентированного графа матрица смежности будет иметь размерность 5×5 и будет выглядеть следующим образом

.

Найдем матрицу достижимости для данного ориентированного графа по формуле (1)

, ,

,

Следовательно,



.

Таким образом, матрица сильной связности, полученная по формуле (2), будет следующей:

.

Присваиваем *p*=1  и составляем множество вершин первой компоненты сильной связности *D*1: это те вершины, которым соответствуют единицы в первой строке матрицы *S*(*D*). Таким образом, первая компонента сильной связности состоит из одной вершины .

Вычеркиваем из матрицы *S*1(*D*) строку и столбец, соответствующие вершине *v*1, чтобы получить матрицу *S*2(*D*):

.

Присваиваем *p*=2. Множество вершин второй компоненты связности составят те вершины, которым соответствуют единицы в первой строке матрицы *S*2(*D*), то есть . Составляем матрицу смежности для компоненты сильной связности  исходного графа *D* − в ее качестве возьмем подматрицу матрицы *A*(*D*), состоящую из элементов матрицы *A*, находящихся на пересечении строк и столбцов, соответствующих вершинам из *V*2:

.

Вычеркиваем из матрицы *S*2(*D*) строки и столбцы, соответствующие вершинам из *V*2 ,чтобы получить матрицу *S*3(*D*), которая состоит из одного элемента:



и присваиваем *p*=3. Таким образом, третья компонента сильной связности исходного графа, как и первая, состоит из одной вершины .

Таким образом, выделены 3 компоненты сильной связности ориентированного графа *D*:

|  |  |  |
| --- | --- | --- |
| *D*1: | *D*2: | *D*3: |

**Максимальный поток и минимальный разрез**

Теорема Форда-Фалкерсона 1 (о максимальном потоке и минимальном разрезе).

В любой сети существует максимальный поток. Величина максимального потока равна пропускной способности минимального разреза.

Теорема Форда-Фалкерсона 2.

Поток, вычисленный с помощью алгоритма Форда-Фалкерсона имеет максимальную величину, а разрез , где -множество вершин, помеченных при последнем помечивании, имеет минимальную пропускную способность.

**Нахождение максимального потока и построение минимального разреза в сети с использованием алгоритма Форда-Фалкерсона**

В данной задаче основным параметром на дугах сети является – *пропускная способность*. Пропускная способность показывает, сколько единиц потока может быть передано по дугам сети. Таким образом, *потоком в сети**D* = [*N*, *M*] называется неотрицательная вещественная функция, удовлетворяющая условиям:

1. *ограниченности*: поток по любой дуге сети не превосходит пропускной способности этой дуги ;

2. *сохранения*: суммарный поток, заходящий в любую вершину сети (кроме истока и стока), равен суммарному потоку, выходящему из этой вершины.

Дуга сети называется *насыщенной*, если поток по этой дуге равен пропускной способности этой дуги, т. е. .

*Разрезом сети*называется множество дуг, удаление которых из сети приводит к тому, что исток и сток оказываются несвязанными.

*Пропускной способностью разреза*называется число, равное сумме пропускных способностей дуг этого разреза. Разрез называется *минимальным*, если имеет наименьшую пропускную способность.

Отыскание минимального разреза – одна из основных задач анализа транспортных сетей. В силу конечности графа минимальный разрез может быть найден перебором всех разрезов, но этот путь, конечно, неприемлем для достаточно больших графов.

Минимальный разрез можно отыскать при помощи теоремы Форда – Фалкерсона: в любой транспортной сети величина любого *максимального потока*равна пропускной способности любого *минимального разреза*.

Для нахождения максимального потока в сети разработан *алгоритм Форда – Фалкерсона*. Перед началом выполнения алгоритма все вершины сети нумеруются произвольным образом, кроме источника и стока (источник получает минимальный номер 1, сток – максимальный , где – число узлов).

**Алгоритм состоит из следующих основных шагов:**

1. Определить начальный поток в сети, сложив потоки по дугам, выходящим из источника.

2. Вершинам сети присвоить целочисленные метки, а дугам – знаки «+» и «–» по следующим правилам:

а) вершине-истоку присвоить метку ;

б) находим *непомеченную*вершину , *смежную помеченной*вершине . Если поток по соединяющей вершины дуге меньше пропускной способности этой дуги, то происходит *помечивание*, иначе переходим к рассмотрению следующей вершины. Если вершина является *образом*помеченной вершины , то происходит *прямое помечивание* **(***дуга в прямом направлении допустима***)**: вершина получает метку, равную номеру вершины , соединяющая вершины дуга получает метку «+», переходим к рассмотрению следующей вершины. Если вершина не имеет ни одного помеченного прообраза, поток по дуге в прямом направлении больше 0, то происходит *обратное помечивание* **(***дуга допустима в обратном направлении***)**: вершина получает метку, равную номеру вершины (являющейся в данном случае ее образом), соединяющая вершины дуга получает метку «–», происходит переход к рассмотрению следующей вершины. Процесс помечивания продолжается до тех пор, пока все удовлетворяющие этим условиям вершины не получат метку.

3. Если в результате процедуры помечивания вершина-сток метки не получила, то текущий поток – максимальный, переход к шагу 5. В противном случае перейти к пункту 4.

4. Рассмотреть последовательность вершин *L*, метка каждой из которых равна номеру следующей за ней вершины, и множество дуг *МL*, соединяющих соседние вершины из *L*.

*Построение нового потока по схеме:*

а) Если дуга принадлежит множеству *МL* (смотри выше) и имеет знак «+», то *новый поток по этой дуге* **=** *старый поток по этой дуге* **+** Δ (схему нахождения смотри далее).

б) Если дуга принадлежит множеству *МL* и имеет знак «–», то *новый поток по этой дуге* **=** *старый поток по этой дуге* **–** Δ.

в) Если дуга не принадлежит множеству *МL*, то поток по дуге оставляем без изменения.

Схема нахождения Δ:

I. **,** *где для нахождения* **** рассматриваются все дуги, принадлежащие множеству *МL* и имеющие знак «+», и для каждой такой дуги вычисляется разность между пропускной способностью дуги и потоком по этой дуге (). Затем из этих значений разностей выбирается минимальное значение и присваивается ****.

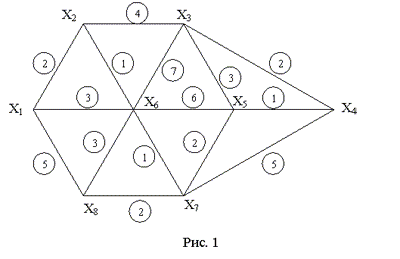
II. *Для нахождения* **** рассматриваются все дуги, принадлежащие множеству *МL* и имеющие знак «–». Затем из этих дуг выбирается дуга с минимальным потоком (), и значение потока по этой дуге присваивается ****.

Перейти к шагу 2.

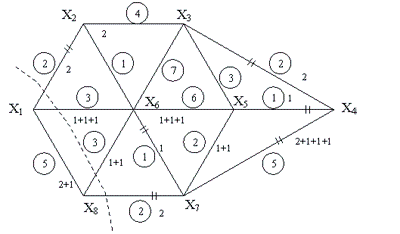
5. Определяем максимальный поток, складывая начальный поток и все полученные изменения потока.

В оптимальном решении, т. е. когда найден максимальный поток, минимальный разрез образуется насыщенными дугами.

**Пример № 1 выполнения задания 2**

На заданной сети найти максимальный поток из X4 в X1 и минимальный разрез.  
   
Решение  
  
Необходимо заполнить таблицу:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| t   X1 | 2, +Х2 | 1, +Х6 | 2, +Х8 | 1, +Х6 | 1, +Х6 | 1,+Х8 |  |
| X2 | 2, +Х3 | 1, +Х3 |  |  | 1, +Х6 |  |  |
| X3 | 2, +Х4 | 1, +Х5 |  |  | 2, +Х5 | 1, +Х5 |  |
| s   X4 | ∞,+Х4 | ∞,+Х4 | ∞,+Х4 | ∞,+Х4 | ∞,+Х4 | ∞,+Х4 | ∞,+Х4 |
| X5 | 1, +Х4 | 1, +Х4 | 2, +Х7 | 2, +Х7 | 2, +Х7 | 1, +Х7 |  |
| X6 | 2, Х3 | 1, +Х5 | 2, +Х7 | 1, +Х7 | 2, +Х5 | 1, Х5 |  |
| X7 | 5, +Х4 | 5, +Х4 | 5, +Х4 | 3, +Х4 | 2, +Х4 | 1, +Х4 |  |
| X8 |  |  | 2, +Х7 |  |  | 1, +Х6 |  |
| V | 2 | 1 | 2 | 1 | 1 | 1 |  |

  
  
После первого шага увеличиваем потоки в дугах, которые сначала были везде = 0.  
Далее продолжаем наращивать поток по цепям:  
     Х4, Х3, Х2, Х1,        V1 = 2  
     Х4, Х5, Х6, Х1,        V2 = 1  
     Х4, Х7, Х8, Х1,        V3 = 2  
     Х4, Х7, Х6, Х1,        V4 = 1  
     Х4, Х7, Х5,              V5 = 1  
     Х4, Х7, Х5, Х6, Х8,    V6 = 1  
     Image, т.е. максимальный поток = 8.  
На последнем, 7-ом шаге, на котором мы не достигли (не можем пометить) вершину t = Х1 находим все помеченные вершины:  
     Xs = {X4}  
     и непомеченные вершины  
     Xt = {X1, X2, X3, X5, X6, X7, X8}.  
Минимальный разрез – ребра, один конец которых лежит в Xs, а другой в Xt.  
В нашем случае это:  
     (Х4, Х3), V(Х4, Х3) = 2;  
     (Х4, Х5), V(Х4, Х5) = 1;  
     (Х4, Х7), V(Х4, Х7) = 5  
      Image          
     Т.е. величина минимального разреза совпадает с максимальным потоком.

**Пример № 2 выполнения задания № 2.**

Постановка задачи поиска максимального потока: найти максимальный поток из в для транспортной сети (рисунок) с помощью алгоритма Форда – Фалкерсона:



Решение:

|  |  |  |
| --- | --- | --- |
| *№* | *Алгоритм* | *Конкретные действия* |
| 1. | *1-я итерация* | 1. .  2.1 (Второй шаг, первая итерация – подобное обозначение идет далее для всех шагов алгоритма). Производим помечивание вершин и дуг, результат показан на рисунке. Вершина 6 получила метку . |
| 2. | *2-я итерация* | 3.1.  .  4.1. ;  ;  .  2.2. Заново осуществляется помечивание. Вершина 6 снова получает метку  (смотри рисунок). |
| 3. | *3-я итерация* | 3.2. .  4.2. ;  ;  .  2.3. Осуществляется помечивание. При этом из вершины 3 прямых допустимых дуг не выходит, однако дуга 2–3 является допустимой в обратном направлении, и вершина 2 получает метку . Вершина 6 получает метку  (смотри рисунок). |
| 4. | *4-я итерация* | 3.3. .  4.3. ;  ;  ;  ;  .  2.4. Осуществляется помечивание. При этом из вершины 3 допустимые дуги не выходят. Вершина 6 не получает метку (смотри рисунок). Переходим к шагу 5. |
| 5. |  | 5. .  Минимальный разрез образуют насыщенные дуги 3–6 и 5–6. Пропускная способность минимального разреза . Условия теоремы Форда – Фалкерсона выполняются   задача решена правильно.  Алгоритм Форда – Фалкерсона используется при решении многих практических задач. Одна из них – *задача об источниках и потребителях*. |

***Задание:***

**1.**С помощью матрицы смежности найти компоненты сильной связности ориентированного графа D. При решении использовать программу Grafoanalizator1.3.3 rus, Простой граф

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Вариант1 | Вариант 2 | Вариант 3 |

**2**. Дана сеть:

****

Определить максимальный поток в сети при начальных значениях дуговых потоков: , , , , , , .

Варианты значений пропускных способностей дуг для задания:



***Вопросы для самоконтроля:***

1. Перечислите основные компоненты связности графов.

2. В чем смысл матрицы достижимости?

3. Запишите теорему Форда-Фалкерсона 1 (о максимальном потоке и минимальном разрезе).