ДК: 519.15

БЕЙШЕКЕЕВ Ж. Ж., БЕЙШЕБАЕВА Ж. К.
КНУ им. Ж. Баласагына, Бишкек.

BEISHEKEEV J.J. BEISHEBAEVA J.K.
J. Balasagyn KNU, Bishkek

ПОТОКИ В СЕТЯХ И АЛГОРИТМЫ РЕШЕНИЯ НА ГРАФАХ

Сеткадагы потоктун графтагы чыгарылышынын алгоритмасы

Flow in networks end algorithm decisions in the graph

Аннотация: *в этой работе для исследования представлена модель сетевого потока в симметричном графовом соединении. Для этой модели разработана программа. Решение удовлетворяет оценке* o(.$|v|^3$0 *алгоритм*
для нахождения максимального потока в сетях.

Аннотация: *симметрялык граф колдонгон сеткадагы поток моделдештирилип бул иште каралган. Моделди иштеткенге программа дайардалган. Сеткадагы максималдык потокту алган бул программа* **o(-1 *v*f -)**
баадагы алгоритмде иштейт.

Annotation: *in this work presentation the model of network flow in testing symmetrical graph connectivity. Prepared for this model program. Decision content oneself with an* **or-l*v*l^3-">** *algorithm for finding maximum flows in networks.*

Ключевые слова: *исток-только исходящие дуги; сток-только заходящие дуги; ребро-дуга; линия; ветвь; дуга - ориентированное ребро.*

Негизги свздвр: *чыгуучу-жалаң чыгуучу дугалар; кирүүчү-жалан кирүүчү дугалар; каптал-дуга; сызыкча; тармактары; дуга- багыты көрсөтүлгөн каптал.*

Keywords: *source - only the proceeding arches; a drain - only the coming arches; edge arch; line; branch; an arch - a directed edge.*

Введение. Каждой ветви и вершине графа может соответствовать некоторое число параметров, характеризующих естественные ограничения и возможности ветвей и вершин. Например, энергетическая система может быть представлена в виде графа, в котором ветви соответствуют линиям электропередачи, а вершины - электростанциям, подстанциям и потребителям. Важные параметры системы включены в модель в виде чисел или весов, приписанных ветвям и вершинам графа. Эти веса могут быть как фиксированными, так и случайными.Так,для энергосистемы типичная вершина, представляющая электростанцию, может иметь следующие веса: максимальную выходную мощность, число генераторов на станции, надежность каждого генератора и стоимость одного киловатт часа. Типичная ветвь может иметь три веса, соответствующих максимально допустимому значению величины передаваемой мощности, надежности линии и ее стоимости. Цель введения весов ветвей и вершин состоит в том, чтобы включить в структурную теоретико-графовую модель неструктурную информацию. Предположим, что дана система фабрик, складов и магазинов, связанных между собой автомобильными и железными дорогами и водными путями [2]. Эта система может быть представлена в виде графа с ветвями, соответствующими транспортным каналам, и вершинами, соответствующими фабрикам, складам и магазинам. В графе фабрики соответствуют входным вершинам, магазины - выходным вершинам, а склады промежуточным вершинам. Можно ввести различия для вершин одного и того же вида, например производство различных товаров для входных вершин. Среди весов вершины можно указать, например, объем производства i-го товара, стоимость производства единицы i-го товара и время, необходимое для производства единицы i-го товара. Для промежуточных вершин может оказаться достаточным указание объема складского помещения. Выходным вершинам могут быть приписаны веса, указывающие типы продаваемых товаров, цена каждого товара, объем местного складского помещения и потребность в товаре каждого вида. Типичными весами ветвей могут быть максимально допустимый объем транспортируемой в единицу времени продукции, стоимость перевозки единицы i-го товара и время перевозки единицы -i-го товара. **Решаемые задачи.** Целесообразность построения модели в виде графа зависит от физической природы решаемой задачи. Наиболее очевидна целесообразность использования графа при решении задачи связности. Для заданной системы и ее графа нас может интересовать, например, вопрос о том, может ли определенный товар быть перевезен из одного места в другое. Нас интересует путь между двумя заданными вершинами, по которому может быть доставлен товар. Может быть поставлена более общая задача, может ли данный товар быть доставлен из любой точки в любую другую. В этом случае необходимо установить, существует ли, по крайней мере, один путь из любой вершины в любую другую. Это задача связности является структурной задачей [6]. Существование пути между двумя вершинами означает

возможность передачи некоторого потока между ними. Однако здесь не указана величина возможного потока. Чтобы включить эту информацию, следует рассматривать взвешенные графы. Предположим, что мы припишем каждой ветви и вершине число, указывающее максимальную величину потока, которую они могут пропустить. Каково же максимальное значение потока, который может быть пропущен между заданной парой вершин? В энергетической системе это число может соответствовать максимуму энергии, поступающей с электростанции к определенному потребителю [4,7]; в системе телеграфной связи оно отображает максимальный объем информации, которая может быть передана при обмене между двумя телеграфными станциями. Задача определения максимального значения потока между *двумя* точками известна как «задача о максимальном потоке». Обобщение этой задачи состоит в том, чтобы найти максимальное значение потоков нескольких товаров, которые одновременно могут быть переданы при обмене между несколькими парами вершин, задача известна как «задача о многотоварном максимальном потоке». Обе эти задачи являются задачами анализа. Задается система и ее модель в виде графа. Мы можем попытаться проанализировать граф и найти максимальное значение потока. Сформулируем также задачу синтеза. Предположим, что заданы множество станций и требования к величине максимального потока. Нужно построить оптимальную систему, удовлетворяющую этим требованиям. Одним из возможных критериев оптимальности может служить минимум стоимости. Задача синтеза имеет большое число вариантов. Например, если предположить, что поведение проектируемой сети может быть точно предсказано, требования к величине потока могут быть полностью удовлетворены. Если же в конструкции или в поведении системы присутствуют элементы случайности, фраза «удовлетворение требований к величине потока должна быть интерпретирована вероятностно>> [3].

Задачи о связности и максимальном потоке тесно связаны с группой задач, которые могут быть объединены под названием задач о живучести или надежности. Речь идет о системе, которая работает в среде, способствующей возникновению условий для повреждений и поломок оборудования. Следствием этого является нарушение связи. Если мы имеем реальную систему, то мы должны проанализировать ее и определить возможные ее повреждения. При заданных критериях

ункционирования мы должны построить систему, в которой возможные нарушения работы были бы минимальны. Как задача анализа, так и задача синтеза могут быть сформулированы и как детерминированная, и как вероятностная задачи. Обычно данную информацию или товар можно доставить разными по качеству путями. Неудачный выбор пути может привести к тому, что будут заблокированы другие потоки, которые в противном случае могли бы быть обслужены. Среди задач, которые будут в дальнейшем рассмотрены, находятся задачи нахождения кратчайшего, наиболее дешевого или самого надежного пути доставки того или иного товара. Во многих физических системах величины поступающих в сеть потоков зависят от времени. Может оказаться, что все возможные пути между двумя некоторыми узлами уже заняты и увеличить поток между ними невозможно. Таким образом, абоненту придется ожидать освобождения канала. Время ожидания может служить важным параметром, характеризующим систему. Типичной задачей анализа являются определение среднего времени ожидания и изучение влияния на его структуры сети и алгоритмов поиска пути доставки информации. Приведенные выше примеры должны были свидетельствовать об универсальности описанной модели и о возможности ее применения к задачам различной физической природы. Для решения этих задач нам понадобятся понятия и результаты теории графов [6].

Транзитивное замыкание между всеми парами вершин.

Задачу определения расстояния между всеми парами вершин можно решить, используя п раз поочередно для каждой вершины от фиксированной вершины.

Таким образом, мы получаем похожий алгоритм со сложностью О(*yi* *)* при использовании метода Форда-Беллмана, или со сложностью *0(yi)* для бесконтурных графов и неотрицательных весов [8]. Однако оказывается, что в общем случае n-кратное использование метода Форда- Беллмана не является наилучшим методом. Покажем теперь один более эффективный метод. Для этого рассмотрим ориентированный граф $G = (V, E)$,

где *V=v(Vl"',"' v n* и предположим, что *E=(£> ,£>)* есть мца весов

бозначив через *£)"* длину кратчайшего пути из V/ до *y.,* содержащего не более m дуг, получаем следующие очевидные уравнения:

$$D^0_y = \begin{cases} O, & если \quad i = j \\ \infty, & если \quad i \neq j \end{cases} \qquad (1)$$

$$£)™ = \min| \; c11^+ вь: \quad \backslash \wedge к < n) \qquad (2).$$

Отметим, что уравнение (2) обнаруживает некоторое сходство с определением произведения двух квадратных матриц. Если операцию min трактовать как «сумму», а операцию «+»- как «произведение», то матрица

£).. является «произведением» матриц ^7 и *Q* . Обозначим такое «произведение» двух матриц через и и отметим, что для этой операции единичным элементом служит матрица *ц* . Тогда из уравнений (1) и (2) теперь легко следует, что

$$Uo=d_y \text{ и } d_{ij}=a_{ij} = dWrVj) \qquad (4)$$

$$\text{или } \breve{и}y \,^{\Phi} a_{ij} \qquad (5)$$

Формула (5) означает, что существует контур отрицательной длины. Произведение (*A*B)* двух матриц размерности nXn можно вычислить за время О ($п^3$). Так как *n* сложений и п-1 сравнений на каждый из $п^2$ элементов произведения *(A*B).* Следовательно, матрицу

Q . и тем самым расстояние между всеми парами вершин можно вычислить за время O *(n^4)*. Пока сложность этого алгоритма такая же, как и для случая n-кратного использования и алгоритма Форда -Беллмана. Однако мы можем ее снизить, если заметим, что операция умножения * ассоциативна (т. е. *(A*B)*C=A*(B*C))*. Этот факт позволяет вычислять произведение матриц, поочередно возводя матрицу *A* в квадрат и тем самым заменяя п- 1 умножение матрицы [log n] умножениями. ***Таким* образом**, мы получаем алгоритм сложности O (n^3 log *n),* отыскивающий расстояния между всеми парами вершин в графе без контуров отрицательной длины. С задачей определения кратчайших путей в графе тесно связана задача транзитивного замыкания бинарного отношения. Вспомним, что под бинарным отношением на множестве *V* мы понимаем произвольное подмножество *E — V X V.* Такое отношение является транзитивным, если удовлетворяется условие, если (x, y) с *E* и *(y, z) ^E,* то *(x, Z) <^E* для произвольных *x,y,z <^E.*

тметим, что бинарное отношение ***J7*** **c**V X V можно однозначно

представить ориентированным графом *G =(V,E).* Для произвольного такого отношения мы определяем *E*= {(x, y): в (V,E)* существует путь ненулевой длины из x в *y}.*

Нетрудно заметить, что E*- транзитивное отношение на множестве **К**и *E* с*E**

Более того, *E** является наименьшим транзитивным отношением, содержащим *E,* т. е. для произвольного транзитивного отношения *F — E* выполняется включение *E*—F.* Отношение *E** называется *транзитивным замыканием* отношения *E.*

Если отношение *E* представить в виде графа (V, **E**), в котором каждая дуга имеет вес 1, то транзитивное замыкание *E** можно вычислить за время O ($п^3$); после завершения работы алгоритма имеем

$$(v,-'V/>^e \ddot{E}^{<}—^{>} Di^jJ] < °°>$$

После завершения модифицированного работы алгоритма имеем

Здесь следует отметить, что известен алгоритм построения транзитивного замыкания, более эффективный, чем предыдущий. Он использует связь этой задачи с умножением матриц, обсуждавшуюся в начале.

Потоки в сетях. Под *сетью* мы будем понимать пару S = <G, c>, где *G=(V, E)*—произвольный ориентированный граф, а ***c*** *(u,v)* функция, которая каждой дуге <u, *v*> ставит в соответствие неотрицательное вещественное число ***c*,** называемое *пропускной способностью* этой дуги. Множества *V* и *E* называются соответственно множеством вершин и множеством дуг сети S.

Для произвольной функции f (u, *v)и* произвольной вершины *v* сети S рассмотрим величину

$$W (f)= \sum_{u:v^u} f(u,v)- \sum_{u.u^v} f(u,v)$$

Если f (w, *v*) мы интерпретируем как поток из **и** в v, то величина W (f) определяет «количество потока», выходящего из вершины **и**. Эта величина может быть положительной (если из вершины v больше выходит, чем входит в нее), отрицательной (если в вершину входит больше, чем выходит из нее, т. е. наступает «накопление» потока в вершине v) и, наконец, равной нулю. Последний случай нас будет интересовать более всего. Выделим в нашей сети две вершины - источник *s* и сток *t (s^t).* Под *потоком* из s в *t* в сети *S* мы будем понимать произвольную функцию вида f (u, *v)*, для которой выполняются условия

O < */(**u**, v)* **<** *c(u, v)* для каждой дуги *(u,v) <^E.* и W(f) = 0 для каждой вершины v ^{e} *V\ {s, t}.* Согласно описанному выше, мы имеем здесь дело с потоком, который не возникает и не накапливается ни в одной из вершин, отличных *от s,t,* и который удовлетворяет следующему условию: через дугу <u,v> мы можем пропустить не более чем *c(u,v)* единиц потока. Такой поток может описывать поведение газа или жидкости в трубопроводе, потоки автомобилей в сети автострад, пересылку товаров по железной дороге (без хранения их на промежуточных станциях), передачу информации в информационной сети. Мы будем интересоваться главным образом нахождением максимального потока в заданной сети. Величина каждого потока из s в t, не превосходит пропускную способность минимального разреза, разделяющего s и t, причем существует поток, достигающий этого значения.

Алгоритмы решения на графах. Все известные алгоритмы построения максимального потока основываются на последовательном увеличении потока, причем модификация потока, увеличивающая его величину, чаще всего опирается на метод увеличивающих цепей.

Напишем условию, что дуга **g.** сети S является допустимой дугой из **y.** и ***y*** относительно потока **£**, если

$$\boldsymbol{eu}=<v,\blacksquare_s\ Vj\blacksquare > \text{и } f(e_{tJ})<_c(e_u) \qquad (6)$$

$$'\bullet J =< \ ', \ J> \text{и } д \ '^{J}) <0 \qquad (7).$$

В зависимости от того, какое из условии имеет место, будет согласованной и не согласованной. Смотрите рисунок один, там взята определенная сеть

Sl^{до}*Sl2* *Sl2<Sin*

от .Для такой сети увеличивающей цепью , т. е. для

данного потока f(^{e}***я***) из произвольная

Vl^{u} ***вы***

знакопеременнаяпоследовательность вершин и дуг . Знание

увеличивающей цепи вида $^{\mathbf{1п}}$позволяет легко увеличить величину потока $/(e,)$ на $^{A(}$-*e)*

$\backslash^{c}$*(e)- f(e*) •*если* ■ *дуга' e* ■ *согласовано*

где ^{A}(^=- \ *o. '. Л$_{e}$) ' е с л и . д у г а . $_{e}$. несогласован*

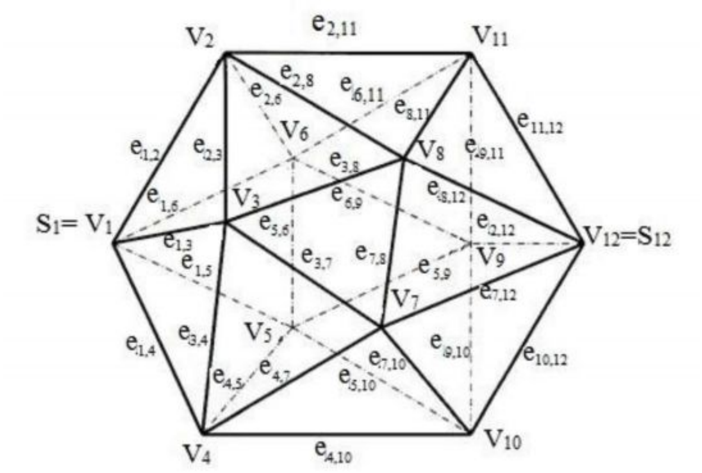

Рисунок 1. Для этого достаточно увеличить на согласованной дуге **£ .**, и **A(/з)** поток по каждой уменьшить на **AQ?**) поток по каждой несогласованной дуге

в,

W(f)= *f(o)* +•• **A(/э**)

W(f)= **/Y/?**) —**A(/э)** Определенная таким образом функция W(f) для дуги **e**, не принадлежащих

цепи ***^ 1"**, является потоком. Для такой определенной сети от^1 ^12 составлена программа.

Мы также использовали из стандартной библиотеки процедуры (Procedure PBC и procedure PPPBC) для построения полного алгоритма максимального потока в сети.

Наша программа правильно строит максимальный поток за время

0(*JJ*), так как для плотных сетей всегда это оценка выполняется. Если

все пропускные способности дуг являются целыми числами, то максимальный поток W(f), построенный программе, является целым числом для каждой дуги e. Отметим, что анализом этого алгоритма мы еще раз показали существование максимального потока в произвольной сети и тем самым предоставили недостающий фрагмент для доказательства фундаментальной задачи о максимальном потоке и минимальном разрезе, однако метод, использованный в этом алгоритме, можно считать обобщением методики увеличивающих цепей. При составлении программы использована следующая литература [1,8]:

```
program Algorithms and Complexity;
uses crt, graph;   const pi=3.14;
var a, b: real;   s,n,x,y,xl,yl,gd,gm,i,h,k,count:integer;
ch:char; z:real; f,fl:array[1..9] of integer;
Function PMPC ( Var s[i],s[l],s[12],v[i],v[l],v[12],v[i] e V, W(f),
c[vl,vl2], vl,vl2eV, s[l],s[12] eV: real);
Begin
for v[l] ev do
for v[12] e y do W[vl,vl2]:=0; (* нулевой начальный поток *)
repeat procedurePBC; if L[S12] Ф oothen (* поток W(f) <maxm *)
begin procedurePPPBC;
(* перенести поток из вспомогательной сети в главную *)
For V/A/GXV do (* XV— множество вершин вспом-ной сети *)
For v[k+n] eXV do
begin W[vl,vl2]:=W[vl,vl2] +XWfvk,v(k+n)J.
if W[vl,vl2] >C[u,v] then
begin
W[v 12,v 1]: = W[v12,v 1]-(W[v 1 ,vl2] -C [u,v]);
W[vl,vl2]:= C [u,v];
End;   End;
End, (* конец фазы *)
until L[S\2] =co; (* поток W[vl,v 12] максимальный*)
Writeln(‘ Максимальный поток W[vl,v 12], vl,vl2eV.’)
End;
begin
k:=i;     s:=s+l;h:=h+l;
if i>100 then h:=h-100*count;
if i>100 then count:=i div 100;
if k>100 then k:=k-100*count;
if k<50 then begin x:=round(320-(200+k)*sin(i));
y:=round(240-(200+k)*cos(i));      {circle(320,240,200+k);}      end;
if s=1 then begin f[ 1]: =x;f 1 [ 1]: =y;end;
if s=2 then begin f[2]:=x;fl[2]:=y;end;
if s=3 then begin f[3]:=x;fl[3]:=y;end;
if s=4 then begin f[4]:=x;fl[4]:=y;end;
if s=5 then begin f[5]:=x;fl[5]:=y;end;
if s=6 then begin f[6]:=x;fl[6]:=y;end;
if s=7 then begin f[7]:=x;fl[7]:=y;end;
if s=8 then begin f[8]:=x;fl[8]:=y;end;
if s=9 then begin f[9]:=x;fl[9]:=y;end;
if s/10=round(s/10) then begin
setcolor(2);
line(f[ 1 ] ,f 1 [ 1 ] ,f[2] ,fl [2]);             line(f[l],fl[l],f[6],fl[6]);
line(f[2],fl[2],f[3],fl[3]);             line(f[6],fl[6],f[5],fl[5]);
```

```
line(f[3],f1[3],f[4],f1[4]); line(f[4],f1[4],f[5],f1[5]);
setcolor(7);
line(f[7],f1[7],f[8],f1[8]);                line(f[7],f1[7],f[9],f1[9]);
line(f[8],f1[8],f[9],f1[9]);
setcolor(5);
line(f[2],f1[2],f[7],f1[7]);                line(f[7],f1[7],f[6],f1[6]);
line(f[2],f1[2],f[8],f1[8]);                line(f[6],f1[6],f[9],f1[9]);
setcolor(6);
line(f[3],f1[3],f[8],f1[8]);                line(f[8],f1[8],f[4],f1[4]);
line(f[4],f1[4],f[9],f1[9]);                line(f[9],f1[9],f[5],f1[5]);
delay(1000);          clearviewport;
s:=0;    end;    end; repeat;
gotoxy(10,12); write(''); until keypressed;   end.
```

Литература

1. Бейшекеев Ж.Ж. Структурное программирование и численные методы на языке Паскаль 6. Издат. КНУ, Бишкек. 2004 г.
2. Бейшекеев Ж.Ж. Динамические программирование с учетом симплекс-метода. Вестник КНУ им. Ж. Баласагынна, 2006г. Естественно-технические науки, Стр. 48.
3. Бейшекеев Ж.Ж. Программирование образов методом Монте-Карло. КГПУ им. Арабаева. Материалы конференции, стр. 106-114. 2003 г.
4. Бейшекеев. Ж. Ж., Жусупбаев. А. Теория графов в энергетических задачах. Труды ИВМ и МГ СО РАН, Выпуск 8. Новосибирск, 2008. Стр. 175.
5. Бейшекеев Ж. Ж. Структурообразование молекул в графах. // Наука и новые технологии. - Бишкек: 2009. №2. –с.225-260.
6. Н. Кристофидес ТЕОРИЯ ГРАФОВ. Алгоритмический подход. Изд. Мир. Москва, 1978.
7. Бейшекеев Ж. Ж. Приложения химических графов в энергетических системах. / / Наука и новые технологии. - Бишкек: 2009. №1. –с.3-8.
8. Форд. Л. Р., Фолкерсон Д. Р. Потоки в сетях - М. Мир. 1966.