Управляемые системы, ИМ, ИК СО АН СССР 1969 г. выпуск 2.
УДК.519.1.

ОБ ОДНОМ АЛГОРИТМЕ НА ГРАФАХ

В.А.Евстигнеев, В.П.Ясакова

Пусть $G=(X,U)$ - неориентированный связный граф с n вершинами, и $x_0 \in X$ - некоторая выделенная вершина. Каждому ребру графа сопоставлен вес, который может быть как числом, так и некоторой функцией. Назовем окрестностью Γ_i вершины x_i множество вершин, смежных с x_i, т.е.

$$\Gamma_i=\{x_i \mid (x_i,x_j)\in U\}.$$

Обозначим через H_n максимальное связывающее дерево графа с n вершинами, а через H_k - связное поддерево H_n, то есть $H_k \subset H_n$.

Пусть $\Gamma^{(k)}=\bigcup\limits_{i\leq k}\Gamma_i \setminus Y_k$,

где Y_k - множество вершин дерева H_k, есть окрестность поддерева H_k.

Рассмотрим уравнение динамического программирования:

$$f_i=\min_{j\neq i}(q_{ij}+f_j),\ i\in J_{n-1}, \qquad /1/$$
$$f_0=0,$$

где $J_{n-1}=\{0,1,2,\dots,n-1\}$.

Числа f_i, приписываемые вершинам сети, вычисляются по следующей рекуррентной схеме:

$$f_i^{(k)}=\min_{j\neq i}(q_{ij}+f_j^{(k-1)}),\ i\in J_{n-1}, \qquad /2/$$
$$f_0^{(k)}=0.$$

Легко видеть, что совокупность ребер (i,j), дающих решение уравнений /1/, образует максимальное связывающее дерево, а случай, когда минимум в /1/ достигается для нескольких ребер $(i,j_1),\dots,(i,j_k)$, не создает трудностей. Используя понятие окрестности $\Gamma^{(k)}$, уравнения /2/ могут быть переписаны в следующем виде:

$$f_i^{(k+1)}=\min_{j\in Y_k}(q_{ij}+f_j^{(k)}),\ i\in\Gamma^{(k)}, \qquad /2'/$$
$$f_0^{(k+1)}=0,$$

что позволяет применить ниже изложенный графический алгоритм решения.

Алгоритм A_1

ШАГ 0. Приписываем вершине x_0 пометку 0.

ШАГ 1. Среди вершин из Γ_0 выбираем вершину i_1 такую, что

$$q_{i_1 0}=\min_{i\in\Gamma_0} q_{i0}$$

и приписываем ей метку

$$f_{i_1}^{(1)} = g_{i_1 0}.$$

ШАГ 2. Образуем поддерево

$$H_1 = (\{x_0, x_{i_1}\}, \{(x_0, x_{i_1})\}).$$

ШАГ 3. Образуем окрестность $\Gamma^{(1)}$ поддерева H_1 :

$$\Gamma^{(1)} = (\Gamma_0 \cup \Gamma_{i_1}) \setminus \{x_0\} \setminus \{x_1\}.$$

ШАГ k .$(k=1,2,\ldots,n)$. Среди вершин из $\Gamma^{(k-1)}$ выбираем вершину i_k такую, что

$$g_{i_k j_0} = \min_{i \in \Gamma^{(k-1)}} \{g_{i_k j} + f_j^{(k-1)}\}, j \in H_{k-1} ,$$

и приписываем ей метку $g_{i_k j}$.

ШАГ $k+1$. Образуем поддерево

$$H_k = (\{Y_{k-1} \cup x_{i_k}\}, \{U_{k-1} \cup (i_k, j_0)\}).$$

ШАГ $k+2$. Образуем окрестность $\Gamma^{(k)}$ поддерева H_k :

$$\Gamma^{(k)} = (\Gamma^{(k-1)} \cup \Gamma_{i_k}) \setminus \{x_{i_k}\}.$$

Т е о р е м а. Алгоритм A_1 дает решение уравнения /2'/ и, следовательно, уравнения /1/.

Д о к а з а т е л ь с т в о. Допустим противное, то есть предположим, что существует решение H_n' уравнения /1/, отличное от решения, полученного с помощью алгоритма A_1. Пусть (i_0, j_0) - ребро, принадлежащее H_n' и не принадлежащее H_n. Тогда (i_0, j_0) образует с ребрами из H_n цикл с входом $\bar{x}$, у которого сумма весов $g(u)$ от $\bar{x}$ до j_0 по пути, состоящем из ребер дерева H_n больше, нежели сумма весов по пути, содержащем (i_0, j_0). Но это невозможно по построению. Теорема доказана.

С л е д с т в и е 1. Если $g(u)$ - длина ребра u, а f_j есть кратчайшее расстояние от 0 до j , то алгоритм A_1 даст дерево кратчайших расстояний от X_0 до остальных вершин сети [2].

С л е д с т в и е 2. Если $g(u)$ - длина ребра u , а $l_j^{(k)} = \sum_{u \in H_k} g(u)$ - сумма длин ребер H_k , то алгоритм A_1 строит кратчайшую связывающую сеть /алгоритм Прима [3] /.

С л е д с т в и е 3. Если g_{ij} есть функция от f_j , то есть $g = g(i, j, f_j)$, то алгоритм A_1 позволяет построить дерево кратчайших путей в сети с длинами ребер, зависящими от длины кратчайшего пути, ведущего из X_0 в начало данного ребра [4].

П р и м е ч а н и е. Алгоритм A_1 легко обобщается на случай ориентированных графов, если под окрестностью вершины X понимать множество вершин, являющихся концами дуг с началом в X .

При реализации алгоритма A_1 на ЭВМ граф задается обобщенной матрицей смежности $L = \|l_{ij}\|$,

где $$l_{ij} = \begin{cases} g_{ij}, & \text{если } i \neq j \text{ и } j \in \Gamma_i \\ \infty & \text{в противном случае.} \end{cases}$$

Ниже приводится программа, реализующая алгоритм A_1 .

Программа

```
начало целый n ; ввод (n);
начало целый i, j, k, n1, μ, ν, α, β;
целый массив p, l, t[0:n], d[1:n, 1:2];
      массив a[0:n, 0:n], min[0:n];
процедура S(j) ; начало a[l[k-1],j]:=a[l[k-1],j]+min[k-1] конец;
ввод (a); k:=0; t[0]:=0; p[0]:=0; l[0]:=0; min[0]:= ⏨18;
для j:=1,...,n   цикл если min[k] > a[i,j]   то
   {min[k]:=a[0,j]; l[k]:=j};
 L : если l[k]<n   то k:=k+1 иначе {n1:=k; на L1};
если t[k-1]≤l[k-1] то t[k]:=l[k-1] иначе {для ν:=2,...,
k цикл если t[k-ν]≤l[k-1] то {для μ:=0,...,ν-2
цикл t[k-μ]:=t[k-μ-1]; t[k-ν+1]:=l[k-1]; на M}};
M : для j:=0,...,t[1]-1 цикл S(j); для α:=2,...,
k цикл для j:=t[α-1]+1,...,t[α]-1 цикл S(j) ; для j:=
t[k]+1,...,n цикл S(j); t[k+1]:=n+1;
min[k]:=⏨18; для β:=0,...,k цикл {i:=t[β]; для α:=
1,...,k цикл {для j:=t[α-1]+1,...,t[α]-1 цикл если min[k]
>a[i,j] то {min[k]:=a[i,j]; l[k]:=j; p[k]:=i}; для j:=
t[α]+1,...,t[α+1]-1 цикл если min[k] ≥ a[i,j]  то
{min[k]:=a[i,j]; l[k]:=j; p[k]:=i}}}; на L ;
 L1: α:=1; L2 : d[α,1]:=p[k]; d[α,2]:=l[k];
если p[k]=0 то на L3 ; для ν:=1,...,n1   цикл
если p[k]=l[k-ν]   то {α:=α+1; k:=k-ν;   на L2};
 L3: вывод (n1, min[n1], d);
конец конец *
```

Поступила в редакцию 7.10.1968 г.

Литература

1. К.Берж. Теория графов и ее применения, ИЛ, М., 1962.

2. Р.Беллман. To the transportation problem,-Quarterly of Applied Mathematics, v. 16, N I, 1958.

3. Р.К.Прим. Кратчайшие связывающие сети и некоторые обобщения. -Кибернетический сборник, ИЛ, М., 1961. вып. 2.

4. K.L.Cooke and Eric Halsey The shortest route through a network with time - dependet internodal transit times. Journal of Mathematical Analysis and Applications, v. 14, 1966, 493-498.