

## БЫСТРОДЕЙСТВУЮЩИЕ СЕРВЕРНЫЕ ПРОЦЕДУРЫ ОПТИМИЗАЦИИ МАРШРУТОВ НА СЕТЯХ

Хаджинова Н.В., Ревотюк М.П.

Белорусский государственный университет информатики и радиоэлектроники  
220013, Минск, ул. П. Бровки, 6, БГУИР, кафедра ИТАС,  
тел. +375(17) 239-88-23, E-mail: rmp@bsuir.unibel.by

Предмет обсуждения – структуры данных объектного представления транспортных сетей и алгоритмы быстродействующих процедур оптимизации маршрутов с поглощением предопределенных решений [1].

Известно, что наиболее предпочтительным представлением слабо связного нагруженного графа  $G(N,A)$ , состоящего из  $N$  вершин и  $A$  дуг, для волновых алгоритмов построения дерева, реализующих схему Дейкстры, является структура смежности *FSF* (*Forward Star Form*):

$$FSF = \langle n, H, B, W \rangle, \quad (1)$$

где  $n$  – количество вершин графа транспортной сети,  $n=|N|$ ;  $H$  – массив указателей списков смежности;  $B$  – массив элементов списков смежности (конечных вершин дуг графа);  $W$  – массив весов дуг.

Для каждой вершины  $x \in N$  множество непосредственно достижимых смежных вершин обозначим  $x'$ ,  $x' = \{k \mid w(x,k) \geq 0\}$ . Если вершины графа пронумерованы числами  $\overline{0, n-1}$ , то элементы (1) фиксируются так:

$$FSF = \begin{cases} H_0 = 0, H_{i+1} = H_i + |i'|, i = \overline{1, n-1}; \\ B(H_i + j) = i'(j), \\ W(H_i + j) = w(i, i'(j)), j = \overline{0, |i'|-1}, i = \overline{0, n-1}. \end{cases} \quad (2)$$

Очевидно, что  $|H|=n+1$ ,  $|B|=|W|=H_n$  (элемент  $H_n$  содержит  $|A|$ ).

Для выделения альтернатив развития дерева их вершины  $x$  в этом случае используется следующий простой алгоритм:

$$x' = \{B(H_x + i) \mid H_x + i < H_{x+1}, i \geq 0\}.$$

Предлагается для каждой вершины графа модели транспортной предварительно выделить и пометить входные дуги минимальной длины:

$$T_j = \left\{ (i, j) : i = \arg \min_{i,j} w(i, j) : (i, j) \in A \right\} j \in N. \quad (3)$$

В случае ветвления процесса дугу из (3) значение расстояния до ее конечной вершины не изменится. Такую вершину можно не включать в очередь, что существенно сокращает время вычислений, особенно при обработке потока запросов на поиск маршрутов.

Для отображения множества (3) достаточно выбрать элементы  $B$  или  $W$  в (1) со свободным двоичным разрядом и пометить дугу бинарным признаком. Таким образом, учет предопределенных решений не требует изменения объема памяти для хранения описания реальной сети.

Схема реализации алгоритма Дейкстры, представленная в стиле кодирования [2], учитывающая predetermined решения (3):

```

Procedure SPT( $r$ ); // Построение дерева с корнем в вершине  $r$ 
begin
  QINIT( $Q, r$ ); //  $Q$  - очередь вершин, вначале включает  $r$ 
  foreach  $i \in N$  do  $D_i = \infty, P_i = i$ ;
   $D_r = 0$ ; // Фиксация корня дерева
  repeat
    QOUT( $Q, i$ ); // Выборка вершины  $i$  из очереди  $Q$ 
    SPG( $i$ ); // Развитие дерева из вершины  $i$ 
  until  $Q = \emptyset$ .
end;
Procedure SPE( $i$ ); // Развитие дерева из вершины  $i$ 
begin
  foreach  $j \in i'$  do
    if  $D_j > D_i + w(i, j)$  then begin
       $D_j = D_i + w(i, j), P_j = i$ ;
      QIN( $Q, j$ ); // Включение вершины  $i$  в очередь  $Q$ 
    end;
  end;
Procedure SPG( $i$ ); // Предлагаемое развитие дерева из вершины  $i$ 
begin
  foreach  $j \in t_i$  do
    if  $D_j > D_i + w(i, j)$  then begin
       $D_j = D_i + w(i, j), P_j = i$ ;
      SPE( $j$ ); // Развитие дерева из вершины  $j$ 
    end;
  foreach  $j \in i' \setminus t_i$  do
    if  $D_j > D_i + w(i, j)$  then begin
       $D_j = D_i + w(i, j), P_j = i$ ;
      QIN( $Q, j$ ); // Включение вершины  $i$  в очередь  $Q$ 
    end;
  end;

```

Эксперименты показывают, что на графе реальной сети автомобильных дорог, где  $|N| \cong 10^4$ ,  $|A| \cong 10^5$ , среднее время поиска кратчайших расстояний между случайными парами вершин сокращается до 40%.

#### ЛИТЕРАТУРА

1. Ревотюк М.П., Застенчик Н.И., Шешко Е.В. Поглощение predetermined решений жадными алгоритмами//Известия Белорусской инженерной академии, № 1(17)/2, 2004. – С. 112-114.
2. Deo N., Chi-yin Pang. Shortest-Path Algorithm: Taxonomy and Annotation// Networks, 1984. – Vol. 14. – P. 275-323.