第14卷 第3期 郑州工学院学报 Vol.14 No.3
1993年 9月 Journal of Zhengzhou Institute of Technology Sep. 1993

对R.W.Floyd算法的一点扩展*

王贺明

(郑州工学院水利系)

摘要: 本文对*Floyd*算法进行了分析，在此基础上提出了一种扩展的*Floyd*算法。该算法在实际工程中有一定的应用价值。扩展的*Floyd*算法已在计算机上实现，计算结果证明该算法正确。

关键词: 算法，最短路径，路选表

中国图书分类号: TP39

1 R.W.Floyd算法分析

定义[1] 设图$G=(V,E)$，对G的某一个边(V_i,V_j)相应地有一个数$L(V_i,V_j)$（可简化为L_{ij}）称为(V_i,V_j)的权，G连同在它边上的权称为赋权图。

在赋权图中给定一个源点V_i及一个终点V_j，最短路径问题就是在(V_i,V_j)通路集合$\{V_i,V_j\}$中找到一条长度最短的路径，这样一条从V_i到V_j的路径称最短路径。

*R.W.Floyd*算法是用来求解图G中任意两点间的最短路径，其算法的基本思想是:

用一个带权的邻接矩阵D表示一个赋权图，用$D[i,j]$表示边(V_i,V_j)上的权值，若(V_i,V_j)不存在，则置$D[i,j]$为∞（在本文的算法实现中，用100代表∞）。假若求从结点V_i到结点V_j的最短路径，如果V_i到V_j有边，则从V_i到V_j存在一条长度为$D[i,j]$的路径，但该路径不一定是最短路径，尚需进行$n-1$次试探，试探过程中首先要判断(V_i,V_1,V_j)是否存在，也就是要判断从V_i到V_1的边以及从V_1到V_j的边是否存在，如果存在，则取(V_i,V_j)与(V_i,V_1,V_j)中较短者作为V_i到V_j中间结点序号不大于1的最短路径。然后在路径上再增加一个结点V_2，如果$(V_i,\cdots,V_2)$和$(V_2,\cdots,V_j)$分别是已经找到的中间结点序号不大于1的最短路径，那么$(V_i,\cdots,V_2,\cdots,V_j)$就有可能是从$V_i$到$V_j$的中间结点的序号不大于2的最短路径，将它与已求得的从V_i到V_j中间结点序号大于1的最短路径相比较，取其短者作为新的最短路径。一般情况下，若$(V_i,\cdots,V_k)$和$(V_k,\cdots,V_j)$分别是从V_i到V_k和从V_k到V_j中间结点序号不大于$k-1$的最短路径，则将中间结点不大于K

* 收稿日期: 1992-11-11

的$(V_i,\cdots V_k,\ V_j)$和已得到的V_i到V_j中间结点序号不大于$K-1$的最短路径相比较，取其较短路径作为V_i到V_j中间序号不大于K的最短路径。这样，经过$n-1$次比较，最后求得即是V_i到V_j的最短路径。

以上求最短路径的具体过程是：根据不断增加路选结点，依次构造一个方阵序列：$D^{(0)}$，$D^{(1)}$，…，先从$D^{(0)}=[L_{ij}]$出发，第K个矩阵$D^{(K)}=〔d_{ij}^{(K)}〕$的元素$d_{ij}^{(K)}$表示从V_i到V_j；而中间结点仅属于V_i到V_K的K个点的所有通路中的最短路径长。

已知：$D^{(K-1)}=〔d_{ij}^{(K-1)}〕$

则：$d_{ij}^{(k)}=min\{d_{ij}^{(K-1)},\ d_{iK}^{(K-1)}+d_{Kj}^{(K-1)}\}$

运算过程从$K=1$开始，让i,j遍取$1\sim N$的所有值，然后K逐步增1，重复进行直到$K=N$为止。这时$D^{(N)}=〔d_{ij}^{(N)}〕$的元素$d_{ij}^{(N)}$就是任意两点间(V_i,V_j)的最短矩离。

例如，图1是一赋权图

它的方阵序列依次是：

$$D^{(0)}=D^{(1)}=D^{(2)}=\begin{bmatrix}0&2&5&1&\infty&\infty\\2&0&3&2&\infty&\infty\\5&3&0&3&1&5\\1&2&3&0&1&\infty\\\infty&\infty&1&1&0&2\\\infty&\infty&5&\infty&2&0\end{bmatrix}$$

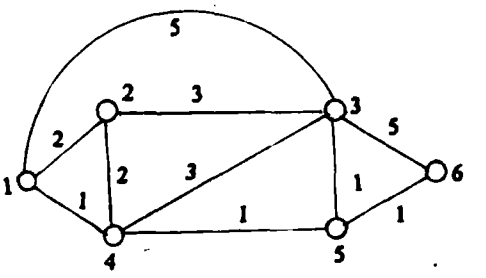

图1　R.W.Floyd算法赋权图

$$D^{(3)}=\begin{bmatrix}0&2&5&1&6&10\\2&0&3&2&4&8\\5&3&0&3&1&5\\1&2&3&0&1&8\\6&0&1&1&0&2\\10&8&5&8&2&0\end{bmatrix}\qquad D^{(4)}=\begin{bmatrix}0&2&4&1&2&9\\2&0&3&2&3&8\\4&3&0&3&1&5\\1&2&3&0&1&8\\2&3&1&1&0&2\\9&8&5&8&2&0\end{bmatrix}$$

$$D^{(5)}=D^{(6)}=\begin{bmatrix}0&2&3&1&2&4\\2&0&3&2&3&5\\3&3&0&2&1&3\\1&2&2&0&1&3\\2&3&1&1&0&2\\4&5&3&3&2&0\end{bmatrix}$$

$D^{(1)}$是根据$D^{(0)}$计算得到，而$D^{(2)}$又是根据$D^{(1)}$计算而得到，依次类推。在这里$D^{(0)}$，$D^{(1)}$，$D^{(2)}$三者相等，而$D^{(5)}$，$D^{(6)}$又相等。

2 对R.W.Floyd算法的扩展

由 $F.W.Floyd$ 算法对一个赋权图进行计算，最后得到一个 N 阶方阵 $D^{(N)}=〔d_{ij}^{(N)}〕$，该方阵只表征了任意两结点 (V_i,V_j) 之间的最短矩离，但没有给出最短路径上的路选结点．该算法用于解决实际问题时并不十分完善．若把我国的铁路线看成是一个取其距离的赋权图，用 $R.W.Floyd$ 算法可以求出从成都到广州的最短铁路长度，但是，是从郑州去广州，还是取道贵阳去广州，尚不可得知．为了解决最短路径同时具有最短路径的路径结点的问题，在此，对 $R.W.Floyd$ 算法加以扩展，使以上提出的问题得以解决．

其算法的具体过程是:

从方阵 $D^{(0)}$ 出发，根据 $R.W.Floyd$ 算法仍然构造一个方阵序列 $D^{(1)}$，$D^{(2)}$，…，$D^{(N)}$，其中 N 为赋权图中结点的个数．为了得到最短路径的路径结点，需要一个元素个数为 n^3 的三维矩阵 G。

已知: $D^{(K-1)}=d_{ij}^{(K-1)}$,$D^{(K)}=d_{ij}^{(K)}$,取 $d_{ij}^{(K-1)}$ 的路径长为 L_{ij},$d_{iK}^{(K-1)}+d_{Kj}^{(K-1)}$ 的路径长为 L_{ikj}，G 矩阵中 i 到 k 结点路径中的结点数为 h_{ik}，当 $L_{ikj}<L_{ij}$ 时

$$d_{ij}^{(K)}=min\{d_{ij}^{(K-1)},\ d_{ik}^{(K-1)}+d_{kj}^{(K-1)}\}$$

$$g_{ijh}=g_{ikh} \qquad g_{ikh}\neq 0$$

$$g_{kj,(h-h_{ik})} \qquad g_{ikh}=0 \qquad (i,j,k,h,\in 1,2,\cdots,n)$$

上式表明，当 $g_{ikh}\neq 0$ 时，用 V_i 到 V_k 及 V_K 到 V_j 路径上的路径结点去代替原 V_i 到 V_j 路径上的路径结点，当 $g_{ikh}=0$ 时，V_k 到 V_j 路径上的结点存于 g_{ijh}。

其中：〔g_{1jx}〕表征的是以结点 V_1 为源点，$V_j(j\neq 1,2,3,\cdots,n)$ 为终点的路选表.

〔g_{2jx}〕表征的是以结点 V_2 为源点，V_j 为终点的路选表.

〔g_{njx}〕表征的是以结点 V_n 为源点，V_j 为终点的路选表.

三维矩阵 G 要根据邻接矩阵 $D^{(0)}$ 提前赋初值。

3 扩展的 R.W.Floyd 算法实现

用高级语言 BASICA 在计算机上实现扩展的 R.W.Floyd 算法。

取图 1 的赋权图，根据赋权图，建立一邻接矩阵 D，根据扩展的 R.W.Floyd 算法，编制程序，程序具有以下功能:

取图 1 的赋权图，根据赋权图，建立一邻接矩阵 D，根据扩展的 R.W.Floyd 算法，编制程序，程序具有以下功能:

①可求出赋权图中任意两结点最短路经及路选结点。

②增加新路径后仍能计算出变化了的任意两结点的最短路径及路选结点。

③断连某一路径后也能计算出变化了的任意两结点间最短路径和路选结点。

以图1的赋权图为例，图2是1的邻接矩阵。

0	2	5	1	100	100
2	0	3	2	100	100
5	3	0	3	1	5
1	2	3	0	1	100
100	100	1	1	0	2
100	100	3	100	2	0

图2　具有6个结点的链路邻接矩阵表

源点	终点	路径	下一结点	最短距离
1	2	2	2	2
	3	4 5 3	4	3
	4	4	4	1
	5	4 5	4	2
	6	4 5 6	4	4

图3　源点为v_1的路选表

图3是当源点取v_1时，维点取$v_j(j\in 2,3,\cdots,n)$时的最短路径和路选表。

图4是当源点取v_2时，终点取$v_j(j\in 1,3;4,\cdots n)$时的最短路径及路选表

源点	终点	路径	下一结点	最短距离
2	1	1	1	2
	3	3	3	3
	4	4	4	2
	5	4 5	4	2
	6	4 5 6	4	5

图4　源点为v_2的路选表

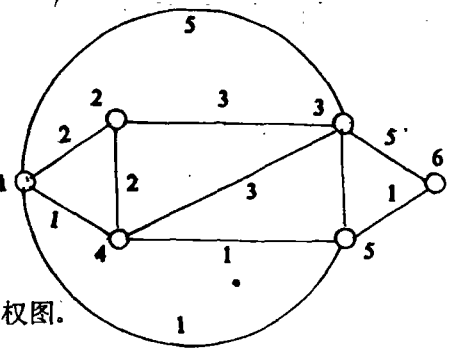

图5　增加$v_1,v_5L_{15}=1$时的赋权图

图5是增加一条从v_1到v_5，$L_{15}=1$的赋权图。

图6是图5的邻接矩阵。

0	2	5	1	1	100
2	0	3	2	100	100
5	3	0	3	1	5
1	2	3	0	1	100
1	100	1	1	0	2
100	100	5	100	2	0

图6　增加v_1到v_5,$L_{15}=1$的邻接矩阵表

源点	终点	路径	下一结点	最短距离
1	2	2	2	2
	3	5 3	5	2
	4	4	4	1
	5	5	5	1
	6	5 6	5	3

图7　增加v_1到v_5，源点为v_1的路选表

图7是增加v_1到v_5新路径时源点取v_1，终点取$v_j(j\in 2,3,\cdots n)$的最短路径和路选结点表。

图8是断连(图1)v_1到v_3路径时的赋权图

图9是图8的邻接矩阵表。

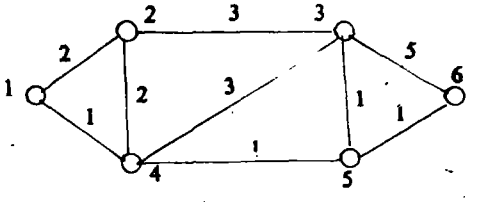

图8　断连v_1到v_3路径的赋权图

0	2	100	1	100	100
2	0	3	2	100	100
100	3	0	3	1	5
1	2	3	0	1	100
100	100	1	1	0	2
100	100	5	100	2	0

图9　断连v_1到v_3后的邻接矩阵表

图10是断连 v_1 到 v_3 之后以 v_1 为源点，以 v_j ($j\in 2,3,\cdots n$)为终点的最短路径及路选表。

源点	终点		路径	下一结点	最短距离
1	2		2	2	2
	3	4	5	4	3
	4	4		4	1
	5	4	5	4	2
	6	4	5	4	4

图10 断连 v_1 到 v_3 后以 v_1 为源点的路选表

从扩展算法的分析和实现中可知，扩展的算法具有很大的灵活性，并能准确给出最短路径的各路径点。由于扩展算法增加了一个三维矩阵，同原算法相比较，运算过程中，计算机内存的开销增加了 n^3 个结点单元。

4 结束语

扩展的算法可用于铁路、公路、航空、城市建设等部门。它不仅可算出最短矩离及其最短路径上的各点，还可以算出最小费用、最短时间及其路选各点。

参考文献

〔1〕 王朝瑞. 图论. 北京工业学院出版社，1987.

〔2〕 潘启敬等编. 计算机网络. 中国铁道出版社，1984.

〔3〕 严蔚民等编. 数据结构. 清华大学出版社，1987.

An Extension of R.W.Floyd's Algorithm

Wang he ming

(Zhengzhou Institute of Technologe)

Abstract: In this paper, Floyd's algorithm is studied, based on this, extented floyd's algorithm is constructed.

Extented algorithm is very useful in practical engineering. Extented Floyd's algorithm have finished on the computer, the conputation results show that extented algorithm is right.

Keywords: Algorithm; shortest-path; path node list