# 最短路径问题的若干算法的编程

王 强
(内蒙古师范大学计算机与信息工程学院 呼和浩特 010022)

**摘 要** 本文讨论了带权图的最短路径问题的三类不同情况,针对每种情况介绍了较好的算法,并在 MATLAB 软件环境下编制了这些算法的程序。通过一个实例,验证了算法的正确性。

**关键词** 最短路径,MATLAB,单源最短路径,单汇最短路径

## 1 引言

生产实践、运输管理以及工程建设的许多方面,诸如各种工艺路线的安排,厂区及货场的布局,管道线网的铺设等问题,都与寻找一个图的最短路径问题密切相关。

设带权图 $G$ 中每条边带的权均大于等于 0。$u$,$v$ 为 $G$ 中任意两个顶点,从 $u$ 到 $v$ 的所有通路中带权最小的通路称为从 $u$ 到 $v$ 的最短路径。

一般来说,最短路径问题可分为以下三类:

(1)单源最短路径问题:寻求从一固定起点到其余各点的最短路径。

(2)单汇最短路径问题:寻求从各点到一固定终点的最短路径。

(3)每对顶点间的最短路径问题:寻求带权图中任两顶点之间的最短路径。

本文基于 MATLAB 给出了这几种问题的算法的编程。

## 2 单源最短路径问题

到目前为止,公认的求单源最短路径的较好的算法是由 E.W.Dijkstra 于 1959 年给出的。该算法能求出任一给定顶点 $v_1$ 到其余顶点的最短路径。用 $W_{ij}$ 表示边 $(v_i,v_j)$ 上的权,若顶点 $v_i$ 与 $v_j$ 不相邻,令 $W_{ij}=\infty$。$P$ 表示具有永久标号的顶点集;对每个顶点,定义两个标记 $l(v)$(表示从顶点 $v_1$ 到 $v$ 的一条路径的权)和 $f(v)$($v$ 的父顶点)。算法如下:

(1)赋初值:$v_1$ 获得 $p$ 标号,$P=\{v_1\}$,$l(v_1)=0$;$\forall v_i\in V-P$,令 $l(v_i)=w_{1i}$,$f(v_1)=v_1$,$v_i$ 获得 $t$ 标号;

(2)求下一个 $p$ 标号顶点:设 $l(v_i)=\min\limits_{v\in V-P}\{l(v)\}$,则 $v_i$ 获得永久性标号,$P=P\cup\{v_i\}$,若 $P=V$,则算法结束,否则转(3)。

(3)修改 $V-P$ 中各顶点的 $t$ 标号:若 $l(v_j)>l(v_i)+w_{ij}$,则 $l(v_j)=l(v_i)+w_{ij}$,$f(v_j)=v\cdot$,其中 $v_i$ 是刚刚获得 $p$ 标号的顶点,转(2)。

求图 1 中顶点 $v_i$ 到 $v_7$ 的最短路径的 Dijkstra 算法如下。

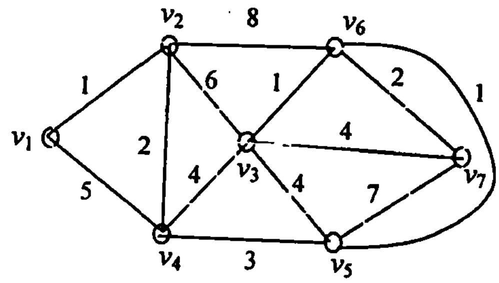


图 1

表 1

| $k$ | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ | $v_6$ | $v_7$ |
|---|---|---|---|---|---|---|---|
| 0 | $\boxed{0}$ | 1 | ∞ | 5 | ∞ | ∞ | ∞ |
| 1 | | $\boxed{1}/v_1$ | 7 | 3 | ∞ | ∞ | ∞ |
| 2 | | | 7 | $\boxed{3}/v_2$ | 6 | ∞ | ∞ |
| 3 | | | 7 | | $\boxed{6}/v_4$ | 7 | 13 |
| 4 | | | $\boxed{7}/v_2$ | | | 7 | 11 |
| 5 | | | | | | $\boxed{7}/v_5$ | 9 |
| 6 | | | | | | | $\boxed{9}/v_6$ |

所以,从表 1 中可知顶点的父顶点是 $v_6$,向前追溯,即得从 $v_1$ 到 $v_7$ 的最短路径是:$v_1$ $v_2$ $v_4$ $v_5$ $v_6$ $v_7$。这条路径的权为 9。

用 MATLAB 编写的 Dijkstra.m 如下:

```
function [l,father] = dijkstra
  w = [0 1 inf 5 inf inf inf;
     1 0 6 2 inf 8 inf;
     inf 6 0 4 4 1 4;
     5 2 4 0 3 inf inf;
     inf inf 4 3 0 1 7;
     inf 8 1 inf inf 0 2;
     inf inf 4 inf 7 2 0];
  n = size(w,1);
  l = w(1,:); father = ones(1,n); ptsign = zeros(1,n);
   ptsign(1) = 1;
  presentp = 1;
  for k = 1:n-1
    min = inf;
    for i = 1:n
      if ~ptsign(i)
        if l(i) < min
          min = l(i); presentp = i;
        end
      end
    end
    ptsign(presentp) = 1;
    for i = 1:n
      if ~ptsign(i)
        if l(i) > l(presentp) + w(presentp,i)
          l(i) = l(presentp) + w(presentp,i); father(i)
              = presentp;
        end
      end
    end
  end
```

在 MATLAB 命令窗中键入:[l, father] = dijkstra

运行结果是:

```
l =
   0   1   7   3   6   7   9
father =
   1   1   2   2   4   5   6
```

## 3 单汇最短路径问题

对于无向带权图 G 而言,单汇问题就是单源问题,二者不存在任何区别。

对于有向带权图 G 而言,可通过改变所有有向边的方向,利用上面的单源最短路径的算法求解。使用上面的 Dijkstra.m 时,只需将矩阵 w 转置即可实现。

## 4 每对顶点间的最短路径问题

显然此问题可由重复 Dijkstra 算法来解决,每次取定一个顶点作起点,但这需要大量重复计算,效率不高。Floyd 提出了比这更好的算法,可一次性地求出任意两点间的最短路径和距离,其基本思路是:从图 G 的带权邻接矩阵 $A^{(0)}=(a_{ij}^{(0)})_{n\times n}$开始,使用递推公式:$A^{(k)}=(a_{ij}^{(k)})_{n\times n}$,其中 $a_{ij}^{(k)}=\min\{a_{ij}^{(k-1)}, a_{ik}^{(k-1)}+a_{kj}^{(k-1)}\}$, $k=1,2,\cdots,n$;依次计算 $n$ 次,最后算出的 $A^{(n)}$称为图 $G$ 的距离矩阵,$A^{(n)}$的 $i$ 行 $j$ 列元素便是顶点 $v_i$ 到顶点 $v_j$ 的最短路径长度。同时再引入一个后继点矩阵 $path$ 来记录两点间的最短路径,$path(i,j)$最终的取值为 $v_i$ 到 $v_j$ 的最短路径上 $v_i$ 的后继点的下标。Floyd 算法如下:

(1)赋初值:当时 $a(i,j)=\infty$ 时,$path(i,j)=0$,否则 $path(i,j)=j$;$k=1$

(2)更新 $a(i,j)$, $path(i,j)$:对所有的 $i,j$,若 $a(i,k)+a(k,j)<a(i,j)$,则

$a(i,j)=a(i,k)+a(k,j)$, $path(i,j)=path(i,k)$。

(3)若 $k=n$,则停止,否则 $k=k+1$,转(2)。

借助 MATLAB 软件,仍以上图为例,求该图中每对顶点间的最短路径的 Floyd 算法的函数 Floyd.m 如下:

```
function [a,path] = floyd
  a = [0 1 inf 5 inf inf inf;
    1 0 6 2 inf 8 inf;
    inf 6 0 4 4 1 4;
    5 2 4 0 3 inf inf;
    inf inf 4 3 0 1 7;
    inf 8 1 inf inf 0 2;
    inf inf 4 inf 7 2 0];
  n = size(a,1);
  path = zeros(n,n);
  for i = 1:n
    for j = 1:n
      if a(i,j) ~= inf
        path(i,j) = j;
      end
    end
  end
  for k = 1:n
    for i = 1:n
      for j = 1:n
        if a(i,k) + a(k,j) < a(i,j)
          a(i,j) = a(i,k) + a(k,j);
          path(i,j) = path(i,k);
        end
      end
    end
  end
```

(下转第 100 页)

段，产生新形体的几何信息和拓扑信息，输出显示该形体。作者用 C 语言从底层开发了三维立体造型拼合运算程序和软件，图 1 是该系统实现的两四面体并运算的输出显示，图 2 是两四面体差运算的输出显示，图 3 为该软件实现的两多面体并运算后的三视图和轴测图，图 4 为该软件实现的两多面体差运算后的三视图和轴测图。

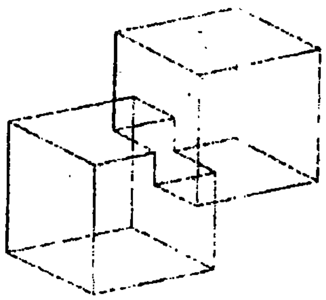

图 1　两四面体并运算的显示

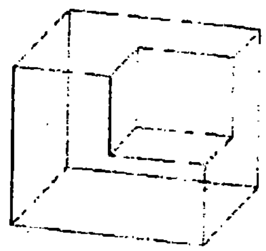

图 2　两四面体差运算的显示

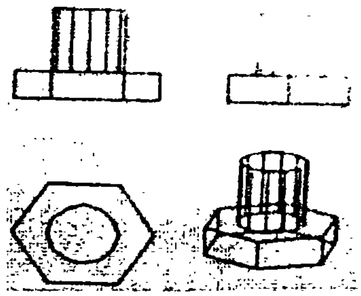

图 3　两多面体并运算后的三视图和轴测图

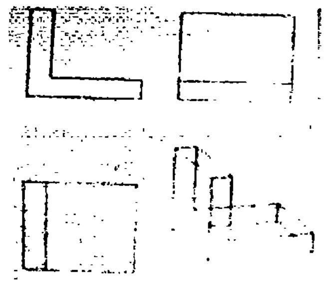

图 4　两多面体差运算后的三视图和轴测图

## 参考文献


1 李晓娟，殷光复.《工程图学》CAI 课件中组合体体素造型方法的研究:[中国农业大学学位论文].1996

2 姜献峰，孙毅.基于立体造型思想的组合体构形.浙江工业大学学报,2000

3 刘振安. 开放式立体造型系统三维图库.微机发展,1999(3)

4 Flickner M. Query by Image and Video Content. The QBIC System Computer , 1995,9

5 肖娜，黄毓瑜. 计算机图形智能评判技术研究与应用[J]. 工程图学学报, 2002 (1) : 62 ~ 71


(上接第 95 页)

在 MATLAB 命令窗中键入:[A,path] = floyd

运行结果是:

A =

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 1 | 7 | 3 | 6 | 7 | 9 |
| 1 | 0 | 6 | 2 | 5 | 6 | 8 |
| 7 | 6 | 0 | 4 | 4 | 1 | 3 |
| 3 | 2 | 4 | 0 | 3 | 4 | 6 |
| 6 | 5 | 2 | 3 | 0 | 1 | 3 |
| 8 | 7 | 1 | 5 | 5 | 0 | 2 |
| 10 | 9 | 3 | 7 | 7 | 2 | 0 |

path =

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| 1 | 2 | 3 | 4 | 4 | 4 | 4 |
| 2 | 2 | 3 | 4 | 5 | 6 | 6 |
| 2 | 2 | 3 | 4 | 5 | 5 | 5 |
| 4 | 4 | 6 | 4 | 5 | 6 | 6 |
| 3 | 3 | 3 | 3 | 3 | 6 | 7 |
| 6 | 6 | 6 | 6 | 5 | 6 | 7 |

所以，由最短距离矩阵 A 和最短路径矩阵 path，容易得出任意两点之间的最短路径及其长度。如从 $v_1$ 到 $v_7$ 的最短路径长度为 A[1,7] = 9，最短路径的下标:1→2→4→5→6→7。因为 path[1,7] = 2, path[2,7] = 4, path[4,7] = 5, path[5,7] = 6, path[6,7] = 7。这个结果与单源最短路径问题中的结果一致。

**结束语**　MATLAB 是一个专为科学和工程计算而设计的高级交互式软件包，它建立在向量、数组和矩阵的基础上，使用方便，人机界面直观，输出结果可视化。本文在 MATLAB 环境下编制了两个求解最短路径的程序，简捷易懂，操作方便。

## 参考文献


1 耿素云，屈婉玲，张立昂.离散数学[M]，第二版.清华大学出版社,2000

2 刘迎春. 一种实用的最短路径求解算法[J].浙江工业大学学报,2000(2)

3 傅鹂，龚劬，刘琼荪，等.数学实验[M].北京:科学出版社,2000

4 楼顺天，于卫，闫华梁.MATLAB 程序设计语言[M].西安:西安电子科技大学出版社,1998