

## 关键路径的稀疏矩阵求解算法

张春生

(内蒙古民族大学 数学与计算机科学学院, 内蒙古 通辽 028043)

(zhangcs\_817128@sina.com)

**摘 要:** 求解 AOE 网的关键路径算法一般基于拓扑排序, 虽然具有较好的时间复杂度 ( $O(n+e)$ ), 但由于必须进行拓扑排序, 同时还要进行拓扑逆序扫描, 使得算法本身比较复杂。针对这个问题提出了一个算法, 算法采用了稀疏矩阵作为数据的存储结构, 为防止关键路径丢失, 采用队列方式进行操作。同经典算法相比, 该算法简单, 时间复杂度相近 ( $O(n+e^2/n)$ )。

**关键词:** AOE 网; 关键路径; 稀疏矩阵

**中图分类号:** TP301.6 **文献标识码:** A

## Sparse matrix algorithm to solve the critical path

ZHANG Chun-sheng

(College of Mathematics and Computer Science, Inner Mongolia University for Nationalities, Tongliao Inner Mongolia 028043, China)

**Abstract:** The algorithm to solve the critical path of the AOE net is generally based on the topological sort. Although this algorithm has good asymptotic time complexity ( $O(n+e)$ ), it is comparatively complex because the topological sort and the topological inverted sequence scanning must be carried on. An algorithm was proposed, using sparse matrix as the storage structure of the data. To prevent the critical path from being lost, the queue method was adopted for the operation. Compared with the classical algorithm, this algorithm is simple, with close asymptotic time complexity ( $O(n+e^2/n)$ ).

**Key words:** AOE net; critical path; sparse matrix

## 0 引言

AOE 网即顶点表示事件、弧表示活动、权表示活动的持续时间的带权有向图。AOE 网在实际应用中经常表示一项工程。在 AOE 网中, 入度为 0 的顶点称为源点, 表示工程的开始; 出度为 0 的点称为汇点, 表示工程的完成。AOE 网中的活动可以并行地进行, 所以完成工程的最短时间是源点到汇点的最长的路径, 这个路径被称为关键路径。

通常求关键路径的算法是基于拓扑排序的<sup>[1]</sup>, 这种算法虽然具有较好的时间复杂度 ( $O(n+e)$ ), 然而算法中不仅要进行拓扑排序, 还要进行拓扑逆序扫描, 同时还要求最早发生时间和最迟发生时间, 所以, 算法本身比较复杂。

在关键路径求解方面, 探讨性的文章较多, 文献[2,3] 采用矩阵解法, 文献[4] 采用十字链表存储结构, 文献[5] 采用多项式约化, 虽然相对于经典算法有所改善, 但也存在着某些错误或局限性。

文献[2] 在求关键路径问题时, 提出了一个  $P$  矩阵求解算法, 该算法原理简单, 但算法本身存在错误。比如, 在本文例子中, 应用文献[2] 求得的关键路径为:  $v_1 v_2 v_5 v_7 v_9$  和  $v_1 v_2 v_5 v_8 v_9$ ; 而实际的关键路径为  $v_1 v_2 v_5 v_7 v_9$ ,  $v_1 v_2 v_5 v_8 v_9$ ,  $v_1 v_3 v_5 v_7 v_9$ ,  $v_1 v_3 v_5 v_8 v_9$ , 丢失了两个关键路径。

本文考虑到 AOE 网的邻接矩阵中 0 元素较多, 为降低时间复杂度, 采用了稀疏矩阵作为数据的存储结构; 为防止关键路径丢失, 采用队列方式进行操作, 算法简单, 时间复杂度低 ( $O(n+e^2/n)$ )。

## 1 相关概念

**定义 1** 设  $G = \langle V, E, W \rangle$  为  $n$  阶带权有向图,  $G$  中无环,  $G$  中存在一个入度为 0 的顶点, 称为源点, 存在一个出度为 0 的顶点, 称为汇点, 弧  $\langle v_i, v_j \rangle$  的权值记为  $W_{ij}$ , 表示活动的持续时间。

**定义 2** 设带权有向图  $G = \langle V, E, W \rangle$  具有  $n$  个节点, 节点的编号为  $v_i (i = 1, 2, 3, \dots, n)$ , 则源点的编号为  $v_1$ , 汇点的编号为  $v_n$ 。

**定义 3** 若一条路径中出现的节点编号不重复, 则此路径为基本路径。

**定义 4** 从源点到汇点的最长的基本路径称为关键路径。

**定理 1** 设  $d$  为带权有向图  $G = \langle V, E, W \rangle$  的关键路径, 则  $d$  的秩  $\leq n-1$ , 且为基本路径。

**证明:** 若  $d$  不是基本路径, 则  $G$  中必定存在环, 这与 AOE 网的定义矛盾, 所以  $d$  一定是基本路径。

因为  $d$  中节点不重复, 又由于  $d$  中最大的节点数为  $n$  个, 由此可知关键路径的最大边数为  $n-1$ , 所以,  $d$  的秩  $\leq n-1$ 。

**定义 5** 设  $L = \langle X, Y, W \rangle$  为图  $G$  的基本路径, 其中,  $X$  是由节点编号  $v_i$  表示的基本路径节点序列,  $Y$  为这个基本路径的终点编号,  $W$  为权值。

**定义 6** 设有基本路径  $L1$  和  $L2$ , 定义  $L1 \times L2$  的规则为:

- 1) 若  $L1.Y \neq L2.X$ , 则:  $L1 \times L2 = \emptyset$ 。
- 2) 若  $L1.Y = L2.X$ , 且  $L2.Y \notin L1.X$  中, 则:  $L1 \times L2 = \langle L1.X + L2.Y, L2.Y, L1.W + L2.W \rangle$ 。
- 3) 若  $L1.Y = L2.X$ , 且  $L2.Y \subset L1.X$ , 则:  $L1 \times L2 = \emptyset$ 。
- 4)  $L1 \times \emptyset = \emptyset$ 。
- 5)  $\emptyset \times L2 = \emptyset$ 。

**定义7** 设  $P^1$  是由 AOE 网中以源点为弧尾的弧并以  $L = \langle X, Y, W \rangle$  形式组成的稀疏矩阵, 其中  $X = v_1 v_i$ ,  $Y = v_i$ ;  $M$  是由 AOE 网中除去以源点为弧尾的弧以外的弧并以  $L = \langle X, Y, W \rangle$  形式组成的稀疏矩阵。

**定理2** 设  $P^k = P^{k-1} \times M$ , 按定义6进行运算, 则  $P^k$  为基本路径, 当  $P^k.L.Y = v_n$ ,  $P^k.L.W$  最大时,  $P^k.L$  为关键路径。

证明:

$\because P^1.L.X = v_1 v_i (i \neq 1)$

$\therefore P^1.L$  为基本路径(由定义3)。

设  $P^{k-1}.L$  为基本路径,  $P^k.L.X = P^{k-1}.L.X + M.L.Y$ , 由定义6, 若生成此路径  $M.L.Y \notin P^{k-1}.L.X$ , 则  $P^k.L.X$  中节点编号不重复, 由定义3可知,  $P^k.L$  为基本路径。

当  $P^k.L.Y = v_n$  时, 说明此基本路径从源点到汇点,  $P^k.L.W$  最大时, 说明权值最大, 由定义4可知,  $P^k.L$  为关键路径。

## 2 算法描述

### 1) 初始化

构造队列  $P^1: P^1$  由  $L \langle X, Y, W \rangle = \langle v_1 v_i, v_i, w_{ij} \rangle$  ( $w_{ij} \neq 0$ ) 构成。

构造  $M$  矩阵:  $M$  矩阵是由  $L \langle X, Y, W \rangle = \langle v_i v_j, v_j, w_{ij} \rangle$  ( $i \neq 1, j \neq 1, w_{ij} \neq 0$ ) 构成。

初始化 rpos[1:n], 该数组记录  $M$  矩阵各行的起始位置。

初始化 p\_length\_old 记录原  $P^k$  的长度。

初始化 p\_length\_new 记录新  $P^k$  的长度。

初始化 w\_max[1:n] 记录源点到各节点的权最大值。

初始化 tag 结束标志, tag = 1 表示结束。

### 2) 开始迭代

求  $P^k = P^{k-1} \times M$

```
do while tag < > 1 //结束标志不等于1, 则循环
{
    for i = 1 to p_length_old //对队列中所有元素循环
    {
        p 元素出队;
        for j = rpos[ P.Y ] to rpos[ P.(Y+1) - 1 ] {
            //对 P 中每个元素, 找对应的 M 中的元素
            P.L.X, Y, W > = < P.L.X + M.L.Y, M.L.Y,
            P.L.W + M.L.W >;
            If P.L.W > w_max[ P.L.Y ] //更新各节点的最大权值
                w_max[ P.L.Y ] = P.L.W;
            p_length_new = p_length_new + 1; //计算新队列中的元素数
            p 元素入队; }
        }
        p_length_old = p_length_new; //更新新队列中的元素数
        p_length_new = 0; //初始化整理后的元素数
        tag = 1;
        for i = 1 to p_length_old { //对当前队列循环
            p 元素出队;
            If P.L.W > w_max[ P.L.Y ] { //确定并选择权值最大的路径
                p 元素入队;
                p_length_new = p_length_new + 1; }
            //计算整理后的队列中的元素数
        }
        If P.L.Y < > v_n //设置更新结束标志
            tag = 0; }
        p_length_old = p_length_new;
        p_length_new = 0;
    }
}
```

### 3) 输出结果

队列  $P$  中路径为关键路径。

for i = 1 to p\_length\_old

printf( P.L.X );

//输出所有的关键路径

## 3 具体实例

有一个 AOE 网如图1所示, 求其关键路径。

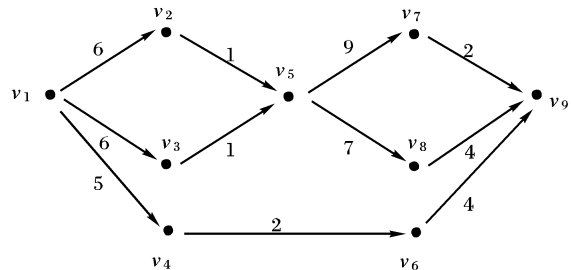


图1 AOE 网结构

### 1) 初始化

$P^1 = \{ \langle v_1 v_2, v_2, 6 \rangle, \langle v_1 v_3, v_3, 6 \rangle, \langle v_1 v_4, v_4, 5 \rangle \}$

$M = \{ \langle v_2 v_5, v_5, 1 \rangle, \langle v_3 v_5, v_5, 1 \rangle, \langle v_4 v_6, v_6, 2 \rangle, \langle v_5 v_7, v_7, 9 \rangle, \langle v_5 v_8, v_8, 7 \rangle, \langle v_6 v_8, v_8, 4 \rangle, \langle v_7 v_9, v_9, 2 \rangle, \langle v_8 v_9, v_9, 4 \rangle \}$

### 2) 求 $P^k = P^{k-1} \times M$

$P^2 = \{ \langle v_1 v_2 v_5, v_5, 7 \rangle, \langle v_1 v_3 v_5, v_5, 7 \rangle, \langle v_1 v_4 v_6, v_6, 7 \rangle \}$

$P^3 = \{ \langle v_1 v_2 v_5 v_7, v_7, 16 \rangle, \langle v_1 v_2 v_5 v_8, v_8, 14 \rangle, \langle v_1 v_3 v_5 v_7, v_7, 16 \rangle, \langle v_1 v_3 v_5 v_8, v_8, 14 \rangle, \langle v_1 v_4 v_6 v_8, v_8, 11 \rangle \}$

$P^4 = \{ \langle v_1 v_2 v_5 v_7 v_9, v_9, 18 \rangle, \langle v_1 v_3 v_5 v_7 v_9, v_9, 18 \rangle, \langle v_1 v_2 v_5 v_8 v_9, v_9, 18 \rangle, \langle v_1 v_3 v_5 v_8 v_9, v_9, 18 \rangle, \langle v_1 v_4 v_6 v_8 v_9, v_9, 15 \rangle \}$

### 3) 求关键路径

$P^4$  最大值为18, 故关键路径为:  $v_1 v_2 v_5 v_7 v_9, v_1 v_2 v_5 v_8 v_9, v_1 v_3 v_5 v_7 v_9, v_1 v_3 v_5 v_8 v_9$ 。

## 4 时间复杂度分析

设节点总数为  $n$ , 弧的总数为  $e$ , 由于  $P^1$  是由 AOE 网中以源点为弧尾的弧组成的稀疏矩阵, 所以,  $P^1$  元素个数最多为  $e/n$ , 同理,  $M$  矩阵的元素个数最多为  $e \times (n-1)/n$ 。

根据稀疏矩阵相乘规则, 得出矩阵相乘的复杂度为  $e^2/n$ , 队列中最大值扫描的复杂度为  $n$ , 所以求关键路径的时间复杂度为  $O(n + e^2/n)$ 。

本算法采用了稀疏矩阵作为数据的存储结构, 为防止关键路径丢失, 采用队列方式进行操作, 同经典算法相比, 算法简单, 时间复杂度相近。

### 参考文献:

- [1] 严蔚敏, 吴伟民. 数据结构[M]. 北京: 清华大学出版社, 1997. 183-186.
- [2] 徐凤生. 一种新的求解关键路径法[J]. 计算机应用与软件, 2005, 22(6): 97-99.
- [3] 徐利民. 关键路径的矩阵算法[J]. 淮南职业技术学院学报, 2001, 1(11): 100-102.
- [4] 徐凤生, 黄倩. 关键路径求解的新算法[J]. 计算机应用, 2004, 24(12): 108-109.
- [5] 赵灵芝, 陈小松. 关键路径新求法[J]. 贵州大学学报, 2002, 19(4): 302-304.