

# AOE 网络与关键路径算法

马欢飞\*

2017年5月

对于一个 AOE(Activity On Edge) 网络, 从源点到汇点的最长路径, 具有最大长度的路径叫**关键路径**。

因此, 对于AOE网络, 如何寻找关键路径, 本质上就是寻找两点间最长简单路径(无环路)问题。前面看到, 寻找最短简单路径的 Dijkstra 算法和 Floyd 算法, 都无法简单地改造为寻找最长路径, 事实上, 对于一般的有向图而言, 寻找最长路径是 NP-hard 的问题, 可以暂时理解为不可能在多项式复杂度内解决。但对于一般有工程背景的 AOE 网络, 事实上这类图可以归为 DAG(Directed Acyclic Graph) 图, 即有向无环图, 对于这类图, 不需要改造 Floyd 算法就可以得到最长路径, 因为有线性时间复杂度的解决方案, 而这方案的一个重要应用就是寻找 AOE 网络的关键路径。

对于一个 DAG 图, 从源点到汇点一定可以找到一个拓扑排序(由于没有环路), 所以得到算法如下:

**算法.** 对于一个 DAG 图  $G(V, E)$  和给定的源点  $s$  和终点  $e$ :

- 从源点  $s$  到汇点  $e$  确定一个拓扑排序序列, 同时初始化

$$\text{dist}(v) = \begin{cases} 0, & v = s \\ -\infty, & \text{else} \end{cases}$$

- 按照拓扑序列中的排序, 对每一个点  $v$  依次做如下操作:

$$\text{dist}(v) = \max_{\langle u, v \rangle \in E} \{ \text{dist}(u) + \omega(u, v) \}$$

则循环完毕后得到的即为  $s$  到  $e$  的最长路径长度。 ■

显然该算法复杂度为  $O(|V| + |E|)$ , 即线性复杂度。进一步的, 如果要确定该最长路径, 则从汇点出发, 按上述逆操作回到源点即可找到路径。

不难发现, 该算法即课本上寻找关键路径的算法。

---

\*hfma@suda.edu.cn