

# AOE网与关键路径

如果在有向图中用顶点表示事件，用弧表示活动，用弧上的权表示活动持续时间，称该带权有向图(即有向网)为边表示活动的网(activity on edge network),简称AOE网。

在AOE网中，只有一个顶点代表的事件发生后，从该顶点出发的各个弧所代表的活动才能开始，只有以弧头关联一个顶点的各个弧所代表的活动都已结束，该顶点所代表的事件才能发生。

一项工程可以由若干个子工程活动组成。用AOV网表示这项工程所关心的是各子工程之间的优先次序，即所得到的拓扑有序序列；而用AOE网表示这项工程所关心的是完成整个工程至少需要多少时间，哪些子工程是影响这项工程进度的关键活动，如何加快整个工程的进度等问题。

由于在AOE网中某些活动可以并行进行，所以完成工程的最短时间是从源点到汇点路径的最大长度(指路径上各活动持续时间之和最大，而不是路径上弧的数目最多)。把从源点到汇点路径长度最大的路径称作关键路径(critical path),关键路径上的活动称作关键活动。关键活动的长度是整个工程的最短工期，加快关键活动的完成是加快工程进度缩短工期地关键。

求AOE网关键路径算法的步骤：

- (1).输入e条弧 $\langle V_j, V_k \rangle$ ，建立AOE网的存储结构。
- (2).从源点 $V_1$ 出发，令 $ve[1]=0$ ;按拓扑有序序列次序求其余各顶点的最早发生时间 $ve[k](2 \leq k \leq n)$ ， $ve[k]=\max\{ve[j]+dut(\langle V_j, V_k \rangle)\}$ ;如果得到的拓扑有序序列中顶点个数小于网中顶点的个数n，说明网中存在环路，不能求关键路径算法终止，否则执行步骤(3)。
- (3).从汇点 $V_n$ 出发,令 $vl[n]=ve[n]$ ,按逆拓扑有序序列求其余各顶点的最迟发生时间 $vl[k](n-1 \geq k \geq 1)$ , $vl[k]=\min\{vl[V_j]-dut(\langle V_k-V_j \rangle)\}$ 。
- (4).根据各顶点的ve值和vl值，求每条弧的最早开始时间e[s]和最迟开始时间l[s];e[s]等于弧s的弧尾顶点 $V_k$ 的最早发生时间 $ve[k]$ ,而l[s]等于弧头顶点 $V_k$ 的最迟发生时间减去弧s的权值；若某条弧s满足 $e[s]=l[s]$ 则为关键活动，由所有关键活动构成的网的一条或几条关键路径。

```
1. void criticalpath(adjlist g[],int n)
2. {
3.     int i,j,k,e,l,dut;
4.     sqstack t;
5.     if(toporder(g,n,t)==0)
6.         printf("the AOE network has a cycle\n");
7.     else
8.     {
9.         for(i=n;i>=1;i--)
```

```
10.     vl[i]=ve[n];
11.     while(empty(t)!=0)
12.     {
13.         j=pop(t);
14.         p=g[j].next;
15.         while(p!=NULL)
16.         {
17.             k=p->adjvex;
18.             dut=p->data;
19.             if(vl[k]-dut<vl[j])
20.                 vl[j]=vl[k]-dut;
21.             p=p->next;
22.         }
23.     }
24.     for(j=1;j<=n;j++)
25.     {
26.         p=g[j].next;
27.         while(p!=NULL)
28.         {
29.             k=p->adjvex;
30.             dut=p->data;
31.             e=ve[j];
32.             l=vl[k]-dut;
33.             if(e==l)
34.                 printf("v%d to v%d:weight is %d\n",j,k,dut);
35.             p=p->next;
36.         }
37.     }
38. }
39. }
```