第6卷 第4期 华侨大学学报 Vol.6 No.4
1985年11月 JOURNAL OF HUACHIAO UNIVERSITY Nov. 1985

生成有向树的U.D.算法及其BASLC源程序

郑建德
（电子工程系）

摘 要

生成伴随有向图中全部给定参考点的有向k-树是以k-树法解有源网络的关键步骤.本文给出生成有向树的一种新算法－U.D.法.该法为迄今所发表的算法中最简单的一种，且具有较高的效率，可以编成简短的BASLC程序，应用于人机对话式的机助网络分析.

一、基 本 原 理

用k-树法解有源网络时，须求网络伴随有向图中全部给定参考节点的生成有向树(以下简称D-树)，这些树数目庞大.但若去掉树中各节点、支路的标号和标向，对应的无标无向树（以下简称U-树）数目却很少.我们可以反过来先构造U-树，再以各种可能方式给U-树的节点、支路加上标识符和标向，从而获得所需的D-树，这就是U.D.法的基本思路.一般认为文[1]，k-树法所处理的有源网络规模最大不超过10个节点，而节点数在10个以下的全部U-树仅200个，实用中，可把它们以数据形式存入计算机，以应付所有的计算需要.

图1给出某有源网络的伴随有向图G，它有6个节点，A节点为参考节点.图2给出具有6个节点的全部U-树，共计6个.以第三个为例，下面给出具体的加标步骤.为叙述方便，令T代表所考虑的U-树，并给它的每一个节点编上号，记编号为i的节点为V_i（图3(a)）.

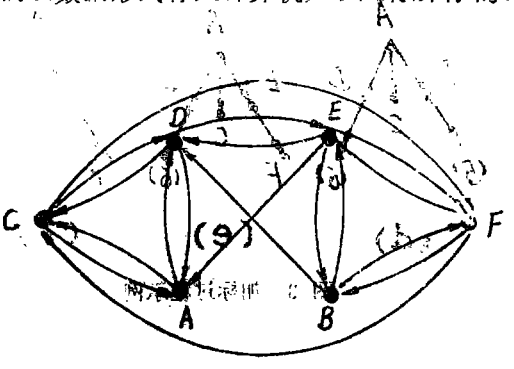

图1 伴随有向图G

图3所示的加标过程由两个步骤组成.

本文1984年12月19日收到.

第一步：在T中选一个节点 V_r，给T的各支路加上标向使之成为一棵以 V_r 为根的有向

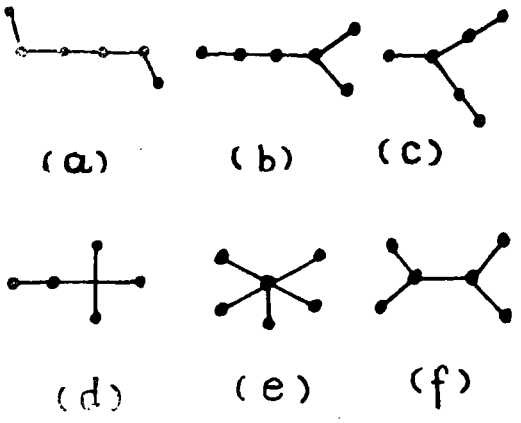

图 2 具有6个节点的公部 U-树

树 $\hat{T}_r$（图3(b)），再按父节点在前、子节点在后的原则给 $\hat{T}_r$ 的节点排上一个顺序，例如：

V_1 V_2 V_3 V_4 V_5 V_6

第二步：依排定的顺序逐个给 $\hat{T}_r$ 的节点加上合适的标识符。一般地说，各节点可加的标识符并不是唯一的。逐一考虑各节点的所有可能加标方式给出如图4所示的搜索树。树中

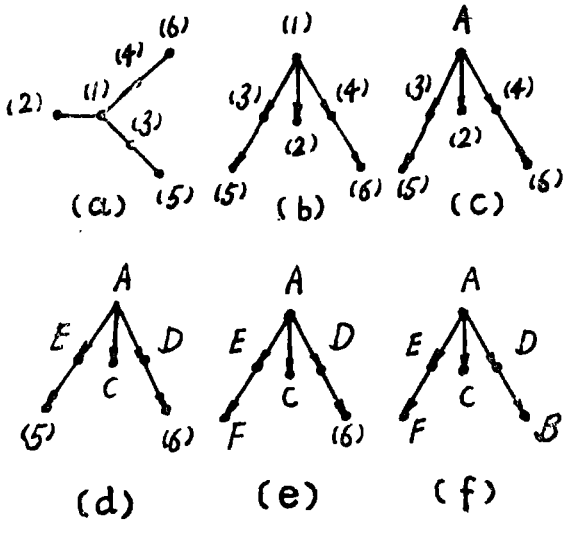

图 3 加标过程示例

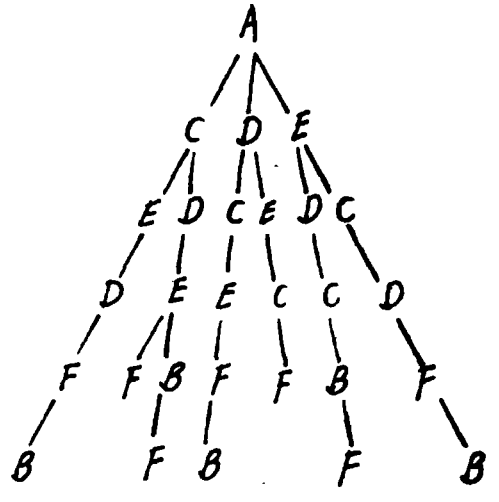

图 4 加标方式搜索树

第i行对应顺序第i个节点的加标方式，从根到最下面一行各节点所引的每条路径都对应G的标识符在 $\hat{T}_r$ 各个节点上的一个排列方式，也对应于G中的一棵D树，如图5所示。

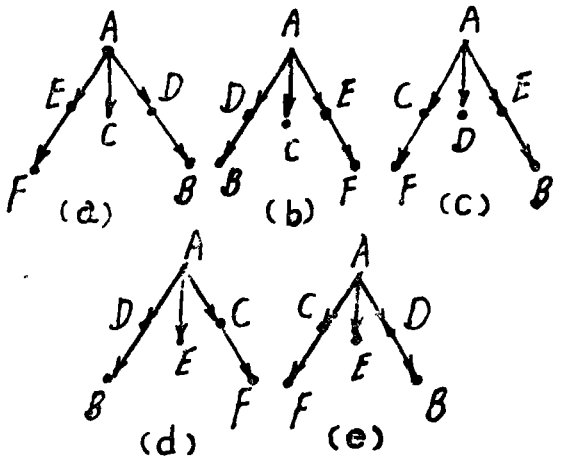

图 5　图 4 确定的 D-树

上述步骤对图 2 中其他 U-树同样适用，反复执行这两个步骤可以穷举图 2 中 6 个 U-树的加标方式从而获得 G 的全部生成有向树。但是，由于许多 U-树存在自同构，这样生成的 D-树也可能是冗余的。例如图 5 中，（b）、（e）分别是重复（a）、（d）的冗余项。

考虑 T 的点群，它由两个自同构组成：

$(V_1)(V_2)(V_3)(V_4)(V_5)(V_6)$

$(V_1)(V_2)(V_3V_4)(V_5V_6)$ 记 T 的点群为 $\Gamma(T)$，根据 $\Gamma(T)$ 可确定集合

$\{V_1, V_2, V_3, V_4, V_5, V_6\}$

上的一个划分

$$\Pi_1(T)=\{\overline{V_1}, \overline{V_2}, \overline{V_3V_4}, \overline{V_5V_6}\}$$

对 $\pi_1(T)$ 的同一等价块中任意两个元素 V_i，V_j，存在置换 $x\in\Gamma(T)$，满足

$$\alpha(V_i)=V_i$$

根据 $\Gamma(\hat{T}_r)$ 及 $\hat{T}_r$ 可以确定 $\{V_1, V_2, V_3, V_4, V_5, V_6\}$

上的另一个划分

$$\Pi_2(\hat{T}_r)=\{\overline{V_1}, \overline{V_2}, \overline{V_3V_4}, \overline{V_5}, V_6\}$$

$\Pi_2(\hat{T}_r)$ 中同一等价块的元素 V_i，V_j 不仅满足

$$\beta(V_i)=V_j \quad \beta\in\Gamma(\hat{T}_r)$$

还必须是 $\hat{T}_r$ 中的兄弟节点。

显然，在算法的第一个步骤中，如把根节点选在 $\Pi_1(T)$ 中同一等价块内的两个节点（如本例的 V_5、V_6 节点）上，第二步的执行结果将给出两组完全相同的 D-树。图5给出产生冗余项的另外一种情况。V_3、V_4 属于 $\pi_2(\hat{T}_r)$ 中同一等价块，仍记 $V_4=\beta(V_3)$，则令

$$L'(V_i)=L(\beta(V_i)) \quad i=1, 2, \cdots 6$$

给出另一种加标方式，但对应的却是同一棵 D-树。为了避免出现上述情况，在 U-D 算法中规定：

(1)在第一步中，$\pi_1(T)$的每个等价块只允许有一个元素被选为根节点。

(2)在第二步中，对同属$\pi_2(\hat{T}_r)$中一个等价块的V_i，V_j，规安$L(V_i) \leqslant L(V_j)$。其中，$\leqslant$是根据字母的自然顺序在{A，B,C,D,E,F,}上定义的一个偏序关系，$L(V_i)$、$L(V_j)$代表V_i、V_j所加的标识符。

在实际问题中，常常会碰到多重有向图。多重有向图的孤无法由所关联的节点唯一确安，不能直接采用上述的算法。处理多重图的好办法是合并平行孤。这一步可由人工完成，也可交由机器完成，如本文给出的程序所设计的那样。

二、关于算法源程序的一些说明

根据U．D算法编成的计算有向生成树BASIC源程序见第三节，本节先就程序的设计思想作一些说明。

1．程序中，有向图G的节点标识符改由数字表示。120、130号语句中由键盘输入的m、n、g、i、j、及k分别代表G的节点数、孤数、参考节点标号、各孤的起、止节点号及孤标号。

2．从$\pi_1(T)$的等价块中各取一个元素，接成链表，以E(i)代表V_i后继节点的编号。根据上节讨论，程序仅选链上节点作为根节点。

3．算法的第一步由子程序SUB A(300号～450号语句)完成。各节点在排好顺序后，其编号被接成另一链表。A(i)、B(i)分别存贮V_i前驱和后继节点标号。取A(n)=n+1，B(1)=-1，当A、B数组的下标上下确界定好之后，可借助于下标溢出对链首链尾作出判断。

4．$\hat{T}_r$的基本数据由F1、C1两个一维n元数组给出。F1(i)存V_i父节点的编号，C1(i)所存的是逻辑参数，C1(i)>0表示编号为B(i)的节点与V_i同属$\pi_2(\hat{T}_r)$的一个等价块，这时需取L(i)>L(B(i))。特别地，当中心被取为根节点时，相应的F1、C1另以F、C表示。E、F、C三个数组作为T的基本数据保存在DATA语句中。根据F、C可以算出任意$\hat{T}_r$对应的F1，C1。

5．算法第二步由子程序SUB B(500号～730号语句)。节点标号以大小为序接成链表，并以P(i)、Q(i)指示前驱及后继标号。取

Q(n)=n+1

在给$\hat{T}$的一个节点标上号后，应把该标号从链上卸下来；而在给某个节点改换标号时，则要把原来的标号重新接入链中。SUB B引入了深度优先搜索法，并利用BASIC Ⅱ的错误收集语句，借助于下标溢出错误，实现流程控制，减少了一些判别语句，提高了计算效率。

三、BASIC 源 程 序 清 单

```
100  REM Main Program
110  INPUT "m, n, g=" m, n, g
120  DIM A(n), B(n), P(n), Q(n), S(n),
       D(n), F(n), E(n), C(n), F1(n),
       C1(n), U(n), Y(n, n)
130  FOR t=1 TO m:
       INPUT "i, j, k=" i, j, k:
       Y(i, j)=Y(i, j)*100+k:
     NEXT t
140  FOR i=3 TO 10:READ U(i):NEXT i
150  FOR i=3 TO n-1:FOR j=1 TO 3*i*u(i):
       READ S1:NEXT j, i
160  FOR i=0 TO n
170  D(i)=i+1:Q(i)=i-1:A(i)=i+1
180  NEXT i
190  FOR i=1 TO n:FOR j=1 TO n:
       IF Y(i, j)>0 THEN D(i)=D(i)+1:
     NEXT j, i
200  FOR h=1 TO U(n)
210  FOR i=1 TO n:
       READ F(i), E(i), C(i):
     NEXT i
220  Y=0:E(0)=1
230  GOSUB 300:IF r=0 GOTO 250
240  GOSUB 500:GOTO 230
250  NEXT n
260  END
300  REM SUB A
310  Y=E(r):IF r=0 THEN RETURN
320  d1=D(r)
330  FOR j=2 TO n:
     IF F(j)=r THEN d1=d1-1:NEXT j
340  IF r>1 THEN d1=d1-1
350  IF d1<0 GOTO 310
360  FOR i=1 TO n:C1(i)=C(i):
```

```
        F1(i)=F(i):NEXT i
370 F1(r)=-1:A(0)=r:k=r
380 IF k=1 THEN 430
390 IF C(k)<3 OR C(k)>-2 THEN 410
400 C1(k)=0:A(k-1)=A(k):F1(F(k))=k
420 A(k)=F(k):k=F(k):GOTO 380
430 FOR k=1 TO n:B(A(k))=k:NEXT k
440 L=g:T=r
450 RETURN
500 REM SUB B
510 ON ERROR GOTO 690
520 P(Q(L))=P(L):Q(P(L))=Q(L):S(T)=L
530 T=A(T):M=S(F1(T))
540 IF C1(T)>0 THEN L=P(L) ELSE L=P(0)
550 IF Y(M, L)>0 THEN 520
560 L=P(L):GOTO 550
570 T=B(T):M=S(F1(T))
580 L=S(T):P(Q(L))=L:Q(P(L))=L
590 GOTO 560
600 FOR i=1 TO n
610 i1=F1(i):IF i1=0 THEN 670
620 k1=Y(i1, i1)
630 IF k1<100 THEN 660 ELSE LPRINT "(",
640 k2=INT(k1/100):LPRINT "Y", k1-100*k2
650 k1=k2:IF k1=0 THEN LPRINT ")",
    ELSE 640
660 IF K1<100 THEN LPRINT "Y", k1;
670 NEXT i
680 LPRLNT
690 IF ERL=550 THEN RESUME 570
700 IF ERL=530 THEN RESUME 600
720 STOP
730 RETURN.
```

参 考 文 献

〔1〕陈树柏等，网络图论及其应用，科学出版社，（1982）。
〔2〕F. 哈拉里著，李慰萱译，图论，上海科学技术出版社，（1980）。

The U.D.Algorithm for the Computation of Directed Trees ——— with a BASIC Program

Zheng Jiande

Abstract

The paper devoeops a new algorithm to accompeish the task of computing all directed trees with given root in a directcd graph, which is the key step tu solve active networks using k-tree technigue. The algovithm is the simplest one among those having beiny put forwowd and being proved to be efficient