第 27 卷 第 5 期
1999 年 10 月
福州大学学报(自然科学版)
Journal of Fuzhou University(Natural Science)
Vol.27 No.5
Oct.1999

文章编号：1000－2243(1999)05－0010－04

若干NP 完全问题的特殊情形

王晓东
(福州大学计算机科学与技术系，福建 福州 350002)

摘要：讨论了图算法中若干NP 完全问题在所给的图是一棵树时的特殊情形．利用树结构的前序编号表示法提出了解树的最大独立集问题、最小顶点覆盖问题和最小支配集问题的线性时间算法．在渐近意义下这些算法都是最优算法．

关键词：图；树；NP 完全问题；计算复杂性
中图分类号：TP3　　**文献标识码：**A

1 引言

在图上定义的许多组合优化问题是NP 完全问题．这类问题属于较难解的问题，至今没有找到多项式时间算法，也很可能根本没有多项式时间算法．遇到这类问题时，通常从以下几个方面来考虑并寻求解决办法．

1) 特殊情形：仔细分析所遇到的NP 完全问题，研究具体实例的特殊性，考虑是否必须在最一般的意义下来解此问题．也许可利用具体实例的特殊性，在特殊条件下解此问题．许多NP 完全问题在特殊情形下可以找到多项式时间算法．例如求图G 的最大团问题是NP 完全问题，而在图G 是平面图的情形下，该问题是多项式时间可解的．

2) 动态规划和分枝限界方法：对于许多NP 完全问题来说，用动态规划和分枝限界方法常可得到较高的解题效率．

3) 概率分析：对于许多NP 完全问题，其困难实例出现的概率很小，因此对这类NP 完全问题常可设计出平均性能很好的算法．

4) 近似算法：通常可以设计出解NP 完全问题的多项式时间近似算法，以近似解来代替最优解．

5) 启发式算法：在用别的方法都不能奏效时，也可以采用启发式算法来解NP 完全问题．这类方法根据具体问题的启发式搜索策略来求问题的解，在实际使用时可能很有效，但有时很难说清它的道理．

本文考虑关于图的若干NP 完全问题的特殊情形．当所给的图G 是一棵树时，许多NP 完全问题可在多项式时间内求解．特别地，对于图G 的最小顶点覆盖问题、最大独立集问题和最小支配集问题等都是NP 完全问题[1]．而在图G 是一棵树时，可以有效地解决．本文利用树的前序编号为工具，提出解决上述问题的 $O(n)$ 时间算法．

收稿日期：1998－09－21
作者简介：王晓东(1957－)，男，教授．

基金项目：福建省自然科学基金资助项目(F9810009－JS98－1)

2　树的前序标号表示法

给定一棵树 T，在计算机中可以有多种表示方法[2]．在图论算法中，树是作为一般的图来表示的，通常采用邻接表表示法．针对所考虑问题的特殊性，提出树的前序标号表示法如下：

对于给定的用邻接表表示的有 n 个顶点的树 T．

①任选一个顶点 r 作为树 T 的根结点；

②对以 r 为根的树作前序遍历，并且在遍历过程中对访问的顶点依次编号；

③用数组parent 记录每个结点的父结点编号．即编号为 i 的结点的父结点编号为parent[i]．

这个表示过程显然可在 $O(n)$ 时间内完成．这种表示法实际上是树在前序标号意义下的父亲数组表示法[2]．它具有以下重要性质：

1) 对于 $i=2,\cdots,n$，有 $i>\text{parent }i$，当 $i=1$ 时parent[i] $=1$；

2) 若将树 T 看作是一般的图 $G=(V,E)$，则有：

$V=\{1,2,\cdots,n\}$，$E=\{(i,\text{parent}[i]),i=2,\cdots,n\}$；

3) 对于任意 j，$2\leqslant j\leqslant n$，定义树 T 的子树 $T_j=(V_j,E_j)$ 为：

$V_j=\{1,2,\cdots,j\}$，$E_j=\{(i,\text{parent}[i]),i=2,\cdots,j\}$，则parent[$1..j$] 是子树 T_j 的前序标号表示．特别地，$T_n=T$；

4) 标号为 j 的结点是子树 T_j 的叶结点，$j=2,\cdots,n$．特别地，标号为 n 的结点是 T 的叶点．

在树的前序标号表示法下，许多关于树的运算得以简化．

3　最小顶点覆盖

顶点集 S 是图 $G=(V,E)$ 的顶点覆盖，当且仅当对任意 $(u,v)\in E$ 有 $u\in S$ 或 $v\in S$．最小顶点覆盖问题是对给定的图 G 找出使 $|S|$ 最小的顶点覆盖 S．当所给的图是一棵树 T 时，可以设计出求 T 的最小顶点覆盖的贪心算法如下：

```
MIN-VERTEX-COVER(T)
begin
    for i := 1 to n do
      cover[i] := 0;
    S := ∅;
    for i := n downto 2 do
      if (cover[i] = 0) and (cover[parent[i]] = 0) then
        begin
          S := S ∪ {parent[i]};
          cover[parent[i]] := 1;
        end
  end; {MIN-VERTEX-COVER}
```

该算法是一个贪心算法．算法中用数组cover 来标记选入覆盖点集的树结点，即当结点 i 被选入覆盖点集，则cover[i] $=1$，否则cover[i] $=0$．为了说明算法的正确性，必须证明关于

树的最小顶点覆盖问题满足贪心选择性质并且具有最优子结构性质[2]．

1) 贪心选择性质：对于树 T，存在一个 T 的最小顶点覆盖 S，使 parent[n] $\in S$．事实上，设 s 是 T 的一个最小顶点覆盖，若 parent[n] $\notin S$，则 $n \in S$．否则 S 就不是 T 的一个顶点覆盖．这种情况下，令 $\bar{S} = S \cup \{parent[n]\} - \{n\}$，则 $\bar{S}$ 仍为 T 的一个顶点覆盖，且 $|\bar{S}| = |S| - 1 + 1 = |S|$．故 $\bar{S}$ 是 T 的一个最小顶点覆盖，且 parent[n] $\in \bar{S}$．

2) 最优子结构性质：对 T 的任一最小顶点覆盖 S，当 $n \in S$ 时，显然 $S - \{n\}$ 是 T_{n-1} 的一个顶点覆盖，下面证明 $S - \{n\}$ 是 T_{n-1} 的一个最小顶点覆盖．不然，存在 T_{n-1} 的一个更小的顶点覆盖 $\bar{S}$，且 $|\bar{S} < |S| - 1$，则 $\bar{S} \cup \{n\}$ 显然是 T 的一个顶点覆盖．但是 $|\bar{S} \cup \{n\}| < |S| - 1 + 1 = |S|$．这与 S 是 T 的一个最小顶点覆盖矛盾．

当 $n \notin S$ 时，设 i = parent[n]，则必有 i $\in S$，令 $S_1 = \{j \mid i \leqslant j \leqslant n\} \cap S$，则 $S - S_1$ 是 T_{i-1} 的一个最小顶点覆盖．事实上，由于 S 是 T 的一个顶点覆盖，则 $S - S_1$ 是 T_{i-1} 的一个顶点覆盖．若在 T_{i-1} 中有一个更小的顶点覆盖 S_{i-1} 使 $|S_{i-1}| < |S| - |S_1|$，则 $S_{i-1} \cup S_1$ 是 T 的一个顶点覆盖，且 $|S_{i-1} \cup S_1 \leqslant |S_{i-1}| + |S|_1 < |S| - |S_1| + |S_1| = |S|$，这与 S 是 T 的最小顶点覆盖矛盾．

根据上述的贪心选择性质和最优子结构性质，容易用数学归纳法证明算法 MIN-VERTEX-COVER 能正确找出 T 的最小顶点覆盖．

算法的 for 循环显然只需要 $O(n)$ 的时间，从而整个算法所需的时间为 $O(n)$．

4 最大独立集

顶点集 S 是图 $G = (V, E)$ 的独立集当且仅当 S 中任何 2 个顶点在 G 中是不相邻的．最大独立集问题是要对给定的图 G 找出使 $|S|$ 达到最大的 G 的独立集 S．当所给的图是一棵树 T 时，我们用树的前序标号表示法表示它，并设计一个找出其最大独立集的贪心算法如下：

```
MAXINDEPENDENT-SET(T)
begin
    for i := 1 to n do
        cover[i] := 0;
    S := Ø;
    for i := n downto 2 do
        if cover[i] = 0 then
            begin
                S := S ∪ {i};
                cover[parent[i]] := 1;
            end;
    if cover[1] = 0 then S := S ∪ {1}
end; {MAX-INDEPENDENT-SET}
```

求树 T 的最大边独立集也可用类似的算法实现如下：

```
MAX-EDGE-INDEPENDENT-SET(T)
begin
    for i := 1 to n do
        cover[i] := 0;
```

```
    S := ∅;
    for i := n downto 2 do
      if (cover[i] = 0) and (cover[parent[i]] = 0) then
        begin
          S := S ∪ {(i, parent[i])};
          cover[parent[i]] := 1
        end
end; {MAX-EDGE-INDEPENDENT-SET}
```

5 最小支配集

顶点集 D 是图 $G=(V,E)$ 的支配集当且仅当对任意 $u \in V-D$ 存在 $v \in D$ 使 $(u,v) \in E$. 最小支配集问题是对给定图 G 找出使 $|D|$ 最小的支配集 D. 当所给的图是一棵树 T 时，我们可以利用树的前序标号表示法设计出求最小支配集 D 的线性时间算法如下：

```
MIN-DOMINATE-SET(T)
begin
    for i := 1 to n do
      cover[i] := 0;
    D := ∅;
    for i := n downto 2 do
      if cover[i] = 0 then
        begin
          D := D ∪ {parent[i]};
          cover[parent[i]] := 1;
          cover[parent[parent[i]]] := 1
        end
end; {MIN-DOMINATE-SET}
```

最小支配集问题同样具有贪心选择性质和最优子结构性质，从而保证了算法 MIN-DOMINATE-SET 的正确性，算法所需的计算时间也是 $O(n)$.

参考文献：

[1] Garey M R, Johnson D S. Computers and intractability, a guide to the theory of NP-completeness [M]. California: Freeman and Co, 1979.

[2] 傅清祥，王晓东. 算法与数据结构[M]. 北京：电子工业出版社，1998.

Some Special Cases of NP-Complete Problems

WANG Xiao-dong

(Department of Computer Science and Technology, Fuzhou University, Fujian Fuzhou 350002, China)

Abstract: This paper discusses some special cases of NP-complete graph problems in which the given graph is a tree. By means of the pre-order labeling presentation of a tree, we present several linear time algorithms for graph problems on trees. These algorithms are all asymptotically optimal.

Keywords: graphs; trees; NP-complete problems; time complexities