第13卷 第5期 电子科学学刊 Vol. 13, No. 5
1991 年 9 月 JOURNAL OF ELECTRONICS Sept., 1991

有源网络完全 k 树多项式的产生及其在网络分析中的应用*

房 大 中
(天津大学自动化系,天津)

摘要 本文算法产生有源网络无源树边的完全 k 树多项式，算法时间复杂度与列写无向图全部树的改进的 Minty 算法相同. 用该算法分析有源网络的符号函数可有效地减少对消冗余项数,同时也避免了对无源完全树边的符号鉴别问题. 文章讨论了算法的合理性,并举例说明了它在网络分析中的应用.

关键词 网络分析;有源网络拓朴图;完全 k 树多项式

1. 前言

本文算法入手处理的是有源网络拓朴图,在不同的递归层次上,按 Minty 算法[1,2]分解算法中当前拓朴图，产生以无源边为完全树边的 k 树多项式. 在上述过程中，不必考虑无源树边的符号鉴别和对消项问题是其主要特点. 伴随上述过程，算法中当前图将收缩为严格有源拓朴图(定义见后)或一顶点. 若算法配有完全树法[3]或有向树法[4]子程序处理算法中的严格有源拓朴图,算法可有效地产生有源网络符号函数.

2. 拓朴图画法

图 1 所示为某一三晶体管二端口网络 N 的小信号线性增广模型网络 $\hat{N}$[5]. 若 $\hat{N}$ 的每一电压控制电流源 (VCCS) 分别用一对权值等于其控制电导的有向电流和有向电压对偶边代换,而 $\hat{N}$ 的每一无源二端元件用一权值为其导纳值的无向边代换，则可得 $\hat{N}$ 的拓朴图 $\hat{G}$ 如图 2 所示.

3. 拓朴图的分解

令 E_p, E_i 和 E_v 分别表示算法当前图 G 的无源电流和电压边集,即

$$G \triangleq G(E_p + E_i) \cup G(E_p + E_v)$$

若 $E_p = \varnothing$，则称 G 为严格有源拓朴图. 同时用符号 $Y \triangleq y_1 + y_2 + \cdots + y_q \in E_p$ 表示 G 的一组无源平行边集[5,6], Y 在算法中被视为一无源边的权值，且当 $q = 1$ 时 Y 退化为无源边 y_1 本身. 令 $G_1 \triangleq G \ominus Y$ 和 $G_2 \triangleq G - Y$ 分别为以 Y 分解 G 所得的缩边图和去边图[5,6],并令 P_G, P_{G_1} 和 P_{G_2} 分别表示图 G, G_1 和 G_2 的完全树多项式,则按文献[3]中完全树的定义和文献[1]中图的分解算法,有

$$P_G = Y P_{G_1} + P_{G_2} \tag{1}$$

1990年3月15日收到，1990年10月15日修改定稿.
* 国家自然科学基金资助项目.

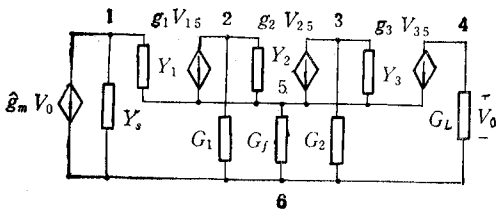

图 1　某二端口网络 N 的增广模型网络 $\hat{N}$

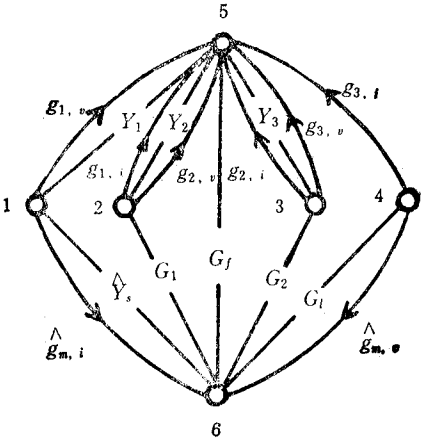

图 2　$\hat{N}$ 的拓朴图 $\hat{G}$

显然当 Y 为图 $G(E_p+E_i)$ 或 $G(E_p+E_v)$ 的桥时，上式退化为

$$P_G = Y P_{G_1} \tag{2}$$

若 G 为初始图 $\hat{G}$，并按上述同样方法分解 G_1 和 G_2 直至其为一个顶点或为严格有源拓朴图为止．如此便可得到由一系列边 Y 构成的无源完全 k 树多项式及其相应的一些严格有源拓朴图(见后例)．注意到(1),(2)两式中没涉及 Y 的符号是重要的．

4. 符号问题

对图 $\hat{G}$ 任意指定其无源边的参考方向和参考节点，由文献[3]有

$$|\hat{Y}_n| = |A_i \hat{Y} A_v^T| = \sum_{\forall k} \varepsilon_k \cdot \text{完全树 } T_k \text{ 树边导纳积} \tag{3}$$

上式右部对 $\hat{G}$ 的全部完全树求和，A_i 和 A_v 分别是 $G(E_p+E_i)$ 和 $G(E_p+E_v)$ 的降阶关联矩阵，$\hat{Y}_n$ 和 $\hat{Y}$ 分别是 $\hat{G}$ 对应网络 $\hat{N}$ 的节点方程系数矩阵和支路导纳矩阵．ε_k 是 T_k 对 $|Y_n|$ 各项贡献的符号，且

$$\varepsilon_k \triangleq |A_{kv,0}|\,|A_{ki,0}| = \pm 1 \tag{4}$$

式中 $A_{kv,0}$ 和 $A_{ki,0}$ 分别是 T_k 对应 A_v 和 A_i 的大子矩阵．不失一般性，设定 $T_k = \{y_{k_1}, y_{k_2}, \cdots, y_{k_q}, y_{k_{q+1}}, \cdots, y_{k_{n-1}}\}$，其中 $\{y_{k_1}, y_{k_2}, \cdots, y_{k_q}\} \subset E_p$，假定 $y_{k_1} = \langle n, s\rangle$，于是示意地可写出：

$$A_{kv,0} = \begin{matrix} \\ n \\ s \end{matrix}\!\left[\begin{array}{c|c} y_{k_1} & \\ 1 & \times \\ -1 & \end{array}\right]; \quad A_{ki,0} = \begin{matrix} \\ n \\ s \end{matrix}\!\left[\begin{array}{c|c} y_{k_1} & \\ 1 & \times \\ -1 & \end{array}\right]$$

若对上述两矩阵分别做：(1)第 s 行各元素分别加至第 n 行各同列元素上，(2) 从所得矩阵删除 s 节点对应行和 y_{k_1} 导纳对应列．最后所得矩阵分别记做 $A_{kv,1}$ 和 $A_{ki,1}$，考虑行列式展开性质有

$$\begin{aligned} \varepsilon_k &= |A_{kv,0}|\,|A_{ki,0}| = (-1)^{w_v} |A_{kv,1}| (-1)^{w_i} |A_{ki,1}| \\ &= (-1)^{2w_i} |A_{kv,1}|\,|A_{ki,1}| = |A_{kv,1}|\,|A_{ki,1}| \end{aligned} \tag{5}$$

式中 $w_v = w_i = 1 + s$ 节点行序号 $+ y_i$ 边列序号．不难验证，$A_{kv,1}$ 和 $A_{ki,1}$ 恰为 $\hat{G} \ominus y_{k_1}$ 缩边图完全树 $\{y_{k_2}, \cdots, y_{k_q}, \cdots, y_{k_{n-1}}\}$ 对其电压图和电流图的降阶关联矩阵．重复上

述过程 q 次有

$$\varepsilon_k = |A_{k_v,q}||A_{k_i,q}| \tag{6}$$

式中 $A_{k_v,q}$ 和 $A_{k_i,q}$ 是由 $\hat{G}$ 依次收缩边 $y_{k_1}, y_{k_2}, \cdots, y_{k_q}$ 后所得缩边图完全树 $\{y_{k_{q+1}}, \cdots, y_{k_{n-1}}\}$ 相应其电压图和电流图的降阶关联矩阵。(6) 式说明 $\hat{G}$ 完全树多项式的符号由算法伴生的严格有源图确定。

5. 算法要点

由(1)、(2)式容易按文献 [2, 6] 描述的递归过程实现产生 $\hat{G}$ 的全部无源 k 树多项式的算法过程。但注意下述各点对提高算法效率是重要的。

(1) 缩边图的处理　当前图中的有源边是成对出现的，则称 $g_{k,i}$ 是 $g_{k,v}$ 的对偶边；反之亦然。在缩边图 $G\ominus Y$ 中，原 G 图中与 Y 平行的有源边将变成自环边，算法应及时将 $G-Y$ 中的自环边及其对偶边从 $G\ominus Y$ 中删除。另一方面 $G\ominus Y$ 中也可能出现由某对对偶边 $\{g_{k,i}, g_{k,v}\}$ 形成的平行边，若这种情况发生应依 $g_{k,i}$ 和 $g_{k,v}$ 是同向还是反向用权值为 g_k 或 $-g_k$ 的无源边替代该对有源边[3]。

(2) 去边图的处理　称图 G 中不同时含有电压边和电流边的割集为 A 类割集。若 G 存在 A 类割集，则算法总是从同一 A 类割集中选择边 $Y\in E_p$ 去分解 G 和 G 的去边图。当该 A 类割集只剩最后一无源边 Y 时，算法不再处理其去边图，而按(2)式分解当前图。当 G 不存在 A 类割集，而存在 B 类割集时，算法总是从同一 B 类割集中选择 $Y\in E_p$ 去分解 G 和 G 的去边图。所谓 B 类割集是仅含一条某类有源边的非 A 类割集。为方便叙述，假定我们选定的 B 类割集仅含一条电流边 $g_{k,i}$，并令 G_t 是从 B 割集去掉所有无源边后所得的去边图，则不难设想 $g_{k,i}$ 必定是 G_t 相应电流图的桥，因而 $g_{k,i}$ 及其对偶边必定为 G_t 的每一完全树的树边。所以应将 $g_{k,i}$ 的对偶边 $g_{k,v}$ 的每一平行边从 G_t 中删除。

(3) 编程方案　为节省计算机内存，算法在每一层次上分解当前图都是将去边图推入堆栈，待缩边图处理完后，再返回处理去边图[2]。

6. 算例

考虑生成图 1 所示网络 $\hat{N}$ 所有完全树问题。对图 2 所示 $\hat{G}$ 依次从 A 类割集 $\{\hat{g}_{m,i}, \hat{Y}_s, G_1, G_f, G_2, g_{3,i}\}$ 选择无源边分解 $\hat{G}$ 及其缩边图。当当前图转变为严格有源拓朴图后，继续用文献[3]中的方法生成其完全树多项式。所得 $\hat{G}$ 完全树多项式如下：

$$\begin{aligned}P_{\hat{G}} = {} & \hat{Y}_s G_L(G_2(G_1(Y_1+Y_2+Y_3+g_1+g_2+g_3+G_f)+Y_2(Y_1+Y_3\\ & +G_f+g_3)+g_1^*(-g_2^*))+G_1(Y_3(Y_1+Y_2+G_f+g_1)+g_2^*(-g_3^*))\\ & +Y_2Y_3(Y_1+G_f)+g_1^*g_2^*g_3^*)+G_1(G_LY_1(G_2(Y_2+Y_3+G_f+g_2+g_3)\\ & +Y_3(Y_2+G_f)+g_2^*(-g_3^*))+G_2(Y_1\hat{g}_m^*g_3^*+\hat{g}_m^*g_1^*g_3^*)\\ & +Y_1\hat{g}_m^*g_2^*(-g_3^*)+\hat{g}_m^*g_1^*g_2^*(-g_3^*))+G_f(G_LY_1Y_2(Y_3+G_2)\\ & +g_m^*g_1^*g_2^*(-g_3^*))+G_2(G_LY_1Y_2(Y_3+g_3)+Y_1Y_2\hat{g}_m^*g_3^*+\hat{g}_m^*g_1^*g_2^*(-g_3^*))\end{aligned}$$

式中带 * 号的完全树边由算法中伴生的严格有源拓朴图产生。由 $P_{\hat{G}}$ 不难看出算法中伴生的每一严格有源拓朴图恰为一完全树。此时只需按文献[3]提供的方法确定其符号即可。由 $P_{\hat{G}}$ 通过分离其中带 $\hat{Y}_s$ 和 $\hat{g}_m$ 的完全树多项式容易写出 $\hat{N}$ 对应的二端口网络 N 的各种符号函数[5]。

7. 结论

本文产生有源网络无源边完全 k 树多项式的系统方法可较好地改进传统的完全树列举算法的有效性。

本文曾经杨山教授审阅，并提出宝贵意见，在此致以谢意。

参　考　文　献

[1] G.J. Minty, *IEEE Trans, on CT*, **CT-12** (1965)3, 119.
[2] 房大中，天津大学学报，1988 年，第 4 期，第 108 页。
[3] W. Mayeda, Graph Theory, John Wiley & Sons, Inc., (1972).
[4] W. K. Chen, *IEEE Trans. on CT*, **CT-12** (1965)3, 85.
[5] 仝茂达，朱英辉编：符号网络函数与不定导纳矩阵，高等教育出版社，1984年。
[6] 黄汝激，电子学报，1987 年，第 5 期，第 8 页。

AN ALGORITHM FOR GENERATING ALL PASSIVE-EDGE COMPLETE k-TREE AND ITS APPLICATION IN THE ANALYSIS OF ACTIVE NETWORKS

Fang Dazhong
(Tianjin University, Tianjin)

Abstract　An efficient algorithm for generating all passive-edge complete k-trees in symbolic form for general linear active networks is presented. This algorithm, which is based on Minty's method, processes the topological graph of an active network directly. It can solve the problem of sign evaluation, and can reduce the number of cancellation terms greatly. Finally, an example is given to show its application in the analysis of linear active networks.

Key words　Network analysis; Topological graph of active networks; Complete k-tree polynomial