第11卷 第3期 电子科学学刊 Vol. 11, No. 3
1989 年 5 月 JOURNAL OF ELECTRONICS May, 1989

产生符号网络函数的主子超图法*

黄 汝 激
(北京钢铁学院自动化系，北京)

摘要 文中引入完全超树和主子超图的概念，并提出产生线性有源网络之符号网络函数的一个新方法——**主子超图法**. 它是主子图法[1]的改进. 它的表达式很紧凑，计算时间复杂度为 $O(ve\sum n_i)$. 通常叶总数 $\sum n_i$ 远小于主子图数 n_p，所以它的计算效率高于主子图法.

关键词 网络，符号网络函数；主子超图法；完全超树

一、引 言

文献[1]提出了主子图概念和分析线性有源网络的**主子图法**. 它是完全树法与有向树法的统一，但是其中主子图是一个一个产生出来的，效率较低. 本文应用文献[2]提出的有向超图理论，引入完全超树和主子超图的概念，并提出产生线性有源网络之符号网络函数的一个新的拓扑方法——**主子超图法**. 通过它可以成批地产生主子图，所以所得表达式较紧凑，而且计算效率较高.

二、主子超图法的原理

考虑一个线性有源网络 N'，它含有 $s-1$ 个互导纳为 g_α 的压控流源 (VCCS) α，$\alpha=1,2,\cdots,s-1$，n 个顶点 $1,2,\cdots,n$ 和 e 条无源边 $y_h=[PE(1,h),\ PE(2,h)]$，$h=1,2,\cdots,e$($2\times e$ 矩阵 PE 存放了所有 y_h 的始终点)，它的输入口为 (i_1,i_2)，输出口为 (j_1,j_2). 求它的符号开路转移阻抗 $Z_{i_1i_2\to j_1j_2}$.

首先，把一个互导纳为 g_t 的正反馈压控流源 $g_tV_{i_1i_2}=-I_{i_1i_2}$ 接到 N' 的两个口，得到 N' 的**增广网络** N^*. 把 N^* 的所有 VCCS 拉出来，剩下的**无源子网络**记作 N，它的图为 G，G 的邻接矩阵和关联矩阵各记作 J 和 A. VCCS 的**电压(v)边**和**电流(i)边**的端点各称为 **v 点**和 **i 点**[1]，总称为**源点**. 设源点总数为 v，则 G 对应于一个 v 端超边[2] (称为**无源超边**) E. VCCS 的 v 边、i 边和伴随有向图[1,3]分别记作 $e_{v\alpha}=(VE(1,\alpha),\ VE(2,\alpha))$，$e_{i\alpha}=(IE(1,\alpha),\ IE(2,\alpha))$ 和 $g_{a\alpha}$，$\alpha=1,\cdots,s$. 若 N^* 中每个 VCCS α 用 $g_{a\alpha}$(其边称为**有向边** $e_{a\alpha}$)，$e_{v\alpha}$ 和 $e_{i\alpha}$ 代替($e_{a\alpha}$，$e_{v\alpha}$ 和 $e_{i\alpha}$ 统称**有源边**)，则它们与 G 构成的

* 1987 年 7 月 7 日收到，1987 年 10 月 27 日修改定稿.

图分别称为 N^* 的**混合图** G_c，**电压图** G_v 和**电流图** G_i，$G_r = G_v \cup G_i$ 称为 N^* 的**合成图**[1]。若把 G 用 E 代替，把每个 $e_{a\alpha}$，$e_{v\alpha}$ 和 $e_{i\alpha}$ 各看成一条二端超边，则图 G_c，G_v，G_i 和 G_r 都变(简化)成超图[2]，分别称为 N^* 的**混合超图** H_c，**电压超图** H_v，**电流超图** H_i 和**合成超图** H_r。H_i 的一个超树[2] T_{ci} 称为**完全电流超树**，如果从 T_{ci} 中移去所有 i 边并加上对应 v 边所得超图是 H_v 的一个超树，后者称为 T_{ci} 的对应**完全电压超树** T_{cv}，它们形成一个**完全超树对**，简称**完全超树**，$T_c = \{T_{ci}, T_{cv}\}$。

其次，求 **VCCS 集** $S_{vc} = \{1, 2, \cdots, s\}$ 的所有子集的集合

$$SS = \{SS(b) \mid b = 0, 1, 2, \cdots, 2^s - 1\},$$
$$SS(b) \triangleq S = \{S(j) \mid S(j) \in S_{vc}, j = 1, \cdots, a\} \tag{1}$$

设 $SS(b)$ 的顶点集为 $V(|V| = m)$，v 边集为 U_v，i 边集为 $U_i(|U_v| = |U_i| = a)$，则 $g_{vb} = (V, U_v)$ 和 $g_{ib} = (V, U_i)$ 各称为 $SS(b)$ 的 ***v* 边图**和 ***i* 边图**，它们的关联矩阵和邻接矩阵各记作 AV，AI 和 JV，JI。对应于满足条件 "g_{vb} 和 g_{ib} 不含回路"的每个源子集 $SS(b)$，求出超边 E 的所有满足下列条件的 $k = a + 1$ 分解[2]

$$T_{c(a+1)}(b, d) = E(F_1, \cdots, F_k) = \{F_p \mid p = 1, \cdots, k\}, d = 1, 2, 3, \cdots, n_d \tag{2a}$$

(存放在两数组 C，D 中，子超边 $F_p = \{D(C(p)), \cdots, D(C(p+1)-1)\}$)，(求法见第三节)；每个 F_p 的伴随子图 $G(F_p)$ 是连通的；

$$T_{ci}(b, d) \triangleq \prod_{j=1}^{a} e_{iS(j)} T_{c(a+1)}(b, d) \text{ 与 } T_{cv}(b, d) \triangleq \prod_{j=1}^{a} e_{vS(j)} T_{c(a+1)}(b, d) \tag{2b}$$

组成完全超树对；称 F_p 为 E 的**连通子超边**。定义 $T_{cv}(b, d)$ 的 $k \times a$ 阶**子超边-*v* 边关联阵** MV 的元素如下：

$$MV(p, j) = \begin{cases} 1, & \text{若 } F_p \text{ 含 } e_{vS(j)} \text{ 的始点} \\ -1, & \text{若 } F_p \text{ 含 } e_{vS(j)} \text{ 的终点} \\ 0, & \text{若 } F_p \text{ 不含 } e_{vS(j)} \text{ 的端点} \end{cases} \tag{3}$$

同样可定义 $T_{ci}(b, d)$ 的**子超边-*i* 边关联阵** MI。令 $M(p,)$ 表示矩阵 M 的第 p 行，显然有

$$MV(p,) = \sum_{i=c(p)}^{c(p+1)-1} AV(D(i),), \quad MI(p,) = \sum_{i=c(p)}^{c(p+1)-1} AI(D(i),), \quad p = 1, \cdots, k \tag{4}$$

根据 MV 和 MI 应用深度优先搜索法 (DFS) 可以判别 T_{cv} 和 T_{ci} 是否含有超回路，即是否为超树。

对于任一超树 T，定义 T 的每一超边关于某参考点 (根) r 的主点和副点如下：赋于 T 的各超边端点以这样的极性，使得 T 变成负根超树[2]，则每条超边的正极和负极分别称为该超边的**主点**和**副点**。T_{ci} 和 T_{cv} 中 i 边 $e_{i\zeta}$ 和 v 边 $e_{v\zeta}$ 的主点和副点分别称为 ***i* 主点** $\triangle_\zeta$，***i* 副点** $\triangle_\zeta$，***v* 主点** $\square_\zeta$ 和 ***v* 副点** $\boxdot_\zeta$，有向边 $e_\zeta = \triangle_\zeta \rightarrow \square_\zeta$ 称为 VCCSζ 的**主边**。定义 $T_{cv}(b, d)$ 的 $k \times a$ 阶**子超边-*v* 主副点关联阵** NV 的元素如下：

$$NV(p, j) = \begin{cases} 1, & \text{若 } F_p \text{ 含 } e_{vs(j)} \text{ 的主点} \\ -1, & \text{若 } F_p \text{ 含 } e_{vs(j)} \text{ 的副点} \\ 0, & \text{若 } F_p \text{ 不含 } e_{vs(j)} \text{ 的端点} \end{cases} \tag{5}$$

同样可定义 $T_{ei}(b,d)$ 的**子超边-i 主副点关联阵** NI. 与文献[4]定理 1 的证明相类似，可证下面定理成立.

定理 1（$T_{e(a+1)}$ 的性质）　超边 E 的对应于源子集 $SS(b)=\{S(j)|j=1,\cdots,a\}$ 的每个满足条件(2)式的 $k=a+1$ 分解 $T_{e(a+1)}(b,d)$ 具有下列性质:

(1) 对于所有 $\zeta\in SS(b)$, $\square_\zeta$ 和 $\boxdot_\zeta$ 不属于同一连通片，$\triangle_\zeta$ 和 $\triangleq_\zeta$ 也不属于同一连通片.

(2) 每个连通片至少含一个 v 点 $\square_\zeta$ 或 $\boxdot_\zeta$ 和一个 i 点 $\triangle_\eta$ 或 $\triangleq_\eta$, $\zeta,\eta\in SS(b)$.

(3) **根片** T_{sr}（含根 r 的连通片）不含任何 v, i 主点，但至少含一个 $\boxdot_\zeta$ 和一个 $\triangle_\eta$, ζ, $\eta\in SS(b)$.

(4) 每个**非根片** $T_{s\zeta}$（不含根 r 的连通片）恰含一条超通路[2] $\square_\zeta$---$\triangle_\eta$ $(\zeta,\eta\in SS(b))$，称为**主超通路**.

(5) 若把 a 条主边 $e_{S(j)}$, $j=1,\cdots,a$, 加到 $T_{e(a+1)}$ 上，它们将与上述 a 条主超通路形成 $c(\leqslant a)$ 条单向超回路 C_μ, $\mu=1,\cdots,c$, 称为**主超回路**，并使 $T_{e(a+1)}$ 变成一个 $c+1$ 片超图，称为 H_e 的对应于 $T_e(b,d)$ 的**主子超图**，记作

$$h_{pr}(b,d)=\prod_{j=1}^{a} e_{S(j)}T_{e(a+1)}(b,d),\quad e_{S(j)}=\triangle_{S(j)}\to\square_{S(j)}, \tag{6}$$

对应的**带号主子超图权** $\tilde{h}_{pr}(b,d,y)$, 简记作 $HW(b,d)$, 定义为

$$HW(b,d)=\tilde{h}_{pr}(b,d,y)\triangleq(-1)^c\prod_{j=1}^{a} e_{S(j)}(y)T_{e(a+1)}(b,d,y)$$

$$=\mathrm{SIGN}\prod_{j=1}^{a} g_{s(j)}EW(b,d) \tag{7}$$

式中 c 为 $h_{pr}(b,d)$ 中的主超回路数（根据 NV 和 NI 应用 DFS 可以搜索到所有主超回路，从而求得 c 值），而

$$e_{s(j)}(y)=(-1)g_{s(j)}MI(x,j)MV(y,j),\quad \triangle_{S(j)}\in F_x,\ \square_{S(j)}\in F_y \tag{7a}$$

$$\mathrm{SIGN}=(-1)^{c+a}\prod_{j=1}^{a} MI(x,j)MV(y,j) \tag{7b}$$

$$T_{e(a+1)}(b,d,y)=EW(b,d)\triangleq P_G(t_{F_1,\cdots,F_k}(y))=\sum_G t_{F_1,\cdots,F_k}(y) \tag{7c}$$

G 的 k 树权多项式 $P_G(t_{F_1,\cdots,F_k}(y))$ 可用文献[5]的算法求得紧凑的表达式.

推论 1（主副点简易判别法）　从定理 1 可推得 v 主副点的简易判别法（从 MV 构造 NV 的方法）如下：先选定一个子超边（例如 F_1）为根片 F_r，找出 F_r 的所有 v 点，标作 v 副点（若 $MV(r,y)\neq 0$，则 $NV(r,y)\leftarrow -1$），找出 v 边 $e_{vS(y)}$ 的另一端点（设它属于 F_x）标作 v 主点（若 $MV(x,y)*(x-r)\neq 0$，则 $NV(x,y)\leftarrow 1$），找出 F_x 中的其他 v 点都标作 v 副点，…，直到标出所有 v 主副点为止. 把上述方法中的 v 换成 i 即得 i 主副点简易判别法（从 MI 构造 NI 的方法）.

推论 2（主子超图法）　给定网络 N'，作出其增广网络 N^*, N^* 的合成图 G_c 和无源子图 G 及其伴随超边 E，构造 VCCS 子集集 $SS=\{SS(b)|b=0,1,\cdots,n_b\}$，对于满足条件“$g_{vb}$ 和 g_{ib} 都不含回路”的每个源子集 $SS(b)$ 求出超边 E 的满足条件(2a)和(2b)

的所有 $k=a+1$ 分解 $T_{c(a+1)}(b,d)$, $d=1,\cdots,n_d$, 应用推论 1 判别 v, i 主副点,作出主子超图 $h_{pr}(b,d)$, 用(7)式算得带号主子超图权 $HW(b,d)$, 最后通过下式求出网络 N' 的开路转移阻抗

$$Z_{i_1i_2\to j_1j_2}=\frac{Y_{(i_1j_1,i_2j_2)}}{Y_{(j_2i_2)}}=\frac{g_s^{-1}\sum_{b=n_b'+1}^{n_b}\sum_{d=1}^{n_d}HW(b,d)}{\sum_{b=0}^{n_b'}\sum_{d=1}^{n_d}HW(b,d)} \tag{8}$$

式中 $n_b=2^s-1$, $n_b'=2^{(s-1)}-1$. 类似地可得其他符号网络函数.

为了证明(8)式,先研究**带号完全树权** $\tilde{t}_c(y)$ 与**带号主子图权** $\tilde{g}_{pr}(y)$ 的关系. 根据文献[1,6]有

$$\tilde{g}_{pr}(y)\triangleq(-1)^c g_{pr}(y)=(-1)^c\prod_{j=1}^{a}e_{s(j)}(y)t_{c(a+1)}(y)=\mathrm{SIGN}\prod_{j=1}^{a}g_{s(j)}t_{F_1,\cdots,F_k}(y), \tag{9}$$

$$\tilde{t}_c(y)\triangleq V(SP)t_c(y)=\det A(t_{cip})\det A(t_{cvp})t_c(y),\quad t_c(y)\triangleq\prod_{j=1}^{a}g_{s(j)}t_{c(a+1)}(y), \tag{10}$$

式中 t_{cip} 和 t_{cvp} 各是 t_{ci} 和 t_{cv} 关于根 r 的伴随主树(星形树), SP 是 t_{ci} 与 t_{ct} 的**符号置换**,来源于 $A^r(t_{cip})$ 和 $A^r(t_{cvp})$. 类似地,基于 $A(t_{cip})$ 和 $A(t_{cvp})$ 可定义**主点置换** VP 如下:

(1) 行 j 代表图 G_j, $j=1,2$, $G_1=t_{cip}$, $G_2=t_{cvp}$;

(2) 列 k 代表边 y_k, $k=1,\cdots,n'$, $n'=n-1$ 为树支数;

(3) 元素(j,k), $j=1,2$, $k=1,\cdots,n'$, 定义为

$$(j,k)=\begin{cases}x, & \text{若 } x \text{ 是 } y_k \text{ 的主点,且在 } G_j \text{ 中 } y_k \text{ 方向背离 } x;\\ x^-, & \text{若 } x \text{ 是 } y_k \text{ 的主点,且在 } G_j \text{ 中 } y_k \text{ 方向指向 } x.\end{cases}$$

VP 的值 $V(VP)\triangleq(-1)^{m+d}$, m 是 VP 中负上标个数, d 是 VP 的判数,即把第二行变成第一行所需对换数. 根据文献[7],VP 可分解成一些独立循环置换 Q_μ, $\mu=1,\cdots,c$ (每个 Q_μ 对应于 g_{pr} 的一条主回路[11]) 与一些不变置换 Q_v, $v=c+1,\cdots,n_s$, 的乘积, $VP=Q_1\cdots Q_cQ_{c+1}\cdots Q_{n_s}$, 而且 $d=n'-n_S=n''-c$, n'' 是 $Q_1,\cdots,Q_c$ 的列数和. 设 m_k 是列 k 的负上标数,则

$$V(VP)=(-1)^{m+n''-c}=(-1)^c\prod_{k=1}^{n''}(-1)^{m_k+1}\prod_{k=n''+1}^{n'}(-1)^{m_k} \tag{11}$$

因 Q_v 中主点不变, $m_k=0$ 或 2,故 $(-1)^{m_k}=1$. 对于 Q_μ 中每条无源边 y_k, 因其主点从一端变到另一端, $m_k=1$, 故 $(-1)^{m_k+1}=1$; 对于 Q_μ 中每条有源边 y_h, $m_h=0$, 1 或 2, 可证 $(-1)^{m_h+1}=\mathrm{sign}(e_{s(h)}(y))$. 考虑到 $V(SP)=\det A^r(t_{cip})\det A^r(t_{cvp})=\det A(t_{cip})\det A(t_{cvp})=V(VP)$ 和式(7a), (7b)可推得

定理 2

$$V(SP)=V(VP)=(-1)^c\prod_{h=1}^{a}\mathrm{sign}(e_{s(h)}(y))=\mathrm{SIGN}, \tag{12}$$

推论 2

$$\tilde{t}_c(y) = V(VP)t_c(y) = (-1)^c g_{pr}(y) = \tilde{g}_{pr}(y). \tag{13}$$

根据(7)式和(9)式可证明下面定理成立.

定理 3　混合超图 H_c 的带号主子超图权 $HW(b, d)$ 与其伴随混合图 G_c 的带号主子图权有下列关系

$$HW(b, d) = \tilde{h}_{pr}(b, d, y) = \Sigma\tilde{g}_{pr}(b, d, y)[g_{pr}(b, d) \subset G_c(h_{pr}(b, d))] \tag{14}$$

式中求和是对 $h_{pr}(b, d)$ 的所有伴随混合图 $G_c(h_{pr}(b, d))$ 中的所有主子图 $g_{pr}(b,d)$ 进行的.

据(13)、(14)式和文献[6]的(8-5-7)式就可推得(8)式.

三、主子超图法的算法

主子超图法的核心是求 $HW(b, d)$. 设 $SS(b)$ 的顶点集 $V=\{V(c)|c=1,\cdots,m\}$, 由 G 的无源边构成的通路称为**无源通路**. 定义 m 阶**无源邻接矩阵** JP 如下: 若 $V(g)$ 与 $V(h)$ 之间存在无源通路而且不存在 $SS(b)$ 中的 v 边和 i 边, 则 $JP(g, h) = JP(h, g) = 1$; 否则, $JP(g, h) = JP(h, g) = 0$. 根据 JP 应用分支-定界搜索法可产生 m 端超边 E 的**超边分解状态空间树**[8] $T(b)$. 若 $T(b)$ 的某节点对应于连通子超边 $F_1, \cdots, F_{p-1}$ 已经产生了, F_p 正在产生中, F_p 的邻点数为 h, 则该节点将有 $h+1$ 个分支: 使 F_p 的 h 个邻点分别熔进 F_p 可形成 h 个分支, 剩下一个分支是 F_p 不熔进任何邻点. 因此根据定理 1 和 LIFO 分支定界搜索法[8]可得**求 $HW(b, d)$ 的主要算法步骤**如下:

(1) $d \leftarrow 0, k \leftarrow a+1, l \leftarrow 0, p \leftarrow 1, q \leftarrow 2, C \leftarrow \phi, C(1) \leftarrow 1, D \leftarrow \phi, D(1) \leftarrow 1, HP \leftarrow JP(1,)$.

(2) w 从 1 到 m 循环进行: 若 $HP(w) = 1$, $q < m$ 或 $q = m$ 但 $p = k$, 则 $D(q) \leftarrow w$, 调用子程序 HYPCIR (根据 MV, MI, 应用 DFS 检查超图 $g_{vb} \cup F_1 \cup \cdots \cup F_p$ 和 $g_{ib} \cup F_1 \cup \cdots \cup F_p$ 是否含超回路, 若都不含, 则 $hc \leftarrow 0$, 否则 $hc \leftarrow 1$). 若 $hc = 0$, 且 $q < m$, 则入栈, 即 $l \leftarrow l+1$, $SC(l,) \leftarrow C$, $SD(l,) \leftarrow D$; 若 $q = m$, $p = k$, 则 E 的一个满足条件(2a)和(2b)的 k 分解 $T_{c(a+1)}(b, d)$ 已经找到, $d \leftarrow d+1$, $C(k+1) \leftarrow m+1$, 调用子程序 PSHGW (根据 MV, MI, NV, NI 和 A, C, D, 应用分支-定界搜索法)求出并打印 $HW(b, d)$.

(3) 若 F_p 至少含一个 v 点和一个 i 点, 则 $z \leftarrow 1$. 若还有 $p < k$, 则产生 F_{p+1}, 即 $p \leftarrow p+1$, $C(p) \leftarrow q$, $D(q) \leftarrow 0$; 若 $q < m$, 或 $q = m$ 但 $p = k$, 则取一个未访问过的顶点 w, $D(q) \leftarrow w$, 调用 HYPCIR, 若 $hc = 0$, 且 $q = m, p = k$, 则 $d \leftarrow d+1$, $C(k+1) \leftarrow m+1$, 调用 PSHGW, 求出并打印 $HW(b, d)$; 若 $q < m$, 则转步骤(5).

(4) 若 $z = 0$ 或 $hc = 1$ 或 $q = m$, 那么若 $l = 0$ (栈空), 则 $n_d \leftarrow d$, 打印 $HW(b) = HW(b, 1) + \cdots + HW(b, n_d)$, 结束; 若 $l > 0$, 则出栈, 即 $C \leftarrow SC(l,)$, $D \leftarrow SD(l,)$, $l \leftarrow l-1$, C 中非零元素数送 p, D 中非零元素数送 q.

(5) $HP \leftarrow JP(D(C(p)),) \vee \cdots \vee JP(D(q),)$, $q \leftarrow q+1$, 转步骤(2).

根据上述原理和算法可设计出用 FORTRAN 语言写成的产生一个网络 N^* 的混合

超图 H_c 的全部带号主子超图权 $HW(b,d)$ 的具体程序——**程序 GSPSHW**，已在微型机上调试通过。它的**计算时间复杂度**为 $O(ve\sum n_l)$ [主要决定于计算 $EW(b,d)$ 的时间]，**空间复杂度**为 $O[(\delta+3)n'_{dl}+(n+s)(n+3e+4s)]$，这里 v 是增广网络 N^* 的源点数，e 是 N^* 的无源边数，$\sum n_l$ 是 N^* 的混合超图 H_c 的所有主子超图的 $EW(b,d)$ 所对应的状态空间树[5]的叶总数，n 是 N^* 的顶点数，s 是 N^* 的 VCCS 数，δ 是 N^* 的无源子图 G 的平均顶点度数，n'_{dl} 是 G 的树多项式所对应的状态空间树的节点数。由于 $\sum n_l \ll n_p$ (n_p 是 N^* 的混合图 G_c 的主子图总数)，而且表达式很紧凑，所以它优于主子图法(尤其是当网络规模较大而源数较少时)。另外，主子超图法还具有直观、简单、便于手算和教学的优点(参看下例)。

四、应用举例

例 用主子超图法求图 1(a) 所示电网络 N' 的开路转移阻抗 $Z_{16\to 26}$。

解 N' 的增广网络 N^* 的合成图 G_r 如图 1(b) 所示，有 $n=6$，$e=9$，$s=3$，$v=5$，

$$VE=\begin{bmatrix}1&4&2\\3&6&6\end{bmatrix},\quad IE=\begin{bmatrix}4&2&1\\3&6&6\end{bmatrix},\quad PE=\begin{bmatrix}1&1&3&3&4&2&2&5&5\\6&3&6&4&6&6&5&6&3\end{bmatrix}$$

$$J=\begin{bmatrix}0&0&1&0&0&1\\0&0&0&0&1&1\\1&0&0&1&1&1\\0&0&1&0&0&1\\0&1&1&0&0&1\\1&1&1&1&1&0\end{bmatrix},\quad A=\begin{bmatrix}1&1&0&0&0&0&0&0&0\\0&0&0&0&0&1&1&0&0\\0&1&1&1&0&0&0&0&1\\0&0&0&1&1&0&0&0&0\\0&0&0&0&0&0&1&1&1\\1&0&1&0&1&1&0&1&0\end{bmatrix}$$

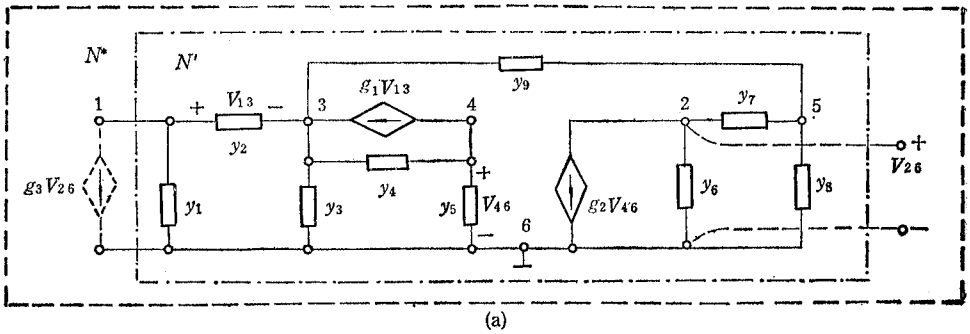
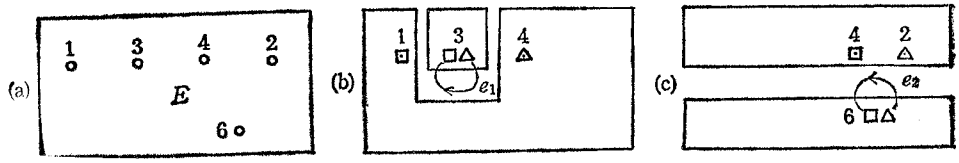


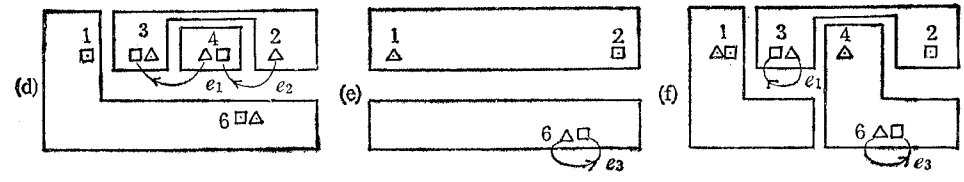
图 1 有源网络 N' 和它的增广网络 N^*(a) 和合成图 G_r(b)

$CS=(1,2,3,5,6,8,10,13)$, $DS=(1;2;1,2;3;1,3;2,3;1,2,3)$, $n_b=7$, $n_S=12$。应用主子超图法可得 N^* 的混合超图 H_c 的所有主子超图如图 2 所示。对应的带号主子超图权如下：

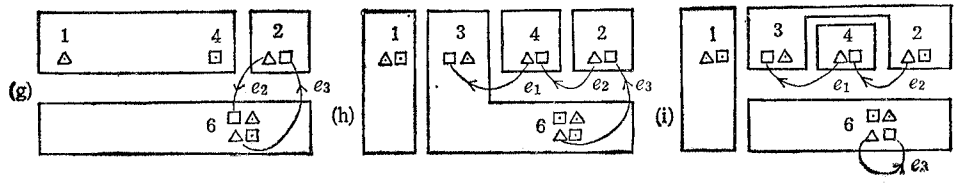
$$HW(0)=\sum_G t(y)=y_1\{(y_2+y_3)(y_4+y_5)[y_6(y_7+y_8+y_9)+(y_8+y_9)y_7]$$
$$+y_5[y_4(y_6(y_7+y_8+y_9)+(y_8+y_9)y_7)+y_6(y_7+y_8)y_9+y_8y_7y_9]$$
$$+y_6(y_7+y_8)y_9y_4+y_8y_7y_9y_4\}+y_2\{y_3(y_4+y_5)[y_6(y_7+y_8+y_9)$$
$$+(y_8+y_9)y_7]+y_4[y_5(y_6(y_7+y_8+y_9)+(y_8+y_9)y_7)$$
$$+y_9(y_7(y_6+y_8)+y_8y_6)]+y_9[y_7(y_6+y_8)y_5+y_8y_5y_6]\}$$



(a) $h_{pr}(0)=E(12346)$　　(b) $h_{pr}(1,1)=e_1E(14,3)$, $e_1(y)=-g_1$　　(c) $h_{pr}(2,1)=e_2E(42,6)$, $e_2(y)=-g_2$



(d) $h_{pr}(3,1)=e_1e_2E(16,32,4)$, $e_1(y)=g_1$, $e_2(y)=-g_2$　　(e) $h_{pr}(4,1)=e_3E(21,6)$, $e_3(y)=-g_3$　　(f) $h_{pr}(5,1)=e_1e_3E(1,32,46)$, $e_1(y)=-g_1$, $e_3(y)=-g_3$



(g) $h_{pr}(6,1)=e_2e_3E(41,6,2)$, $e_2(y)=g_2$, $e_3(y)=g_3$　　(h) $h_{pr}(7,1)=e_1e_2e_3E(1,36,4,2)$, $e_1(y)=g_1$, $e_2(y)=-g_2$, $e_3(y)=g_3$　　(i) $h_{pr}(7,2)=e_1e_2e_3E(1,32,4,6)$, $e_1(y)=g_1$, $e_2(y)=-g_2$, $e_3(y)=-g_3$

图 2　增广网络 N^* 的混合超图 H_c 的全部主子超图

$$HW(1,1)=(-1)e_1(y)\sum_G t_{14,3}(y)=g_1y_1y_5[y_6(y_7+y_8+y_9)+(y_8+y_9)y_7]$$

$$HW(2,1)=(-1)e_2(y)\sum_G t_{42,6}(y)=g_2y_4(y_2y_5y_7+y_9y_7y_1)$$

$$HW(3,1)=(-1)e_1(y)e_2(y)\sum_G t_{16,32,4}(y)=g_1g_2y_1y_9y_7$$

$$HW(4,1)=(-1)e_3(y)\sum_G t_{21,6}(y)=g_3y_7y_9y_2(y_4+y_5)$$

$$HW(5,1)=(-1)^2e_1(y)e_3(y)\sum_G t_{1,32,46}(y)=g_1g_3y_9y_7y_5$$

$$HW(6,1)=(-1)e_2(y)e_3(y)\sum_G t_{41,6,2}(y)=-g_2g_3y_4y_2(y_7+y_8+y_9)$$

$$HW(7,1)=(-1)e_1(y)e_2(y)e_3(y)\sum_G t_{1,36,4,2}(y)=g_1g_2g_3[y_3(y_7+y_8+y_9)+y_9y_8]$$

$$HW(7,2)=(-1)^2e_1(y)e_2(y)e_3(y)\sum_G t_{1,32,4,6}(y)=g_1g_2g_3y_9y_7$$

因 $n_b=2^{(s-1)}-1=3$, $n_b=2^s-1=7$, 根据(8)式可求得

$$Z_{16\to 26}=\frac{g_3^{-1}[HW(4,1)+HW(5,1)+HW(6,1)+HW(7,1)+HW(7,2)]}{HW(0)+HW(1,1)+HW(2,1)+HW(3,1)}$$

参 考 文 献

[1] 黄汝激, 电子科学学刊, **8**(1986), 335—342.
[2] 黄汝激, 电子科学学刊, **9**(1987), 244—255.
[3] W. K. Chen, Applied Graph Theory, Amsterdam, North-Holland, 1976.
[4] 黄汝激, 电子科学学刊, **7**(1985), 254—266.
[5] 黄汝激, 电子学报, 1987 年,第 5 期,第 8—13 页.
[6] W. Mayeda, Graph Theory, John Wiley and Sons, Inc., 1972, ch. 8.
[7] 黄汝激, 北京钢铁学院学报, 1982 年,第 2 期, 第 83—89 页
[8] E. Horowize, S. Sahni, Fundamentals of Computer Algorithms, Computer Science Press, Potomac, Maryland, 1978, chs. 1,7,8.

PRIMARY SUBHYPERGRAPH METHOD FOR GENERATING SYMBOLIC NETWORK FUNCTIONS

Huang Ruji

(Beijing College of Iron and Steel Technology, Beijing)

Abstract The concepts of complete hypertree and primary subhypergraph are introduced, and a new method——primary subhypergaph method is presented for generating symbolic network functions. It is an improvement of primary subgraph mothod. Its resulting expressions are very compact, and its computinp time complexity is $O(ve\Sigma_{n_l})$. In general, the total number of leaves Σ_{n_l} is far less than the numberof primary subgraphs n_p, hence its computation efficiency is higher than that of primary subgraph method.

Key words Network; Symbolic network function; Primary subhypergraph method; Complete hypertree