DOI:10.13374/j.issn1001-053x.1996.06.014

第18卷 第6期 北京科技大学学报 Vol.18 No.6
1996年 12月 Journal of University of Science and Technology Beijing Dec. 1996

陈氏基本互补划分法的改进和发展

黄汝激

北京科技大学自动化信息工程学院,北京 100083

摘要 应用有向超图理论引入超边的分解与收缩概念,把解决二个无源网络级联问题的陈氏基本互补划分(ECP)法改进为更有效的分解 − 收缩对(DCP)法,发展为有向(正根)分解 − 收缩对(PDCP)法,用于解决二个有源网络级联问题.在此基础上,进一步发展为一般分解 − 收缩(GDC)法,用于解决多个有源网络互联时求符号网络函数的问题.

关键词 符号网络函数,有向超图理论,分解 − 收缩对法

中图分类号 O157.5

在电子线路的分析与设计中经常用到符号网络函数.产生符号网络函数的 关键是产生网络图的全部树.陈惠开教授[1]提出了2个多端无源网络级联时产生全部树的基本互补划分(ECP)法.文献[2～5]又对ECP法作了进一步研究,但无突破性进展.文献[6]阐明了 m 点集 $V(m)$ 的一个ECP对应于无向二超边超图 $H^{(m)}=(V^{(m)},E)$ 的一棵超树 T.本文首先应用有向超图理论[7]引入超边的分解与收缩概念,把陈氏ECP法改进为更有效的方法,以解决2个有源网络级联及多个有源网络互联时求符号网络函数的问题.

1 超边的分解与收缩

根据有向超图理论[7],$e\geq 2$ 个多端有源网络互联形成的超网络 N 的伴随超图 $H=(V,E)$, $V=\{1,\cdots,v\}$, $E=\{E_1,\cdots,E_e\}$.设超边 $E_j=A_j\cup B_j$, $A_j\cap B_j=\phi$,记作 $E_j=E_j[A_jB_j]$.A_j 称为被开路,若 A_j 的所有引线被移去使得 E_j 变成 $E_j[B_j]=E_j-A_j$.A_j 称为被收缩,若 A_j 及其伴随图 $G(A_j)$ 被收缩成一点 $\overline{A_j}$(称超端点)使得 E_j 变成 $|E_j-A_j|+1$ 端超边,记作 $E_j[\overline{A_j}B_j]=E_j\circ A_j$.$m$ 端无向超边 E_j 的一个 h 分解 $D_j=D_j(m,h)=E_j[F_1,\cdots,F_h]=\{F_1,\cdots,F_h\}$ 是把 E_j 划分成 h 个互不相交的非空子集 F_i, $i=1,\cdots,h$.F_i 称为 E_j 的 i 号子超边,并以其端点号串为其简化表示.D_j 的权 $D_j(y)$ 定义为 E_j 的伴随图 $G(E_j)$ 中所有 h 树权 $t(F_1,\cdots,F_h;y)$ 之和形成的多项式 $P_j[t(F_1,\cdots,F_h;y)]$,即:

$$D_j(y)\triangleq P_j\left[t(F_1,\cdots,F_h;y)\right]=\Sigma t(F_1,\cdots,F_h;y)|_{G(E_j)} \tag{1}$$

m 端无向超边的 h 分解的数目记作 $[m,h]$.无向超边 $E[1,\cdots,m]$ 的分解集 SD(m).h 分解集 SD(m,h) 和 a 号 h 分解的表达式参看文献[7]定理2.应用该定理可递归推得 SD(m), $m=1,2,3,4,\cdots$(见文献[7]中推论1).$B_m=|\mathrm{SD}(m)|$ 称为贝尔(Bell)数,$B_1=1,B_2=2,B_3=5,B_4=15,B_5=52,\cdots$.$m$ 端有向超边 E_j 的一个正根 h 分解 $\mathrm{PD}_j=\mathrm{PD}_j(m,h)_r=E_j(r_1w_1,\cdots,r_hw_h)=\{r_1w_1,\cdots,r_hw_h\}$ 是把 E_j 划分成 h 个互不相交的具有正根 r_i 的非空正根子超边 r_iw_i, $i=1,\cdots,$

1996-01-02 收稿 第一作者 男 56岁 教授

h. PD_j的权$PD_j(y)$定义为E_j的伴随有向图$G(E_j)$中所有正根h树权$t(r_1w_1,\cdots,r_hw_h;y)$之和形成的多项式$P_j[t(r_1w_1,\cdots,r_hw_h;y)]$，即：

$$PD_j(y) \triangleq P_j[t(r_1w_1,\cdots,r_hw_h;y)] = \Sigma t(r_1w_1,\cdots,r_hw_h;y)|_{G(E_j)} \tag{2}$$

m端有向超边$E(1,\cdots,m)$的具有给定正根集$p=\{p_1,\cdots,p_n\}$的正根分解集$SPD(m)_p$是它的所有具有正根集$r\supseteq p$的正根分解$PD(m,h,c)_r$的集合，即：

$$SPD(m)_p = \{PD(m,h,c)_r | r\supseteq p;\ h=|p|,\cdots,m;c=1,\cdots,[m,h]_p\} \tag{3}$$

其中$r=\{r_1,\cdots,r_h\}=\{p_1,\cdots,p_n,r_{n+1},\cdots,r_h\}$，$[m,h]_p$是$E(1,\cdots,m)$的具有给定正根集$p$的正根$h$分解$PD(m,h,c)_r$的数目．例如$SPD(3)_{2,3}=\{\{21,3\},\{2,31\},\{2,3,1\}\}$．根据$SPD(m)_p$和$SD(m)$的定义，应用似SPARKS语言[8]，可设计算法ASPD如下．

算法 ASPD // 从$SD(m)$求$SPD(m)_p$ //

1．$p\leftarrow\{p_1,\cdots,p_n\}$; $n\leftarrow|p|$; $SPD(m)_p\leftarrow\phi$; input m and $SD(m)$

2．for $h\leftarrow n$ to m do $c\leftarrow 0$; $SPD(m,h)\leftarrow\phi$

for $a\leftarrow 1$ to $[m,h]$ do from $SD(m)$ take out $D(m,h,a)$;

if $D(m,h,a)=\{p_1w_1,\cdots,p_nw_n,F_{n+1},\cdots,F_h\}$ then

for $i_{n+1}\leftarrow 1$ to $|F_{n+1}|$ do $r_{n+1}\leftarrow F_{n+1}(i_{n+1})$; $w_{n+1}\leftarrow F_{n+1}-F_{n+1}(i_{n+1})$

⋮

for $i_h\leftarrow 1$ to $|F_h|$ do $r_h\leftarrow F_h(i_h)$; $w_h\leftarrow F_h-F_h(i_h)$; $c\leftarrow c+1$;

$PD(m,h,c)\leftarrow\{p_1w_1,\cdots,p_nw_n,r_{n+1}w_{n+1},\cdots,r_hw_h\}$;

$SPD(m,h)\leftarrow SPD(m,h)\cup\{PD(m,h,c)\}$

repeat

⋮

repeat

endif

repeat; $SPD(m)_p\leftarrow SPD(m)_p\cup SPD(m,h)$

repeat

3．end ASPD

推论 1 令$SPD(m)=SPD(m)_1$，从算法ASPD可推得

$SPD(1)=\{\{1\}\}$

$SPD(2)=\{\{1\ 2\},\{1,2\}\}$

$SPD(3)=\{\{1\ 2\ 3\},\{1\ 2,3\},\{1\ 3,2\},\{1,2\ 3\},\{1,3\ 2\},\{1,2,3\}$

$SPD(4)=\{\{1\ 2\ 3\ 4\},\{1\ 2\ 3,4\},\{1\ 2\ 4,3\},\{1\ 2,3\ 4\},\{1\ 2,4\ 3\},\{1\ 3\ 4,2\},$
$\{1\ 3,2\ 4\},\{1\ 3,4\ 2\},\{1\ 4,2\ 3\},\{1\ 4,3\ 2\},\{1,2\ 3\ 4\},\{1,3\ 2\ 4\},$
$\{1,4\ 2\ 3\},\{1\ 2,3,4\},\{1\ 3,2,4\},\{1,2\ 3,4\},\{1,3\ 2,4\},\{1\ 4,2,3\},$
$\{1,2\ 4,3\},\{1,4\ 2,3\},\{1,2,3\ 4\},\{1,2,4\ 3\},\{1,2,3,4\}\}$

2 DCP 法和 PDCP 法

考虑一超网络N，它的伴随超图$H=(V,E)$，$V=\{1,2,\cdots,m\}$，$E=\{E_1,\cdots,E_e\}$，H和E_j的伴

随混合图分别记作 $G = G(H)$和 $G_j = G(E_j), j = 1, \cdots, e$.要求 H 的超树权多项式 $P[T(y)]$[7] . 下面先讨论 2 种特殊情况.

情况 1　H 是无向二超边超图— DCP 法

若 H 的一对分解超边 $D_1 = E_1[F_1, \cdots, F_h]$和 $D_2 = E_2[F_1', \cdots, F_k']$ $(h + k = m + 1)$构成一棵超树[6] T，则称 D_1和 D_2为 V 的一对基本互补划分，记作 ECP $= (D_1, D_2)$. H 的一个分解超边 $D_1 = E_1[F_1, \cdots, F_h]$与其对应收缩超边 $C_2 = E_2[\overline{F_1}, \cdots, \overline{F_h}]$组成的对称为 H 的一个分解 − 收缩对，记作 DCP $= (D_1, C_2)$，它的权 DCP$(y) \triangleq D_1(y)C_2(y)$.

定理 1　无向二超边超图 H 的超树权多项式 $P[T(y)]$等于 H 的所有分解 − 收缩对权 DCP(y)之和，即:

$$P[T(y)] = \Sigma \mathrm{DCP}(y) = \Sigma D_1(y)C_2(y) \ (D_1 \in \mathrm{SD}_1) \tag{4}$$

式中，求和是对于 E_1 的所有分解超边 $D_1 \in \mathrm{SD}_1$进行的.

证: 设与 D_1 基本互补的分解超边 D_2 有 b 个，记作 $D_2^i, i = 1, \cdots, b$，则含 D_1 的部分超树权多项式 $P[T(y)|D_1] = D_1(y) \sum_{i=1}^{b} D_2^i(y)$. 因 F_j 内的点已被 D_1 连通，F_j 与 $F_k (j \neq k)$未被连通，故据超树定义，在 D_2^i 中，F_j 内的点应不连通，F_j 与 $F_k (j \neq k)$的点或者由 D_2^i 连通，或者通过 D_1 连通. 因此把 $F_j (j = 1, \cdots, h)$收缩后所有 $D_2^i (i = 1, \cdots, b)$都将变成同一个收缩超边 C_2，从而有:

$$\sum_{i=1}^{b} D_2^i(y) = C_2(y) \tag{5}$$

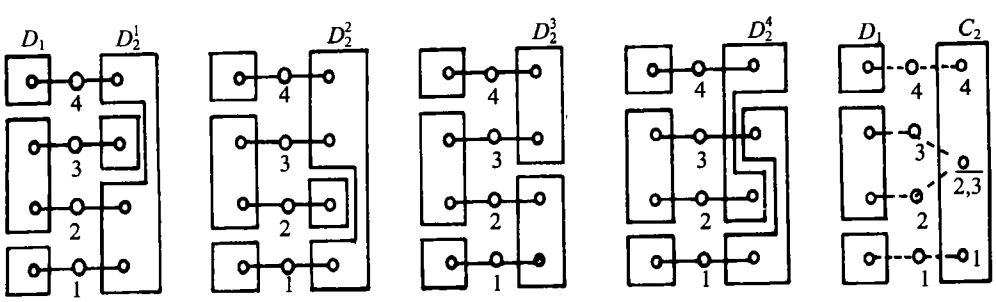
$$P[T(y)] = \Sigma P[T(y)|D_1] = \Sigma D_1(y) \sum_{i=1}^{b} D_2^i(y) \qquad (\mathrm{ECP}\ 法) \tag{6-1}$$

$$P[T(y)] = \Sigma D_1(y)C_2(y) = \Sigma DCP(y) \qquad (\mathrm{DCP}\ 法) \tag{6-2}$$

例如，$m = 4$ 和 $D_1 = E_1[1, 23, 4]$的无向超树集和 DCP，见图 1 . 这里 $\sum_{i=1}^{b} D_2^i(y) = \sum_{i=1}^{4} D_2^i(y) = C_2(y)$. DCP 法与 ECP 法的 $P[T(y)]$项数 B_m 与 $N_m = 2(m+1)^{m-2}$的比较如表 1 所示. 可见，随着 m 的增大，B_m 可比 N_m降低几个数量级以上，因此 DCP 法的计算效率远高于 ECP 法.

情况 2　H 是有向二超边超图—— PDCP 法

陈惠开教授[1]未解决有源网络的分解法问题. 若 N 是有源网络，H 是有向超图，必须将陈氏 ECP 概念发展如下. 若 H 的一对正根分解超边 $PD_1 = E_1(r_1w_1, \cdots, r_hw_h)$和 $PD_2 = E_2(r_1'w_1', \cdots, r_k'w_k')$ $(h + k = m + 1,\ r_1 = r_1' = 1)$构成一棵正根超树[6,7] T_1(这里选点 1 为正根)，则称



(a) $D_2^1 = E_2[124,3]$　(b) $D_2^2 = E_2[134,2]$　(c) $D_2^3 = E_2[12,34]$　(d) $D_2^4 = E_2[13,24]$　(e) $C_2 = E_2[1\overline{234}]$

图 1 $m = 4$ **和** $D_1 = E_1[1,23,4]$**的无向超树集** $\{T' = D_1 \cup D_2^i | i = 1, \cdots, 4\}$ **和** **DCP** $= (D_1, C_2)$

PD_1和PD_2为V的的一对有向(正根)基本互补划分,记作 PECP = (PD_1, PD_2).H的一个正根分解超边 $PD_1 = E_1(r_1w_1, \cdots, r_hw_h)$(设 $r_1 = 1$)与其对应正根收缩超边 $PC_2 = E_2(\overline{r_1w_1^+}, \cdots, \overline{r_hw_h^+})$组成的对称为$H$的一个正根分解 – 收缩对,记作 PDCP = (PD_1, PC_2),它的权 $PDCP(y) \triangleq PD_1(y)PC_2(y)$,式中 w_j^+ ($j = 1, \cdots, h$)的右上标"+"表示点集 w_j中所有点在PD_2中应为正极(因它们在PD_1中为负极),收缩前应将 $G(E_2)$的 w_j中所有点的射入边都去掉,以保证 w_j中所有点在PD_2中为正极.

表1 DCP法与ECP法的 $P[T(y)]$ 项数B_m与N_m的比较

m	DCP 法项数B_m	ECP 法项数$N_m=2(m+1)^{m-2}$
2	2	2
3	5	8
4	15	50
5	52	432
6	203	4 802
7	877	65 536
8	4 140	1 062 882
9	21 147	2×10^7

定理 2 有向二超边超图 H 的正根超树权多项式 $P[T_1(y)]$等于 H 的所有正根分解 – 收缩对权 PDCP(y)之和,即:

$$P[T_1(y)] = \Sigma\, PDCP(y) = \Sigma\, PD_1(y)PC_2(y)\ (PD_1 \in SPD_1) \tag{7}$$

式中求和是对于 E_1 的所有正根分解超边 $PD_1 \in SPD_1$ 进行的.

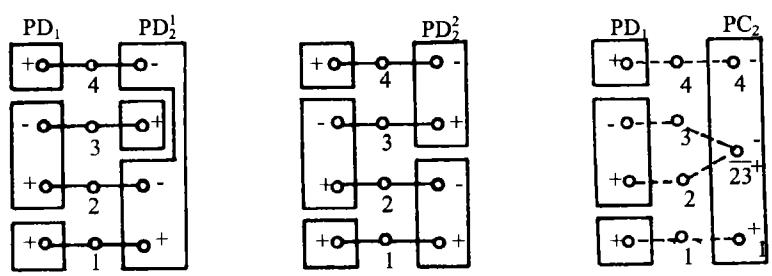
证: 与定理 1 的类似,设与 PD_1正根基本互补的分解超边 PD_2有 a 个 ($a \le b$),记作 PD_2^i, $i = 1, \cdots, a$,则有:

$$\sum_{i=1}^{a} PD_2^i(y) = PC_2(y) \tag{8}$$

$$P(T_1(y)) = \Sigma\, P(T_1(y)|PD_1) = \Sigma\, PD_1(y)\sum_{i=1}^{a} PD_2^i(y) \quad (\text{PECP 法}) \tag{9-1}$$

$$P(T_1(y) = \Sigma\, PD_1(y)PC_2(y) = \Sigma\, PDCP(y) \quad (\text{PDCP 法}) \tag{9-2}$$

文献 [6]中第三节的方法就是 PECP 法,这里把它改进为 PDCP 法,大大简化了计算和提高了计算效率. 例如,$m = 4$ 和 $PD_1 = E_1(1, 23, 4)$的正根超树集和 PDCP 如图 2 所示.这里 $\sum_{i=1}^{a} PD_2^i(y) = PD_2^1(y) + PD_2^2(y) = PC_2(y)$.



(a) $PD_2^1 = E_2(124,3)$ (b)$PD_2^2 = E_2(12,34)$ (c)$PC_2 = E_2(1\overline{23^+}4)$

图 2 $m = 4$ 和$PD_1 = E_1(1,23,4)$的正根超树集$\{T_1' = PD_1 \cup PD_2^i \mid i = 1,2\}$和PDCP = (PD_1, PC_2)

3 一般分解 – 收缩 (GDC)法

现在把 PDCP 法推广到超边数 $e > 2$ 的一般有向超图情况.这时每处理(收缩 – 分解)完

一条超边,都要对其后所有超边进行收缩处理.因此有:

定理 3　有向超图 $H(e>2)$的正根超树权多项式 $P[T_1(y)]$可表达成:

$$P(T_1(y)) = \Sigma\, \mathrm{PD}_1(y)\cdots \Sigma\, \mathrm{PC}_j D_j(y)\cdots \mathrm{PC}_e(y) \tag{10}$$

式中,$\mathrm{PC}_j D_j(1<j<e)$称为 E_j的 $(j-1)$重收缩 − 分解超边,它是 E_j依据前 $(j-1)$条处理过的超边进行 $(j-1)$重收缩后再分解(若能分解且不破坏所得超图的连通性)得到的.PC_e称为 E_e的 $(e-1)$重收缩超边,它是 E_e依据前 $(e-1)$条处理过的超边进行 $(e-1)$重收缩得到的.求和是对所有不破坏所得超图连通性的可能分解 PD_1和 $\mathrm{PC}_j D_j$, $1<j<e$ 进行的.若 $E_1(E_j$或 $E_e)$是无向超边,则 $\mathrm{PD}_1(\mathrm{PC}_j D_j$或 $\mathrm{PC}_e)$可用 $D_1(C_j D_j$或 $C_e)$代替.

根据定理 3,应用分支定界法和状态空间树 (SST)[8]概念,可设计一个通过 $P_H(T_1)=P(T_1)$ 产生 $P_G(t_1)$的一般分解 − 收缩 (GDC)算法如下:

Algorithm SSTGDCM // Find $P_G(t_1)$ by SST and $P_H(T_1)$ of $H=(V,E)$ //

Integer m, n, t, p, fl, ns, v, e

Array EM, EN, DM, E_j, PD_j, NS, FL, Y, r_i, w_i, SN, SE, SDM

1. Initialize: $v \leftarrow |V|$; $e \leftarrow |E|$; $m \leftarrow 0$; $n \leftarrow 0$; $t \leftarrow 0$; $r_1 \leftarrow \{1\}$; $p \leftarrow 1$; NS $\leftarrow Z$; FL $\leftarrow \phi$; $Y \leftarrow \phi$; $Y(0) \leftarrow 1$; DM $\leftarrow \phi$; EN $\leftarrow \phi$; EM $\leftarrow E$. If EM=ϕ then go to step 6.

2. Process state node m: From EM find a hyperedge E_j incident on the positive root r_1, if such E_j can not be found then go to step 6 else find SPD_j of E_j by Corollary 1; DM $\leftarrow \mathrm{SPD}_j$; EM $\leftarrow$ EM $-\{E_j\}$; EN $\leftarrow$ EM.

3. Generate state node n: From DM take out a $\mathrm{PD}_j = E_j(r_1 w_1, \cdots, r_h w_h)$. For each $E_k \in$ EN and each $v_i \in E_k$, if $v_i \in w_1 \cup \cdots \cup w_h$ then v_i must be a positive pole of E_k for generating T_1, denote it by v_i^+ to indicate that in $G(E_k)$ all edges incident into v_i^+ must be deleted, then contract each subhyperedge $r_i w_i$ of PD_j into one hyperterminal $\overline{r_i w_i}$, stored in set r_i, to yield a contracted hypergraph H_n expressed by EN and corresponding to a new state node n with weight PD_j and father node label m, that is, $n \leftarrow n+1$; $\mathrm{NS}(m) \leftarrow \mathrm{NS}(m)+1$; $\mathrm{FL}(n) \leftarrow m$; $Y(n) \leftarrow \mathrm{PD}_j$; DM $\leftarrow$ DM $-\{\mathrm{PD}_j\}$; $r_j \leftarrow r_i \cup w_i$, $i=1,\cdots,h$; $p \leftarrow |r_1|$.

4. If DM $\neq \phi$ then $t \leftarrow t+1$; $\mathrm{SN}(t) \leftarrow m$; SE(t) $\leftarrow$ EM; $\mathrm{SDM}(t) \leftarrow$ DM.

5. If EN $\neq \phi$ then $m \leftarrow n$; EM $\leftarrow$ EN; go to step 2.

6. If $p=v$ then state node n is a leaf node, go to step 9, else go to step 7.

7. Delete state node n: fl $\leftarrow \mathrm{FL}(n)$; $\mathrm{NS(fl)} \leftarrow \mathrm{NS(fl)}-1$; $Y(N) \leftarrow \phi$; $n \leftarrow n-1$.

8. If NS(fl)=0 and fl $\neq \mathrm{SN}(t)$ then go to step 7.

9. If $t>0$ then $m \leftarrow \mathrm{SN}(t)$; EM $\leftarrow \mathrm{SE}(t)$; DM $\leftarrow \mathrm{SDM}(t)$; $t \leftarrow t-1$; go to step 3.

10. If $t=0$ then the search process forms a SST. From SST the required $P_H(T_1)$ and $P_G(t_1)=P[T_1(y)]$ can be found by the following Theorem 4 and above Theorem 3 respectively.

11. End SSTGDCM.

定理 4　从算法 SSTGDCM 产生的状态空间树 SST 按照节点标号 n 的顺序写下它的所有状态节点权 $Y(n)$,并插入某些符号"+","["和"]"使得每个子树权等于该子树根点权乘以其所有子子树权之和,结果形成一多项式 P.那么 $P_H(T_1)=P$.P 的展开式的每一项 P_t是 H 的

一个正根超树子集，并且对应于 SST 中从根点 0 到一叶点的一条路径（称为树路）.

证：注意到伴随每一树路 p 的项 P_i 对应于一个分解超图子集 $\{H_d\}$.例如，设树路 $p(0 \to 6)$ 对应于项 $P_2 = E_1(12,3)E_2(\overline{12}3)E_3(\overline{23}) = E_1(12,3)[E_2(1,23) + E_2(13,2)]E_3(2,3)$，它包含 2 个 H_d: $E_1(12,3)E_2(1,23)E_3(2,3)$ 和 $E_1(12,3)E_2(13,2)E_3(2,3)$，见图 3 .步骤 3，5，6，7 和 8 保证每个 H_d 是 H 的一个 T_1 .对于关联根点 r_1 的每条超边 $E_j \in$ EM 只有 |DM| 类 T_1，每类 T_1 含 E_j 的一个分解 PD_j .所以步骤 1，2，3，4 和 9 保证无重复地产生 H 的全部 T_1 .因此 $P_H(T_1) = P$.证毕.

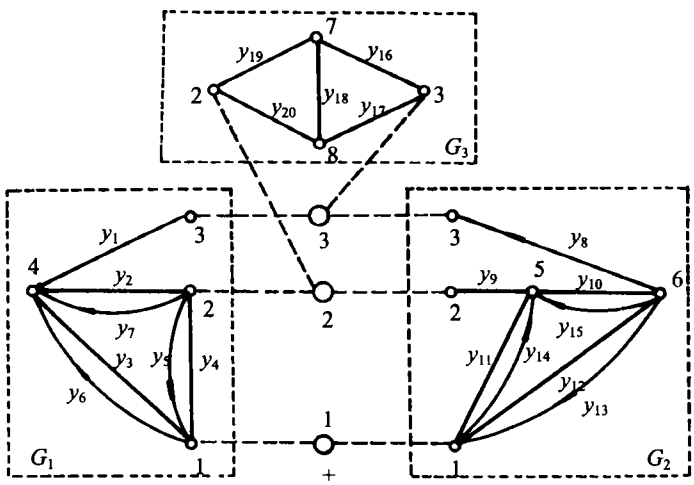


图3 超网络N的混合图G

推论 4.1 在算法 SSTGDCM 的步骤 2 中，若 E_j 是无向的，仅需找出 E_j 的无向分解集 SD_j，$DM \leftarrow SD_j$，并且在步骤 3 中当从 DM 取出一个 D_j 时可以把赋于 $E_k \in EN$ 的端点 $v_i \in [E_k \cap (w_1 \cup \cdots \cup w_h)]$ 以“+”极性的过程省去.

算法 SSTGDCM 中求 SST 部分的计算时间复杂度（决定于步骤 2 ～9）是 $O(en_t)$，e 是 H 的超边数，n_t 是 SST 的叶点数.叶点权 $Y(n) = P_j(rw)$ 是用算法 DKTPCG[9] 计算的，其复杂度参看文献 [9] .

需要进一步研究的问题有：(1)PDCP 法和 GDC 法中 $P[T_1(y)]$ 的项数的计算和估计；(2)分解 – 收缩法在超图理论其他问题中的应用；(3)分解 – 收缩法与并行算法的结合.

参 考 文 献

1 Chen W K. Computer Generation of Trees and Cotree in a Cascade of Multiterminal Networks. IEEE Trans on CT, 1969, CT – 16(11): 518 ～526

2 Chen W K, Goyal I C. Tables of Essential Complementary Partitions. IEEE Trans on CT, 1971, CT – 18(9): 562 ～563

3 Kajitani T, Ueno S, Chen W K. On the Number of Essential Complementary Partitions. IEEE Trans on CAS, 1982, CAS – 29(8): 572 ～574

4 吴新余.论 $k = 6$ 时基本互补划分表的产生.见：中国电子学会线路与系统学会“一般理论专题讨论会”.

北京,1983

5 马昭彦,全一男.$k=7,8,9,10$ 的基本互补划分的生成.见:中国电子学会线路与系统学会第五届年会论文集.西安,1984.255～259

6 黄汝激.通过有向 k 超树产生有向图的有向 k 树多项式.电子学报,1987,15(1):1～9

7 黄汝激.超网络的有向 k 超树分析法.电子科学学刊,1987,9(3):244～255

8 Horowitz E.Fundamentals of Computer Algorithms.Potomac:Computer Science Press, 1978.370

9 黄汝激.混合图的有向 k 树多项式的产生和状态空间树.电子学报,1987,15(5):8～14.

Improvement and Development for Chen's Essential Complementary Partition Method

Huang Ruji

College of Automation and Information Engineering, USTB, Beijing 100083, PRC

ABSTRACT By applying the direted hypergraph theory, the concepts of the decomposition and contraction of a hyperedge are introduced. Chen s essential complementary partition (ECP) method for solving the cascade problem of two passive networks is improved to form a more efficient decomposition – contraction pair (DCP) method, then it is developed to form the directed (positive-rooted) decomposition – contraction pair (PDCP) method, to solve the cascade problem of two active networks. Furthermore, it is developed to form the general decomposition – contraction (GDC) method, to solve the problem of finding symbolic network functions for the interconnection of several active networks.

KEY WORDS symbolic network function, directed hypergraph theory, decomposition-contraction pair method