第7卷 第2期 电子科学学刊 Vol. 7, No. 2
1985 年 3 月 JOURNAL OF ELECTRONICS Mar., 1985

关于 Coates 图的 1-因子的算法*

陆生勋 朱加生
(杭州大学)

提 要

本文是作者 1982 年一文的继续，根据该文使用星积产生 1-因子的定理，我们先讨论了一类特殊图，阐明星积和求排列的关系，然后推广到一般情况. 最后，给出求 Coates 图的 1-因子的算法.

一、引 言

利用拓扑方法分析有源网络能控制舍入误差，便于研究灵敏度或其他某些参量对电路设计的影响，也便于分析和综合某些非线性网络[1]. 这种方法往往归结为求有向图的某类子图，如 1-因子、半因子、有向树等. 作者[2]曾提出使用星积产生 1-因子和 1-因子连通. 本文的目的是阐明算法的具体步骤.

二、概念和定理

凡本文未定义的术语和符号均与文献[3]相同. 例如边的"积" $e_{i_1}e_{i_2}\cdots e_{i_k}$ 表示这些边及其端点构成的图. e_1 和 e_2 的和图 (sum graph) $e_1 \cup e_2$ 记作 e_1e_2. 设 $n \times n$ 阶矩阵 $A=[(i,j)]$ 的全部元素 (i,j) 均不为零，则伴随的 Coates 图 $G_c(A)$ 称为全图.

有向图 G 的子图集合 h_1, h_2 的星积定义为:

$$h_1 * h_2 = \{x \cup y;\ x \in h_1,\ y \in h_2,\ \text{且}\ \deg^+_{x\cup y}(i) < 2,\ \deg^-_{x\cup y}(i) < 2\}, \tag{1}$$

其中 i 是边的端点.

$$h_1 * \phi = \phi * h_1 = \phi. \tag{2}$$

定理 1[2] 设 $G_c(V,E)$ 为 Coates 图，$V=\{1,2,\cdots,n\}$, $S_k=\{(k,t);(k,t)\in E,\ t\in V\}$ (S_k 是节点 k 为始点的边集)，则 S_k 的星积给出 $G_c(V,E)$ 的全部 1-因子，

$$C = S_1 * S_2 * \cdots * S_n. \tag{3}$$

推论 1 C 等于从 $S_1, S_2, \cdots, S_n$ 的笛卡儿乘积中删去有两条以上的边具有相同终点的那些子图.

证 (1)在式 (3) 求星积的运算中，各 S_i 的出边的始点都不相同，而且在每个 S_i 中

* 1983 年 7 月 11 日收到，1984 年 11 月 5 日修改定稿.

恰取一个元素，显然满足式 (1) 中的条件 $\deg^{+}_{x\cup y}(i) < 2$. (2) 注意到笛卡儿乘积的定义为

$$S_1 \times S_2 = \{g_i \cup g_j ; g_i \in S_1, g_j \in S_2\}, \tag{4}$$

从中删去上述子图意味着删去不满足条件 $\deg^{-}_{x\cup y}(i) < 2$ 的子图. 推论得证.

设集合 $V = \{1, 2, \cdots, n\}$, 通常称这 n 个数码的一个有序列 $i_1 i_2 \cdots i_n$ 为排列，i_k 是排列的第 k 位置的元素，称为排列的第 k 个元素.

为了本文算法的需要，将排列分为两种：(1)无约束排列，简称排列，(2)有约束排列.

定义 1　设集合 $V = \{1, 2, \cdots, n\}$, 将 V 复制成 n 个集合 $V_i (i = 1, 2, \cdots, n)$, $V_i = V$. 求这 n 个集合的笛卡儿积

$$V_1 \times V_2 \times \cdots \times V_n,$$

删去集中含有重复数码的各项，剩下的全体定义为 V 的全部无约束排列.

例 1　$V = \{1, 2, 3\}$, 作 $V_1 = V_2 = V_3 = V$.

$$\begin{aligned} V_1 \times V_2 \times V_3 = \{&111,112,113,121,122,123,131,132,133,211,\\ &212,213,221,222,223,231,232,233,311,312,\\ &313,321,322,323,331,332,333\}, \end{aligned}$$

删去含有重复数码的各项得全部无约束排列：

$$123,\ 132,\ 213,\ 231,\ 312,\ 321. \tag{5}$$

定义 2　设集合 $V = \{1, 2, \cdots, n\}$, $V_i \subset V$, $i = 1, 2, \cdots, n$, 其中至少有一个 V_i 是 V 的真子集，求这 n 个子集合的笛卡儿积

$$V_1 \times V_2 \times \cdots \times V_n,$$

删去含有重复数码的各项，剩下的全体定义为 V 的全部有约束排列.

例 2　$V = \{1, 2, 3\}$, 子集 $V_1 = \{1, 2, 3\}$, $V_2 = \{2, 3\}$, $V_3 = \{1, 2\}$.

$$\begin{aligned} V_1 \times V_2 \times V_3 = \{&121, 122, 131, 132, 221, 222, 231, 232, 321,\\ &322, 331, 332\}, \end{aligned}$$

删去含有重复数码的各项后得全部有约束排列

$$132,\ 231,\ 321 \tag{6}$$

命题 1.　V 共有 $n!$ 个无约束排列.

命题 2.　设 V 共有有约束排列 p 个，则

$$p < n! \tag{7}$$

三、算　法

以下先阐明 1-因子的产生可以归结为求排列问题，然后给出求排列的算法. 为清楚起见，先讨论一类特殊图，再推广到一般情况.

(一) 1-因子和排列

1. 全图的 1-因子　设 G_c 是 n 个节点的全图，各节点的出边集为：

$$\left.\begin{aligned} S_1 &= \{(1,1),(1,2),\cdots,(1,n)\},\\ S_2 &= \{(2,1),(2,2),\cdots,(2,n)\},\\ &\vdots\\ S_n &= \{(n,1),(n,2),\cdots,(n,n)\}. \end{aligned}\right\} \tag{8}$$

根据推论 1.

$$C = \{(1,1),(1,2),\cdots,(1,n)\}\times\{(2,1),(2,2),\cdots,(2,n)\}\times\cdots \times\{(n,1),(n,2),\cdots,(n,n)\},\text{且删去 } \deg^-(i)\geqslant 2 \text{ 的子图.} \tag{9}$$

此结果记作

$$C = \{(1,i)(2,i_2)\cdots(n,i_n)\}. \tag{10}$$

令上式圆括号中的始点恒按自然数的顺序排列，并且注意到当且仅当子图$(1,i_1)(2,i_2)\cdots(n,i_n)$有 $\deg^-(i)\geqslant 2$ 的节点时，其中圆括号的终点所构成的集合 $\{i_1,i_2,\cdots,i_n\}$ 才有两个或两个以上元素是相同的. 显然，每个子图都有如此对应的终点集合. 式(9)中删去 $\deg^-(i)\geqslant 2$ 的子图，意味着同时删去了有相同元素的那些终点集合，剩下的是没有相同元素的终点集合. 因此计算式 (9) 等价于求终点集合 $\{1,2,\cdots,n\}$ 的全部无约束排列 $\{i_1i_2\cdots i_k\cdots i_n\}$，再将始点 k 和 i_k 配对，得

$$\{(1,i_1)(2,i_2)\cdots(n,i_n)\}. \tag{11}$$

这便是全部 1-因子.

例 3 G_c 为三个节点的全图，$V=\{1,2,3\}$. 全部无约束排列为

$$123,\ 132,\ 213,\ 231,\ 312,\ 321$$

将 1，2，3 依次与排列的第 1，2，3 三个元素配对，得

$$C = \{(1,1)(2,2)(3,3),(1,1)(2,3)(3,2),(1,2)(2,1)(3,3),(1,2)(2,3)(3,1),(1,3)(2,1)(3,2),(1,3)(2,2)(3,1)\}.$$

2. 任意 Coates 图的 1-因子 设 G_c 是除全图外有 n 个节点的任意 Coates 图，则各出边集为

$$\left.\begin{aligned} S_1 &= \{(1,i_1),(1,i_2),\cdots\},\\ S_2 &= \{(2,i_1),(2,i_2),\cdots\},\\ &\vdots\\ S_n &= \{(n,i_1),(n,i_2),\cdots\}, \end{aligned}\right\} \tag{12}$$

其中 $\{i_1,i_2\cdots\}\subset\{1,2,\cdots,n\}$，且至少有一个真子集.

同样，可证明按推论 1 求 1-因子等价于先求全部有约束排列 $\{i_1,i_2,\cdots,i_k,\cdots\}$，再使自然数 k 和排列的第 k 个元素配对成 $\{(1,i_1)(2,i_2)\cdots\}$.

例 4 设 G_c 的节点数是 3，$S_1=\{(1,1),(1,2),(1,3)\}$，$S_2=\{(2,2),(2,3)\}$，$S_3=\{(3,1),(3,2)\}$，则全部有约束排列如式 (6)，因此，配对后得

$$C = \{(1,1)(2,3)(3,2),(1,2)(2,3)(3,1),(1,3)(2,2)(3,1)\}.$$

综上所述，求 Coates 图的 1-因子的算法归结为求全部无约束排列或有约束排列问题，关键在于寻求一种产生排列的好算法.

(二) 产生排列的算法

1. 算法 1 无约束排列的产生为了计算 $V'=\{1,2,\cdots,n\}$ 的全部无约束排列，引

入 $n \times n$ 阶矩阵 V 和两个 n 维向量 B, W。矩阵 V 提供构成排列的元素，排列的第 i 个元素恒取自 V 的第 i 行。开始时置 V 的各行元素都与 V' 相同，

$$V = \begin{bmatrix} 1 & 2 & \cdots & n \\ 1 & 2 & \cdots & n \\ \vdots & & & \\ 1 & 2 & \cdots & n \end{bmatrix}. \tag{13}$$

W 存贮求得的一个排列。B 是按下述规则变化的向量，由 B 的变化控制排列的产生。当 B 的每个元素 $B_i = i - 1$ 时，称 B 处于初态；当 B 的第 i 个元素 $B_i = i - 1$ 时，称 B_i 处于初态。如果 B 的每个元素 $B_i = 0, i = 1, 2, \cdots, n$，称 B 处于末态。

(1) 设 B 处于某态，从 B_1 开始依下标增序检查 B_i 元素，记 B_k 是第 1 个遇到的非零元素，置 $B_k = B_k - 1$。令 B_k 以后的元素 $B_{k+1}, B_{k+2}, \cdots, B_n$ 保持不变，B_k 以前的元素 $B_1, B_2, \cdots, B_{k-1}$ 回到初态。如此得到 B 的新态后，转入 (2)。

(2) 检查 B 是否到达末态。是，停止运算。否，则转入 (1)。

例如 $B = [000345]$ 时，先变到 $[012245]$，再变到 $[002245]$，依此类推。

产生 $V' = \{1, 2, \cdots, n\}$ 的全部无约束排列的算法分五步：

第 1 步，置 V 如式 (13)，

$$B = [0\ 1\ 2 \cdots n-1],\quad k = n.$$

第 2 步，从矩阵 V 的第 k 行到第 1 行每行选一个元素存入 W 作为 W_k 到 W_1 的 k 个元素。要求 (1) 是非零元素，(2) 使 W 的 n 个元素均不相同。若某行有若干元素供选择时，在程序中可作具体规定，譬如选下标最大的元素。其次，将 V 中被选中的元素置零。最后，从 W 输出一个排列后，转入第 3 步。

第 3 步，依次检查 $B_1, B_2, \cdots$，若均为零，停止运算，否则，设第 1 个遇到的非零元素为 B_k，记录 k。

第 4 步，恢复 V 中第 $k-1$ 行到第 1 行的元素，使这些行的元素与式 (13) 相应行的元素相同。然后转入第 5 步。

第 5 步，根据 B 的变化规则将 B 变到下一状态后，转入第 2 步。

例 5　求集合 $\{1, 2, 3, 4\}$ 的全部排列。

第 1 步，令

$$V = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{bmatrix},\quad B = [0\ 1\ 2\ 3],\quad k = 4.$$

第 2 步，从 V 的第 $k = 4$ 行到第 1 行各取一个元素。第 4 行有 4 个元素可供选择，选最大下标 $V_{44} = 4$ 存入 W_4。第 3 行有 3 个元素 V_{31}, V_{32}, V_{33} 可供选择，选出了存入 W_3。其余类此，得到

$$W = [1234].$$

置

$$V=\begin{bmatrix}0&2&3&4\\1&0&3&4\\1&2&0&4\\1&2&3&0\end{bmatrix},$$

输出 W，为简便起见，得到的 W 和 B 常记作

$$\left.\begin{matrix}1&2&3&4\\(0&1&2&3)\end{matrix}\right\}$$

第 1 行是排列，第 2 行是 B.

第 3 步，检查 B_i，因 $B_2\neq 0$，可以 $k=2$，$k=1=1$.

第 4 步，恢复 V 中第 $k-1=1$ 到第 1 行元素，得

$$V=\begin{bmatrix}1&2&3&4\\1&0&3&4\\1&2&0&4\\1&2&3&0\end{bmatrix}.$$

第 5 步，根据 B 的变化规则，置 $B=[0\ 0\ 2\ 3]$，转入第 2 步

第 2 步，从矩阵 V 的第 $k=2$ 行到第 1 行各选一个元素存入 W_2，W_1，得

$$W=[2\ 1\ 3\ 4],$$

置

$$V=\begin{bmatrix}1&0&3&4\\0&0&3&4\\1&2&0&4\\1&2&3&0\end{bmatrix},$$

如此循环，产生如下无约束排列：

$$\left.\begin{matrix}1&2&3&4\\(0&1&2&3)\end{matrix}\right\},\ \left.\begin{matrix}2&1&3&4\\(0&0&2&3)\end{matrix}\right\},\ \left.\begin{matrix}1&3&2&4\\(0&1&1&3)\end{matrix}\right\},\ \cdots,\ \left.\begin{matrix}4&3&2&1\\(0&0&0&0)\end{matrix}\right\}. \tag{14}$$

2. 证明 先证明一个命题.

命题 3. B 从初态到末态共有 $n!$ 态.

证 用归纳法. 显然，$n=3$ 时成立，共有 3! 个态. 假设 $n=k$ 时成立，有 $k!$ 个态：

$$\left.\begin{matrix}0&1&\cdots&k-1\\0&0&\cdots&k-1\\\vdots&&&\\0&0&\cdots&0\end{matrix}\right\} \tag{15}$$

不妨视式(15)为 $k!\times k$ 阶矩阵，并将它复制成 $(k+1)$ 个矩阵. 对每个矩阵增添一列使第 i 个矩阵的第 $k+1$ 列的元素都是 $k-i+1$. 于是有

$$\begin{bmatrix}0&1&\cdots&k-1&k\\0&0&\cdots&k-1&k\\\vdots&&&&\\0&0&\cdots&0&k\end{bmatrix},\begin{bmatrix}0&1&\cdots&k-1&k-1\\0&0&\cdots&k-1&k-1\\\vdots&&&&\\0&0&\cdots&0&k-1\end{bmatrix},\ \cdots,$$

$$\begin{bmatrix} 0 & 1 & \cdots & k-1 & 0 \\ 0 & 0 & \cdots & k-1 & 0 \\ \vdots & & & & \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}. \tag{16}$$

依次考察各矩阵的行，正符合 B 的变化规则，且有 $(k+1)!$ 个态．命题得证．

现在用归纳法证明全部无约束排列的产生，集合 $V'=\{1,2,3\}$ 时，有

$$\left\{\begin{matrix}1&2&3\\(0&1&2)\end{matrix}\right\},\ \left\{\begin{matrix}2&1&3\\(0&0&2)\end{matrix}\right\},\ \left\{\begin{matrix}1&3&2\\(0&1&1)\end{matrix}\right\},\ \left\{\begin{matrix}3&1&2\\(0&0&1)\end{matrix}\right\},\ \left\{\begin{matrix}2&3&1\\(0&1&0)\end{matrix}\right\},\ \left\{\begin{matrix}3&2&1\\(0&0&0)\end{matrix}\right\}.$$

假设 $V'=\{1,2,\cdots,k\}$ 时成立，有以下 $k!$ 个排列：

$$\left\{\begin{matrix}1&2&3&\cdots&k\\(0&1&2&\cdots&k-1)\end{matrix}\right\},\ \left\{\begin{matrix}2&1&3&\cdots&k\\(0&0&2&\cdots&k-1)\end{matrix}\right\},\ \cdots,\ \left\{\begin{matrix}k&k-1&\cdots&1\\(0&0&\cdots&0)\end{matrix}\right\}. \tag{17}$$

证明 $V''=\{1,2,\cdots,k+1\}$ 时也成立．先按式 (13) 写出 $(k+1)\times(k+1)$ 阶增广矩阵 V．

$$V=\begin{bmatrix} 1 & 2 & \cdots & k & k+1 \\ 1 & 2 & \cdots & k & k+1 \\ \vdots & & & & \\ 1 & 2 & \cdots & k & k+1 \end{bmatrix}. \tag{18}$$

其次，对式(17)各项增添元素 $\left\{\begin{matrix}k+1\\(k)\end{matrix}\right\}$ 写成

$$\left\{\begin{matrix}1&2&3&\cdots&k&k+1\\(0&1&2&\cdots&k-1&k)\end{matrix}\right\},\ \left\{\begin{matrix}2&1&3&\cdots&k&k-1\\(0&0&2&\cdots&k-1&k)\end{matrix}\right\},\ \cdots,$$
$$\left\{\begin{matrix}k&k-1&\cdots&1&k+1\\(0&0&\cdots&0&k)\end{matrix}\right\}. \tag{19}$$

注意到式(17)第 1 行是 V' 的全部排列，$k+1\in V'$，所以式(19)各项的第 1 行正是 V'' 的 $k!$ 个排列．根据算法 1 不难看出，这些排列是第一批产生的 $k!$ 个排列．其次，由算法 1 第 2 步给出 $V_{k+1,k+1}=0$，第 3 步给出 $B_{k+1}\neq 0$，第 4 步恢复第 k 行到第 1 行的元素得

$$V=\begin{bmatrix} 1 & 2 & \cdots & k & k+1 \\ 1 & 2 & \cdots & k & k+1 \\ \vdots & & & & \\ 1 & 2 & \cdots & k & k+1 \\ 1 & 2 & \cdots & k & 0 \end{bmatrix}. \tag{20}$$

然后转入第 5 步，再到第 2 步．这时因 $V_{k+1,k+1}=0$，所以从第 $k+1$ 行中选取其它元素 k 存入 W_{k+1}，结果 $W_{k+1}=k$ 又排斥其它 $W_i(i=1,2,\cdots,k)$ 等于 k，故在 B 的控制下产生第二批 $k!$ 个排列．类此，直到 V 中第 $k+1$ 行的非零元素被取尽，共产生 $(k+1)!$ 个排列，同时 B 也到达末态，结束运算．以上证明了求无约束排列的正确性．

3. 讨论　(1) 我们感兴趣的是产生排列而不注意先产生哪一个排列，所以当第 i 行有若干元素供选择时，也可以规定其它选取方式，例如选下标最小的元素存入 W_i．其次，从式(14) B 的变化可以看出 B 的左端的元素变化较频繁，或称之为自左至右的变化．

上述算法中受 B 控制的 W 也是自左至右的变化，其实修改算法将 W 改为自右至左的变化也能得到全部排列(参见算法 2)。

(2) 欲求例 2 的有约束排列时，如果将算法 1 式(13)第 i 行内不属于子集 V_i 的元素置零，使

$$V=\begin{bmatrix}1&2&3\\0&2&3\\1&2&0\end{bmatrix}, \tag{21}$$

则因在产生全部无约束排列的运算中筛除含零的排列，故能得到例 2 的结果。当 V 是稀疏矩阵时，这样处理的主要缺点是浪费机时，譬如有 6×6 阶矩阵 V，其中第 4 行非零元素甚少，根据算法第 2 步拟从第 $k=4$ 到第 1 行选取 $W_4, W_3, \cdots$ 时，如果第 4 行元素已均为零，无元素可选，而 $B=0\ 0\ 0\ 3\ 4\ 5$，根据 B 的变化：

$$0\ 0\ 0\ 3\ 4\ 5,\ 0\ 1\ 2\ 2\ 4\ 5,\ 0\ 0\ 2\ 2\ 4\ 5,\ \cdots,\ 0\ 0\ 0\ 0\ 4\ 5.$$

仍继续由前 4 行中选取元素，显然，仍无元素可选，直到 $B=0\ 0\ 0\ 0\ 4\ 5$ 时，才允许自前 5 行中选取元素。为了避免这种情况，可令 B 从 $0\ 0\ 0\ 3\ 4\ 5$ 直接变到 $0\ 0\ 0\ 0\ 4\ 5$，约节省 $3!$ 个 B 的变化。

4. 算法 2 有约束排列的产生 设 $V'=\{1, 2, \cdots, n\}$，$V'_i\subset V'$，$\max\{|V'_i|;\ i=1, 2, \cdots, n\}=m$，其中至少有一个 V'_i 是 V' 的真子集。考虑到标准 FORTRAN 的循环参量的步长不允许负值，本算法采用 B 自右至左控制 W 的变化，具体步骤基本上和算法 1 相同，分以下五步：

第 1 步，置

$$V=\begin{bmatrix}i_1&i_2&\cdots&i_m\\i_1&i_2&\cdots&i_m\\\vdots&&&\\i_1&i_2&\cdots&i_m\end{bmatrix}, \tag{22}$$

V 是 $n\times m$ 阶矩阵，其中每行第 1 到第 $|V'_i|$ 个元素同 V'_i，第 $|V'_i|+1$ 到第 m 个元素置零。令 $B=[0\ 1\ 2\cdots m-1]$，$l=1$。

第 2 步，(1) 从矩阵 V 的第 $l(=n-k+1)$ 行到第 n 行，每行选一个元素作为 $W_l, W_{l+1}, \cdots, W_n$ 的元素，要求与算法 1 相同。其次，将 V 中被选中的元素置零，输出一个排列后转入第 3 步。(2)如果第 j 行 $(l\leqslant j\leqslant n)$ 已无元素可选，置 $B_1, B_2, \cdots, B_{n-j+1}$ 等于零。转入第 3 步。

第 3 步，依此检查 $B_1, B_2, \cdots, B_n$，若均为零，停止运算。否则，若第 1 个遇到的非零元素为 B_k，置 $l=n-k+1$。

第 4 步，恢复 V 中第 $l+1$ 行到第 n 行的元素。

第 5 步，将 B 变到下一状态后，转入第 2 步。

这里与算法 1 的主要区别是在第 2 步中增加步骤 (2)。在计算稀疏矩阵时，这一步骤尤为重要，因为它能节省机时，原因已在讨论(2)中举例说明过。其次，是引入 $n\times m$ 阶矩阵 V 代替式(13)，以便节省存贮。本算法采用 B 自右至左控制 W 的变化，所以在第 j 行无元素可取时，置 $B_1, B_2, \cdots, B_{n-j+1}$ 等于零而不是置 $B_1, B_2, \cdots, B_j$ 等于零。同理，在

第 1 步中置 $l = 1$ 代替算法 1 第 1 步置 $k = n$.

证明　同算法 1，从略.

参　考　文　献

[1] P. M. Lin and G. E. Alderson SNAP, A Computer Program for Generating Symbolic Network Function, Thesis, Purdue University, 1970.
[2] 陆生勋等，电子学通讯，4(1982)，198.
[3] W. K. Chen, Applied Graph Theory, North-Holland, Amsterdam, 1976.

AN ALGORITHM TO FIND 1-FACTORS OF A COATES GRAPH

Lu Shengxun, Zhu Jiasheng
(Hangzhou University)

This paper continues the study in autherls former paper (1982). According to the theorem for generating 1-factors by star product described in former paper, we consider a class of special graphs at first, for which the relations between star product and permutation are established and then the relations are extened to the general case. Finally, we give an algorithm to find the 1-factors of a Coates graph.