第9卷 第6期
1987年11月

电子科学学刊
JOURNAL OF ELECTRONICS

Vol.9, No. 6
Nov., 1987

生成有向图全部有向回路的一个有效的回路向量空间算法*

熊 德 琰
(同 济 大 学)

提 要

本文提出一种生成有向图全部有向回路的、有效的回路向量空间算法，其中每个有向回路都由一个连支定义的基本回路(有向回路或半回路)和一组已获得的有向回路的环和产生，同时可将每个有向回路用一个选定的有向回路基集的线性组合表示.

一、引 言

线图G的全部回路和不共边回路之并加上空集，构成一个定义在整数模2域$GF(2)$上的向量空间，其向量加法定义为回路边集的环和. 图G的回路基集定义为回路空间的基. 回路空间维数等于图G的零度μ. 回路空间包含2^μ个向量.

回路向量空间法，就是从一个回路基集出发，通过环和运算，生成全部回路的一种算法. 如果对运算的空间不加任何限制，就必须计算除基本回路和零元以外的$2^\mu-\mu-1$个向量，其中含有大量冗余项[1].

多年来有许多图论学者致力于研究只计算全部向量的一个子集，就能产生无向图的全部回路的方法，即所谓"删节技术". 提出这样的算法，有 Welch[2]、Hsu 和 Honkanen[3]、Gibbs[4]、Mateti 和 Deo[5]等，可是，在最坏情形下，仍必须计算所有的回路向量.

不久前 Maciej M. Sysl'o[6] 提出一个应用范围仅限于平面无向图的、有效的回路向量空间算法.

对于有向图，原理上也可以采用回路向量空间法. 但是，有向回路集合通常只是回路集合中一个很小的子集，有时是空集. 因此，人们一直认为对于有向图，回路向量空间法是不适用的[1].

最近，本文作者提出一个生成有向图的有向回路基集的有效算法，和一个在环和运算中判别有向回路和不共点有向回路之并的简单方法[7-9]. 本文是上述工作的继续，在所提出的算法中，每个有向回路由一个连支定义的基本回路和已经获得的有向回路的一个子集的环和产生. 每作一次环和运算，即得一个有向回路，没有冗余项. 算法关键在于系统

1986年4月23日收到，1987年2月11日修改定稿.

地生成参加环和运算的有向回路子集．为此必须存储生成的全部有向回路的信息．

二、算法的理论基础

从有向图的任一顶点出发作一次深度优先搜索（DFS），可以得到一个有向林或一个有向树，识别全部最大强连通子图（MSC），同时对全部有向边进行划分．首先划分二种边集，一种属于 MSC，另一种不属于任何 MSC，即属于某个单向割集，在此种割集中所有边和割集方向一致．每个 MSC 有一个外向树，即根顶点的入度为零的有向树，它将 MSC 的边集划分为树支和连支． 有向图的两个顶点 v 与 w 间若存在一条由 v 到 w 的有向树路径，即全由树支构成的有向路径(以下简称树路径)，则称 v 是 w 的祖先，而 w 是 v 的子孙；当边 (v, w) 是一个树支时，则称 v 为 w 的父顶点，w 为 v 的子顶点．于是，连支按照其始端 v 与末端 w 的关系又可划分为三类：(1)后退连支，w 是 v 的祖先或父顶点；(2)横跨连支，v 与 w 间不存在有向树路径(图 1(a))；(3)前进连支，v 是 w 的祖先或父顶点(图1(b))．只有属于 MSC 的树支和连支才能构成有向回路．

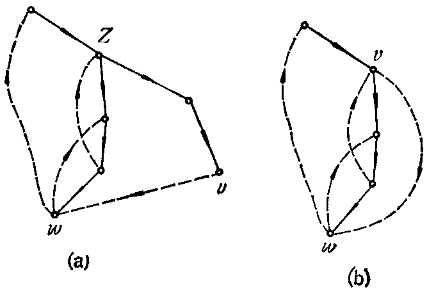


图 1 (v, w) 是横跨连支(a)和前进连支(b)的情形

在 DFS 搜索过程中，由已访问过的顶点和边构成的有向图的子图称为搜索图．文献[8]中证明了有向图的下列性质．当搜索图上每添加一个不属于单向割集的连支时，搜索图上便增加一组包含这个连支的新的有向回路；因为一般有向回路包含不止一个连支，这个新添加的连支便称为这组回路的“主导连支”；这组有向回路被定义为“共主导连支有向回路集”．由后退连支定义的共主导连支有向回路集只含一个回路，而由横跨连支或前进连支定义的至少含有一个回路．文献[8]并得出以下结论：(1)共主导连支有向回路集是非空和不相交的；(2)所有共主导连支有向回路集的总和，构成全部有向回路集合；(3)从每个共主导连支有向回路集任意取一个回路，可得一个有向回路基集；(4)强连通图的有向回路基集的基数等于它的零度．任意有向图的有向回路基集等于它所有强连通子图的有向回路基集的总和；(5)图 1(a) 和 (b) 分别表示，在 DFS 过程中被检查的主导连支 (v, w) 是横跨连支和前进连支的情形． 用符号 $Z(v)$ 表示情形 (a) 中 v 与 w 的最近共同祖先 Z 或情形 (b) 中的 v 点．已经证明，由 $Z(v)$ 到 w 含有连支 (v, w) 的有向路径和每一个由 w 到 $Z(v)$ 的有向路径，构成以 (v, w) 为主导连支的共主导连支有向回路集中的每一个回路[7,8]．

在以上结果的基础上，发展了一种生成全部有向回路的回路向量空间法．下面介绍此法的原理．

为叙述方便起见，引入以下术语．MSC 中，由单一的主导连支 (v, w) 与树支构成的回路称为基本回路，记为 C_f．当主导连支是后退连支时，它是一个有向回路，其它两种情形(图 1(a) 和 (b)) 中，C_f 是个“半回路”． 有向图基本回路的这个定义与无向图的相应定义一样．当 C_f 是半回路时，它包含两条有向路径：一条从 $Z(v)$ 到 w，包含主导连支

(v, w) 的有向路径，称为"主导路径"，记为 P_d，另一条由 $Z(v)$ 到 w 的有向树路径，称为"正则路径"，记为 P_p．若它们都以边集表示，则有

$$C_f = P_d \cup P_p = P_d \oplus P_p. \tag{1}$$

任意一条从 w 到 $Z(v)$ 的有向路径称为"后退路径"，记为 P_i，$i = 1, 2, \cdots, s$，设有 s 条后退路径．主导路径 P_d 与每一条后退路径 P_i 构成一个有向回路，记为 C_{di}，

$$C_{di} = P_d \cup P_i = P_d \oplus P_i. \tag{2}$$

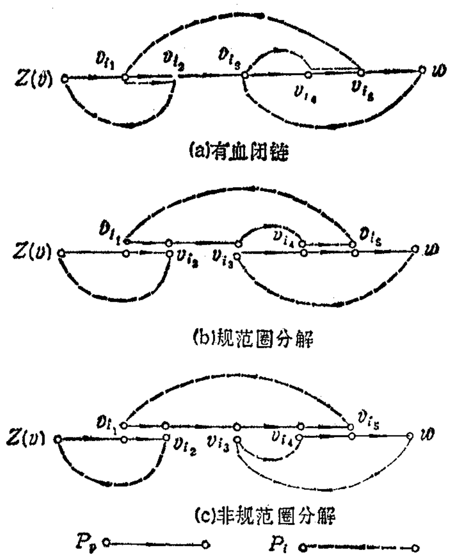
正则路径 P_p 与每一条后退路径 P_i 构成一个有向闭链，记为 L_i，$L_i = P_p + P_i$，加法"＋"定义为代数和，L_i 中的边和顶点至多重复一次．特殊情况下，L_i 是个简单的有向回路．

引理 1　设 L_i 是由正则路径 P_p 和后退路径 P_i 构成的有向闭链，L_i 总是可以分解为一系列(至少一个)有向回路 C_i，$i \in I = \{1, 2, \cdots, r\}$，而且

$$L_i = P_p + P_i = C_1 + C_2 + \cdots + C_r, \tag{3a}$$

$$P_p \oplus P_i = C_1 \oplus C_2 \oplus \cdots \oplus C_r = \oplus_{i \in I} C_i. \tag{3b}$$

证明　若 P_p 和 P_i 除 $Z(v)$ 和 w 以外无共同顶点，则 L_i 是个有向回路，引理 1 成立．若 P_p 和 P_i 在 $Z(v)$ 和 w 外有共同顶点和边，则设沿正则路径，共同顶点顺序为 $v_{i_1}, v_{i_2}, \cdots, v_{i_p}$ (图 2)．取 P_p 中从 $Z(v)$ 到 v_{i_1} 的子路径和 P_i 中从 v_{i_1} 到 $Z(v)$ 的子路径构成一个有向回路．从闭链 L_i 中拿掉这个回路，剩下的仍是一个闭链．新闭链比原来的闭链的相交顶点数目至少减少了一个，即 $Z(v)$．在新闭链中，顶点 v_{i_1} 代替了 $Z(v)$ 的位置．照上述方法分解下去，在有限次数内，使闭链只剩下一个有向回路，这样便完成了闭链的圈分解．所以，式 (3a) 成立；又因任意一条边至多在两个不同的有向回路中出现，所以式 (3b) 也成立．证毕．



(a) 有向闭链
(b) 规范圈分解
(c) 非规范圈分解
图 2　有向闭链的圈分解

一般而言，闭链的圈分解不是唯一的，但按证明中所述方法和步骤进行圈分解，其结果是唯一的，因为分解结果由形成闭链的两条路径的交点分布唯一地决定．这种圈分解，我们称它为"规范圈分解"．图 2 中 (b) 和 (c) 分别是图 2(a) 中闭链经过规范圈分解和非规范圈分解得到的两种不同结果．

容易证明，闭链 L_i 通过规范圈分解产生的有向回路集，具有以下四条性质：(1)每个有向回路与正则路径 P_p 至少在一个地点相交，按正则路径边顺序，第一个相交处是一段至少包含一个树支的树路径．这段树路径称为该有向回路的"有效段"．(2) 没有一个回路的有效段是另一个回路的有效段的子集，而且所有有向回路的有效段之并等于正则路径 P_p，即有向回路集覆盖整个正则路径．(3)正则路径上的边和顶点，除顶点 $Z(v)$ 和 w 以外，在集合的所有回路中，至多出现两次．(4)不在正则路径上的边和顶点以及顶点 $Z(v)$

和 w，在集合的所有回路中，至多出现一次.

定义 1 主导连支 (v, w) 定义的基本回路 C_f 的正则路径 P_p 的规范覆盖集(简称规范覆盖集或 NCS) 是搜索图中有向回路的、满足以上四个条件的一个子集.

在主导连支 (v, w) 是后退连支的情形中，C_f 是个有向回路，也是共主导连支有向回路集中的唯一回路.我们认为此时 C_f 的正则路径 P_p 收缩成一点，定义此时的规范覆盖集是一个空集. 于是得到一个统一的定理.

定理 1 一个主导连支所定义的基本回路 C_f 和它的正则路径的每一个规范覆盖集中的全部回路的环和，产生该主导连支定义的共主导连支有向回路集的每一个回路，既不重覆，也无遗漏.

证明 正则路径 P_p 的一个规范覆盖集中回路之和(代数和)构成一个有向闭链.(1) 由规范覆盖集的性质 2、3 和 4 可知，所形成的有向闭链可分解为正则路径 P_p 和另一条由 w 到 $Z(v)$ 的简单的后退路径 P_i. (2) 这条后退路径 P_i 与正则路径 P_p 构成的有向闭链，通过规范圈分解仍还原为 P_p 的规范覆盖集. 因此，后退路径与规范覆盖集是一一对应的.根据公式 (1)、(2) 和 (3b)，定理 1 成立. 证毕.

值得指出，按照定义 1，规范覆盖集不包含访问主导连支 (v, w) 时尚未出现在搜索图中，而以后可能出现的满足定义 1 的四个条件的回路集合，这样的回路集合对应一条尚未出现在搜索图中的，从 w 到 $Z(v)$ 的有向路径. 这条路径与当前的主导路径 P_d 形成有向回路，然而，按共主导连支有向回路集定义，它不属于当前主导连支 (v, w) 定义的此种回路集. 它包含着另一条将来才出现的主导连支，并将由该主导连支的正则路径的规范覆盖集产生出来.

DFS 过程第一个产生的有向回路是由后退连支产生的基本回路.第一个产生的非空的规范覆盖集是全由已获得的后退连支产生的基本回路构成的. 从此以后，在 DFS 过程的任何时候搜索图上的全部回路都可以根据定理 1 产生出来，所以，组成规范覆盖集的有向回路都是已获得的有向回路，而且每个有向回路都可用基集元素的线性组合表示.

设 $\{C_i | i \in I = \{1, 2, \cdots, c\}\}$ 是已经获得的有向回路集合. 正则路径有 s 个规范覆盖集. $I_1, I_2, \cdots, I_s$ 是整数集合 I 的子集. 第 j 个规范覆盖集表示为 $\{C_i | i \in I_j\}$.则有

$$C_f \oplus_{i \in I_j} C_i = C_f \oplus P_p \oplus P_j = P_d \oplus P_j = C_{dj}.$$

C_{dj} 为一有向回路.

新产生的共主导连支有向回路集的第一个有向回路是

$$C_{d1} = C_f \oplus_{i \in I_1} C_i. \tag{4a}$$

若选择 C_{d1} 为基本有向回路，并记为 C_{db}，则其余有向回路可由 C_{db} 表出:

$$\begin{aligned} C_{d2} &= C_f \oplus_{i \in I_2} C_i, \\ &= C_{db} \oplus_{i \in I_1} C_i \oplus_{i \in I_2} C_i, \\ C_{d3} &= C_{db} \oplus_{i \in I_1} C_i \oplus_{i \in I_3} C_i, \\ &\cdots\cdots\cdots\cdots \\ C_{ds} &= C_{db} \oplus_{i \in I_1} C_i \oplus_{i \in I_s} C_i. \end{aligned} \tag{4b}$$

当第一次利用非空的规范覆盖集求有向回路时，即第一次搜索到横跨型或前进型主

导连支时，公式 (4a) 和 (4b) 中的 C_i 都是已获得的由后退连支定义的有向回路，它们是当然的基本有向回路，因此公式 (4b) 同时给出新有向回路的、以有向回路基集的线性组合表示的公式。从此以后，每获得一个新有向回路，它的以有向回路基集的线性组合表示的公式都可利用公式 (4b) 求得，因为 (4b) 中所有 C_i 的以基集的线性组合表示的公式都是已知的。

三、算　　法

算法的主过程是对有向图进行一次深度优先搜索 (DFS)，同时实现以下目标：(1) 找一个有向生成树或林，(2)划分边集，(3)找出主导连支定义的基本回路 C_f 及其 P_p 和 P_d，(4) 识别所有 MSC. 图 3 给出了算法主过程的流程图(假设有向图不存在自环)。下面仅对主过程作简要说明.

在计算机程序中设立两个栈，亦即两个顶点集合：

(1) STACK 1　顶点 v 从被访问开始进栈，直到所属最大强连通子图被找出后集体出栈.

(2) STACK 2　顶点 v 从被访问开始进栈，当 v 的所有射出边检查完毕时出栈.

从图 3 的流程图可见，边的类型可以根据(1)DFN(v) 和 DFN(w) 的先后和 (2)w 是否存在于 STACK 1 和 STACK 2 中来判断. 若遇属于单向割集的边 (对应于流程图中的连线 a 和 b)，不作处理.

当顶点 v 退出 STACK 2 时，若 LOW(v) = DFN(v)，即不改初值，则 v 是一个 MSC 的树根，STACK 1 中从 v 以下的顶点都属同一 MSC.

当执行图 3 的方框 A，处理横跨型主导连支时，根据下式找出 v 与 w 的最近共同祖先 Z：

$$\mathrm{DFN}(Z) = \mathrm{MAX}(\mathrm{DFN}(x)/x \in \mathrm{STACK}\ 2\ \text{且}\ \mathrm{DFN}(x) \leqslant \mathrm{DFN}(w))$$

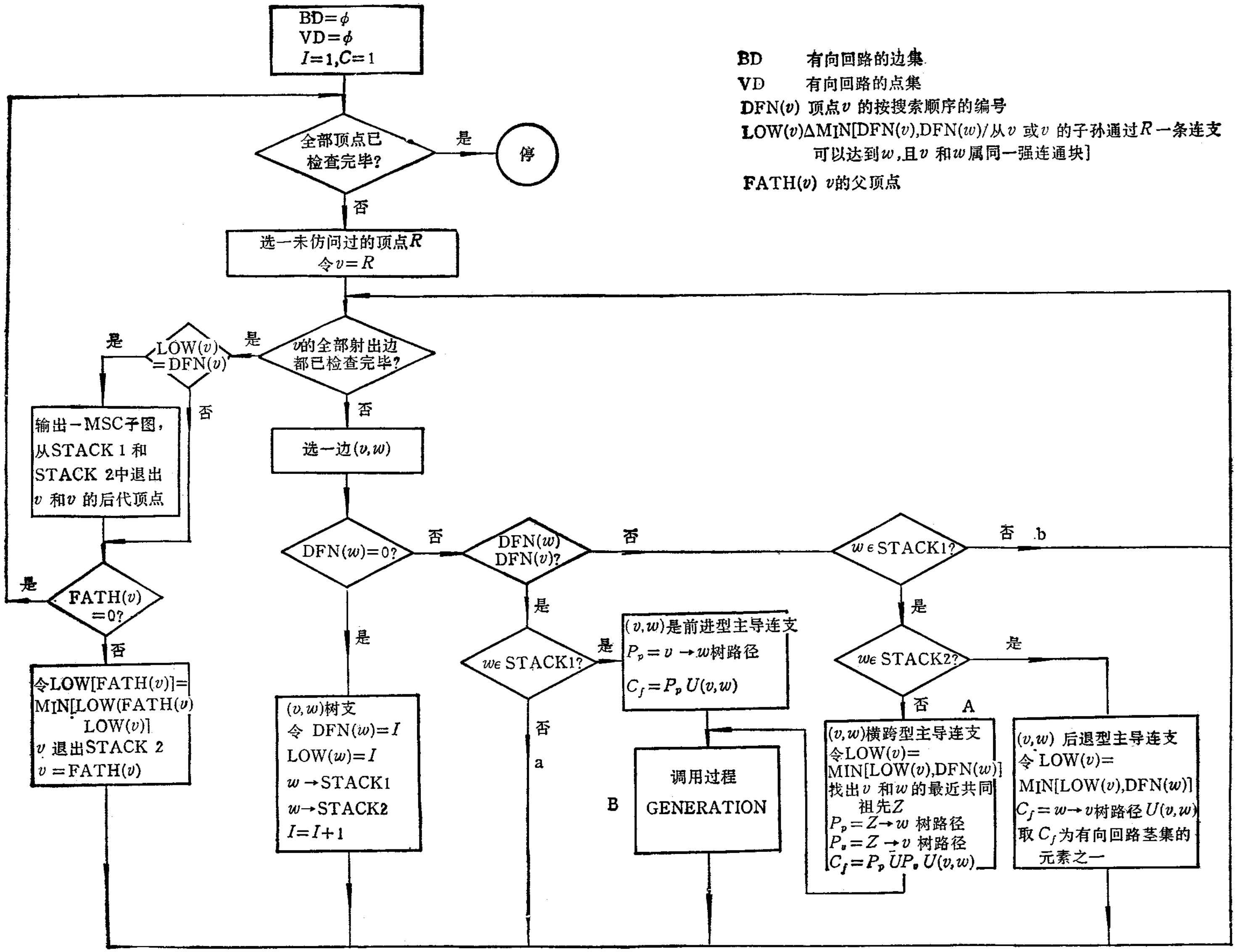
寻找 Z 是一个沿当前树路径的回溯过程，与此同时可获得图 1(a) 中的从 Z 到 v 的和从 Z 到 w 的树路径，即 $P_d \backslash (v, w)$ 和 P_p.

图 3 的方框 B，子过程 GENERATION 是本算法的核心，其流程图见图 4. 在这个子过程中，寻找规范覆盖集 NCS 和它与 C_f 的环和运算相结合地进行. RINGSUM 中纪录 C_f 与正在生长的 NCS 的环和. 子过程 GENERATION 实质上是在一个纪录已获得的有向回路的数据结构中，查找符合 NCS 定义的回路子集. 显然，NCS 的元素只存在于进入 GENERATION 之前就已获得的，且属于同一 MCS 的有向回路集合中. 进入过程 GENERATION 之后生成的有向回路不必考虑，因为它们都包含了主导路径 P_d，从而不可能具有覆盖 P_p 的有效段.

过程 GENERATION 建立在回溯法基础上，它的核心是图 4 的方框 C. 其主要操作步骤如下：

第 1 步　按照 P_p 的边和点序列来检查 C_a，查明是否属于以下情形之一：

情形 A　C_a 所含属于 P_p 中的任一顶点已含 NCS 于的两个回路中；



BD 有向回路的边集
VD 有向回路的点集
DFN(v) 顶点v 的按搜索顺序的编号
LOW(v)ΔMIN[DFN(v),DFN(w)/从v 或v 的子孙通过R一条连支可以达到w,且v 和w属同一强连通块]
FATH(v) v的父顶点

图 3 生成有向图的全部有向回路的回路向量空间算法流程图

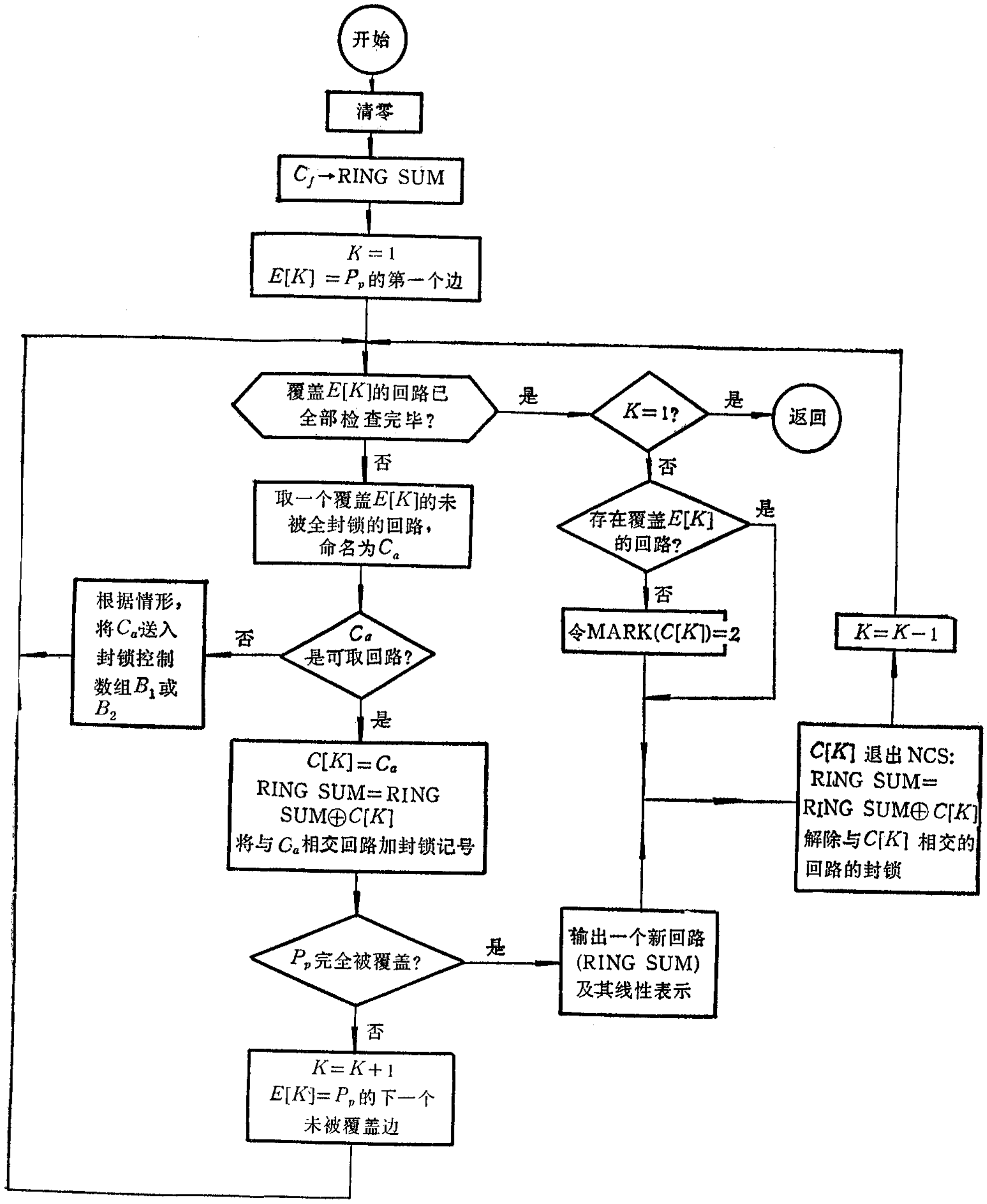


图4　过程 GENERATION 流程图

情形 B　C_a的覆盖 $E[K]$ 的连续边序列包含 $C[K-1]$ 的有效段；

情形 C　C_a的覆盖 $E[K]$ 的连续边序列不是 C_a 的有效段，即在此边序列的前面，C_a 和 P_p 有公共顶点；

情形 D　C_a的有效段未覆盖 P_p 的最末一条边，但 C_a 却通过顶点 w；

若属以上情形之一，C_a是不可取回路.

第 2 步　按照 C_a的边和点集检查是否属于

情形 E　C_a在 P_p 以外，与回路集中已找到的任何一个回路有相交顶点，

若是，C_a 是不可取回路.

在执行第一步(情况B)时,可以同时找到 P_p 的下一个未被覆盖的边.

过程 GENERATION 的流程图中有两个闭环，隐含有无效搜索．为限制无效搜索的重覆,可采用下述删节技术．对每个回路 x 设一个标记 MARK(x)．当 MARK$(x)=0$ 时，回路 x 未被封锁，可选作 NCS 的后补元素；当 MARK$(x)\geq 2$ 时，回路 x 被完全封锁，禁止再次选用；当 MARK$(x)=1$ 时，回路 x 处于半封锁状态，再来一次半封锁就变成全封锁．对每个回路 x 设置二个数组 $B_1(x)$ 与 $B_2(x)$，以控制封锁的实施：若回路 $y\in B_1(x)$，当 x 被接受加入 NCS 时令 MARK(y) = MARK$(y)+1$；若 $y\in B_2(x)$，当 x 被接受加入 NCS 时令 MARK(y) = MARK$(y)+2$．在执行图4的方框 C 时，若经查明 C_a 是不可取回路,则分别作如下处理:

(1) 情形A　设 C_a 与回路 x 和 y 在 P_p 上相交,将 C_a 送入 $B_1(x)$ 和 $B_1(y)$；

(2) 情形B　将 C_a 送入 $B_2(C[K-1])$；

(3) 情形C和D　令 MARK$(C_a)=2$；

(4) 情形E　设 C_a 与回路 x 在 P_p 以外有相交顶点,将 C_a 送入 $B_2(x)$.

四、举　例

图5所示是一个强连通图．实线表示树支，虚线表示连支．并联边用 t 和 c 加以区别．运算过程和结果列于表1．表1的行顺序与 DFS 过程顺序一致．表1的第7和第

表1　图5所示有向图应用本算法的运算过程和结果

1	2	3	4	5	6	7	8
回路集 No.	主导连支	基本回路	正则路径	No.	规范覆盖集	有向回路	有向回路基集的线性组合
1	(4,1)	(4,1)(1,2)(2t3)(3t4)	ϕ	1	ϕ	(4,1)(1,2)(2t3)(3t4)	C_1
2	(5,2)	(5,2)(2t3)(3t4)(4,5)	ϕ	2	ϕ	(5,2)(2t3)(3t4)(4,5)	C_2
3	(3c4)	(3c4)(3t4)	(3t4)	3	$\{C_1\}$	(3c4)(4,1)(1,2)(2t3)	C_3
				4	$\{C_2\}$	(3c4)(4,5)(5,2)(2t3)	$C_3\oplus C_1\oplus C_2$
4	(2c3)	(2c3)(2t3)	(2t3)	5	$\{C_1\}$	(2c3)(3t4)(4,1)(1,2)	C_5
				6	$\{C_2\}$	(2c3)(3t4)(4,5)(5,2)	$C_5\oplus C_1\oplus C_2$
				7	$\{C_3\}$	(2c3)(3c4)(4,1)(1,2)	$C_5\oplus C_1\oplus C_3$
				8	$\{C_4\}$	(2c3)(3c4)(4,5)(5,2)	$C_5\oplus C_2\oplus C_3$
5	(6,5)	(1,2)(2t3)(3t4)(4,5)(1,6)(6,5)	(1,2)(2t3)(3t4)(4,5)	9	$\{C_1, C_2\}$	(6,5)(5,2)(2t3)(3t4)(4,1)(1,6)	C_9
				10	$\{C_3, C_2\}$	(6,5)(5,2)(2t3)(3c4)(4,1)(1,6)	$C_9\oplus C_1\oplus C_3$
				11	$\{C_5, C_2\}$	(6,5)(5,2)(2c3)(3t4)(4,1)(1,6)	$C_9\oplus C_1\oplus C_5$
				12	$\{C_7, C_2\}$	(6,5)(5,2)(2c3)(3c4)(4,1)(1,6)	$C_9\oplus C_3\oplus C_5$

8 列是按照公式 (4a) 和 (4b) 进行环和运算的结果.

有向图有个 5 连支，定义 5 个共主导连支有向回路集，选择 C_1、C_2、C_3、C_5 和 C_9 组成有向回路基集. 值得指出的是：虽然 $\{C_3, C_4\}$ 也覆盖第 5 个基本回路的正则路径，而且 $C_{f5} \oplus C_3 \oplus C_4 = C_{12}$，但 C_4 的有效段是空边集，所以 $\{C_3, C_4\}$ 不是规范覆盖集，C_{12} 不致于重复产生.

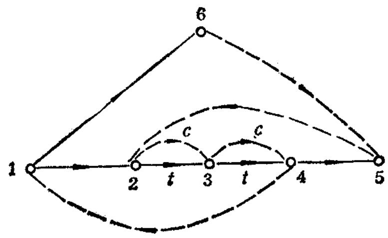


图 5　例图

作者感谢中国科学技术大学研究生院左垲教授、631 研究所李芸芳工程师和本文审阅者，仔细阅读本文初稿并提出许多有益建议.

参　考　文　献

[1] P. Mateti and N. Deo, *SIAM J. Comput.* 5(1976)1, 90.
[2] J. T. Welch, *J. ACM,* 13(1966), 205.
[3] H. T. Hsu and P. A. Honkanen, A fast minimal storage Algorithm for Determing all the Elementary Cycles of a Graph, Computer Sci. Dept., Pennsylvania State Univ., University Park, 1972.
[4] N. E. Gibbs, *J. ACM,* 16(1969), 564.
[5] P. Mateti and N. Deo, On Algorithm for Enumerating all Circuits of a Graph, UIUCDCD-R-73585 (revised), Dept. of Computer Sci., University of Illinois, Urbana, 1973.
[6] Maciej M. Sysl'o, *SIAM J. Comput.,* 10(1981)4, 797.
[7] D. Y. Xiong (熊德琰), Some Properties about Digraph and a Search Algorithm for Finding Simultaneously all Directed Circuits and the Basic Sets, Proceeding of China 1985 International Conference on Circuits and Systems, Ed. by IEAS, pp. 132—135.
[8] 熊德琰，电子学报，1986 年，第 6 期，第 42 页.
[9] 熊德琰，关于生成有向图的全部有向回路的回路向量空间法，中国电机工程学会第一届理论电工学术讨论会，1985 年 3 月.

AN EFFICIENT CIRCUIT VECTOR SPACE ALGORITHM FOR GENERATING ALL DIRECTED CIRCUITS OF A DIGRAPH

Xiong Deyan
(Tongji University)

In this paper, an efficient circuit vector space algorithm is presented for enumerating directed circuits of a directed graph, by which every directed circuit is generated by the ring sum of a fundamental circuit (directed circuit or semi-circuit) defined by a chord of the cotree and a subset of the previously obtained directed circuits. At the same time every directed circuit in the digraph can be represented by a linear combination of a selected basic set of directed circuits.