

文章编号: 0255- 8297(2002) 03- 0258- 05

有向图的同构判定算法: 出入度序列法

李 锋, 商慧亮

(复旦大学 电子工程系, 上海 200433)

摘 要: 提出了有向图的同构判定新算法: 出入度序列法. 该算法可应用于许多可用有向图描述的模式识别等实际问题中.

关键词: 图的同构; 图论算法; 入度序列; 出度序列

中图分类号: TB126 文献标识码: A

An Isomorphism Testing Algorithm for Directed Graphs the In-degree and Out-degree Sequence Method

LI Feng, SHANG Hui-liang

(Department of Electronic Engineering, Fudan University, Shanghai 200433, China)

Abstract In this paper, a new isomorphism testing algorithm for directed graphs—the in-degree and out-degree sequence method—is presented. The algorithm can be used in many pattern recognition problems that can be described by directed graphs.

Key words graph isomorphism, graph theory algorithm, in-degree sequence, out-degree sequence

图的同构判定是图论学科的基本问题之一. 作者在文献 [1] 中对任意无向图的同构判定提出了新的算法——关联度序列法. 本文讨论任意有向图的同构判定算法, 将文献 [1] 的结果从无向图推广到有向图. 本文对讨论的有向图没有任何限制, 即包括平面图与非平面图, 连通图与非连通图, 简单图与非简单图 (含有自环或平行边).

1 有向图的同构概念

为了讨论方便, 先简要介绍一下有向图的同构概念^[2].

定义 1 有向图的同构 如果有向图 G 与 G' 的点之间保持一一对应, 边之间保持一一对应, 而且对应点与对应边之间保持相同的关联关系, 那么称 G 与 G' 同构, 记为 $G = G'$.

定义 1 也可等价地表述为: 如果有向图 G 与 G' 可以写出相同的关联矩阵 A , 则 G 与 G' 同构. 例如图 1 中的两个有向图, 它们的关联矩阵分别是

$$A = \begin{matrix} & \begin{matrix} b_1 & b_2 & b_3 & b_4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & -1 & -1 & 1 \\ -1 & 0 & 1 & -1 \end{bmatrix} \end{matrix}$$

$$A' = \begin{matrix} & \begin{matrix} b'_3 & b'_4 & b'_1 & b'_2 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & -1 & -1 & 1 \\ -1 & 0 & 1 & -1 \end{bmatrix} \end{matrix}$$

因 $A = A'$, 所以 G 与 G' 同构. 需要指出的是, 两个同构有向图, 只有当它们对应的点和对应的边处于关联矩阵行和列的相同位置时, 它们的关联矩阵才会相同. 这一点给我们利用 A 来判定两个有向图是否同构带来了困难. 因为我们事先并不知道哪一点与

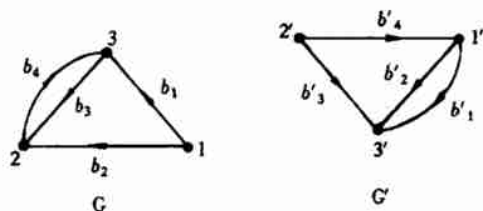


图1 两个同构有向图

哪一点对应, 哪一边与哪一边对应. 一个笨拙的办法是: 先写出 G 和 G' 的关联矩阵 A 与 A' , 然后对 A 进行行与行、列与列的交换, 如能使 $A=A'$, 则判定 $G=G'$; 如所有可能的行、列交换都不能使 $A=A'$, 则判定 $G \neq G'$. 所有可能的行、列交换意味着行的全排列与列的全排列, 所以行、列交换的总次数将达到 $N! \cdot B!$ (N 为图的顶点数, B 为图的边数). 因此, 在最坏情况下判定两个有向图是否同构需要进行 $N! \cdot B!$ 次行、列交换运算, 这是一个比指数时间复杂性还要复杂的算法.

图的同构判定在模式识别等领域有许多实际应用^[3~5]. 因此研究和进一步改进图的同构判定算法, 不仅是一个理论问题, 而且是一个实际问题, 是值得图论学者和算法研究人员探索的课题.

2 有向图的同构判定算法: 出入度序列法

本文将提出一个新的有向图的同构判定算法: 出入度序列法. 为了能使本算法也适用于含有自环的有向图, 我们先对含有自环的有向图做一个“预处理”: 在每一个自环的边上增加一个点, 如图2所示. 这样一个自环就化成了两条有向边, 图中不再存在自环. 我们称这种预处理为自环预处理. 很容易证明: 若对同构图 G_1 与 G_2 做了自环预处理, 所得图 G'_1 与 G'_2 仍然是同构的. 反之, 若 G'_1 与 G'_2 是 G_1 与 G_2 经自环预处理得到的两个同构图, 则 G_1 与 G_2 也是同构的.

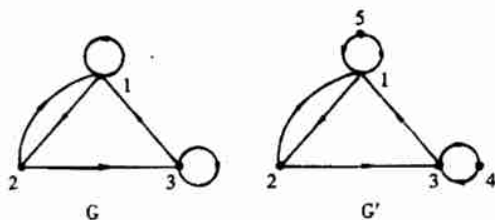
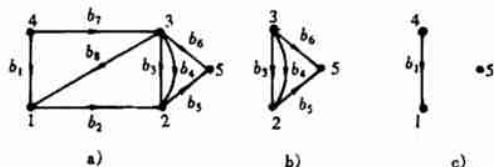


图2 自环预处理图示

有了自环预处理, 我们仅需讨论不含自环的有向图的同构判定问题.

定义2 n 点连通子图 如果有向图 G 中 n ($1 \leq n < N$, N 为 G 的点数) 个点以及连接这 n 个点之间的边组成的子图是连通的, 那么这个子图称为图 G 的 n 点连通子图. 记为 $G(V_n)$, V_n 是这 n 个点的集合.

图3 n 点连通子图举例

例如图3a中, 点2、3、5以及连接于这3点之间的边 b_3, b_4, b_5, b_6 组成的子图3b, 由于它是连通的, 所以该子图是一个3点连通子图, 记为 $G(2, 3, 5)$. 点1、4、5以及连接于这3点之间的边 b_1 组成的子图如图3c所示, 由于它是不连通的, 所以该子图不是3点连通子图.

定义3 n 点连通子图的出入度 在图 G 中, 与一个 n 点连通子图 $G(V_n)$ 相连接的边 (不包括 $G(V_n)$ 的内部边) 称为关联边. 关联边中方向指向这个子图的边称为入边, 方向离开这个子图的边称为出边. 入边的数目称为 $G(V_n)$ 的入度, 记为 $d_i(V_n)$, 出边的数目称为 $G(V_n)$ 的出度, 记为 $d_o(V_n)$.

例如图3a中, 与3点连通子图 $G(2, 3, 5)$ 相连接边有3条: b_2, b_7, b_8 , 所以 $G(2, 3, 5)$ 的关联边是 b_2, b_7, b_8 , 其中入边是 b_2, b_7 , 出边是 b_8 , 所以该子图的入度 $d_i(2, 3, 5)=2$, 出度 $d_o(2, 3, 5)=1$.

定义4 n 点连通子图的出入度序列 若图 G 的 n 点连通子图 G 一共有 P 个, 则将这 P 个 n 点连通子图的出度按从大到小的顺序排列起来所得到的有序集合称为 n 点连通子图的出度序列, 记为 $S_o(n)$; 将这 P 个 n 点连通子图的入度按从大到小的顺序排列起来所得到的有序集合称为 n 点连通子图的入度序列, 记为 $S_i(n)$.

图3a共有5个点, 3点连通子图可能的点的组合是: $(1, 2, 3), (1, 2, 4), (1, 2, 5), (1, 3, 4), (1, 3, 5), (1, 4, 5); (2, 3, 4), (2, 3, 5), (2, 4, 5); (3, 4, 5)$.

经考察, 组合 $(1, 4, 5)$ 与 $(2, 4, 5)$ 不构成3点连通子图, 故舍去.

各连通子图的入度和出度如下:

$$\begin{aligned} d_i(1, 2, 3) &= 2, & d_i(1, 2, 4) &= 3, \\ d_i(1, 2, 5) &= 5, & d_i(1, 3, 4) &= 0 \end{aligned}$$

$$\begin{aligned}d_i(1,3,5) &= 3, & d_i(2,3,4) &= 1, \\d_i(2,3,5) &= 2, & d_i(3,4,5) &= 1 \\d_o(1,2,3) &= 2, & d_o(1,2,4) &= 2, \\d_o(1,2,5) &= 0, & d_o(1,3,4) &= 4 \\d_o(1,3,5) &= 3, & d_o(2,3,4) &= 4, \\d_o(2,3,5) &= 1, & d_o(3,4,5) &= 4\end{aligned}$$

将 $d_i(V_3)$ 按从大到小的顺序排列起来得 3 点连通子图的入度序列:

$$S_i(3) = [5, 3, 3, 2, 2, 1, 1, 0]$$

将 $d_o(V_3)$ 按从大到小的顺序排列起来得 3 点连通子图的出度序列:

$$S_o(3) = [4, 4, 4, 3, 2, 2, 1, 0]$$

定义 5 点的合并 将点 i 与 j 重合并使原接于 i 的边都连接到 j , 再消去这一过程中可能出现的自环. 这样一个操作称为 i 与 j 的合并.

例如, 我们将图 3a 中的点 5 与 2 合并, 先得图 4a 所示的图, 再消去自环 b_5 , 则得到图 4b 所示的合并结果.

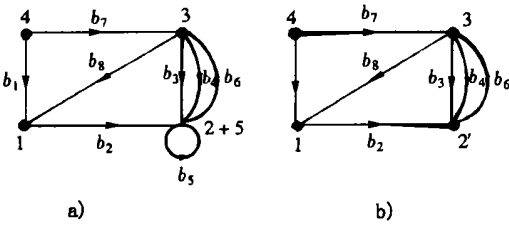


图 4 点的合并图示

点的合并, 可以通过对图的关联矩阵的等价操作来完成. 例如, 图 3a 的关联矩阵为

$$A = \begin{matrix} & \begin{matrix} b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & -1 & -1 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & -1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \end{bmatrix} \end{matrix}$$

如果要将点 5 与点 2 合并, 先将 A 的第 5 行“加”到第 2 行, 即得

$$A_1 = \begin{matrix} & \begin{matrix} b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 \end{matrix} \\ \begin{matrix} 1 \\ 5+2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & -1 & -1 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & -1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{matrix}$$

再删去全 0 的第 5 列 (对应于消去自环 b_5) 即得

$$A_2 = \begin{matrix} & \begin{matrix} b_1 & b_2 & b_3 & b_4 & b_6 & b_7 & b_8 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & -1 & -1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & -1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{matrix}$$

A_2 正是图 4b 中图的关联矩阵.

一般地说, 如果要将图中 n 个点合并, 就是将关联矩阵中这 n 个点对应的行执行加法运算, 且将全 0 的列删去.

如果将 n 点连通子图的 n 个点合并, 则一个 n 点连通子图就收缩为一个“新点”. 显然, 这个“新点”的入度和出度就是这个 n 点连通子图的入度和出度, 例如图 3a 中 $G(2,5)$ 的入度 $d_i(2,5)=4$, 出度 $d_o(2,5)=0$. 而点 5 与 2 合并后, $G(2,5)$ 就收缩为新点 $2'$, 如图 4b, 新点 $2'$ 的入度 $d_i(2')=4$, 出度 $d_o(2')=0$, 显然 $d_i(2,5)=d_i(2')$, $d_o(2,5)=d_o(2')$.

定理 1 设 G_1 与 G_2 是同构有向图, 如果 G_1 中 n 个点与 G_2 中 n 个点一一对应, 那么这 n 个点合并后所得的新图 G'_1 与 G'_2 仍是同构图, 且原有的点之间仍保持一一对应, 合并后所得的新点之间也保持互相对应.

证明 设图 G_1 与 G 的对应点、对应边处在相同的行、列位置上, 则 G_1 与 G_2 有相同的关联矩阵, 即 $A_1=A_2$. 因为 G_1 中 n 个点与 G_2 中 n 个点一一对应, 所以这 n 个点对应着 A_1, A_2 中相同的行. 而 n 个点合并即是将这 n 个点对应的行执行加法运算, 且删去全 0 的列. 因执行的是相同的运算, 所以运算结果有 $A'_1=A'_2$. A'_1 对应着新图 G'_1 , A'_2 对应着新图 G'_2 , 所以 G'_1 与 G'_2 是同构图. 在 A'_1 与 A'_2 中, 原有的对应行未变, 合并后新点对应的行是 n 行加法运算产生的新行, 所以, 对新图 G'_1 与 G'_2 而言, 原有的点之间保持一一对应, 合并所得新点之间保持互相对应.

定理 2 设 G 与 G' 是同构有向图, 如果 G 中有一个 n 点连通子图的入度为 $d_i(V_n)$, 出度为 $d_o(V_n)$, 那么, G' 中必有一个 n 点连通子图, 其入度和出度相同, 即 $d'_i(V_n)=d_i(V_n)$, $d'_o(V_n)=d_o(V_n)$.

证明 因为 G 与 G' 同构, 所以 G' 中必有 n 个点与 G 中的 n 个点一一对应, 因为 G 中的 n 个点可构成连通子图, 所以 G' 中的这 n 个点也必定构成连通子图. 根据定理 1, 这 n 个点合并以后所得的两个新图是同构的, 且合并所得的新点之间保持互相对应. 设这两个新图中的新点分别是 P 与 P' , 那么有: $d_i(P)=d'_i(P')$, $d_o(P)=d'_o(P')$. 由前分析知, P 点的入度和出度等于 G 中 n 点连通子图的入度和出

度, 即 $d_i(V_n) = d_i(P)$, $d_o(V_n) = d_o(P)$. P' 点的入度和出度等于 G 中 n 点连通子图的入度和出度, 即 $d'_i(V_n) = d'_i(P')$, $d'_o(V_n) = d'_o(P')$. 所以有 $d'_i(V_n) = d_i(V_n)$, $d'_o(V_n) = d_o(V_n)$.

定理 3 如果有向图 G 与 G' 同构, G 与 G' 的 n 点连通子图的入度和出度序列分别为 $S_i(n)$, $S_o(n)$, 和 $S'_i(n)$, $S'_o(n)$, 那么对任一 n 都有 $S_i(n) = S'_i(n)$, $S_o(n) = S'_o(n)$, $n = 1, 2, \dots, N-1$. 这里 N 是图 G 与 G' 的点数.

证明 设 $n = k$, $k \in (1, 2, \dots, N-1)$. 因为 G 与 G' 同构, G 中 k 点连通子图的总数应与 G' 中 k 点连通子图的总数相同, 所以 $S_i(k)$ 与 $S'_i(k)$ 中元素的总数相同, $S_o(k)$ 与 $S'_o(k)$ 中元素的总数相同. 另外, 根据定理 2, 图 G 中任何一个 k 连通子图对应着图 G' 中入度和出度与之相同的 k 连通子图, 反之亦一样. 这样 $S_i(k)$ 中的元素与 $S'_i(k)$ 的元素一一对应. 最后, 由于 $S_i(k)$ 与 $S'_i(k)$ 和 $S_o(k)$ 与 $S'_o(k)$ 中的元素排列都是按从大到小顺序. 所以 $S_i(n) = S'_i(n)$, $S_o(n) = S'_o(n)$.

举一例子. 图 5 中的 G 与 G' 实际上是两个同构图, 由于它们点的标号及画法不同, “形状” 显得很不相同. 分别求图 G 与 G' 的出、入度序列, 结果列于表中.

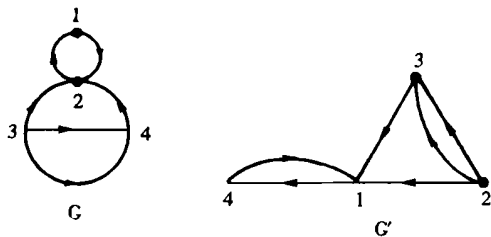


图 5 两个同构有向图

从表中可见 G 与 G' 对应的 $S_i(n)$ 都相等 ($n = 1, 2, 3$), 对应的 $S_o(n)$ 也都相等 ($n = 1, 2, 3$). 因为本例

表: G 与 G' 出入、度计算结果

G					G'				
$d_i(V_1)$	$d_i(1)=1$	$d_i(2)=3$	$d_i(3)=0$	$d_i(4)=2$	$d_i(1)=3$	$d_i(2)=0$	$d_i(3)=2$	$d_i(4)=1$	
$d_o(V_1)$	$d_o(1)=1$	$d_o(2)=1$	$d_o(3)=3$	$d_o(4)=1$	$d_o(1)=1$	$d_o(2)=3$	$d_o(3)=1$	$d_o(4)=1$	
$d_i(V_2)$	$d_i(1,2)=2$	$d_i(2,3)=2$	$d_i(2,4)=4$	$d_i(3,4)=0$	$d_i(1,2)=2$	$d_i(1,3)=4$	$d_i(1,4)=2$	$d_i(2,3)=0$	
$d_o(V_2)$	$d_o(1,2)=0$	$d_o(2,3)=3$	$d_o(2,4)=1$	$d_o(3,4)=2$	$d_o(1,2)=3$	$d_o(1,3)=1$	$d_o(1,4)=0$	$d_o(2,3)=2$	
$d_i(V_3)$	$d_i(1,2,3)=1$	$d_i(1,2,4)=3$	$d_i(2,3,4)=1$		$d_i(1,2,3)=1$	$d_i(1,2,4)=1$	$d_i(1,3,4)=3$		
$d_o(V_3)$	$d_o(1,2,3)=2$	$d_o(1,2,4)=0$	$d_o(2,3,4)=1$		$d_o(1,2,3)=1$	$d_o(1,2,4)=2$	$d_o(1,3,4)=0$		
$S_i(1)$	[3, 2, 1, 0]				[3, 2, 1, 0]				
$S_o(1)$	[3, 1, 1, 1]				[3, 1, 1, 1]				
$S_i(2)$	[4, 2, 2, 0]				[4, 2, 2, 0]				
$S_o(2)$	[3, 2, 1, 0]				[3, 2, 1, 0]				
$S_i(3)$	[3, 1, 1]				[3, 1, 1]				
$S_o(3)$	[2, 1, 0]				[2, 1, 0]				

中顶点数 $N=4$, 所以 $n=1, 2, 3$. 这个例子说明定理 3 是正确的.

定理 3 揭示了同构有向图的重要拓扑性质. 实际上, 定理 3 的逆定理也是成立的, 即如果两个图 G 与 G' 的 n 点连通子图的出入度序列都相等, 即 $S_i(n) = S'_i(n)$, $S_o(n) = S'_o(n)$, $n = 1, 2, \dots, N-1$, 那么 G 与 G' 同构. 这是因为入度和出度序列 $S_i(n)$ 和 $S_o(n)$ 反映了各个子图层次上的点与边的关联关系. 如果 G 与 G' 在结构上有稍许细微的差别, 都将造成某一层或某些层次的 $S_i(n) \neq S'_i(n)$ 和 (或) $S_o(n) \neq S'_o(n)$, 这与假定条件相矛盾.

定理 3 及其逆定理可以作为图的同构判据: 如果 $S_i(n) = S'_i(n)$, $S_o(n) = S'_o(n)$, $n = 1, 2, \dots, N-1$, 则判定 $G = G'$, 只要有一个 $S_i(n) \neq S'_i(n)$ 或 $S_o(n) \neq S'_o(n)$, 则判定 $G \neq G'$.

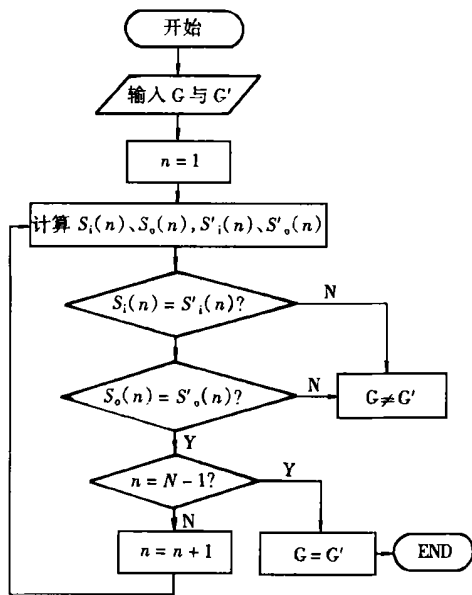


图 6 同构判定算法框图

$S'_i(n)$, 则判定 $G \neq G'$. 据此我们可得到有向图的同构判定算法框图. 如图 6 所示.

下面我们举一例子, 用本算法判定图 7 中 G 与 G' 是否同构.

先计算 G 与 G' 的 $S_i(1)$ 、 $S_o(1)$ 和 $S'_i(1)$ 、 $S'_o(1)$:

$$S_i(1) = [2, 2, 1, 1, 1, 1]$$

$$S_o(1) = [2, 2, 1, 1, 1, 1]$$

$$S'_i(1) = [2, 2, 1, 1, 1, 1]$$

$$S'_o(1) = [2, 2, 1, 1, 1, 1]$$

$$S_i(1) = S'_i(1), S_o(1) = S'_o(1)$$

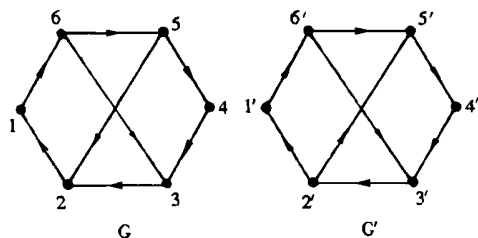


图 7 两个不同构的有向图

再计算其 $S_i(2)$ 、 $S_o(2)$ 和 $S'_i(2)$ 、 $S'_o(2)$:

$$S_i(2) = [3, 2, 2, 2, 2, 1, 1]$$

$$S_o(2) = [3, 2, 2, 2, 2, 2, 1, 1]$$

$$S'_i(2) = [2, 2, 2, 2, 2, 2, 1, 1]$$

$$S'_o(2) = [2, 2, 2, 2, 2, 2, 1, 1]$$

因 $S_i(2) \neq S'_i(2)$, $S_o(2) \neq S'_o(2)$, 故判定 $G \neq G'$.

3 算法分析

下面我们粗略地估计一下该算法的时间复杂性. 本算法的主要步骤是计算 n 点连通子图出入度 $S_i(n)$ 和 $S_o(n)$, 当 $n=i$ 时, i 点连通子图可能的最大数目为 C_N^i (N 为图的点数), 当 n 从 1 变到 $N-1$ 时, 全部连通子图的最大数目为 $\sum_{i=1}^{N-1} C_N^i$, 而 $\sum_{i=1}^{N-1} C_N^i \approx 2^N$, 可见, 利用本算法判断两个图同构时, 在最坏情况下, 其计算时间复杂性仍为指数的, 但是, 该算法

与第 1 部分介绍的关联矩阵行、列交换算法相比较, 有不少改进. 因为关联矩阵行、列交换算法的时间复杂性是 $N! \cdot B!$, 而 $N! \cdot B! \approx (N/e)^N \sqrt{2\pi N} (N/e)^B \sqrt{2\pi B}$, 当 N, B 稍大时, (例如, $N \gg 2e, B \gg 2e$ 时) $2^N \ll (N/e)^N \sqrt{2\pi N} \cdot (N/e)^B \sqrt{2\pi B}$. 可见, 本算法确实要比行列交换算法好许多.

需要指出的是, 利用本算法判断两个图不同构时, 所需的计算时间更要少得多. 因为我们只要找到一个 $S_i(k) \neq S'_i(k)$, 或 $S_o(k) \neq S'_o(k)$, 就可以判定两个图不同构, 并不需要计算所有的 $S_i(n)$ 和 $S_o(n)$.

此时连通子图的最大数目仅为 $\sum_{i=1}^k C_N^i$, 因 $k \ll N$, 所以计算时间复杂性远远低于 2^N (与多项式时间复杂性接近). 但是, A 的行、列交换算法只有完成所有的行列交换仍未找到 $A_1 = A_2$ 时才能判定两个图不同构, 其时间复杂性仍为 $N! \cdot B!$. 因此, 本算法在判定两个图不同构时非常有效.

最后指出, 本算法还是适合并行计算的算法, 因为计算每一个 $S_i(n)$ 和 $S_o(n)$ 都是可独立进行的, 如果使用并行计算机系统来实现本算法, 计算速度将有更大的提高.

本算法可应用于许多用有向图描述的模式识别等实际问题中.

参考文献:

- [1] 李锋. 图的同构判定算法: 关联度序列法及其应用[J]. 复旦学报(自然科学版) 2001, 40(3): 318-325.
- [2] 陈树柏, 左垲, 张良震. 网络图论及其应用[M]. 北京: 科学出版社, 1982. 176-181.
- [3] 李锋, 包闻亮. 一种汉字签名双重识别方法[J]. 模式识别与人工智能, 1998, 11(1): 67-74.
- [4] 李锋, 周新伦. 甲骨文自动识别的图论方法[J]. 电子科学学刊, 1996, 18: 41-47.
- [5] Li Feng, Woo Peng-Yung. Chinese signature verification: the topological approach and the waveform matching approach [J]. International Journal of Imaging Systems and Technology, 1999, 10(4): 355-361.