

在 7 级混洗交换网络中实现 16×16 的可重排性

戴 浩¹, 沈孝钧²

(1. 中国电子设备系统工程公司研究所, 北京 100036

2. 密苏里州立大学堪萨斯分校计算机与工程学院, MO 64110, 美国)

摘 要: 长期以来, 人们猜想 $(2n-1)$ 级的均匀混洗交换网络 Ω 对置换 $2^n \times 2^n$ 是可重排的. 若干论文企图从理论上给出其充分性证明, 但都没有成功, 包括最近的一次证明^[24], 仍然是错误的, 但还没有人指出. 本文的目的之一是澄清这一点. 当 $n=3$ 时已有学者给出了证明^[1, 2]. 本文针对 $n=4$ 时的 7 级 Ω 网络, 给出了实现 16×16 可重排性的构造性证明. 论文提出了避免内部冲突的平衡树模型, 置换的连接图、回路图表示和对称图形、同解变换等概念, 并基于图形压缩、图形剖分等方法, 将 16×16 置换分为五种情况, 共给出五种赋值算法. 这些算法比较简洁, 易于编程实现. 本文提出的思想对研究高阶网络的可重排性也有一定参考价值.

关键词: 多级互连网络; 混洗交换网络; 内部冲突; 可重排性; 同解变换

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112(2007)10-1875-11

Rearrangeability of the 7-Stage 16×16 Shuffle Exchange Network

DAI Hao¹, SHEN Xiaojun²

(1. Institute of China Electronic Equipment System Engineering Company, Beijing 100036, China;

2. School of Computing and Engineering, University of Missouri-Kansas City, Kansas City, MO 64110, USA)

Abstract: It is a long outstanding conjecture that any $(2n-1)$ -stage shuffle exchange network (Ω network) is rearrangeability for $2^n \times 2^n$. Since then many researchers have attempted to solve this conjecture without success. A new result on the sufficiency proof has been established by Hasan, however, nobody have pointed out this proof to be incorrect. To clarify this fact is one of the objectives of this paper. The case $n=3$ was proved by many researchers. Using a constructive approach, this paper also proves that when $n=4$ the 7-stage 16×16 shuffle exchange network is rearrangeability. The model of the balanced tree to avoid internal conflict, the repetition of permutations using connective graph and cycle graph, the concept of symmetry graph and equivalent transform are also presented in this paper. Based on the graph composition and bipartition the permutations 16×16 are divided into five classes, and total five assignment algorithms are proposed. These algorithms are more simple and clear and to be programmed. The techniques used for $n=4$ may provide useful hints for general case $n > 4$.

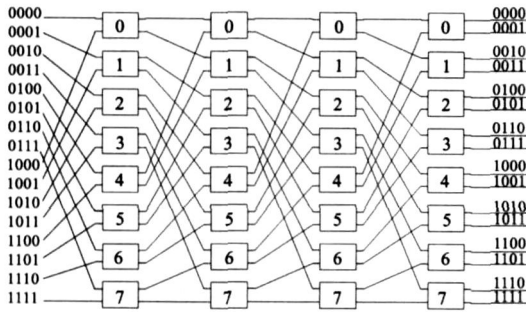
Key words: multistage interconnection network (MIN); shuffle exchange network (SEN); internal conflict; rearrangeability; equivalent transform

1 引言

1971 年由 Stone 提出的混洗交换 (SE: Shuffle/Exchange) 网络在并行处理机和各类通信交换机中得到了广泛的研究和应用^[3]. 输入输出各为 $N=2^n$ 线 (简记为 $N \times N$) 的混洗交换网络通常是不低于 n 级 (阶) 的多级互连网络 MIN (Multistage Interconnection Network), 其中每级有 $N/2$ 个 2×2 的基本交换单元 (Switch Element), 又称开关. 每个开关只能是两种基本状态之一: 直通 (through) 或交叉 (cross), 以实现交换功能; 相邻两级开

关间的连线体现了均匀混洗 (Perfect Shuffle) 的特征^[3]. Ω 网络 (简记为 Ω) 是均匀混洗交换网络的典型代表^[4]. 在 n 级 Ω 网络中, 一对入出线之间不仅存在唯一的路由, 而且具有自路由 (self routing) 能力, 即根据输入输出地址的比特串, 可确定开关的状态. 图 1 给出了入出各为 16 线的 4 级 Ω 网络拓扑结构的表示方法.

在多级互连网络中, 可以将 N 条入出线地址间的一一对应关系, 看作是 N 个自然数的一组置换 (Permutation). N 个数的置换共有 $N!$ 种. 在消除了出线冲突后, N 条入线同时通过 Ω 网络时, 仍有可能存在内部阻

图1 Ω 网络拓扑结构表示方法

塞(blocking)或冲突(conflict).可以证明,对于某些置换来说,例如等同置换或线性置换^[5],能在 Ω 网络中找到 N 条无冲突的路由.然而,这类置换在 $N!$ 种置换中仅占很少一部分.对 $N!$ 种置换均无内部冲突的多级混洗交换网络称为可重排列网络(Rearrangeable Network)或可置换网络.

在 n 级 Ω 网络中,实现可重排的方法有两种.一是将 N 条入线分批,每批入线能无阻塞地一次性地通过 Ω 网络.这种办法归结为寻求最少批数^[6].二是增加 Ω 网络的级数,使不同去向的 N 个信元均能无阻塞地一次性地通过.此时每对入出线通过网络的路由不再惟一,从而有可能避开 n 级 Ω 网络中难以避免的内部冲突.这种办法归结为寻求最小级数问题.

随着集成电路技术的进步及元器件成本下降,人们更多地关注后一种解决办法.30多年来,许多学者致力于这方面的研究^[7~12].1975年,Benes猜测 $(2n-1)$ 级混洗交换网络 Ω 是实现可重排列的充要条件^[13].猜测的必要性很容易证明,而猜测的充分性证明则相当困难.虽然有人用扩展回路算法证明了: $(2n-1)$ 级的Benes网络能保证其重排列性^[15],但 $(2n-1)$ 级 Ω 网络是否为可重排列的充分条件,长期以来不得而知.人们曾尝试用布尔代数、有限状态机模型、图论、矩阵等方法解决这个难题,并先后取得 $2n^2$, $\left\lfloor \frac{3}{2}n^2 - \frac{n}{2} \right\rfloor$, n^2 ,

$3n$, $(3n-1)$, $(3n-3)$, $(3n-4)$ 等较佳下界^[3,7,9,14,16~19],但仍然没有达到 $(2n-1)$ 的极限.有人曾运用拓扑等价的方法证明各类SE网络的同构性^[20~22],但后来发现证明过程中有错^[5,23].论文^[24]证明 $(2n-1)$ 级 Ω 网络可重排的思路是:构造 $N \times (2n-1)$ 阶平衡矩阵集,并从中逐一筛选符合特定模板(frame)要求的矩阵.但论文没有证明对任一置换经过筛选后的平衡矩阵集非空,也没有给出构造性算法实例.从工程角度出发,需要研究一种简捷高效的路由算法.

当 $n=3$ 时,早在1987年就有了5阶 Ω 网络可重排的构造性算法^[1,2,8],从而证明了猜想的充分性,但以

后一直没有突破性进展.文献[5]报导曾用计算机程序验证了 $n=4$ 的各种情况,但没有给出分析性证明.有人用分类方法求解 $n=4$ 的问题,但划分的类别太多,算法过于繁琐.针对 $n=4$ 时的7级 Ω 网络,本文提出一种可重排的构造性算法.它借助于置换的图形表示及其等效变换原理,将 $16!$ 种置换归结为5大类,对每类置换给出相应的路由算法.

论文的其余部分是这样组织的.第2节介绍置换的矩阵表示、转移矩阵概念、内部冲突的窗口检测法和平衡树模型等基本知识.第3节介绍置换的图形表示法、连接图的压缩与分解、等效变换和置换的分类.第4节介绍置换的对称性和两种基本的赋值算法.第5节是本文的核心,主要介绍同解变换概念、回路图表示法和三种基于同解变换的赋值算法.在最后的结束语中,总结了连接图或回路图的判分原则,并对任意级 Ω 网络的可重排性作了初步探讨.

2 基本知识

2.1 置换的矩阵表示

$$P = \begin{pmatrix} S_1 & S_2 & S_3 & S_4 & D_1 & D_2 & D_3 & D_4 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} S_1 & S_2 & S_3 & S_4 & D_1 & D_2 & D_3 & D_4 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

图2 置换的矩阵表示法

本文用四位二进制数分别表示输入地址 $S = S_1S_2S_3S_4$,输出地址 $D = D_1D_2D_3D_4$,对应的一组置换可以表示为 16×8 的矩阵 $P = [S_1 S_2 S_3 S_4 D_1 D_2 D_3 D_4]$,矩阵中每一个元素均为比特值0或1,故又称 $\{0,1\}$ 矩阵.矩阵中的行自上而下编号为0到15.矩阵 P 中任意两行位置对换后,仍表示同一置换.图2为同一置换的两种矩阵表示形式:一种是按输入地址的逆向次序排列,即 $S_4S_3S_2S_1$ 从0000,0001,0010……一直到1111;另一种是按输出地址的正常次序排列,即 $D_1D_2D_3D_4$ 从0000,0001,0010……一直到1111.

2.2 内部冲突的检测

设 Ω 网络的级从1开始,从左到右顺序编号(图3).在4级 Ω 网络中,路由由算法决定了各级开关的状

态. 对给定一组由入出地址组成的置换来说, 第 i 级开关的状态取决于 $S_i \oplus D_i$ 值. 当输入地址中第 i 位与输出地址中的第 i 位相同时, 开关为直通状态, 否则为交叉状态. 或者说, 输出地址 D_i 的比特值决定了第 i 级开关的输出, 通常 0 表示从开关的上端输出, 1 表示从开关的下端输出. 所谓内部冲突, 就是进入开关的两条入线要求有相同的出口, 或者进入开关的同一入线要求有不同的出口, 从而使开关处于既要直通又要交叉的状态. 例如图 3 所示的 Ω 网络中同时有两对连接: $(0010 \rightarrow 0111)$, $(1010 \rightarrow 0001)$. 他们通过第 1 级开关后, 均要求从输出地址 0100 进入下一级. 显然, 这两条连接是无法同时建立的.

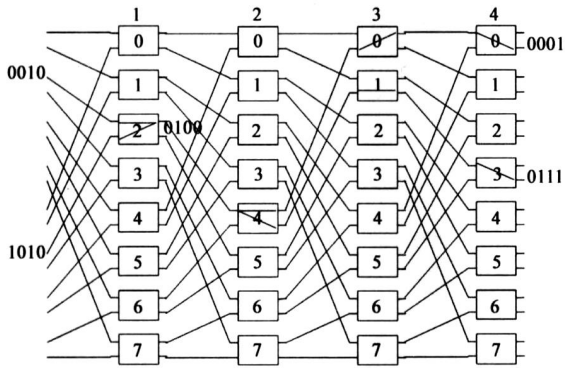


图 3 两条路由的内部冲突

为了检查多条连接的路由是否冲突, 通常采用窗口检验法. 在 16×8 的置换矩阵 P 中, 将相邻的 4 列子矩阵看作一个窗口 (Window). 如果窗口中逐行的二进制串各不相同, 换言之, 如果是 0000 到 1111 的全排列, 则对应的这级开关就不会产生冲突, 因为它们不再有不同的输出地址. 由于第一个窗口和最后一个窗口分别是输入地址和输出地址, 各行的二进制串肯定不同, 所以我们只需检查中间的 3 个窗口即可. 一般来说, 由输入地址和输出地址组成的置换矩阵 P , 中间 3 个窗口很难做到全排列.

2.3 置换的转移矩阵

如将 4 级 Ω 网络增加一级, 对某个连接来说, 可供选择的途径就增加一倍; 增加两级, 可选路径就增加到 4; 增加三级, 可选路径就增加到 8, 所以扩展后的 7 级 Ω 网络有可能避免内部冲突, 使 16 条路由一次性地通过. 此时共有 $8 \times 7 = 56$ 个开关, 每个开关有 2 种状态, 总共可形成 $2^{56} = 6.4 \times 10^{16}$ 量级的解空间, 远大于 $16! = 2.1 \times 10^{13}$ 量级的问题空间. 关键在于如何为连接寻找一条确定的、互不冲突的路由, 或者说, 要确定路径上逐级开关的输出地址. 为此需要将置换矩阵 $P[16, 8]$ 扩展成转移矩阵 $T[16, 11]$, 即在中间增加 3 列, 记作 $[XYZ]$, 于是有:

$$T = [S_1, S_2, S_3, S_4, X, Y, Z, D_1, D_2, D_3, D_4]$$

如果 $T[16, 11]$ 能通过窗口检验, 即其中任何连续 4 列的子矩阵 (窗口) 中逐行的比特串各不相同, 则 16 条路由在 7 级 Ω 网络中必定无冲突^[18]. 这样, 可重排性问题归结为如何填写 XYZ 各列中的比特值, 即确定 16 个三位的二进制数, 每个数字恰好出现两次. 对于某特定连接来说, 相当于在每行输出地址串前增加 3 位地址, 又称路由前缀或扩展的输出地址. 一条路由由中各级开关的状态取决于 $T_{ij} \oplus T_{i,j+4}$ ($0 \leq i \leq 15, 1 \leq j \leq 7$) 的值.

2.4 无冲突的平衡树模型

为保证转移矩阵 T 中间的 6 个窗口 $[S_2S_3S_4X]$, $[S_3S_4XY]$, $[S_4XYZ]$, $[XYZD_1]$, $[YZD_1D_2]$ 和 $[ZD_1D_2D_3]$ 均为全排列, 即每行的二进制数值各不相同, 我们给出多维地址空间的概念, 并对 XYZ 三列的比特值有如下要求:

- 一维空间左平衡: 当 $S_2S_3S_4$ 相同时, 对应的 X 不同;
 - 二维空间左平衡: 当 S_3S_4 相同时, 对应的 XY 不同;
 - 三维空间左平衡: 当 S_4 相同时, 对应的 XYZ 不同.
 - 一维空间右平衡: 当 $D_1D_2D_3$ 相同时, 对应的 Z 不同;
 - 二维空间右平衡: 当 D_1D_2 相同时, 对应的 YZ 不同;
 - 三维空间右平衡: 当 D_1 相同时, 对应的 XYZ 不同.
- 满足上述条件的 XYZ 称之为合理赋值. 如将 T 按上下左右分成 4 个象限, 每个象限表示一个三维地址空间, 则可用平衡树模型表述这些要求. 图 4 和图 5 给出的 2 棵树分别表示第 II 象限 ($S_4 = 0$) 和第 I 象限 ($D_1 = 0$) 的平衡条件, 每棵树有 3 层. 另两棵平衡树的结构类同.

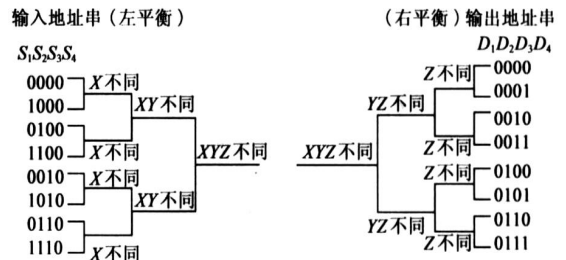


图 4 $N=16$ 无冲突树型模型之一 图 5 $N=16$ 无冲突树型模型之二

混洗交换网络具有一题多解的特点. 例如, 如果向量组 XYZ 满足平衡树的要求, 那么将 X 、 Y 、 Z 独立或同时取反, 形成新的向量组 $\bar{X}YZ$ 、 $X\bar{Y}Z$ 、 $\bar{X}\bar{Y}Z$ 、 $X\bar{Y}\bar{Z}$ 、 $X\bar{Y}Z$ 、 $\bar{X}YZ$ 也是合法解.

3 连接图的分解和置换的分类

3.1 置换的图形表示

可以用正则的二部图表示任一 16×16 置换, 左列节点用二进制的输入地址表示, 右列节点用二进制的

输出地址表示, 节点之间的连线(边)表示一对置换. 每条边都对应合理的 XYZ 值. 图 6(a) 为前述置换 P 对应的二部图.

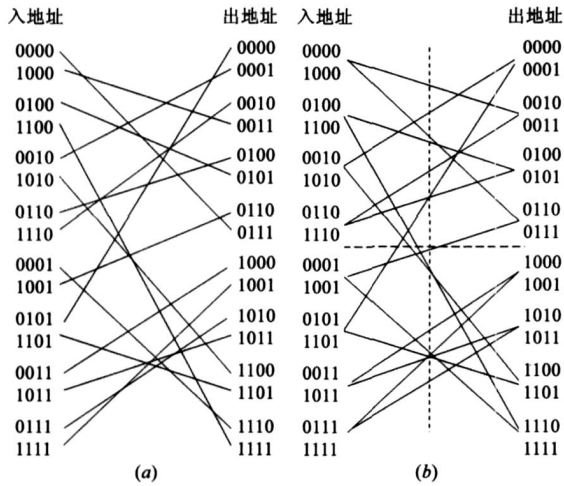


图 6 置换对应的二部图及压缩后的连接图

可以对二部图作适当压缩, 将属于一维空间的节点($S_2S_3S_4$ 相同或 $D_1D_2D_3$ 相同) 合二而一, 形成左右各有 8 个节点的连接图, 见图 6(b), 并记为 G 或 $G(16)$, G 中每个节点的秩(度)为 2. 这样, 汇聚在同一节点的两条边属于相同的一维空间. 属于相同二维空间(S_3S_4 或 D_1D_2 相同) 的两个节点保持上下相邻的关系. G 图和二部图表示了同一组置换 P , 以后简称 G 为 P 的连接图, 并可用笛卡尔坐标轴将 G 图分成 4 个象限.

定义 1 输入/输出地址在第 III/第 I 象限的边或在第 II/第 IV 象限的边称之为交叉边. 输入/输出地址分别分别在第 II/第 I 象限或分别分别在第 III/第 IV 象限的边称之为非交叉边.

推论 1 第 III/第 I 象限的交叉边和第 II/第 IV 象限的交叉边数量相同.

推论 2 G 图中交叉边的对数 $m \leq 8$.

3.2 连接图的等效变换

混洗交换网络除一题多解外, 还具有多题一解的特点. 例如, 利用下述引理 1~ 4 对输入/输出地址进行对换和组合后, 会得到若干新的置换, 但它们具有相同的解, 从而可以将置换归类. 下面是在拓扑同构前提下, 连接图的一些等效变换法则:

引理 1 在连接图中, 属于同一节点(一维空间) 的两个入/出地址可以对换其位置.

引理 2 在连接图中, 属于同一二维空间的两组一维空间的入/出地址可以对换其位置.

引理 3 在连接图中, 属于同一三维空间的两组二维空间的入/出地址可以对换其位置.

引理 4 在连接图中, 属于同侧的上下两个象限(三维空间) 的地址可以成组对换其位置.

根据引理 4 对连接图施行等效变换后, 可以使交叉边与非交叉边对调. 若对调前 G 图有 m 对交叉边, 在对调后则有 $(8-m)$ 对交叉边. 不失一般性, 在以后的讨论中, 我们约定 $m \leq 4$.

引理 5 在连接图中, 对换左右两侧的输入地址和输出地址后, 新的置换关系为原置换之逆(将 XYZ 换为 ZYX).

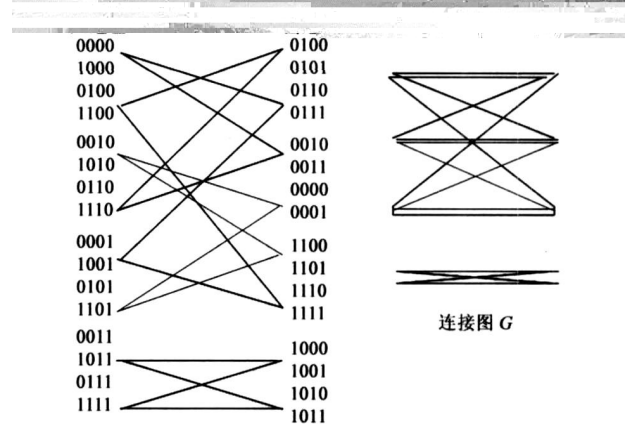


图 7 调整后的连接图及经过二次压缩的连接图

基于上述等效变换法则, 适当调整图 6(b) 连接图的输出地址位置, 尔后再一次进行压缩, 将属于相同二维空间的两个节点合二而一, 形成的正则连接图 G , 如图 7 所示. 根据引理 1~ 4, 在以后的连接图中, 将不再标注节点的二进制地址, 集中研究抽象的 G 图中 16 条边的 XYZ 合理赋值问题.

3.3 连接图的分解

连接图 G 中的 16 条边可以组成一个回路, 也可以组成长度不等的多个回路. 通过沿回路交替取边法, 可将图 $G(16)$ 一分为二, 形成 G_1 和 G_2 . 在 G_1 和 G_2 中, 将属于相同二维空间的边汇聚成在一起, 再次运用沿回路交替取边法, 将 G_1 分成两个 4×4 的二部图 G_{11} 和 G_{12} , 将 G_2 分成两个 4×4 的二部图 G_{21} 和 G_{22} , 每个二部图中有 4 条边, 以下简称 $G_{ij}(i, j = 1, 2)$ 为四元组. 上例中的图 G 有两种判分方法, 对应的 G_1 和 G_2 及其四元组分列于右, 如图 8 所示.

推论 3 设回路长度为 L , 当 $G(G_i)$ 为单回路时, $G_i(G_j)$ 的分法唯一的; 当 $G(G_i)$ 由多条 $L \geq 4$ 的回路组成时, $G_i(G_j)$ 的分法是不唯一的.

推论 4 四元组 G_{ij} 中的交叉边均是成对出现的, 对某个 G_{ij} 来说, 交叉边的对数 ≤ 2 .

从理论上讲, 不同拓扑结构的 G_{ij} 共有 $4! = 24$ 种, 比较典型的有 13 种, 如图 9 所示. 其中有一对交叉边的 G_{ij} 有 7 种. 当四元组 G_{ij} 的交叉边的对数为 0 和 2 时, 其 XYZ 的赋值方法没有什么差异. 因此, 我们主要关注有一对交叉边的 G_{ij} . 以后如无特殊说明, 有交叉边的 G_{ij} 均是指只有一对交叉边的 G_{ij} .

3.4 置换的分类

将图 G 分解成四个 G_{ij} 后, 根据有交叉边的 G_{ij} 个数 k , 可将 G 分成五种情况.

- CASE 0: 有交叉边 G_{ij} 的个数 $k=0$
- CASE 1: 有交叉边 G_{ij} 的个数 $k=1$
- CASE 2: 有交叉边 G_{ij} 的个数 $k=2$
- CASE 3: 有交叉边 G_{ij} 的个数 $k=3$
- CASE 4: 有交叉边 G_{ij} 的个数 $k=4$

注意: 这种分类方法和 G 图中交叉边的对数 (m) 不完全相同, 表 1 列出了两者的对应关系. 表中的“=”, “×”和“#”分别代表 G_{ij} 有 0 对、1 对和 2 对交叉边.

表 1 不同交叉边对数的 G 图在五种类别中的分布

G 中交叉边对数 m	CASE 0	CASE 1	CASE 2	CASE 3	CASE 4
0	=====				
1		×=====			
2	=====		××=====		
3		×=====		×××=====	
4	=====		×××=====		××××=====

下节将说明, 对 CASE 0 和 CASE 1 可直接运用赋值算法 1; 对 CASE 4 可直接运用赋值算法 2. 当 CASE 2 和 CASE 3 的 G 图具有对称性时, 可直接运用赋值算法 1. 当 CASE 2 和 CASE 3 的 G 图为非对称时, 需适当调整交叉边或非交叉边的端接位置, 即实施同解变换后, 再运用赋值算法 1 或 2. 因此, 算法 1 和 2 是基本的赋值算法.

4 基本赋值算法

4.1 置换的对称性

为引入 G 图的对称性概念, 我们将 G 中每个象限分为上下两个分区(每个区是属于相同二维地址空间的各边的端接处), 从上至下记为 A, B, C, D , 左右象限对应位置的区号相同, 如图 10 所示. $G_p, G_{ij}(i, j$

$= 1, 2)$ 中分区的位置与 G 相同.

定义 2 G 的对称性

在 CASE 2 和 CASE 3 中, 如果诸 G_{ij} 的一对交叉边的右(左)端点分别落在 A, D 区和/或 B, C 区(见图 11a), 或分别落在 A, C 区和/或 B, D 区(见图 11b), 则称 G 的诸交叉边端接点分布是右(左)对称. 如果 G 是左对称和/或右对称, 则称 G 中交叉边的分布具有对称性; 否则称 G 为非对称, 即既非左对称又非右对称(见图 11c).

在 CASE 1 中, G 具有天然的对称性. 在 CASE 0 和 CASE 4 中, G 的对称性概念不适用.

根据引理 2~ 引理 5, 可以将 G 进行规范化. 不失一般性, 以后假定对称图 G 具有右对称性, 且交叉边端点成对地落在右侧 A, D 区和/或 B, C 区, 如图 11(a) 所示.

4.2 两个基本的赋值算法

算法 1 (适用于 CASE 2、CASE 3 中对称的 G , 包括 CASE 0 和 CASE 1)

赋值思路如下: X 值按 G_{11}, G_{12} 和 G_{21}, G_{22} 分组的方式赋不同的值, Y 值按 G_{11}, G_{22} 和 G_{21}, G_{12} 分组的方式赋

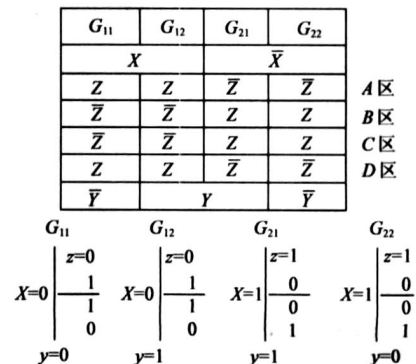


图 12 赋值算法 1 的示意图

不同的值; Z 值按右侧 $A D$ 区、 $B C$ 区的分组方式, 赋以不同的值. 赋值通式如图 12 所示. 图中还给出了一个具体赋值的例子.

注意: 当 G_j 为两对交叉边时, 其赋值方法与没有交叉边的 G_j 完全相同. 从通式中可以看出, 用这种方法为 16 条边赋 XYZ 值, 其结果是不惟一的. 但不论如何赋值, 每个四元组中交叉边的 XYZ 值总是对应相等.

定理 1 用算法 1 对 CASE 0、CASE 1 和其他对称 G 赋值一定能满足平衡树模型.

定理的正确性不难检验.

算法 2 (仅适用于 CASE 4)

四个 G_j 中各边的 XYZ 值如图 13 所示. 图 (a) 是一般的赋值方法, 图 (b) 是一个赋值的特例. 其中 X, Z 值分别由每条边左右侧所在位置决定. Y 的赋值方式同算法 1.

	G_{11}	G_{12}	G_{21}	G_{22}
AB 区	XZ	$\bar{X}Z$		
CD 区	$\bar{X}Z$	XZ		
	Y	\bar{Y}	Y	

(a)

G_{11}	G_{12}	G_{21}	G_{22}
0 0	0 0	1 1	1 1
1 1	1 1	0 0	0 0
$y=0$	$y=1$	$y=1$	$y=0$

(b)

图 13 赋值算法 2 的示意图

定理 2 用算法 2 对 CASE 4 中的 G_j 赋值一定是合理的.

证明: 略.

注意: CASE 4 的 G_j 中, 两条交叉边的 XYZ 值并不相同, G_{11} 与 G_{22} 的两组交叉边是对应相等的. 对 G_{12} 和 G_{21} 来说, 也是如此.

5 基于同解变换的赋值算法

5.1 对同解变换的要求

同解变换是指图 G 在变更拓扑结构后仍具有相同的解空间. 为实现 CASE 2 和 CASE 3 中非对称图的赋值, 需对引理 2~ 引理 4 的等效变换原理加以推广.

引理 6 交换二维空间内两条边的端接位置(同解变换 1)

将 G 中位于相同二维空间(区)但分属不同一维空间的两条边右端接位置对调, 生成新的 G' . 如果 G' 能合理赋值, 并保证对调的两条边有相同的 Z 值, 则 G 也能有合理赋值.

根据引理 6

可以对图形进行“棍棒化”处理. 参见图 14 (a), 当 e_1, e_3 两条边左端接, 右端在不同的

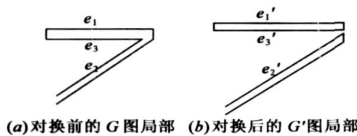


图 14 互换两条边端点形成“棍棒”

换右端接点, 可以将回路一分为二, 形成由 (e'_1, e'_3) 组成的、回路长度为 2 的“棍棒”, 如图 14 (b) 所示. 显然, 当 (b) 图有合理赋值时, “棍棒”中两条边 e'_1, e'_3 的 XYZ 值可以对换, 且 Z 值一定相反, 可以保证 e'_1 边的 Z 值与 e'_2 边的 Z 值相同. 引理 6 表明, “棍棒化”后对 G 图的合理赋值不会产生影响.

引理 7 交换三维空间内两条边的端接位置(同解变换 2)

固定 G 中互不端接两条边的左端点, 将位于同一三维空间, 但分属不同二维空间的右端点位置对调(A, B 区互换, 或 C, D 区互换)后, 生成新的 G' . 如果 G' 能合理赋值, 并保证对调后的两条边有相同的 YZ 值, 则 G 也能有合理赋值.

参见图 15. 当 e_1, e_2 两条边对换右端接位置后, 形成 e'_1, e'_2 .

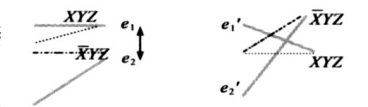


图 15 互换两条边端点位置的条件

如果这两条边的赋值是 XYZ 和 $\bar{X}YZ$, 并能保证图 G' 一维、二维

空间右平衡, 那么将 G 中 e_1, e_2 两条边分别赋值 XYZ 和 $\bar{X}YZ$ 后, 也能保证图 G 一维、二维空间右平衡.

引理 8 交换不同三维空间内两条边的端接位置(同解变换 3)

固定 G 中互不端接两条边的左端点, 仅在不同三维空间之间对换右端接位置(从 I 象限对换到 IV 象限或反之), 生成新的 G' . 如果 G' 能有合理赋值, 且对换端点的两条边有相同的 XYZ 值, 则 G 与 G' 同解, 即 G 也有合理赋值.

引理 6, 7, 8 同样适用于两条边右端点固定, 左端点对换位置的情况.

在以下的算法中, 如无特殊说明, 我们将需搬移端点位置的交叉边和非交叉边分别记作 e_1 和 e_2 . 如果 e_1, e_2 在 G 中属于同一回路, 可用算法 3 或算法 5 对 e_1, e_2 边实施同解变换; 否则, 需用算法 4 对两对边同时进行同解变换.

5.2 回路图与同解变换

定义 3 回路中边的距离

两条边的距离 d 为从一条边的起点开始, 沿回路到达另一条边的起点, 中间所经过的较少边的数量. d 为偶数时称为偶距, 为奇数时称为奇距. 如果将回路图 G 的边二着色, 或交替标以“+”, “-”号, 则回路中同色的两条边为偶距, 异色的两条边为奇距.

鉴于在多回路的连接图中回路相互交错, 难以分清, 为此设计与连接图拓扑等价的回路图, 其中横坐标上四个点代表右侧分区 $ABCD$, 纵坐标上四个点代表

左侧分区 $ABCD$. 对于某个给定的回路, 将连接图中的“边”用回路图中的“点”来表示, 其横坐标为该边在右侧分区的端接位置, 纵坐标为该边在左侧分区的位置; 连接图中的端接点用回路图中的直角边来表示, 回路图中水平连线表示连接图中的左端接点, 垂直连线表示连接图中的右端接点. “棍棒”在回路图中为重合的两个点, 用“:”表示之. 回路图中每行每列刚好有 4 个点.

图 16(a) 给出的回路图与图 7 的连接图等价, 图中有 2 条长度分别等于 8 和 4 的回路, 另有 1 条长度等于 4 的回路集中在右下角的一个坐标点上, 我们用小矩形示之. 同样沿回路交替取“点”, 可将回路图 G 一分为二, 形成 $G_1(8)$ 和 $G_2(8)$. 并和连接图一样, 有两种分法, 如图 16(b) 和 (c) 所示, 它们分别与图 8 上下两个子图 G_1, G_2 相对应.

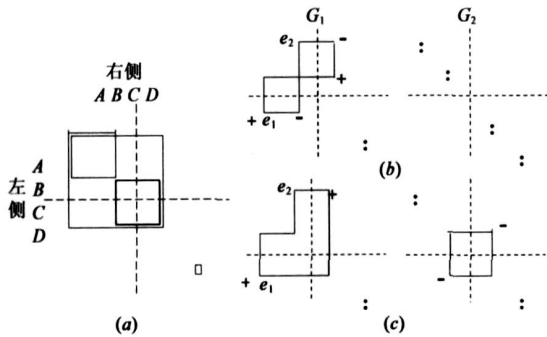


图 16 与连接图等价的回路图

可以用笛卡儿坐标将回路图分为上下左右四部分. 不难看出, 连接图中的交叉边对应回路中 I、II 象限的点, 我们称这些点为“交叉点”; 回路图中 II、IV 象限中的点相当于连接图中的非交叉边, 我们称这些点为“非交叉点”.

推论 5 当回路中 ($L \geq 4$) 偶距的两条边越区对换右(左)端点位置时, 将分裂为两条新的回路, 每个新回路包含其中的一条边.

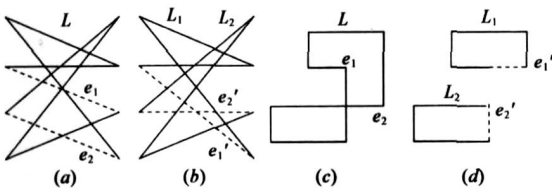


图 17 同解变换后将回路 L 一分为二

图 17(a)、(b) 示例说明了连接图中对换两条右边端接点后回路的变化. 选择同号的边 e_1, e_2 变换右端接点后, 原回路 L 一分为二, 拆成两个较小的回路 L_1 和 L_2 , 其中 $e'_1 \in L_1, e'_2 \in L_2$. 此时各边的标号(着色)不变. 但变换后的两条边处在不同的回路中, e'_1, e'_2 也可着不同的颜色. 图 17(c)、(d) 是对应的回路图的变化.

推论 6 当回路 ($L \geq 6$) 中奇距的两条边越区对换

右(左)端点位置时, 新生成的两条边仍是同回路中奇距, 且保留异着色, 但它们之间的两条路径中, 有一条路径上的各边要变更颜色.

图 18 的示例说明了变换前后回路图的变化. 其中 e_1, e_2 在同一回路中奇距, 交换前后两点的着色不变, 但它们之间在右下角路径上的点变更了颜色.

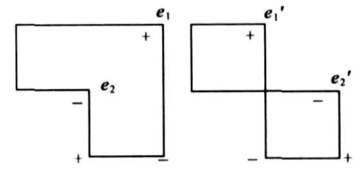


图 18 同一回路中奇距的两点跨区对换

推论 7 位于两个回路中的两条边对换端点位置后, 两个回路合并成一个回路, 这两条边在新的回路中为偶距(同色). 这是推论 5 之逆.

在下面的行文中, 越区或异区对换, 对连接图来说, 指两条边在同象限的两个区中互换端接点位置; 对回路图来说, 指同侧或上半部/下半部的两个点相对平移.

5.3 同回路异区的同解变换

算法 3 (同回路中交叉边和非交叉边的同解变换)

设非对称 G 图中需要更右端接位置的交叉边为 e_1 , 在相同三维空间、但在不同二维空间的非交叉边为 e_2 , 且 e_1, e_2 在 G 的同一回路中 ($L \geq 4$), 则可对换 e_1, e_2 右端接位置, 形成交叉边满足右对称要求的 G' 图, 并与 G 图同解.

step 1: 保持 e_1, e_2 左端点位置不变, 右端点位置对换, 生成新的边记作 $e'_1, e'_2 \in G'$. 如果 e_1, e_2 在 G 中同回路偶距, 根据推论 5, 在 G' 中将会生成两条新回路, 分别包含 e'_1, e'_2 ; 如果 e_1, e_2 在 G 中同回路奇距, 根据推论 6, e'_1, e'_2 在 G' 中仍保持同回路奇距.

step 2: 采用沿回路交替取边法, 将 G' 一分为二, 若 e'_1, e'_2 在 G' 中同回路奇距, 则自动分到 G'_1 和 G'_2 中; 若 e'_1, e'_2 在 G' 中属于不同回路时, 有意识地将它们分到 G'_1 和 G'_2 中.

step 3: 对 G'_1 和 G'_2 实施沿回路交替取边法, 生成右对称的 $G''_{11}, G''_{12}, G''_{21}$ 和 G''_{22} . 不失一般性, 假设 $e'_1 \in G''_{11}, e'_2 \in G''_{22}$.

step 4: 调用算法 1 为 G' 赋值, 并令 G''_{11} 和 G''_{22} 的 Y 值相等.

step 5: G 中未变动的边与 G' 中对应的边赋值相同, $(e'_1) \Rightarrow e_1, (e'_2) \Rightarrow e_2$.

图 16 所示的非对称图 G 可用同解变换来求解, 如图 19 所示. 图中, 交叉点 e_1 在 A 区, 需变更到 B 区, 非交叉点 e_2 在 B 区, 相应变更到 A 区. 变换后的 G' 有两种判分方法, 我们将 e'_1 留在 G'_1 中, e'_2 安排到 G'_2 中, 以保证 e'_1 和 e'_2 能有相同的 YZ 值. 图 19 的右下方给出了

四个四元组的一组赋值. 从中可以看到, e_1' 的 XYZ 值为 000, e_2' 的 XYZ 值为 100, 其 YZ 值相同.

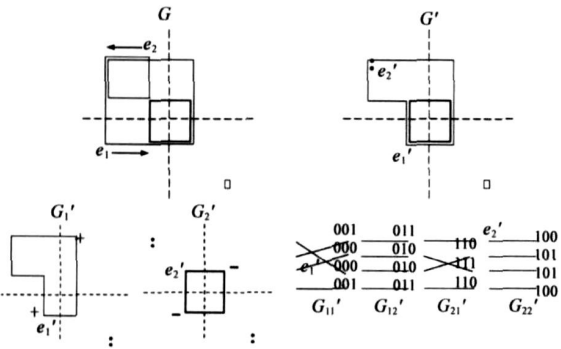


图 19 对换交叉边和非交叉边的右端接位置

定理 3 在 CASE 2 和 CASE 3 中, 如果交叉点 e_1 和非交叉点 e_2 同回路 ($L \geq 4$) 异区, 则非对称图 G 一定有合理的赋值.

证明: 略.

算法 3 通过变更交叉点的分区位置, 实现了非对称图向对称图的转化, 同回路的条件保证了交叉点 e_1' 和非交叉点 e_2' 赋予相同的 YZ 值, 故 e_1, e_2 也有相同的 YZ 值. 又因为 G' 赋值 XYZ 后满足平衡树要求, 根据引理 7, 图 G 的 XYZ 值也一定能满足平衡树要求.

图 20 是图 6 所示例子 XYZ 的最终解. 图中还给出了 7 级 Ω 网络的拓扑结构, 以及每级 8 个开关的状态.

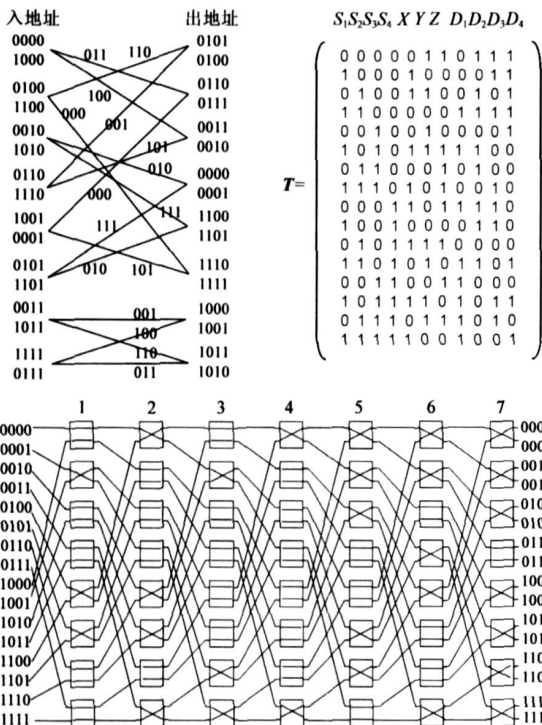


图 20 连接图的解决定了 Ω 网络的路由

5.4 同回路异区的判别准则

当一个交叉点 e_1 需要越区对换位置时, 首先要保

证 e_1 所在的回路中有一个异区的非交叉点 e_2 . 因此, 判定非对称图 G 中是否存在同回路、异区的非交叉点, 是运用算法 3 的前提. 考虑到交叉点(或非交叉点)在回路中总是成对出现的, 于是有下面两个引理.

引理 9 在一条 $L \geq 4$ 的回路中, 如果含有同一象限的两个交叉点, 则在该回路中一定能找到一对偶距、异区的交叉点与非交叉点.

在图 21 中所示的两类回路图中, 均有同象限的两个交叉点和两个非交叉点, 从而能符合算法 3 的要求. 根据引理 9 可得出如下推论.

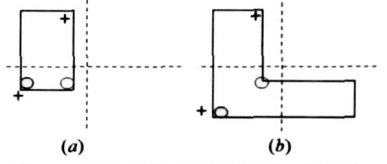


图 21 同象限的交叉点在同一回路中

推论 8 当回路图中某一回路仅有的两个交叉点位于同象限时, 仅当 $L = 2$ 时(棍棒化)时, 在该回路中才找不到与 e_1 对应的偶距、异区的非交叉点 e_2 .

引理 10 在一条 $L \geq 6$ 的回路中, 如果含有分处两象限的两个(或两个以上的)交叉点, 则在该回路中一定能找到一对偶距(或奇距)、异区的交叉点与非交叉点.

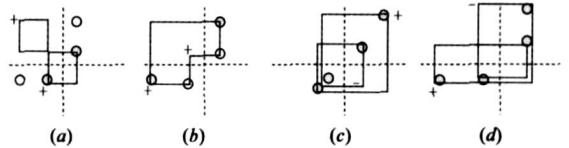


图 22 不同象限的交叉点在同一回路中

图 22(a)、(b) 示意在 $L = 6$ 的回路中, 一对异区的交叉点与非交叉点的分布. 图 22(c)、(d) 所示的是 $L = 8$ 的回路, 其中含不同象限的 4 个交叉点, 还有 4 个非交叉点. 因为非交叉点与交叉点皆为奇距, 所以一定可以在回路中找到一对奇距、异区的交叉点与非交叉点, 从而能符合算法 3 的要求. 根据引理 10 可得出如下推论.

推论 9 当回路图中的回路有两个或两个以上的交叉点位于异象限时, 仅当 $L = 4$ 时, 在该回路中才找不到与 e_1 对应的异区的非交叉点 e_2 .

5.5 异回路的同解变换

判定 G 是否为非对称图, 是运用算法 3 的前提条件之一. 在 CASE 2 时, 当左边和/或右边的两对交叉边集中在 B、C 两个分区(0 2 2 0 型), 或均匀分布在四个分区(1 1 1 1 型)时, G 图是对称的. 只有当交叉边分布为 0 2 1 1 型, 才能为非对称. 于是有如下引理.

引理 11 在 CASE 2 中, 当交叉边的分布如图 23 所示时, 图 G 为非对称.

从上下、左右两个方向看回路图 23, 其中交叉点的分布均呈 0 2 1 1 型. 四个交叉点中任一个进行跨区对换, 都可以使 G 变成对称图.

在图 23 中, 对于选定的交叉点 e_1 来说, 如果找不到同回路异区的非交叉点 e_2 , 根据推论 9, 只有一种可能, 即不同象限的一对交叉点形成了独立的 $L=4$ 回路, 如图 24 所示。

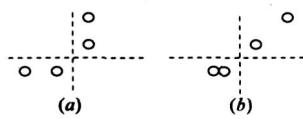


图 23 非对称图 ($K=2$) 交叉点在回路图中的分布

处理图 24 这类特殊的非对称图需要有新的算法。为将 G 变为对称图, 交叉边 e_1 欲变更分区位置, 但异区的非交叉边 e_2 属于另一回路, 此时需要运用推论 7 和推论 5, 先交换异回路中 e_1 和 e_2 端点的位置, 再交换新回路中另一对偶距的非交叉边的端点位置, 使 e_1 和 e_2 仍在异回路中, 即需要同时变更两个回路中的两对边。

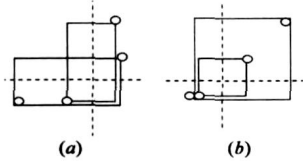


图 24 需特殊处理的非对称图 (CASE 2)

算法 4 (异回路中两对边的同解变换)

step 1: 在图 25 所示的 G 中, 设需要改变右端接位置的交叉边为 e_1 , 在右 B 区与非交叉边为 e_3 端接, 在右 A 区选择一对端接的非交叉边 e_2 和 e_4 , 且这四条边不在 G 图的同一回路中。

step 2: 将 e_2, e_4 的右端接点交换到 B 区, 并保持两者的端接关系, 记新生成的边为 e_2' 和 e_4' ; 保持 e_1, e_3 的右端接的关系, 并将它们交换到 A 区, 记作 e_1', e_3' 。

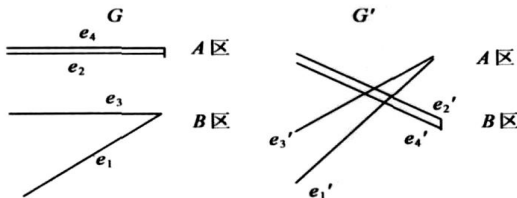


图 25 两对边的同解变换

step 3: 在 G' 中用交替取边法将其一分为二, 由于 G' 是由两个以上的回路组成, 所以有两种以上的剖分方法。不失一般性, 令 $(e_1', e_2') \in G_1', (e_3', e_4') \in G_2'$ 。

step 4: 将 G_i' 一分为二, 形成右对称的 G_{ij}' , 此时 e_1', e_2' 在 G_{ij}' 的分布有两种可能: ①若 $(e_1', e_2') \in G_1'$ 的不同回路, 需根据 e_3', e_4' 的分布情况决定 e_1', e_2' 的分布: e_3', e_4' 同在 G_{22}' 时, 令 e_1', e_2' 同在 G_{11}' ; 若 $e_3' \in G_{21}', e_4' \in G_{22}'$ 时, 则使 e_1', e_2' 分布在不同 G_{ij}' 中; ②若 $(e_1', e_2') \in G_1'$ 同回路偶距 (奇距), 则要求 e_1', e_2' 在同一 (不同) G_{ij}' 中, e_3', e_4' 也要在同一 (不同) G_{ij}' 中。

step 5: 调用算法 1 为 G' 赋值, 此时 e_{ij}' 四条边的 Y 值或者全部相同; 或者两两相同, 即 e_1' 与 e_4' 的 Y 值同, e_3' 与 e_2' 的 Y 值同。

step 6: 为 G 赋值, $(e_i') \Rightarrow e_i (i=1, 2, 3, 4)$, 此外, G 中未变动的边与 G' 同值。

图 26 是算法 4 的一个例子。它用连接图表示交换前后 G 图的变化。其中被交换的两对边集中在两个四元组中, 并赋予相同的 Y 值。

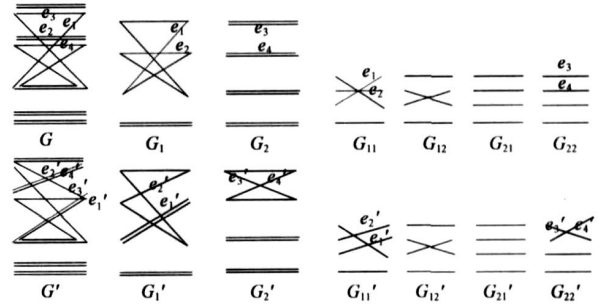


图 26 不同回路中两对边互换端点时, Y 值全部相等, Z 值两两相同

定理 4 在 CASE 2 的非对称图中, 运用算法 4 在两条回路中实施同解变换, 一定能使 (e_1', e_4') (e_2', e_3') 分别赋予相同的 YZ 值, 并使 G 有合理赋值。

证明: 当出现图 24 所示的情况时, 图 G 由 2 个以上回路组成, 所以有可能让 4 个交叉点都分到 G_1 中。由于回路图 G_1 是由 8 个点组成的, 分布在不同行和/或列的两个交叉点一定能和分处两区的两非交叉点处于同回路中 (见图 27, 虚线为 G 中回路, 实线为 G_1 中回路)。而保证 (e_1, e_2) 在 G_1 中同回路则是调用算法 4 处理非对称 CASE 2 的关键。

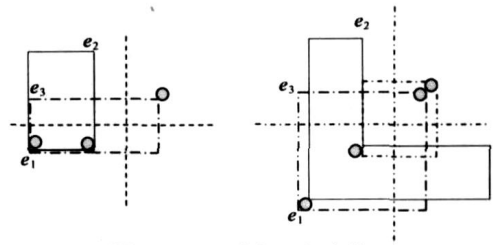


图 27 两对边的同解变换

因为 (e_1, e_2) 在 G_1 中同回路偶距, 根据推论 5, 在跨区交换端接点后, (e_1', e_2') 在新的 G_1' 中属于不同的回路, 这时 (e_1', e_2') 可以都分到 G_{11}' 中, 也可将 e_1' 分到 G_{11}' , 将 e_2' 分到 G_{12}' 中, 这取决于 (e_3', e_4') 在 G_2' 中的分布情况。总之, 为了使它们 Z 值相同, (e_1', e_4') 要在不同四元组的不同区, (e_2', e_3') 也要在不同四元组的不同区。因为 $(e_1', e_2') \in G_1'$ 的同一回路的情况不会出现, 从而避免了算法 4 中 step 4 的第 ②种可能性。

下面证明四条边的 Y 值两两相同。如果 $(e_1', e_2') \in G_{11}', (e_3', e_4') \in G_{22}'$, 则令这两个四元组 Y 同值 (见图 28); 否则, 令 e_1', e_4' 所在的两个四元组 Y 同值, e_2', e_3' 所在的两个四元组 Y 同值。

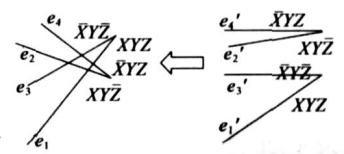


图 28 四条边的 Y 赋值法

5.6 同回路异象限的同解变换

在讨论 CASE 3 非对称图前,先考察三对交叉点在回路图中的分布.在一个象限中,上下两区交叉点的分布有 03 或 21 两种.从 G 图左右或上下看,有 0330 型,1221 型和 0321 型三种.其中,0330 型肯定是对称的;0321 型肯定是非对称的;1221 型有可能是对称的,也有可能是非对称的.判定图形 G 为非对称,一定要保证它既不是左对称,又不是右对称,因此从任何一个侧面看,都不能有 0330 型.仅当从左右或上下看为 1221 型和 0321 型时,才有可能为非对称.此时有三种组合方式.

引理 12 在 CASE 3 中,当出现图 29 所示的三类 6 种情况时,图 G 为非对称.

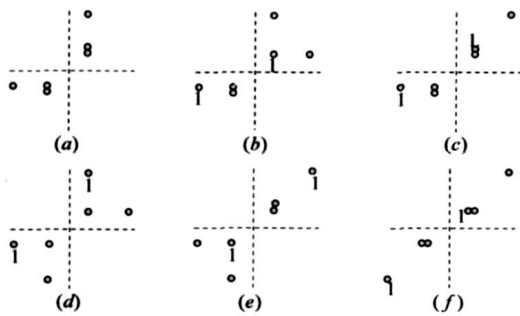


图 29 CASE 3 非对称 G 的回路图中交叉点的分布

对 G 二着色.首先三对交叉点不可能同色,否则在分割 G 时,可将三对交叉点都分在 G_1 中, G 就属于 CASE1 了.因此一定有一对交叉点的着色与另两对交叉点不同,图中用“1”示之.从左右、上下两个方向看回路图 29,在图 (a) 中,交叉点的分布都是 0321 型,因而非对称图.在图 (b)、(c) 中,交叉点的分布是混合型,即上下看是 0321 型,左右看是 1221 型;当有一对交叉点分布如图所示时,余下的两对交叉点分布类同图 23 (b),所以从左右看也是非对称.在图 (d)、(e)、(f) 中,交叉点的分布都是 1221 型.当一对交叉点在 G_j 中的配对如图所示时,余下的两对交叉点分布, (d) 类同图 23 (a), (e), (f) 类同图 23 (b),所以从上下、左右看都是非对称的.

在一般情况下,用算法 3 可解决 CASE 3 的非对称问题.但对给定的交叉点 e_1 来说,如果不存在处于同回路、异区的非交叉点 e_2 时,根据推论 8 和 9,只有两种可能:同象限同位置的两个交叉点在 $L=2$ 的回路上;不同象限的两个交叉点在 $L=4$ 的回路上.因为回路中的交叉点是成对出现的,如果有同象限的两个交叉点组成了 $L=2$ 的回路,同象限的第三个交叉点只能和异象限的某个交叉点组成 $L=4$ 的回路.所以在非对称的 G 图中,至少有一条 $L=4$ 的回路,其中两个交叉点和两个非交叉点分布在四个象限中.此时可运用引理 8(同

解变换 3) 来解决.

算法 5 (同回路中一对非交叉边的同解变换)

当 CASE 3 中非对称图 G 属于图 29 所示的情况时,可在连接图 G 中选择同一回路中偶距、异象限的非交叉边 e_1, e_2, e_1 横越第 II 和第 I 象限, e_2 横越第 II 和第 IV 象限.

step 1: 保持 e_1, e_2 左端点不变,右端点位置对换,生成新的交叉边 e'_1, e'_2 ,在 G' 中属于不同的回路.

step 2: 采用沿回路交替取边法将 G' 一分为二,并将 e'_1 分到 G'_1 ,将 e'_2 分到 G'_2 .

step 3: 采用沿回路交替取边法继续对 G'_1 和 G'_2 一分为二,生成四个有交叉边的 G'_j .不失一般性,假设 $e'_1 \in G'_{11}, e'_2 \in G'_{22}$.

step 4: 调用算法 2 为 G' 赋值,并令 G'_{11} 和 G'_{22} 的 Y 值相等,从而保证 e'_1, e'_2 边的 XYZ 值完全相同.

step 5: (e'_1) \Rightarrow $e_1, (e'_2) \Rightarrow e_2, G$ 中未变动的边与 G' 中对应的边赋同值.

图 30 给出的例子表明:此时的非对称的 G 图无法运用算法 3,只能用算法 5 赋值.

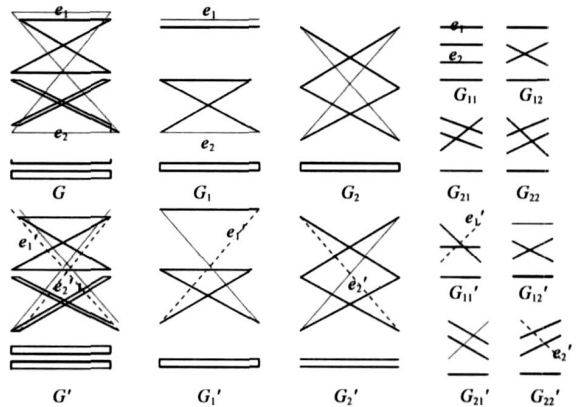


图 30 将两条非交叉边对换为交叉边

定理 5 在 CASE 3 中,非对称的 G 图一定能有的合理的赋值.

证明:(略)

最后说明一点,经过算法 5 的变换后, G' 会不会蜕变为 CASE 2.首先 G 不可能有三条含一对异象限交叉点的 $L=4$ 的回路,否则在分割 G 时,可以将三对交叉点集中在 G_1 中,使 G_1 的两个四元组为 $\# \times$ 型,这与 CASE 3 前提矛盾.引理 9 说明同象限的两个交叉点如在同一条 $L>4$ 的回路中,就能用算法 3.

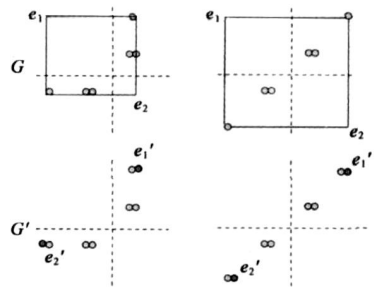


图 31 必须用定理 5 处理的 G 图中交叉点的分布

所以必须用算法 5 的 G 图, 只能有一条 $L=4$ 的回路, 其中含一对交叉点. 其余四个交叉点形成两个“棍棒”, 如图 31 所示. 选择非交叉点 e_1, e_2 后, 实施算法 5 变换, G' 中的 e'_1 和 e'_2 与原有的一个交叉点又形成“棍棒”. 当剖分 G' 时, G'_1 和 G'_2 中交叉点分布完全相同.

6 结束语

混洗交换网络的可重排性是交换机的核心问题. 本文借助连接图和回路图的表示方法, 交替运用图形分解与图形压缩手段, 直观地给出了 16×16 的 7 级 Ω 网络的无阻路由解. 虽然 16×16 的可能置换数多达 21 万亿, 但我们将置换归并成 5 类, 并提出了 5 种赋值算法. 在 CASE 2 和 CASE 3 中, 首要选择是算法 1 和算法 3, 算法 4 和算法 5 并不适用于任何对称的 G 图, 它仅是算法 3 的补充和非对称图赋值的最后手段. 实践证明, 当非对称图通过同解变换后都能变为对称图, 实质上这是重新安排交叉边或非交叉边在四元组中的配对.

如何借助于本文提出的分类思想和同解变换方法, 将赋值算法进一步推广到 $n \geq 5$ 的多级 Ω 网络中, 仍然是一个极富挑战性的课题. 当 $n \geq 5$ 后, 置换的分类数量将会增大, 此时需要寻找能递归的构造型算法, 才有可能证明 $(2n-1)$ 级混洗交换网络的可重排性.

参考文献:

- [1] K Kim, C S Raghavendra. A Simple Algorithm to Route Arbitrary Permutations on 8 input 5 stage Shuffle/Exchange Network [A]. Proceeding 5th International Parallel Processing Symposium [C]. May 1991. 398–403.
- [2] C S Raghavendra, A Varma. Rearrangeability of the 5 Stage Shuffle/Exchange Network for $N=8$ [A]. Proceedings of 1986 International Conference on Parallel Processing [C]. 1987. 119–122.
- [3] H S Stone. Parallel Processing with the Perfect shuffle [J]. IEEE Transactions on Computers, 1971, C-20(2): 153–161.
- [4] D H Lawrie. Access and alignment of data in an array processor [J]. IEEE Transactions on Computers, December 1975, C-24(12): 1145–1155.
- [5] C S Raghavendra. On the Rearrangeability Conjecture of $(2\log 2N-1)$ -stage Shuffle/Exchange Network [A]. IEEE computer Society Technical Committee on Computer Architecture Newsletter [C]. Winter 1994–95. 10–12.
- [6] C Wu, T Feng. On a Class of Multistage Interconnection Networks [J]. IEEE Transactions on Computers, 1980, 29(8): 694–702.
- [7] C Wu, T Feng. The universality of the shuffle/exchange network [J]. IEEE Transactions on Computers, 1981, 30(5): 324–332.
- [8] V E Benes. Proving the rearrangeability of connecting networks by group calculations [J]. Bell system Tech Jour 1975, 54: 421–434.
- [9] D S Parker. Notes on shuffle/exchange type switching networks [J]. IEEE Transactions on Computers, Mar. 1980, c 29: 213–222.

- [10] N Linial, M Tarsi. Efficient generation of permutations with the shuffle/exchange network [R]. Technical Report, UCLA computer science dept. 1982.
- [11] F Sovis. Uniform Theory of the Shuffle Exchange Type Permutation Networks [A]. in Proceeding 10th Ann Int Symposium Computer Architecture [C]. June 1983. 185–191.
- [12] C L Wu, T Y Feng. The universality of the shuffle exchange network [J]. IEEE Transaction on Computers, May 1981, c 30: 324–332.
- [13] K Y Lee. On the rearrangeability of the $2\log N-1$ stage permutation network [J]. IEEE Trans Comp, May 1985, C-34(5): 412–425.
- [14] S T Huang, S K Tripathi. Finite state model and compatibility theory: new analysis tools for permutation networks [J]. IEEE Transaction on Computers, July 1986, c 35(7): 591–601.
- [15] S Andresen. The looping algorithm extended to base 2i rearrangeable switching networks [J]. IEEE Transaction on communication, 1977, COM-25: 1057–1063.
- [16] S T Huang, S K Tripathi. Finite state model and compatibility theory: new analysis tools for permutation networks [J]. IEEE Trans Comp, 1986, 35(7): 509–601.
- [17] A Varma, C Raghavendra. Rearrangeability of multistage shuffle/exchange network for $N=8$ [J]. IEEE Trans Comm, 1988, COM-36(10): 1138–1147.
- [18] N Linial, M Tarsi. Interpolation between bases and the shuffle exchange network [J]. European J Combin, 1989, 10(1): 29–39.
- [19] T Feng, S Seo. A new routing algorithm for a class of rearrangeable network [J]. IEEE Trans Comp, 1994, 43(11): 1270–1280.
- [20] F Sovis, On rearrangeable networks of the shuffle exchange network [A]. Proc of frontiers of massively parallel computation [C]. 1990. 304–314.
- [21] H Cam, J A Fortes. rearrangeability of shuffle/exchange network [A]. Proc of frontiers of massively parallel computation [C]. 1990. 303–314.
- [22] A Abdennadher, T Feng. On rearrangeability of Omega Omega networks [A]. in Proc. 1992 International Conference on Parallel Proc [C]. 1992. F 159 165.
- [23] M Kim, H Yook, S Maeny. On the correctness of inside out routing algorithm [J]. IEEE Trans Comp, 1997, 46(7): 820–823.
- [24] Hasan Can. Rearrangeability of $(2r-1)$ -stage shuffle exchange networks [J]. SIAM j comput, 2003, 32(3): 557–585.

作者简介:

戴 浩 中国电子设备系统工程公司研究所研究员, 博士生导师, 中国工程院院士. 主要研究方向为电子信息系统、计算机网络.

E-mail: dai_hao@sina.com

沈孝钧 美国密苏里州立大学堪萨斯分校计算机与工程学院教授, 博士生导师. 主要研究方向为计算机结构与算法、计算机网络.