二分图约束的顶点覆盖问题的快速算法

何 峰, 车文刚
(昆明理工大学 信息工程与自动化学院, 云南 昆明　650051)

摘要: 对超大规模集成电路芯片(VLSI)的缺陷修复可归结为受二分图约束的顶点覆盖问题, 该问题属于NP 完全问题. 目前仍不能在多项式时间内对该问题求解. 本文应用参数计算理论, 将问题化简为与输入问题规模无关的问题来求解. 并利用二分图的特性, 提出了一种简单、高效的算法, 大大提高了修复速度.

关键词: 可修复阵列; 二分图; 顶点覆盖; 匹配; 参数计算
中图分类号:TP393　　文献标识码: A　　文章编号: 1007- 855X(2003)05- 0085- 05

An Efficient Algorithm for Constraint Bipartite Vertex Cover

HE Feng, CHE Wen-gang
(Faculty of Infomation Engineering and Automation, Kunming University of Science and Technology, Kunming 650051, China)

Abstract: The fault coverage problem for reconfigurable arrays has received as constraint bipartite vertex cover problem, which is proved as a NP- complete. It can't be resolved in polynomial running time. The theory of parameterised computation is used to reduce the kernel of problem which is independent of the original problem. Furthermore, the algorithm's running time is bounded by the study for the characteristics of the bipartite graph.

Key words: reconfigurable arrays; bipartite graph; vertex cover; graph matching; parameterised computation

0 引 言

随着超大规模集成电路芯片生产技术的发展, 单片芯片的集成度越来越高. 要想一次生产出没有任何缺陷的芯片已不太可能. 为了提高芯片的成品率, 目前一种较为常用的方法是在超大规模集成电路芯片内部集成备用行(SR)和备用列(SC)用以替换阵列中有缺陷的行和列[5, 6], 采用激光修复工艺可以很容易地实现这一修复过程[6]. Kuo 和 Fuchs[5]对该问题进行了开拓性的研究并给出了一些极有价值得结论: 芯片可以抽象为一个 $M \times N$ 的阵列, 对该阵列进行修复可以归结为用SR 个行和SC 个列对 $m \times n$ 的二分图进行覆盖. 另外, Kuo 和 Fuchs 还对如何修复给出了两个算法: 一个是近似算法, 到目前为止, 该算法仍然是最高效的顶点覆盖问题的求解算法之一. 但该算法的缺点在于不能保证求出顶点覆盖的最优解. 也就是说, 当一片芯片在事实上是可修复时却有可能在近似算法中得到不可修复的结论. 另一个算法是启发式的分支- 界定算法, 但该算法仅仅在中等规模的二分图中可行(SR, SC 小于 20). 笔者之一在此前也曾提出过一种基于启发式的错误谱算法[1], 有效的提高了求解的效率. 但该算法仍然存在不能保证最优解的问题.

由于顶点覆盖问题的精确求解在实际应用中的需要, 20 世纪 90 年代末期提出了参数计算理论, 并很好的应用于顶点覆盖问题的求解上. Buss[9]提出了第一个参数算法, 并证明了该算法的运行时间复杂度为 $O(kn + 2^k k^{2K+2})$, 该算法的时间复杂度随后被 Downey 和 Fellows[10]改进为 $O(kn + 1.322\,718^k k^2)$. Balasubramanian[2]率先在顶点覆盖问题的精确求解上突破了以 2 为底的指数复杂度. 他提出的新算法证明了顶点覆盖问题可在的时间复杂度内被求解[2]. 最近, Chen 和 Kanj[3]又提出了时间复杂度为 $O(kn + 1.271^k k^2)$ 的固定参数算法[3]. 这是迄今为止已发表的最快的顶点覆盖问题算法.

收稿日期: 2003- 04- 30.

第一作者简介: 何峰(1974. 8~), 男, 硕士; 主要研究方向: 图论及其运用. **E- mail:** kmhefeng@hotmail.com

但是,以上的算法并不能直接应用于受二分图约束的顶点覆盖问题(Constraint Bipartite Vertex Cover, CBVC)求解.这是由于电路芯片的缺陷修复的实际问题造成的.例如,在传统的图顶点覆盖问题中,如果有一个顶点的邻接点的度数为1,则该顶点必然包含于最小顶点覆盖集中.从而减小了搜索树的范围.但在CBVC问题中,该法则却未必能用,因为在CBVC问题中最小覆盖集同时受到SR,SC(即 k_1, k_2)的约束.但是,二分图本身也有许多有用的特性可以减小问题的求解范围.

本文利用二分图的特殊性,应用固定参数算法,提出来一种简单但却十分有效的算法,可在时间复杂度为 $O(1.26^{k_1+k_2}+(k_1+k_2)n)$ 求得CBVC的精确解.

1 参数计算理论

参数复杂理论目前主要用于解决"是、非"判定性问题.一个典型的判定问题如:"给定一个实例 I,I 是否具有性质 P."很明显,一个图的顶点覆盖问题是一个判定问题.在此给出一个受二分图约束下的顶点覆盖问题的数学表示:给出一个二分图 $G=(V_1, V_2, E)$ 和两个正整数 k_1, k_2.能否得到两个顶点集 $C_1 \subseteq V_1$, $C_2 \subseteq V_2$,并且 $|C_1| \leqslant k_1$, $|G_2| \leqslant k_2$,使得 E 中的每一条边都至少有一个顶点在 $C_1 \cup C_2$ 中?在固定参数算法中,有两个方法常用来提高算法的效率:化简问题核心和限定搜索树.这两种方法针对二分图的应用也是本文所提出算法的核心.

2 CBVC算法

2.1 化简问题核心

化简问题核心是参数复杂算法中最常用,也是最有效的方法.其目的是将原始问题经过一系列的化简,最终得到一个规模更小、但能得出原始问题解的"核心"问题.虽然"核心"问题仍然是指数复杂的,但可得到一个规模依赖于固定参数 K 而与原始问题无关的一个"核心"问题,从而可以大大提高求解的效率.

首先,在二分图中删除掉所有在 V_1 中但顶点度大于 k_2,在 V_2 中但顶点度大于 k_2 的顶点和与这些顶点相关的边.满足上述条件的顶点必然包含于最小顶点覆盖集中.这是因为在图的顶点覆盖问题中,如果一个顶点 v 每有被包含于一个顶点覆盖集中,则它的所有相邻顶点必然被包含于该顶点覆盖集中.所以,如果在而没有被包含于最小顶点覆盖集中,则表明该二分图在 V_2 中的最小顶点覆盖集大于 k_2,也就是说该芯片不可修复.同理,在 V_1 中但顶点度大于 k 的顶点也必然被包含于最小顶点覆盖集中.

令 $S_1=\{v \in V_1 \mid \deg(v) > k_2\}$, $S_2=\{w \in V_2 \mid \deg(w) > k_1\}$,,则可令 $k'_1 = k_1 - |S_1|$, $k'_2 = k_2 - |S_2|$,从而原始问题便转换为在一个顶点数最多为 $k'^2_1 + k'^2_2$ 的图 G' 中求约束参数为 k'_1 和 k'_2 的最小顶点覆盖.下面,针对二分图的一些特性,对问题进行进一步的化简.首先给出一些以后将会用到的概念:

定义3.1 图的匹配:

设 M 是 $E(G)$ 的一个子集,如果 M 中任意两条边在 G 中均不邻接,则称 M 是 G 的一个匹配.如果不存在其它匹配 M' 使得 $|M'| > |M|$,则称 M 是最大匹配.其中 $|M|$ 表示匹配 M 的边数.如果在匹配 M 中包含了图 G 中的所有顶点,则称该匹配为完备匹配.

定义3.2 饱和与非饱和:

若匹配 M 的某条边与顶点 v 关联,则称 M 饱和顶点 v 并且称 v 是 $M-$ 饱和的.

定义3.3 交互道:

若 M 是二分图 $G=(V, E)$ 的一个匹配.设从图 G 中的一个顶点到另一个顶点存在一条道路,这条道路是由属于 M 的边和不属于 M 的边交替出现组成的,则称这条道路为交互道.

定义3.4 可增广道路:

若一交互道的两端点为关于 M 非饱和顶点时,则称这条交互道是可增广道路。显然,一条边的两端点非饱和,则这条边也是可增广道路.

在一个二分图中构造一个最大匹配需要 $O(m\sqrt{n})$ 的时间复杂度[8]. 已经证明在二分图中, 最大匹配 M 中边的数量等于该二分图中最小顶点覆盖集中顶点的数量[7]. 所以, 对 G' 构造一个最大匹配M, 并判断 $|M|$ 是否大于 $k'_1+k'_2$. 如果是, 则问题无解; 否则, 则继续进行下面的化简.

令 D 为G' 中至少不属于一个最大匹配的顶点的集合; 令A 为包含于$V-D$ 但至少有和D 中的一个顶点相邻的顶点集合; 令 $C=V-D-A$.

命题 3.1　[Gallai－Edmonds 结构定理] ([7], 定理 3.2.4)

设二分图 G 包含如上所定义的子集D, A 以及C, 那么

1) D 是二分图G 中的一个独立集(independent set) 且D 中没有一个顶点被包含于任何一个最小顶点覆盖集中;

2) A 是G 中所有最小顶点覆盖集的交集;

3) 子图 $G(C)$ 中必有一个完全匹配.

在 *CBVC* 问题中, 如何找到一个D 顶点集, 成为了进一步解决问题的关键. 下面给出一个简单的算法来找到 D. 令 W 为一个最大匹配M 中非饱和的顶点集, W' 为所有从W 中的顶点v 出发, 经过偶数条交互道可到达的顶点集. 详见图 1. W' 即为图中虚线框住部分. W' 就是上面定义的集合D. 集合 D 已知, 则集合A 和C 很容意便可求得.

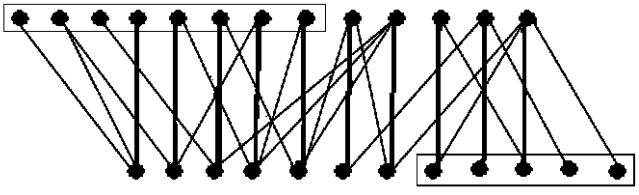

图 1　二分图中的 W' 集合

由命题3.1可知, 顶点集D 中的顶点没有一个被包含于最小顶点覆盖集中, 而顶点集A 中则包含了所有最小顶点覆盖集的交集. 故可以构造一个图 $G'=G(C)=(V'_1\cup V'_2, E')$. 如果顶点集$A$ 中有h_1个 V_1 顶点, h_2 个 V_2 顶点, 令 $k''_1=k'_1-h_1$, $k''_2=k'_2-h_2$. 这样, 二分图 G 在k_1, k_2 参数约束下有顶点覆盖当且仅当二分图 G' 在k''_1, k''_2参数约束下有顶点覆盖. 由命题 3.1 的结论(3) 可知, 图 G' 中存在一个完全匹配, 由二分图的性质可知, 图 G' 的最小顶点覆盖包含 $|G'|/2$ 个顶点.

下面来分析上述算法的时间复杂度. 首先, 在一个有 n 个顶点, m 条边的二分图中找到一个最大匹配的时间复杂度为$O(m\sqrt{n})$[8]. 而利用最大匹配构造集合 D, A 和C 都只需要线性时间复杂度. 只有在二分图 G' 中寻找最小顶点覆盖集是指数复杂的. 令所需时间为: $t(k''_1+k''_2)$, 则整个算法的时间复杂度为: $O(m\sqrt{n}+t(k''_1+k''_2))$

在Hasan 和Liu 的论文[4] 中提出了一个临界集(critical sets) 的概念, 在该文中, 临界集是指图 G 中所有最小顶点覆盖集的交集. 很明显, 临界集就是命题2.1中的A 集合. 但Hasan 和Liu 仅将临界集应用于减小分支－界定算法的规模, 并没有对集合 D 和C 进行更深一步的研究.

此外, 由于命题2.1明确指出二分图 G 删除集合D 和A 后剩下的子图G' 中必然包含有一个完全匹配, 这不但立刻缩小了问题的规模, 也为下一步求解提供了极大的方便.

2.2 限定搜索树

根据上面的分析, 可以将问题化简为求二分图 G' 在k''_1, k''_2参数约束下的最小顶点覆盖, 且 G' 中存在一个完全匹配. 下面应用限定搜索树的方法来提高算法的效率.

如果一个二分图 B 中的每一条边都包含于B 中的一个完全匹配, 则把这个二分图称为基本二分图. 基本二分图有一个很有用的性质可以极大地提高搜索顶点覆盖的过程: 在一个基本二分图中, 最小顶点覆盖要么完全包含二分图的上半部分, 要么完全包含二分图的下半部分, 没有其他可能.

在Dulmage－Mendelsohn 分解理论中[7] 证明了一个包含完全匹配的二分图可以被分解为一系列基本二分图 $B_i=(V_{1i}\cup V_{2i}, E_i)$, $i=1, 2, \cdots, r$, 且在图 G 中从子图B_i 到另一个子图B_j, $(i<j)$ 的任意一条边必然一个顶点在 B_i 的上半部分, 另一个顶点在 B_j 的下半部分. 这些基本二分图 B_i 称为(基本) 块, 当块 B_i 中 $|V_{1i}|=|V_{2i}|=d$ 时, 把它称为 d－块. 各个块之间的边称为块间边.

下面给出如何利用Dulmage－Mendelsohn 分解理论来分解二分图 G', 并分析算法的时间复杂度. 算

法见图2

算法的第一步是构造图 G 的一个完全匹配 M, 当图有 n 条边时, m 条边时, 其时间复杂度为 $O(m\sqrt{n})$; 由块间边的定义可知, 二分图 G 中的任意边 $e = [u, v]$ 为块间边当且仅当该边不属于 G 中的任何一个完全匹配. 而任意边 $e = [u, v]$ 属于二分图 G 的一个完全匹配当且仅当二分图 $G - \{u, v\}$ 有一个完全匹配. 故算法中的第二步可直接将图 G 中的所有块间边标记出来. 步骤二的计算复杂度主要在于在图 G' 中判断是否有完全匹配. 由于已经有了一个完全匹配 M, 只需要判断在 G' 中相对于 M 是否存在可增广道路, 便可判定 G' 中是否存在完全匹配. 而这一过程可以用广度优先算法很容易地实现. 其时间复杂度为 $O(n^2)$. 步骤三、四均可在线性时间内完成. 故整个算法的时间复杂度为: $O(n^2)$.

DM - 分解算法.
输入: 一个存在完全匹配的二分图 G
输出: $B_i = (V_{1i} \cup V_{2i}, E_i)$, $i = 1, 2, \cdots, v$,
1. 在 G 中构造一个完全匹配 M;
2. For each edge $e = [u, v]$ in G do
If the graph 没有完全匹配
Then 将 e 标记为块间边;
3. 删除所有的块间边;
4. 剩下的无关联子图即为基本块;

图2 DM- 分解算法

下面对整个算法中唯一需要指数时间复杂度的部分, 应用限定搜索树的方法, 提高算法的效率. 以上已经应用 DM 分解算法, 将具有一个完全匹配的二分图 G' 分解成为 B_i 个基本块. 因为每一个基本块都只有两种情况被包含于最小顶点覆盖, 所以可以以 B_i 为顶点, 块间边为边, 构造出一个新的图 D. 令 $F(k_1 + k_2)$ 为搜索树中叶子的数量. 下面将分情况讨论在 D 中搜索的时间复杂度:

情况一: 图 D 中的每一个顶点 B_i 至少为 3- 块;

假设图 D 中的每一个顶点 $B_i = (V_{1i} \cup V_{2i}, E_i)$, 且 $d = |V_{1i}| = |V_{2i}| \geqslant 3$. 因为 B_i 具有一个完全匹配, 故 B_i 内部的顶点覆盖要么包含全部上半部分的顶点, 要么包含全部下半部分的顶点, 没有其他可能. 所以在对搜索树进行搜索的时候, 每次至少可以将 3 个顶点加入到最小顶点覆盖集中去. 也就是说, 在对搜索树进行搜索的递归次数满足下面的关系式:

$$F(k_1 + k_2) \leqslant 2F(k_1 + k_2 - 3) \quad (1)$$

情况二: 图 D 中有一个度数为 1 的顶点 B_i 为 2- 块;

在这种情况下, 由于 B_i 的顶点度为 1, 假设是它的上半部分的顶点被包含于最小顶点覆盖中, 如果块间边是从 V_{1i} 连向和它相邻的顶点 B_j 的 V_{2i}, 则马上可以判定 B_j 顶点的 V_{1j} 部分必然也被包含于最小顶点覆盖集中. 反之亦然. 因此在每次搜索时也至少可以判定 3 个顶点. 也就是说, 在图 D 中有度数为 1 的顶点 B_j 时, 对搜索树进行搜索的递归次数满足关系式(1).

当图 D 中有度数大于 1 的 2- 块顶点时, 很容易得出对搜索树的限定关系满足下式:

$$F(k_1 + k_2) \leqslant 2F(k_1 + k_2 - 4) \quad (2)$$

情况三: 图 D 中有度数为 1 的 1- 块顶点 B_i, 且和该顶点相邻的顶点 B_j 也为 1- 块顶点.

当图 D 中有度数为 1 的 1- 块顶点时, 如果和它相邻的顶点为 2- 块顶点, 则搜索的递归次数关系式满足关系式(2); 所以仅分析当其相邻的顶点也为 1- 块顶点的情况. 而其邻接点 B_j 又可分为度数等于 1 和大于 1 两种情况:

1) B_j 的度数等于 1:

则表明 B_i 和 B_j 为一个无关联子图, 与块间边相连的两个顶点必然被包含于最小顶点覆盖集中. 对它的搜索可在线性时间内完成.

2) B_j 的度数大于 1:

也就是说 B_j 至少有一个块间边与其他块 B_k 相连. 如果 B_k 为 2- 块顶点, 则搜索的递归次数关系式满足关系式(2); 如果 B_k 也为 1- 块顶点时, 由完全匹配的性质, B_i、B_j 和 B_k 中的最小顶点覆盖集可以一次判定出来, 因而满足关系式(1).

由固定参数理论可知, 1.259 同时满足关系式(1)、(2). 即 $F(k_1 + k_2) = 1.259^{k_1 + k_2}$. 也就是说在 G' 中进行完全搜索的时间复杂度为 $O(1.259^{k_1 + k_2})$.

2.3 CBVC 算法整体分析

基于上面的分析, 下面给出完整的 CBVC 算法(见图 3).

步骤 1– 2 仅使用了线性时间, 化简了问题的规模, 使问题被限定在最多有 $k'^2_1 + k'^2_2 +$ 条边的一个二分图中, 使问题变成了与输入无关而仅与参数 k'_1, k'_2 相关.

步骤 3 应用 Gallai– Edmonds 结构定理进一步化简问题规模. 在删去了集合 D 和 A 后, 不但立即缩小了问题的规模, 还得到了一个具有特殊性质的子图 $G(C)$. 从而可以在第四步应用 DM 分解算法. 该步骤的时间复杂度为 $O(m\sqrt{n})$. 步骤 4 实际上是为了步骤 5 作准备, 其目的是为了在完全搜索中减少搜索树的规模. 步骤 5 是整个算法中唯一指数复杂的, 但由 2.3 节的分析可知, 该步骤的时间复杂度为 $O(1.259^{k_1+k_2})$.

综上所述, 整个算法的时间复杂度为 $O(1.259^{k_1+k_2} + (k_1 + k_2)n)$.

CBVC 算法:
输入: 二分图 $G = (V_1 \cup V_2, E)$, 两个正整数 k_1, k_2;
输出: 二分图 G 的一个满足参数 k_1, k_2 约束的最小顶点覆盖 K, 或者报告不存在满足条件的最小顶点覆盖
1. For $V1$ 中每一个顶点度大于 k_2 的顶点 u
将 u 包含于 K 并将 u 从二分图 G 中删除;
$k_2 = k_2 - 1$;
2. For V_2 中每一个顶点度大于 k_1 的顶点 v
将 v 包含于 K 并将 v 从二分图 G 中删除;
$k_1 = k_1 - 1$;
3. 应用 Gallai– Edmonds 结构定理化简问题规模
4. 应用 DM 分解算法将二分图 G' 分解为一系列基本块 $B_i = (V_{1i} \cup V_{2i}, E_i)$, $i = 1, 2, \cdots, r$,
5. 用完全搜索算法找到最小顶点覆盖
6. 集成解

图 3　CBVC 算法

3 结束语

参数计算理论在算法设计和分析中是一种极为有效的理论工具, 特别是在 NP 完全问题的求解, 目前已有许多成功的应用. 在本文中, 除了对参数计算理论的应用外, 还针对阵列修复的特殊性, 将问题进一步化简, 得到了较好的结果. 但本算法没有考虑到实际工业应用中如何减少备用行和备用列的使用以提高生产率. 这是今后算法改进的方向.

参考文献:

[1] Wengang Che and Israel Koren. Fault Spectrum Analysis for Fast Spare Allocation in Reconfigur– able Arrays[J]. in Proceedings of the IEEE International Workshop on Defect and Fault Tolerance in VLSI Systems, 1992.

[2] R. Balasubramanian, M. R. Fellows and V. Raman. An improved fixed parameter algorithm for vertex cover[J]. Information Processing Letters, 1998, 65: 163~ 168.

[3] Jianer Chen and A. Kanj and WEIJIA. Vertex Cover: further observations and further improvement, Proc. 25th Int. Workshop on Graph– Theoretic Concepts in Computer Science (WG' 99)[J]. Lecture Notes in Computer Science 1999, 1665: 313~ 324.

[4] N. Hasan and C. L. Liu. Minimum Fault Coverage in Reconfigurable Arrays[J]. in IEEE International Symposium on Fault– Tolerant Computing, 1988. 348~ 353.

[5] S. Y. Kuo and W. K. Fuchs. Efficient Spare Allocation for Reconfigurable Arrays[J]. IEEE Design and Test, 1987(2): 24~ 31.

[6] R. T. Smith, J. D. Chlipala, J. F. Bindels, R. G. Nelson, F. H. Fischer and T. F. Mantz. Laser Programm– able Redundancy and Yield Improvement in 64K DRAM[J]. IEEE Journal of Solid– State Circuits, vol SC– 16, 1981, (10). 506~ 513.

[7] L. Lovasz and M. D. Plummer. Matching Theory[J]. Annals of Discrete Mathematics 29, North– Holland, 1986.

[8] S. Micali and V. Vazirani. An algorighm for finding maximum matching in general graphs, Proc[J]. 21st IEEE Symposium on the Foundation of Computer Science (FOCS' 80), 1980. 17~ 27.

[9] J. F. Buss, J. Goldsmith. Nondeterminism within P[J]. In SIAM Journal of Computing, 1993, 22, 560~ 572.

[10] R. G. Downey, M. R. Fellows and U. Stege. Parameterized complexity: A framework for systematically confronting computational intractability[J]. in Contemporary trends in Discrete Mathematics: From DIMACS and DIMATIA to the Future, F. Roberts, J. Kratochvil, and J. Nesetril, eds., AMS– DIMACS Procedings Series 49, AMS, 1999. 49~ 99