一种基于链暗示技术的二分图受约束最小点覆盖问题的近似算法*)

许小双 王建新 刘云龙 陈建二

（中南大学信息科学与工程学院 长沙 410083）

摘 要 二分图受约束最小点覆盖问题作为一个 NP-完全问题，无法在多项式时间内得到最优解，除非 P=NP。基于此，本文提出了一种基于链暗示技术的二分图受约束最小点覆盖问题的近似算法，具体为：当二分图受约束最小点覆盖问题实例中存在满足约束条件的最小点覆盖(k_u,k_l)时，对任意给定的近似率$\delta=1+\varepsilon>1$，一定可以找到一个受约束近似点覆盖(k_u^*,k_l^*)，对应的近似率为$\max\{k_u^*/k_u,k_l^*/k_l\}\leqslant 1+\varepsilon$，整个近似算法的运行时间复杂度为$O(2^{2/\varepsilon})$。显然，它是二分图受约束最小点覆盖问题的一个多项式时间近似方案（polynomial time approximation scheme，PTAS 算法）。

关键词 二分图的受约束最小点覆盖，近似算法，参数计算，PTAS 算法

An Approximation Algorithm Based on Chain Implication for Constrained Minimum Vertex Cover in Bipartite Graphs

XU Xiao-Shuang WANG Jian-Xin LIU Yun-Long CHEN Jian-Er

(School of Information Science and Engineering, Central South University, Changsha 410083)

Abstract Constrained minimum vertex cover in bipartite graphs (Min-CVCB) problem is a NP-complete problem, it can't be solved in polynomial time, unless P=NP. In this paper, we provide an approximation algorithm which is based on chain implication to solve this problem. When the instance of Min-CVCB problem has a minimum vertex cover (k_u, k_l), for any given constant, $\delta=1+\varepsilon>1$, we can get a constrained approximate vertex cover (k_u^*, k_l^*), the approximation ratio is $\max\{k_u^*/k_u, k_l^*/k_l\}\leqslant 1+\varepsilon$, and the algorithm runs in time $O(2^{2/\varepsilon})$. Obviously, it is a polynomial time approximation scheme for the Min-CVCB problem.

Keywords Constrained minimum vertex cover in bipartite graphs, Approximation algorithm, Parameterized computation, Polynomial time approximation scheme

1 引言

随着 VLSI 技术的发展，电路芯片的规模越来越大，在制作工艺中引入错误的可能也随之增加。在芯片集成度不断增加的情况下，制作过程中出现错误存储单元是不允许的。一种较好的解决办法是使用可修复阵列，即在每一个存储矩阵的行和列旁边留有一定数目的备用行和列；当矩阵中发现有错误单元时，便用它们来替换矩阵中含有错误单元的行和列，从而使得矩阵恢复正常。替换的行和列所构成的集合即是一个点覆盖。由于修复一个芯片的代价正比于所使用替换行和列的数目，所以人们常常替换尽量少的行和列来修复矩阵，也即寻找满足一定约束条件下二分图的最小点覆盖[3]。为便于进一步准确描述、讨论该问题，在此先给出它的定义。

定义 1 二分图受约束最小点覆盖问题（Min-CVCB 问题）：给定一个二分图 $G=(U,L,E)$ 以及两个正整数 k_u 和 k_l，构造图 G 中的一个最小点覆盖 C，使得 C 中包含至多 k_u 个 U 中结点和 k_l 个 L 中结点（或者报告没有这样的最小点覆盖存在）。

Min-CVCB 问题是一个 NP-完全问题[4]。从定义可以看出，Min-CVCB 问题是求满足约束条件的解最小，因此是一个 NP-优化问题。

最近 20 年，人们对这个问题做了大量的研究。早期人们主要采用启发式算法对该问题进行研究，具体可见文[2,3,7～11]。其中，Hasan 和 Liu[3] 提出了 critical set 的概念，引入了一种人工智能方面的基于 A* 的分枝定界算法来解决 Min-CVCB 问题，但它无法在预定时间内给出问题的解答，在最坏情况下运行时间是 $\Omega(k_u+k_l+mn^{1/2})$)，其中 n 为图 G 结点个数，m 为图 G 中边条数。

最近几年主要围绕如何利用参数计算理论来求解 Min-CVCB 问题的精确结果进行研究[4,5]。文[5]通过分析二分图结构后使用分枝搜索技术提出了一种时间复杂度为 $O((k_u+k_l)n+1.3999^{k_u+k_l})$ 的算法，其中 n 为图 G 结点个数。文[4]将经典匹配理论和最新参数计算技术结合并扩展开来，最后提出了一个运行时间为 $O((k_u+k_l)|G|+1.26^{k_u+k_l})$ 的算法，它是当今有关二分图受约束最小点覆盖问题精确算法的最佳结果。

然而，上述算法中启发式算法无法在预定时间内给出问题的解答。精确算法虽然保证得到最优解，但是运行时间是

*）国家自然科学基金重点项目：生物信息学中的相关组合理论和算法研究（60433020）。**许小双** 硕士研究生，主要研究领域为参数计算、计算机理论；**王建新** 博士，教授，博士生导师，CCF 高级会员，主要研究领域为计算机算法、网络优化理论、生物信息学；**刘云龙** 硕士研究生，主要研究领域为生物信息学；**陈建二** 博士，长江学者特聘教授，主要研究领域为生物信息学、计算机理论、计算复杂性及优化。

指数复杂度的，因此难以适应大规模问题实例的求解。而近似算法虽然不能保证得到最优解，但它可以在多项式时间内得到近优解，从而成为在很多实际应用需要中所能接受的求解决方案。基于此，本文首先运用参数计算理论中降核心阶技术处理问题实例，而后通过利用二分图的经典匹配理论进一步深入分析二分图结构，充分利用链暗示技术来寻找受约束的近似点覆盖，从而寻找到求解该问题的一个近似率为 $1+\epsilon$ 的基于链暗示技术的二分图受约束最小点覆盖近似算法(Approximation Algorithm based on Chain Indication for constrained minimum vertex cover in bipartite graphs，简称 AACI 算法)，其运行时间复杂度为 $O(2^{2/\epsilon})$。同时，它也是二分图受约束最小点覆盖问题的一个多项式时间近似方案(polynomial time approximation scheme，PTAS 算法)。AACI 算法具体为：当 Min-CVCB 问题实例存在受约束最小点覆盖 (k_u, k_l) 时，对于任意给定的 $\delta=1+\epsilon$，一定可在 $O(2^{2/\epsilon})$ 时间内找到一个受约束近似点覆盖 (k_u^*, k_l^*)，其中 $\max\{k_u^*/k_u, k_l^*/k_l\}\leqslant 1+\epsilon$。因此在下文中我们假设给定问题实例的受约束最小点覆盖总是存在的。

本文后面部分的组织如下：第 2 部分给出了 Min-CVCB 问题与近似算法之间的关系；第 3 部分给出了 AACI 算法的相关定义和定理；第 4 部分是描述当 Min-CVCB 问题实例中存在一个受约束最小点覆盖时，如何利用 AACI 算法在 DAG D 中寻找到受约束近似点覆盖；第 5 部分是 AACI 算法各步综合及其近似率和时间复杂度分析；最后是结论。

2 Min-CVCB 问题与近似算法

我们首先给出近似算法理论中近似率的一些基本概念。这方面的详尽讨论可见文[12]。

定义 2 给定一个 NP 优化问题 P，我们说一个算法 A 的近似率为 $f(n)$ 是指：对于 P 的任何一个实例 I，算法 A 求出的目标函数值 $A(I)$ 与该实例的最优解 $Opt(I)$ 的比值满足：

$$\mathrm{Max}\{A(I)/Opt(I), Opt(I)/A(I)\}\leqslant f(Size|I|)$$

其中 $Size|I|$ 指实例 I 的规模，$f(|n|)$ 也可以为常数。通常，我们称一个近似率达到 $f(Size|I|)$ 的算法为 $f(Size|I|)$ 近似算法。

定义 3 设 x 为 NP 优化问题 P 的一个实例，ϵ 为一个大于 0 的常数，如果有一个算法 A 将 (x,ϵ) 作为输入，得到 x 的一个近似率为 $1+\epsilon$ 的可行解 y，而且对于任意固定大小的 $\epsilon>0$，算法 A 的运行时间为 $|x|$ 的多项式，则称算法 A 是 NP 优化问题 P 的一个 PTAS 算法。

根据前面对近似率和 PTAS 算法的介绍，由于 Min-CVCB 问题是一个 NP-优化问题，可对该问题的 PTAS 算法可作如下描述：

定义 4 设 (x, k_u, k_l) 为 Min-CVCB 问题的一个实例，ϵ 为一个大于 0 的常数，如果有一个算法 A_Q 将 (x, k_u, k_l, ϵ) 作为输入，则当 Min-CVCB 问题实例存在一个受约束最小点覆盖 (k_u, k_l) 时，均可得到 x 的一个近似率为 $1+\epsilon$ 的受约束近似点覆盖 (k'_u, k'_l)，它满足 $\max\{k'_u/k_u, k'_l/k_l\}\leqslant 1+\epsilon$，同时算法 A_Q 的时间复杂度为关于 $|x|$ 的多项式，则称算法 A_Q 是 Min-CVCB 问题的一个 PTAS 算法。

3 AACI 算法的相关定义与定理

Min-CVCB 问题的 AACI 算法包含以下两步：

(1)使用参数计算技术减小问题实例的核心阶[1]，同时结合使用二分图的经典匹配理论对问题实例作等价转换。减小问题实例核心阶的就是减少图 G 中顶点个数，定理 3.1 中用一个多项式时间算法把 Min-CVCB 问题实例转换为顶点个数至多为 $2(k'_u+k'_l)$ 且具有完美匹配二分图的等价问题。而定理 3.2 则是在此基础上利用经典匹配理论对问题实例做进一步等价转换，使其成为一个以基本二分子图为单位的有向无环图。

(2)通过利用链暗示技术对权值大于某一限定值的基本二分子图的所有可能覆盖情形进行列举来寻找受约束近似点覆盖。它使得在当 Min-CVCB 问题实例中存在一个受约束最小点覆盖时，一定可以在多项式时间内找到满足约束条件的近似点覆盖。

为便于我们进一步深入描述讨论 AACI 算法，在此先给出 AACI 算法的一些相关定义和定理。

定义 5 二分图 设无向图 $G=(V,E)$ 的结点集合 V 能分成两个子集 U 和 L，满足 $U\cup L=V, U\cap L=\varnothing$，而且对任意一条边 $e=(v_i,v_j)\in E$，均有 $v_i\in U, v_j\in L$，则称 G 为二分图，并称 V 和 U 为 G 的互补结点子集。在二分图中，对应的 U 中结点称为 U-部分结点(或简称 U-部点)，L 中结点称为 L-部分结点(或简称 L-部点)。

在拥有完美匹配的二分图 $G=(U\cup L,E)$ 中，如果每条边都包含在一个完美匹配中，则称图 G 是一个基本二分子图。由文[6]可知，基本二分子图有且仅有两个最小点覆盖 U 和 L，而没有其它可能性。显然，利用基本二分子图的特性可以极大提高寻找到受约束近似点覆盖的效率。

定理 3.1[4] 设 G 是一个由 n 个结点、m 条边组成的二分图，则解决一个 Min-CVCB 问题实例 $\langle G,k_u,k_l\rangle$ 的时间复杂度为 $O(mn^{1/2}+t(k_u+k_l))$，其中 $t(k_u+k_l)$ 是解决 Min-CVCB 问题实例 $\langle G',k'_u,k'_l\rangle$ 的时间，$k'_u<k_u, k'_l<k_l$，图 G' 有一个完美匹配，它至多包含 $2(k'_u+k'_l)$ 个结点。

定理 3.2[6] (The Dulmage-Mendelsohn Decomposition Theorem)具有完美匹配的二分图 $G=(U\cup L,E)$，可以在时间 $O(|E|^2)$ 内完成分解并标记为基本二分子图 $B_i=(U_i\cup L_i, E_i), i=1,2,\cdots,r$ 的有向无环图(简称 DAG D)，使得图 G 中每一条连接 B_i 到 B_j 的边 $(i<j)$ 有一端点在 B_i 的 U-部点，而另一端点在 B_j 的 L-部点。

通常，我们把经过 Dulmage-Mendelsohn 定理分解出来的基本二分子图 B_i 称为块。如果 $|U_i|=|L_i|=d_i$，则称 B_i 是一个权值为 d_i 的块。在 DAG D 中，连接不同块之间的边称为块间边，与块 B_i 的 U-部点相连的边数称为 B_i 的出度，与块 B_i 的 L-部点相连的边数称为 B_i 的入度。

定理 3.3[4] 设图 G 是一个具有完美匹配的二分图，B_1，B_2，…B_r 是图 G 经过 Dulmage-Mendelsohn Decomposition 定理分解之后的块，则图 G 的受约束最小点覆盖，就是块 B_1，B_2，…，B_r 最小点覆盖的并集。

通常情况下，在 Min-CVCB 问题实例 $\langle G,k_u,k_l\rangle$ 中，图 $G=(U\cup L,E)$ 是一个具有完美匹配的二分图。首先由定理 3.1 知，要解决一个 Min-CVCB 问题实例 $\langle G, k_u, k_l\rangle$，只需解决另一个 Min-CVCB 问题实例 $\langle G', k'_u, k'_l\rangle$，其中 G' 是一个拥有完美匹配的二分图，它至多有 $2(k'_u+k'_l)$ 个结点；然后由定理 3.2 可知，Min-CVCB 问题实例 $\langle G', k'_u, k'_l\rangle$ 可以分解成一个或多个 DAG D；由定理 3.3 可知，只要找到每个 DAG D 中满足约束条件的各个块的最小点覆盖，就可以解决 Min-

CVCB 问题实例$\langle G', k'_u, k'_l \rangle$。在将 Min-CVCB 问题实例做等价转换后，接下来我们开始在 DAGD 中寻找满足约束条件的近似点覆盖。

4 AACI-DAG D 算法

AACI-DAG D算法是在第 3 部分中将 Min-CVCB 问题实例做等价转换为 DAG D 的基础上来寻找受约束近似点覆盖(k_u^*, k_l^*)，它是通过对 DAG D 中权值大于或等于 $1+\varepsilon k$ 的块的覆盖情形的列举来实现的，其中 $k=\max\{k_u, k_l\}$。

在 DAG D 中，当每一个块的权值都小于 $1+\varepsilon k$ 时，则由 DAG D 的有向拓扑结构的性质可知：只需从 DAG D 中的各个块 B_i（其中 $1\leqslant i\leqslant k_u+k_l$）按下标从小到大的顺序将其 U-部点取入点覆盖中，使得点覆盖中 U-部点个数 k_u^* 满足 $k_u\leqslant k_u^*\leqslant(1+\varepsilon)k_u$，然后将剩余块的 L-部点放入点覆盖中，即可找到受约束近似点覆盖(k_u^*, k_l^*)。而当 DAG D 中包含有权值大于或等于 $1+\varepsilon k$ 的块时，对其中某一权值大于或等于 $1+\varepsilon k$ 的块，则可能当该块的未放入点覆盖中时，$k_u^*<k_u$，$k_l^*<k_l$；而取该块 L-部点到点覆盖中时，使得 $k_l^*>(1+\varepsilon)k_l$；当取该块 U-部点到点覆盖中时，使得 $k_u^*>(1+\varepsilon)k_u$。因而对权值大于或等于 $1+\varepsilon k$ 的块，唯有将其覆盖情形一一列举，才能找到可能存在的受约束近似点覆盖。在 AACI-DAG D 算法中，正是通过对权值大于或等于 $1+\varepsilon k$ 的块的覆盖情形一一列举来寻找受约束近似点覆盖。

在对权值大于或等于 $1+\varepsilon k$ 的块的覆盖情形列举过程中，我们将应用链暗示技术[4]，它是充分利用块的相邻关系来加速寻找目标点覆盖的过程。设$[B_1{}'B_2{}'\cdots B'_h]$是 DAG D 中一条有向路径，如果目标点覆盖 K 包含路径前面部分某块的 L-部点，则 K 一定包含中间部分与之相连块的 L-部点，也一定包含后面部分与之相连的相应块的 L-部点；如果 K 包含路径前面部分某块的 U-部点，则中间部分与之相连的块可能 U-部点包含在 K 中，也可能 L-部点包含在 K 中。而如果是 L-部点包含在 K 中，则后面部分与之相连的相应块 L-部点一定也包含在 K 中。链暗示技术就是利用块的相邻关系来加速寻找目标点覆盖的过程。

由链暗示技术可知：在 DAG D 中，当限定某块的 U-部点在点覆盖中时，所有通过有向路径可到达该块的块的 U-部点也一定在点覆盖中；而当限定某块的 L-部点在点覆盖中时，所有该块通过有向路径可到达的块的 L-部点也一定在点覆盖中。因此，AACI-DAG D 算法的具体操作分为如下两步：

step1. 在 DAG D 中，找出每一个权值大于或等于 $1+\varepsilon k$ 的块 B_i，然后找出所有可通过有向路径到达块 B_i 的块构成的块集合 C_i，以及该块通过有向路径可到达的所有块构成的集合 C'_i，其中 $1\leqslant i<2/\varepsilon$。

step2. 对 DAG D 中所有权值大于或等于 $1+\varepsilon k$ 的块的覆盖情形进行一一列举。对于已确定 U-部点在点覆盖中块，找出其对应的块集合 C_i，而对于已确定其 L-部点在点覆盖中块，也找出其对应的块集合 C'_i。假设我们在列举时用 B_s 表示限定其 U-部点在点覆盖中的权值大于或等于 $1+\varepsilon k$ 的块集合，C_s 表示块集合 B_s 中各个块在限定其 U-部点在点覆盖中时相对应的 C_i 构成的块集合；B'_r 表示限定其 L-部点在点覆盖中的权值大于或等于 $1+\varepsilon k$ 的块集合，C'_r 表示与块集合 B'_r 中各个块在限定其 L-部点在点覆盖中时相对应的 C'_i 构成的块集合，G_0 表示 DAG D 剩下的尚未确定其覆盖情形的块，则 $G_0=G\text{-}B_s\cup C_s\cup B_r{}'\cup C'_r$。若$(B_s\cup C_s)\cap(B'_r\cup C'_r)\neq\Phi$，则说明在 DAG D 中同时要求某块的 U-部点和 L-部点都包含在点覆盖中。此时产生"冲突"，可跳过本次列举，直接进行下一次列举。否则可根据$|B_s|+|C_s|$的大小，分为如下三种情况来处理：

case1：$k_u\leqslant|B_s|+|C_s|\leqslant(1+\varepsilon)k_u$ 或 $k_l\leqslant|B'_r|+|C'_r|\leqslant(1+\varepsilon)k_l$，则将 G_0 中各块 L-部点（或 U-部点）放入点覆盖中，即可找到受约束近似点覆盖，算法结束；

case2：$|B_s|+|C_s|>(1+\varepsilon)k_u$ 或 $|B'_r|+|C'_r|>(1+\varepsilon)k_l$，则此时不存在受约束近似点覆盖，从而进行下一次列举；

case3：$|B_s|+|C_s|<(1+\varepsilon)k_u$ 且 $|B'_r|+|C'_r|<(1+\varepsilon)k_l$，则此时 G_0 一定满足$|G_0|+|C_s|+|B_s|>(1+\varepsilon)k_u$。由于 G_0 中任意块的权值均小于 $1+\varepsilon k$，因而可在 G_0 中按各块下标从小到大取其 U-部点到点覆盖中，使得点覆盖中 U-部点个数 k_u^* 满足 $k_u\leqslant k_u^*\leqslant(1+\varepsilon)k_u$，然后将剩余块的 L-部点放到点覆盖中，即可找到受约束近似点覆盖，算法结束。

综上，在 DAG D 中寻找受约束近似点覆盖的 AACI-DAG D算法如图 1 所示：

输入：经过第三部分中 Dulmage-Mendelsohn Decomposition 定理分解生成的 DAG D, k_u, k_l, ε。
输出：DAG D 中的一个受约束近似点覆盖：它至多包含$(1+\varepsilon)k_u$个 U-部分结点和$(1+\varepsilon)k_l$个 L-部分结点。
1. 寻找 DAG D 中每一个权值大于或等于 $1+\varepsilon k$ 的块 B_i 及其对应 C_i'和 C_i'，其中 $1\leq i<2/\varepsilon$；
2. 对所有权值大于或等于 $1+\varepsilon k$ 的块的覆盖情况进行一一列举：若$(B_s\cup C_s)\cap(B_r'\cup C_r')\neq\varnothing$，则产生"冲突"，直接进行下一次列举；否则分以下三种情况来讨论：
case1: $k_u\leq\lvert B_s\rvert+\lvert C_s\rvert\leq(1+\varepsilon)k_u$ 或 $k_l\leq\lvert B_r'\rvert+\lvert C_r'\rvert\leq(1+\varepsilon)k_l$，则将 G_0 中各块 L 部点（或 U-部点）放到点覆盖中，算法结束；
case2: $\lvert B_s\rvert+\lvert C_s\rvert\geq(1+\varepsilon)k_u$ 或$\lvert B_r'\rvert+\lvert C_r'\rvert\geq(1+\varepsilon)k_l$，则直接进行下一次列举；
case3: $\lvert B_s\rvert+\lvert C_s\rvert<(1+\varepsilon)k_u$ 且$\lvert B_r'\rvert+\lvert C_r'\rvert<(1+\varepsilon)k_l$，则在 G_0 中按各块下标从小到大取其的 U-部点到点覆盖中，使得点覆盖中 U-部点个数 k_u^*满足 $k_u\leq k_u^*\leq(1+\varepsilon)k_u$，然后将 G_0 中剩余块的 L-部点放到点覆盖中，算法结束。

图 1　AACI-DAG D 算法

定理 4.1　当 Min-CVCB 问题实例存在一个受约束最小点覆盖(k_u, k_l)时，AACI-DAG D 算法寻找到的受约束近似点覆盖的近似率为 $1+\varepsilon$。

证明：当 Min-CVCB 问题实例存在一个受约束最小点覆盖(k_u, k_l)时，AACI-DAG D 算法是通过对权值大于或等于 $1+\varepsilon k$ 的块的覆盖情形的列举来寻找受约束近似点覆盖的。在 AACI-DAG D 算法中，step1 是对权值大于或等于 $1+\varepsilon k$ 的块进行预处理；step2 中开始寻找满足约束条件的近似点覆盖，其中 case1 和 case3 是能寻找到满足约束条件的近似点覆盖的情形，而 case2 则是不能找到时的情形。因此，我们主要是对 step2 中 case 1 和 case3 进行分析。case1 是当 $k_u\leqslant|B_s|+|C_s|\leqslant(1+\varepsilon)k_u$ 或 $k_l\leqslant|B'_r|+|C'_r|\leqslant(1+\varepsilon)k_l$ 时，将 G_0 中各块 L-部点（或 U-部点）放到点覆盖中，即可寻找到满足约束条件的近似点覆盖(k_u^*, k_l^*)。此时点覆盖(k_u^*, k_l^*)一定满足：$k_u^*\leqslant(1+\varepsilon)k_u$，$k_l^*\leqslant(1+\varepsilon)k_l$；case3 是当$|B_s|+|C_s|<(1+\varepsilon)k_u$ 并且$|B'_r|+|C'_r|<(1+\varepsilon)k_l$ 时，此时 G_0 一定满足$|G_0|+|C_s|+|B_s|>(1+\varepsilon)k_u$。由于 G_0 中任意块的权值均小于 $1+\varepsilon k$，故可在 G_0 中按各块下标从小到大顺序取其 U-部点到点覆盖中，使得点覆盖中 U-部点个数 k_u^* 满足 $k_u\leqslant k_u^*\leqslant(1+\varepsilon)k_u$。而后将 G_0 中剩余块 L-部点放到点覆盖中，显然此时寻找到的点覆盖(k_u^*, k_l^*)也一定满足：$k_u^*\leqslant(1+\varepsilon)k_u$，$k_l^*\leqslant(1+\varepsilon)k_l$。因此，当 Min-CVCB 问题实例存在一个受约束最小点覆盖时，使用 AACI-DAG D 算法所寻找到的受约束近似

点覆盖的近似率为 $\max\{k_u^*/k_u, k_l^*/k_l\} \leqslant 1+\varepsilon$。

定理 4.2 AACI-DAG D 算法的时间复杂度为 $O(2^{2/\varepsilon})$。

证明：在 AACI-DAG D 算法中，我们是通过对 DAG D 中权值大于或等于 $1+\varepsilon k$ 的块的覆盖情形的列举来寻找受约束近似点覆盖的。在 DAG D 中包含的权值大于或等于 $1+\varepsilon k$ 的块个数至多为 $2/\varepsilon$，而每一个块的覆盖仅有两种可能：要么 U-部点在点覆盖中，要么 L-部点在点覆盖中。因此在 AACI-DAG D 算法的 step2 中，对权值大于或等于 $1+\varepsilon k$ 块的覆盖情况，最多有 $2^{2/\varepsilon}$ 次列举，而对于剩下的权值小于 $1+\varepsilon k$ 的块，可在线性时间内决定其覆盖情形。又因为 step1 中寻找每一个权值大于或等于 $1+\varepsilon k$ 的块 B_i 及其对应的 C_i 和 C'_i 也是在线性时间内完成的，故 AACI-DAG D 算法的时间复杂度为 $O(2^{2/\varepsilon})$。

5 AACI 算法各步综合及其分析

综合前面各个部分的讨论，Min-CVCB 问题的 AACI 算法的各步实现过程如图 2 所示。

输入：二分图 G=(U, L, E)，两个整数 k_u, k_l, ε；
输出：图 G 的一个受约束近似点覆盖 K，它至多有 $(1+\varepsilon)k_u$ 个 U-部分结点，$(1+\varepsilon)k_l$ 个 L-部分结点。

1. K=$\varnothing$；
2. 对于 U-部分中度大于 k_l 的结点，包含到 K 中并将其从图 G 中去掉，k_u=k_u-1；
3. 对于 L-部分中度大于 k_u 的结点，包含到 K 中并将其从图 G 中去掉，k_l=k_l-1；
4. 应用定理 3.1 来将 Min-CVCB 问题实例转换为一个具有完美匹配的二分图 G，它至多有 $2(k_u+k_l)$ 个结点，(k_u, k_l) 以及 K 都是更新之后的；
5. 应用 Dulmage-Mendelsohn Decomposition 定理将图 G 分解成块 B_1, B_2,…B_r 的有向无环图(DAG D)；
6. 使用第四部分中 AACI-DAG D 算法在 DAG D 中寻找满足约束条件的近似点覆盖。

图 2 Min-CVCB 问题的 AACI 算法各步综合示意图

以上各步中，第一步是点覆盖 K 初始化，第二步对图 G 中 U-部分中高度结点的处理。如果 U-部分中有度大于 k_l 的结点不包含在 K 中，则其 L-部分的邻接点都在 K 中，显然此时没有满足条件的点覆盖存在。对于第三步中，对图 G 中 L-部分的高度结点也有类似处理。如果此时 k_u 或者 k_l 变为负数，则返回不存在满足条件的点覆盖。处理完高度结点后，剩下图中结点数至多 $(2k_u^* k_l+k_u+k_l)$，边数至多为 $k'(k_u+k_l) \leqslant (k_u+k_l)^2$，其中 $k'=\max\{k_u, k_l\}$。第四步是应用定理 3.1 来将问题实例转化为 $2(k_u+k_l)$ 大小的等价问题。经过第四步后，核心阶大小至多为 $2(k_u+k_l)$。第五步是应用 Dulmage-Mendelsohn Decomposition 定理将图 G 的核心阶分解成块 $B_1, B_2, \cdots B_r$ 的有向无环图 (DAG D)。以上五步是对 Min-CVCB 问题实例的做等价转化处理。第六步是是运用第 AACI-DAG D 算法在 DAG D 中寻找到满足约束条件的近似点覆盖 (k_u^*, k_l^*)，它是整个 AACI 算法的核心部分。

定理 5.1 AACI 算法的时间复杂度为 $O(2/^{2/\varepsilon})$。

证明：在图 2 中 AACI 算法各步对应的时间复杂度如下：第一步点覆盖 K 初始化和第二、三步对图 G 中高度结点的处理的时间复杂度为 $O((k_u+k_l)|G|)$；第四步中应用定理 3.1 减小问题实例核心阶的时间复杂度为 $O((k_u+k_l)^3)$；第五步中应用 Dulmage-Mendelsohn Decomposition 定理将图 G 分解成块 $B_1, B_2, \cdots, B_r$ 的 DAG D 的时间复杂度为 $O(|E|^2)$，其中 $|E|$ 是图 G 中边的条数。显然 $|E| \leqslant (k_u+k_l)^2$，也即第五步对应的时间复杂度至多为 $O((k_u+k_l)^4)$；第六步则是运用 AACI-DAG D 算法在 DAG D 中寻找到满足约束条件的近似点覆盖，由定理 4.2 可知，其对应的时间复杂度为 $O(2/^{2/\varepsilon})$。综上可得，AACI 算法的时间复杂度为

$$O((k_u+k_l)|G|)+(k_u+k_l)^3+(k_u+k_l)^4+2^{2/\varepsilon})=O(2^{2/\varepsilon})。$$

定理 5.2 当 Min-CVCB 问题实例存在一个受约束最小点覆盖 (k_u, k_l) 时，AACI 算法的寻找到的受约束近似点覆盖的近似率为 $1+\varepsilon$。

证明：在图 2 中，AACI 算法的前五步是 Min-CVCB 问题实例的等价转换，因此它对近似率没有影响。在第六步，开始在 DAG D 中寻找满足约束条件的近似点覆盖，由定理 4.1 可知：当 Min-CVCB 问题实例存在一个受约束最小点覆盖时，AACI-DAG D 算法的近似率为 $1+\varepsilon$，也即 AACI 算法的近似率为 $1+\varepsilon$。故当 Min-CVCB 问题实例存在一个受约束最小点覆盖时，AACI 算法的寻找到的受约束近似点覆盖的近似率为 $1+\varepsilon$。

定理 5.3 如果 Min-CVCB 问题实例存在一个受约束最小点覆盖 (k_u, k_l)，则 AACI 算法一定可在 $O(2^{2/\varepsilon})$ 时间内找到一个近似率为 $1+\varepsilon$ 的受约束近似点覆盖 (k_u^*, k_l^*)，即 AACI 算法是 Min-CVCB 问题的一个运行时间为 $O(2^{2/\varepsilon})$ 的 PTAS 算法。

证明：对于图 G 中的一个受约束最小点覆盖 (k_u, k_l)，需要寻找的点覆盖是在图 G 中 U-部点个数不超过 k_u，L-部点个数不超过 k_l 的最小点覆盖。而对于图 G 中一个近似率为 $1+\varepsilon$ 的受约束近似点覆盖 (k_u^*, k_l^*)，需要寻找的是在图 G 中 U-部点个数不超过 $(1+\varepsilon)k_u$，L-部点个数不超过 $(1+\varepsilon)k_l$ 的点覆盖。

当 Min-CVCB 问题实例存在一个受约束最小点覆盖时，DAG D 中每个块的覆盖情形都是确定的：仅取其 U-部点或 L-部点之一在点覆盖中。由前面的 AACI-DAG D 算法可知，在寻找受约束近似点覆盖的过程中，难点是寻找权值大于或等于 $1+\varepsilon k$ 的块的覆盖情形。由于在 DAG D 中包含的权值大于或等于 $1+\varepsilon k$ 的块个数至多为 $2/\varepsilon$，因此在 AACI-DAG D 算法中最多只需要列举 $2^{2/\varepsilon}$ 次，即可找到受约束最小点覆盖中各个权值大于或等于 $1+\varepsilon k$ 的块的覆盖情形；而对于剩下的权值小于 $1+\varepsilon k$ 的块，则可在线性时间内找到对应的覆盖情形。

在 AACI-DAG D 算法的具体执行中，step2 开始寻找受约束近似点覆盖。当 $(B_s \cup C_s) \cap (B'_r \cup C'_r) \neq \Phi$ 时，说明要求某块要使其 U-部点和 L-部点同时在点覆盖中，显然此时一定不会有受约束最小点覆盖，从而跳过本次列举对最后寻找近似点覆盖不会产生影响；在 step2 的 case2 中，当 $|B_s|+|C_s|>(1+\varepsilon)k_u$ 或 $|B'_r|+|C'_r|>(1+\varepsilon)k_l$，也一定不会存在一个受约束最小点覆盖，从而跳过本次列举也不会对最后寻找受约束近似点覆盖产生影响。排除掉以上两种不存在受约束近似点覆盖的情形，仅剩下的 case1 和 case3 中，由算法的具体过程可知：如果满足判定条件就一定可找到受约束近似点覆盖。因此，只要 Min-CVCB 问题实例存在一个受约束最小点覆盖，AACI 算法一定可找到一个受约近似束点覆盖。由定理 5.2 可知：AACI 算法所得近似率为 $1+\varepsilon$，也即寻找到的受约束点覆盖近似率为 $1+\varepsilon$。又由定理 5.1 可知：AACI 算法时间复杂度为 $O(2^{2/\varepsilon})$。

（下转第 282 页）

由表2、表3数据分析可得：LBMR算法与UMR、MI、XMI同类算法相比，60%以上实验结果性能得到明显提高。同UMR算法相比，LBMR算法可根据网络带宽实际情况，选择不同的多路调度算法来完成任务分配、计算，更加符合实际计算环境需求；与MI、XMI算法相比，除算法的自适应性更强以外，LBMR算法可以近似求解调度所需的最优路数M，而MI、XMI算法只能人为指定循环路数大小，任务调度性能不能得到保证。

结　论　UMR算法是一种高效可分割任务调度算法，该算法实现了任务调度时间跨度最优化，给定了求解最优路数M的方法。由于实际网络带宽的有限性，大大限制了UMR算法的应用范围，本文在分析UMR算法基础上，引入网络带宽约束条件，提出了适用于有限网络带宽的可分割任务调度算法LBMR。该算法将结点的处理能力和网络流情况协同考虑，分别针对充足带宽和受限带宽提出相应调度算法，相比单纯的充足带宽多路算法和受限带宽多路算法，本文所提算法避免了纯充足带宽多路算法在处理任务时不能满足可适应性问题，又克服了纯受限带宽多路算法在任务处理过程中工作节点利用率低等问题的缺点。实验表明LBMR算法较先前提出的可分割任务调度算法，调度性能明显提高，该算法易于调度性强，具有更高的应用价值。

参考文献

1 Davis T, Chalmers A, Jensen H W. Practical parallel processing for realistic rendering. ACM SIGGRAPH, 2000

2 Lee C, Hamdi M. Parallel Image Processing Applications on a Network of Workstations. Parallel Computing, 1995, 21: 137～160

3 Altilar D, Paker Y. An Optimal Scheduling Algorithm for Parallel Video Processing. In: Proceedings of the IEEE International Conference on Multimedia Computing and Systems, 1998

4 Blazewicz J, Drozdowski M, Markiewicz M. Divisible Task Scheduling - Concept and Verification. Parallel Computing, 1999, 25: 87～98

5 Drozdowski M, Wolniewicz P. Experiments with Scheduling Divisible Tasks in Clusters of Workstations. In: Proceedings of Europar'2000, 2000. 311～319

6 Rosenberg A L. Sharing Partitionable Workloads in Heterogeneous NOWs: Greedier Is Not Better. In: Proceedings of the 3rd IEEE International Conference on Cluster Computing, 2001. 124～131

7 Hummel S F. Factoring: a Method for Scheduling Parallel Loops. Communications of the ACM, 1992, 35(8): 90～101

8 Yang Y, Casanova H. UMR: A Multi-Round Algorithm for Scheduling Divisible Workloads. In: Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'03), 2003

9 Yang Y, Casanova H. Multi-Round Algorithm for Scheduling Divisible Workload Applications: Analysis and Experimental Evaluation: [Technical Report CS2002-0721]. Dept. of Computer Science and Engineering, 2002

10 Bharadwaj V, Ghose D, Mani V, Robertazzi T G. Scheduling Divisible Loads in Parallel and Distributed Systems, chapter 10. IEEE Computer Society Press, 1996

11 Hagerup T. Allocating Independent Tasks to Parallel Processors: An Experimental Study. Journal of Parallel and Distributed Computing, 1997, 47: 185～197

12 Beaumont O, Carter L, Ferrante J, Legrand A, Robert Y. Bandwidth-centric allocation of independent tasks on heterogeneous platforms. In: Proc. Int'l Parallel and Distributed Processing Symp., 2002

13 Buyyal R, Murshed M. GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing. Concerrency and computation: Practice and experience, 2002, 14: 1175～1220

14 Yang Y, Casanova H. Extensions to the Multi-Installment Algorithm: Affine Costs and Output Data Transfers: [Technical Report CS2003-0754]. 2003

15 Howell F, McNab R. SimJava: A Discrete Event Simulation Package For Java With Applications In Computer Systems Modelling. In: First International Conference on Web-based Modelling and Simulation, Society for Computer Simulation, 1998

（上接第273页）

综上可知，对一个Min-CVCB问题实例，如果存在一个受约束的最小点覆盖，则通过AACI算法一定可在$O(2^{2/\epsilon})$时间内找到一个近似率为$1+\epsilon$的受约束点覆盖。

显然，AACI算法满足前面定义4，故它是Min-CVCB问题的一个运行时间为$O(2^{2/\epsilon})$的PTAS算法。

结论　本文主要是研究在VLSI芯片生产中有着广泛应用的Min-CVCB问题。由于传统的启发式算法和精确算法都不能满足实际应用的需要，本文从一个全新角度出发，提出了一个近似率为$1+\epsilon$的近似算法。它首先运用参数计算理论中降核心阶技术处理问题实例，然后利用二分图的经典匹配理论进一步深入分析二分图结构，最后充分利用链暗示技术通过穷举寻找到受约束的近似点覆盖。具体为：当Min-CVCB问题实例存在受约束最小点覆盖时，对于任意给定的$\delta=1+\epsilon>1$，一定可以在$O(2^{2/\epsilon})$时间内找到一个近似率为$1+\epsilon$的受约束近似点覆盖。同时，它也是Min-CVCB问题的一个运行时间复杂度为$O(2^{2/\epsilon})$的PTAS算法。显然，当近似率$1+\epsilon$大小一定时，随着问题规模增大，AACI算法的时间复杂度不变；而当问题实例的规模一定时，随着近似率$1+\epsilon$的减小，AACI算法的时间复杂度越高。

由AACI-DAG D算法中的step2中case1和case3中判定过程可知：DAG D中每一个块都是仅取其U-部点或L-部点之一到受约束近似点覆盖中，故最后找到的近似点覆盖也是一个最小点覆盖，即$k_u^*+k_l^*=k_u+k_l$。它与原来受约束最小点覆盖(k_u,k_l)的区别是：(k_u^*,k_l^*)的取值范围比(k_u,k_l)要大，其中$k_u^*\leqslant(1+\epsilon)k_u$，$k_u^*\leqslant(1+\epsilon)k_u$，但时间复杂度却由指数时间降为多项式时间。目前，我们正在通过进一步深入运用本文中AACI算法，来处理Min-CVCB问题实例不存在受约束最小点覆盖时，如何快速寻找到Min-CVCB问题受约束近似解。

参考文献

1 Downey R G, Fellows M R. Parameterized Complexity. Springer, New York, 1999

2 Kuo S-Y, Fuchs W. Efficient spare allocation for reconfigurable arrays. IEEE Design and Test, 1987, 4: 24～31

3 Hasan N, Liu C L. Minimum Fault Coverage in Reconfigurable Arrays. In: Proceedings of the 18th International Sysmposium on Fault-Tolerant Computing (FTCS'88), IEEE Computer Society Press, Los Alamitos, CA, 1988. 348～353

4 Chen J, Kanj I A. Constrained minimum vertex cover in bipartite graphs: complexity and parameterized algorithms. Journal of Computer and System Science, 2003, 67: 833～847

5 Fernau H, Niedermeier R. An efficient exact algorithm for constraint bipartite vertex cover. Algorithms, 2001, 38: 374～410

6 Lovasz L, Plummer MD. Matching Theory. Annals of Discrete Mathematics, Vol 29. North-Holland, Amsterdam, 1986

7 Blough D M. On the reconfigurable of memory arrays containing clustered faults. In Fault Tolerant Computing, pages, Los Alamitos, Ca., USA, IEEE Computer Society Press, June 1991. 444～451

8 Blough D M, Pelc A. Complexity of fault diagnosis in comparison models. IEEE Trans Comput, 1992, 41(3): 318～323

9 Hasan N, Liu C L. Fault covers in reconfigurable PLAs [A]. In: 20th Int Symp on Fault-Tloerant Computing (FTCS'90), IEEE Computer society Press, 1990. 166～173

10 Low C P, Leong H W. A new class of efficient algorithms for reconfiguration of memery arrays. IEEE Trans Comput, 1996, 45 (5): 614～618

11 Smith M D, Mazumder P. Generation of minimal vertex cover for row/column allocation in self-repairable arrays [J]. IEEE Trans Comput, 1996, 45: 109～115

12 Cormen T H, Leiserson C E, Rovest R L. Introduction to Algorithms. New York: McGraw-Hill Book Company, 1992