考虑多冗余通孔插入的 DSA 引导槽分配

李兴权[1], 曾艺玲[1], 朱文兴[2]

(1.闽南师范大学 数学与统计学院, 福建 漳州 363000; 2.福州大学 离散数学研究中心, 福建 福州 350108)

摘　要：作为一种新兴的制造技术,嵌段共聚物定向自组装(DSA)有望用于超大规模集成电路通孔层制造. 同时,冗余通孔插入被认为是提高产量的关键步骤. 为了获得更好的可靠性和可制造性,本文同时考虑在布线后阶段考虑多冗余通孔插入的 DSA 引导槽分配. 首先,通过分析引导槽的结构特性,提出了一种基于积木块的解表达方式. 然后,遵循紧凑的解表达方式,本文为版图构造冲突图,然后将问题描述成整数线性规划. 为了更好的平衡解的质量和运行时间,本文将整数线性规划松弛成一个无约束非线性规划问题. 最后,本文提出了一种基于线搜索的优化迭代算法来求解无约束非线性规划. 实验结果验证了本文的解表达方式的有效性和提出的算法的效率.

关键词：VLSI 通孔层；多冗余通孔插入；DSA 引导槽分配；整数线性规划；线搜索优化算法

中图分类号：TU4　**文献标志码**：A　**文章编号**：2095-7122(2019)01-0034-10

DOI:10.16007/j.cnki.issn2095-7122.2019.01.005

DSA Guiding Template Assignment with Multiple Redundant via Insertion

LI Xingquan[1], ZENG Yiling[1], ZHU Wenxing[2]

(1.School of Mathematics and Statistics, Minnan Normal University, Zhangzhou, Fujian 363000, China; 2.Center for Discrete Mathematics and Theoretical Computer Science, Fuzhou University, Fuzhou, Fujian 350108, China)

Abstract: As an emerging manufacture technology, block copolymer directed self-assembly(DSA) is promising for via layer fabrication. Meanwhile, redundant via insertion is considered as an essential step for yield improvement. For better reliability and manufacturability, in this paper, we concurrently consider DSA guiding template assignment with multiple redundant via insertion at post-routing stage. Firstly, by analyzing the structure property of guiding templates, we propose a building-block based solution expression to discard redundant solutions. Then, honoring the compact solution expression, we construct a conflict graph, and then formulate the problem to an integer linear programming(ILP). To make a good trade-off between solution quality and runtime, we relax the ILP to an unconstrained nonlinear programming (UNP). Finally, a line search optimization algorithm is proposed to solve the UNP. Experimental results verify the effectiveness of our new solution expression and the efficiency of our proposed algorithm.

Key words: VLSI via layer; multiple redundant via insertion; DSA guiding template assignment; integer linear programming; line search optimization algorithm

作为一种新兴技术,嵌段共聚物定向自组装技术(DSA)被认为是最有希望的超大规模集成电路(VLSI)通孔层制造技术[1-2]. 已有的工作在 DSA 的制造,建模,仿真和图形外延方面取得了许多显着的进步[3]. 这些改进使 DSA 技术能够模拟制造通孔. 要使用 DSA 技术制造分布不规则的通孔,需要用引导槽来引导通孔[4-5]. 这些引导槽将由传统的光刻技术制造,因此引导槽的制造质量受到引导槽间距的限制.

由于随机缺陷,切割不对称,电迁移和热/机械应力等原因[6],单个通孔可能发生故障. 通孔故障会严重

收稿日期：2019-02-27
基金项目：国家自然科学基金面上项目(61672005)；闽南师范大学科研启动项目(KJ18009)
作者简介：李兴权(1990-)，男，福建省龙岩市人，讲师.

影响集成电路的功能和产量[7–8]. 因此，为了提高电路可靠性和良率[9–10]，插入冗余通孔被认为是减少通孔故障的必要步骤. 冗余通孔插入技术是在每个通孔旁边插入一个冗余通孔[11]. 此外，插入的冗余通孔不能产生电路短路，也就是说，插入的冗余通孔不能与来自其他线网的任何金属线重叠. 如图 1a 所示，给定具有三个通孔的布局，可以分别为通孔 v_2 和 v_3 插入冗余通孔 r_2 和 r_3. 由于通孔 v_1 的周围位置是非法的，不能为 v_1 插入任何冗余通孔.

为了提高分辨率，可以将相邻的通孔(包括冗余通孔)放入多孔引导槽[12–14]. 基本上，应该为每个通道和冗余通道分配一个引导槽. 图 1b 显示了图 1a 中布局的引导槽分配. 在图 1b 中，通孔 v_3 和冗余通孔 r_3 分配给 1×2 孔槽，通孔 v_1 由单孔槽引导，而由于间距限制，通孔 v_2 和冗余通孔 r_2 不能被引导和制造. 理想的结果是将图 1a 中的 5 通孔和冗余通孔分配到相同的引导槽以进行制造，如图 1c 所示. 然而，图 1c 中的这种不规则引导槽很容易产生重叠误差，为了保证精度，只有一些形状规则的引导槽可供使用.

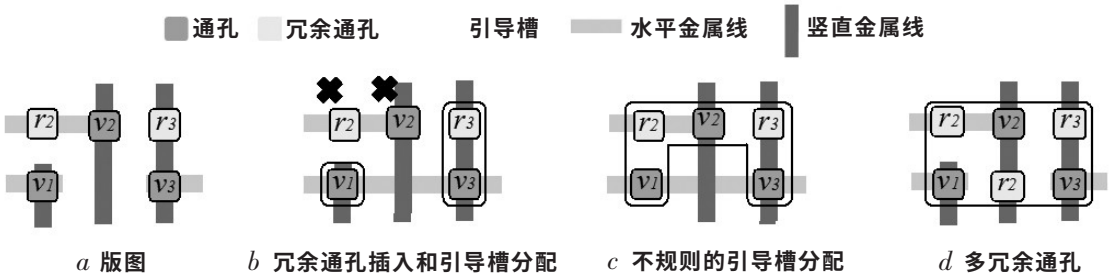

图 1 多冗余通孔插入和引导槽分配

Fig. 1 Multiple redundant via insertion and guiding template assignment

在传统的设计过程中，冗余通孔插入和通孔层的制造在两个独立的阶段中处理. Fang 等[15]首先同时考虑了冗余通过插入和 DSA 引导槽分配问题. 对于这种同时考虑，可插入的通孔的数量 (Insertion Rate, IR)和可制造的通孔的数量(Manufacture Rate, MR)都增加. 为了提高插入率和制造率，以前的工作[16–19]中的主流技术可以包括以下两种类型：1) 通过多重图案化来疏松通孔间距[16–17]；2) 提高冗余通孔和引导槽的自由度[18–19]. 对于方式 2)，Fang 等[18]研究了弯曲金属线的冗余通孔插入和 DSA 引导槽分配问题，通过局部扰动一些金属线，它以增加线长为代价插入冗余通孔. 但在先进的 1–D 金属层设计中，弯曲金属线是不可靠的. 为了避免这个缺点，Hung 等[19]研究了虚拟通孔插入的问题，通过插入了一些虚拟的通孔来辅助形成可用的引导槽.

为了进一步提高通孔插入率 IR 和制造率 MR，本文考虑多冗余通孔插入问题 DSA 引导槽分配问题，如图 1 中的情况，如果为通孔 v_2 插入两个冗余通孔 r_2，如图 1d 所示，则这 6 个通孔和冗余通孔可以由一个 2×3 规则槽引导. 本文引入积木块来表达一个 DSA 引导槽分配解. 在积木块的帮助下，本文将考虑多冗余通孔插入的 DSA 引导槽分配问题建模成一个基于冲突图的整数线性规划问题. 通过引入 Sigmoid 函数，本文进一步将整数线性规划问题放松成一个无约束非线性规划，以便在解的质量和运行时间之间实行较好的权衡. 最后，本文引入了一个基于线搜索的优化算法来求解无约束非线性规划，该算法可以实现局部最优解.

1　准备工作

1.1　引导槽分配

对于 DSA 技术，引导槽用于指导通孔在 DSA 技术下的制造. 由于不规则的引导槽产生重叠误差的概

率较高,为保证重叠精度,只有一些少孔且规则的引导槽才能被使用. 图 2 列出了本文可使用的七种类型的引导槽. 此外, 为了保证引导槽的分辨率和焦深, 相邻引导槽之间的间距不应小于最小光学极限间距 d_s. 通常,d_s 设置为不小于冗余通孔与其相关通孔之间的距离,并且不小于引导槽中的孔间距. 需要决定引导槽的分配,以便引导槽可以包围更多的通孔和冗余通孔.

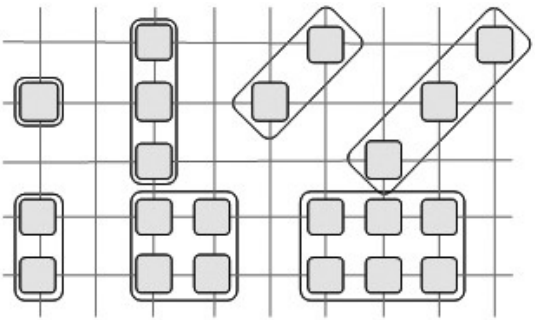

图 2 七种可用的引导槽

Fig. 2 Seven type available guiding templates

1.2 问题陈述

考虑多冗余通孔插入的 DSA 引导槽分配问题是为每个通孔插入一个或多个冗余通孔,并通过 DSA 技术下制造所有通孔及其冗余通孔,使得冗余通孔的插入率 IR 和制造率 MR 最大. 正式的问题描述如下.

给定后布线阶段的通孔层版图、可使用的引导槽类型和最小光学分辨率极限间距 d_s;需要为每个通孔插入一个或多个冗余通孔,并将每个通孔和冗余通孔分配给某个可使用的引导槽,使得:1)插入的冗余通孔位置是合法的;2)相邻引导槽之间的间距大于 d_s;目标是为了最大化 MR+β·IR,其中 β 是加权参数.

2 构造冲突图

在找到所有候选的冗余通孔之后,已有的工作[18-19]先检查所有可能的引导槽分配方案,然后用引导槽分配方案来表示解. 但是检测所有可能的引导槽分配方案将花费很多计算时间. 而且引导槽分配方案是指数级的,求解 DSA 引导槽分配问题是非常耗时的.

为了获得紧凑的解决方案表达式, 本文引入了积木块的概念. 积木块由一些通孔和候选冗余通孔组成,并且积木块可用于组成各种类型的引导槽. 图 3 中示出了九种类型的积木块,其中积木 1 包含一个通孔; 积木 2 包含一个冗余通孔;积木 3 包含一个通孔和一个冗余通孔;积木 4 包括两个通孔;积木 5 包括两个冗余通孔;积木 6 包含一个通孔和一个冗余通孔在对角位置;积木 7 包含两个通孔在对角位置;积木 8 包含两个冗余通孔在对角位置;积木 9 包含六个通孔或冗余通孔形成一个 2×3 引导槽形状. 然后,通过对积木块进行组合,可以形成图 2 中的七种类型的可用引导槽,如图 4 所示(仅列出竖直情况).

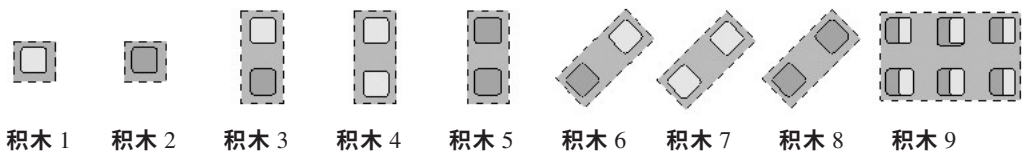

图 3 九种积木块

Fig. 3 Night type building-blocks

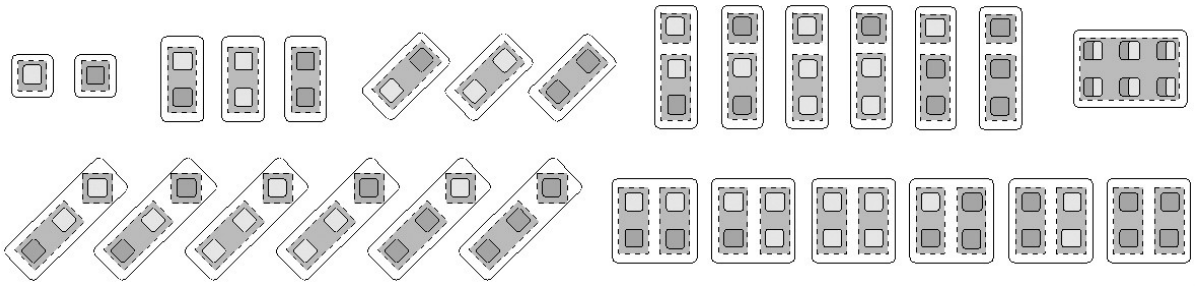

图 4 所有可能的积木组合

Fig. 4 All possible combinations of building-blocks

给定图 5a 所示的版图，假设检测出所有可能的候选冗余通孔如图 5b. 根据本文的积木块规则，可以识别出如图 5c 所示的 5 个积木块 a、b、c、d、e，这些积木块可以看作是冲突图中的顶点. 基于这些顶点，在它们之间引入了一些边.

定义 1(重叠边)如果积木 i 和 j 彼此重叠，则它们之间存在重叠边 e_{ij}，设 E_O 为重叠边的集合.

定义 2(冲突边)如果两个积木 i 和 j 之间的距离小于 ds，并且它们之间不存在重叠边，则积木 i 和 j 之间存在冲突边 e_{ij}，设 E_C 为冲突边的集合.

定义 3(引导槽边)对于两个积木 i 和 j，如果可以将 i 和 j 分配给相同的引导槽而不违反任何设计规则，并且 i 和 j 之间存在冲突边 $e_{ij} \in E_C$，则 e_{ij} 也称为引导槽边，设 E_T 是引导槽边的集合. 显然，$E_T \subseteq E_C$.

定义 4(冲突图)冲突图 $G(V, E)$是无向图，其中顶点是积木块，$e_{ij} \in E$ 是边，$E=(E_C-E_T) \cup E_O$. E_C，E_T 和 E_O 分别是冲突边、引导槽边和重叠边的集合.

图 5d 是版图 5b 的冲突图，在图 5d 中，黑色边是重叠边，红色边是连续边，绿色虚线边是引导槽边. 从图 5b 中，可以知道积木块 b 和 c 在通孔 v_2 处相互重叠，因此在它们之间存在重叠边，如图 5d 所示. 积木块 b 和 e 之间的距离不大于 d_s，因此它们之间存在一个冲突边. 由于积木块 b 和 d 可以分配到 1×3 孔引导槽，如图 5d 所示，因此在 a 和 c 之间有一个引导槽边.

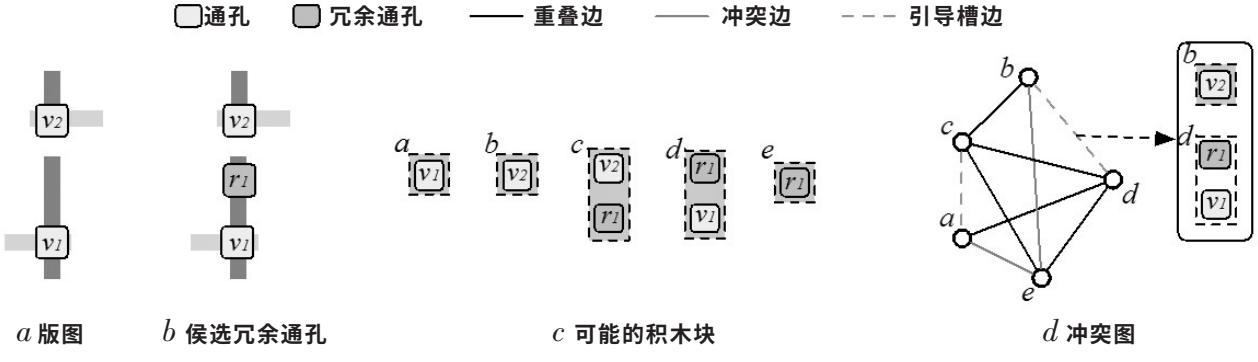

图 5 冲突图

Fig. 5 Conflict graph

3　局部最优算法

本节首先将原问题描述成一个整数线性规划，接着，将整数线性规划松弛成一个无约束非线性规划问题，并提出基于线搜索的优化算法来求解无约束非线性规划.

3.1 整数线性规划

在冲突图中,如果两个积木 i 和 j 彼此重叠,即 $e_{ij}\in E_O$,那么它们中只有一个可以被制造. 另外,如果两个积木的间距小于 d_s,即 $e_{ij}\in E_C$,则由于分辨率的限制,它们中的也仅有一个能被制造,除非将两个积木可以被分配到相同的引导槽中,即 $e_{ij}\in E_T$.

如果两个积木 i 和 j 之间有引导槽边,则它们可能被分配给相同的引导槽. 特别地,对于图 6a 中所示的结构,积木 i 和 j 通过两个引导槽边连接到 k,但是 i,k 和 l 不能同时分配到同一个引导槽,因为没有一个这样形状的引导槽(图 6b 和图 6c 与图 6a 类似). 本文称这些无序三元组(i,k,l)为冲突结构,其定义如下.

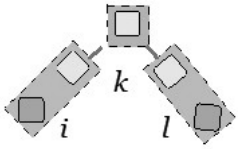

图 6 冲突结构

Fig. 6 Conflict structure

定义 5(冲突结构,Conflict Structure)该结构由三个积木块 i,k 和 l 组成,其中 e_{ik} 和 e_{kl} 是引导槽边,并且 i 和 l 之间不存在任何边,设 CS 表示为冲突结构的集合.

原问题的目的是最大化 MR+$\beta\cdot$IR, 即最大化可制造通孔数量和插入冗余通孔数量的加权和. 不同的积木块包括不同数量的通孔和冗余通孔. 假设通孔和冗余通孔的权重分别为 1 和 β. 通过给每个积木块 i 设置如下权重 w_i 来同时最大化 MR 和 IR.

$$w_i=N_v+\beta\cdot N_r, \tag{1}$$

其中,N_v 和 N_r 是积木块 i 包含的通孔和冗余通孔的数量,令 W 是每个积木块权重的集合,则冲突图可以写成加权形式 $G(V, E, W)$. 给出原问题的一个整数线性规划形式.

$$\max \quad \sum_{i\in V} w_i x_i , \tag{2}$$

$$s.t. \quad x_i + x_j \le 1, \qquad \forall e_{ij}\in E; \tag{2a}$$

$$x_i + x_k + x_l \le 2, \quad \forall (i,k,l)\in CS; \tag{2b}$$

$$x_i \in \{0,1\} \qquad \forall i\in V, \tag{2c}$$

在上述整数线性规划中,约束(2a)表示如果积木块 i 和 j 之间有边 $e_{ij}\in E_O$ 或者 $e_{ij}\in E_C$–E_T,那么积木块 i 和 j 至多只有一个可以被制造;约束(2b)表示如果积木块 i,k 和 l 是冲突结构,那么 i,k 和 l 至多只有两个可以被制造.

3.2 局部最优算法

由于求解整数线性规划是非常耗时的,本文将上述问题(2)转化成一个无约束非线性规划问题,并提出一个基于线搜索的算法进行求解. 首先上述整数规划问题(2)可以写成如下等价形式.

$$\max \quad \sum_{i\in V} w_i x_i , \tag{3}$$

$$s.t. \quad x_i x_j = 0, \qquad \forall e_{ij}\in E; \tag{3a}$$

$$x_i x_k x_l = 0, \quad \forall (i,k,l) \in CS; \tag{3b}$$

$$x_i \in \{0, 1\} \quad \forall i \in V, \tag{3c}$$

其中 $\boldsymbol{w} = (w_1, w_2, \ldots, w_n)^{\mathrm{T}} \in R^n$, $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^{\mathrm{T}} \in \{0, 1\}^n$, $n = |V|$. 因为约束(3*a*)和(3*b*)是等式，它们可以直接被整合到目标函数. 那么问题(3)可以进一步被写成如下非线性规划.

$$\max \quad \sum_{i \in V} \{w_i x_i \prod_{\substack{j \in V \\ e_{ij} \in E}} (1 - x_j) \prod_{\substack{k,l \in V \\ (i,k,l) \in CS}} (1 - x_k x_l)\}, \tag{4}$$

$$s.t. \quad x_i \in \{0,1\} \quad \forall i \in V, \tag{4a}$$

在问题(4)，$e_{ij} \in E$ 和 $(i,k,l) \in CS$ 是用于描述积木块之间的关系. 令矩阵 $B=(B_{ij}) \in \{0,1\}^{n \times n}$ 是冲突图 G 的邻接矩阵. 如果 $e_{ij} \in E$，那么 $B_{ij}=1$，否则 $B_{ij}=0$；令张量 $C=(C_{ikl}) \in \{0, 1\}^{n \times n \times n}$ 来表示版图中的冲突结构. 如果 $(i,k,l) \in CS$，那么 $C_{ikl}=1$，否则 $C_{ikl}=0$. 进而，目标函数(4)可以通过邻接矩阵和张量写成更简便的形式如下.

$$\max \quad \sum_{i \in V} \{w_i x_i \prod_{j \in V} (1 - x_j)^{B_{ij}} \prod_{k,l \in V} (1 - x_k x_l)^{C_{ikl}}\}, \tag{5}$$

$$s.t. \quad x_i \in \{0,1\} \quad \forall i \in V, \tag{5a}$$

问题(5)虽与问题(2)等价，但它仍然是整数规划形式. 为了快速求解，问题(5)进一步被松弛成一个连续形式. 首先，引入一个辅助向量 $y=(y_i) \in R^n$ 通过 Sigmoid 函数来近似 0–1 变量 x_i 如下.

$$x_i \approx \sigma(y_i) = (1 + e^{-\gamma y_i})^{-1}, \tag{6}$$

其中 γ 是参数，用于刻画 Sigmoid 函数的平缓程度，如果 γ 越小，则 Sigmoid 函数越平缓. 本文取 $\gamma=8$. 见图 7.

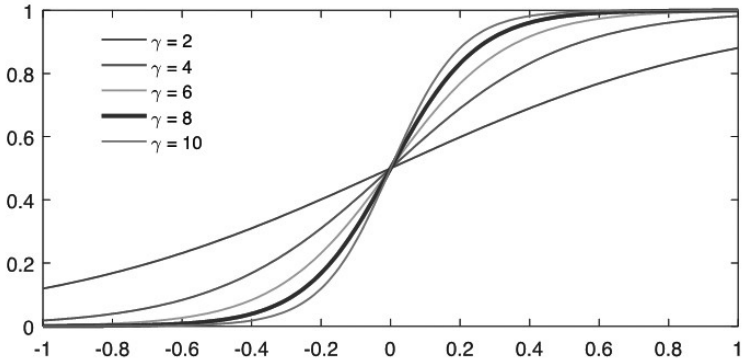

图 7 Sigmoid 函数在不同 γ 下的图像

Fig. 7 Sigmoid function on different γ

那么问题(5)可以转化成如下问题(7).

$$\max \ f(\boldsymbol{y}) = \sum_{i \in V} \{w_i \sigma(y_i) \prod_{j \in V} (1 - \sigma(y_j))^{B_{ij}} \prod_{k,l \in V} (1 - \sigma(y_k)\sigma(y_l))^{C_{ikl}}\}, \tag{7}$$

如果求解问题(7)得到一个解 $\boldsymbol{y}^*$，可以通过 Sigmoid 函数 $\sigma(\boldsymbol{y}^*)$ 近似到最近的整数得到问题(2)的一个解 x. 问题(7)是一个无约束非线性规划问题. 令

$$g_i(\boldsymbol{y}) = \sigma(y_i) \prod_{j \in V} (1 - \sigma(y_j))^{B_{ij}} \prod_{k,l \in V} (1 - \sigma(y_k)\sigma(y_l))^{C_{ikl}}, \tag{8}$$

且$g_i = g_i(\boldsymbol{y})$,然后问题(7)的目标可以简写为.

$$f(\boldsymbol{y}) = \sum_{i \in V} w_i g_i, \tag{9}$$

为了得到问题(7)的最大解$\boldsymbol{y}=(y_i) \in R^n$. 在每次迭代 t,解由$f(\boldsymbol{y})$的梯度方向更新.

$$\boldsymbol{y}^{(t+1)} = \boldsymbol{y}^{(t)} + \alpha \nabla f(\boldsymbol{y}^{(t)}), \tag{10}$$

其中,α 是步长,它由 Wolfe–Powell 非精确线搜索[20]得到,而$f(\boldsymbol{y}^{(t)})$的梯度由如下公式计算得到.

$$[\nabla f(\boldsymbol{y}^{(t)})]_i = \frac{\partial f(\boldsymbol{y}^{(t)})}{\partial y_i} = \gamma w_i g_i^{(t)} \{(1-\sigma(y_i^{(t)})) - \sum_j B_{ij} \sigma(y_j^{(t)}) - \sum_k \sum_l C_{ikl} \frac{\sigma(y_k^{(t)})(1-\sigma(y_k^{(t)}))\sigma(y_l^{(t)})}{1-\sigma(y_k^{(t)})\sigma(y_l^{(t)})}\}, \tag{11}$$

其中,$g_i^{(t)} = g_i(\boldsymbol{y}^{(t)})$,可以证明通过公式(10)迭代,$f(\boldsymbol{y}^{(t)})$可以收敛到局部最大值. 算法 1 中提出了针对求解无约束非线性规划问题(7)的线搜索迭代优化方法,其中目标值在每次迭代时都会增加,并收敛到最大解. 实验上, 算法 1 只需要迭代几次就可以达到终止条件. 此外, 从公式 (11) 知道算法 1 的算法复杂度为$O(\max\{|V|\cdot|E|, |V|\cdot||C||_0\})$,其中$||C||_0$是张量$C$中非零元素的数量.

算法 1: 线搜索算法

输入: 冲突图 $G(V, E, W)$的连通分支,收敛阈值 $\delta=10^{-4}$;

输出: 整数线性规划问题(2)的解 x^*;

1: 初始化 $t=0$;

2: 产生初始解 $x^{(0)}$;

3: 如果 $x_i^{(0)}=1$,则 $yi^{(0)}=1$;否则 $yi^{(0)}=-1$;

4: repeat

5: $\forall i \in V$,计算 $gi^{(t)}$;

6: 计算$\nabla f(\boldsymbol{y}^{(t)})$;

7: 对$\boldsymbol{y}^{(t)}$进行线搜索得到 α;

8: $\boldsymbol{y}^{(t+1)}=\boldsymbol{y}^{(t)}\boldsymbol{y}+\alpha\nabla f(\boldsymbol{y}^{(t)})$;

9: $t=t+1$;

10: until$||\nabla f(\boldsymbol{y}^{(t)})||<\delta$

11: 近似 $\sigma(y_i^*)$到最近的整数得到 x_i^*

引理 1 $\sum_i w_i \Delta g_i \geq 0$.

证明 根据公式(11), $\sum_i w_i \Delta g_i = \sum_i \gamma w_i g_i \{(1-\sigma(y_i)\Delta y_i) - \sum_j B_{ij}\sigma(y_j)\Delta y_j - \sum_k C_{ikl} \sum_l \frac{\sigma(y_k)(1-\sigma(y_k))\sigma(y_l)}{1-\sigma(y_k)\sigma(y_l)} \Delta y_k\}$, 令 $u_i = w_i g_i$, $\boldsymbol{u} = (u_i) \in R^l$, 和辅助矩阵$A = (A_{iJ}) \in R^{n \times n}$: 如果 $i=j$, $A_{ij} = 1-\sigma(y_i)$; 如果 $e_{ij} \in = E$, $A_{ij} = -\sigma(y_i)$; 如果$(i, k, l) \in = CS$. $A_{ik} = -\sum_l \frac{\sigma(y_k)(1-\sigma(y_k))\sigma(y_l)}{1-\sigma(y_k)\sigma(y_l)}$. 然后$\sum_i w_i \Delta g_i = \gamma \boldsymbol{u}^T A \Delta y$. 从公式(10)知道, $\Delta y = y(t+1) - y(t) = a\nabla f(y)$. 根据$\nabla f(y)$和$A_{ij}$的定义, 有 $\Delta y = \alpha\gamma A^T \boldsymbol{u}$. 因此, $\sum_i w_i \Delta g_i = \alpha\gamma^2 \boldsymbol{u}^T A A^T \boldsymbol{u} \geq 0$.

根据引理 1, 每次迭代有$\sum_i w_i \Delta g_i \geq 0$, 因此有下列定理 1.

定理 1 每次迭代，$f(\boldsymbol{y})$不下降.

定理 2 算法 1 收敛到局部最大值.

证明 因为$\forall i, x_i = \sigma(y_i): R \to [0,1]$，然后从公式(11)可以推理出$\forall i, g_i: R \to [0,1]$. 最后有$\sum_i w_i g_i \leq \boldsymbol{w}^T \boldsymbol{1}$成立. 另外根据定理 1，$f(\boldsymbol{y}) = \sum_i w_i g_i$总是递增. 因此当$\| \nabla f(\boldsymbol{y}) \| = 0$时算法 1 收敛到局部最大值.

3.3　加速技术

从图 5 可以看出，即使给定一个简单的版图结构如图 5a，最终构造的冲突图也是相当的复杂. 这是因为在之前整个冲突图的构造过程中考虑到了所有可能性. 从算法 1 的分析可以知道，其算法复杂度依赖于冲突图包含的点和边度数目. 如果冲突图的点的数目和边的数目多，那么算法运行会花较大的代价. 出于这个动机，本文提出基于点删除的算法来加速算法 1. 整数线性规划问题(2)的目标是为了最大化选择点的权重之和，所以对于每个连通分支，如果得到的解所包含点的权重之和不可能是最大的，那显然可以把这种解剪枝掉. 图 8 给出了一个例子，假设该冲突图包含 6 个点 a, b, c, d, e 和 f，权重分别为 1, 2, 2, 2, 2 和 3. 图 8a 中点 f 与其它剩余的 5 个点有冲突边或者重叠边相连，那么如果选择了点 f，则其它的点都不能够被选择；另外可以知道点 f 的权重不是最大的(b 与 d 由引导槽边相连，它们可以被同时选择，它们的权重和为 4). 因此最优解不可能包含点 f，进而它可以被删除，最终冲突图 8a 可以被简化成冲突图 8b 的形式. 同样图 8b 中点 e 也可以被删除，最终得到冲突图 8c. 与冲突图 8a 相比，冲突图 8c 求解相对快速很多. 下面给出发冲突图中可删除点的定义.

定义 6(可删除点)如果冲突图 $G(V, E)$的某个连通分支中的点 i 与分支内其它顶点都相连；另外通过多项式算法在剩余顶点中所找到的一组独立集 $j_1, j_2, \cdots, j_k$，如果$\sum_{j=j_1}^{j_k} w_j > w_i$，那么称点 i 为可删除点.

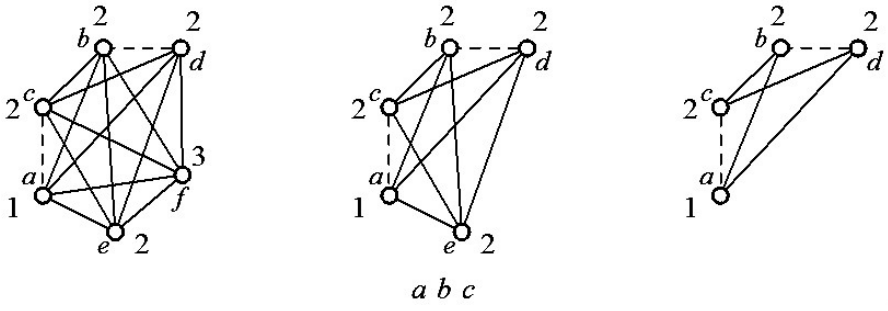

图 8 点删除

Fig. 8 Vertices removal

4　实验结果

本文提出的算法用 C ++实现，在具有 2.7GHz CPU，8GB 内存和 Unix 操作系统的个人计算机上运行. 所测试的实例集合是由 Fang 等[15]提供的 MCNC 基准测试实例和工业电路 Faraday 测试实例. 首先，所有测试实例的版图都转换为网格模型，其中网格尺寸为一个金属线间距. 在实验中，通孔与其冗余通孔之间的距离设定为一个金属线间距，相邻引导槽的最小光学分辨率极限间距 d_s 也设定为一个金属线间距. 参数 β 设置为 1.

为了评估本文所提方法的性能，将得到的结果与 TCAD'17[15]和 ASPDAC'17[19]中的 ILP 结果进行了比较. 实验比较见表 1，"TCAD'17"和"ASPDAC'17"分别是文献[15]和[19]中的结果，"Ours"中的结果是通过

求解 3.2 节中的算法获得的. 此外,表 1 中,列“#V”列出了通孔的数量,列“CPU(s)”是第 2 个的运行时间. “MR(%)”和“IR(%)”分别是制造速率和冗余通孔插入率,其中 #MV 是可制造的通孔的数量(不包括冗余通孔),#RV 是插入的有效冗余通孔的数量(每个通孔至多算一个有效冗余通孔).

$$\mathrm{MR}=\frac{\#\mathrm{MV}}{\#\mathrm{V}}\times 100\%\ ,\quad \mathrm{IR}=\frac{\#\mathrm{RV}}{\#\mathrm{V}}\times 100\%\ , \tag{12}$$

Ours 和 TCAD'17 中的结果之间的差异在于本文的工作考虑插入多个冗余通孔, 但 TCAD'17 没有. 如行表 1 中“比率”行可以看出,本文所提出的算法分别提高 MR 和 IR 均达到 7%. 当然,考虑多冗余通孔插入将极大地增加解空间的大小. 尽管如此,本文的算法的运行时间比 TCAD'17 的算法少了 5. 4 倍.

ASPDAC'17 和 Ours 的方法都考虑了多冗余通孔插入来提高插入率和制造率. 由于 ASPDAC'17 的方法的直接求解整数线性规划问题,所以 ASPDAC'17 的结果应该是最优的. 从表 1 中的比较可以发现本文的算法实现了与 ASPDAC'17 中几乎相同的结果, 这也表明本文所提出的局部最优算法可以实现接近最优的结果. 必须注意的是,ASPDAC'17 中整数线性规划求解时间比本文的方法慢了 17. 78 倍. 运行时的改进归功于本文的紧凑的解表达方式和线搜索算法, 以及本文的基于点删除的加速技术. 从图 9 可以直观的看出本文算法在运行时间上的优势.

表 1 实验结果比较

Tab. 1 Experimental comparison

测试实例	#V	TCAD'17			ASPDAC'17			Ours		
		MR/%	IR/%	CPU/s	MR/%	IR/%	CPU/s	MR/%	IR/%	CPU/s
struct	12 551	99. 86	99. 42	30. 00	99. 96	99. 79	40. 47	99. 95	99. 69	8. 62
primary1	8 764	99. 80	99. 21	28. 00	99. 92	99. 33	29. 14	99. 80	99. 45	10. 20
s5378	8 649	81. 10	61. 78	11. 00	96. 41	75. 27	34. 41	96. 42	75. 09	0. 44
s13207	18 780	84. 35	66. 93	23. 00	97. 13	79. 28	99. 96	97. 19	79. 01	1. 48
s15850	22 694	82. 70	64. 41	28. 00	96. 63	77. 35	121. 94	96. 61	77. 58	2. 18
s38417	54 225	84. 00	65. 71	65. 00	96. 83	78. 13	320. 35	96. 92	78. 59	10. 45
s38584	74 155	81. 53	63. 01	88. 00	96. 37	76. 83	416. 11	96. 18	77. 12	19. 50
dsp1	30 317	99. 05	97. 57	53. 00	99. 74	98. 45	176. 42	99. 45	98. 25	8. 42
dsp2	31 301	98. 50	96. 68	52. 00	99. 76	98. 74	179. 37	99. 35	98. 16	6. 84
risc1	43 858	98. 77	96. 93	75. 00	99. 70	98. 04	216. 61	99. 34	97. 90	15. 14
risc2	44 385	98. 79	96. 91	77. 00	99. 70	97. 98	229. 22	99. 32	97. 89	14. 95
平均	31 789	91. 68	82. 60	48. 18	98. 38	89. 02	158. 76	98. 23	88. 98	8. 93
比率		0. 93	0. 93	5. 40	1. 00	1. 00	17. 78	1. 00	1. 00	1. 00

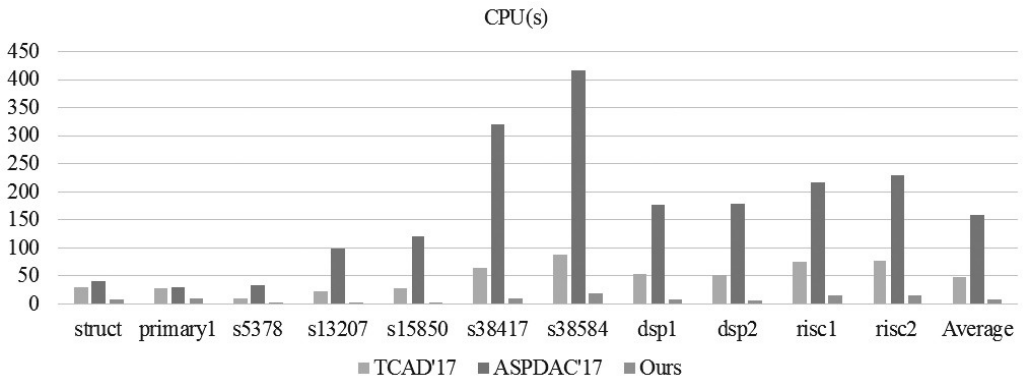

图 9 算法运行时间比较

Fig. 9 Comparison on runtime

5　结束语

本文考虑了多冗余通孔插入的 DSA 引导槽分配问题. 本文引入积木块，并构造相应的冲突图. 由于积木块，可以有效地减少冲突图中的顶点数. 基于冲突图，本文将原问题建模为整数线性规划问题，并将其松弛为无约束非线性规划. 本文开发了一种线搜索优化算法来获得局部最优解，并设计点删除方法来加快求解速度. 实验结果证明了本文的解表达方式和提出的算法的有效性.

参考文献:

[1] Yi H, Bao X-Y, Zhang J, et al. Contact-hole patterning for random logic circuits using block copolymer directed self-assembly[C]//Proceedings of SPIE, 2012: 8323.

[2] Ou J, Yu B, Gao J-R, et al. Directed self-assembly cut mask assignment for unidirectional design[J]. Journal of Micro/Nanolithography. MEMS MOEMS, 2015, 14(3): 1-8.

[3] Wong H S P, Bencher C, Yi H, et al. Block copolymer directed self-assembly enables sub-lithographic patterning for device fabrication[C]//Proceedings of SPIE, 2012: 8323.

[4] Bates C M, Seshimo T, Maher M J, et al. Polarityswitching top coats enable orientation of sub-10-nm blockcopolymerdomains[J]. Science, 2012, 338(6108): 775-779.

[5] Shim S, Chung W, Shin Y, Redundant via insertion for multiple- patterning directed-self-assembly lithography[C]//Proceedings of the 53rd Annual Design AutomationConference, 2016: 410-417.

[6] Zorian Y, Gizopoulos D, Vandenberg C, et al. Guest editors introduction: design for yield and reliability[J]. IEEE Transactions on Computer-AidedDesign of Integrated Circuits and Systems, 2004, 21(12): 2197-2208.

[7] Xu G, Huang L D, Pan D Z, et al. Redundant-via enhanced maze routing for yield improvement[C]//Proceedings of the 10th Asia and South PacificDesign AutomationConference, 2005: 1148-1151.

[8] Ou J, Yu B, Xu X, et al. DSAR: DSA aware routing with simultaneous DSA guiding pattern and double patterning assignment[C]//Proceedings of theInternational Symposium on Physical Design, 2017: 91-98.

[9] Chen H Y, Chiang M F, Chang Y W, et al. Full-chip routing considering double-via insertion[J]. IEEE Transactions on Computer-AidedDesign of Integrated Circuits and Systems, 2008, 27(5): 844-857.

[10] Lee K Y, Koh C K, Wang T C, et al. Fast and optimal redundant via insertion[J]. IEEE Transactions on Computer-AidedDesign of Integrated Circuits and Systems, 2008, 27(12): 2197-2208.

[11] Lin S T, Lee K Y, Wang T C, et al. Simultaneous redundant via insertion and line end extension for yield optimization[C]//Proceedings of the 16th Asia and South PacificDesign AutomationConference, 2011: 633-638.

[12] Badr Y, Torres A, Gupta P. Mask assignment and synthesis of DSA-MP hybrid lithography for sub-7nm contacts/vias[C]//Proceedings of the 52nd Annual Design AutomationConference, 2015: 1-6.

[13] Xiao Z, Du Y, Tian H, et al. Directed self-assembly (DSA) template pattern verification[C]//Proceedings of the 52nd Annual Design AutomationConference, 2014: 1-6.

[14] Kuang J, Ye J J, Young F Y. Simultaneous template optimization and mask assignment for DSA with multiple patterning[C]//Proceedings of the 21st Asia and South PacificDesign AutomationConference, 2016: 75-82.

[15] Fang S Y, Hong Y X, Lu Y Z. Simultaneous guiding template optimization and redundant via insertion for directed self-assembly[J]. IEEE Transactions on Computer-AidedDesign of Integrated Circuits and Systems, 2017, 36(1): 156-169.

[16] Ou J, Yu B, Pan D Z. Concurrent guiding template assignment and redundant via insertion for DSA-MP hybrid lithography[C]//Proceedings of theInternational Symposium on Physical Design, 2016: 39-46.

[17] Li X, Yu B, Ou J, et al. Graph based redundant via insertion and guiding template assignment for DSA-MP[J]. IEEE Transactions on Very Large-Scale Integration Systems, 2018, 26(11): 2504-2517.

[18] Fang S Y, Hong Y X. Design optimization considering guiding template feasibility and redundant via insertion for directed self-assembly[C]//Proceedings of IEEE Asia Pacific Conference on Circuits and Systems, 2016: 1-4.

[19] Hung C Y, Chou P Y, Mak W K. Optimizing DSA-MP decomposition and redundant via insertion with dummy vias[C]//Proceedings of the 22nd Asia and South PacificDesign AutomationConference, 2017: 1-6.

[20] Noceda J, Wright S J. Numerical Optimization[M]. 2nd Edition, Springer, 2006.

[责任编辑：钟国翔]