利用命题逻辑最大可满足性的冗余通孔最优插入方法

杨成[1,2)], 杨骏[1)], 张亚东[3)]

[1)] (中国科学院微电子研究所汽车电子中心 北京 100029)
[2)] (中国科学院大学 北京 100049)
[3)] (北京华大九天科技股份有限公司 北京 100102)
(yangcheng@ime.ac.cn)

摘 要: 在纳米尺度的集成电路设计中, 冗余通孔插入是减轻通孔失效造成良率降低问题的常用技术. 文中将最优冗余通孔插入问题规约到命题逻辑最大逻辑可满足性(maximum satisfiability, Max SAT)问题, 并利用完备求解器求取最优解. Max SAT 问题是一个 NP 困难问题, 采用 2 种方法来降低求解难度; 一是预选取方法, 将提前确定的不与其他通孔产生冲突的冗余通孔作为部分解来降低问题的规模; 二是分治法, 根据连通分量将原问题划分成多个子问题分别求解, 降低求解的复杂度. 同时, 从理论上证明这 2 种方法能够保证解的最优性. 在 2019 年国际物理设计研讨会(ISPD)举办的详细布线比赛基准测试集上进行实验的结果表明, 所提出的插入方法带来的时间开销不到详细布线时间的 5%, 算法的最优性保证了最大化解决插入冲突后的插入率, 在所有可插入通孔中, 冗余通孔的插入率为 67%~87%.

关键词: 冗余通孔插入; 命题逻辑最大可满足性问题; 版图后优化; 可制造性设计
中图法分类号: TP391.41 **DOI:** 10.3724/SP.J.1089.2023.19482

Method for Optimal Redundant via Insertion with Maximum Satisfiability

Yang Cheng[1,2)], Yang Jun[1)], and Zhang Yadong[3)]

[1)] (*Automotive Electronics Center, Institute of Microelectronics of the Chinese Academy of Sciences, Beijing* 100029)
[2)] (*University of Chinese Academy of Sciences, Beijing* 100049)
[3)] (*Empyrean Technology Co., Ltd, Beijing* 100102)

Abstract: In nanoscale integrated circuit design, redundant via insertion is a common technique to alleviate the yield loss caused by via failure. In this paper, the optimal redundant via insertion problem is converted to a maximum satisfiability (Max SAT) problem, and the optimal solution is obtained by using a complete solver. Since the Max SAT problem is NP-hard, two techniques are used in this paper to reduce the difficulty of finding solutions. One is the pre-selection, which reduces the size of the problem by identifying the redundant vias that do not conflict with other vias as partial solutions. The other is problem partitioning, which divides the original problem into several sub-problems with respect to the connected components to reduce the solution complexity. The paper also proves that these two methods can guarantee the optimality of the solution. Finally, the algorithm experiments on the benchmarks of the detailed routing competition of the International Symposium on Physical Design (ISPD) 2019. The optimality of the algorithm ensures that the insertion rate is maximized while resolving insertion conflicts, and the insertion rate of redundant vias is 67%—87% of all insertable vias.

收稿日期: 2021-11-22; 修回日期: 2022-03-10. **杨成**(1996—), 男, 硕士研究生, 主要研究方向为集成电路物理设计自动化; **杨骏**(1966—), 男, 博士研究生, 研究员, 硕士生导师, 主要研究方向为信号处理、人工智能; **张亚东**(1987—), 男, 硕士研究生, 主要研究方向为集成电路物理设计自动化.

Key words: redundant via insertion; maximum satisfiability problem; post-layout optimization; design for manufacturing

在集成电路制造过程中，通孔失效是一个影响良率的重要问题，冗余通孔插入作为可制造性设计技术之一，能够有效地减轻通孔失效问题对良率的影响. 为了保证良率，芯片的物理设计中应当尽可能地使用双通孔[1]. 已有许多文献讨论冗余通孔插入问题，并提出多种模型和算法. 文献[2]将插入问题构造成最小独立集问题，并提出一种启发式方法进行规模缩减并求解；文献[3]进一步构造整数线性规划问题求解该问题；文献[4]提出二分图匹配法，并证明了该方法在3层布线层下能够保证解的最优性. 此外，文献[5]是首个在详细布线阶段考虑通孔插入的工作，而基于多商品流的详细布线器将冗余通孔插入问题转化为线性约束[6]，在布线阶段加入了对插入操作的考虑. 近年来，随着定向自组装(directed self-assembly, DSA)光刻技术的出现，研究者们针对此类冗余通孔插入问题提出一些新的算法. 文献[7]提出整数线性规划的方法解决多重图形下DSA的通孔插入问题；文献[8]提出一种通过微调金属线来进行冗余通孔插入的方法. 由于缺乏相关工艺以及版图数据，因此 DSA 不在本文讨论范围之内. 国际物理设计研讨会(international symposium on physical design, ISPD)举办的详细布线比赛提供了真实的芯片设计作为测试集[9]，并且至今仍可开放获取，该测试集的线网规模远大于前述工作. 大部分测试用例中提供了双通孔的工艺参数，但比赛中鲜有队伍较好地支持冗余通孔插入. 因此，适应于现代集成电路设计中大规模版图中的冗余通孔插入问题值得被进一步探讨.

本文在文献[10]提出的详细布线算法的基础上进行布线后的冗余通孔插入，将问题构造为最大可满足性(maximum satisfiability, Max SAT)问题，并利用预选取缩减问题规模，用连通分量划分对问题进行分治，在证明采用加速策略保证问题的最优性的基础上降低了问题求解的复杂度.

为验证算法的有效性和运行效率，本文在2019年的ISPD详细布线比赛的测试集[9]上进行实验，结果表明，本文提出的冗余通孔插入方法的运行时间不超过总运行时间的5%，运行时间开销较小.

1　问题描述

首先介绍冗余通孔插入的问题形式，以及 Max SAT 问题的定义；然后将不同通孔插入时产生的设计规则违例(design rule violation, DRV)视为冲突关系，构建出表示冲突关系的无向图；最后根据该图构造出 Max SAT 问题.

1.1　冗余通孔插入问题的一般形式

冗余通孔插入是物理设计的环节之一，同样需要满足设计规则约束. 本文将冗余通孔插入导致的 DRV 分为 2 类：第 1 类是与任何当前版图图形、布线结果以及默认通孔产生的 DRV. 由于本文方法在布线后进行通孔插入，若通孔插入引入该类违例，违例将无法被修复，因此不允许此情况发生；第2类是与其他冗余通孔的插入发生设计规则违例. 若发生这种情况，则放弃该插入操作即可避免违例的产生.

在文献[10]提出的详细布线算法中，详细布线结果被记录在多叉树中，称为布线结果的拓扑结构，拓扑结构中的所有代表通孔的结点 n 构成结点集 N. 进行冗余通孔插入时可以选取 R 种冗余通孔类型，构成冗余通孔类型集合 $T=\{t_i \mid i=1,2,\cdots,R\}$.

由于拓扑结构上的任意代表通孔的结点之间均不应产生设计规则违例，定义第 1 条约束(即 DRV 约束)为 $\sum\limits_{n\in N}\sum\limits_{m\in N-n}\mathrm{DRV}(n,m)=0$. 其中，$\mathrm{DRV}(n,m)$ 表示结点 n 和结点 m 之间存在的设计规则违例数量. 此外，对任意结点 $n\in N$，最终只可应用唯一的插入方式或不应用任何插入方式，定义 $x_{n,i}\in\{0,1\}$ 表示在结点 n 处，是否插入冗余通孔 t_i，定义第 2 条约束(即插入规则约束)为

$$\sum_{i=1}^{R}x_{n,i}\leqslant 1,\quad \forall n\in N.$$

优化目标是尽可能多地进行冗余通孔插入，综合上述 2 条约束，得到该约束优化问题的一般形式为

$$\begin{cases}\max & \sum\limits_{n\in N}\sum\limits_{i=1}^{R}x_{n,i}\\ \text{s.t.} & \sum\limits_{n\in N}\sum\limits_{m\in N-n}\mathrm{DRV}(n,m)=0\\ & \sum\limits_{i=1}^{R}x_{n,i}\leqslant 1,\quad \forall n\in N\end{cases}\tag{1}$$

1.2 Max SAT 问题

Max SAT 问题是命题逻辑公式可满足性判定(propositional satisfiability problem, SAT)的推广问题. SAT 问题回答是否存在使合取范式(conjunction normal formulation, CNF)为真的变量取值, CNF 是可满足性问题的标准形式, 由一系列析取子句的合取构成, 则 CNF 为真, 等价于所有析取子句均为真. 因此, SAT 问题等价于判断是否存在使所有子句均为真的变量取值. 然而, SAT 仅能提供能否使所有子句成立的信息, 而其推广问题 Max SAT 定义为尽可能找到使更多的子句为真的变量取值, 因此相较于 SAT, Max SAT 能够描述更多现实世界的问题. 进一步地, 许多时候优化问题是带约束的, 定义硬子句为必须满足的子句, 而软子句为原 Max SAT 中所希望的尽可能满足的子句, 即得到 Max SAT 的增广形式部分最大可满足性(partial Max SAT)问题, 此问题定义为在满足所有硬子句为真的前提下, 最大化软子句为真的数目. 显然, partial Max SAT 更加自然地描述了组合优化问题的一般形式, 诸多组合优化问题可以被规约到该问题[11]. 本文将冗余通孔插入问题抽象为 partial Max SAT 问题, 并使用开源的 Max SAT 完备求解器 EvalMaxSAT 进行求解[12].

1.3 构造 Max SAT 问题

首先将原问题抽象成表示冲突关系的无向图, 称为冲突图. 冲突图 $G(V,E)$ 的顶点集 V 中每个顶点 $v=\{n,t\}$ 唯一地表示了一个不与任何当前版图图形、布线结果以及默认通孔产生 DRV 的结点-类型对. 其中, 结点是布线结果拓扑结构中的结点 n, 类型是冗余通孔的类型 t. 如图 1a 所示, 结点 b 是通孔结点, 而在该处选取 t_1 类型的通孔不会导致违例的产生, 因此对于该结点-类型对构成的顶点 $u=\{b,t_1\}$, 有 $u\in V$, 即顶点 u 所代表的插入操作存在于冲突图顶点集中; 结点 a 也是一个通孔结点, 但是在该处选取 t_1 类型的通孔会导致其与同层障碍物产生设计规则违例, 因此对于顶点 $v=\{a,t_1\}$, 有 $v\notin V$, 即顶点 v 所代表的插入操作不是冲突图的顶点.

冲突图的无向边集 E 由 2 个子集构成: 第 1 个子集是违例边集 E_{vio}, 其中的每条边 $e_{\text{vio}}=\{u,v\}$ 表示在结点 *v.n* 处选取冗余通孔类型 *v.t* 会与结点 *u.n* 处选取冗余通孔类型 *u.t* 产生 DRV. 如图 1b 所示, 记顶点 $v=\{a,t_1\}$ 表示在 a 处插入 t_1 类型的冗余通孔, 顶点 $u=\{b,t_2\}$ 表示在 b 处插入 t_2 类型的冗余通孔, 由于这 2 个插入操作会导致 DRV 的产生, 故它们不可同时进行, 因此在顶点 u 和顶点 v 间添加违例边 (u,v); 第 2 个子集是自冲突边集 E_{intr}, 对任意自冲突边 $e_{\text{intr}}=(u,v)$, 有 $e_{\text{intr}}=(u,v)$, 即自冲突边的 2 个顶点处于相同的通孔结点. 如图 1c 所示, 在同一个结点 a 处有 2 种可选的冗余通孔类型 t_1 和 t_2, 分别对应于顶点 $v=\{a,t_1\}$ 和顶点 $u=\{a,t_2\}$, 对结点 a 处的通孔, 仅可选取唯一的冗余通孔插入类型. 因此, 通过添加边 (u,v) 来表示对任一结点, 只能有 1 种冗余通孔类型被选中的约束.

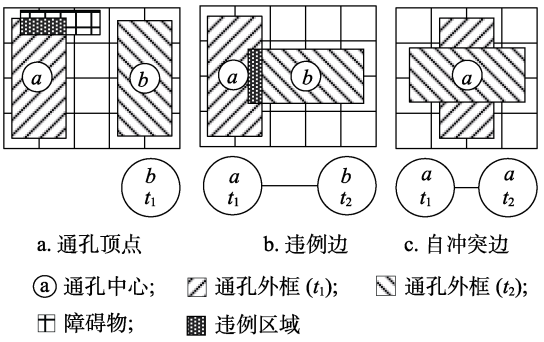


图 1 冗余通孔插入尝试与冲突图的对应关系

按照以上规则构建冲突图 $G(V,E)$, 下文介绍如何将冲突图表示的制约关系转化为命题逻辑问题. 以逻辑“真”值表示可以应用冲突图中某个顶点所表示的插入行为, “假”则表示不采取该插入操作. 在冲突图中, 每条边的 2 个顶点不可被同时选取, 于是对每条冲突边 (u,v) 定义析取子句 $\neg u\vee\neg v$. 其中, u 和 v 分别表示冲突边的 2 个顶点. 该子句施加了 2 个变量不可同时取值为真的约束, 即 2 种插入不可被同时采用. 将该子句定义为硬子句, 即对于任何可行解该子句必须满足, 从而保证了造成违例的一对插入操作不会被同时采用, 同时每个布线网格结点处只会应用一种可行的插入操作. 通过定义该析取子句为硬约束, 于是 Max SAT 问题的任何可行解必须使得该子句为真, 保证了最终选取的候选冗余通孔之间不会产生 DRV.

由于优化目标是尽可能多地插入冗余通孔, 通过对冲突图顶点集 V 中的每个顶点 v 定义软子句 v, 使尽可能更多的顶点的取值为真. 于是, 约束优化问题式(1)被构造成为 Max SAT 问题.

2 本文方法

本文提出的冗余通孔插入方法步骤如下:

输入. 布线拓扑中的通孔结点集 N.

输出. 冲突图 G.

Step1. 初始化冲突图 $G(V,E)=\varnothing$.

Step2. 对于所有的通孔结点 $n\in N$ 进行冗余通孔插入尝试:

Step2.1. 初始化结点 n 处的插入操作集 $C=\varnothing$;

Step2.2. 遍历所有可用的通孔类型，检查当前冗余通孔类型 i 是否会导致违例. 若不会，则将结点-类型对加入到插入操作集 $C=C\cup\{n,t_i\}$;

Step2.3. 遍历插入操作集 C 中顶点，检查 $v\in C$ 是否会与当前冲突图中的顶点 $u\in V$ 造成设计规则违例. 若会，则添加冲突边 $\{u,v\}$;

Step2.4. 若 $|C|>1$，则为所有 $u,v\in C$ 且 $u\neq v$ 添加冲突边 (u,v)，并对原顶点集与插入操作集做并集 $V=V\cup C$.

Step3. 遍历顶点集 C，进行预选取:

Step3.1. 若顶点 v 的任一邻居 $u\in N_G(v)$ 都与之表示同一布线拓扑的结点，即 $u.n=v.n$，转 Step3.2; 否则，跳过当前顶点;

Step3.2. 遍历顶点 v 的所有邻居顶点，收集邻居节点 u 的所有邻居 $w\in N_G(u)$ 为次邻居集合 N'，并删除所有邻居结点;

Step3.3. 对于所有 $w\in N'$，令 $v=w$，转 Step3.1.

Step4. 对图 G 的所有连通分量分别构造 Max SAT 问题，并求解.

下面先介绍冲突图的构建方法，并分析其复杂度，然后证明 Step3 的预选取和 Step4 将原问题划分为子问题后仍能保证解是最优的.

2.1　构建冲突图

本文应用文献[10]提出的基于查找表的悲观查询策略，预先构建多张查找表(lookup table, LUT)，提供时间复杂度为常数阶的通孔 DRV 的检查方法. 与文献[3]中基于树结构进行区域查询的检查方法相比，本文方法中，Step2.2 和 Step2.3 中检查 DRV 的时间复杂度从对数阶降低至常数阶.

所有在布线拓扑中使用的通孔类型都被记录在通孔处的结点上，通过遍历布线拓扑即可访问到通孔结点集合 N 中的所有元素. 对于每个通孔结点在所有通孔类型上进行插入尝试，每次插入都查询 LUT 来确认是否有违例产生，将不与原始结果产生冲突的插入操作添加顶点到冲突图中，故顶点插入的时间复杂度为 $O(|N|\cdot|T|)$. 其中，T 是可选冗余通孔类型的集合. 对于产生 DRV 的 2 个插入操作，在对应顶点之间加入冲突边，当冲突图是完全图时此操作复杂度较高. 但实际问题中，冲突常是局部的，因此构建冲突边的复杂度为 $O(|V|\cdot\alpha|V|)$，其中，$\alpha(\cdot)$ 是一个增长非常缓慢的函数[13]，而图顶点数 $|V|$ 与 $|N|\cdot|T|$ 是同阶的，且 $|T|$ 是一个常数，因此构建冲突图的复杂度为 $O(|N|\cdot(1+\alpha(|N|)))$. 至此，冲突图构建完毕.

由于 Max SAT 是一个 NP 困难问题，而原问题的规模较大，因此求解速度非常慢; 而在此过程中有许多通孔插入操作并不与其他结点通孔插入产生冲突，预先确定这些位置的插入不仅能剔除冗余的顶点以缩小问题规模，也会移除不必要的冲突关系. 由于冲突是局部的，可以将问题划分为子问题并分别求解; 而预先确定的方法会潜在地增加连通分量的数目，可进一步缩减每个子问题的规模. 本文沿用文献[3]中对此类方法的命名，并在附录 A 中证明了此方法在 Max SAT 问题下解的最优性.

2.2　预选取

有许多通孔插入操作存在一个不与任何其他结点产生冲突的冗余通孔类型，即对于任一结点，与其他结点的冲突并不总是存在的，这样的结点-类型对可被预先确定为解的一部分; 并且这种选取与原 Max SAT 问题有相同的最优值，即成立的软子句个数相同，证明见附录 A.

在冲突图构建完成后，Step3 使用广度优先遍历的方式从任一顶点开始遍历其邻居，若当前顶点与其所有邻居表示的结点相同，则将其所有邻居顶点从冲突图中删除，并将其邻居的所有邻居(除自身外)添加到待访问队列中; 继续进行广度优先遍历直至队列为空，并遍历其他邻居. 为每个结点添加一个布尔类型的变量来记录其是否曾被访问过，若被访问过，则不再遍历其邻居. 这样做能够保证冲突图上的每条边至多被访问 2 次: 一次是对顶点的初次访问; 另一次是由于该顶点的邻居被删除而对其进行访问，而每个顶点本身会因广度优先遍历而被访问，因此预选取的时间复杂度为 $O(2|E|+|V|)$，又因为冗余通孔构建的冲突图是稀疏的，所以复杂度可记为 $O(|V|)$，根据第 2.1 节，也可记为 $O(|N|)$.

2.3　根据连通分量划分原问题

Max SAT 是一个 NP 困难问题，从原始图构造出的 Max SAT 问题规模并不小; 根据实际问题中通孔插入操作的冲突具有局部性的特点，可以考虑对问题分而治之. 可以证明，根据连通分量构造出的多个 Max SAT 问题的最优解与原图构造出的 Max SAT 问题最优解相同，最优性的证明见附录

A. 因此, Step4 通过广度或深度优先探索得到无向图的所有连通分量, 并分别构造 Max SAT 问题. 因此, 本步骤的时间复杂度为 $O(|E|)$.

实际运行中, 即使对于测试用例中最小的版图, 原问题的求解时间也非常长, 甚至超过详细布线阶段, 而经过预选取和问题划分对问题规模进行缩减后, 运行时间得到了极大的改善.

3 实验结果与分析

本文在 2.60 GHz Intel(R) Core(TM) i5-11400 CPU, 32 GB 内存, Ubuntu 20.04 操作系统的个人电脑上进行实验; 数据集采用 ISPD19 详细布线比赛的测试用例, 测试集的相关参数如表 1 所示. 本文的冗余通孔插入方法在文献[10]提出的详细布线算法基础上实现, 其中, 详细布线算法以 4 线程运行, 本文的冗余通孔插入方法没有多线程实现, 故在完成详细布线后单线程运行.

表 1 ISPD19 基准测试集参数

测试用例	布线层	线网数	裸片尺寸/mm^2
test2	9	72 410	0.873×0.589
test3	9	8 953	0.195×0.195
test4	5	151 612	1.604×1.554
test5	5	29 416	0.906×0.906
test6	9	179 863	1.358×1.325

图 2 所示为从读入 LEF/DEF 到冗余通孔插入完成的运行时间对比. 其中, 黄色部分表示详细布线的运行时间, 绿色部分表示通孔插入的运行时间, 折线图中的数据点表示通孔插入算法运行时间占总运行时间的比例.

从图 2 可以看出, 求解冗余通孔插入的运行时间不超过总运行时间的 5%, 远小于详细布线所需时间. 因此, 本文提出的冗余通孔插入方法带来的时间开销是足够小的, 原因有 2 个: (1) 由于冗余通孔插入的冲突的局部性很强, 虽然在实现中每个通孔有 6 种冗余通孔插入方式, 即最多会产生 6 个顶点, 但观察发现连通分量的顶点数一般在 100 以内, 因此根据连通分量构造的子问题变量数量足够小, 求解子问题较为容易, 求解速度也可得到保证; (2) 使用基于查找表的违例查询方式, 对于每个插入违例判断的时间复杂度是常数阶的, 避免了基于树结构的区域查询方法在检查是否与版图上其他图形产生违例时所需的对数时间开销, 在图形较多时, 基于查找表的方法仍然能够保证较高的效率.

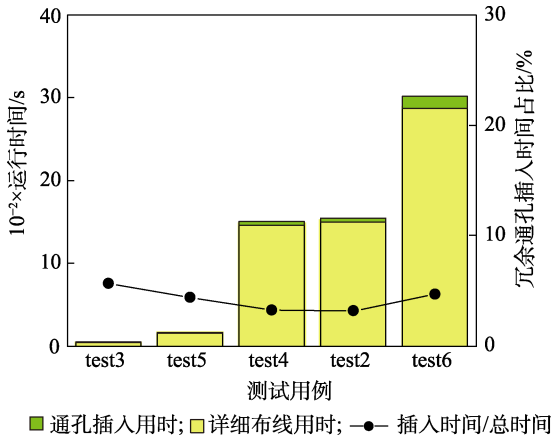


图 2 冗余通孔插入运行时间占比

本文还对通孔插入的效果进行实验. 值得注意的是, 由于文献[10]的详细布线算法在布线时并不考虑冗余通孔, 因此不会为冗余通孔插入预留空间, 也不会考虑插入冲突. 本文算法只能在保证所有插入的冗余通孔均不会导致与版图图形、布线结果和默认通孔等产生 DRV, 同时在避免插入间冲突的条件下最大化能够插入的冗余通孔数量, 但不能对原本就无法插入冗余通孔的位置进行处理. 因此, 实验中还利用文献[10]的算法在拆线重布迭代过程中仅拆解有 DRV 的线网重新布线的特点, 简单尝试了在布线迭代时对没有 DRV 的线网进行冗余通孔插入的方法(本文将此方法称为“布线迭代时考虑冗余通孔插入”法), 以保证在布线迭代过程中 DRV 线网在重新布线时能够绕开之前已经插入的合理线网冗余通孔. 表 2 中, 带有 w/o 标志的行表示仅在布线完成后进行冗余通孔插入的结果, 不带 w/o 标志代表采用布线迭代时考虑冗余通孔插入方法的结果.

表 2 冗余通孔插入结果统计

测试用例	通孔总数	可插入数	实际插入数	插入率/%	
				有效	实际
test2 (w/o)	788 220	584 047	392 017	67	49
test2	889 540	743 960	540 203	73	61
test3 (w/o)	64 858	51 409	39 583	77	61
test3	70 065	62 096	50 957	82	73
test4 (w/o)	1 031 463	889 409	705 505	79	68
test4	1 192 001	1 100 691	890 301	81	75
test5 (w/o)	153 545	146 687	128 354	87	84
test5	159 628	156 943	144 982	92	91
test6 (w/o)	1 986 781	1 586 646	1 133 843	71	57
test6					

从表 2 中带有 w/o 的结果可以看出，虽然在详细布线阶段没有考虑为冗余通孔插入预留空间，但是本文插入算法得到的解的最优性保证了较高的有效插入率. 然而，对于布线后的冗余通孔插入方法，尽管本文插入方法能够最大化冗余通孔的插入数目，但实际插入率在可进行冗余通孔插入的通孔数目和冗余通孔之间的冲突 2 方面因素的影响下并不是很高.

从表 2 中不带 w/o 的结果可以看出，通过布线迭代时考虑冗余通孔插入的方法，有效插入率和实际插入率都得到了不同程度的提升. 但预留空间和布线迭代中对冗余通孔插入影响的因素很多，需要在今后的工作中全面考虑各种影响因素，以更好地解决此问题.

表 3 所示为 ISPD19 详细布线比赛结果中公开的参赛队伍使用冗余通孔数目的情况[9]，可以看出，本文方法有效地提升了插入数目.

表 3　本文与不同方法[9]在 test6 中冗余通孔数目对比

方法	冗余通孔数
方法 1	0
方法 2	269
方法 3	0
方法 4	2314
本文	1133843

4　结　语

本文提出采用 Max SAT 模型解决布线后的最优冗余通孔插入问题的方法，利用冲突的局部性对原问题进行划分以降低 Max SAT 的规模，在保证了解的最优性的前提下，能够以不超过总运行时间 5%的时间开销完成冗余通孔插入.

实验结果表明，布线后进行的冗余通孔插入会受制于布线结果，如果在布线阶段对此加以考虑，则能够提升冗余通孔的插入率. 但是，布线迭代时考虑冗余通孔插入的方法与现有的在布线的同时通过添加代价来考虑冗余通孔插入的方法有所不同，未来将对 2 种方法分别进行更细致地分析，并综合考查插入均衡性以及插入导致的线长变化、通孔数量等指标，寻找综合表现更优的布线时考虑冗余通孔插入的策略.

由于 Max SAT 各个子问题之间是独立的，也可以考虑并行求解子问题，进一步降低通孔插入阶段的运行时间.

参考文献(References):

[1] Zorian Y, Gizopoulos D, Vandenberg C, *et al*. Guest editors' introduction: design for yield and reliability[J]. IEEE Design Test of Computers, 2004, 21(3): 177-182

[2] Lee K Y, Wang T C. Post-routing redundant via insertion for yield/reliability improvement[C] //Proceedings of the Asia and South Pacific Conference on Design Automation Conference. Los Alamitos: IEEE Computer Society Press, 2006: 303-308

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

[4] Chen H Y, Chiang M F, Chang Y W, *et al*. Novel full-chip gridless routing considering double-via insertion[C] //Proceedings of the 43rd Annual Design Automation Conference. New York: Association for Computing Machinery, 2008: 755-760

[5] Xu G, Huang L D, Pan D Z, *et al*. Redundant-via enhanced maze routing for yield improvement[C] //Proceedings of the 2005 Asia and South Pacific Design Automation Conference. New York: Association for Computing Machinery. 2005: 1148-1151

[6] Jia X T, Cai Y C, Zhou Q, *et al*. MCFRoute 2.0: a redundant via insertion enhanced concurrent detailed router[C] //Proceedings of the 26th Edition on Great Lakes Symposium on VLSI. New York: Association for Computing Machinery, 2016: 87-92

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

[8] Lin K L, Fang S Y, Hong Y X. Design optimization considering guiding template feasibility and redundant via insertion for directed self-assembly[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2017, 63(12): 3172-3182

[9] Liu W H, Mantik S, Chow W K, *et al*. ISPD 2019 initial detailed routing contest and benchmark with advanced routing rules[C] //Proceedings of the International Symposium on Physical Design. New York: Association for Computing Machinery, 2019: 147-151

[10] Chen G J, Pui C W, Li H C, *et al*. Dr.CU: detailed routing by sparse grid graph and minimum-area-captured path search[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2020, 39(9): 1902-1915

[11] Li C M, Manyà F. MaxSAT, hard and soft constraints[M] //Handbook of Satisfiability, vol 336. 2rd ed. Amsterdam: IOS Press, 2021

[12] Avellaneda.F. EvalMaxSAT[CP/OL] [2021-11-22].https://github.com/FlorentAvellaneda/EvalMaxSAT

[13] Cormen T H, Leiserson C E, Rivest R L, *et al*. Introduction to algorithms[M]. Cambridge: The MIT Press, 2009: 573

附录 A 最优性证明

首先证明 3 个基本定理，并由定理推导得到最优性的证明.

定理 1. 记由冲突图 G 构造出的 Max SAT 问题最优值为 $\mathrm{MaxSAT}(G)$. 则对于图 G 顶点集 V 的任意子集 V_{sub} 构成的导出子图 G' 及补集 $V-V_{\mathrm{sub}}$ 构成的导出子图 G''，有

$$\mathrm{MaxSAT}(G)\leqslant \mathrm{MaxSAT}(G')+\mathrm{MaxSAT}(G'') \tag{A1}$$

其中，等号成立的充分条件是任取顶点 $v\in V_{\mathrm{sub}}$，$u\in V-V_{\mathrm{sub}}$，有 $(u,v)\notin E$，E 是图 G 的边集.

证明. 若任取顶点 $v\in V_{\mathrm{sub}}$，$u\in V-V_{\mathrm{sub}}$，对于图 G 的边集 E，有 $(u,v)\notin E$，记 G' 的边集为 E'，G'' 的边集为 E''，则有 $E=E'\cup E''$，即两子图边集的并集与原图边集相等，因此构造出的 Max SAT 问题子句集与图 G 构造出的 Max SAT 问题的子句集是相同的，则两问题等价，等号成立；若存在 $v\in V_{\mathrm{sub}}$，$u\in V-V_{\mathrm{sub}}$，使得 $(u,v)\in E$，此时存在硬子句 $\neg v\vee\neg u$，则图 G 构成的原 Max SAT 问题中，软子句 u 和软子句 v 不可同时成立，于是原问题中子句成立个数不会大于两子图构成的问题子句成立数之和. 证毕.

定理 2. 对于图 G 的任一子图 G'，总有

$$\mathrm{MaxSAT}(G')\leqslant \mathrm{MaxSAT}(G) \tag{A2}$$

证明. 图 $G(V,E)$ 的补图为 $\bar{G}(V,\bar{E})$，对于任意顶点 $v\in V_{\mathrm{sub}}$，$u\in V-V_{\mathrm{sub}}$，若存在边 $(u,v)\in\bar{E}$，则与子图 G' 构成的 Max SAT 问题相比，原图构成的 Max SAT 问题中多出一个可满足的软子句 u，记这类边的集合为 E'，当 $E'\neq\varnothing$ 时，有下确界

$$\mathrm{MaxSAT}(G')+1\leqslant \mathrm{MaxSAT}(G) \tag{A3}$$

当 $E'=\varnothing$ 时，没有可以额外成立的子句，式(A2)等号成立.

证毕.

定理 3. 若图 G 为完全图，则其对应的 Max SAT 问题取得最优解时，软子句成立个数为 1.

证明. 在完全图 G_{complete} 中，任意 2 个顶点之间均存在一条冲突边. 任取 $\forall u,v\in G_{\mathrm{complete}}$，在由 G_{complete} 构造的 Max SAT 问题中，总有硬子句 $\neg u\vee\neg v$ 存在.

利用反证法，假设最优解至少有 2 个变量取值为真，则硬子句 $\neg u\vee\neg v$ 不成立，因此不是一个可行解，最优值不会比 1 更大. 对于由完全图构成的 Max SAT 问题，取得最优解时的软子句成立个数为 1. 证毕.

引理 1. 对于由完全图构造的 Max SAT 问题，可以直接确定其最优值为 1，且最优解为任一变量为真，其余变量为假.

证明. 由定理 3 易知. 证毕.

引理 2. 若存在顶点集的子集 $V_{\mathrm{sub}}\subset V$，使其构成的图 G 的导出子图 G' 为完全图，且存在 $v\in V_{\mathrm{sub}}$，$u\in V-V_{\mathrm{sub}}$，有 $(u,v)\notin E$，则 v 为真，对所有 $v'\in V_{\mathrm{sub}}-\{v\}$ 均为假，可为原问题最优解的一部分.

证明. 由定理 3 可知 $\mathrm{MaxSAT}(G')=1$. 记由 $V-V_{\mathrm{sub}}$ 构成的导出子图为 G''，因 $(u,v)\notin E$，故式(A1)等号成立，可得

$$\mathrm{MaxSAT}(G)=1+\mathrm{MaxSAT}(G'') \tag{A4}$$

因此，对每个这样的顶点集，均可直接确定相关联变量在原问题中最优解的一部分. 证毕.

引理 3. 按照连通分量构造出的多个 Max SAT 问题最优值之和与原图 G 构造出问题的最优值相等.

证明. 任意连通分量 $G_1,G_2,G_3,\cdots,G_n$ 均为原图的导出子图，由于任取 $v\in V_i$，$u\in V_j$，其中 $1\leqslant i,j\leqslant n$，且 $i\neq j$，有 $(u,v)\notin E$，故式(A1)的等号成立，可得

$$\mathrm{MaxSAT}(G)=\sum_{i=1}^{n}\mathrm{MaxSAT}(G_i) \tag{A5}$$

因此，求解原图构造的 Max SAT 问题等价于求解每个连通分量构造的 Max SAT 问题. 证毕.