# Two-Stage Layout Decomposition for Hybrid E-Beam and Triple Patterning Lithography

XINGQUAN LI and WENXING ZHU, Fuzhou University, China

Hybrid e-beam lithography (EBL) and triple patterning lithography (TPL) are advanced technologies for the manufacture of integrated circuits. We propose a technology that combines the advantages of EBL and TPL, which is more promising for the pattern product industry. Layout decomposition is a crucial step in this technology. In this article, we propose a two-stage decomposition flow for the hybrid e-beam and triple patterning lithography of the general layout decomposition (HETLD) problem. At the first stage, we formulate two optimization problems: the e-beam and stitch-aware TPL mask assignment (ESTMA) problem and the extended minimum weight dominating set for $R_4$ mask assignment (MDS$R_4$MA) problem. Binary linear program formulations of the two problems are solved by the cutting plane approach. At the second stage, solutions of the first stage problems are legalized to feasible solutions of the HETLD problem by stitch insertion and e-beam shot. To speed up decomposition, we reduce the problem size by removing some vertices and some minor conflict edges before decomposition. Experimental results show the effectiveness of our decomposition methods based on ESTMA and MDS$R_4$MA.

CCS Concepts: • **Hardware → Electronic design automation (EDA)**; **Design for manufacture (DFM)**; **Hybrid layout decomposition**;

Additional Key Words and Phrases: Triple patterning lithography, e-beam lithography, variable shaped beam, stitch

**ACM Reference format:**
Xingquan Li and Wenxing Zhu. 2017. Two-Stage Layout Decomposition for Hybrid E-Beam and Triple Patterning Lithography. *ACM Trans. Des. Autom. Electron. Syst.* 23, 1, Article 6 (July 2017), 23 pages.
https://doi.org/10.1145/3084683

## 1 INTRODUCTION

The major bottleneck that hinders faster and more powerful processor development is the design and manufacture technologies of integrated circuits (IC). At present, many manufacture technologies have been developed (Wilson 2013), such as the 193nm ArF immersion optical lithography (ArF-IOL) and the related multiple patterning lithography (MPL), electron beam lithography (EBL), directed self-assembly (DSA), and extreme ultraviolet lithography (EUVL). EUVL is considered a promising technology for next-generation lithography. However, due to various obstacles, EUVL still cannot be put into mass IC manufacture (Chang et al. 2015). Currently, MPL is among the most popular, because it is high throughput and low optimal exposure (Borodovsky 2009). On the



This work is supported by the National Natural Science Foundation of China under Grants 61672005 and 11331003.
Authors' addresses: X. Li and W. Zhu, Center for Discrete Mathematics and Theoretical Computer Science, Fuzhou University, Fuzhou 350108, China; emails: {n130320024, wxzhu}@fzu.edu.cn.

https://doi.org/10.1145/3084683

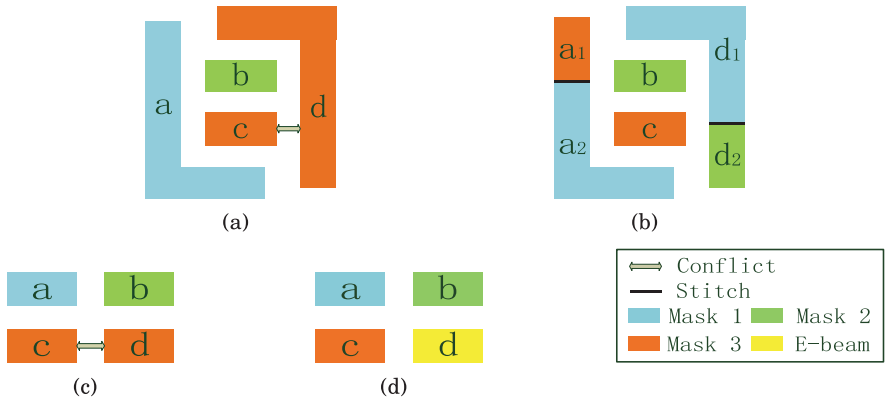


Fig. 1. Hybrid e-beam and triple patterning lithography layout decomposition. (a) A mask assignment for triple patterning lithography layout decomposition with a conflict. (b) An example of layout with stitch insertion for eliminating the conflict. (c) A mask assignment for TPL layout decomposition with a conflict. (d) A mask assignment for HETLD.

contrary, e-beam lithography has lower throughput but good for random complex patterns (Chase and Smith 2001). Since a solo lithography technology cannot achieve the measures (cost, throughput, timing, etc.) very well simultaneously, hybrid lithography has been introduced recently. One of the most promising technologies is the combination of MPL with EBL, which is a novel and practical choice for manufacture of IC (Steen et al. 2005). Compared with a solo lithography technology, hybrid lithography can produce better-quality circuit boards (Tian et al. 2014b; Yang et al. 2016).

LELELE-style lithography, that is, triple patterning lithography (TPL), is one type of MPL that has been proposed for quite a few years. To obtain high resolution, many methods have been proposed for TPL layout decomposition (TPLLD). For a general layout, Yu et al. (2015b) described the TPLLD problem and proved that it is NP-hard. To solve the problem, they introduced a semidefinite programming relaxation-based decomposition method. In Fang et al. (2014), Kuang and Young (2013), and Li et al. (2017), the authors proposed different heuristic methods to obtain decomposition solutions fast. In particular, for standard cells with a row structure layout, decomposers in Tian et al. (2014a), Yu et al. (2015a), and Chien et al. (2015) have considered layout decomposability in physical design stages.

Compared with MPL, EBL is a flexible lithography technique that prints patterns by use of mass electron beams. The conventional electron beam is a variable-shaped beam (VSB), which is a rectangle-based e-beam, and can only print rectangular patterns. As a result, EBL is low throughput. To raise throughput, many methods (Yuan et al. 2012; Yu et al. 2013; Mak and Chu 2014) have been proposed for the character-shaped beam (CSB). However, the character-shaped beam technique cannot substantially resolve the throughput issue of EBL. Hence EBL should be used as little as possible.

Figure 1 shows an example of hybrid e-beam and triple patterning lithography layout decomposition. Suppose patterns $a$, $b$, $c$, and $d$ in Figure 1(a) and Figure 1(c) are too close to each other. In Figure 1(a), patterns are assigned to three masks, but a conflict occurs between patterns $c$ and $d$. To eliminate the conflict, two stitches are inserted into patterns $a$ and $d$ to split them into two sub-patterns, respectively, as shown in Figure 1(b). However, for some dense layouts, stitch insertion may not eliminate conflicts. As shown in Figure 1(c), there is a conflict between patterns $c$ and $d$, and the conflict cannot be eliminated by inserting stitches. A manufacture plan is shown as Figure 1(d), in which pattern $d$ is printed by e-beam.

Over the years, some works have been done for hybrid lithography layout decomposition. For 1D layout structure, Du et al. (2012) constructed a mathematical formulation and proposed an iterative Integer Linear Program (ILP) algorithm to assign cuts for hybrid lithography with e-beam and 193nm immersion (193i) single exposure. For 2D layout, Ding et al. (2014) investigated the layout decomposition for hybrid self-aligned double patterning lithography (SADP) and EBL by solving an elegant ILP formulation; Gao et al. (2014) considered simultaneously the e-beam cut cost and the trim cut cost and introduced a matching method for hybrid SADP and EBL layout decomposition.

Hybrid EBL and TPL was first investigated by Tian et al. (2014b). Their method is only for layouts with standard cells on rows. Recently, Yang et al. (2016) considered the hybrid EBL and TPL of general layout decomposition problem (HETLD) and proposed a random-initialized improvement local search method basing on their hybrid EBL and double patterning lithography decomposer. Before decomposition, their method divides every pattern into several rectangles using candidate stitches. This operation leads to the following two issues: (i) It would increase the size of the decomposition problem, and (ii) since candidate stitches are inserted at the corners of the patterns, the locations of the stitches are illegal because they would cause side effects (Yu et al. 2015b; Fang et al. 2014; Kuang and Young 2013; Li et al. 2017) and increase the manufacturing costs.

The HETLD problem looks like the TPLLD problem; however, there are many differences between the HETLD problem and the TPLLD problem. First, the TPLLD problem uses stitch insertion for eliminating conflicts, while the HETLD problem uses e-beam and stitch insertion. Second, the primary objective of the HETLD problem is minimizing the sum of VSB numbers, while the TPLLD problem is minimizing the total number of conflict edges. Furthermore, when decomposing a layout for the HETLD problem, if a pattern is assigned to e-beam, then stitch cannot be used for reducing the VSB number of the pattern. While for the TPLLD problem, if a pattern conflicts with some other patterns, then stitch still can be used to reduce the number of conflicts. Finally, for the HETLD problem, if a pattern is assigned to e-beam, then the pattern has no conflicts with other patterns. Hence, the discrete relaxation-based decomposition method for the TPLLD problem (Li et al. 2017) cannot be used directly for the HETLD problem.

In this article, we consider the hybrid e-beam and TPL of general layout decomposition problem. We propose a two-stage decomposition flow for the problem. At the first stage, we consider an e-beam and stitch-aware TPL mask assignment (ESTMA) problem, and then the problem is formulated as a binary linear program and solved by the cutting plane approach. At the second stage, the solution is legalized to a feasible solution of the HETLD problem by stitch insertion and e-beam shot. In addition, some graph reduction techniques proposed by previous TPL layout decomposers (Yu et al. 2015b; Fang et al. 2014; Kuang and Young 2013; Li et al. 2017) are used to reduce the problem size. Moreover, a new graph reduction that deletes some minor conflict edges is proposed to further speed up the decomposition flow. Furthermore, to obtain a better solution with a smaller VSB number, we propose an extended minimum weight dominating set for the $R_4$ mask assignment (MDS$R_4$MA) problem, which is also formulated as an ILP. In the first stage, if we solve the MDS$R_4$MA problem instead of the ESTMA problem, then more patterns can be assigned to TPL masks by inserting stitches. Experimental results show the effectiveness of the ESTMA and the MDS$R_4$MA-based decomposition methods. In addition, it must be noted that the two issues in Yang et al. (2016) are avoided in this article.

The rest of this article is organized as follows. In Section 2, we describe the problem, introduce the concepts of conflict pattern and native conflict structure, and analyze some properties of the problem. In Section 3, we propose two problems: e-beam and stitch-aware TPL mask assignment problem and extended minimum weight dominating set for $R_4$ mask assignment problem and formulate them as integer linear programming problems. In Section 4, we detail the stitch and

e-beam assignment for the patterns that are not resolved in the first decomposition stage. Section 5 introduces some graph reduction techniques and shows our hybrid decomposition flow. Experimental results are presented in Section 6, and conclusions are made in Section 7.

## 2 PRELIMINARIES

In this section, first we describe the HETLD problem, and then we introduce some concepts and analyze properties of the HETLD problem.

### 2.1 Hybrid e-beam and TPL Layout Decomposition Problem

Given a layout $L$, the minimum coloring spacing $\min_{cs}$ rule is that if the distance between two patterns is less than $\min_{cs}$, then there exists a conflict between the two patterns. The HETLD problem is that the patterns in $L$ are assigned to three TPL masks with stitch insertions to eliminate most of the conflicts, and, to totally eliminate conflicts, e-beam shots are used to print some patterns.

The HETLD problem can be such that, all patterns in $L$ are assigned to four different colors, that is, three colors for TPL and one color for EBL. For the three colors for TPL, stitch insertion can be used to split patterns into several sub-patterns, and the distance between any two patterns or sub-patterns in the same TPL color should be greater than $\min_{cs}$.

Since stitch will lead to potential functional errors of a chip, and increase the manufacture cost (Tian et al. 2014b), the number of stitches should be minimized. In addition, the EBL system mainly applies the VSB technique to print patterns, which is low throughput due to a one-by-one print process (Yu et al. 2013; Maruyama et al. 2012). Hence, to maximize the throughput, the number of VSB shots should be minimized. It must be noted that VSB is a rectangle-shaped electron beam, hence a pattern should be split into a set of rectangles (Du et al. 2012; Ding et al. 2014; Fujimura 2010; Inanami et al. 2003). That is, the number of rectangles of patterns printed by an e-beam is equal to the number of VSBs (Tian et al. 2014b). Thus, in the hybrid TPL and EBL layout decomposition, the number of VSBs and the number of stitches should be minimized for low cost and high throughput of manufacture. Formally, the HETLD problem is described as follows.

**Hybrid e-beam and triple patterning lithography layout decomposition problem $P_0$:**

**Given:** Layout $L$, the minimum coloring spacing $\min_{cs}$, the minimum pattern size $\min_{ps}$, and the minimum overlap margin $\min_{ov}$.

**Find:** A color assignment of patterns in layout $L$ to three colors for TPL with stitch insertions and one color for EBL, subject to the following: (i) any two patterns or sub-patterns within $\min_{cs}$ should not be assigned to the same TPL color; (ii) a pattern in $L$ should be printed only by TPL or EBL; and (iii) the location of stitch insertion should be legal.

**Objective:** Mainly minimize the number of VSBs, that is, $|VSB|$, and, second, minimize the number of stitches, that is, $|S|$.

In the problem, the second constraint means that, sub-patterns of a pattern should be produced by the same manufacture technique, that is, either by TPL or by EBL. This is due to the fact that producing a pattern using two different manufacturing techniques will induce greater manufactureing costs (Yang et al. 2016). Here, we show an illegal mask assignment as Figure 2. In the figure, sub-pattern $d_1$ is printed by an e-beam shot, but $d_2$ is printed by TPL exposure.

For the location of an inserted stitch, it should satisfy that (Li et al. 2017) (i) a generated sub-pattern must be larger than the minimum pattern size $\min_{ps}$, (ii) the location of an inserted stitch should not be near any corner of a pattern, and (iii) the overlap length should be greater than the minimum overlap margin $\min_{om}$. Here the overlap length means that an inserted stitch can be moved within some range without causing any new conflict, and the length of the range is called the overlap length (Kuang and Young 2013).

To show the complexity of the HETLD problem, first we consider the minimum weight vertex removal 3-coloring (MWVR3C) problem. The decision problem of the MWVR3C problem is

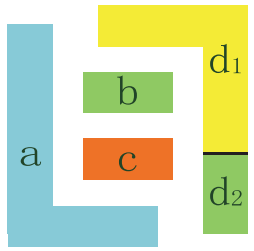


Fig. 2. An illegal mask assignment for HETLD.

that, given a vertex-weighted undirected graph $G(V, E)$ and a constant $K$, we ask if there exists a subset $V'$ of $V$, such that the sum of weights of vertices in $V'$ is less than or equal to $K$, and the induced subgraph $G[V - V']$ of $G$ is 3-colorable. Since deciding whether a graph is 3-colorable is NP-complete (Garey et al. 1976), the MWVR3C problem is NP-hard. Furthermore, the MWVR3C problem can be reduced easily to the HETLD problem. The detail of reduction is the same as that in Yu et al. (2015b), in which the planar graph 3-coloring problem is reduced to the triple patterning layout decomposition problem. Hence, the HETLD problem is NP-hard, which means that it cannot be solved in polynomial time unless P = NP.

### 2.2 Conflict Pattern and Native Conflict

Given a layout $L$, according to the minimum coloring spacing $\min_{cs}$ rule, we transform the geometric layout structure to a conflict graph $CG(V, E)$, where $V$ is the set of patterns, and $E$ is the set of conflict edges between any two patterns. In the conflict graph $CG$, two patterns where a conflict edge exists should be assigned to different masks. However, there might be some patterns that cannot be assigned to different masks to eliminate conflicts. They should be inserted stitches or assigned to an e-beam. Furthermore, there might be conflicts at some patterns that cannot be totally eliminated by inserting stitches. Some of these patterns should be assigned to an e-beam.

We present a graph structure where conflicts between patterns cannot be resolved by inserting stitches. Before that, some definitions from Li et al. (2017) are introduced as follows.

*Definition 2.1 (Li et al. 2017) (Conflict Region (CR)).* The conflict region of a pattern is defined as a two-dimensional (2D) region around the pattern, which is within the minimum coloring spacing $\min_{cs}$ of the pattern.

Figure 3(a) shows an example of a conflict region, where the round rectangle region of pattern $c$ is the conflict region of $c$, and the red dashed box in Figure 3(a) is the intersection of pattern $d$ and $CR$ of pattern $c$.

As we know, $K_4$ is the smallest 3-uncolorable structure, and a 3-uncolorable graph would generate conflicts for TPL layout decomposition. Actually, for TPL, most of conflicts would be generated from $K_4$. The $K_4$ structure is introduced as follows.

*Definition 2.2 (Li et al. 2017) (Conflict Pattern (CP)).* Pattern $v$ is called a conflict pattern if it satisfies the two conditions: (i) On pattern $v$ there is an intersection of conflict regions of three other different patterns, and (ii) the sub-graph induced by pattern $v$ and the three patterns is a $K_4$ graph. The three patterns are called conflict adjacent patterns $CAP$ of pattern $v$.

*Definition 2.3 (Li et al. 2017) ($K_4$ Conflict Structure ($K_4$CS)).* A graph structure is a $K_4$ conflict structure if (i) it is a $K_4$ structure, (ii) all of the four patterns are conflict patterns $CP$, and (iii) the four patterns are $CAP$ each other.

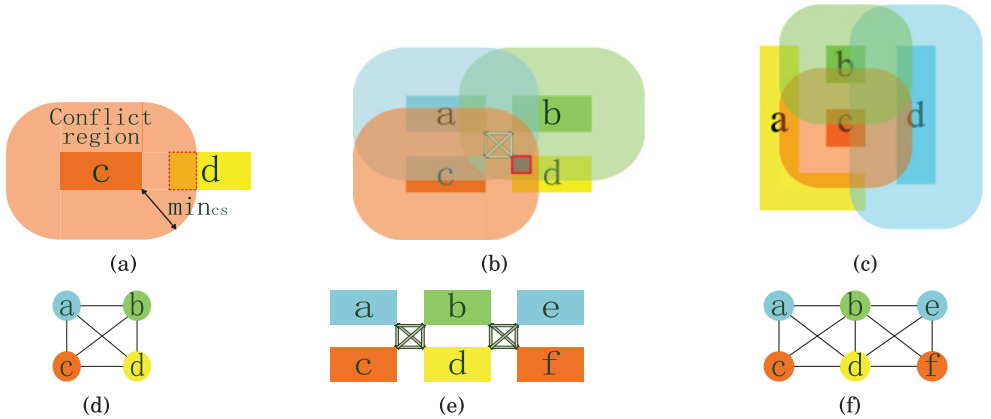


Fig. 3. Conflict region and conflict patterns. (a) Conflict region. (b) Conflict pattern. (c) An example of non-conflict pattern. (d) $K_4$ conflict structure. (e)(f) Native conflict structure.

As Figure 3(b) shows, pattern $d$ is a *CP*, since there is a red box on pattern $d$, which is the intersection of conflict regions of patterns $a$, $b$, and $c$. Actually, patterns $a$, $b$, and $c$ also are *CP*, and the four patterns compose a $K_4$. Furthermore, the four patterns are *CAP* each other. Thus, the structure composed by patterns $a$, $b$, $c$, and $d$ is a $K_4CS$. For comparison, we show a non-conflict pattern *NCP* as the pattern $a$ in Figure 3(f), in which pattern $a$ belongs to a $K_4$ graph, but on pattern $a$ there does not exist an intersection of conflict regions of patterns $b$, $c$, and $d$.

*Definition 2.4 (Native Conflict Structure (NCS)).* A native conflict structure *NCS* consists of one or more connected $K_4$ conflict structures. A $K_4CS$ is the smallest *NCS*.

By Definition 2.4, the structures in Figure 3(b) and Figure 3(d) are native conflict structures *NCS*. Figure 3(b) consists of one $K_4CS$ $\{a, b, c, d\}$, and Figure 3(d) consists of two $K_4CS$s $\{a, b, c, d\}$ and $\{b, e, d, f\}$, since $\{a, b, c, d\}$ and $\{b, e, d, f\}$ are connected at vertices $b$ and $d$.

Next, we show a relationship between *NCS* and the e-beam as follows.

Theorem 2.5. *At least a pattern in $K_4CS$ should be printed by e-beam shots.*

Proof. Since $K_4CS$ is a $K_4$ structure, it is not 3-colorable, and there is at least a conflict between two of the four patterns after coloring. Moreover, from Li et al. (2017), we know that the conflicts between the four patterns cannot be totally eliminated by stitch insertions. Hence at least a pattern in $K_4CS$ cannot be printed by TPL, which must be printed by e-beam. □

According to the definition of *NCS* and Theorem 2.5, we have a corollary as follows.

Corollary 2.6. *At least a pattern in NCS should be printed by e-beam shots.*

Given a layout, we cannot distinguish which conflict can be eliminated by stitch insertion, but Theorem 2.5 and its corollary provide sufficient conditions for finding patterns that should be printed by e-beam. The two sufficient conditions are critical for our e-beam and stitch-aware TPL mask assignment in the next section. Hence, before coloring, it is significant to check the conflict patterns and native conflict structures, by which we can find potential unresolvable conflicts in a layout.

Checking the native conflict structures in a layout can be done by the Breadth First Search (BFS) algorithm, which traverses all vertices and finds all conflict patterns. For a conflict graph $CG(V, E)$, suppose $D = \max\{d_v, v \in V\}$ is the maximum degree of vertices, and then the computing time of

determining whether there exists a structure containing a pattern $v$ is $K_4$ is in time $O(D^3)$. It is easy to know that the runtime complexity of checking all $NCS$ in $CG(V, E)$ is $O(D^3 * |V|)$.

## 3 HYBRID E-BEAM AND TPL MASK ASSIGNMENT METHODS

In this section, we introduce the first layout decomposition stage. For the HETLD problem, the minimum number of VSBs is the primary objective, and the minimum number of stitches is the secondary objective. For a large-scale case of the HETLD problem, it is not good that every pattern in a layout is split into several sub-patterns by candidate stitches before solving the problem, since this would increase the size of the problem. Hence, we introduce two mask assignment methods: (1) the ESTMA method and (2) the extended minimum weight dominating set for $R_4$ mask assignment (MDS$R_4$MA) method. The two methods consider implicitly the e-beam and stitch insertion in the first decomposition stage, and the concrete e-beam and stitch insertion will be considered in the second decomposition stage. Some involved notations are introduced as follows:

- $V$, the set of patterns in a conflict graph;
- $E$, the set of conflict edges in a conflict graph;
- $W$, the set of weights of patterns in a conflict graph;
- $VSB_i$, the number of VSBs (or the number of rectangles) of pattern $i$;
- $C_1, C_2, C_3$, the colors (masks) of TPL;
- $R_4$, the set of uncolored patterns;
- $\beta$, the weighting parameter between VSB and stitch numbers, which is set as $\beta = 0.01$ as in Yang et al. (2016).

### 3.1 ESTMA

For the HETLD problem, e-beam and stitch insertion can be seen as two kinds of conflict eliminating techniques, where the cost of e-beam is higher than that of stitch insertion. Hence, if we can distinguish which pattern will use stitch insertion to eliminate conflicts, and which pattern must use e-beam to eliminate conflicts, then the HETLD problem can be well addressed. However, we do not know about that before coloring. In this subsection, we introduce the ESTMA problem by assigning weights to all patterns. The objective of the ESTMA problem is minimizing the sum of weights of uncolored patterns, which implies minimizing the total cost of VSBs and stitches.

According to the analysis in Section 2.2 for conflict pattern $CP$ and $K_4$ conflict structure $K_4CS$, we know that for a $CP$, stitch insertion is almost useless for eliminating all conflicts, unless some of its conflict adjacent patterns $CAP$ are assigned to e-beam or stitches are inserted into its $CAP$. However, if a pattern is not a $CP$, stitch insertion is more likely to eliminate all conflicts. Thus, the weights of patterns are set as

$$w_i = \begin{cases} VSB_i, & \text{if } i \text{ is a conflict pattern;} \\ \beta VSB_i, & \text{if } i \text{ is not a conflict pattern.} \end{cases}$$

We divide patterns into four color classes without considering e-beam and stitch insertion directly. The objective is to minimize the sum of weights of patterns in $R_4$, that is, $\sum_{i \in R_4} w_i$. Thus, the ESTMA problem (1) is formulated as

$$\begin{aligned} \min \quad & \sum_{i \in R_4} w_i \\ \text{s.t.} \quad & \text{if } i, j \in C_k, \text{ then } i \notin A(j), \ k = 1, 2, 3; && \text{(1a)} \\ & i \in C_1 \cup C_2 \cup C_3 \cup R_4. && \text{(1b)} \end{aligned}$$

In the above formulation, $A(j)$ is the set of adjacent patterns of $j$ in the conflict graph $CG$; $i \in C_k$ means pattern $i$ is assigned to color $k$, $k = 1, 2, 3$; and $i \in R_4$ means pattern $i$ is uncolored. Constraint (1a) is used to force that any two touch patterns should be assigned different colors.

To solve problem (1) and obtain a solution, we formulate it as a binary linear program (BLP). Let $(x_{i1}, x_{i2})$ be a two-dimensional binary variable, which is used to represent the color of vertex $i$. When $(x_{i1}, x_{i2}) = (0, 1)$, it means $i \in C_1$; similarly, when $(x_{i1}, x_{i2}) = (1, 0)$, it means $i \in C_2$; and when $(x_{i1}, x_{i2}) = (1, 1)$, it means $i \in C_3$. However, when $(x_{i1}, x_{i2}) = (0, 0)$, it means $i \in R_4$. Then problem (1) is equivalent to the following binary problem (2).

$$\begin{aligned}
\min \quad & \sum_{i \in V} w_i y_i & & \\
\text{s.t.} \quad & x_{i2} - x_{i1} + x_{j2} - x_{j1} \le 1, & \forall e_{ij} \in E; & \quad (2a)\\
& x_{i1} - x_{i2} + x_{j1} - x_{j2} \le 1, & \forall e_{ij} \in E; & \quad (2b)\\
& x_{i1} + x_{i2} + x_{j1} + x_{j2} \le 3, & \forall e_{ij} \in E; & \quad (2c)\\
& 1 - x_{i1} - x_{i2} \le y_i, & \forall i \in V; & \quad (2d)\\
& (x_{i1}, x_{i2}) \in \{0, 1\}^2, y_i, \in \{0, 1\}, & \forall i \in V. & \quad (2e)
\end{aligned}$$

In the above equation, constraints (2a)–(2c) are equivalent to constraint (1a). That is, any two patterns within the same color class $C_k$ are not conflicting, $k = 1, 2, 3$. Constraint (2d) is used to force that, if $x_{i1} = 0$ and $x_{i2} = 0$, then $y_i = 1$; otherwise $y_i = 0$, since the objective is minimization and $w_i > 0$.

### 3.2 MDS$R_4$MA

According to the weighting rule for the ESTMA problem (1), it can be seen that for a solution of problem (1), if a conflict pattern $i$ is assigned to $R_4$, then $VSB_i$ will be added to the objective value. Actually, the *CP*s in $R_4$ might be assigned to TPL masks using stitch insertion, and then the total number of VSBs will decrease. We take an example to show this as follows.

For the layout given in Figure 4(a), all patterns are *CP*, and an optimal solution of the ESTMA problem (1) is shown in Figure 4(b), where conflict patterns $c$ and $g$ are assigned to $R_4$. In the second decomposition stage, since patterns $c$ and $g$ cannot be inserted stitches to eliminate conflicts, they are assigned to e-beam as in Figure 4(c). Then the total decomposition cost of the HETLD problem is the number of VSBs, that is, $|VSB| = 2$. However, there exists a feasible solution of problem (1) as shown in Figure 4(d), where conflict patterns $a$ and $g$ are assigned to $R_4$. For this solution, if pattern $g$ is assigned to e-beam, and pattern $a$ is inserted one stitch for coloring in the second stage, then $|VSB| = 1$, $|S| = 1$, and the total cost of the HETLD problem is $1 + \beta$, which is smaller than the solution shown in Figure 4(b).

Note that, for a feasible solution of problem (1), if a conflict pattern $i$ is in $R_4$, then it cannot be inserted stitches for TPL color assignment, unless some of its conflict adjacent patterns *CAP*s are assigned to e-beam. The reason is that if at least one of its *CAP*s are assigned to e-beam, then stitches might be inserted into $i$ for assigning the sub-patterns of $i$ to TPL colors. Thus, comparing with assigning some irrelevant *CP*s to $R_4$, like $c$ and $g$ in Figure 4(b), it is better to assign some of the *CP*s and their *CAP*s to $R_4$ simultaneously at the first decomposition stage, like $a$ and $g$ in Figure 4(d), if needed. And the second decomposition stage will deal with the patterns in $R_4$ by stitch insertion or e-beam shot.

For two patterns $i$ and $j$, if $i$ is a *CAP* of $j$ or $j$ is a *CAP* of $i$, then we call the edge between $i$ and $j$ a conflict adjacent edge and denote it by $cae_{ij}$. For example, every edge in Figure 4(f) is a conflict adjacent edge. Let $E_{ca}$ be the set of conflict adjacent edges. We introduce a graph $G_{R_4}(R_4, E_{R_4})$,

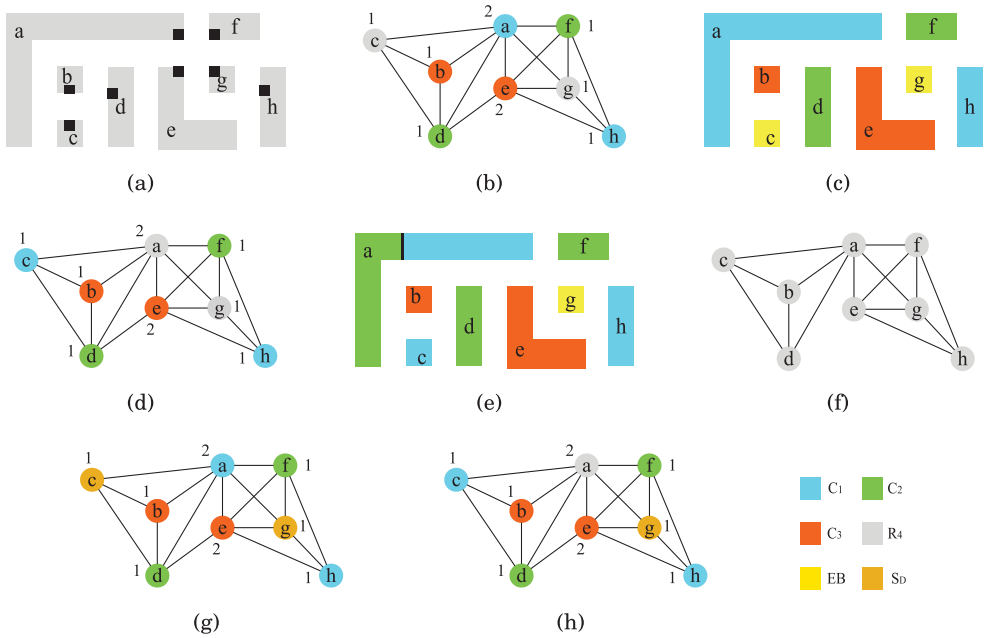


Fig. 4. A comparison of the ESTMA problem and the MDS$R_4$MA problem. (a) A layout where all patterns are *CP*. (b) A feasible solution of the ESTMA problem. (c) The decomposition result of (b) and (g). (d) A feasible solution of the ESTMA problem. (e) The decomposition result of (d) and (h). (f) The set of conflict adjacent edges $E_{ca}$. ((g) and (h)) Two feasible solutions of the MDS$R_4$MA problem.

which is induced by $R_4$ from graph $G(V, E_{ca})$, where $E_{R_4} \subset E_{ca}$. Since if a pattern $i \in R_4$ is assigned to e-beam, then some other patterns in $R_4$ connected to $i$ by $cae_{ij}$ may be assigned to TPL masks using stitch insertion. Hence we hope to assign the conflict patterns connected by conflict adjacent edges to $R_4$ at the first decomposition stage if needed.

Following this motivation, we propose the extended minimum weight dominating set for the $R_4$ mask assignment (MDS$R_4$MA) problem. A dominating set $S_D$ of a graph $G_{R_4}(R_4, E_{R_4})$ is a subset of $R_4$ such that every vertex not in $S_D$ is adjacent to at least one member of $S_D$. For the MDS$R_4$MA problem, every pattern in a layout would be assigned to one of the sets $C_1$, $C_2$, $C_3$, $S_D$, and $R_4 - S_D$, such that the conflict spacing rule is satisfied. The main objective is to minimize the total VSB number of patterns in $S_D$, and the secondary objective is to minimize the size of the set $R_4 - S_D$. The MDS$R_4$MA problem can be seen as a hybrid of the minimum weight dominating set problem and the 3-coloring problem, which can be formulated as

$$\begin{aligned} \min \quad & \sum_{i \in S_D} w_i + \beta(|R_4| - |S_D|) \\ \text{s.t.} \quad & \text{if } i, j \in C_k, \text{ then } i \notin A(j),\ k = 1, 2, 3; && \text{(3a)} \\ & i \in C_1 \cup C_2 \cup C_3 \cup R_4; && \text{(3b)} \\ & S_D \text{ is a dominating set of } G_{R_4}(R_4, E_{R_4}). && \text{(3c)} \end{aligned}$$

Obviously, any vertex in $R_4$ is either in $S_D$ or adjacent to at least a vertex in $S_D$. The objective of the problem is to minimize the sum of weights of vertices in the dominating set $S_D$ and $\beta(|R_4| - |S_D|)$, where the weight $w_i$ of vertex $i$ is the same as that in problem (1).

Consider the MDS$R_4$MA problem in the graph in Figure 4(g), where the coloring schemes of vertices in Figures 4(g) and 4(h) are two feasible solutions, respectively. For Figure 4(g), $G_{R_4}(R_4, E_{R_4})$ is the graph with $R_4 = \{c, g\}$ and $E_{R_4} = \emptyset$. $S_D = \{c, g\}$ is a dominating set of $G_{R_4}$, and the objective value of the MDS$R_4$MA problem is $\sum_{i \in S_D} w_i + \beta(|R_4| - |S_D|) = w_c + w_g = 2$. For Figure 4(h), $G_{R_4}(R_4, E_{R_4})$ is the graph with $R_4 = \{a, g\}$ and $E_{R_4} = \{(a, g)\}$. $S_D = \{g\}$ is a dominating set of $G_{R_4}$, and the objective value the MDS$R_4$MA problem is $\sum_{i \in S_D} w_i + \beta(|R_4| - |S_D|) = w_g + \beta(|R_4| - |S_D|) = 1 + \beta$. In fact, the solution as Figure 4(h) is an optimal solution of the MDS$R_4$MA problem.

Problem (3) is not in numerical form. To solve the problem, we formulate it as a BLP,

$$\begin{aligned}
\min \quad & \sum_{i \in V} w_i z_i + \beta \sum_{i \in V} (y_i - z_i) \\
\text{s.t.} \quad & x_{i2} - x_{i1} + x_{j2} - x_{j1} \le 1, && \forall e_{ij} \in E; && \text{(4a)} \\
& x_{i1} - x_{i2} + x_{j1} - x_{j2} \le 1, && \forall e_{ij} \in E; && \text{(4b)} \\
& x_{i1} + x_{i2} + x_{j1} + x_{j2} \le 3, && \forall e_{ij} \in E; && \text{(4c)} \\
& 1 - x_{i1} - x_{i2} \le y_i, && \forall i \in V; && \text{(4d)} \\
& x_{i1} + x_{i2} - 2 \le -2z_i, && \forall i \in V; && \text{(4e)} \\
& 1 - x_{i1} - x_{i2} - x_{j1} - x_{j2} \le \sum_{m \in A_{ca}(i) \cup \{i\}} z_m, && \forall i \in V, j \in A_{ca}(i) \cup \{i\}; && \text{(4f)} \\
& (x_{i1}, x_{i2}) \in \{0, 1\}^2, y_i, z_i \in \{0, 1\}, && \forall i \in V. && \text{(4g)}
\end{aligned}$$

In the above formulation, $(x_{i1}, x_{i2})$ is used to denote a color as in problem (2). $A_{ca}(i)$ is the set of vertices connected to $i$ by conflict adjacent edges. Constraints (4a)–(4c) are an equivalent formulation of constraint (3a). That is, any two patterns within the same color class $C_k$ are not conflicting, $k = 1, 2, 3$. Constraint (4d) is used to force that if $(x_{i1}, x_{i2}) = (0,0)$, that is, $i \in R_4$, then $y_i = 1$; otherwise, $y_i = 0$, since the objective is minimization and $w_i > 0$. Constraint (4e) is used to force that only if pattern $i \in R_4$, then $z_i$ may be equal to 1; otherwise, $z_i = 0$. Constraint (4f) is used to find a dominating set $S_D$ of $G_{R_4}(R_4, E_{R_4})$. If $z_i = 1$, then $i \in S_D$. If all adjacent patterns of pattern $i \in R_4$ are not in $R_4$, that is, all $j \in A_{ca}(i)$ are not in $R_4$, then $z_i = 1$. If there exists $j \in A_{ca}(i) \cap R_4$ such that $z_j = 1$, then $z_i = 0$, since the objective is minimization. If for all $j \in A_{ca}(i) \cap R_4$, $z_j = 0$, then $z_i = 1$, which means $i$ is a vertex in the dominating set.

We use the cutting plane approach in the software package GUROBI (Optimization 2014) to solve problems (2) and (4). Problems (2) and (4) are hard to solve in the large-scale case, especially for problem (4), since it has more variables and constraints. However, the graph reduction techniques in Section 5 can cut down the size of the problem such that it is easy to solve using the cutting plane approach. Here, we have the following result for problems (2) and (4).

Theorem 3.1. *Suppose $M$ is the number of native conflict structures NCS in a layout $L$. We have*

*(i) for the HETLD problem, its VSB number is at least $M$;*
*(ii) suppose $x^{R*}$ is an optimal solution of problem (2) or (4), then it holds that VSB$(x^{R*}) \ge M$.*

Proof. (i) For the initial layout $L$, by Theorem 2.5, at least $M$ patterns should be printed by e-beam, since there are $M$ native conflict structures $NCS$ in the layout $L$. Moreover, every one of these patterns should be printed by at least a VSB shot. Hence at least $M$ VSB shots should be used to eliminate the conflicts. So the total VSB number for the HETLD problem is not less than $M$.

(ii) For problem (2) or problem (4), all vertices of a $K_4CS$ in $E^R$ satisfy constraints (2a)–(2c) or (4a)–(4c). Since an $NCS$ is not 3-colorable, an optimal solution $x^{R*}$ of problem (2) or problem (4) includes at least $M$ components with $x_i^{R*} \in R_4$, and $VSB_i \geq 1$ ($\forall i \in M$) holds. Thus $VSB(x^{R*}) \geq M$. □

## 4 LEGALIZATION BY STITCH INSERTION, E-BEAM SHOT, AND BACKTRACK COLORING

In the second decomposition stage, a solution $x^R = (x_1^R, x_2^R, \ldots, x_n^R)$ of problems (2) and (4) may be infeasible for the HETLD problem. This is caused by the following: First, there may exist some conflicts between patterns due to the edge deletion of the relaxed conflict graph $RG(V, E^R, W)$ (introduced in Section 5.1.2). Second, the patterns in $R_4$ are uncolored and should be assigned to TPL masks by stitch insertion or printed by e-beam shot.

Hence an infeasible solution $x^R$ must be legalized to a feasible solution $x^H = (x_1^H, x_2^H, \ldots, x_n^H)$ of the HETLD problem. First, we deal with the conflicts between patterns by stitch insertion. If a conflict cannot be eliminated by stitch insertion, then for the two patterns causing the conflict edge, we assign one of them with a smaller VSB number to e-beam shot. After all conflicts having been eliminated, we assign the patterns in $R_4$ to TPL masks by stitch insertion or assign to e-beam. In addition, to achieve a better solution, a backtrack coloring method is used, which is a local swap-based method.

### 4.1 Conflict Elimination

In an optimal solution $x^R$ of problem (2) or problem (4), most of the components satisfy constraint (1a) or constraint (3a), but there still exist some components that violate the constraint. Suppose that components $p$ and $q$ of $x^R$ satisfy $x_p^R = x_q^R \in C_k$ ($k \in \{1, 2, 3\}$), and $(p, q) \in E$. That means $(p, q)$ is a conflict edge and $p$ and $q$ are assigned to the same TPL color, and there is a conflict between $p$ and $q$. We introduce stitch insertion or e-beam shot to patterns $p$ or $q$ to eliminate the conflict. There are three cases:

(1) Stitches are inserted into one of the patterns $p$ or $q$, say, $p$. Pattern $p$ is split into several sub-patterns $p_1, p_2, \ldots, p_m$. These sub-patterns $p_i$, $i = 1, 2, \ldots, m$, can be divided into two classes according to the distance between $p_i$ and $q$, $i = 1, 2, \ldots, m$: (i) one class of patterns is that the distance is less than or equal to $\min_{cs}$, and then the TPL colors of these sub-patterns should differ from $q$; (ii) another one is that the distance is greater than $\min_{cs}$, and then the TPL colors of these sub-patterns may be the same as that of pattern $q$.
(2) Both patterns $p$ and $q$ are split into several sub-patterns using stitches. Too close sub-patterns (within $\min_{cs}$) should be assigned different TPL colors.
(3) Stitch insertion cannot eliminate the conflict between the two patterns. That means one of $p$ and $q$ should be printed by e-beam. We choose the pattern with the smaller VSB number for e-beam shot.

Note that if one of patterns $p$ and $q$, say, $p$, is a $CP$, and $p$ and three patterns from its $CAP$ compose a $K_4$, then inserting stitches into $p$ cannot eliminate conflicts. Motivated by this fact, for patterns $p$ and $q$ that satisfy $(p, q) \in E$ and $p$ and $q$ are assigned to the same TPL color in the infeasible solution, we propose a conflict elimination algorithm as Algorithm 1 to insert stitches to patterns $p$ or $q$ or assign e-beam to $p$ or $q$.

In Algorithm 1, the function $genCSI()$ (line 2) is used to generate potential candidate stitch insertions for every pair of patterns $p$ and $q$ that satisfy that $(p, q) \in E$ and $p$, $q$ are assigned to the same TPL color. In Li et al. (2017), the authors proposed an algorithm to generate all possible candidate

**ALGORITHM 1:** Conflict Elimination

**Input:** Solution $x^R$ of problem (2) or (4) on $RG$.
**Output:** Solution $x^{R'}$ without conflict.
**for** *every* $(p, q) \in E$ *and* $x_p^R = x_q^R \in C_k$ **do**
    $CSIs = genCSI(p, q)$ for patterns $p$ and $q$;
    $NS_0 \leftarrow +\infty$;
    **for** *every CSI of* $p$ *and* $q$ **do**
        **if** *conflict between* $p$ *and* $q$ *is eliminated and there is no new conflict generated by inserting CSI* **then**
            **if** $NS_{CSI} < NS_0$ **then**
                $NS_0 \leftarrow NS_{CSI}$, and store current $CSI$;
            **end**
        **end**
    **end**
    **if** $NS_0 < +\infty$ **then**
        insert $CSI$ with stitch number $NS_{CSI} = NS_0$ to patterns $p$ or $q$, and the generated sub-patterns are assigned to TPL masks $C_k$, $k = 1, 2, 3$;
    **else**
        **if** $VSB_p < VSB_q$ **then**
            pattern $p$ is assigned to e-beam;
        **else**
            $q$ is assigned to e-beam;
        **end**
    **end**
**end**

stitch insertions $CSI$ for a pattern $i$, we call it $genCSI(i)$. The algorithm first finds all conflict regions on pattern $i$ and then finds the horizontal or vertical line segments on pattern $i$, which are tangent to a conflict region. A line segment at pattern side is called a Checking Stitch Edge ($CSE$). The combination of one or more $CSE$s on pattern $i$ is called a Candidate Stitch Insertion ($CSI$). A Candidate Stitch Insertion on a pattern will produce sub-patterns with the minimum number of conflicts.

In this article, we use the corresponding algorithm in Li et al. (2017) as the function $genCSI()$ to generate all candidate stitch insertions of patterns $p$ and $q$, that is, $CSIs = genCSI(p, q)$ (line 2). A candidate stitch insertion $CSI$ is a stitch insertion plan that may include some stitches on patterns $p$ and $q$. $NS_{CSI}$ (line 6) is the stitch number of a certain $CSI$. $NS_0$ is an intermediate variable. Of course, stitches generated by $genCSI()$ should satisfy the stitch location condition: A candidate stitch insertion is not near the periphery or the corner of the pattern and should be in the overlap margin of the pattern that is greater than $\min_{om}$.

In Algorithm 1, first we check stitch insertions for a pair of patterns $p$ and $q$ (lines 2–10). If there are some insertion plans that can totally eliminate the conflict between $p$ and $q$, then we choose the insertion plan with the minimum stitch number to eliminate the conflict (lines 11 and 12); otherwise, we consider e-beam shot to print one of $p$ and $q$ that has a smaller VSB number (lines 13–15), and let another one be printed by TPL.

### 4.2 Assignment of Patterns in $R_4$

After removing all conflicts of the solution $x^R$, to obtain a feasible solution $x^H$ of the HETLD problem, we assign the patterns in $R_4$ to TPL colors or e-beam shot. The patterns in $R_4$ are dealt with one by one. We check whether every pattern $i \in R_4$ can be divided into several sub-patterns

by stitch insertion such that these sub-patterns can be assigned to TPL masks without generating conflicts. If not, then $i$ is assigned to e-beam. The details are as Algorithm 2 shows, which is similar to Algorithm 1, where we only consider a pattern $i$ instead of a pair of patterns $p$ and $q$. The explanations are the same as those of Algorithm 1 and are skipped here. It must be noted that Algorithm 2 shows the assignment of patterns in $R_4$ for the ESTMA problem (2). For the MDS$R_4$MA problem (4), the statement "every pattern $i \in R_4$" (line 1) in Algorithm 2 would be replaced by "every pattern $i \in R_4 - S_D$, then every pattern $i \in S_D$."

**ALGORITHM 2:** Assignment of Patterns in $R_4$

**Input:** Legalized solution $x^{R'}$ from Algorithm 1.
**Output:** Feasible solution $x^H$.
**for** *every pattern* $i \in R_4$ **do**
  $CSIs = genCSI(i)$ for pattern $i$;
  $NS_0 \leftarrow +\infty$;
  **for** *every CSI of* $i$ **do**
    **if** *there is no conflict generated from* $i$ *by inserting CSI* **then**
      **if** $NS_{CSI} < NS_0$ **then**
        $NS_0 \leftarrow NS_{CSI}$, and store current $CSI$;
      **end**
    **end**
  **end**
  **if** $NS_0 < +\infty$ **then**
    insert $CSI$ with stitch number $NS_{CSI} = NS_0$ to pattern $i$, and the generated sub-patterns are assigned to TPL masks $C_k$, $k = 1, 2, 3$;
  **else**
    pattern $i$ is assigned to e-beam;
  **end**
**end**

### 4.3 Backtrack Coloring

In the above legalization algorithms, the quality of an obtained solution depends on the legalization order. Moreover, due to stitch insertion, some removed patterns at the graph reduction stage introduced in the next section may have to be assigned to e-beam. However, the VSB number resulting from the above two cases could be reduced by a backtrack coloring method, which tries to further reduce the VSB number of the solution $x^H$ obtained from Sections 4.1 and 4.2. This method is detailed as Algorithm 3.

Algorithm 3 aims at finding a better solution with a smaller total VSB number by searching the adjacent patterns of pattern $i \in EB$ of $x^H$. In this algorithm, $totalVSB(C_k, i)$ is the total VSB (rectangle) number of patterns in color class $C_k$ connected to $i$, and $VSB_i$ is the VSB (rectangle) number of pattern $i$. Line 8 is used to perform stitch insertions or e-beam assignment for pattern $j \in C_{k_0}$, which is connected to $i$.

## 5 GRAPH REDUCTION AND DECOMPOSITION FLOW

In this section, first, we introduce some vertex removal techniques and propose a new graph reduction technique that removes some edges. And then we show our decomposition flow.

**ALGORITHM 3:** Backtrack Coloring

**Input:** Solution $x^H$ obtained from Algorithm 2.
**Output:** Another feasible solution $x^{H'}$.
**for** *every pattern i assigned to EB in* $x^H$ **do**
  calculate $totalVSB(C_k, i)$ of all patterns in $C_k$ connected to $i$, $k = 1, 2, 3$;
  $k_0 \leftarrow \text{argmin}_{k=1,2,3}\{totalVSB(C_k, i)\}$;
  **if** $totalVSB(C_{k_0}, i) < VSB_i$ **then**
    $C_{k_0} = C_{k_0} \cup \{i\}$, $EB = EB - \{i\}$;
    **for** *every j connected to i and* $j \in C_{k_0}$ **do**
      $C_{k_0} = C_{k_0} - \{j\}$;
      Use lines 2-15 of Algorithm 2 for pattern $j$ to perform stitch insertions or e-beam assignment;
    **end**
  **end**
**end**

### 5.1 Graph Reduction

Generally, it is hard to solve directly the large-scale HETLD problem of general layout, due to the complexity of the problem. For tackling the large-scale problem, some techniques should be utilized first to preprocess the conflict graph to reduce the size of the problem.

*5.1.1 Vertex Removal.* We introduce some tricks to delete some easily colored vertices from the conflict graph, which have been popularly used to reduce the size of the TPL layout decomposition problem (Yu et al. 2015b; Fang et al. 2014; Kuang and Young 2013; Li et al. 2017):

- —Vertex with degree less than three removal (Yu et al. 2015b; Fang et al. 2014; Kuang and Young 2013; Li et al. 2017);
- —Contained vertex removal (Li et al. 2017);
- —Connected component calculation (Yu et al. 2015b; Fang et al. 2014; Kuang and Young 2013; Li et al. 2017).

These techniques are highly effective for reducing the problem size of the HETLD problem. Since vertices with degree less then three are easily colored for the 3-coloring problem, the operation *Vertex with degree less than three removal* will not lose the solution quality of the HETLD problem and is used repeatedly in our decomposition flow. The operation *Contained vertex removal* was introduced in Li et al. (2017), which aims at deleting contained vertices. The definition of *Contained vertex* is as follows.

*Definition 5.1 (Li et al. 2017) (Contained Vertex).* Given a graph $G(V, E)$, for a pair of vertices $i, j \in V$, suppose that $(i, j) \notin E$ and $A(i) \subseteq A(j)$, where $A(i)$ and $A(j)$ are the set of adjacent vertices of vertices $i$ and $j$, respectively. Then we call that vertex $i$ is contained in vertex $j$, vertex $i$ is called a contained vertex, and $j$ is called a containing vertex.

A contained vertex can be prior assigned the color of its containing vertex. The vertices with degree less than three and the contained vertices are deleted from the conflict graph before the TPL mask assignment stage. And they are colored as soon as the TPL mask assignment stage finishes. The order of coloring these vertices is in the reverse order of deleting them.
Another graph reduction technique is *connected component calculation*. Since the HETLD problem in different connected components is independent, we need to calculate the connected components for solving the problem more easily. Actually, the conflict graph can be divided into a

number of connected components after the operations *Vertex with degree less than three removal* and *Contained vertex removal*. Furthermore, our algorithm deals with the HETLD problem on the connected components one by one.

*5.1.2 Edge Deletion.* For the HETLD problem, e-beam and stitch are used to eliminate conflicts. To minimize VSB and stitch numbers, it is necessary to figure out the relationship among the conflict, stitch, and VSB. That is, we must decide which conflict edge can be eliminated by inserting stitches in a pattern and which conflict edge must be eliminated by VSB. According to our analysis in Section 2.2 and our empirical experiments, we have three meaningful observations:

(1) Conflicts are mainly generated due to $K_4$ structure, which is the smallest 3-uncolorable graph;
(2) For the native conflict structure *NCS* (including $K_4CS$), the conflicts in it cannot be totally eliminated by stitch insertions;
(3) Suppose that there are conflicts at pattern $i$. If $i$ is a conflict pattern (*CP*), then the conflicts at pattern $i$ cannot be totally eliminated by inserting stitches into this pattern; if $i$ is a non-conflict pattern (*NCP*), then the conflicts at pattern $i$ might be eliminated by inserting stitches into this pattern.

Inspired by the above observations, we construct a relaxed conflict graph $RG(V, E^R, W)$ by deleting some minor conflict edges of the weighted conflict graph $CG(V, E, W)$. In the weighted conflict graph, if a $K_4$ structure is not a $K_4CS$, and a conflict edge $(i, j)$ in the $K_4$ structure satisfies one of the following conditions, then $(i, j)$ is considered minor:

(1) at least one of $i$ and $j$ is not *CP*;
(2) both $i$ and $j$ are *CP*, but at least one of them has that its *CAP*s are not all in the $K_4$ structure.

For a $K_4$ structure, there may be more than one conflict edge satisfying the above conditions, but we only delete the minor conflict edge $(i, j)$ with the sum of weights of vertices $w_i + w_j$ less than those of the other minor conflict edges.
According to our statistics, many conflict edges could be deleted at this step. The relaxed conflict graph is constructed at the conflict pattern *CP* and $K_4$ conflict structure identifying step (Section 2.2). After that, a relaxed conflict graph $RG(V, E^R, W)$ is generated, where $E^R \subseteq E$. Then our ESTMA and MDS$R_4$MA problems are solved on $RG(V, E^R, W)$, respectively. The relaxed conflict graph is sparser than the original conflict graph by deleting some conflict edges. Thus, solving the ESTMA and MDS$R_4$MA problems on $RG(V, E^R, W)$ is easier than on $CG(V, E, W)$.

## 5.2 Flow for Hybrid EBL and TPL Layout Decomposition

For the HETLD problem of general layout, our decomposition flow is summed up as Figure 5. Given a layout $L$, according to the minimum coloring spacing $\min_{cs}$ rule, we transform the geometric layout structure to a conflict graph $CG(V, E)$, where $V$ is the set of patterns, and $E$ is the set of conflict edges between any two patterns. Some graph reduction techniques are used to reduce the size of the conflict graph. After that, the surface projection-based method (Li et al. 2017) is introduced to calculate conflict patterns and detect native conflict structures. Then we introduce an e-beam and stitch-aware TPL mask assignment problem for the reduced graph.

Based on the native conflict structure, a relaxed conflict graph is constructed by removing some conflict edges. On the relaxed conflict graph, the ESTMA problem and the MDS$R_4$MA problem are formulated. Furthermore, the 0-1 linear program of one of the above two problems are formulated for obtaining a solution. At last, stitch insertion and e-beam assignment are introduced to eliminate

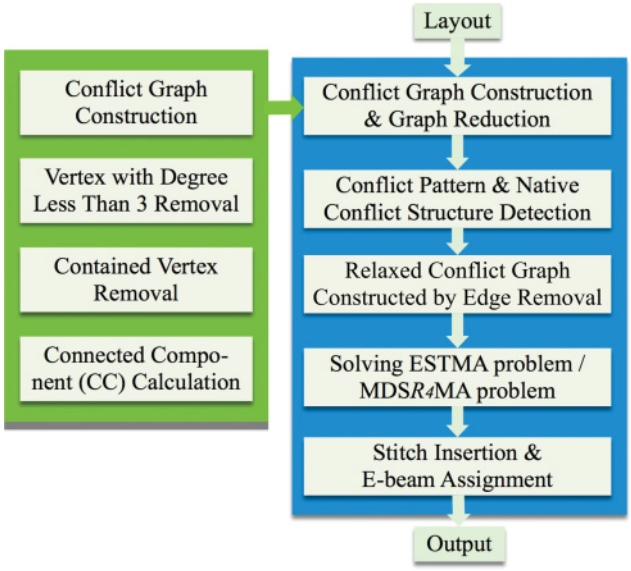


Fig. 5. Our HETLD decomposition flow.

all conflicts and obtain a higher resolution for the layout decomposition problem. The details of the decomposition flow are illustrated in the above sections.

### 5.3 An Example of the Two-Stage Decomposition Algorithm

In this section, for the HETLD problem, we give an example to illustrate the ESTMA-/MDS$R_4$MA-based decomposition methods described in Sections 2, 3, and 4.

Figure 6(a) is a layout with patterns $\{a, b, c, d, e, f, g\}$. According to the minimum coloring spacing rule, a conflict graph $CG(V, E)$ is constructed as in Figure 6(b), where

$$\begin{aligned} V &= \{a, b, c, d, e, f, g\}; \\ E &= \{(a,b), (a,c), (a,d), (b,c), (b,d), (b,e), (c,d), \\ &\quad (c,e), (c,f), (c,g), (d,e), (d,f), (e,f), (f,g)\}. \end{aligned}$$

Here, we skip the graph reduction techniques, focusing on the two decomposition stages.

*5.3.1 Decomposition Stage.* Before coloring, using the conflict identification method, we can find all $K_4$ subgraphs: $\{a, b, c, d\}$, $\{b, c, d, e\}$, $\{c, d, e, f\}$; all conflict patterns $CP$: $b, c, d, e, f$; and all $K_4CS$s: $\{b, c, d, e\}$, $\{c, d, e, f\}$, which are shown in Figure 6(c). Then all patterns are weighted by our weighting rule: $w(a) = 0.01$, $w(b) = 1$, $w(c) = 2$, $w(d) = 3$, $w(e) = 1$, $w(f) = 1$, $w(g) = 0.01$.

For the $K_4$ subgraph $\{a, b, c, d\}$, according to the relaxed conflict graph construction method, since pattern $a$ is not a $CP$, edge $(a, b)$ satisfies the conflict edge deletion condition, and the weights of $a$ and $b$ are $w(a) = 0.01$ and $w(b) = 1$, respectively, and $(a, b)$ will be deleted. Then we obtain the relaxed conflict graph $RG(V, E^R, W)$ as Figure 6(d), where $E^R = E - \{(a, b)\}$.

By solving problem (2) or problem (4) (the two problems have the same solution in this case), we obtain a solution $x^R$ with the minimum sum of weights: $x^R = (c_3, c_3, c_2, c_1, r_4, c_3, c_1)$, where $c_k$ means the corresponding vertex is in the color class $C_k$, $k = 1, 2, 3$, and $r_4$ means the corresponding vertex is not colored. Hence $C_1 = \{d, g\}$, $C_2 = \{c\}$, $C_3 = \{a, b, f\}$, $R_4 = \{e\}$.

As Figure 6(e) shows, color $C_1$ is blue, $C_2$ is green, $C_3$ is orange, and $R_4$ is uncolored.

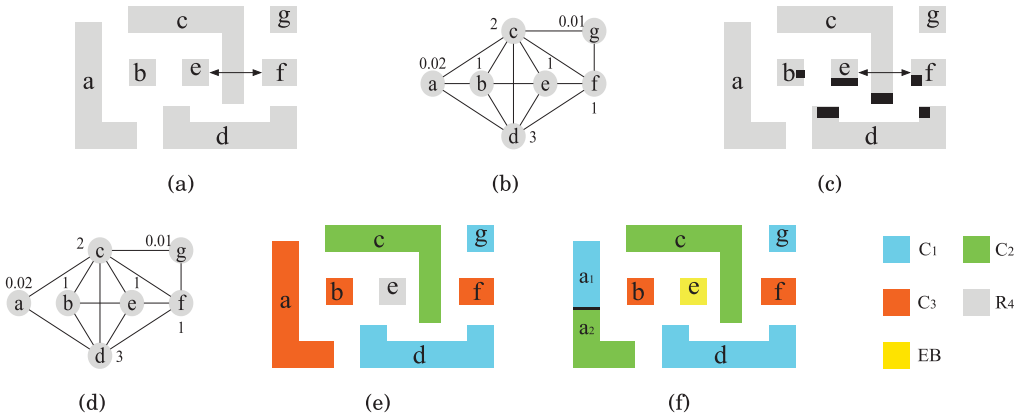


Fig. 6. A sample of HETLD flow. (a) Initial layout. (b) Conflict graph. (c) Conflict pattern identification. (d) Relaxed conflict graph. (e) Solution of problem (2) or problem (4). (f) Feasible solution of HETLD.

*5.3.2 Legalization Stage.* However, patterns $a$ and $b$ in $C_3$ are infeasible for the HETLD problem. Hence, we first consider inserting stitches to patterns $a$ and $b$. By Algorithm 1, we find a stitch insertion plan as Figure 6(f), where a stitch is inserted into pattern $a$, and $a$ is split into two sub-patterns $a_1$ and $a_2$. Then $a_1$ is assigned to color class $C_1$, and $a_2$ is assigned to color class $C_2$. Furthermore, stitch insertion is invalid for pattern $e$ by calling Algorithm 2. Thus $e$ is assigned to e-beam. Finally, we obtain a feasible solution $x^H$ of the HETLD problem:

$$x^H = (\{c_1, c_2\}, c_3, c_2, c_1, eb, c_3, c_1);$$
$$C_1 = \{a_1, d, g\}, C_2 = \{a_2, c\}, C_3 = \{b, f\}, EB = \{e\}.$$

For the solution $x^H$, the VSB number is $\sum_{i \in EB} VSB_i = VSB_e = 1$, and the stitch number is 1. Actually, it can be seen that the solution is an optimal solution of the HETLD problem for this layout.

## 6 EXPERIMENTAL RESULTS

Our decomposition methods for the hybrid e-beam and triple patterning lithography of general layout is programmed in C++ and run on a personal computer with 2.7GHz CPU, 8GB memory, and the Unix operating system. We test our method on the ISCAS-85 and 89 benchmarks provided by Yu et al. (2015b). In this article, the minimum coloring spacing is set as 160nm, and the minimum pattern size $\min_{ps}$ and the overlap margin $\min_{om}$ are set as 10nm.

Since this article aims at hybrid decomposition for general layout, we ignore the row structures of the test benchmarks. Note that EBL is low throughput, and the general e-beam is a VSB, which means that if a pattern is printed by e-beam shot, then it would be printed by several VSBs. Therefore, for the purpose of throughput, we use the number of VSBs mainly to evaluate the performance of the compared methods. Moreover, stitch may lead to potential functional errors of a chip during manufacture, hence the number of stitches is another comparison criterion.

### 6.1 Statistics and Analysis

Since one concern of this article is reducing the size of the problem, we compare some statistics of the initial conflict graphs and the relaxed conflict graphs. The statistics on all benchmarks are listed in Table 1. The data in the columns "#P" and "#E" are the numbers of patterns and conflict edges in the conflict graphs, respectively. The data in the column "Ratio" of "initial conflict graph" and "after graph reduction" are the ratios between the numbers of edges and patterns. Every datum

Table 1. Statistics of Hybrid e-beam and TPL Layout Decomposition Benchmarks with $\min_{cs}$ = 160nm

| | Initial conflict graph | | | | After graph reduction | | | | After Relaxing | |
|---|---|---|---|---|---|---|---|---|---|---|
| Benchs | #P | #E | Ratio | #ANC | #P | #E | Ratio | #ANC | $\#E^R$ | $\text{Ratio}^R$ |
| C432 | 1109 | 2160 | 1.95 | 65 | 478 | 966 | 2.02 | 19 | 890 | 1.86 |
| C499 | 2126 | 4590 | 2.16 | 41 | 1342 | 2827 | 2.11 | 21 | 2549 | 1.9 |
| C880 | 2411 | 4434 | 1.84 | 35 | 1187 | 2282 | 1.92 | 17 | 2162 | 1.82 |
| C1355 | 3262 | 5906 | 1.81 | 36 | 1323 | 2583 | 1.95 | 17 | 2453 | 1.85 |
| C1908 | 5125 | 8846 | 1.73 | 36 | 1580 | 3049 | 1.93 | 14 | 2914 | 1.84 |
| C2670 | 7933 | 14480 | 1.83 | 36 | 3896 | 7687 | 1.97 | 18 | 7302 | 1.87 |
| C3540 | 10189 | 17798 | 1.75 | 38 | 4389 | 8171 | 1.86 | 13 | 7758 | 1.77 |
| C5315 | 14603 | 26467 | 1.81 | 40 | 6686 | 12768 | 1.91 | 17 | 11940 | 1.79 |
| C6288 | 14575 | 26038 | 1.79 | 38 | 5251 | 10033 | 1.91 | 16 | 9476 | 1.8 |
| C7552 | 21253 | 37930 | 1.78 | 43 | 9033 | 17336 | 1.92 | 17 | 16365 | 1.81 |
| S1488 | 4611 | 8769 | 1.9 | 35 | 3020 | 5877 | 1.95 | 26 | 5461 | 1.81 |
| S38417 | 67696 | 126215 | 1.86 | 105 | 28978 | 61522 | 2.12 | 17 | 57507 | 1.98 |
| S35932 | 157455 | 317832 | 2.02 | 133 | 88188 | 192619 | 2.18 | 27 | 179621 | 2.04 |
| S38584 | 168319 | 314785 | 1.87 | 83 | 70121 | 151141 | 2.16 | 16 | 141199 | 2.01 |
| S15850 | 159952 | 309753 | 1.94 | 77 | 87216 | 184948 | 2.12 | 22 | 172449 | 1.98 |
| Avg. | 42708 | 81733 | 1.87 | 56 | 20846 | 44254 | 2.00 | 18 | 41336 | 1.88 |
| Ratio | 1.00 | 1.00 | 1.00 | 1.00 | 0.49 | 0.54 | 1.07 | 0.33 | 0.50 | 1.01 |

in column "#ANC" is the average number of patterns on the number of connected components. Every datum in column "$\#E^R$" is the number of conflict edges in the relaxed conflict graph, and every datum in column "$\text{Ratio}^R$" is the ratio between the number of edges and patterns in the relaxed conflict graph.

From Table 1, compared with the initial conflict graph, the number of patterns and the number of conflict edges are only half left after graph reduction, which shows that the graph reduction techniques used are effective. Note that conflict edges are removed at two stages, one is at the graph reduction stage and another one is at the relaxed conflict graph (*RG*) construction stage. From the column "Ratio" of "initial conflict graph," the average value is 1.87, which means the number of total conflict edges is nearly 1.87 times as many as the number of total patterns for every benchmark. From the column "Ratio" of "after graph reduction," the average value is 2.00, which means that after graph reduction the conflict graph is denser than the initial conflict graph.

The graph reduction stage mainly removes vertices with degree less than three and contained vertices and remove the incident edges, while the *RG* construction stage aims at removing conflict edges from dense structures. Comparing the column $\#E^R$ with the column #E in "after graph reduction," it can be found that the number of edges of #E is reduced to $\#E^R$ by 0.54/0.50=8%. This implies that removing edges at the *RG* construction stage is effective for dense graph structures. Furthermore, comparing the data in the two columns "#ANC" indicates that, the average number of patterns in every connected component in the relaxed conflict graph is only one-third of the initial graph. Thus, it is small enough for solving 0-1 linear programs (2) and (4) on every connected component.

The most time-consuming computation in our two-stage decomposition method is solving the binary linear program (2) or program (4) by the cutting-plane approach in the software package GUROBI. To speed up the computation, we set the parameter *gap* in GUROBI (Optimization 2014) as a larger value for larger connected components. The parameter *gap* is used to control the

Table 2. Comparison of Decomposition Results with Smaller Gap and Larger Gap for the ILP of the ESTMA Based Decomposition, $\min_{cs} = 160$nm

| | CC number and size | | | #CP | Smaller gap | | | Larger gap | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Benchs | #CC | #≥60 | #≥100 | | #VSB | #S | CPU(s) | #VSB | #S | CPU(s) |
| C432 | 25 | 1 | 0 | 275 | 85 | 7 | 9.77 | 86 | 6 | 2.24 |
| C499 | 63 | 4 | 1 | 1042 | 279 | 33 | 109.06 | 281 | 31 | 7.89 |
| **C880** | 68 | 4 | 0 | 525 | 110 | 89 | 26.08 | 110 | 89 | 7.44 |
| C1355 | 76 | 3 | 0 | 549 | 146 | 62 | 21.30 | 146 | 62 | 7.98 |
| C1908 | 116 | 3 | 0 | 561 | 152 | 86 | 25.42 | 152 | 86 | 11.28 |
| C2670 | 214 | 2 | 2 | 1549 | 423 | 271 | 200.10 | 430 | 264 | 21.09 |
| C3540 | 347 | 5 | 0 | 2080 | 431 | 362 | 55.51 | 431 | 362 | 30.27 |
| C5315 | 399 | 8 | 1 | 3363 | 858 | 344 | 176.45 | 862 | 341 | 37.32 |
| C6288 | 338 | 5 | 2 | 2673 | 738 | 236 | 273.80 | 740 | 234 | 32.98 |
| C7552 | 534 | 8 | 3 | 3840 | 1007 | 524 | 365.33 | 1019 | 515 | 51.85 |
| S1488 | 116 | 6 | 2 | 1488 | 416 | 182 | 256.75 | 419 | 180 | 24.60 |
| S38417 | 1723 | 26 | 5 | 17218 | 4048 | 1331 | 966.22 | 4069 | 1314 | 177.65 |
| S35932 | 3243 | 297 | 74 | 52982 | N/A | N/A | >3600 | 13258 | 2688 | 718.53 |
| S38584 | 4463 | 54 | 3 | 46445 | 10021 | 3270 | 1321.14 | 10063 | 3233 | 269.90 |
| S15850 | 4012 | 164 | 42 | 50086 | N/A | N/A | >3600 | 13011 | 3689 | 487.90 |
| Avg. | 1049 | 39 | 9 | 12312 | 1439 | 523 | 292.84 | 1447 | 517 | 52.50 |
| Ratio | | | | | 0.99 | 1.01 | 5.58 | 1.00 | 1.00 | 1.00 |

termination criterion in the cutting plane approach, which is

$$\frac{|f(x^c) - LB|}{|f(x^c)|} \leq gap,$$

where $f(x^c)$ is the currently minimal value, and $LB$ is the lower bound obtained by linear program relaxation of the binary linear program (2) or program (4).

For every relaxed conflict graph, we count the number of connected components with vertex number between 60 and 100, and the number of connected components with vertex number not less than 100, respectively, and put them in columns "#≥60" and "#≥100" in Table 2, respectively. We test our hybrid decomposition method with different *gap*s in the cutting plane approach for the ILP of the ESTMA problem: (i) smaller gap, $gap = 10^{-4}$ for all connected components, and (ii) larger gap, $gap = 0.4$ for connected components with vertex number smaller than 60, $gap = 0.5$ for connected components with vertex number between 60 and 100, and $gap = 0.6$ for connected components with vertex number not less than 100. The test results of the ESTMA-based decomposition method are listed in Table 2.

In Table 2, the data in the column "#CP" are the numbers of conflict patterns in the conflict graphs. These patterns should be prior assigned to TPL masks, since conflicts at these patterns can hardly be eliminated by stitch insertions. In columns "#VSB," "#S," and "CPU(s)," we list the total VSB numbers, the total stitch numbers, and the runtimes by our ESTMA-based decomposition method, respectively. On the one hand, the average runtime of the ESTMA-based method with a smaller gap is 5.61 times more than with that with lager gap. Moreover, the ESTMA-based method with a smaller gap is very slow for the connected components with many vertices, especially for benchmarks S35932 and S15850, which cannot be solved in 1 hour.

On the other hand, it can be seen that our ESTMA-based decomposition method with a larger gap produces slightly more VSB numbers than that with a smaller gap for almost all benchmarks. More

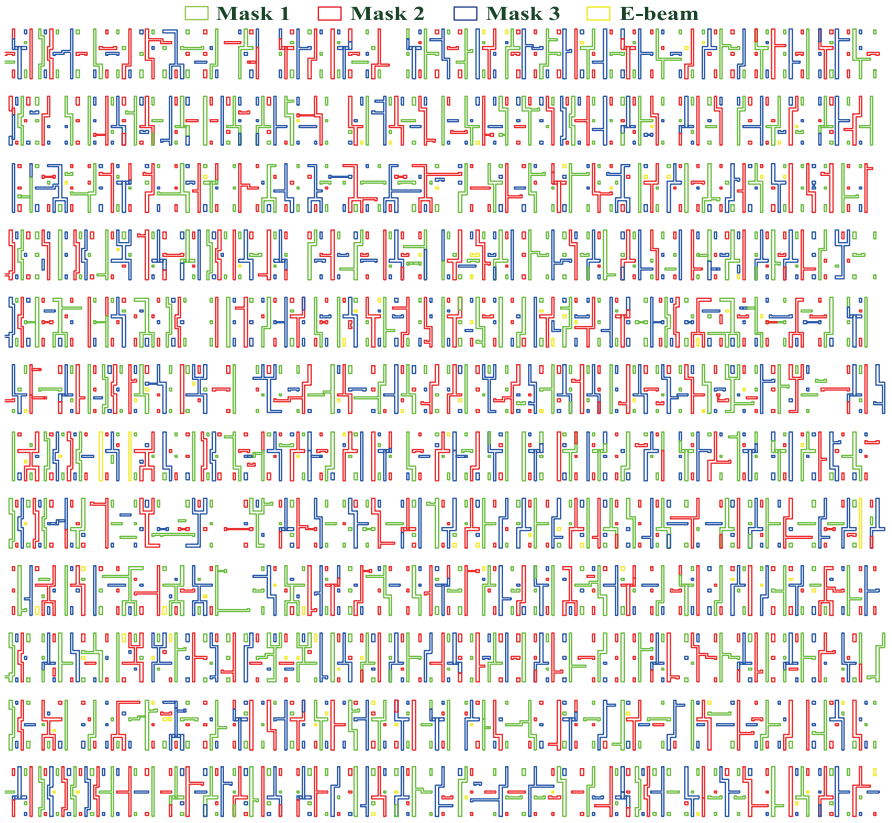


Fig. 7. Decomposed layout for benchmark C880 with $\min_{cs}$=160nm.

precisely, the ESTMA-based method with a larger gap achieves nearly the same good solutions as that with a smaller gap for the benchmarks. Actually, this is due to the cutting plane method and our backtrack coloring algorithm. First, although the gap is not small enough, the cutting plane method may still obtain an sub-optimal solution of the binary programming problem. Second, the backtrack coloring algorithm can reduce the number of VSBs by local swapping.

Figure 7 presents our decomposition result for benchmark **C880** with $\min_{cs}$ = 160nm, which is obtained by the ESTMA-based method with larger gap. In the figure, the green, red, and blue colors denote Mask 1, Mask 2, and Mask 3, respectively, and yellow denotes E-beam.

### 6.2 Comparisons

In this subsection, we compare four hybrid e-beam and TPL decomposers: "TCAD'16" is the decomposer of Yang et al. (2016), "Extended TOC'17" is the decomposer extended from Li et al. (2017), and "ESTMA" and "MDS$R_4$MA" are the two decomposition methods proposed in this article based on problems ESTMA and MDS$R_4$MA, respectively.

Yang et al. first focused on hybrid e-beam and multiple patterning lithography for general layout decomposition and considered two different objectives to evaluate the throughput of EBL: minimum total VSB number and minimum total area for e-beam shot. For the e-beam direct writing strategy, the total area of e-beam shot may not be an essential factor (Yu et al. 2013; Mak and Chu 2014), and the writing time is the main factor for evaluating the throughput of EBL (Tian et al.

Table 3. Comparison Results of the Four HETLD Decomposers, $\min_{cs} = 160$nm

| | TCAD'16 | | | Extended TOC'17 | | | ESTMA | | | MDS$R_4$MA | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Benchs | #VSB | #S | CPU(s) | #VSB | #S | CPU(s) | #VSB | #S | CPU(s) | #VSB | #S | CPU(s) |
| C432 | 108 | 17 | 5 | 162 | 9 | 0.89 | 86 | 6 | 2.24 | 81 | 11 | 3.01 |
| C499 | 333 | 46 | 17.19 | 634 | 25 | 4.99 | 281 | 31 | 7.89 | 274 | 42 | 160.29 |
| C880 | N/A | N/A | N/A | 297 | 73 | 3.11 | 110 | 89 | 7.44 | 118 | 94 | 13.48 |
| C1355 | 207 | 99 | 10.74 | 315 | 86 | 3.56 | 146 | 62 | 7.98 | 145 | 78 | 20.14 |
| C1908 | N/A | N/A | N/A | 383 | 56 | 2.42 | 152 | 86 | 11.28 | 156 | 98 | 172.47 |
| C2670 | N/A | N/A | N/A | 876 | 172 | 12.14 | 430 | 264 | 21.09 | 411 | 294 | 90.96 |
| C3540 | 574 | 380 | 45.21 | 768 | 368 | 4.89 | 431 | 362 | 30.27 | 429 | 449 | 166.32 |
| C5315 | 1013 | 415 | 69.83 | 1589 | 262 | 11.68 | 921 | 341 | 37.32 | 909 | 386 | 176.45 |
| C6288 | N/A | N/A | N/A | 1299 | 271 | 9.42 | 740 | 234 | 32.98 | 722 | 264 | 119.27 |
| C7552 | 1321 | 640 | 94.06 | 2439 | 420 | 17.52 | 1119 | 515 | 51.85 | 1110 | 604 | 266.03 |
| S1488 | N/A | N/A | N/A | 785 | 170 | 27.49 | 419 | 180 | 24.6 | 411 | 196 | 115.19 |
| S38417 | 4634 | 2403 | 315.44 | 8395 | 1386 | 62.84 | 4269 | 1314 | 177.65 | 4235 | 1457 | 298.39 |
| S35932 | 13963 | 7184 | 1117.9 | 26998 | 3516 | 1019.8 | 13258 | 2688 | 718.53 | 12235 | 3258 | 2768.38 |
| S38584 | 10957 | 6132 | 773.7 | 20459 | 3779 | 90.12 | 10063 | 3233 | 269.9 | 9936 | 3709 | 684.51 |
| S15850 | 14292 | 6983 | 800 | 25841 | 3897 | 547.17 | 13011 | 3689 | 487.9 | 12322 | 4370 | 1140.39 |
| Avg.1 | | | | 6083 | 966 | 121.2 | 3029 | 873 | 125.93 | 2900 | 1021 | 413.02 |
| Avg.2 | 4740 | 2430 | 324.91 | | | | 4359 | 1224 | 179.15 | | | |
| Ratio | 1.09 | 1.99 | 1.81 | 2.01 | 1.11 | 0.96 | 1.00 | 1.00 | 1.00 | 0.96 | 1.17 | 3.28 |

2014b; Du et al. 2012; Ding et al. 2014; Gao et al. 2014). Furthermore, for VSB e-beam shot, the writing time mainly depends on the number of VSBs (rectangles). Hence we mainly compare our method with Yang et al. (2016) on this aspect. Yang et al. (2016) only lists results of 10 benchmarks. We cite them directly for comparisons, since the code of their method is not available to us.

In Table 3, the results in the column "Extended TOC'17" are adapted from Li et al. (2017). Since the objectives in Li et al. (2017) are conflict number and stitch number, we directly use e-beam shot to eliminate conflicts in the decomposition results by Li et al. (2017). For fair comparison, we use VSB as few as possible to eliminate conflicts. The data in columns "ESTMA" and "MDS$R_4$MA" are the results of the two methods with a larger gap.

The results of the four decomposers are reported in Table 3. Data in the columns "#VSB" and "#S" are the total VSB numbers and the total stitch numbers by the decomposers on the tested benchmarks, respectively. Data in the columns "CPU(s)" are the runtimes by the respective decomposers. In row "Avg.1," we list the average results on all the benchmarks. Since Ref. Yang et al. (2016) did not report the test results on the benchmarks C880, C1908, C2670, C6288, and S1488, we do not list the average results of "TCAD'16" on all the benchmarks. However, we list in row "Avg.2" the average results of "TCAD'16" on all the benchmarks except the five benchmarks. For fair comparison, we also list in row "Avg.2" the average test results of "ESTMA" on all the benchmarks except the five benchmarks. In the last row "Ratio" of Table 3, we list the ratios of the average results of "TCAD'16," "Extended TOC'17," and "MDS$R_4$MA" based on the results of "ESTMA." It must be remarked that the data in the last row "Ratio" of "TCAD'16" are calculated based on the data in row "Avg.2."

From Table 3, it can be seen that the average VSB number by Yang et al. (2016) is 9% more than that by our ESTMA-based method, and the average stitch number is twice more than that by our ESTMA-based method. Moreover, it can be seen that the runtime of the method in Yang et al. (2016) is 1.81 times more than that of our ESTMA-based method. Furthermore, it must be noted

that their decomposer was run on a workstation with 3GHz CPU and 4GB memory, which is better than ours. This together with the comparison results demonstrates that our ESTMA-based method works better than the decomposer in Yang et al. (2016) on the test benchmarks.

Comparing the data of columns "Extended TOC'17" and "ESTMA" in the row "Ratio," it can be seen that the average VSB number of "Extended TOC'17" is twice more than that of the ESTMA-based method. This is due to the fact that the method in TOC'17 focuses on the minimum conflict number instead of the VSB number, while a pattern may be printed by more than one VSB shot. Furthermore, the average stitch number is 11% more than that by the ESTMA-based method. Hence, extending the TPL layout decomposition method in Li et al. (2017) directly to solve the HETLD problem is not a good choice.

At last, we compare the data of columns "ESTMA" and "MDS$R_4$MA" in the row "Ratio." As expected by theory, the average number of VSB in "MDS$R_4$MA" is 4% less than that in "ESTMA," and the average number of stitches in "MDS$R_4$MA" is 17% more than that in "ESTMA." This demonstrates that the MDS$R_4$MA-based method assigns more pairs of a *CP* and its *CAP*s to $R_4$ simultaneously at the first decomposition stage, and then there are more patterns in $R_4$ are assigned to TPL masks by inserting stitches. Finally, the average runtime of MDS$R_4$MA is 3.28× of ESTMA. This is due to the fact that the ILP formulation in the MDS$R_4$MA-based method has more variables and constraints than the ILP formulation in the ESTMA-based method.

## 7 CONCLUSIONS

Hybrid e-beam and triple patterning lithography is a new technology for manufacture of VLSI circuit, which combines the advantages of e-beam and TPL. Layout decomposition is a core problem in the hybrid lithography, which is NP-hard on the general layout. In this article, we propose a two-stage layout decomposition flow for the HETLD problem that achieves decomposition by two steps. First, we consider the ESTMA problem, and then the problem is relaxed by deleting some conflict edges, which is used to quickly obtain a solution with some conflicts. Second, the infeasible solution with conflicts is legalized to a feasible one of the HETLD problem by stitch insertion and e-beam shot. To speed up decomposition, we reduce the problem size by removing some vertices and some edges before decomposition.

Furthermore, to obtain a better solution with less VSB number, we propose the extended minimum weight dominating set for $R_4$ mask assignment (MDS$R_4$MA) problem. By solving the MDS$R_4$MA problem in the first decomposition stage, we can obtain a solution with the patterns in $R_4$ more likely being assigned to TPL masks by stitch insertion. However, the ILP formulation of the MDS$R_4$MA problem has many more variables and constraints than the ILP formulation of the ESTMA problem.

In the decomposition process, our objective is maximizing e-beam throughput (minimizing VSB number) and minimizing stitch number. Experimental results show the effectiveness of the ESTMA and the MDS$R_4$MA-based decomposition methods, comparing with the state-of-the-art decomposer.

## ACKNOWLEDGMENTS

The authors thank the editor and anonymous reviewers, whose suggestions and comments helped improving the quality of the manuscript greatly.

## REFERENCES

Yan Borodovsky. 2009. Lithography 2009 overview of opportunities. In *Semicon West.*

Yao-Wen Chang, Ru-Gun Liu, and Shao-Yun Fang. 2015. EUV and E-beam manufacturability: Challenges and solutions. In *Proceedings of the 52nd Annual Design Automation Conference (DAC'15).* ACM, 198.

J. Geoffrey Chase and Bram W. Smith. 2001. Overview of modern lithography techniques and a MEMS-based approach to high throughput rate electron beam lithography. *J. Intell. Mater. Syst. Struct.* 12, 12 (2001), 807–817.

Hsi-An Chien, Ye-Hong Chen, Szu-Yuan Han, Hsiu-Yu Lai, and Ting-Chi Wang. 2015. On refining row-based detailed placement for triple patterning lithography. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst.* 34, 5 (2015), 778–793.

Yixiao Ding, Chris Chu, and Wai-Kei Mak. 2014. Throughput optimization for SADP and e-beam based manufacturing of 1D layout. In *Proceedings of the 51st Annual Design Automation Conference (DAC'14)*. ACM, 1–6.

Yuelin Du, Hongbo Zhang, Martin D. F. Wong, and Kai-Yuan Chao. 2012. Hybrid lithography optimization with e-beam and immersion processes for 16nm 1D gridded design. In *Proceedings of 17th Asia and South Pacific Design Automation Conference (ASP-DAC'12)*. IEEE, 707–712.

Shao-Yun Fang, Yao-Wen Chang, and Wei-Yu Chen. 2014. A novel layout decomposition algorithm for triple patterning lithography. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst.* 33, 3 (2014), 397–408.

Aki Fujimura. 2010. Design for e-beam: Design insights for direct-write maskless lithography. In *Proceedings of SPIE*. 7823 (2010), 137–140.

Jhih-Rong Gao, Bei Yu, and David Z. Pan. 2014. Self-aligned double patterning layout decomposition with complementary e-beam lithography. In *Proceedings of 19th Asia and South Pacific Design Automation Conference (ASP-DAC'14)*. IEEE, 143–148.

Michael R. Garey, David S. Johnson, and Larry Stockmeyer. 1976. Some simplified NP-complete graph problems. *Theor. Comput. Sci.* 1, 3 (1976), 237–267.

Ryoichi Inanami, Shunko Magoshi, Shouhei Kousai, Atsushi Ando, Tetsuro Nakasugi, Ichiro Mori, Kazuyoshi Sugihara, and Akira Miura. 2003. Maskless lithography: Estimation of the number of shots for each layer in a logic device with character projection-type low-energy electron-beam direct writing system. In *Proceedings of SPIE*. 5037 (2003), 1043–1050.

Jian Kuang and Evangeline F. Y. Young. 2013. An efficient layout decomposition approach for triple patterning lithography. In *Proceedings of the 50th Annual Design Automation Conference (DAC'13)*. ACM, 69.

Xingquan Li, Ziran Zhu, and Wenxing Zhu. 2017. Discrete relaxation method for triple patterning lithography layout decomposition. *IEEE Trans. Comput.* 66, 2 (2017), 285–298.

Wai-Kei Mak and Chris Chu. 2014. E-beam lithography character and stencil co-optimization. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst.* 33, 5 (2014), 741–751.

Takashi Maruyama, Yasuhide Machida, Shinji Sugatani, Hiroshi Takita, Hiromi Hoshino, Toshio Hino, Masaru Ito, Akio Yamada, Tetsuya Iizuka, and Satoshi Komatsu. 2012. CP element based design for 14nm node EBDW high volume manufacturing. In *Proceedings of SPIE*. 8323 (2012), 8323–14.

Gurobi Optimization. 2014. Inc., Gurobi Optimizer Reference Manual, 2014. *Retrieved March 27, 2014 from www.gurobi.com.*

S. E. Steen, S. J. McNab, L. Sekaric, I. Babich, J. Patel, J. Bucchignano, M. Rooks, D. M. Fried, A. W. Topol, J. R. Brancaccio, and others. 2005. Looking into the crystal ball: Future device learning using hybrid e-beam and optical lithography (keynote paper. In *Proceedings of the International Society for Optics and Photonics on Microlithography 2005*. 26–34.

Haitong Tian, Yuelin Du, Hongbo Zhang, Zigang Xiao, and Martin D. F. Wong. 2014a. Triple patterning aware detailed placement with constrained pattern assignment. In *Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD'14)*. IEEE, 116–123.

Haitong Tian, Hongbo Zhang, Zigang Xiao, and Martin D. F. Wong. 2014b. Hybrid lithography for triple patterning decomposition and e-beam lithography. In *Proceedings of SPIE*. 9052 (2014), 90520P.

Linda Wilson. 2013. International technology roadmap for semiconductors (ITRS). In *Proceedings of the Semiconductor Industry Association* (2013).

Yunfeng Yang, Wai-Shing Luk, David Pan, Hai Zhou, Changhao Yan, Dian Zhou, and Xuan Zeng. 2016. Layout decomposition co-optimization for hybrid e-beam and multiple patterning lithography. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst.* 35, 9 (2016), 1532–1545.

Bei Yu, Xiaoqing Xu, Jhih-Rong Gao, Yibo Lin, Zhuo Li, Charles J. Alpert, and David Z. Pan. 2015a. Methodology for standard cell compliance and detailed placement for triple patterning lithography. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst.* 34, 5 (2015), 726–739.

Bei Yu, Kun Yuan, Duo Ding, and David Z. Pan. 2015b. Layout decomposition for triple patterning lithography. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst.* 34, 3 (2015), 433–446.

Bei Yu, Kun Yuan, Jhih-Rong Gao, and David Z. Pan. 2013. E-BLOW: E-beam lithography overlapping aware stencil planning for MCC system. In *Proceedings of the 50th Annual Design Automation Conference (DAC'13)*. ACM, 70.

Kun Yuan, Bei Yu, and David Z. Pan. 2012. E-beam lithography stencil planning and optimization with overlapped characters. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst.* 31, 2 (2012), 167–179.

Received September 2016; revised March 2017; accepted April 2017