

Title:	Layout Decomposition for Triple Patterning
Name:	Bei Yu, David Z. Pan
Affil./Addr.	Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX, USA
Keywords:	Layout decomposition; Lithography; Graph coloring; Mathematical programming
SumOriWork:	2011; Yu, Yuan, Zhang, Ding, Pan

# Layout Decomposition for Triple Patterning

BEI YU, DAVID Z. PAN

Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX, USA

## Years and Authors of Summarized Original Work

2011; Yu, Yuan, Zhang, Ding, Pan

## Keywords

Layout decomposition; Lithography; Graph coloring; Mathematical programming

## Problem Definition

Layout decomposition is a key stage in triple patterning lithography manufacturing process, where the original designed layout is divided into three masks. There will be three exposure/etching steps, through which the circuit layout can be produced. When the distance between two input features is less than certain minimum distance  $min_s$ , they need to be assigned to different masks (colors) to avoid coloring conflict. Sometimes coloring conflict can be resolved by splitting a pattern into two different masks. However this introduces stitches, which lead to yield loss because of overlay error. Therefore, two of the main objectives in layout decomposition are conflict minimization and stitch minimization. An example of triple patterning layout decomposition is shown in Fig. 1, where all features are divided into three masks without any conflict and one stitch is introduced.

Given an input layout, a *conflict graph* is constructed to transfer initial geometrical relationship into an undirected graph with a set of vertices  $V$  and two sets of edges, which are the *conflict edges* ( $CE$ ) and *stitch edges* ( $SE$ ), respectively.  $V$  has one or more vertices for each polygonal shape and each vertex is associated with a polygonal shape. An edge is in  $CE$  iff the two corresponding vertices are within minimum coloring distance  $min_s$ . An edge is in  $SE$  iff there is a stitch candidate between the two vertices which are associated with the same polygonal shape.

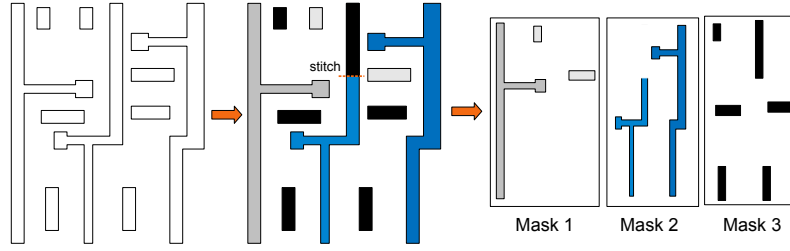


Fig. 1. Layout decomposition for triple patterning lithography (TPL).

## Problem 1 (Layout Decomposition for Triple Patterning)

INPUT: The decomposition graph where each vertex represents one polygonal shape, and all possible conflicts and stitches are in the conflict edge set  $CE$  and the stitch edge set  $SE$ , respectively.

OUTPUT: A three color assignment to the conflict graph, such that the weighted cost of conflicts and stitches are minimized. The additional constraints may include color balancing, overlay control, and color preference.

## Key Results

Given input layout, the conflict graph is constructed. Based on the conflict graph, the layout decomposition for triple patterning can be formulated as an integer linear programming (ILP) formulation [5]. As shown in (1), the objective function in the ILP formulation is to minimize the weighted cost function of conflict and stitch numbers simultaneously:

$$\min \sum_{e_{ij} \in CE} c_{ij} + \alpha \sum_{e_{ij} \in SE} s_{ij} \quad (1)$$

where  $\alpha$  is a parameter for assigning relative cost of stitch versus conflict. Typically  $\alpha$  is much smaller than 1, for example 0.1 as resolving conflict is the most important objective during layout decomposition. Although the ILP formulation can solve the above layout decomposition problem optimally, it is not scalable to deal with large layouts in modern VLSI designs as the ILP problem is NP-complete.

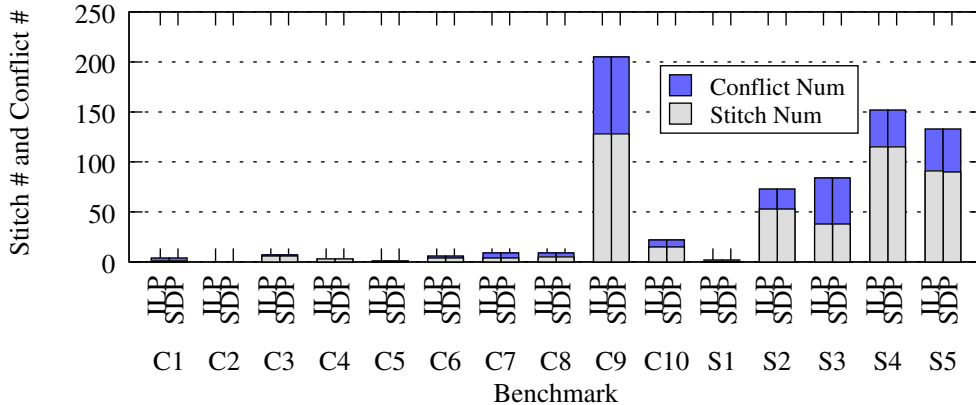
In [5], a semidefinite programming (SDP) based algorithm was proposed to achieve good runtime and solution quality. Instead of using a two binary variables to represent three masks, three unit vectors  $(1, 0)$ ,  $(-\frac{1}{2}, \frac{\sqrt{3}}{2})$  and  $(-\frac{1}{2}, -\frac{\sqrt{3}}{2})$  are proposed to represent them. Note that the angle between any two vectors of the same color is 0, while the angle between any two vectors with different colors is  $2\pi/3$ . The inner product of two  $m$ -dimension vectors  $\mathbf{v}_i$  and  $\mathbf{v}_j$  is defined as  $\mathbf{v}_i \cdot \mathbf{v}_j = \sum_k v_{ik}v_{jk}$ . Then for any two vectors  $\mathbf{v}_i, \mathbf{v}_j \in \{(1, 0), (-\frac{1}{2}, \frac{\sqrt{3}}{2}), (-\frac{1}{2}, -\frac{\sqrt{3}}{2})\}$ , the following property holds:

$$\mathbf{v}_i \cdot \mathbf{v}_j = \begin{cases} 1, & \mathbf{v}_i = \mathbf{v}_j \\ -\frac{1}{2}, & \mathbf{v}_i \neq \mathbf{v}_j \end{cases}$$

Based on the vector representation, the layout decomposition for triple patterning problem can be written as the following vector programming.

$$\min \sum_{e_{ij} \in CE} \frac{2}{3}(\mathbf{v}_i \cdot \mathbf{v}_j + \frac{1}{2}) + \frac{2\alpha}{3} \sum_{e_{ij} \in SE} (1 - \mathbf{v}_i \cdot \mathbf{v}_j) \quad (2)$$

$$\text{s.t. } \mathbf{v}_i \in \{(1, 0), (-\frac{1}{2}, \frac{\sqrt{3}}{2}), (-\frac{1}{2}, -\frac{\sqrt{3}}{2})\} \quad (2a)$$



**Fig. 2.** For ISCAS benchmark suite, the results of ILP and SDP based methods are very comparable.

It shall be noted that  $\mathbf{v}_i$  here is discrete, which is very expensive to solve. Then the discrete vector program is relaxed to the corresponding continuous formulation, which can be solved as a standard semidefinite programming (SDP), as shown below.

$$\min \sum_{e_{ij} \in CE} \frac{2}{3} (\mathbf{y}_i \cdot \mathbf{y}_j + \frac{1}{2}) + \frac{2\alpha}{3} \sum_{e_{ij} \in SE} (1 - \mathbf{y}_i \cdot \mathbf{y}_j) \quad (3)$$

$$\text{s.t. } \mathbf{y}_i \cdot \mathbf{y}_i = 1, \quad \forall i \in V \quad (3a)$$

$$-\frac{1}{2} \leq \mathbf{y}_i \cdot \mathbf{y}_j, \quad \forall e_{ij} \in CE \quad (3b)$$

$$Y \succeq 0 \quad (3c)$$

The resulting matrix  $Y$ , where  $y_{ij} = \mathbf{y}_i \cdot \mathbf{y}_j$ , essentially provides the relative coloring guidance between two layout features (nodes in the conflict graph). It will be used to guide the final color assignment. If  $y_{ij}$  is close to 1, nodes  $i$  and  $j$  should be in the same mask; if  $y_{ij}$  is close to -0.5, nodes  $i$  and  $j$  tend to be in different masks. The results show that with reasonable threshold such as  $0.9 < y_{ij} \leq 1$  for the same mask, and  $-0.5 \leq y_{ij} < -0.4$  for different masks, more than 80% of nodes/polygons are decided by the global SDP. For the rest values, heuristic mapping algorithms will be performed to assign all vertices to their final colors.

A set of graph simplification techniques have been proposed to achieve speed-up [5][1][2][8]. For example, one technique is called *iterative vertex removal*, where all vertices with degree less than or equal to two are detected and removed temporarily from the conflict graph. After each vertex removal, the degrees of other vertices would be updated. This removing process will continue until all the vertices have degree three or more. All the vertices that are temporarily removed are stored in a stack  $S$ . After the color assignment of the remained conflict graph is solved, the removed vertices in  $S$  are added back for coloring assignment. For row-based structure layout, specific graph based algorithms are proposed to provide fast layout decomposition solutions [3][7].

Triple patterning layout decomposition has been actively studied in the last few years, with many interesting results reported. In [5], the performances between ILP and SDP based methods were compared. As shown in Fig. 2, SDP based method can achieve the same optimal solutions as obtained by ILP for 14 out of 15 test cases. However, the runtime of ILP-based algorithm is prohibitive when the problem size is big and the layout is dense. Graph simplification techniques are very effective to speed up the layout decomposition process as that can effectively reduce the ILP and SDP problem size. The coloring density balance was integrated into the SDP formulation in

[6]. In [4], the SDP framework was further extended to handle quadruple patterning or more general multiple patterning lithography with new vector definition and linear runtime heuristic algorithms.

## URLs to Code and Data Sets

Programs and benchmark suites can be found through <http://www.cerc.utexas.edu/utda/download/MPLD/>.

## Recommended Reading

1. Fang SY, Chen WY, Chang YW (2012) A novel layout decomposition algorithm for triple patterning lithography. In: IEEE/ACM Design Automation Conference (DAC), pp 1185–1190
2. Kuang J, Young EF (2013) An efficient layout decomposition approach for triple patterning lithography. In: IEEE/ACM Design Automation Conference (DAC), pp 69:1–69:6
3. Tian H, Zhang H, Ma Q, Xiao Z, Wong M (2012) A polynomial time triple patterning algorithm for cell based row-structure layout. In: IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp 57–64
4. Yu B, Pan DZ (2014) Layout decomposition for quadruple patterning lithography and beyond. In: IEEE/ACM Design Automation Conference (DAC)
5. Yu B, Yuan K, Zhang B, Ding D, Pan DZ (2011) Layout decomposition for triple patterning lithography. In: IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp 1–8
6. Yu B, Lin YH, Luk-Pat G, Ding D, Lucas K, Pan DZ (2013) A high-performance triple patterning layout decomposer with balanced density. In: IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp 163–169
7. Yu B, Xu X, Gao JR, Pan DZ (2013) Methodology for standard cell compliance and detailed placement for triple patterning lithography. In: IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp 349–356
8. Zhang Y, Luk WS, Zhou H, Yan C, Zeng X (2013) Layout decomposition with pairwise coloring for multiple patterning lithography. In: IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp 170–177