Efficient DSA-DP Hybrid Lithography Conflict Detection and Guiding Template Assignment

Jiaojiao Ou[†], Brian Cline[‡], Greg Yeric[‡] and David Z. Pan[†]
[†]ECE Dept., University of Texas at Austin, Austin, TX USA
[‡] ARM Inc.
Email[†]: {jiaojiao.ou}@utexas.edu, {dpan}@ece.utexas.edu
Email[‡]: {brian.cline,greg.yeric}@arm.com

ABSTRACT

In recent years, directed self-assembly (DSA) has demonstrated tremendous potential to reduce cost for multiple patterning with fewer masks, especially for via patterning. DSA is considered as one of the next generation lithography candidates or complementary lithography techniques to extend 193i lithography further for the sub-7 nm nodes. In this work, we focus on the simultaneous DSA guiding template assignment and decomposition with DSA and double patterning (DSA-DP) hybrid lithography for 7nm technology node. We first analyze the placement error of DSA patterns with different shapes and sizes. We then propose a graph-based approach to reduce the problem size and solve the problem more efficiently without affecting the optimality of the results. The experimental results demonstrate that we can achieve a 50% reduction in both the number of variables and constraints compared to previous work, which leads to a 50X speed up in runtime.

1. INTRODUCTION

With the continuing scaling of technology nodes, directed self-assembly (DSA) is considered to be one of the next generation lithography candidates and complementary lithography techniques for extending 193 nm immersion lithography (193i) further for sub-7 nm nodes, especially for via layers.[1–4] Additionally, because of the pitch-multiplication and hole-shrink ability of DSA, it can also be applied to the cut/block/trim mask for layers used in self-aligned multiple patterning lithography.[5–7] Because of the chemistry involved in DSA lithography and the inherent properties of the block-copolymers used, DSA patterns are typically limited to certain topologies and sizes. Additionally, DSA-aware design techniques, such as DSA-aware redundant via insertion[8–10] and DSA-aware detailed routing[11–13] are needed to improve the yield and printability of the DSA layers. Furthermore, multiple patterning on the DSA guiding templates is employed in ultra scaled technology nodes.[14] In order to take full advantage of the properties of DSA and maximize the achievable mask count reduction (which can result in significant manufacturing cost savings since mask cost is on the order of millions of dollars in sub-7nm nodes), DSA decomposition that simultaneously groups vias and assigns colors to guiding templates is needed. Badr et al. approached the problem with an integer linear programming (ILP) and a heuristic method with maximal cardinality matching.[15] However, the proposed ILP formulation contains too many variables and it does not consider the properties of the conflict graph, which can greatly affect the runtime for large designs. Kuang et al. proposed a heuristic method to solve the simultaneous DSA pattern decomposition with multiple patterning, but the optimality of the result cannot be guaranteed.[16]

In this work, we will focus on DSA guiding template assignment and decomposition with DSA-DP hybrid lithography for 7nm design. we first analyze the cost of DSA patterns with different shapes and sizes. As the cost of DSA patterns can affect the final mask manufacturability and yield, DSA patterns with low cost are preferred during the decomposition stage. We then propose a graph-based approach to reduce the problem size and solve the DSA guiding template assignment with double patterning (DSA-DP) problem more efficiently without affecting the optimality of the final result. The experimental results demonstrate that we achieve a 50% reduction on both the number of variables and constraints compared to previous work of Badr et al.,[15] which enables up to a 50X speed up in runtime.

Our major contributions are listed as follows.

1. We measure and analyze the placement error for different DSA patterns;

Design-Process-Technology Co-optimization for Manufacturability XI, edited by Luigi Capodieci, Jason P. Cain, Proc. of SPIE Vol. 10148, 101480C · © 2017 SPIE
CCC code: 0277-786X/17/$18 · doi: 10.1117/12.2258156

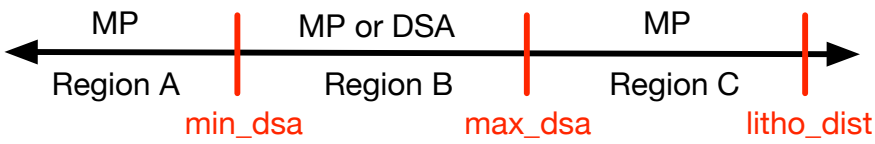


Figure 1. DSA design rule

2. We propose a more efficient mathematical formulation with optimal solution compared to previous work;

The rest of the paper is organized as follows. In Section 2, we discuss the DSA design rules and analyze DSA pattern placement error. In Section 3, we explain the DSA double patterning guiding template assignment and conflict detection. We discuss the conflict graph construction and simplification and we also discuss the mathematical formulation to solve the problem. The experimental results are presented in Section 4 and we conclude the paper in Section 5.

2. PRELIMINARIES

2.1 DSA Design Rules

Based on the design rules from Badr et al.,[15] the most stable distance between two holes in the same guiding template is the natural pitch, L_o, of the block copolymer materials. In order to obtain a larger distance between two holes, more energy is required on the guiding template, leading to more complicated masks. It may also increase the placement error between the target via pattern and final via pattern. Therefore, a threshold region is defined to limit the distance between adjacent holes in the same DSA guiding template. As shown in Figure 1, min_dsa and max_dsa are used to indicate the minimum and maximum distance between adjacent holes. min_dsa is usually equal to L_o. For adjacent vias, they can be grouped in the same guiding template if their distance is within $[min\_dsa, max\_dsa]$. Otherwise, they have to be decomposed into different masks if their distance is smaller than $litho\_dist$, which is the minimum distance between two patterns on the same mask. In addition, some DSA pattern topologies are forbidden, such as forks and circular shaped patterns, as they are difficult to synthesize and the resulting guiding templates are too complex. So only linear type DSA patterns where all the patterns are located in a nearly straight line are allowed.

2.2 DSA Pattern Placement Error Analysis

In order to determine the maximum number of vias in the same DSA guiding template, we analyze the cost of DSA patterns with different sizes. The cost of DSA patterns is defined as the average placement error between target and simulated vias inside the guiding template by the DSA tool.[17] The placement error of a single hole, pe, and the placement error of the entire DSA pattern, PE, is calculated as follows:

$$pe = |C_{target} - C_{simulated}| \tag{1a}$$

$$PE = \frac{\sum_{i=1}^{n} pe_i}{n} \tag{1b}$$

where n is the number of holes of the DSA pattern, C_{target} indicates the (x,y) coordinates of the center of target via, $C_{simulated}$ indicates the (x,y) coordinates of the center of the simulated via. To calculate the PE of different DSA patterns, we randomly generate some DSA test patterns in which the number of holes in the same guiding template varies from 1 to 5. Figure 2 depicts how pe is calculated for a single hole, and Figure 2(b)-(e) show examples of the test pattern configurations. The test patterns are then put into a DSA simulation flow, which is shown in Figure 3. In the simulation flow, DSA synthesis first generates the ideal shapes for the DSA guiding templates based on the locations of the target vias. Then Optical Proximity Correction (OPC) is performed on the ideal guiding templates with 3 lithography process conditions: nominal, minimum and maximum condition. Finally, the DSA simulation is performed on the guiding templates to generate the final vias.[14,17]

Box plots for the placement error of different DSA patterns grouped by size is shown in Figure 4. As the number of randomly placed vias within a group increases, the resulting placement error also increases. Therefore, in order to improve the yield, we limit the maximum DSA pattern size to 3.

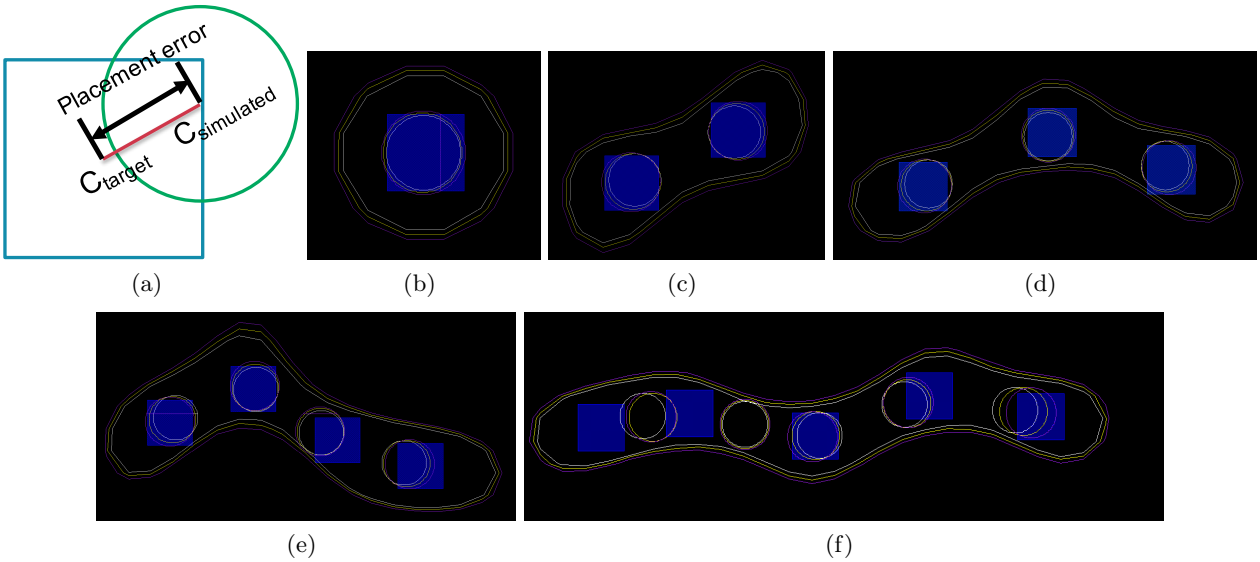


Figure 2. (a) Placement error example. (b-f) Placement error example for DSA GPs with 1 hole to 5 holes.

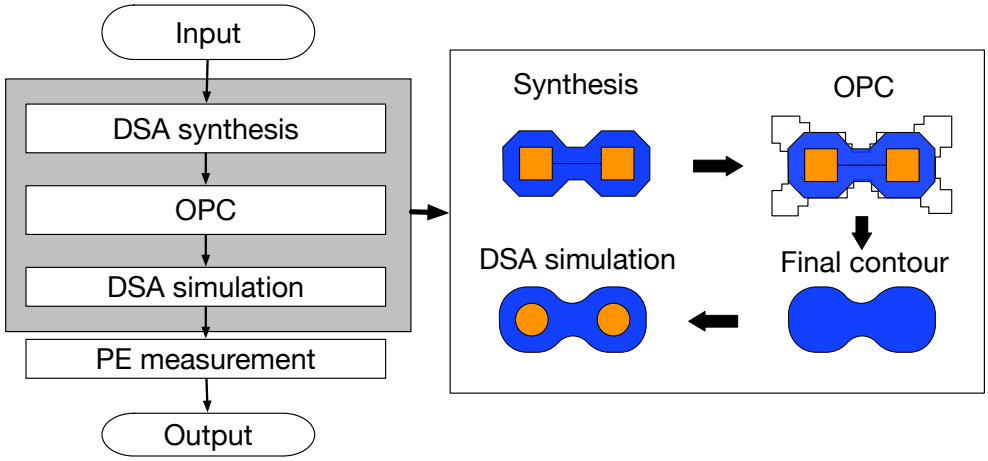


Figure 3. Placement error measurement process

3. DSA GUIDING TEMPLATE ASSIGNMENT AND CONFLICT DETECTION WITH DOUBLE PATTERNING

3.1 Overall Flow

The overall flow of our DSA-DP aware guiding template assignment and conflict detection is shown in Figure 5. First, we will construct a conflict graph followed by a graph simplification for the via layer based on the DSA and lithography design rules. Then, based on whether the graph is a planar graph or not, we formulate corresponding mathematical formulations to solve the DSA guiding template grouping and decomposition. Each step will be explained in detail in the following sections.

3.2 Conflict Graph Construction and Simplification

Once we have the DSA design rules, we can construct the conflict graph for the via layer. A conflict edge is added between adjacent vias if their distance is less than "*litho_dist*", as shown in Figure 6. The conflict edge is a solid line if the distance is within the DSA groupable region. Otherwise, the conflict edge is a dashed line, meaning that the two vias have to be assigned to different masks.

We then perform the graph simplification after the conflict graph construction to make the problem size much smaller. It is noticed that the edges that are not contained in any circles in the conflict graph will not affect

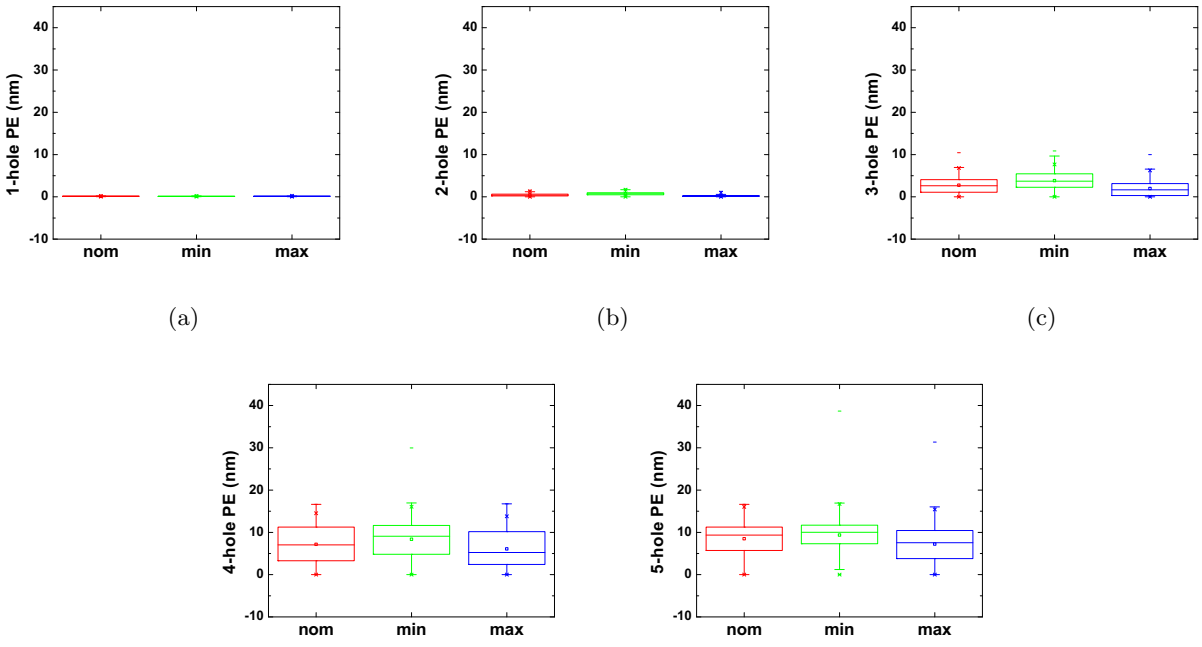


Figure 4. Placement error for different DSA patterns: (a) 1-hole patterns. (b) 2-hole patterns. (c) 3-hole patterns. (d) 4-hole patterns. (5) 5-hole patterns.

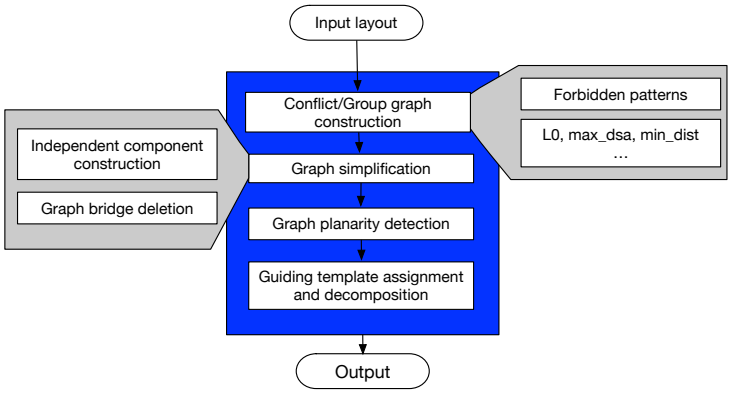
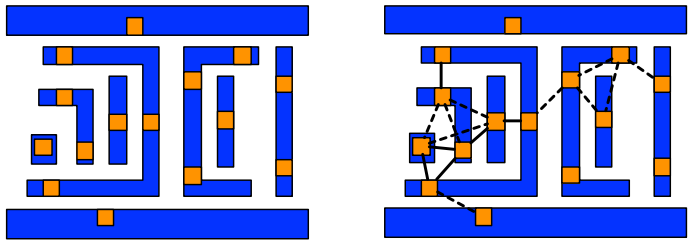


Figure 5. Flow of DSA-DP decomposition

the DSA grouping and decomposition result as we can always find two different colors for vias on the two ends of these edges. These kind of edges are also called bridges in graph theory. Thus by deleting the bridges we can obtain several independent components of the conflict graph. The DSA grouping and decomposition can be performed concurrently on the independent components.

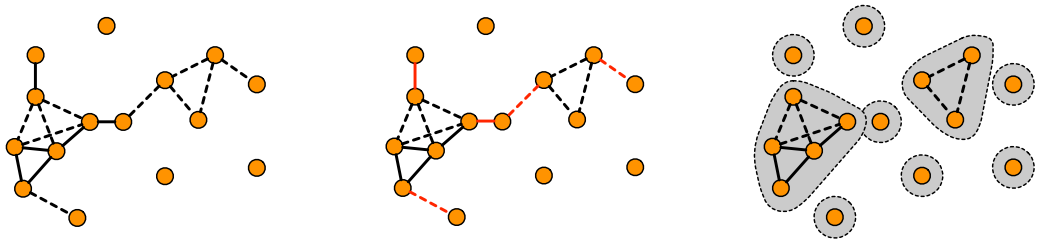
3.3 Constrained Edge Bipartization

Without DSA, for the 7nm technology used in this study, four or more masks are needed to print the densely distributed via layers, leading to an increase of mask cost. With DSA, we can combine critical vias in the same guiding template. In the conflict graph, the grouping of vias is similar to deleting the conflict edges. When adjacent vias are grouped in the same guiding template, the conflict edge between them will be deleted. It is also observed that the final graph after edge deletion should be a bipartite graph in order to be 2-colorable. Together with the edge deletion step and DSA constraints, the DSA grouping and decomposition problem can



(a) (b)

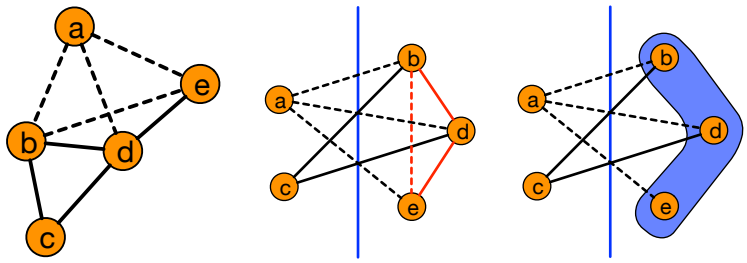
Figure 6. (a) Contact/via distribution. (b) Conflict graph construction.



(a) (b) (c)

Figure 7. Conflict graph simplification. (a) Original conflict graph. (b) Bridge removal (marked in red). (c) Independent components.

be converted to the constrained edge bipartization problem, which is a known NP-complete problem.[18] To solve this problem, we can assign a large weight to the dashed edge which indicates that the vias on the two ends cannot be grouped by DSA. Then we divide the vertices in the conflict graph into two parts so that the cost of edges connecting the vertices on different parts is maximized. Edges that connect vertices on the same side are deleted, and vertices that are connected by the deleted edges with small weight are grouped in the same guiding template. It should be noted that during the edge bipartization process, it is necessary to take into account the DSA pattern constraints: the maximum group size, forbidden patterns and etc. Thus by assigning the vertices on the same side to the same color, we can obtain the final DSA grouping and decomposition results with minimized conflicts. For example, as shown in Figure 8, via a and c are assigned to one mask, via b, d and e are assigned to another mask, and b, d and e are grouped in the same DSA guiding templates.



(a) (b) (c)

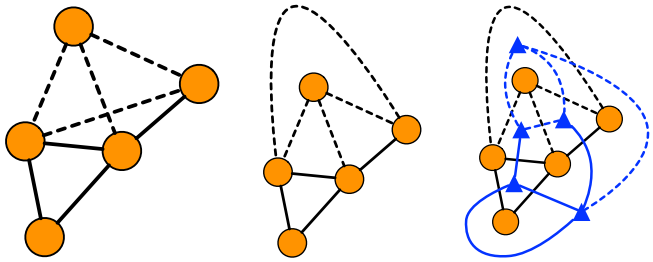
Figure 8. Constrained edge bipartization. (a) Original conflict graph. (b) Edge bipartization. Node a and c are on the same side, other nodes are on the other side. (c) Final result. Node b, d and e are grouped in the same DSA guiding template.

We also observed that the problem can be further simplified when the original conflict graph is a planar graph. A planar graph is a graph that can be embedded in the plane where the edges can not cross each other. The dual graph of the planar graph is a graph that has a vertex for each face of the planar graph. The edge of the dual graph connects two faces of the planar graph when they are separated by an edge, as illustrated in

Table 1. Notations

E	Edge set
V	Vertex set
E_g	Groupable edge set. The two vertices on the ends of an edge in E_g can be grouped
E_c	Conflict edge set. The two vertices on the ends of an edge in E_c cannot be grouped
E_v	Out edges of vertex v
E_f	Edge set of forbidden DSA pattern
e_{uv}	Binary parameters, edge that connects vertices (u, v)
M	Large coefficient
s_u, s_v	Binary parameters. $s_u = s_v$ indicates vertices u and v are on the same side
max_g	Maximum group number

Figure 9. Instead of minimizing the deleted edges of the original graph during the edge bipartization process, we can make the degree of each vertices of the dual graph an even number by deleting edges. Since the number of faces is much smaller than the number of edges of the planar graph, the problem size can be greatly reduced.



(a) (b) (c)

Figure 9. Spacial case: (a)-(b) Original planar graph. (c) Dual graph (blue) of the planar graph.

3.4 Mathematical Formulation

Based on general conflict graph and the dual graph for the special case, we can derive the mathematical formulation for constrained edge bipartization to solve the DSA grouping and decomposition problem. The notations of the variables are illustrated in Table.1. For the general conflict graph:

$$\min \quad \sum e_{uv}, \quad e_{uv} \in E_c \tag{2a}$$

$$\text{s.t.} \quad s_u + s_v + e_{uv} \geq 1, \qquad \forall (u, v) \in E, \tag{2b}$$

$$s_u + s_v - e_{uv} \leq 1, \qquad \forall (u, v) \in E, \tag{2c}$$

$$\sum_i e_{uv_i} \leq 2, \qquad \forall u \in V, \ \forall e_{uv_i} \in E_g \tag{2d}$$

$$e_{uv} + e_{uw} \leq 1, \qquad e_{uv}, e_{uw} \in E_g, \ \forall \ conflicted \ e_{uv} \ and \ e_{vw}, \tag{2e}$$

$$\sum e_{u_i u_{i+1}} \leq max_g, \qquad e_{u_i u_{i+1}} \in E_g \tag{2f}$$

$$e_{uv} \in (0, 1), \tag{2g}$$

As we mentioned earlier that some of the edges will be deleted in order to convert the original conflict graph to a bipartite graph. Vertices on the two ends of the deleted edge that can be grouped in the same DSA guiding template are preferred. Otherwise, if vertices on the deleted edge can not be grouped in the same guiding template, a conflict occurs. Therefore, in order to DSA conflicts, our objective 2a is to minimize the cost of deleted edges that cannot be grouped by DSA during the edge bipartization process. $e_{uv} = 1$ indicates that this edge is deleted. Constraints 2b and 2c are used to ensure that the two vertices of each edge are divided into two sides. If $s_u = s_v$, then vertices u and v are on the same side. So the edge, e_{uv}, should be deleted, $e_{uv} = 1$.

Table 2. Notations

EP_g	groupable edge set in dual graph
EP_c	conflict edge set in dual graph
EP	edge set in dual graph
VP	vertices set in dual graph
ep_{uv}	edge in dual graph
max_g	maximum group number

Otherwise, if $s_u \neq s_v$, vertices u and v are on different sides. There is no need to delete e_{uv}. Equation 2d is used to forbid the fork patterns in which there are more than two out-edges of a vertex. This can be realized by limiting the degree of deleted edges of each node less than or equal to 2. Equation 2e is used to forbid the forbidden patterns, such as a DSA pattern with acute angles. Equation 2f is used to limit the maximum DSA group size. The number vertices from connected deleted edges should be less than or equal to max_g.

As explained in previous section 2.2, the placement error is quite large for DSA groups whose size is larger than 4. Thus the maximum group size in our design is less than or equal to 3. Therefore, circular patterns are automatically avoided with equation 2e.

For the planar graph, the corresponding notations are illustrated in Table 2.

$$\max \quad \sum ep_{uv}, \quad ep_{uv} \in EP_c, \tag{3a}$$

$$\text{s.t.} \quad \sum_{u_i} ep_{u_i v} = 2 \times m_v, \quad \forall v \in VP, \tag{3b}$$

$$\sum_{u_i} (1 - ep_{vu_i}) \leq 2, \quad \forall v \in VP, \ \forall ep_{vu_i} \in EP_g, \tag{3c}$$

$$(1 - ep_{uv}) + (1 - ep_{uw}) \leq 1, \quad \forall \ conflicted \ ep_{uv} \ and \ ep_{uw} \tag{3d}$$

$$\sum (1 - ep_{v_i v_{i+1}}) \leq max_g, \tag{3e}$$

$$ep_{uv} \in (0, 1), \tag{3f}$$

$$m_v \in (0, 1, 2, 3, ...), \tag{3g}$$

The mathematical formulation is different from the one for the general graphs. It is noted that the edge is denoted as "0" if it is deleted, $ep_{uv} = 0$, thus in order to minimize the number of DSA conflicts, we can maximize the existing edges whose vertices cannot be grouped by DSA as shown in Equation 3a. Equation 3b ensures that the degree of each node is an even number. Equation 3c is used to limit the forbidden fork patterns in which the vertices have more than 2 out-edges. Equation 3d is used to avoid the forbidden DSA patterns, such as acute DSA patterns that are difficult to synthesize. Equation 3e is used to limit the maximum group size.

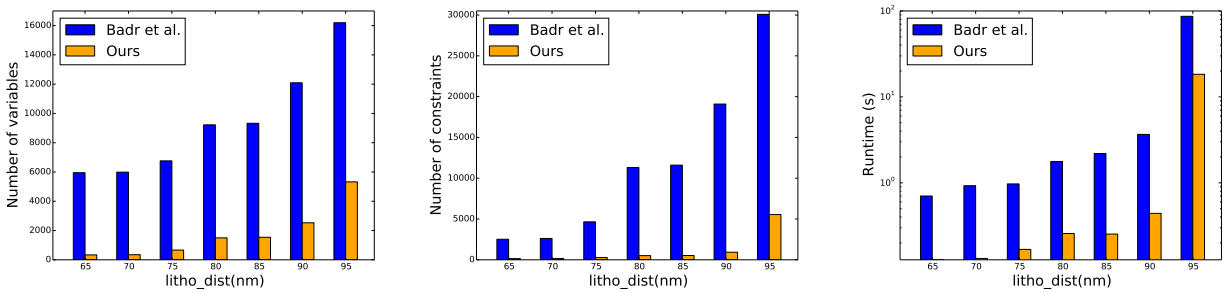
4. DSA-DP EXPERIMENTAL RESULTS

We implement our method in C++ on a Linux workstation, and test it on four benchmarks of via layer from an industrial 7nm design. In order to compare the result with previous works, we also implement the work of Badr et al.[15]

First, the computational analysis is performed in terms of the number of variables and constraints between our work and Badr et al.[15] Since the original conflict graph in our work and Badr et al.[15] is the same, we assume that the total number of edges and the total number of vertices are E and V of the conflict graph. F is the number of planar faces if the conflict graph is a planar graph. Let M denote the estimated number of constraints for the mathematical formulation. The analysis is shown in Table 3. It can be observed that our formulation can have much less number of variables and constraints than Badr et al.[15] Because the number of planar faces is much less than the number of original vertices in the conflict graph, the planar graphs can save a significant number of variables and constraints compared to the general mathematical formulation and Badr et al.[15]

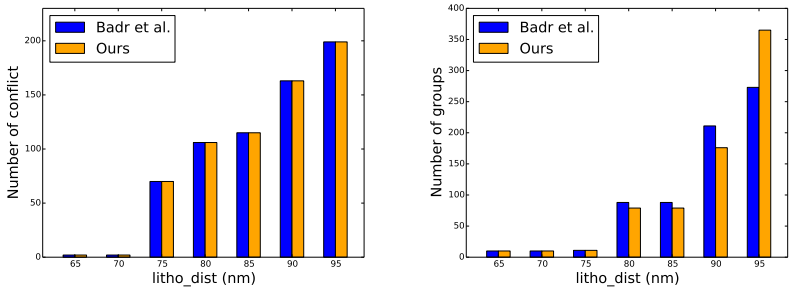
Table 3. Number of variables and constraints

	#Variables	#Constraints
General	E+V	2E+M
Planar	E+F	F+M
Badr et al.[15]	3E+2V	11E+M



(a) (b) (c)

Figure 10. Number of variables and constraints, runtime comparison



(a) (b)

Figure 11. Conflict and group number comparison

In order to test the performance of the formulation with different "*litho_dist*" values, we vary it from 65nm to 95nm. Thus the complexity of the conflict graphs increases accordingly. The comparison results are shown in Figure 10 and Figure 11. In Figure 10, we compare the number of variables and constraints, as well as the runtime. Figure 10 shows that our new formulation has more than 4× less variables and 6× less constraints than Badr et al.[15] As a result, our runtime is about 5× faster.

We also compare the number of conflicts and DSA groups in Figure 11. Since both of our formulations are trying to minimize the number of decomposition conflicts, and both formulations can achieve the optimal result, the number of conflicts are the same for both formulations. However, the number of final DSA groups are different between our method and Badr et al.[15] since we do not restrict the number of groups.

5. CONCLUSION

With the scaling down of technology nodes, DSA is a promising alternative lithography technique for the fabrication of via layers. In this work, we analyze the placement error of DSA patterns with different shapes and sizes. We then propose a graph-based approach to reduce the problem size and solve the DSA-DP problem more efficiently. The experimental results demonstrate that we can achieve a 50% reduction on both the number of variables and constraints compared to previous work, which results in a 50X speed up in runtime.

REFERENCES

[1] Gronheid, R. and Nealey, P., [*Directed Self-assembly of Block Co-polymers for Nano-manufacturing*], Woodhead Publishing (2015).

[2] Wong, H.-S. P., Yi, H., Tung, M., and Okabe, K., "Physical layout design of directed self-assembly guiding alphabet for IC contact hole/via patterning," in [*ACM International Symposium on Physical Design (ISPD)*], 65–66 (2015).

[3] Ma, Y., Wang, Y., Word, J., Lei, J., Mitra, J., Torres, J. A., Hong, L., Fenger, G., Khaira, D., Preil, M., et al., "Directed self assembly (DSA) compliant flow with immersion lithography: from material to design and patterning," in [*Proceedings of SPIE*], **9777** (2016).

[4] Yu, B., Xu, X., Roy, S., Lin, Y., Ou, J., and Pan, D. Z., "Design for manufacturability and reliability in extreme-scaling vlsi," *Science China Information Sciences* (2016).

[5] Ou, J., Yu, B., Gao, J.-R., Pan, D. Z., Preil, M., and Latypov, A., "Directed self-assembly based cut mask optimization for unidirectional design," in [*ACM Great Lakes Symposium on VLSI (GLSVLSI)*], 83–86 (2015).

[6] Lin, Z.-W. and Chang, Y.-W., "Cut redistribution with directed self-assembly templates for advanced 1-D gridded layouts," in [*IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*], 89–94 (2016).

[7] Xiao, Z., Du, Y., Wong, M. D. F., and Zhang, H., "DSA template mask determination and cut redistribution for advanced 1D gridded design," in [*Proceedings of SPIE*], **8880** (2013).

[8] Ou, J., Yu, B., and Pan, D. Z., "Concurrent guiding template assignment and redundant via insertion for DSA-MP hybrid lithography," in [*ACM International Symposium on Physical Design (ISPD)*], 39–46 (2016).

[9] Shim, S., Chung, W., and Shin, Y., "Redundant via insertion for multiple-patterning directed-self-assembly lithography," in [*ACM/IEEE Design Automation Conference (DAC)*], (2016).

[10] Fang, S.-Y., Hong, Y.-X., and Lu, Y.-Z., "Simultaneous guiding template optimization and redundant via insertion for directed self-assembly," in [*IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*], 410–417 (2015).

[11] Du, Y., Xiao, Z., Wong, M. D. F., Yi, H., and Wong, H.-S. P., "DSA-aware detailed routing for via layer optimization," in [*Proceedings of SPIE*], **9049** (2014).

[12] Ou, J., Yu, B., Xu, X., Mitra, J., Lin, Y., and Pan, D. Z., "Dsar: Dsa aware routing with simultaneous dsa guiding pattern and double patterning assignment," in [*ACM International Symposium on Physical Design (ISPD)*], (2017).

[13] Su, Y.-H. and Chang, Y.-W., "Vcr:simultaneous via-template and cut-template-aware routing for directed self-assembly technology," in [*IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*], (2016).

[14] Ma, Y., Lei, J., Torres, J. A., Hong, L., Word, J., Fenger, G., Tritchkov, A., Lippincott, G., Gupta, R., Lafferty, N., He, Y., Bekaert, J., and Vanderberghe, G., "Directed self-assembly (DSA) grapho-epitaxy template generation with immersion lithography," in [*Proceedings of SPIE*], **9423** (2015).

[15] Badr, Y., Torres, A., and Gupta, P., "Mask assignment and synthesis of DSA-MP hybrid lithography for sub-7nm contacts/vias," in [*ACM/IEEE Design Automation Conference (DAC)*], 70:1–70:6 (2015).

[16] Kuang, J., Ye, J., and Young, E. F. Y., "Simultaneous template optimization and mask assignment for DSA with multiple patterning," in [*IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*], 75–82 (2016).

[17] Graphics, M., "Calibre computational lithography."

[18] Guo, J., Gramm, J., Fuffner, F., Niedermeier, R., and Wernicke, S., "Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization," *Journal of Computer and System Sciences* **72**, 1386–1396 (2996).