A Unified Framework for Simultaneous Layout Decomposition and Mask Optimization

Yuzhe Ma, Wei Zhong, Shuxiang Hu, Jhih-Rong Gao, Jian Kuang, Jin Miao, and Bei Yu, *Member, IEEE*

***Abstract*—In advanced technology nodes, layout decomposition (LD) and mask optimization (MO) are two key stages in integrated circuit design. Due to the inconsistency of the objectives of these two stages, the performance of conventional layout and MO may be suboptimal. To tackle this problem, in this article, we propose a unified framework, which seamlessly integrates LD and MO. We propose a gradient-based approach to solve the unified mathematical formulation, as well as a set of discrete optimization techniques to avoid being stuck in local optimum. The conventional optimization process can be accelerated as some inferior decomposition results can be smartly pruned in early stages. The experimental results show that the proposed unified framework can achieve more than 34× speed-up compared with the conventional two-stage flow, meanwhile, it can dramatically reduce EPE violations by more than 8×, and thus maintain better design quality.**

***Index Terms*—Design for manufacturing, layout decomposition, mask optimization, VLSI design.**

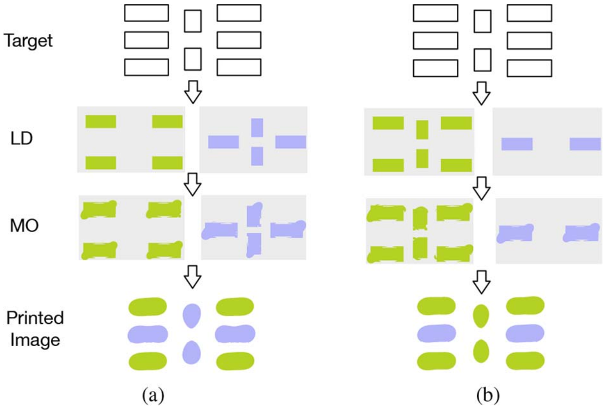


Fig. 1. Same quality LDs can achieve different EPE violation number after MO. (a) Solution 1 with #EPE violation = 3. (b) Solution 2 with #EPE violation = 1.

I. Introduction

In ACCORDANCE with the Moore's law, through extreme scaling, the transistor number on a chip has increased exponentially in the last five decades. However, the continued scaling of the transistor feature size has pushed the conventional 193-nm wavelength lithography system into its resolution limit, thus, the whole semiconductor industry is facing severe manufacturing challenges [1]. To overcome these issues, resolution enhancement techniques (RETs) on layout and mask levels toward better printability and yield are of great importance [2].

Two of the most critical RET stages are layout decomposition (LD) and mask optimization (MO). In the first stage, LD divides target image into several masks so that the coarser pitches on every mask can be manufactured through 193-nm wavelength lithography. According to different processes, LD can be classified into litho-etch-litho-etch (LELE)-type or spacer-type [1]. Since the LELE-type manufacturing process can support complex and flexible design patterns, in this article, we concentrate on LELE-type LD. Depending on the total mask number available, the problem is also called double patterning LD (two masks) or triple patterning LD (three masks). In the second stage, each decomposed mask needs to be further refined by MO, e.g., optical proximity correction (OPC), to reduce edge placement error (EPE). Finally, all optimized masks go through the lithography process separately to generate the corresponding printed image, and all printed images are combined together to generate target.

In emerging technology nodes, the conventional two-stage flow (i.e., LD followed by MO) cannot achieve good printability on their own. The reasons are twofold.

1) The LD and MO are separated from each other and each problem is solved independently, which may lose a global view.
2) Due to the inconsistency between the objectives of the two stages, decomposed results with identical quality may cause diverse printed image qualities after MO.

That is, the LD is based on simple design or coloring rules, which are just coarse regression of complicated lithography model; while the MO is verified by accurate and sophisticated lithography simulation. Fig. 1 gives an example on such inconsistency. Given the identical target, two different

Manuscript received August 18, 2019; revised December 9, 2019 and January 30, 2020; accepted March 9, 2020. Date of publication March 17, 2020; date of current version November 20, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61906029, in part by the Fundamental Research Funds for the Central Universities, and in part by the Research Grants Council of Hong Kong under Grant CUHK24209017. The preliminary version has been presented at the IEEE International Conference On Computer Aided Design (ICCAD) in 2017. This article was recommended by Associate Editor I.-H.-R. Jiang. *(Corresponding authors: Wei Zhong; Bei Yu.)*

Yuzhe Ma and Bei Yu are with the Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong (e-mail: byu@cse.cuhk.edu.hk).

Wei Zhong and Shuxiang Hu are with the International School of Information Science and Engineering, Dalian University of Technology, Dalian 116024, China (e-mail: zhongwei@dlut.edu.cn).

Jhih-Rong Gao, Jian Kuang, and Jin Miao are with the DSG Group, Cadence Design Systems, San Jose, CA 95134 USA.

Digital Object Identifier 10.1109/TCAD.2020.2981457

LD results are found (LD stage in the figures), and both of them satisfy all design rules and coloring rules. After the MO (MO stage in the figures) on each mask, however, it can be observed that the qualities of the printed images are diverse: Fig. 1(a) has three EPE violations, while Fig. 1(b) has only one EPE violation. Therefore, there is an increasing need to bridge the gap between LD and MO by a unified design framework.

There is a wealth of literature on LELE-type LD. The general LD problem minimizing both conflict and stitch can be optimally solved through integer linear programming (ILP) [3]–[5]. Due to the computationally intractable of ILP solutions, there are several speed-up techniques under different scenarios. For the double patterning scenario, Xu and Chu [6] formulated the problem into a maximum-cut problem and proposed an ILP formulation on stitch minimization, while Tang and Cho [7] computed a stitch graph and proved that the stitch graph is planar, and proposed min-cut-based approaches to minimize the stitch number optimally. For the triple patterning or general multiple patterning scenarios, Yu *et al.* [5] and Yu and Pan [8] proposed semidefinite programming (SDP) formulation to reduce the ILP formulation. Fang *et al.* [9] discussed several graph division techniques and proposed a stitch-aware mask assignment algorithm which can find the mask assignment such that the conflicts in the same mask are more likely to be resolved by inserting stitches; Tian *et al.* [10] proposed a polynomial-time algorithm for row-based design and the mask assignment was found by searching for a shortest path. Lin *et al.* [11] developed a linear programming-based relaxation. Kuang and Young [12] applied efficient graph matching. Chang and Jiang [13] considered more complex coloring rules for triple patterning LD, where the graph coloring problem was reduced to an exact cover problem. Kuang and Young [14] applied fixed-parameter tractable (FPT) algorithm to LD. There are few studies considering the quality of printed images in the early LD stage. Yu *et al.* [15] optimized the local balanced density, while Chen *et al.* [16] introduced the concept of spacing-based density. However, both density measurements are just coarse estimate of complicated lithography model, thus their fidelity to the MO is not guaranteed.

There are also investigations on MO or OPC, which can be classified into three types: 1) rule-based OPC; 2) model-based OPC; and 3) inverse lithography technique (ILT). Rule-based OPC requires comprehensive experiments determining design rules to compensate nondesired patterns, thus can only be applied to less aggressive designs [17]. Model-based OPC segments pattern edges into small parts and moves them slightly to make corrections for printed images. However, it is heavily based on lithography simulation so usually is time-consuming [18]–[20]. ILT has become a promising OPC solution, which handles the MO as an inverse problem of the lithography system. Poonawala and Milanfar [21] proposed a systematic formulation. Jia and Lam [22] applied a stochastic gradient descent method to update the mask in each iteration. Gao *et al.* [23] proposed a speed-up technique for lithography simulation, as well as a formulation for accurate EPE calculation. Note that some previous [24]–[26] studied multiple exposure effects in ILT framework, but none of them addressed the LD problem because they only considered the multiple exposures on a single mask.

TABLE I
NOTATIONS USED IN THIS ARTICLE

$\boldsymbol{Z}_t$	Target image
$\boldsymbol{Z}_1$, $\boldsymbol{Z}_2$	Binary images under the nominal condition
$\boldsymbol{Z}$	Printed image under the nominal condition
$\boldsymbol{Z}^{(n)}$	The printed image under the n-th process condition
$\boldsymbol{M}_1$, $\boldsymbol{M}_2$	Output masks
$\boldsymbol{P}_1$, $\boldsymbol{P}_2$	Unconstrained variables
$\boldsymbol{I}_1$, $\boldsymbol{I}_2$	Aerial images under the nominal condition
$\boldsymbol{I}_1^i$, $\boldsymbol{I}_2^i$	Aerial images under the i-th process condition
$\boldsymbol{H}$	A set of optical kernels $\{\boldsymbol{h}_1, \ldots, \boldsymbol{h}_K\}$
$\boldsymbol{H}^*$	The conjugate transpose of $\boldsymbol{H}$
$\odot$	Element-wise matrix multiplication operator
$\otimes$	Convolution operator

In this article, we propose a unified optimization framework which aims at solving LD and MO simultaneously. Combining the two processes together leads to a larger solution space, which has the potential to obtain a higher quality mask design. In addition, through effective pruning techniques, our framework can avoid exhaustive MO on all LD solutions, therefore, the overall efficiency can be significantly improved. The main contributions of this article are listed as follows.

1) To the best of our knowledge, this is the first work handling multiple patterning LD and MO simultaneously.
2) We propose a unified problem formulation and develop a gradient-based optimization approach, while the process variation issue is studied to guarantee mask robustness.
3) We further apply a set of discrete optimization techniques (e.g., semi-definite programming, randomized rounding, and pruning) to avoid being stuck in local optimum.
4) The experimental results verify the effectiveness of the proposed framework.

The remainder of this article is organized as follows. Section II introduces lithography models and evaluation criteria. Section III formulates the problem mathematically and describes algorithmic details in our framework. Section IV lists the experimental results to support our methodologies, followed by a discussion in Sections V and VI concludes this article.

II. PRELIMINARIES

In this section, we provide preliminaries on lithography models, and then introduce the evaluation criteria. For convenience, notations used in this article are listed in Table I. Note that in this article we focus on double patterning scenario, but the problem formulation and the corresponding methodologies can be extended to triple patterning counterpart.

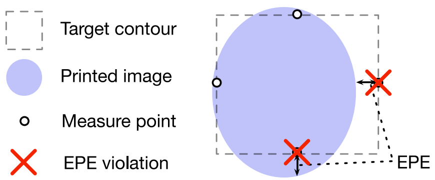


Fig. 2. Illustration of EPE measurement.

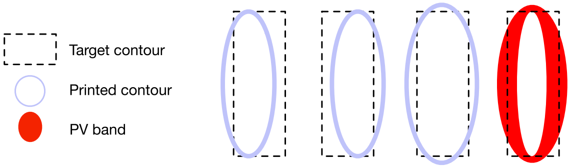


Fig. 3. Example of PV Band.

A. Forward Lithography Models

Two models are needed to transform mask patterns into the printed image: optical lithography model and photo-resist model. First, an aerial image $\boldsymbol{I}$ is generated by convolving the mask $\boldsymbol{M}$ with a set of optical kernels [27], which is represented as

$$\boldsymbol{I} = f_{\text{optical}}(\boldsymbol{M}) = \sum_{k=1}^{K} w_k \cdot |\boldsymbol{M} \otimes \boldsymbol{h}_k|^2 \tag{1}$$

where $\boldsymbol{h}_k$ is the kth optical kernel, w_k is the weight of $\boldsymbol{h}_k$, and K is the total kernel number.

Then a resist model is applied to the aerial image. In this article, a constant threshold resist model is used, which sets an intensity threshold I_{th} to binarize the aerial image, denoted by $\boldsymbol{Z}$ in the following equation:

$$\boldsymbol{Z}(x, y) = f_{\text{resist}}(\boldsymbol{I}) = \begin{cases} 1, & \text{if } \boldsymbol{I}(x, y) \geq I_{\text{th}} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Finally, binary images $\boldsymbol{Z}_1 = f_{\text{resist}}(\boldsymbol{I}_1)$ and $\boldsymbol{Z}_2 = f_{\text{resist}}(\boldsymbol{I}_2)$ are combined to form the printed image. Considering that the printed image is binary as well, the process can be represented by performing logical OR operation as follows:

$$\boldsymbol{Z}(x, y) = \boldsymbol{Z}_1(x, y) \vee \boldsymbol{Z}_2(x, y). \tag{3}$$

B. Evaluation Metrics

Given a target layout and the printed image, the EPE and process variation band (PV Band) are defined as follows.

Definition 1 (EPE): EPE is defined as the geometric displacement of the image contour from the edge of the target image on the layout. A violation is introduced if the perpendicular displacement is greater than an EPE threshold value, as shown in Fig. 2.

Definition 2 (PV Band): PV Band is measured as the area between the outermost printed edge and the innermost printed edge among all process conditions, as shown in Fig. 3. It is used to evaluate the mask robustness against process variations.

In our implementation, the EPE threshold value is set to 10 nm. An example of our EPE measurement is given in Fig. 2: to facilitate the computation, a set of measure points are sampled on each edge and EPE violation will be checked at the measure points. For PV Band, we use XOR operations to compute the region among all possible printed images.

III. Methodologies

Given the above notations, the problem of LD and MO (LDMO) is defined as follows.

Problem 1 (LDMO): Given target image $\boldsymbol{Z}_t$, two optimized masks, $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$, are generated. The objective is to minimize the difference between the final printed image $\boldsymbol{Z}$ and the target image $\boldsymbol{Z}_t$.

A. Mathematical Formulation for LDMO

Since we are seeking a pair of masks which can form printed image with high fidelity, the LDMO problem can be formulated as an optimization problem as follows:

$$\min_{\boldsymbol{M}_1, \boldsymbol{M}_2} \quad F = \|\boldsymbol{Z}_t - \boldsymbol{Z}\|_2^2 \tag{4a}$$

$$\text{s.t.} \quad \boldsymbol{M}_1(x, y) \in \{0, 1\} \quad \forall x, y \tag{4b}$$

$$\boldsymbol{M}_2(x, y) \in \{0, 1\} \quad \forall x, y \tag{4c}$$

$$\boldsymbol{I}_1 = \sum_{k=1}^{K} w_k \cdot |\boldsymbol{M}_1 \otimes \boldsymbol{h}_k|^2 \tag{4d}$$

$$\boldsymbol{I}_2 = \sum_{k=1}^{K} w_k \cdot |\boldsymbol{M}_2 \otimes \boldsymbol{h}_k|^2 \tag{4e}$$

$$\boldsymbol{Z} = f_{\text{resist}}(\boldsymbol{I}_1) \vee f_{\text{resist}}(\boldsymbol{I}_2). \tag{4f}$$

The problem is strongly nonconvex with discrete constraints, thus is hard to be solved directly. In this section, we propose a unified flow for solving the LDMO problem.

B. Numerical Optimization

The gradient-based method has been widely adopted in solving numerical optimization problems. However, there are nondifferentiable discrete constraints in our formulation. Therefore, it is necessary to do relaxation before deriving the gradient.

In the formulation of LDMO, the variables $\boldsymbol{M}_1$, $\boldsymbol{M}_2$, $\boldsymbol{Z}_1$, and $\boldsymbol{Z}_2$ are binary, which are nondifferentiable. One possible method is to relax them into floating values with a feasible region of [0,1], which cannot be solved by the gradient-based method directly. Alternatively, the binary constraints can be replaced with *sigmoid* function for relaxation so that the variables become unconstrained, and hence convenient to derive the gradient. To apply *sigmoid* function, we need to introduce new variables $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$. Then $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ can be relaxed by applying *sigmoid* function on $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$, respectively,

$$\boldsymbol{M}_1(x, y) = \text{sig}(\boldsymbol{P}_1(x, y)) = \frac{1}{1 + \exp\left[-\theta_M \boldsymbol{P}_1(x, y)\right]} \tag{5}$$

$$\boldsymbol{M}_2(x, y) = \text{sig}(\boldsymbol{P}_2(x, y)) = \frac{1}{1 + \exp\left[-\theta_M \boldsymbol{P}_2(x, y)\right]}. \tag{6}$$

Similarly, the *sigmoid* function can be applied to $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ to relax $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$ as shown in (7) and (8). θ_M and θ_Z

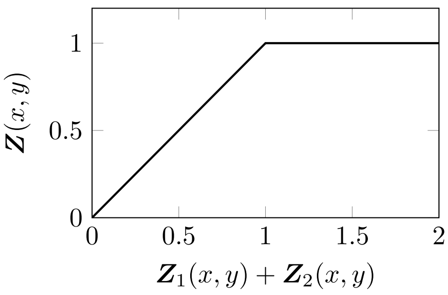


Fig. 4. Relation between $\boldsymbol{Z}(x, y)$ and $\boldsymbol{Z}_1(x, y) + \boldsymbol{Z}_2(x, y)$.

are user-defined parameters which represent the steepness of *sigmoid* functions, and I_{th} is the threshold in the resist model

$$\boldsymbol{Z}_1(x, y) = \text{sig}(\boldsymbol{I}_1(x, y)) = \frac{1}{1 + \exp\left[-\theta_Z(\boldsymbol{I}_1(x, y) - I_{\text{th}})\right]} \quad (7)$$

$$\boldsymbol{Z}_2(x, y) = \text{sig}(\boldsymbol{I}_2(x, y)) = \frac{1}{1 + \exp\left[-\theta_Z(\boldsymbol{I}_2(x, y) - I_{\text{th}})\right]}. \quad (8)$$

Note that $\boldsymbol{Z}$ is also a binary value, but different from $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$, it is calculated by logical OR. We relax constraint (4f) to

$$\boldsymbol{Z}(x, y) = \min\{\boldsymbol{Z}_1(x, y) + \boldsymbol{Z}_2(x, y), 1\}. \quad (9)$$

Considering that the maximum value of $\boldsymbol{Z}$ before relaxation is 1, here, we set an upper bound to 1, which may reduce the error when calculating the objective value. The relation between $\boldsymbol{Z}(x, y)$ and $(\boldsymbol{Z}_1(x, y) + \boldsymbol{Z}_2(x, y))$ is shown in Fig. 4. Then it is easy to derive the gradient formulation of $\boldsymbol{Z}$ with respect to $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$, denoted by $\boldsymbol{B}$, which is given by

$$\frac{\partial \boldsymbol{Z}(x, y)}{\partial \boldsymbol{Z}_1(x, y)} = \frac{\partial \boldsymbol{Z}(x, y)}{\partial \boldsymbol{Z}_2(x, y)} = \boldsymbol{B}(x, y) = \begin{cases} 1, & \text{if } \boldsymbol{Z}(x, y) \leq 1 \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

After relaxation, we can formulate the relaxed LDMO problem as follows:

$$\begin{aligned} \min_{\boldsymbol{P}_1, \boldsymbol{P}_2} \quad & F = \|\boldsymbol{Z}_t - \boldsymbol{Z}\|_2^2 \\ \text{s.t.} \quad & \text{(4d) and (4e), (5)–(9).} \end{aligned} \quad (11)$$

Now variables $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ are unconstrained, and functions in (5)–(9) are differentiable. We obtain the gradient according to the chain rule

$$\begin{aligned} \frac{\partial F}{\partial \boldsymbol{P}_1(x, y)} &= \frac{\partial \sum_{i,j} (\boldsymbol{Z}_t(i, j) - \boldsymbol{Z}(i, j))^2}{\partial \boldsymbol{P}_1(x, y)} \\ &= 2 \sum_{i,j} (\boldsymbol{Z}(i, j) - \boldsymbol{Z}_t(i, j)) \cdot \frac{\partial \boldsymbol{Z}(i, j)}{\partial \boldsymbol{Z}_1(i, j)} \cdot \frac{\partial \boldsymbol{Z}_1(i, j)}{\partial \boldsymbol{P}_1(x, y)} \end{aligned} \quad (12)$$

where

$$\begin{aligned} \frac{\partial \boldsymbol{Z}_1(i, j)}{\partial \boldsymbol{P}_1(x, y)} &= \theta_M \theta_Z \boldsymbol{Z}(i, j)(1 - \boldsymbol{Z}(i, j)) \\ &\quad \times \left\{\left[\boldsymbol{M}_1(i, j) \otimes \boldsymbol{H}^*(i, j)\right] \boldsymbol{H}(i - x, j - y) \right. \\ &\qquad \left. + \left[\boldsymbol{M}_1(i, j) \otimes \boldsymbol{H}(i, j)\right] \boldsymbol{H}^*(i - x, j - y)\right\} \\ &\quad \times \boldsymbol{M}_1(i, j)\left[1 - \boldsymbol{M}_1(i, j)\right]. \end{aligned} \quad (13)$$

Algorithm 1 Numerical Optimization Flow

```
1: Initialize $P_1$, $P_2$, maximum iteration $T$, tolerance $\epsilon$;
2: for $i = 1, \ldots, T$ do
3:     Compute the printed image according to current $P_1$ and $P_2$;
4:     if printed image is illegal then
5:         Discrete optimization ;                    ▷ Section III-E
6:     else
7:         MaskUpdate($P_1$, $P_2$);                  ▷ Algorithm 2
8:         if $\text{RMS}(\nabla_{P_1} F) + \text{RMS}(\nabla_{P_2} F) \leq \epsilon$ then
9:             break;
10:        end if
11:    end if
12: end for
```

Algorithm 2 Gradient-Based Mask Update

```
function MaskUpdate($P_1$, $P_2$)
    Initialize stepsize $t$;
    Compute the relaxed masks $M_1, M_2$;
    Compute $Z$ according to current $P_1$ and $P_2$;
    Compute the gradient $\nabla_{P_1} F$, $\nabla_{P_2} F$ through (4d) and (4e), (17) and (18);
    $P_1 \leftarrow P_1 - t \times \nabla_{P_1} F$;
    $P_2 \leftarrow P_2 - t \times \nabla_{P_2} F$;
    return $P_1$,$P_2$, $\nabla_{P_1} F$, $\nabla_{P_2} F$;
end function
```

Then we can compute the gradient of F with respect to $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ as follows:

$$\begin{aligned} \nabla_{\boldsymbol{P}_1} F &= 2\theta_M \theta_Z \times \boldsymbol{M}_1 \odot (1 - \boldsymbol{M}_1) \\ &\quad \odot \left\{\boldsymbol{H} \otimes \left[(\boldsymbol{Z} - \boldsymbol{Z}_t) \odot \boldsymbol{B} \odot \boldsymbol{Z} \odot (1 - \boldsymbol{Z}) \odot \left(\boldsymbol{M}_1 \otimes \boldsymbol{H}^*\right)\right] \right. \\ &\qquad \left. + \boldsymbol{H}^* \otimes \left[(\boldsymbol{Z} - \boldsymbol{Z}_t) \odot \boldsymbol{B} \odot \boldsymbol{Z} \odot (1 - \boldsymbol{Z}) \odot (\boldsymbol{M}_1 \otimes \boldsymbol{H})\right]\right\} \end{aligned} \quad (14)$$

$$\begin{aligned} \nabla_{\boldsymbol{P}_2} F &= 2\theta_M \theta_Z \times \boldsymbol{M}_2 \odot (1 - \boldsymbol{M}_2) \\ &\quad \odot \left\{\boldsymbol{H} \otimes \left[(\boldsymbol{Z} - \boldsymbol{Z}_t) \odot \boldsymbol{B} \odot \boldsymbol{Z} \odot (1 - \boldsymbol{Z}) \odot \left(\boldsymbol{M}_2 \otimes \boldsymbol{H}^*\right)\right] \right. \\ &\qquad \left. + \boldsymbol{H}^* \otimes \left[(\boldsymbol{Z} - \boldsymbol{Z}_t) \odot \boldsymbol{B} \odot \boldsymbol{Z} \odot (1 - \boldsymbol{Z}) \odot (\boldsymbol{M}_2 \otimes \boldsymbol{H})\right]\right\}. \end{aligned} \quad (15)$$

The numerical optimization algorithm is described in Algorithm 1. First, we initialize $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$, the maximum iteration number T and the tolerance ϵ (line 1). An intuitive initial solution is that $\boldsymbol{P}_1$ is initialized so that the corresponding $\boldsymbol{M}_1$ is identical to the target image, and $\boldsymbol{P}_2$ is initialized so that $\boldsymbol{M}_2$ is empty. In each iteration, the printed image is obtained based on variables $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ (line 3). The violation checking will be carried out in every iterations, which will be introduced in Section III-D. If the printed image is illegal, a discrete optimization step will be executed, which will be introduced in Section III-E. Otherwise, the function `MaskUpdate` will be called to update the masks. To save the runtime, the loop will exit early if the sum of the root mean square (RMS) of the gradient is less than a tolerance ϵ, which indicates that the objective function may be very close to the optimal value. The loop will finally terminate when the maximum number of iteration T is achieved. In our implementation, T is set to 40. θ_M and θ_Z are set to 85 and 4, respectively.

The procedure for mask update is described in Algorithm 2. To derive the gradient, the lithography simulation is conducted first to compute the corresponding printed image based on current masks (line 4). Next, gradient of objective F with respect to $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ are computed (line 5), followed by variables update (lines 6 and 7). In our implementation, the tolerance ϵ is set to 0.01 and stepsize t is set to 0.4.

C. Process Variation Aware Mask Optimization

Process variation is a critical issue in realistic manufacturing stage since it can impact the yield directly. Therefore, not only image fidelity but also mask robustness should be taken into considerations. In this section, we extend the LDMO problem to consider process variation issue. Based on the above notations, the problem of process variation-aware LD and MO is defined as follows.

Problem 2 (PV-LDMO): Given target image $\boldsymbol{Z}_t$, two optimized masks, $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$, are generated. The objectives are twofold: 1) minimize the difference between the final printed image $\boldsymbol{Z}$ and the target image $\boldsymbol{Z}_t$ and 2) the obtained masks, $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$, are robust to process variation.

PV Band is a commonly applied criterion to evaluate the mask robustness, which can be naturally integrated into the proposed unified optimization framework. The illustration of the PV Band is depicted in Section II. It can be observed that the calculation of PV Band requires a series of boolean operations among printed images under all possible process conditions. In order to make the calculation more tractable, we resort to minimize the difference between possible images and the target image, which is in accordance with the target of process variation optimization. The conventional OPC scenario optimizes single mask, therefore, the possibility of printed image is the same as the number of process corners N_p. Differently, PV-LDMO considers dual masks, thus, the possibility of printed image is more than N_p due to the possible combination on the separate mask. The PV-LDMO problem can be formulated as follows:

$$\min_{\boldsymbol{M}_1,\boldsymbol{M}_2} \quad F_{pv} = \|\boldsymbol{Z}_t - \boldsymbol{Z}\|_2^2 + \sum_{l=1}^{N_p^2} \|\boldsymbol{Z}_t - \boldsymbol{Z}_l\|_2^2 \tag{16a}$$

$$\text{s.t.} \quad \boldsymbol{I}_1^n = \sum_{k=1}^{K} w_{nk} \cdot |\boldsymbol{M}_1 \otimes \boldsymbol{h}_{nk}|^2, \quad n = 1 \ldots N_p \tag{16b}$$

$$\boldsymbol{I}_2^n = \sum_{k=1}^{K} w_{nk} \cdot |\boldsymbol{M}_2 \otimes \boldsymbol{h}_{nk}|^2, \quad n = 1 \ldots N_p \tag{16c}$$

$$\boldsymbol{Z}_l = f_{\text{resist}}\left(\boldsymbol{I}_1^i\right) \vee f_{\text{resist}}\left(\boldsymbol{I}_2^j\right)$$

$$i, j = 1 \ldots N_p, \quad l = 1 \ldots N_p^2 \tag{16d}$$

$$\text{(4b)–(4f)} \tag{16e}$$

where N_p is the number of process windows under consideration. $\boldsymbol{I}_{n1}$ and $\boldsymbol{I}_{n2}$ are the aerial images under the nth process window from $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$, respectively. $\boldsymbol{Z}_l$ is the lth possible combined printed image. Similar to solving Formulation 4, the same relaxation scheme can be applied here. Then we can derive the gradient based on the extended objective function and the chain rule as follows:

$$\begin{aligned}\nabla_{\boldsymbol{P}_1} F = {} & 2\theta_M\theta_Z \times \boldsymbol{M}_1 \odot (1-\boldsymbol{M}_1) \\ & \odot \Big\{\Big[\boldsymbol{H} \otimes \big((\boldsymbol{Z}-\boldsymbol{Z}_t) \odot \boldsymbol{B} \odot \boldsymbol{Z} \odot (1-\boldsymbol{Z}) \odot \big(\boldsymbol{M}_1 \otimes \boldsymbol{H}^*\big)\big) \\ & \quad + \boldsymbol{H}^* \otimes ((\boldsymbol{Z}-\boldsymbol{Z}_t) \odot \boldsymbol{B} \odot \boldsymbol{Z} \odot (1-\boldsymbol{Z}) \odot (\boldsymbol{M}_1 \otimes \boldsymbol{H}))\Big] \\ & \quad + \sum_{l=1}^{N_p^2}\Big[\boldsymbol{H} \otimes \big((\boldsymbol{Z}_l-\boldsymbol{Z}_t) \odot \boldsymbol{B} \odot \boldsymbol{Z}_l \odot (1-\boldsymbol{Z}_l) \odot \big(\boldsymbol{M}_1 \otimes \boldsymbol{H}^*\big)\big) \\ & \qquad + \boldsymbol{H}^* \otimes ((\boldsymbol{Z}_l-\boldsymbol{Z}_t) \odot \boldsymbol{B} \odot \boldsymbol{Z}_l \odot (1-\boldsymbol{Z}_l) \odot (\boldsymbol{M}_1 \otimes \boldsymbol{H}))\Big]\Big\}\end{aligned} \tag{17}$$

$$\begin{aligned}\nabla_{\boldsymbol{P}_2} F = {} & 2\theta_M\theta_Z \times \boldsymbol{M}_2 \odot (1-\boldsymbol{M}_2) \\ & \odot \Big\{\boldsymbol{H} \otimes \Big[(\boldsymbol{Z}-\boldsymbol{Z}_t) \odot \boldsymbol{B} \odot \boldsymbol{Z} \odot (1-\boldsymbol{Z}) \odot \big(\boldsymbol{M}_2 \otimes \boldsymbol{H}^*\big)\Big] \\ & \quad + \boldsymbol{H}^* \otimes [(\boldsymbol{Z}-\boldsymbol{Z}_t) \odot \boldsymbol{B} \odot \boldsymbol{Z} \odot (1-\boldsymbol{Z}) \odot (\boldsymbol{M}_2 \otimes \boldsymbol{H})]\Big\} \\ & \sum_{l=1}^{N_p^2}\Big[\boldsymbol{H} \otimes \big((\boldsymbol{Z}_l-\boldsymbol{Z}_t) \odot \boldsymbol{B} \odot \boldsymbol{Z}_l \odot (1-\boldsymbol{Z}_l) \odot \big(\boldsymbol{M}_2 \otimes \boldsymbol{H}^*\big)\big) \\ & \quad + \boldsymbol{H}^* \otimes ((\boldsymbol{Z}_l-\boldsymbol{Z}_t) \odot \boldsymbol{B} \odot \boldsymbol{Z}_l \odot (1-\boldsymbol{Z}_l) \odot (\boldsymbol{M}_2 \otimes \boldsymbol{H}))\Big]\Big\}.\end{aligned} \tag{18}$$

Then we follow the numerical optimization algorithm presented in Algorithm 1 to solve the extended PV-LDMO problem with new gradient calculation steps.

D. Violation Detection

If two features of the same mask are too close to each other, there will be a violation in the printed image after lithography simulation and it is difficult to be legalized only through gradient-based optimization. Intuitively, the violation can be resolved if the violated patterns are assigned to different masks. However, we need to locate where the violations occur. To do so, a Hanan-like grid is built based on the geometry of the bounding box of each target image. Each bounding box shares the same centroid with the pattern inside. Considering that the further assignment is also based on the grid and extra pattern may be generated around the original pattern during the MO, each bounding box is set to be a bit larger than the pattern. In our implementation, the extra width and extra height are both set to 20 nm. All the grids are then categorized into *pattern grid* and *spacing grid* depending on their positions on the target image. Different from conventional Hanan grid in which all the grids are aligned horizontally and vertically, the adjacent pattern grids in our Hanan-like grid will be merged so that a single pattern will not be split by grids. In addition, the spacing grids between two patterns are also merged into one grid. The orientation of each merged spacing grid is set according to its relative position to the two pattern grids, as shown in Fig. 5. The H, V, and D in Fig. 5 represent the orientation of horizontal, vertical, and diagonal, respectively.

As mentioned before, the violation checking is conducted by every w iterations rather than by each iteration such that the efficiency of the whole flow is maintained. The violation detection is performed through the following way. The printed image is first mapped to the Hanan-like grids. Since all the violations happen at the region between patterns, i.e., the spacing grids, we only check the spacing grids between two patterns

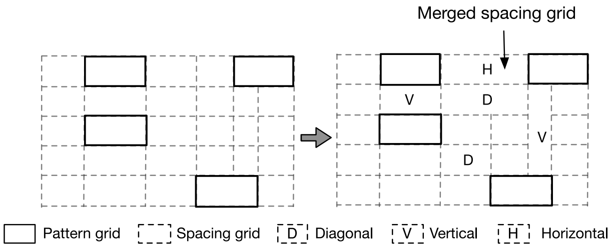


Fig. 5. Pattern grids and spacing grids.

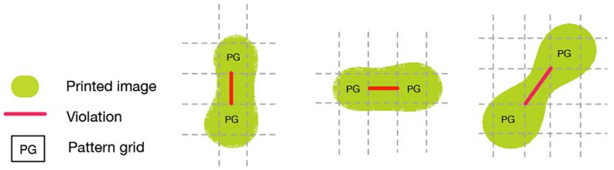


Fig. 6. Different kinds of violation.

(see Fig. 6). For each spacing grid, we initialize a matrix $\boldsymbol{A}$ with the same size as the spacing grid, so that each pixel in the grid corresponds to an entry of $\boldsymbol{A}$. Then we find all printable pixels in the grid and set the corresponding entries of the matrix to 1 and set the rest of the entries to 0. An intuitive observation is that a horizontal or vertical violation is caused by a printable line in spacing grids which connects an edge to the opposite side. Diagonal violation is due to the printable lines diagonally connecting two corners. For vertical or horizontal violation, we can check the sum of each row and each column of the matrix $\boldsymbol{A}$. Combined with the orientation, the violation can be determined. If the sum of one row is equal to the width W or the sum of one column is equal to the height H, there exists a violation. For diagonal violation, we compute the diagonal length of diagonal spacing grid. If the diagonal length is less than a spacing threshold, there is a diagonal violation. In our implementation, the threshold is set as 110 nm.

E. Discrete Optimization

The masks $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ are updated in each iteration to reduce the objective value. Since both LDMO and PV-LDMO are highly nonconvex problems, the gradient-based method can only achieve local optimum with poor quality. One reason is that the gradient-based method actually performs a greedy search and it is hard to escape from local optimum. To tackle this problem, we further propose a discrete optimization method to collaborate with the numerical optimization flow.

The most critical issue in multiple patterning lithography is the violations in printed images. Therefore, a critical step of the proposed framework is to resolve violations in the printed image and obtain high quality and robust masks. Generally, resolving violations can be achieved by assigning violated pattern grids to different masks. However, since the correlation between EPE violation and the distribution of patterns is unknown, it is difficult to derive a mathematical formulation to bridge the gap.

To overcome this issue, in this article, we develop a discrete optimization approach seeking a two-way partitioning of the pattern grids considering image violation, EPE violation, as well as potential spacing rules. Note that the proposed approach here can be easily extended to handle triple patterning lithography, where a three-way partitioning is adopted. With the position of printed image violation and the position of EPE violation, a weighted graph $G(V, E)$ can be constructed, where the vertex v_i represents ith pattern grid and the edges with weight 1 connecting two vertices are conflict with each other. In addition, we add edges with weight β between the vertices which have large EPE, where $0 < \beta < 1$. Therefore, the objective of discrete optimization is to find a cut of the graph so that the total weight of the edges between the cut and its complement is maximized. We use a vector $\boldsymbol{x}$ to denote the assignment of pattern grids, where $x_i = 1$ means v_i is assigned to mask 1 and $x_i = -1$ means v_i is assigned to mask 2. Moreover, in order to further ease the MO process, the graph is refined with one more kind of edge. Besides the violation edge and EPE edge, a set of *spacing edges* are added to the graph, which are similar to the diagonal violation edges in the sense that their existence is determined by the distance between each pair of nodes. The threshold for determining a spacing edge is 130 nm. Then the two-way partitioning problem can be formulated as follows:

$$\max_{\boldsymbol{x}} \sum_{(i,j)\in E} w_{ij}\big(1 - x_i x_j\big) \tag{19a}$$

$$\text{s.t.} \quad x_i \in \{-1, 1\} \quad \forall v_i \in V \tag{19b}$$

where w_{ij} defines the edge weight as follows:

$$w_{ij} = \begin{cases} 1, & \text{if } v_i \text{ and } v_j \text{ have violation} \\ \beta, & \text{if } v_i \text{ and } v_j \text{ both have large EPE} \\ \gamma, & \text{if } v_i \text{ and } v_j \text{ are close but not conflicted} \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

In our implementation, if the sum of EPE violations of two grids is greater than seven and they are not violated patterns, they will be connected by an edge of weight β, and β is set as 0.1.

Formulation in (19) can be approximated to an SDP with (21), which can be solved efficiently while maintaining high accuracy

$$\min_{\boldsymbol{X}} \quad \boldsymbol{W} \cdot \boldsymbol{X} \tag{21a}$$

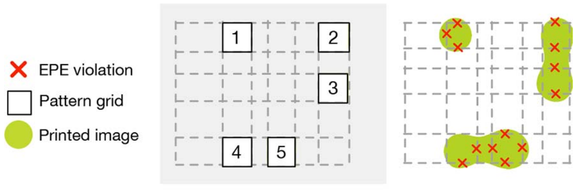
$$\text{s.t.} \quad \operatorname{diag}(\boldsymbol{X}) = \boldsymbol{e} \tag{21b}$$

$$\boldsymbol{X} \succeq \boldsymbol{0} \tag{21c}$$

where $\boldsymbol{e} = [1, 1 \ldots, 1]^\top$.

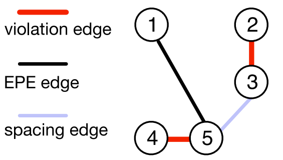
The optimal solution $\boldsymbol{X}^*$ of (21) need not to be in the form of $\boldsymbol{x}\boldsymbol{x}^\top$, and hence it does not yield a feasible solution to (19) immediately. However, we can extract from $\boldsymbol{X}^*$ a solution via randomized rounding [28]. First, we compute Cholesky factorization $\boldsymbol{X}^* = \boldsymbol{U}^\top \boldsymbol{U}$ of $\boldsymbol{X}^*$. The ith column of $\boldsymbol{U}$, denoted by $\boldsymbol{u}_i$, corresponds to the assignment of grid i. Let $\boldsymbol{r}$ be a vector uniformly distributed over the unit sphere (i.e., $\|\boldsymbol{r}\|_2 = 1$). Then we can set x_i as follows:

$$x_i = \operatorname{sgn}(\boldsymbol{u}_i^\top \boldsymbol{r}) = \begin{cases} 1, & \text{if } \boldsymbol{u}_i^\top \boldsymbol{r} \geq 0 \\ -1, & \text{otherwise.} \end{cases} \tag{22}$$



(a) (b)

Fig. 7. (a) Pattern on the mask. (b) Corresponding lithography printed image with EPE violations.



$$\boldsymbol{W}=\begin{bmatrix}0&0&0&0&\beta\\0&0&1&0&0\\0&1&0&0&\gamma\\0&0&0&0&1\\\beta&0&\gamma&1&0\end{bmatrix}$$

Fig. 8. Conflict graph and the corresponding weighted matrix $\boldsymbol{W}$.

In other words, we partition the grids according to whether their corresponding vectors lie "above" or "below" the hyperplane. The grids are, therefore, assigned to different masks according to the value of x_i.

An example of graph construction is given in Figs. 7 and 8. After solving the SDP, the solution $\boldsymbol{X}^*$ and $\boldsymbol{U}$ are given by

$$\boldsymbol{X}^*=\begin{bmatrix}1&0&0&1&-1\\0&1&-1&0&0\\0&-1&1&0&0\\1&0&0&1&-1\\-1&0&0&-1&1\end{bmatrix}$$

$$\boldsymbol{U}=\begin{bmatrix}1&0&0&1&-1\\0&1&-1&0&0\\0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\end{bmatrix}.$$

With a random vector $\boldsymbol{r}=[r_1,r_2\cdots r_5]^\top$ and (22), vector $\boldsymbol{x}$ is obtained by

$$\boldsymbol{x}=\left[\mathrm{sgn}(r_1),\mathrm{sgn}(r_2),\mathrm{sgn}(-r_2),\mathrm{sgn}(r_1),\mathrm{sgn}(-r_1)\right]^\top. \quad (23)$$

By solving SDP we can obtain multiple solutions which are useful to avoid being stuck in local optimum during the succeeding numerical optimization process. Furthermore, the runtime cost of solving SDP is much smaller than the lithography simulation. Therefore, SDP is an efficient method for our discrete optimization. Once the SDP is solved, we can obtain multiple solutions for $\boldsymbol{x}$. Then all these obtained solutions are further optimized through numerical optimization flow without violation checking, where these solutions will go through a pruning process and suboptimal ones will be removed.

The detailed procedure is shown in Algorithm 3. First, an empty set P is initialized to store the potential solutions (line 1). Then S solutions are obtained by randomized rounding, from which we can get the corresponding assignment solution for two masks, i.e., $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$. Each pair of $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ is treated as a 2-tuple which is stored in P (lines 2–6). Next, gradient-based mask update illustrated in Algorithm 3 is performed for T iterations for each solution in P. After that, the legality of the printed image of each solution will be checked. The solutions generating illegal printed images will be discarded first. Then the number of EPE violations of all the solutions will be compared and half of the solutions with larger EPE violation will be discarded (lines 8–15). The pruning process will be repeated until only one element in P is left. In order to balance the runtime and performance, T and S are both set to 5 in our implementation.

Algorithm 3 Pruning

```
Require: SDP solution $\boldsymbol{X}^*$.
Ensure: $\boldsymbol{P}_1$, $\boldsymbol{P}_2$.
 1: Initialize set P;
 2: for $i \leftarrow 1, \ldots, S$ do
 3:     Randomized rounding;                          ▷ Equation (22)
 4:     Get corresponding $\boldsymbol{P}_1^i$, $\boldsymbol{P}_2^i$;    ▷ $\{\boldsymbol{P}_1, \boldsymbol{P}_2\}$ is a 2-tuple
 5:     Save $\{\boldsymbol{P}_1^i, \boldsymbol{P}_2^i\}$ in P;
 6: end for
 7: while P.size()> 1 do
 8:     for all $\{\boldsymbol{P}_1^i, \boldsymbol{P}_2^i\} \in$ P do
 9:         for $j \leftarrow 1, \ldots, T$ do
10:             MaskUpdate($\boldsymbol{P}_1^i, \boldsymbol{P}_2^i$);          ▷ Algorithm 2
11:         end for
12:         Check legality of the printed image and get the number of EPE violations;
13:     end for
14:     Remove solutions with illegal printed image;
15:     Remove half of solutions with larger EPE violations;
16: end while
17: return the remaining $\{\boldsymbol{P}_1, \boldsymbol{P}_2\}$;
```

Compared with the graph construction method in [29], our conflict graph construction can reflect the lithography principles better by considering the spacing rules. The advantage is twofold.

1) By adding the spacing edges with weight γ, the solution space of the SDP is tighter than that in [29], which helps the search procedure to focus more on the superior solutions.
2) The runtime of the search and pruning routine (Algorithm 3) depends on the number of the solutions we obtained from the SDP solver.

Constraining the solutions space in a superior region also accelerates the algorithm to converge. These advantages can be verified in our experimental results.

F. Overall Flow

From the problem formulation, it is clear that the LDMO and PV-LDMO are both strongly nonconvex problems without analytic structure, which makes them numerically hard to solve. In order to solve the problem, we design a unified optimization flow which includes a numerical optimization flow and a discrete optimization flow. These two engines are collaborative with each other. Basically, the masks are optimized numerically with gradient-based optimization. Once the violation is detected in the printed image, another SDP-based

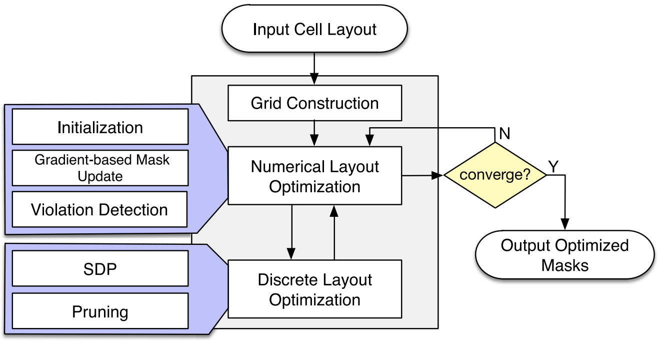


Fig. 9. Overall flow.

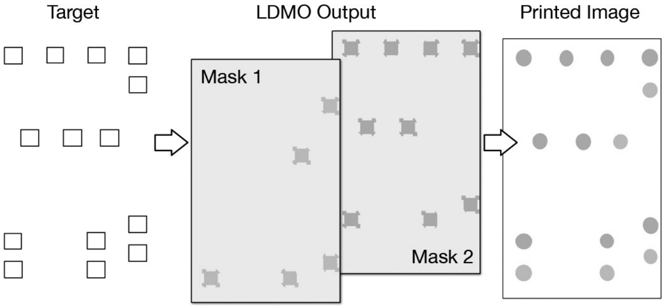


Fig. 10. Example of our LD and MO on cell `OR2_X1`.

discrete optimization engine is triggered to resolve the violations. Multiple solutions are obtained from the solution of SDP, which can help to jump out of local optimum and act as a guidance of numerical optimization. The solutions returned by SDP will be numerically optimized until a pair of masks with the highest quality is selected, which will be further optimized by numerical optimization flow. The overall flow is presented in Fig. 9.

IV. Experimental Results

A. Environment and Implementation Details

We implement our algorithms with C++ on an Intel Core 2.6-GHz Linux machine with 48-GB RAM. To solve SDP we use Csdp, a package for specifying and solving SDP problems [30]. We use an open-source lithography simulator and EPE checker [31], where the intensity threshold is set to 0.039. The EPE violation threshold value is set to 10 nm, which is more strict than the 15 nm used in [31]. The experiments are conducted using NanGate, an open-source standard cell library [32]. We test the proposed unified framework on contact layers where stitches are forbidden. Fig. 10 gives an example of our output masks and the printed images for cell `OR2_X1`. All the solutions obtained are legal in our experiments. We implement an LD engine, where branch-&-bound methodology is applied to search for all legal coloring solutions. In LD, the coloring distance is set to 110 nm, so all contact layers are double patterning friendly. We obtain a modified binary of the MO engine from [23].

B. Results of LDMO

In the first experiment, we compare the proposed framework with an exhaustive optimization flow, where all legal LD solutions are enumerated, and all the solutions are fed into the MO engine [23]. The results of the exhaustive optimization are shown in the merged column "ENUM + [23]" of Table II. The column "#LD" represents the total number of enumerated LD solutions. Considering that it will take extremely long time if we run MO on all LD solutions, we set an upper bound of runtime, which is 36 000 s (i.e., 10 h). The column "#Complete" lists the total number of solutions that have been finished within the runtime limit. Then we can obtain the best decomposed layout with the least EPE violations. The columns "#EPEV" and "RT(s)" list the best EPE violation number and the total MO runtime in seconds. Note that compared to expensive MO, the runtime of LD is usually ignorable. The corresponding results of our unified framework are shown in the merged column "ours." Compared with the exhaustive optimization flow, our unified framework can effectively reduce the EPE violations number by 8×, meanwhile, it can achieve more than 35× speed-up on average.

In order to avoid the unrealistic runtime cost of the exhaustive optimization, heuristic selection methods were proposed in previous work by Yu *et al.* [15] and Chen *et al.* [16]. In the second experiment, we use these two strategies to select from the exhaustive LD solutions, and feed the selected solutions to MO engine of [23]. Then we compare the quality of corresponding printed patterns with ours. The corresponding results are shown in merged columns "[15] + [23]" and "[16] + [23]" in Table II. Here, columns #EPEV and RT (s) represent the EPE violation number and the runtime of MO on the selected LD solutions. From the table we can see that our proposed framework can achieve around 65% and 66% EPE violation reduction compared to the heuristic selection in [15] and [16], respectively. The experimental results show that the density-based LD strategy may not promise an optimal printed image quality after MO.

Compared with the preliminary work [29], new violation graph is constructed as described in Section III-E, which boosts the performance in terms of both image quality and runtime. On average, with the new techniques, LDMO achieves 4× reduction on the number of EPE violations, and more than 2× speed-up.

In addition, we also have explored the how the proposed framework perform under stricter EPE violation threshold conditions. We further use 8-nm and 5-nm as the threshold for determining an EPE violation, and the results are presented in Tables III and IV. It can be observed that the proposed framework can still outperform all the baselines in terms of the quality of the obtained masks and the running time.

For comprehensive comparison among the three flows listed in Table II, we plot the distribution of EPE violations of all enumerated solutions of a cell according to the results of "ENUM+[23]" (see Fig. 11). We can find that different solutions of LD result in diverse EPE cost after MO. The solution obtained by different methods are marked in the figure. It can also be seen that among all the potential solutions, most coloring solutions are actually suboptimal, while methods proposed in [15] and [16] do not select the optimal ones which correspond to the leftmost bar in each chart. Take Fig. 11(a) as

TABLE II
Comparison Between Previous Flow and Unified Flow. EPE$_{\text{THRE}}$ = 10 nm

Cell	ENUM + [23]				[15] + [23]		[16] + [23]		[29]		Ours	
	#LD	#Complete	#EPEV	RT (s)	#EPEV	RT (s)	#EPEV	RT (s)	#EPEV	RT (s)	#EPEV	RT (s)
INV_X1	10	10	1	11814	1	1183	1	1183	1	1995	0	695
NOR2_X1	18	18	2	25117	4	1405	8	1421	1	1996	0	642
BUF_X1	22	22	1	23072	5	1068	5	867	1	1990	0	773
CLKBUF_X1	30	30	1	30486	1	737	1	1046	0	1996	0	890
OAI211_X1	34	34	3	>36000	7	1207	5	1213	6	1996	2	950
AOI211_X1	42	31	3	>36000	8	1080	5	1048	1	1989	1	937
AND2_X1	52	23	3	>36000	13	1061	13	1080	1	1993	0	720
OR2_X1	82	33	0	>36000	3	1220	7	1046	0	1997	0	760
NAND4_X1	86	31	3	>36000	6	1176	5	1173	1	1997	0	809
NAND3_X2	252	35	6	>36000	8	774	7	1171	3	1998	2	680
OR4_X1	1282	33	1	>36000	2	1233	5	1188	0	1987	0	730
NOR3_X2	3016	37	3	>36000	4	1168	5	773	7	1999	3	924
OAI33_X1	6178	40	4	>36000	10	922	11	1017	6	1998	0	811
Average			2.55	>34595.35	6.09	1058.73	6.27	1056.55	2.15	1994.62	0.46	793.92
Ratio			5.54	>43.34	13.23	1.45	13.63	1.42	4.67	2.51	**1.00**	**1.00**

TABLE III
Comparison Between Previous Flow and Unified Flow. EPE$_{\text{THRE}}$ = 8 nm

Cell	ENUM + [23]				[15] + [23]		[16] + [23]		[29]		Ours	
	#LD	#Complete	#EPEV	RT (s)	#EPEV	RT (s)	#EPEV	RT (s)	#EPEV	RT (s)	#EPEV	RT (s)
INV_X1	10	10	3	11865	6	1201	6	1201	4	1994	0	684
NOR2_X1	18	18	6	25217	10	1496	11	1436	4	1993	0	623
BUF_X1	22	22	4	23988	12	1165	9	1063	4	1990	0	706
CLKBUF_X1	30	30	1	30486	1	878	1	936	3	1990	1	846
OAI211_X1	34	34	3	>36000	7	1288	5	1265	7	1999	5	911
AOI211_X1	42	31	3	>36000	13	1195	10	1132	6	1992	4	897
AND2_X1	52	23	6	>36000	17	1123	18	1126	9	1993	3	708
OR2_X1	82	33	2	>36000	4	1309	10	1160	1	1992	1	751
NAND4_X1	86	31	9	>36000	9	1256	13	1221	6	1999	2	792
NAND3_X2	252	35	10	>36000	11	903	15	1209	6	1998	2	659
OR4_X1	1282	33	6	>36000	7	1320	9	1236	1	1987	1	719
NOR3_X2	3016	37	6	>36000	10	1197	11	963	9	1999	4	906
OAI33_X1	6178	40	8	>36000	13	1010	14	1123	5	1998	2	799
Average			5.15	>31965.85	9.23	1180.08	10.15	1059.31	5.00	1994.15	1.92	769.31
Ratio			2.68	>41.55	4.80	1.53	5.28	1.51	2.60	2.59	**1.00**	**1.00**

TABLE IV
Comparison Between Previous Flow and Unified Flow. EPE$_{\text{THRE}}$ = 5 nm

Cell	ENUM + [23]				[15] + [23]		[16] + [23]		[29]		Ours	
	#LD	#Complete	#EPEV	RT (s)	#EPEV	RT (s)	#EPEV	RT (s)	#EPEV	RT (s)	#EPEV	RT (s)
INV_X1	10	10	15	12936	21	1200	21	1201	8	1992	7	674
NOR2_X1	18	18	21	26214	27	1496	22	1436	12	1993	15	652
BUF_X1	22	22	19	25632	29	1153	24	1100	19	1994	7	720
CLKBUF_X1	30	30	18	31995	23	896	23	953	12	1991	4	855
OAI211_X1	34	34	18	>36000	27	1263	27	1250	25	1999	23	898
AOI211_X1	42	31	21	>36000	38	1185	33	1136	21	1999	17	909
AND2_X1	52	23	25	>36000	30	1163	33	1130	21	1999	12	782
OR2_X1	82	33	21	>36000	24	1286	26	1230	15	1998	4	753
NAND4_X1	86	31	32	>36000	38	1263	41	1236	18	1998	10	799
NAND3_X2	252	35	42	>36000	50	889	52	1213	19	1999	13	702
OR4_X1	1282	33	28	>36000	30	1213	34	1250	15	1998	11	730
NOR3_X2	3016	37	31	>36000	38	1163	37	956	18	1997	12	931
OAI33_X1	6178	40	38	>36000	41	1110	52	1020	24	1999	15	821
Average			25.31	>32367.46	32.00	1175.38	32.69	1162.38	17.46	1994.62	11.54	786.62
Ratio			2.19	>41.15	2.77	1.49	2.83	1.48	1.51	2.54	**1.00**	**1.00**

an example. There are 22 DPLD solutions, while the number of EPE violations of these solutions ranges from 1 to 9, among which half of the solutions have five EPE violations, including the solutions selected by methods proposed in [15] and [16]. The masks generated by our method can achieve 0 violation. The same observation can also be found from

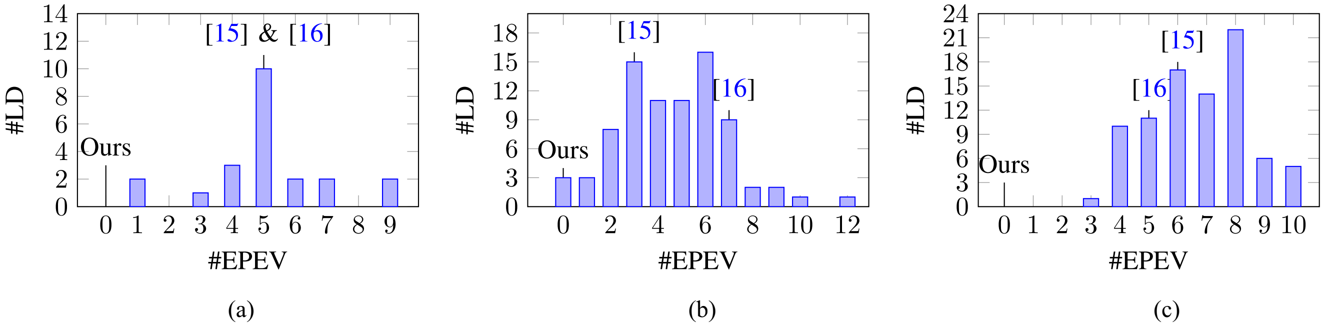


Fig. 11. (a) Distribution of #EPE violations of BUF_X1. (b) Distribution of #EPE violations of OR2_X1. (c) Distribution of #EPE violations of NAND4_X1.

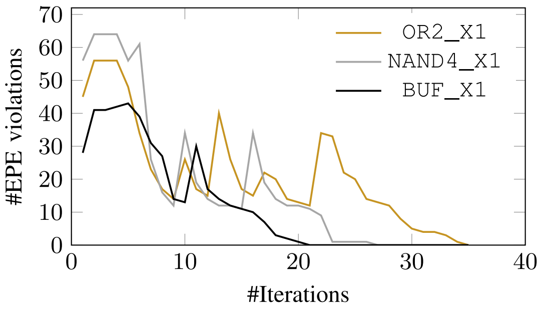


Fig. 12. Convergence of #EPE violations.

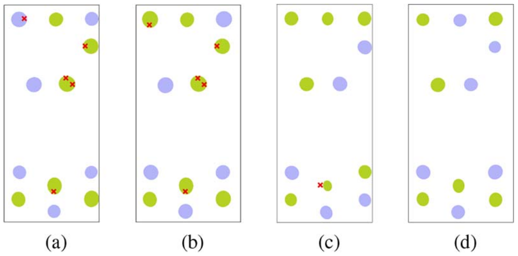


Fig. 13. Printed image of cell BUF_X1. (a) [15] +[23], #EPEV = 5. (b) [16] +[23], #EPEV = 5. (c) [29], #EPEV = 1. (d) Ours, #EPEV = 0.

Fig. 11(c). We can see that the quality of the masks obtained by unified optimization can even outperform all the solutions obtained by conventional two-stage flow.

Fig. 12 demonstrates the convergence of the EPE violation number. Since the optimization process will not get stuck in local optimum, it can be seen that the number of EPE violations goes up on some iterations. Eventually, it will converge to a solution with fewer EPE violations.

Examples of printed images of the contact layers are given in Figs. 13–15 in which the EPE violations are marked with red cross on the pattern. It can be seen more explicitly that the proposed algorithm can find masks with higher quality, which have fewer EPE violations on printed images.

C. Results of PV-LDMO

In practical manufacturing process, the process variation is also a critical issue to be considered. The proposed PV-LDMO flow is designed for this target. Next, we demonstrate the experimental results of the PV-LDMO flow. The statistics are presented in Fig. 16–18. In the figures, "NG" and "PV" denote new graph construction and process variation-aware optimization, respectively. We use the fundamental LDMO flow proposed by the preliminary work [29] as baseline. It can be noted that by considering the new PV Band optimization objective and applying new layout representation graph, the number of EPE violations and the PV Band area are reduced

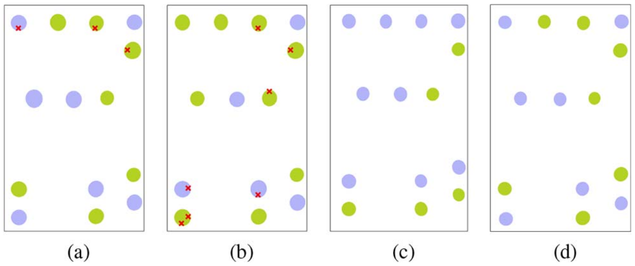


Fig. 14. Printed image of cell OR2_X1. (a) [15] +[23], #EPEV = 3. (b) [16] +[23], #EPEV = 7. (c) [29], #EPEV = 0. (d) Ours, #EPEV = 0.

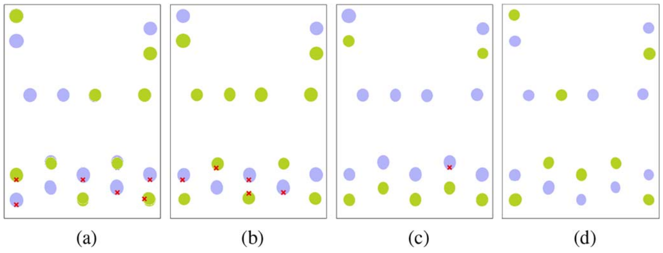


Fig. 15. Printed image of cell NAND4_X1. (a) [15] +[23], #EPEV = 6. (b) [16] +[23], #EPEV = 5. (c) [29], #EPEV = 1. (d) Ours, #EPEV = 0.

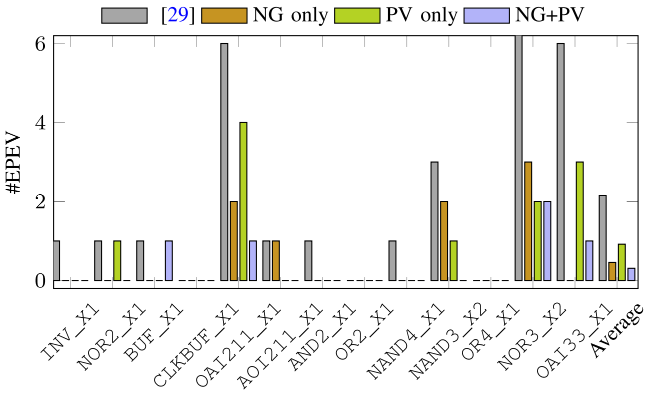


Fig. 16. Comparison on the number of EPE violations.

TABLE V
RESULTS RECORD OF MULTIPLE EXPERIMENTAL RUNS

Cell	1		2		3		4		5		Average	
	EPEV #	PVB (nm^2)	EPEV #	PVB (nm^2)	EPEV #	PVB (nm^2)	EPEV #	PVB (nm^2)	EPEV #	PVB (nm^2)	EPEV #	PVB (nm^2)
INV_X1	0	14247	0	13944	0	13956	0	14332	0	14392	0.0	14174
NOR2_X1	0	19278	1	18667	0	19278	1	18802	0	19289	0.4	19063
BUF_X1	0	19070	1	19011	0	18917	0	19023	0	19064	0.2	19017
CLKBUF_X1	0	17139	0	17936	0	17140	0	17633	0	16949	0.0	17359
OAI211_X1	1	31038	0	31214	1	31333	0	31292	2	31075	0.8	31190
AOI211_X1	0	30455	1	30920	0	30478	0	30386	0	30936	0.2	30635
AND2_X1	0	23493	0	23590	4	22780	0	23468	0	23402	0.8	23347
OR2_X1	0	22327	0	22038	0	22505	0	22327	1	22641	0.2	22368
NAND4_X1	0	29235	1	28607	0	28524	0	28359	1	28430	0.4	28667
NAND3_X2	0	36531	0	36753	0	36766	1	36255	0	36226	0.2	36506
OR4_X1	0	28398	0	28569	0	28972	0	28639	0	28644	0.0	28644
NOR3_X2	0	30969	2	31457	2	31272	0	31435	3	31574	1.4	31341
OAI33_X1	0	33295	0	33460	0	33363	0	33704	1	33464	0.2	33583

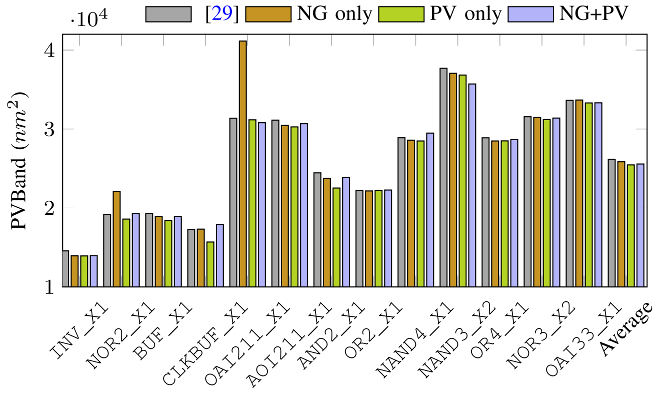


Fig. 17. Comparison on PV band.

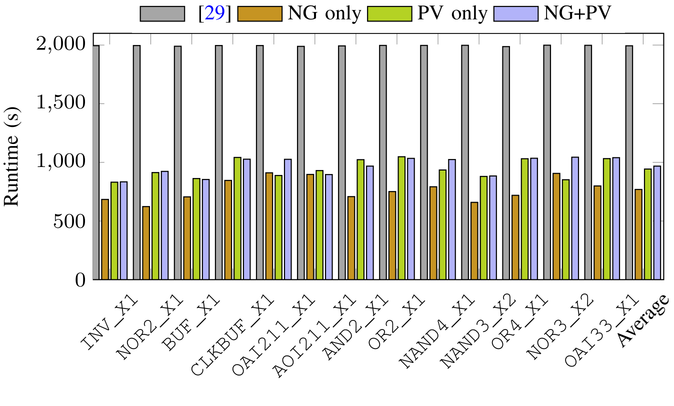


Fig. 18. Comparison on runtime.

significantly. Compared with the preliminary work [29], the number of EPE violation is reduced by 84%, while the PV Band area is improved by 2%. What is more, the new flow achieves 2× speedup on runtime.

We verify the effectiveness of each new presented technique of the PV-LDMO flow by conducting comprehensive ablation study. To do so, each newly designed optimization technique is enabled separately to justify the benefit of its own. The results can also be observed in Figs. 16–18. It is shown that applying only new graph construction will result in the shortest runtime and smaller number of EPE violations. Solely applying PV Band optimization will result in the smallest PV Band area, while the number of EPE violation will increase. Leveraging both of them can take the good side of each and lead to satisfying results on all aspects.

D. Robustness and Pattern Density

In the experiment, we use a straight-forward method for initialization. $\boldsymbol{M}_1$ is initialized with target image $\boldsymbol{Z}$ and $\boldsymbol{M}_2$ is initialized as a 0 matrix (empty mask), which is a conventional initialization approach for MO [19], [23]. The decomposition/assignment happens in the discrete optimization. Different initial assignments may be obtained at the first discrete optimization process due to the randomized rounding and pruning, which may lead to different final solutions. In order to analyze the sensitivity of the initial decomposition to the proposed framework, we conduct multiple runs and record all the results which is shown as in Table V. It can be seen that there are slight vibration among different runs while the overall results is robust to the randomness.

In double patterning process, the pattern density uniformity between two masks is of great importance in manufacturing. The final on-wafer image is generated by etching process which is sensitive to the pattern density. Since the proposed framework is targeting at simultaneous LD and MO in multiple patterning process, pattern density issue should be taken into consideration. It can be realized in two ways: 1) pruning solutions based on a combined metric considering both printed image quality and pattern density uniformity (e.g., a weighted summation of these two metrics) and 2) set a hard constraint on the uniformity and directly pruned unsatisfied ones. To verify the idea, we implement a uniformity-aware discrete optimization and conducted the experiments. Since in our experiments the total regions of the two masks are the same, we use the pattern area on each mask to represent the pattern density. Denote the pattern density on the two masks as D_1 and D_2, respectively. They can be calculated using $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$, or can be approximately calculated using the solution (23). The constraint is set to be $|D_1 - D_2| \leq 20\% \times \mathcal{D}$, where $\mathcal{D}$ is

TABLE VI
PERFORMANCE OF PATTERN UNIFORMITY-AWARE LDMO

Cell	NG only		PV only		NG+PV	
	EPEV #	PVB (nm^2)	EPEV #	PVB (nm^2)	EPEV #	PVB (nm^2)
INV_X1	0	14189	0	13946	0	13951
NOR2_X1	0	19253	0	18742	0	18987
BUF_X1	0	19142	0	19055	0	18996
CLKBUF_X1	0	17742	0	17296	0	16992
OAI211_X1	1	31354	1	30897	1	31244
AOI211_X1	1	30561	2	30496	0	30478
AND2_X1	0	23861	0	23869	2	23798
OR2_X1	0	22478	0	22190	0	22148
NAND4_X1	0	28429	4	26795	0	28210
NAND3_X2	1	37232	1	35931	0	35894
OR4_X1	0	28478	1	28173	0	28403
NOR3_X2	3	31280	2	31096	2	30703
OAI33_X1	1	33467	0	33282	1	33472
Average	0.54	26805	0.85	25520	0.46	25636
Ratio	1.17	1.04	1.85	0.99	**1.0**	**1.0**

the total area of the target patterns. The results are shown in Table VI after applying the uniformity-aware constraint.

V. DISCUSSION

A. Handling Variable Threshold Resist Model

The relaxation approach applied in (7) and (8) assumes a constant threshold model (CTM) when forming a printed image from an aerial image. The proposed Algorithms 1 and 2 are also applicable to the scenario where a variable threshold model (VTM) is used. Recall that CTM is relaxed using sigmoid function to incorporate with numerical optimization [(7) and (8)]. Similar approaches can be used if the resist model is VTM. Essentially, we just need to obtain pixel-wise threshold before binarizing the aerial image to printed image. Suppose a VTM is given as $I_{\text{vth}} = C_1 + C_2 \times I_{\max}$ [33], where $I_{\max}$ is the maximum intensity of a local region of aerial image, I_{vth} is the threshold of the pixels within the same region. C_1 and C_2 are the parameters in the model. After aerial images $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ are generated, two threshold matrices $\boldsymbol{I}_{vth1}$ and $\boldsymbol{I}_{vth2}$ can be obtained, which have the same dimension as the corresponding aerial images. The entries in $\boldsymbol{I}_{vth1}$ and $\boldsymbol{I}_{vth2}$ indicate the threshold of the corresponding pixels in $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$, respectively. Then (7) and (8) can be written as

$$\boldsymbol{Z}_1(x,y) = \frac{1}{1+\exp\left[-\theta_Z(\boldsymbol{I}_1(x,y) - \boldsymbol{I}_{vth1}(x,y))\right]} \tag{24}$$

$$\boldsymbol{Z}_2(x,y) = \frac{1}{1+\exp\left[-\theta_Z(\boldsymbol{I}_2(x,y) - \boldsymbol{I}_{vth2}(x,y))\right]}. \tag{25}$$

Considering the $\boldsymbol{I}_{vth1}$ and $\boldsymbol{I}_{vth2}$ are only correlated with the maximum local intensity in $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$, thus, they can be treated as constants when deriving the gradient based on above equations, which makes the gradient calculation the same as (17) and (18).

B. Handling More Advanced Multiple Patterning Lithography

The proposed framework contains a gradient-based numerical optimization, as well as a set of discrete optimization. The main step for gradient-based numerical optimization is the mathematical relaxation and derivation of the gradient for each individual mask. For triple patterning process, the problem formulation can be rewritten with an additional variable $\boldsymbol{M}_3$ and additional constraints

$$\boldsymbol{M}_3(x,y) \in \{0,1\} \quad \forall x,y \tag{26}$$

$$\boldsymbol{I}_3 = \sum_{k=1}^{K} w_k \cdot |\boldsymbol{M}_3 \otimes \boldsymbol{h}_k|^2 \tag{27}$$

$$\boldsymbol{Z} = f_{\text{resist}}(\boldsymbol{I}_1) \vee f_{\text{resist}}(\boldsymbol{I}_2) \vee f_{\text{resist}}(\boldsymbol{I}_3). \tag{28}$$

It can be observed that the gradient derivation is not limited to only 2 masks, since all the mathematical relaxation methods can be applied to the newly added variables and constraints, and a similar equation can be derived for $\boldsymbol{M}_3$. The discrete optimization requires a slightly change to handle triple patterning lithography cases which is essentially a three-way partition. For double patterning process, the problem is formulated into a max-cut problem and relaxed to an SDP formulation. For triple patterning process, discrete optimization solutions can be generated based on vector programming which also can be relaxed to an SDP formulation, as proposed in [5] and [34]. Considering that there may be some patterns that have multiple assignment choices, we can also obtain multiple solutions from the three-way partition step. Similarly, the pruning step is applied to select the most promising solution containing a 3-tuple, and proceed as shown in Fig. 9.

C. Handling Other Layers

In addition, although the experiments are conducted with contacted layers in this article, the optimization methodologies proposed in this framework are general. Specifically, the gradient-based numerical optimization is a conventional technique which is generally applicable to any layers. The discrete optimization is designed to assist the conflicting pattern separation and help to explore higher quality solutions, therefore, the fundamental idea is applicable to other layers. In order to perform LDMO for other layers, there are a few detailed modifications can be applied to some of the submodules. Particularly, the grid construction needs to incorporate other issues like stitch insertion, which essentially requires more engineering efforts for stitch insertion [3], [5] and grid merging, such that the violation detection, discrete optimization and pruning can still perform the functionality as they are.

VI. CONCLUSION

In this article, we have proposed a unified framework solving LD and MO problem, while taking process variation issues into consideration. In this framework, we designed two collaborative flows for optimization: a gradient-based numerical optimization, as well as a set of discrete optimizations to jump out of local optimum. The experimental results show that our proposed framework outperforms conventional flow in terms of both runtime and EPE violation number. To the best of our knowledge, this is the first work trying to handle multiple patterning LD and MO simultaneously. Note that our framework is general and it can be extended to handle triple or quadruple patterning lithography coloring rules. We hope this article can stimulate more future work into this field.

ACKNOWLEDGMENT

The authors would like to thank Jing Su, Chenxi Lin, Yi Zou from ASML, and anonymous reviewers for helpful comments.

REFERENCES

[1] D. Z. Pan, B. Yu, and J.-R. Gao, "Design for manufacturing with emerging nanolithography," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 10, pp. 1453–1472, Oct. 2013.

[2] B. Yu, X. Xu, S. Roy, Y. Lin, J. Ou, and D. Z. Pan, "Design for manufacturability and reliability in extreme-scaling VLSI," *Sci. China Inf. Sci.*, vol. 59, pp. 1–23, May 2016.

[3] A. B. Kahng, C.-H. Park, X. Xu, and H. Yao, "Layout decomposition approaches for double patterning lithography," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 29, no. 6, pp. 939–952, Jun. 2010.

[4] K. Yuan, J.-S. Yang, and D. Z. Pan, "Double patterning layout decomposition for simultaneous conflict and stitch minimization," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 29, no. 2, pp. 185–196, Feb. 2010.

[5] B. Yu, K. Yuan, D. Ding, and D. Z. Pan, "Layout decomposition for triple patterning lithography," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 3, pp. 433–446, Mar. 2015.

[6] Y. Xu and C. Chu, "GREMA: Graph reduction based efficient mask assignment for double patterning technology," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, 2009, pp. 601–606.

[7] X. Tang and M. Cho, "Optimal layout decomposition for double patterning technology," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, 2011, pp. 9–13.

[8] B. Yu and D. Z. Pan, "Layout decomposition for quadruple patterning lithography and beyond," in *Proc. ACM/IEEE Design Autom. Conf. (DAC)*, 2014, pp. 1–6.

[9] S.-Y. Fang, Y.-W. Chang, and W.-Y. Chen, "A novel layout decomposition algorithm for triple patterning lithography," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 33, no. 3, pp. 397–408, Mar. 2014.

[10] H. Tian, H. Zhang, Q. Ma, Z. Xiao, and M. D. F. Wong, "A polynomial time triple patterning algorithm for cell based row-structure layout," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, 2012, pp. 57–64.

[11] Y. Lin, X. Xu, B. Yu, R. Baldick, and D. Z. Pan, "Triple/quadruple patterning layout decomposition via novel linear programming and iterative rounding," in *Proc. SPIE Adv. Lithography*, vol. 9781, 2016, Art. no. 97810M.

[12] J. Kuang and E. F. Y. Young, "An efficient layout decomposition approach for triple patterning lithography," in *Proc. ACM/IEEE Design Autom. Conf. (DAC)*, 2013, pp. 1–6.

[13] H.-Y. Chang and I. H.-R. Jiang, "Multiple patterning layout decomposition considering complex coloring rules," in *Proc. ACM/IEEE Design Autom. Conf. (DAC)*, 2016, pp. 1–6.

[14] J. Kuang and E. F. Y. Young, "Fixed-parameter tractable algorithms for optimal layout decomposition and beyond," in *Proc. ACM/IEEE Design Autom. Conf. (DAC)*, 2017, pp. 1–6.

[15] B. Yu, Y.-H. Lin, G. Luk-Pat, D. Ding, K. Lucas, and D. Z. Pan, "A high-performance triple patterning layout decomposer with balanced density," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, 2013, pp. 163–169.

[16] Z. Chen, H. Yao, and Y. Cai, "SUALD: Spacing uniformity-aware layout decomposition in triple patterning lithography," in *Proc. IEEE Int. Symp. Qual. Electron. Design (ISQED)*, 2013, pp. 566–571.

[17] J.-S. Park *et al.*, "An efficient rule-based OPC approach using a DRC tool for 0.18 μm ASIC," in *Proc. IEEE Int. Symp. Qual. Electron. Design (ISQED)*, 2000, pp. 81–85.

[18] A. Awad, A. Takahashi, S. Tanaka, and C. Kodama, "A fast process variation and pattern fidelity aware mask optimization algorithm," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, 2014, pp. 238–245.

[19] J. Kuang, W.-K. Chow, and E. F. Y. Young, "A robust approach for process variation aware mask optimization," in *Proc. IEEE/ACM Design Autom. Test Europe (DATE)*, 2015, pp. 1591–1594.

[20] Y.-H. Su, Y.-C. Huang, L.-C. Tsai, Y.-W. Chang, and S. Banerjee, "Fast lithographic mask optimization considering process variation," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, no. 8, pp. 1345–1357, Aug. 2016.

[21] A. Poonawala and P. Milanfar, "Mask design for optical microlithography—An inverse imaging problem," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 774–788, Mar. 2007.

[22] N. Jia and E. Y. Lam, "Machine learning for inverse lithography: Using stochastic gradient descent for robust photomask synthesis," *J. Opt.*, vol. 12, no. 4, pp. 1–9, 2010.

[23] J.-R. Gao, X. Xu, B. Yu, and D. Z. Pan, "MOSAIC: Mask optimizing solution with process window aware inverse correction," in *Proc. ACM/IEEE Design Autom. Conf. (DAC)*, 2014, pp. 1–6.

[24] A. Poonawala and P. Milanfar, "Double-exposure mask synthesis using inverse lithography," *J. Micro/Nanolithography MEMS MOEMS*, vol. 6, no. 4, 2007, Art. no. 043001.

[25] X. Li, G. Luk-Pat, C. Cork, L. Barnes, and K. Lucas, "Double-patterning-friendly OPC," in *Proc. SPIE*, vol. 7274, 2009, Art. no. 727414.

[26] S. Banerjee, K. B. Agarwal, and M. Orshansky, "Simultaneous OPC and decomposition for double exposure lithography," in *Proc. SPIE*, vol. 7973, 2011, Art. no. 79730E.

[27] N. B. Cobb, "Fast optical and process proximity correction algorithms for integrated circuit manufacturing," Ph.D. dissertation, Elect. Eng. Comput. Sci., Univ. California at Berkeley, Berkeley, CA, USA, 1998.

[28] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *J. ACM*, vol. 42, no. 6, pp. 1115–1145, 1995.

[29] Y. Ma, J.-R. Gao, J. Kuang, J. Miao, and B. Yu, "A unified framework for simultaneous layout decomposition and mask optimization," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, 2017, pp. 81–88.

[30] B. Borchers, "CSDP, a C library for semidefinite programming," *Optim. Methods Softw.*, vol. 11, nos. 1–4, pp. 613–623, 1999.

[31] S. Banerjee, Z. Li, and S. R. Nassif, "ICCAD-2013 CAD contest in mask optimization and benchmark suite," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, 2013, pp. 271–274.

[32] (2008). *NanGate FreePDK45 Generic Open Cell Library*. [Online]. Available: http://www.si2.org/openeda.si2.org/projects/nangatelib

[33] J. Randall, K. G. Ronse, T. Marschner, A.-M. Goethals, and M. Ercken, "Variable-threshold resist models for lithography simulation," in *Proc. Opt. Microlithography XII*, vol. 3679, 1999, pp. 176–182.

[34] B. Yu, K. Yuan, B. Zhang, D. Ding, and D. Z. Pan, "Layout decomposition for triple patterning lithography," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, 2011, pp. 1–8.

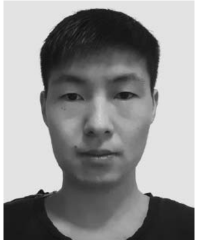
Yuzhe Ma received the B.E. degree from the Department of Microelectronics, Sun Yat-sen University, Guangzhou, China, in 2016. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong.

He has Interned with Cadence Design Systems, San Jose, CA, USA, NVIDIA Research, Austin, TX, USA, and Tencent YouTu X-Lab, Shenzhen, China. His research interests include VLSI design for manufacturing, physical design, and machine learning on chips.

Mr. Ma received Best Student Paper Award from ICTAI in 2019, the Best Paper Award Nomination from ASPDAC in 2019, and the Best Poster Research Award from Student Research Forum of ASPDAC in 2020.

Wei Zhong received the B.S. degree from the Dalian University of Technology, Dalian, China, in 2008, and the M.S. and Ph.D. degrees from Waseda University, Tokyo, Japan, in 2010 and 2014, respectively.

He served as a Research Assistant with the Information, Production and Systems Research Center, Waseda University from 2010 to 2011, an Associate Specialist with the Central Research Laboratory of Ricoh Company, Tokyo, from 2011 to 2014, and a Chief Designer and the Director of the Institute of Image Processing Technology, State Key Laboratory of Digital Multimedia Technology, Hisense Group, Qingdao, China, from 2014 to 2018. From 2015 to 2017, he also served as a Postdoctoral Researcher with the University of Science and Technology of China, Hefei, China. He is currently an Associate Professor with the International School of Information Science and Engineering, Dalian University of Technology. His research interests include several aspects of computer vision and image processing algorithms, VLSI design automation, networks on chips, and hardware-software co-design of embedded systems.



Shuxiang Hu received the B.E. degree from the Department of Software, Hefei University of Technology, Hefei, China. He is currently pursuing the M.S. degree with the International School of Information Science and Engineering, Dalian University of Technology, Dalian, China.

His research interests include VLSI automation and deep learning on chips.

Jhih-Rong Gao received the B.S. and M.S. degrees in computer science from National Tsing Hua University, Hsinchu, Taiwan, in 2005 and 2007, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Texas at Austin, Austin, TX, USA, in 2014.

She was an Research and Development Engineer with Synopsys, Inc., Hsinchu, from 2007 to 2009. In 2014, she joined Cadence Design Systems in Austin, Austin, where she is a Principle Software Engineer working on improving the algorithms and interactions for placement, routing, and clock tree synthesis.

Dr. Gao was awarded BACUS Photomask Scholarship from SPIE in 2013.

Jian Kuang received the B.E. degree from Sun Yat-sen University, Guangzhou, China, in 2012, and the Ph.D. degree in computer science and engineering from the Chinese University of Hong Kong, Hong Kong, in 2016.

He is currently with Cadence Design Systems, San Jose, CA, USA. His current research interests include VLSI computer-aided design, physical design automation, and design for manufacturability.

Jin Miao received the B.S. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 2010, and the Ph.D. degree in electrical and computer engineering from the University of Texas at Austin, Austin, TX, USA, in 2014.

He is currently a Principle Software Engineer with Cadence Design Systems, San Jose, CA, USA. He has broad interests in emerging science and technologies, including approximate computing, hardware security, and machine learning.

Dr. Miao has been serving as a Reviewer or a TPC Member for a number of journals or conferences, including TC, TCAD, TVLSI, DAC, ASPDAC, and NEWCAS.

Bei Yu (Member, IEEE) received the Ph.D. degree from the University of Texas at Austin, Austin, TX, USA, in 2014.

He is currently an Assistant Professor with the Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong.

Dr. Yu received Six Best Paper Awards from ICTAI 2019, Integration, the VLSI Journal in 2018, ISPD 2017, the SPIE Advanced Lithography Conference 2016, ICCAD 2013, ASPDAC 2012, and Five ICCAD/ISPD Contest Awards. He is an Editor of IEEE TCCPS Newsletter. He has served as the TPC Chair of ACM/IEEE Workshop on Machine Learning for CAD, and in many journal editorial boards and conference committees.