CTM-SRAF: Continuous Transmission Mask-Based Constraint-Aware Subresolution Assist Feature Generation

Ziyang Yu, Peiyu Liao, Yuzhe Ma, *Member, IEEE*, Bei Yu, *Senior Member, IEEE*, and Martin D. F. Wong, *Fellow, IEEE*

***Abstract*—In the lithography process, subresolution assist features (SRAFs), as an essential resolution enhancement technique (RET), is applied to improve the pattern fidelity and enlarge the process window. In this article, we propose a robust constraint-aware SRAF generation method based on continuous transmission mask (CTM). The intensity distribution on the CTM is extracted to guide the SRAF generation. The SRAF insertion also honors the design rules, which is formulated as integer programming with quadratic constraints and solved by a fast yet efficient algorithm. A fast probe-based SRAF evolution method is proposed to determine the shapes of SRAFs. The effectiveness and efficiency are demonstrated based on the experimental results.**

***Index Terms*—Continuous transmission mask (CTM), design for manufacturability, subresolution assist feature (SRAF) generation.**

I. Introduction

In the past decades, the feature size of the integrated circuit has been continuously scaled in accordance with Moore's law. The optical lithography has entered the low k-1 regime [1], [2], and the wavelength of light used has remained 193 nm. As a result, it becomes more and more challenging to acquire high pattern fidelity and mask printability using the traditional lithography process. Besides, the printed wafer image becomes highly sensitive to minor variations of the lithography conditions. To mitigate these issues, the requirements for resolution enhancement techniques (RETs) in optical lithography become stricter [3], [4].

One of the most widely adopted RETs is optical proximity correction (OPC) [5], [6], [7], [8], [9]. In traditional OPC, the lithography mask is predistorted for main patterns to compensate for the undesired distortion of printed wafer images. However, as the critical size shrinks and the target patterns become more complex, using OPC alone is hard to acquire a satisfying printed image under adequate process windows. Subresolution assist feature (SRAF) [10], as a particularly significant RET, can be applied. The SRAFs are small shapes inserted in specific regions around the main patterns. They enhance the contrast of the aerial images by providing supplementary spatial frequency components to the main features without being printed themselves. In order to comply with this constraint, mask manufacturing rule checks (MRCs) are leveraged to guide the SRAF insertion, which imposes requirements on the size of each SRAF pattern and the insertion location.

In general, the insertion location and geometrical features can affect the quality of the SRAFs significantly. Therefore, a wide range of SRAF insertion methods were explored these years, which can be divided into three categories: 1) rule-based SRAF [2], [11], [12], [13]; 2) model-based SRAF [14], [15], [16], [17], [18], [19]; and 3) machine-learning-based SRAF [20], [21], [22]. In the rule-based SRAF method, a set of predefined rules govern the shapes, sizes, and positions of the SRAFs. This kind of method is efficient and effective for simple design patterns. However, the SRAF rules vary significantly under different lithography process conditions, and they heavily depend on the engineer's expertise. The rule table is cumbersome and maintains prohibitive when faced with complex design targets. Model-based SRAF methods generate the SRAF using different computational lithography models as guidance. These methods can get rid of the demand for human experience and attain a high accuracy even with complex patterns, at the cost of long computational time. Very recently, powerful machine-learning techniques have been introduced to solve SRAF insertion problems. Machine-learning-based SRAF methods train models using features extracted from historical results. For a new target pattern, the probability of SRAF existence on each site is predicted, based on which the SRAFs are generated and then simplified. This technique demonstrates its high computational efficiency. Nevertheless, the performance highly depends on the generality of the training set and feature engineering. Besides, the learning model needs to be retrained when targeting designs under different layers or processes.

Most of the methods mentioned above generate the SRAFs using the binary mask information. In the binary mask, the value on each grid site can only be chosen from either 0 or 1, representing the blockage or transparency of this position. Continuous transmission mask (CTM) [23], as widely adopted in source-mask optimization (SMO) [24], [25], [26],

Manuscript received 19 July 2022; revised 17 November 2022; accepted 31 December 2022. Date of publication 25 January 2023; date of current version 20 September 2023. This work was supported in part by the Research Grants Council of Hong Kong, SAR, under Project CUHK14208021. This article was recommended by Associate Editor L. Behjat. *(Corresponding author: Ziyang Yu.)*

Ziyang Yu, Peiyu Liao, Bei Yu, and Martin D. F. Wong are with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, SAR (e-mail: yuziyang96@gmail.com).

Yuzhe Ma is with the Microelectronics Thrust, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou 511458, China.

Digital Object Identifier 10.1109/TCAD.2023.3239559

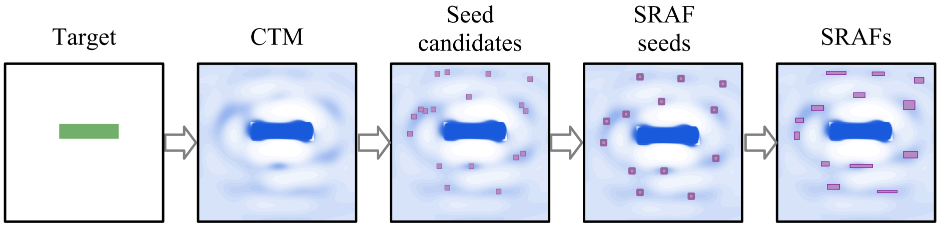

Fig. 1. Workflow of the proposed SRAF generation.

contains more information that can be used to guide the SRAF insertion. In the CTM, a floating intensity value is assigned on each site, representing the degree of transmissivity. The site is blocked if the value is 0 and transparent if the value reaches the maximum. The intensity leverages more information than the binary value, which could guide the SRAF insertion. The previous work using CTM generates complex SRAF shapes, which requires additional simplification post-process to meet the mask manufacturing rules [27].

However, the information contained in CTM can be more effectively extracted. In this article, we propose a robust CTM-based constraint-aware SRAF insertion method. As shown in Fig. 1, the SRAF generation process is mainly composed of three stages: 1) CTM generation; 2) SRAF seed positioning; and 3) SRAF shape evolution. For a given target, the CTM is generated by a relaxed inverse lithography technique (ILT) algorithm. Based on the intensity distribution on CTM, the SRAF seed candidates are roughly selected as the local maximum regions, and the SRAF seeds are selected from the candidates with highest priorities without violating the SRAF spacing rules. Although seed selection is an NP-hard problem, we formulate the problem into a quadratic program and solve it efficiently. The SRAFs are then evolved to desired shapes based on the selected seeds under the guidance of CTM. The generated SRAFs can be co-optimized with the main patterns through OPC to generate better lithography masks. Our method also has complementary advantages to machine-learning-based SRAF methods in the aspect of generating high-quality training data.

The major contributions of the proposed method are summarized as follows.

1) We generate the CTM with relaxed ILT and extract the latent information to guide the SRAF information.
2) We formulate the SRAF seed selection problem as an optimization problem considering the location and distance constraints and provide a novel efficient algorithm to solve it.
3) We provide a fast probe-based SRAF shape evolution method to consider the SRAF size constraints.
4) We co-optimize the generated SRAF with main target pattern with level set-based ILT method to evaluate the effectiveness of SRAF.
5) We perform experiments on ICCAD 2013 contest benchmarks, and the performance is superior to recent state-of-the-art OPC methods.

The remainder of this article is organized as follows. In Section II, we give the description of the problem. In Section III, we explain the algorithms in detail. The experimental results are shown in Section IV, followed by the conclusion in Section V.

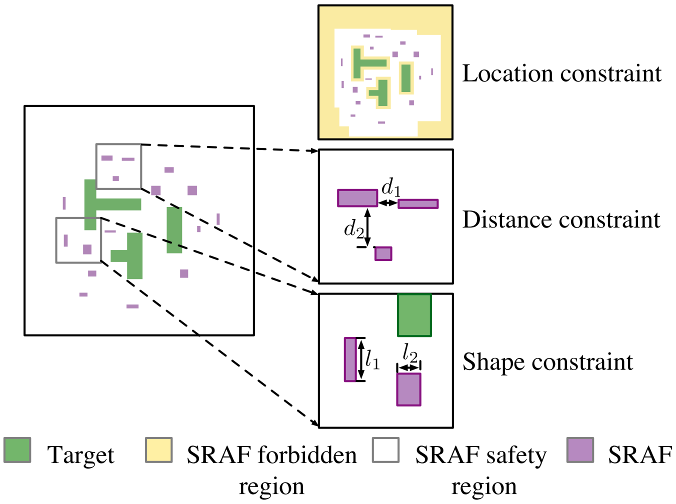

Fig. 2. Constraints-aware SRAFs with target shapes and three types of constraints.

II. Problem Formulation

Compared with the mask containing only main features, the SRAFs inserted around the main features increase the light propagated through the mask and introduce additional spatial frequency components into the imaging process. The reason behind the resolution enhancement brought by SRAFs is, by adequately choosing the additional transparent pitch, the transmitted light has the same phase as the light passing through the main feature pattern [28]. In a CTM, the value on each site represents the degree of transmissivity of the site. The SRAFs are better at covering where the degree of transmissivity is higher.

However, each inserted SRAF needs to obey several constraints, as are shown in Fig. 2. The location constraint says the SRAFs can only locate in the safety region. SRAF cannot be too far away from the main pattern so that the lithography interaction between the SRAF and the main pattern will lead to resolution enhancement; neither can SRAF be too close to the main pattern, in case the SRAF is considered as part of the main pattern and be printed out on the wafer plane. The distance constraint makes sure the distance between different SRAFs is larger than the threshold. The shape constraint controls the size of each SRAF pattern.

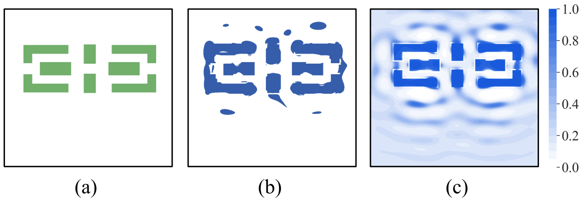

Fig. 3. (a) Target image. (b) Binary mask optimized by pixel-based ILT. (c) Generated CTM, in which on each pixel the value ranges from 0 to 1.

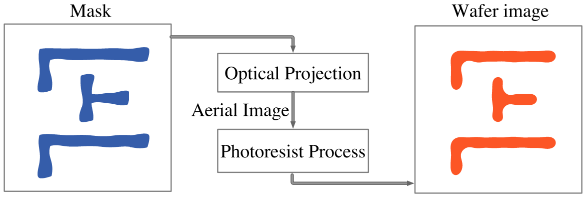

Fig. 4. Schematic of the optical lithography simulator.

Problem 1 (SRAF Generation): Given a specific lithography system and target images, the photomasks are optimized to generate high-quality printed images. The objective of SRAF generation is to insert SRAFs onto the mask plane with decided positions and shapes meanwhile not violating the SRAF constraints. With the help of the added SRAFs, mask optimization through OPC could be further refined, generating better-printed wafer images.

III. SRAF Generation Algorithm

In this section, we discuss our CTM-based constraint-aware SRAF generation method in detail. The overview of our proposed SRAF generation flow is shown in Fig. 1. The SRAF generation is strongly dependent on the lithography process condition. In the first step, the target pattern is transformed into CTM by the relaxed ILT method within a few iterations. The effect of the lithography process conditions can be considered in CTM generation. By utilizing the intensity distribution on CTM, the SRAF seed candidates can be initiated and placed on the mask plane. In case of violating the SRAF design rules, the SRAF seeds are sophisticatedly selected from all the candidates. In the final step, the SRAFs are evolved automatically into desired sizes and shapes based on these SRAF seeds and CTM.

A. Continuous Transmission Mask

With carefully designed SRAFs, the frequency-domain distribution of electric field amplitude for main patterns can be enlarged, contributing to higher wafer fidelity. However, determining the size and locations of SRAF is a dilemma. On the one hand, we want the density of SRAFs to be as large as possible so that more high-frequency diffraction components can be involved to better compensate for mask distortion. On the other hand, the density of SRAFs cannot be too large in case of being printed out on the wafer. The practical effects of different SRAFs lead to different priorities, which can be decided with the guide of ILT, in the form of CTM [28].

Fig. 3 shows an example of a target pattern, the optimized binary mask and the generated CTM. Different from the binary mask in which each location only takes discrete values, e.g., 0 and 1, representing the complete blockage and transparency of the site. The value on each site in a CTM can be a continuum, ranging between a minimum value representing the blockage and a maximum value representing the transparency. Other values between these two extreme numbers can be deemed to be the degree of transmissivity. Even though a CTM cannot be used directly in the practical manufacturing process, it contains more information than the binary mask. SRAF seeds are extracted from CTM where transmissivity is higher, so that the light passing through SRAFs has the same phase as the light passing through the main pattern and, thus, enhances the resolution.

Fig. 4 illustrates the diagram of the optical lithography system. In the lithography system, the incident light is transmitted through the photomask and optical projector successively onto the photoresist material, forming the aerial image. The Kth approximation of light intensity distribution of the aerial image can be represented using the Hopkins diffraction theory

$$\boldsymbol{I}(x, y) = \sum_{k=1}^{K} \mu_k |\boldsymbol{h}_k(x, y) \otimes \boldsymbol{M}(x, y)|^2 \tag{1}$$

where x and y are 2-D coordinates, and $\boldsymbol{h}_k$ is the optical kernel function.

If the light intensity of aerial image $\boldsymbol{I}(x, y)$ is larger than a threshold I_{th}, the photoresist material will develop, leaving the shape printed image R on the wafer plane. This photoresist process can be expressed as a constant threshold model: $\boldsymbol{R}(x, y) = 1$ if $\boldsymbol{I}(x, y) \geq I_{th}$; otherwise, $\boldsymbol{R}(x, y) = 0$. In practical implementation, the discontinuous constant threshold model is approximated by a sigmoid function for deriving the partial difference

$$\boldsymbol{R}(\boldsymbol{M}) = \mathrm{sig}(\boldsymbol{I}) = \mathrm{sig}\left[\sum_{k=1}^{K} \mu_k |\boldsymbol{h}_k \otimes \boldsymbol{M}|^2\right]. \tag{2}$$

In pixel-based ILT mask optimization flow, the mask $\boldsymbol{M}$ is updated iteratively to minimize the cost function $L(\boldsymbol{M})$, which is the linear combination of nominal conditioned pattern deviation and process variation band (PVB) evaluation

$$L(\boldsymbol{M}) = \left|\left|\boldsymbol{R}(\boldsymbol{M}) - \boldsymbol{R}^*(\boldsymbol{M})\right|\right|_{\text{nom}}^2 + \alpha \sum_{k=1}^{N_p} ||\boldsymbol{R}_k(\boldsymbol{M}) - \boldsymbol{R}^*(\boldsymbol{M})||^2 \tag{3}$$

where α is the weight coefficient, N_p is the number of lithography process conditions, and $|| \cdot ||$ represent the L2 norm. In order to apply gradient descent without boundary constraint, the mask is relaxed using unconstrained continuous variable $\boldsymbol{P}(x, y)$ and a smooth sigmoid transformation with relative small steepness $\theta_P = 1$

$$\boldsymbol{M} = \mathrm{sig}(\boldsymbol{P}) = \frac{1}{1 + e^{-\theta_P \cdot \boldsymbol{P}}}. \tag{4}$$

In every iteration, $\boldsymbol{P}(x, y)$ is updated using the gradient of cost function $L(\boldsymbol{M})$

$$\boldsymbol{P}' = \boldsymbol{P} - \Delta t \frac{\partial L(\boldsymbol{M})}{\partial \boldsymbol{P}}. \tag{5}$$

In this way, the mask value transformed by (4) on each site is bounded between 0 and 1, but can take values between 0 and 1 more smoothly. Because the purpose is to get the relative intensity distribution on the mask plane, we need a few iterations to generate the satisfying CTM as the guidance to insert SRAF.

B. SRAF Seed Candidate Generation

For CTM-based SRAF generation, the SRAFs are better located where the intensity values of CTM are larger. In our SRAF generation method, the SRAF seed candidates are labeled on the local maximal sites. In order to reduce the risk of being printed out on the wafer plane after the final mask optimization stage, the distance from the inserted SRAFs to the main pattern should be larger than the predefined threshold D_{opc}. Besides, for SRAFs inserted further away from the main pattern than the threshold D_{sraf}, the imaging interaction between SRAFs and the main pattern becomes negligible, resulting in no resolution enhancement. To obey the above-mentioned location constraint, we expand the edges of main patterns by D_{opc} and D_{sraf} to form the OPC safe region $\mathscr{R}_{\text{opc}}$ and SRAF effective region $\mathscr{R}_{\text{eff}}$, respectively. The XOR of $\mathscr{R}_{\text{opc}}$ and $\mathscr{R}_{\text{eff}}$ is the SRAF safe region $\mathscr{R}_{\text{sraf}}$, in which SRAF seed candidates are legal and effective.

For each site, the first order geometrical difference map of CTM is calculated, based on which we select all the local maximum points with larger CTM pixel values than its four neighbors (i.e., upper, lower, left, and right neighbors). Centering on the selected points, the SRAF seed candidates are shaped in squares with edge length v. In the following SRAF seed selection step, all these seed candidates are where SRAF seeds are possibly selected from.

C. SRAF Seed Selection

In the practical manufacturing process, a series of SRAF geometrical constraints are set to ensure that the mask rule check (MRC) is met. When the space between adjacent SRAF seed candidates is less than a threshold, violating the SRAF distance rule, we tend to select the seeds from candidates with maximum general transmissivity. Selecting the seeds is an NP-hard problem, and it is hard to approximate. This section proposes a fast and robust algorithm, guaranteed to converge to optimum within a few iterations.

1) Formulation: We define *weight* for SRAF seed candidates as the main objective for our SRAF seed selection algorithm.

Definition 1 (SRAF Seed Candidate): For a CTM $\boldsymbol{M}_{\text{C}}(x, y)$, the *weight*, or *priority*, of the ith SRAF seed candidate is defined as the average intensity

$$w_i = \frac{1}{v^2} \sum_{x=-\lfloor \frac{v}{2} \rfloor}^{\lfloor \frac{v}{2} \rfloor} \sum_{y=-\lfloor \frac{v}{2} \rfloor}^{\lfloor \frac{v}{2} \rfloor} \boldsymbol{M}_{\text{C}}(x_i + x, y_i + y) \tag{6}$$

where $\boldsymbol{p}_i := (x_i, y_i)$ represents the coordinates of the central point in the ith SRAF seed candidate, and the scalar value v is the fixed edge length that determines the area of this candidate.

Each SRAF seed candidate is assigned a unique weight value. We expect to select those candidates with the highest weights. Besides, there should be enough distance between each pair of the selected ones. Let $[n] = \{1, 2, \dots, n\}$ be the index set, and each SRAF seed candidate is assigned a unique index. We consider that a pair of candidates conflict with each other if they are too close.

Definition 2 (Conflict): A pair of candidates $(i, j) \in [n]^2$ conflict with each other under distance $d(\cdot, \cdot)$ and a threshold $\delta > 0$ if and only if $d(\boldsymbol{p}_i, \boldsymbol{p}_j) < \delta$.

In other words, two candidates have a conflict if and only if the distance of their center points is less than a specific threshold. Note that the distance d and the threshold δ mentioned in Definition 2 are customized according to specific requirements as hyperparameters, which will be discussed in detail in Section IV.

Given a distance function d and a threshold δ, we are able to construct the conflict matrix $\boldsymbol{H} \in \{0, 1\}^{n \times n}$ with each binary entry $h_{ij} \in \{0, 1\}$ indicating whether the ith and jth candidates conflict with each other under d and δ. More specifically, $h_{ij} = 1$ if the ith and jth candidates conflict with each other, and $h_{ij} = 0$ otherwise. Specially, $h_{ii} = 0$ is true for the diagonal entries. Our interest is to select a subset $S \subseteq [n]$ to maximize $\sum_{k \in S} w_k$ such that for any $i, j \in S$ $(i \neq j)$, we have $h_{ij} = 0$. To avoid ambiguity, we use $\boldsymbol{z}$ to denote the *indicator vector* of the subset S. The problem is formulated as the following integer programming under a quadratic constraint:

$$\boldsymbol{z}^* = \operatorname{argmax} \boldsymbol{w}^\top \boldsymbol{z}, \quad \text{s.t.} \quad \boldsymbol{z}^\top \boldsymbol{H} \boldsymbol{z} = 0; \quad z_i = \{0, 1\} \quad \forall i \in [n]. \tag{7}$$

Here, $\boldsymbol{w} \in \mathbb{R}^n$ is the weight vector with entry w_i being the weight of the ith SRAF seed candidate, defined in Definition 1. Note that (7) is NP-complete, so some techniques should be applied, here, to obtain a good solution.

2) Problem Reformulation: We move the quadratic constraint in (7) into the objective function to make it a 0-1 integer quadratic programming problem.

Theorem 1: For the following 0-1 integer quadratic programming problem:

$$\boldsymbol{z}^*(\lambda) = \operatorname{argmax} \boldsymbol{w}^\top \boldsymbol{z} - \frac{1}{2} \lambda \boldsymbol{z}^\top \boldsymbol{H} \boldsymbol{z}, \quad \text{s.t.} \ z_i = \{0, 1\} \quad \forall i \in [n]. \tag{8}$$

There exists a parameter $\lambda > 0$ such that the maximizer $\boldsymbol{z}^*(\lambda)$ is equal to the maximizer $\boldsymbol{z}^*$ of (7). Here, λ is called the positive regularization parameter.

According to Theorem 1, we can directly work on the integer quadratic programming (8). The existence of such a λ preserving maximizer is guaranteed. It is remarkable that the reformulation (8) requires components of $\boldsymbol{z}$ to be binary to preserve the maximizer.

3) Optimization Algorithm: Let $L_\lambda(\boldsymbol{z}) = \boldsymbol{w}^\top \boldsymbol{z} - (1/2) \lambda \boldsymbol{z}^\top \boldsymbol{H} \boldsymbol{z}$ be the reformulated objective. To facilitate the calculation, we use the auxiliary continuous variables $\boldsymbol{u} \in [0, 1]^n$ to represent the states of all candidates, and approximate $L_\lambda(\boldsymbol{z})$

to a simpler form as guidance to update the possible discrete solution.

Starting from an initial $\boldsymbol{u}^{(0)}$, the iterative algorithm updates $\boldsymbol{u}$ to $\boldsymbol{u}^{(k)}$ at the kth iteration ($k \geq 0$) and gives an approximate discrete solution $\boldsymbol{z}^{(k)}$. Around $\boldsymbol{u}^{(k)}$, the objective is approximated by its first-order Taylor expansion

$$L_\lambda(\boldsymbol{z}) \approx L_\lambda\left(\boldsymbol{u}^{(k)}\right) + \left(\boldsymbol{z} - \boldsymbol{u}^{(k)}\right)^\top\left(\boldsymbol{w} - \lambda\boldsymbol{H}\boldsymbol{u}^{(k)}\right). \tag{9}$$

In (9), the only term related to $\boldsymbol{z}$ is $g_k(\boldsymbol{z}) := \boldsymbol{z}^\top(\boldsymbol{w} - \lambda\boldsymbol{H}\boldsymbol{u}^{(k)})$. The possible discrete solution in the next iteration should maximize $L_\lambda(\boldsymbol{z})$, which is equivalent to maximizing $g_k(\boldsymbol{z})$ when $\boldsymbol{z}$ locates inside a small neighborhood of $\boldsymbol{u}^{(k)}$, i.e., we have $\boldsymbol{z}^{(k)} = \operatorname{argmax}_{\boldsymbol{z}\in\{0,1\}^n} \boldsymbol{z}^\top(\boldsymbol{w} - \lambda\boldsymbol{H}\boldsymbol{u}^{(k)})$.

Theorem 2: At the kth iteration, if the optimal solution to

$$\underset{\boldsymbol{z}\in\{0,1\}^n}{\operatorname{argmax}}\, \boldsymbol{z}^\top\left(\boldsymbol{w} - \lambda\boldsymbol{H}\boldsymbol{u}^{(k)}\right)$$

is $\boldsymbol{z}^{(k)}$ then the ith component of $\boldsymbol{z}^{(k)}$ is

$$z_i^{(k)} = \begin{cases} 1, & \text{if } \lambda\left(\boldsymbol{H}\boldsymbol{u}^{(k)}\right)_i < w_i \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

At each iteration, we have a real-valued guess $\boldsymbol{u}^{(k)}$ and a binary guess $\boldsymbol{z}^{(k)}$. If the function evaluation $L_\lambda(\boldsymbol{z}^{(k)}) \geq L_\lambda(\boldsymbol{u}^{(k)})$ is true, we consider $\boldsymbol{z}^{(k)}$ to be a reliable guess, then set it to be $\boldsymbol{u}^{(k+1)}$ and proceed to the next iteration. Note that, in (9), the second-order term is neglected. The minimizer of $L_\lambda(\boldsymbol{z})$ and $g_k(\boldsymbol{z})$ will no longer be relevant when $d(\boldsymbol{z}, \boldsymbol{u}^{(k)})$ is large. Therefore, the binary guess $\boldsymbol{z}^{(k)}$ may become unreliable, i.e., $L_\lambda(\boldsymbol{z}^{(k)}) < L_\lambda(\boldsymbol{u}^{(k)})$. Intuitively, we find a linear interpolation between $\boldsymbol{u}^{(k)}$ and $\boldsymbol{z}^{(k)}$ to be $\boldsymbol{u}^{(k+1)}$ for the next iteration such that the function value is nondecreasing. More specifically, for any $k \geq 0$, the linear interpolation problem can be expressed as

$$\boldsymbol{u}^{(k+1)} = \begin{cases} \boldsymbol{z}^{(k)}, & \text{if } L_\lambda(\boldsymbol{z}^{(k)}) \geq L_\lambda(\boldsymbol{u}^{(k)}) \quad \text{(11a)} \\ \theta_k\boldsymbol{z}^{(k)} + (1-\theta_k)\boldsymbol{u}^{(k)}, & \text{otherwise} \quad \text{(11b)} \end{cases}$$

which is controlled by the stepsize θ_k.

Theorem 3: In the kth iteration, if the θ_k of (11) is given by following equation:

$$\theta_k = \frac{\left(\boldsymbol{w} - \lambda\boldsymbol{H}\boldsymbol{u}^{(k)}\right)^\top\left(\boldsymbol{z}^{(k)} - \boldsymbol{u}^{(k)}\right)}{\lambda\left(\boldsymbol{z}^{(k)} - \boldsymbol{u}^{(k)}\right)^\top\boldsymbol{H}\left(\boldsymbol{z}^{(k)} - \boldsymbol{u}^{(k)}\right)} \tag{12}$$

then the following must be true.

1) θ_k satisfies $0 \leq \theta_k < 1$ when $L_\lambda(\boldsymbol{z}^{(k)}) < L_\lambda(\boldsymbol{u}^{(k)})$.
2) The function value is nondecreasing, i.e., $L_\lambda(\boldsymbol{u}^{(k+1)}) \geq L_\lambda(\boldsymbol{u}^{(k)})$.

Theorem 3 guarantees that the iterative process defined in (11) gives a sequence $\{\boldsymbol{u}^{(k)}\}_{k\geq 0}$ with nondecreasing function values $\{L_\lambda(\boldsymbol{u}^{(k)})\}_{k\geq 0}$. Note that the function values are upper bounded, then the function must converge to a finite value according to the *monotone convergence theorem*. We summarize the procedure in Algorithm 1.

D. Fast SRAF Shape Evolution

The selected SRAF seeds are in the same $v \times v$ square shape. Each of the seeds is then evolved into rectangles of different sizes. The object is the evolved rectangle SRAFs cover high-transmissivity region on CTM while not exceeding the SRAF shape constraint size. In this article, we proposed a probe-based fast evolution method. Starting from the center of each SRAF seed with the largest CTM value, as is shown in Fig. 5(a), four probes search leftward, rightward, upward, and downward until they reach the points where CTM values are less than the evolution intensity threshold I_{evo}. The distances between the ending points and center points are reported as l'_l, l'_r, l'_u, and l'_d, respectively. Considering the rectangle with size $(l'_l + l'_r) \times (l'_u + l'_d)$ may cover too much region area with less CTM values. We use an adjustment coefficient to fine-tune the size

$$c = \frac{l'_l + l'_r}{l'_l + l'_r + l'_u + l'_d} \tag{13}$$

$$l_l = c \cdot l'_l, \quad l_r = c \cdot l'_r, \quad l_u = (1-c)\cdot l'_u, \quad l_d = (1-c)\cdot l'_d. \tag{14}$$

If the total length in horizontal or vertical is larger than the maximum length L, the lengths are then compressed proportionally

$$l_l^* = \frac{l_l}{l_l + l_r}L, \quad l_r^* = \frac{l_r}{l_l + l_r}L, \quad \text{if } l_l + l_r > L$$
$$l_u^* = \frac{l_u}{l_u + l_d}L, \quad l_d^* = \frac{l_d}{l_u + l_d}L, \quad \text{if } l_u + l_d > L. \tag{15}$$

Algorithm 1 SRAF Seed Selection

Require: SRAF seed candidate weight $\boldsymbol{w}$, conflict matrix $\boldsymbol{H}$, convergence threshold ϵ, regularization parameter λ.
Ensure: selected SRAF seeds set $\boldsymbol{S}$.

```
Initialization: $\boldsymbol{u}^{(0)} \leftarrow \text{random}\{0,1\}^n$, $k \leftarrow 0$.
current best solution: $\boldsymbol{z}^* \leftarrow \boldsymbol{0}$.
repeat
    Update function: $g_k(\boldsymbol{z}, \boldsymbol{u}^{(k)}) \leftarrow \boldsymbol{z}^\top(\boldsymbol{w} - \lambda\boldsymbol{H}\boldsymbol{u}^{(k)})$.
    Possible discrete solution: $\boldsymbol{z}^{(k)}$;          ▷ Equation (10)
    if $L_\lambda(\boldsymbol{z}^{(k)}) \geq L_\lambda(\boldsymbol{u}^{(k)})$ then.
        Update: $\boldsymbol{u}^{(k+1)} \leftarrow \boldsymbol{z}^{(k)}$;
    else
        Select parameter: $\theta_k$;          ▷ Equation (12)
        Update: $\boldsymbol{u}^{(k+1)} \leftarrow \theta_k\boldsymbol{z}^{(k)} + (1-\theta_k)\boldsymbol{u}^{(k)}$;
    end if
    if $L_\lambda(\boldsymbol{z}^{(k)}) > L_\lambda(\boldsymbol{z}^*)$ then;
        $\boldsymbol{z}^* \leftarrow \boldsymbol{z}^{(k)}$;
    end if
    Iteration: $k \leftarrow k+1$;
until $\left|L_\lambda(\boldsymbol{u}^{(k+1)}) - L_\lambda(\boldsymbol{u}^{(k)})\right| < \epsilon$
$\boldsymbol{S} \leftarrow \boldsymbol{z}^*$;
```

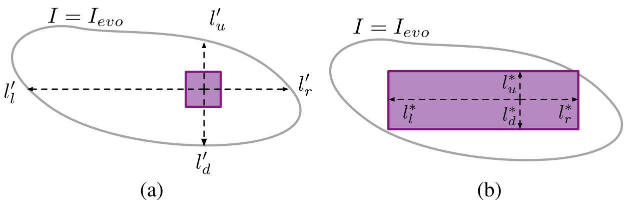

Fig. 5. SRAF fast evolution. (a) Initial SRAF seed and evolution intensity threshold contour. (b) Evolved SRAF.

TABLE I
COMPARISON OF DIFFERENT SRAF METHODS WITH LEVEL SET POST MASK OPTIMIZATION

	Rule-based SRAF [6]				CTM ridge-based SRAF [27]				Dictionary learning-based SRAF [22]				Ours			
ID	#EPE	PVB	RT	Cost	#EPE	PVB	RT	Cost	#EPE	PVB	RT	Cost	#EPE	PVB	RT	Cost
B1	4	61983	56	267988	4	58794	120	255296	3	62343	96	264468	3	57584	121	245457
B2	1	51298	56	210247	1	47862	121	196569	1	49640	64	203624	1	45756	93	188117
B3	29	101497	141	591129	29	99340	230	582590	30	105472	113	602001	28	92660	179	555819
B4	0	30088	111	120463	1	27467	209	115077	1	29400	117	122717	0	26061	128	104372
B5	1	56800	63	232263	1	56714	103	231959	1	57161	48	233692	1	54553	73	223285
B6	0	52970	44	211924	1	51050	103	209303	1	51405	47	210667	1	48134	72	197608
B7	0	46253	63	181076	0	44512	92	178140	0	45982	47	183975	0	43947	78	175866
B8	1	23077	17	97325	1	22057	88	93316	1	22847	48	96436	1	20657	66	87694
B9	0	64316	41	257305	0	62154	113	248729	0	62498	48	250040	0	60754	74	243090
B10	0	18026	17	72121	0	18795	66	75246	0	18661	38	74682	0	17426	57	69760
Avg.				224584				218623				224230				**209107**
Ratio				1.074				1.045				1.072				**1.000**

This is shown in Fig. 5(b). The size of each evolved SRAF is calculated using (14) and (15), the distance between the left edge, right edge, upper edge, and lower edge and the center point are l_l^*, l_r^*, l_u^*, and l_d^*, respectively. The evolved SRAFs are finally co-optimized with main pattern in mask optimization flow.

IV. Experimental Results

Our CTM-based SRAF generation method is implemented in C/C++. The lithography simulator is from the ICCAD 2013 CAD contest [3]. To verify the effectiveness of our SRAF generation method, we employ the public benchmarks released by IBM, for the same contest containing ten 32-nm Metal 1 layers. Compared with via layers containing vias of the same size, these benchmarks represent more challenging shapes to insert SRAFs. Each pattern is 2048 nm × 2048 nm with the resolution of 1 nm^2 per pixel. We adopt the level set-based ILT binary program from Yu et al. [30] to conduct the post SRAF-main pattern co-optimization and evaluate the performance of final printed images. All the experiments are conducted on Linux system with a 2.6-GHz CPU and a single Nvidia Titan X GPU. We set the initial SRAF width v to 30 nm, the maximum width to 100 nm. The minimum space between SRAFs is set to 150 nm. The SRAF safe region parameters D_{opc} and D_{sraf} are 35 and 350 nm, respectively. It is worth mentioning that since the lithography simulator and benchmarks are offered for academic purposes. There is little industrial guidance for choosing the exact value of these SRAF constraint parameters. However, our method can be generalized to a different set of parameters without changing the optimization flow.

A. Comparison w. Other SRAF Methods

In the first experiment, to prove the superiority of our CTM-based constraint-aware SRAF generation algorithm, we also implement different SRAF generation methods and co-optimize the masks with SRAFs using the same level set ILT engine [30]. The performance is evaluated from the final printed images using the cost formula that is officially defined in the contest [3]. The cost is the linear combination of the number of EPE (#EPE), PVB area, runtime (RT), and the number of shape violations

$$\text{Cost} = \text{RT} + 4 \times \text{PVB} + 5000 \times \#\text{EPE} + 10000 \times \text{ShapeViol}. \tag{16}$$

The definition of each metric in (16) can also be found in [3]. The EPE violation threshold is set to 15 nm. EPE is measured on the probe points located on the pattern edges every 40 nm. The ShapeViol is visually checked from the printed images. In our method, the runtime is evaluated on the entire procedure consists of CTM generation, SRAF generation, and post mask co-optimization. Generally speaking, the pattern fidelity and the process variational window are tradeoff in mask optimizations. For this cost, the weight of (#EPE) is much larger than the weight of PVB. We decrease α in cost function (3) when optimizing the mask.

The detailed results are listed in Table I. Compared with the rule-based SRAF method in [6], our result improves by 7.4%. The rule-based method claimed to add one layer of SRAFs around the main pattern and inserted the rectangular SRAFs with widths and distances toward the main patterns manually decided. This method is hard to obtain good performance, especially, when the shapes of the main patterns are complex. Compared with a recent CTM ridge-based SRAF generation with simple shape evolution and manual post-legalization [27], which is a model-based SRAF generation method adopted in real industry, our method still achieves a 4.5% improvement. It is worth mentioning that although the method in [27] also adopts the CTM as the guidance of SRAF generation, the following SRAF insertion optimization process depends heavily on manual adjustment. The method in [27] filters out all the ridge parts on CTM as the initial SRAF, then instead of selecting high-value SRAF seeds, it goes through serval stages of cleanup and simplification operations to shrink the shape of broadly distributed initial SRAFs to meet the SRAF constraints. Finally, all irregular shapes are divided into serval clusters and converted to rectangles. Compared with this method, our proposed method is an end-to-end optimization flow, which could automatically select SRAF seeds with high averaged weighted transmittance and evolve to desired shapes without violating the constraints. The performance improvements are mainly attributed to the reason that our method could distinguish the SRAF seeds with high value and, thus, enlarge the SRAF shapes evolving from them, while the compared method cannot distinguish between different regions and simplifies all irregular SRAF shapes equally. Thanks to Geng et al. [22], we are able to reimplement one state-of-the-art dictionary learning-based SRAF generation method with our lithography simulator on the ten Metal 1 benchmarks. We

TABLE II
CONTEST SCORE COMPARISON WITH OPTIMIZATION-BASED SOTA

	MOSAIC [29]				robust OPC [6]				PVOPC [7]				Level set [30]				SRAF-Level set			
ID	#EPE	PVB	RT	Cost	#EPE	PVB	RT	Cost	#EPE	PVB	RT	Cost	#EPE	PVB	RT	Cost	#EPE	PVB	RT	Cost
B1	9	56890	1707	274267	0	66218	278	265150	2	58269	164	243240	4	62693	123	270895	3	57584	121	245457
B2	4	48312	1245	214493	0	53434	142	213878	0	52674	130	210826	1	50724	81	207977	1	45756	93	188117
B3	52	84608	2523	600955	18	146776	152	677256	47	81541	360	561367	29	100945	214	598994	28	92660	179	555819
B4	3	24723	1269	115161	0	33266	307	133371	0	26960	265	108030	0	29831	184	119508	0	26061	128	104372
B5	2	56299	2167	237363	1	65631	189	267713	4	61820	62	267342	1	56510	76	231116	1	54553	73	223285
B6	1	49285	2084	204224	0	62068	353	248625	0	55090	54	220414	1	51204	65	209881	1	48134	72	197608
B7	0	46280	1641	186761	0	51069	219	204495	0	51977	74	207982	0	45056	64	180288	0	43947	78	175866
B8	2	22342	663	100031	0	25898	99	103691	0	22869	65	91541	1	22757	67	96095	1	20657	66	87694
B9	3	62529	3022	268138	1	75387	119	306667	0	70713	55	282907	0	64597	63	258466	0	60754	74	243090
B10	0	18141	712	73276	0	18536	61	74205	0	17846	41	71425	0	18769	64	75140	0	17426	57	69760
Avg.				227467				249505				226507				224836				**209107**
Ratio				1.088				1.193				1.083				1.075				**1.000**

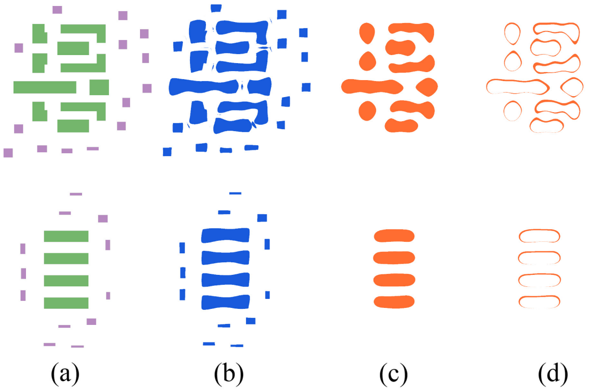

Fig. 6. Mask optimization results. (a) Target (green polygons) and SRAFs (purple rectangles). (b) Optimized masks. (c) Wafer images. (d) Process variational bands.

set the minimum space rule the same as ours. It is worth mentioning that those learning-based SRAF generation methods only support serval kinds of SRAF shapes. In our implementation, we adopt SRAFs of shape (*width*, *lengths*) with three choices: (40 nm, 40 nm), (40 nm, 80 nm), and (80 nm, 40 nm) without breaking our minimum and maximum width rule. We use the other nine benchmarks with SRAFs generated with our proposed method as the training set and predict on the left one benchmark. Compared with this dictionary learning-based method, our method achieves a 7.2% improvement. The reason can be mainly attributed to the high information extraction efficiency on complex pattern shapes using CTM, and the SRAF evolution strategy, making our method free from being limited to serval prefixed shapes.

Fig. 6 shows the examples of SRAF insertion and mask optimization results. The first row corresponds to a complex pattern (test case 1) and the second corresponds to a simple pattern (test case 10). The SRAFs are positioned and evolved around the main patterns flexibly and are not printed out on wafer images.

B. Runtime Breakdown Analysis

Fig. 7 shows the average runtime breakdown of level set-based ILT with and without SRAF generation. Our CTM-based SRAF generation needs extra runtime cost to generate the CTM with a few more iterations of simulated lithography calculation, which takes up 38.4% of total time. In order to leverage proper and distinct transmissivity distribution features, it takes a few iterations of relaxed ILT to optimize the CTM first, which takes up almost half of the total time. Based on the CTM, the SRAF can be generated very efficiently, which only takes up 1.3% of total time. It should be noted that with the help of SRAFs, the number of iterations for post mask optimization is largely reduced, and consequently the average runtime will be reduced by 6.5%.

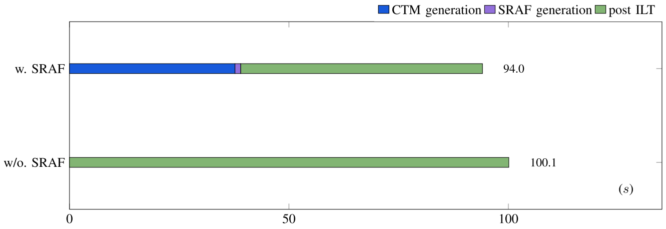

Fig. 7. Average runtime breakdown of level-set-based ILT with and without SRAF.

C. Comparison w. SOTA OPC Methods

Next, we embed our SRAF generation into a level set-based ILT engine released by [30] and compare the optimization results with different SOTA OPC methods. For each test target pattern, the generated SRAFs are identical to the SRAFs in Section IV-A.

The methods listed in Table II adopt the contest score (16) as the optimization objective. It is worth mentioning that we obtain the detailed data from the authors of robust OPC [6] and calculate the cost value based on it. Compared with high performance version of conventional ILT: MOSAIC [29], our cost is reduced by 8.8%. Compared with the rule-based SRAF inserted OPC: robust OPC [6], our algorithm could attain a 19.3% improvement. PVOPC [7] is a process variation-aware OPC algorithm with systematic SRAF inserted, compared with which our cost is reduced by 8.3%. Compared with the level set ILT [30] without SRAF, the SRAF generation could further refine the optimized mask and gain a 7.5% improvement.

D. Mask Printability Comparison With Learning-Based Methods

In this experiment, we compare the printability of our SRAF-aided optimized mask with recent SOTA machine-learning-based OPC methods. The metrics adopted are squared

TABLE III
MASK PRINTABILITY COMPARISON WITH LEARNING-BASED SOTA

ID	Neural-ILT [31]		PGAN-OPC [32]		Develset [33]		SRAF-Level set	
	L2	PVB	L2	PVB	L2	PVB	L2	PVB
B1	50795	63695	52570	56267	49142	59607	46405	56510
B2	36969	60232	42253	50822	34489	52012	33481	45455
B3	94447	85358	83663	94498	93498	76558	77734	83090
B4	17420	32287	19965	28957	18682	29047	13183	27439
B5	42337	65536	44733	59328	44256	58085	41569	55961
B6	39601	59247	46062	52845	41730	53410	38608	50477
B7	25424	50109	28609	47981	25797	46606	24191	44801
B8	15588	25826	19477	23564	15460	24836	15178	22038
B9	52304	68650	52613	65417	50834	64950	49073	62678
B10	10153	22443	22415	19893	10140	21619	8231	19833
Avg.	38504	53338	39949	49957	38403	48673	**34765**	**46828**
Ratio	1.108	1.139	1.149	1.067	1.105	1.039	**1.000**	**1.000**

TABLE IV
INFLUENCE OF SRAF SEED LOCATIONS

δ	#EPE	PVB	L2
0	3.0	57584	46196
2	3.6	59547	46378
5	4.0	61022	46990
8	4.2	61083	46663

L2 loss (L2) as defined in [31] and PVB, which are in accordance with the compared methods. To attain a better balance between L2 and PVB, we increase the weight coefficient α in (3). The detailed results are listed in Table III. The squared L2 loss reflects the general fidelity of printed images. Compared with the on-neural-network ILT framework Neural-ILT [31], our L2 loss is reduced by 10.8%. Compared with the GAN-based mask optimization flow PGAN-OPC [32], our method reduces L2 loss by 14.9%. For the latest DNN-accelerated level set-based OPC framework Develset [33], we still achieve a 10.5% improvement. Moreover, our method attains the best process window. The PVB of our optimized mask is 13.9%, 6.7%, and 3.9% smaller compared with Neural-ILT, PGAN-OPC, and Develset, respectively. Since the compared machine-learning-based methods listed only the inference time in their original papers, which is not comparable with our end-to-end optimization time, we do not list and compare the runtime in this section. The superior of our SRAF generation methods in mask fidelity and robustness are compatible with these machine-learning-based mask optimization frameworks. Our methods could generate high-quality mask training samples with violation-free SRAFs inserted. Besides our SRAF generation method could also be directly embedded in these machine-learning optimization flows.

E. Ablation Study on Seed Location on SRAF Generation Performance

An ablation study is conducted to evaluate the significance of SRAF seed locations on the SRAF generation performance. In this experiment, we chose a complex pattern B1 in the benchmark suite as the target pattern. The shape of the pattern can be visualized in the first row of Fig. 6. To evaluate the significance of seed locations, we randomly move each seed in four directions up, down, left, and right in fixed offset distance, or remain it in place. The final SRAFs are evolved based on the disturbed SRAF seeds, and the lithography performance is evaluated after the level set mask optimization. We choose the offset distance δ to be 2, 5, and 8 pixels, respectively. For each fixed offset distance, we disturb the SRAF seeds locations five times and calculate the average evaluation metrics. The detailed results are listed in Table IV. We can find that the SRAFs evolved from offset seeds lead to worse lithography performance compared with the optimal SRAFs. Especially, when the offset is relatively large, e.g., 5 or 8 pixels, the final lithography performance suffers from severe deterioration. This proves the significance of finding the optimal SRAF seeds.

V. CONCLUSION

In this article, we propose a CTM-based constraint-aware SRAF generation method, which could provide a developed initial SRAF-inserted mask or prepare high-quality training data for machine-learning-based SRAF methods. Assisted by the generated SRAF, the performance and robustness of mask optimization can be significantly improved. The experiment results on ICCAD 2013 contest benchmarks demonstrate that we can achieve fewer edge placement errors number, less squared L2 loss, and less PVB area on printed images.

APPENDIX

PROOF TO THEOREM 1

Proof: Note that we are working on $\boldsymbol{z} \in \{0, 1\}^n$. Therefore, we know that

$$\boldsymbol{z}^\top \boldsymbol{H}\boldsymbol{z} = \sum_{i,j} h_{ij} z_i z_j \in \mathbb{Z}_+$$

is a non-negative integer, because we always have $h_{ij}, z_i, z_j \in \{0, 1\}$. Providing that candidate weights are non-negative, $\boldsymbol{w}^\top \boldsymbol{z}$ has a finite upper bound

$$M = \max_{z \in \{0,1\}^n} \boldsymbol{w}^\top \boldsymbol{z} = \sum_{i=1}^{n} w_i.$$

The upper bound is achieved if and only if all candidates are taken, which is impossible when $\boldsymbol{H}$ is not a zero matrix. Now, for any $\boldsymbol{z} \in \{0, 1\}^n$ such that $\boldsymbol{z}^\top \boldsymbol{H}\boldsymbol{z} \neq 0$, it can be guaranteed that $\boldsymbol{z}^\top \boldsymbol{H}\boldsymbol{z} \geq 1$, and then we have

$$\boldsymbol{w}^\top \boldsymbol{z} - \frac{1}{2}\lambda \boldsymbol{z}^\top \boldsymbol{H}\boldsymbol{z} \leq M - \frac{1}{2}\lambda.$$

Therefore, when taking $\lambda > 2(M - \boldsymbol{w}^\top \boldsymbol{z}^*)$, the maximizer of (8) must appear inside $\{\boldsymbol{z} \in \{0, 1\}^n | \boldsymbol{z}^\top \boldsymbol{H}\boldsymbol{z} = 0\}$, and that is exactly $\boldsymbol{z}^*$ in (7). ■

PROOF TO THEOREM 2

Proof: At the kth iteration, denote $\{i \in [n] | (\boldsymbol{w} - \lambda \boldsymbol{H}\boldsymbol{u}^{(k)})_i > 0\}$ by S_k. The objective at the current iteration is written as

$$\begin{aligned} g_k(\boldsymbol{z}) &= \sum_{i \in S_k} z_i \left| \left(\boldsymbol{w} - \lambda \boldsymbol{H}\boldsymbol{u}^{(k)}\right)_i \right| - \sum_{i \notin S_k} z_i \left| \left(\boldsymbol{w} - \lambda \boldsymbol{H}\boldsymbol{u}^{(k)}\right)_i \right| \\ &\leq \sum_{i \in S_k} \left| \left(\boldsymbol{w} - \lambda \boldsymbol{H}\boldsymbol{u}^{(k)}\right)_i \right|. \end{aligned}$$

The equality holds if and only if $z_i = 1$ for index $i \in S_k$ and $z_i = 0$ for index $i \notin S_k$. ■

Proof to Theorem 3

Proof: Consider the kth iteration ($k \geq 0$). We have the current real-valued guess $\boldsymbol{u}^k \in [0, 1]^n$. The discrete guess $\boldsymbol{z}^{(k)}$ is given by (10). If $\boldsymbol{z}^{(k)}$ is a reliable guess, i.e., $L_\lambda(\boldsymbol{z}^{(k)}) \geq L_\lambda(\boldsymbol{u}^{(k)})$, it is trivial that

$$L_\lambda\left(\boldsymbol{u}^{(k+1)}\right) - L_\lambda\left(\boldsymbol{u}^{(k)}\right) = L_\lambda\left(\boldsymbol{z}^{(k)}\right) - L_\lambda\left(\boldsymbol{u}^{(k)}\right) \geq 0.$$

We focus on cases where $L_\lambda(\boldsymbol{z}^{(k)}) < L_\lambda(\boldsymbol{u}^{(k)})$. Consider the following real-valued function $\varphi : \mathbb{R} \to \mathbb{R}$:

$$\varphi(\theta) = L_\lambda\left(\theta \boldsymbol{z}^{(k)} + (1 - \theta)\boldsymbol{u}^{(k)}\right). \tag{17}$$

It is continuously differentiable. The derivative is given by

$$\begin{aligned}\varphi'(\theta) &= \nabla L_\lambda\left(\theta \boldsymbol{z}^{(k)} + (1-\theta)\boldsymbol{u}^{(k)}\right)^\top \left(\boldsymbol{z}^{(k)} - \boldsymbol{u}^{(k)}\right) \\ &= \left(\boldsymbol{w} - \lambda \boldsymbol{H}\left(\theta \boldsymbol{z}^{(k)} + (1-\theta)\boldsymbol{u}^{(k)}\right)\right)^\top \left(\boldsymbol{z}^{(k)} - \boldsymbol{u}^{(k)}\right) \\ &= \left(\boldsymbol{w} - \lambda \boldsymbol{H}\boldsymbol{u}^{(k)}\right)^\top \boldsymbol{v}^{(k)} - \theta\lambda \boldsymbol{v}^{(k)\top}\boldsymbol{H}\boldsymbol{v}^{(k)}\end{aligned}$$

where $\boldsymbol{v}^{(k)} = \boldsymbol{z}^{(k)} - \boldsymbol{u}^{(k)}$. Denote $\{i \in [n] | (\boldsymbol{w} - \lambda\boldsymbol{H}\boldsymbol{u}^{(k)})_i > 0\}$ by S_k. According to (10), we obtain

$$\begin{aligned}&\left(\boldsymbol{w} - \lambda \boldsymbol{H}\boldsymbol{u}^{(k)}\right)^\top \boldsymbol{v}^{(k)} \\ &= \sum_{i \in S_k} v_i^{(k)}\left(w_i - \lambda\left(\boldsymbol{H}\boldsymbol{u}^{(k)}\right)_i\right) + \sum_{i \notin S_k} v_i^{(k)}\left(w_i - \lambda\left(\boldsymbol{H}\boldsymbol{u}^{(k)}\right)_i\right) \\ &= \sum_{i \in S_k} v_i^{(k)}\left|w_i - \lambda\left(\boldsymbol{H}\boldsymbol{u}^{(k)}\right)_i\right| - \sum_{i \notin S_k} v_i^{(k)}\left|w_i - \lambda\left(\boldsymbol{H}\boldsymbol{u}^{(k)}\right)_i\right| \\ &= \sum_{i \in S_k}\left(1 - u_i^{(k)}\right)\left|w_i - \lambda\left(\boldsymbol{H}\boldsymbol{u}^{(k)}\right)_i\right| + \sum_{i \notin S_k} u_i^{(k)}\left|w_i - \lambda\left(\boldsymbol{H}\boldsymbol{u}^{(k)}\right)_i\right| \\ &\geq\ 0.\end{aligned}$$

That is because $z_i^{(k)} = 1$ for index $i \in S_k$ and $z_i^{(k)} = 0$ for index $i \notin S_k$. Therefore, if $\boldsymbol{v}^{(k)\top}\boldsymbol{H}\boldsymbol{v}^{(k)} \leq 0$ in the second term of $\varphi'(\theta)$, it must be true that $\varphi'(\theta) \geq 0$ for $\theta \in [0, 1]$, and then $L_\lambda(\boldsymbol{z}^{(k)}) = \varphi(1) \geq \varphi(0) = L_\lambda(\boldsymbol{u}^{(k)})$. This conflicts with the assumption that $L_\lambda(\boldsymbol{z}^{(k)}) < L_\lambda(\boldsymbol{u}^{(k)})$. Now, under this assumption, we have $\boldsymbol{v}^{(k)\top}\boldsymbol{H}\boldsymbol{v}^{(k)} > 0$.

Take θ_k defined in (12), then $\theta_k > 0$ must be true, and we have $\varphi'(\theta) \geq 0$ for $0 \leq \theta \leq \theta_k$ and $\varphi'(\theta) < 0$ for $\theta > \theta_k$. If $\theta_k \geq 1$, $\varphi(\theta)$ is nondecreasing within $\theta \in [0, 1]$, which implies $L_\lambda(\boldsymbol{z}^{(k)}) = \varphi(1) \geq \varphi(0) = L_\lambda(\boldsymbol{u}^{(k)})$. Therefore, it must be satisfied that $0 \leq \theta_k < 1$, and

$$L_\lambda\left(\boldsymbol{u}^{(k+1)}\right) = \varphi(\theta_k) \geq \varphi(0) = L_\lambda\left(\boldsymbol{u}^{(k)}\right).$$

It is remarkable that $u_i^{(k+1)} = \theta_k z_i^{(k)} + (1 - \theta_k)u_i^{(k)} \in [0, 1]$ for $i \in [n]$, so this iterative process is valid and produces nondecreasing objective values. ■

References

[1] M. Rothschild, "A roadmap for optical lithography," *Opt. Photon. News*, vol. 21, no. 6, pp. 26–31, 2010.

[2] Y. Ping et al., "Process window enhancement using advanced RET techniques for 20nm contact layer," in *Proc. SPIE*, 2014, pp. 460–469.

[3] 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 Des. (ICCAD)*, 2013, pp. 271–274.

[4] 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 Des. (ICCAD)*, 2014, pp. 238–245.

[5] 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. Des. (ISQED)*, 2000, pp. 81–85.

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

[7] 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.

[8] A. Poonawala and P. Milanfar, "Mask design for optical microlithography—An inverse imaging problem," *IEEE Trans. Image Process.*, vol. 16, pp. 774–788, 2007.

[9] 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 Des. (ICCAD)*, 2017, pp. 81–88.

[10] C. H. Wallace, P. A. Nyhus, and S. S. Sivakumar, "Sub-resolution assist features," U.S. Patent 7 632 610, 2009.

[11] R. Viswanathan, J. T. Azpiroz, and P. Selvam, "Process optimization through model based SRAF printing prediction," in *Proc. SPIE Adv. Lithogr.*, 2012, Art. no. 83261A.

[12] J. Jun et al., "Layout optimization with assist features placement by model based rule tables for 2x node random contact," in *Proc. SPIE*, 2015, Art. no. 94270D.

[13] C. Kodama, T. Kotani, S. Nojima, and S. Mimotogi, "Sub-resolution assist feature arranging method and computer program product and manufacturing method of semiconductor device," U.S. Patent 8 809 072, Aug. 19, 2014.

[14] L. D. Barnes, B. D. Painter, and L. S. Melvin, III, "Model-based placement and optimization of subresolution assist features," in *Proc. Opt. Microlithogr. XIX*, 2006, pp. 789–795.

[15] K. Sakajiri, A. Tritchkov, and Y. Granik, "Model-based SRAF insertion through pixel-based mask optimization at 32nm and beyond," in *Proc. SPIE*, 2008, pp. 325–336.

[16] J. Ye, Y. Cao, and H. Feng, "System and method for model-based sub-resolution assist feature generation," U.S. Patent 7 882 480, Feb. 1, 2011.

[17] S. D. Shang, L. Swallow, and Y. Granik, "Model-based SRAF insertion," U.S. Patent 8 037 429, Oct. 11, 2011.

[18] L. Pang, Y. Liu, T. Dam, K. Mihic, T. Cecil, and D. Abrams, "Inverse lithography technology (ILT): Keep the balance between SRAF and MRC at 45 and 32 nm," in *Proc. SPIE*, 2007, pp. 1615–1624.

[19] B.-S. Kim et al., "Pixel-based SRAF implementation for 32nm lithography process," in *Proc. SPIE*, 2008, pp. 266–276.

[20] X. Xu et al., "Subresolution assist feature generation with supervised data learning," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 6, pp. 1225–1236, Jun. 2018.

[21] M. B. Alawieh, Y. Lin, Z. Zhang, M. Li, Q. Huang, and D. Z. Pan, "GAN-SRAF: Subresolution assist feature generation using conditional generative adversarial networks," in *Proc. ACM/IEEE Des. Autom. Conf. (DAC)*, 2019, pp. 1–6.

[22] H. Geng, W. Zhong, H. Yang, Y. Ma, J. Mitra, and B. Yu, "SRAF insertion via supervised dictionary learning," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 10, pp. 2849–2859, Oct. 2020.

[23] J.-C. Yu, P. Yu, and H.-Y. Chao, "Model-based sub-resolution assist features using an inverse lithography method," in *Proc. SPIE*, 2008, pp. 254–264.

[24] L. Pang et al., "Considering MEEF in inverse lithography technology (ILT) and source mask optimization (SMO)," in *Proc. SPIE*, 2008, pp. 631–644.

[25] S. Hsu et al., "An innovative source-mask co-optimization (SMO) method for extending low k1 imaging," in *Proc. SPIE*, 2008, Art. no. 714010.

[26] S. Hsu, Z. Li, L. Chen, K. Gronlund, H.-Y. Liu, and R. Socha, "Source-mask co-optimization: Optimize design for imaging and impact of source complexity on lithography performance," in *Proc. SPIE*, 2009, pp. 101–111.

[27] P. Gao, L. Zhang, and Y. Y. Wei, "SRAF generation based on SGM/CTM contour line," in *Proc. SPIE*, 2021, pp. 171–176.

[28] P. Gao, X. Su, W. Shi, Y. Wei, and T. Ye, "Sub-resolution assist feature cleanup based on grayscale map," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 4, pp. 583–588, Nov. 2019.

[29] 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 Des. Autom. Conf. (DAC)*, 2014, pp. 1–6.

[30] Z. Yu, G. Chen, Y. Ma, and B. Yu, "A GPU-enabled level set method for mask optimization," in *Proc. IEEE/ACM Des. Autom. Test Europe (DATE)*, 2021, pp. 1835–1838.

[31] B. Jiang, L. Liu, Y. Ma, H. Zhang, E. F. Y. Young, and B. Yu, "Neural-ILT: Migrating ILT to neural networks for mask printability and complexity co-optimization," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des. (ICCAD)*, 2020, pp. 1–9.

[32] H. Yang, S. Li, Z. Deng, Y. Ma, B. Yu, and E. F. Y. Young, "GAN-OPC: Mask optimization with lithography-guided generative adversarial nets," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 10, pp. 2822–2834, Oct. 2020.

[33] G. Chen, Z. Yu, H. Liu, Y. Ma, and B. Yu, "DevelSet: Deep neural level set for instant mask optimization," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des. (ICCAD)*, 2021, pp. 1–9.

Ziyang Yu received the B.S. degree from the Department of Physics, University of Science and Technology of China, Hefei, China, in 2018, and the M.Phil. degree from the Department of Physics, University of Hong Kong, Hong Kong, in 2020. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong.

His current research interests include design space exploration in electronic design automation and machine learning on chips.

Peiyu Liao received the B.S. degree from the School of Mathematical Sciences, Zhejiang University, Hangzhou, China, in 2017, and the M.S. degree from the School of Engineering, The Hong Kong University of Science and Technology, Hong Kong, in 2019. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, and the School of Integrated Circuits, Peking University, Beijing, China.

His current research interests include high-performance computing and numerical optimization in physical design.

Yuzhe Ma (Member, IEEE) received the B.E. degree from the Department of Microelectronics, Sun Yat-sen University, Guangzhou, China, in 2016, and the Ph.D. degree from the Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong, in 2020.

He is currently an Assistant Professor of Microelectronics Thrust with The Hong Kong University of Science and Technology (Guangzhou), Guangzhou. His research interests include agile VLSI design methodologies, machine-learning-aided VLSI design, and hardware-friendly machine learning.

Dr. Ma received the Best Paper Awards from ICCAD 2021, ASPDAC 2021, and ICTAI 2019, and the Best Paper Award Nomination from ASPDAC 2019.

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

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

Dr. Yu received nine Best Paper Awards from DATE 2022, ICCAD 2021 and 2013, ASPDAC 2021 and 2012, ICTAI 2019, *Integration, the VLSI Journal* in 2018, ISPD 2017, and SPIE Advanced Lithography Conference 2016, and six ICCAD/ISPD contest awards. 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. He is an Editor of IEEE TCCPS Newsletter.

Martin D. F. Wong (Fellow, IEEE) received the B.Sc. degree in mathematics from the University of Toronto, Toronto, ON, Canada, in 1979, and the M.S. degree in mathematics and the Ph.D. degree in computer science from the University of Illinois at Urbana–Champaign (UIUC), Champaign, IL, USA, in 1981 and 1987, respectively.

He was a Faculty Member with The University of Texas at Austin (UT-Austin), Austin, TX, USA, from 1987 to 2002 and UIUC from 2002 to 2018. He was a Bruton Centennial Professor of Computer Science with UT-Austin and an Edward C. Jordan Professor of ECE with UIUC. From August 2012 to December 2018, he was the Executive Associate Dean of the College of Engineering, UIUC. Since January 2019, he has been with The Chinese University of Hong Kong, Hong Kong, as the Dean of Engineering and a Choh-Ming Li Professor of Computer Science and Engineering. His main research interest is in electronic design automation (EDA). He has published around 500 papers and graduated over 50 Ph.D. students in EDA.

Prof. Wong is a Fellow of ACM.