Graph-Based Subfield Scheduling for Electron-Beam Photomask Fabrication

Shao-Yun Fang, Wei-Yu Chen, and Yao-Wen Chang, *Fellow, IEEE*

Abstract—Electron beam lithography has shown great promise in photomask fabrication; however, its successive heating process centralizing in a small region may cause a severe problem of critical dimension (CD) distortion. Consequently, subfield scheduling that reorders the sequence of the writing process is needed to avoid successive writing of neighboring subfields. In addition, the writing process of a subfield raises the temperature of neighboring regions and may block other subfields for writing. This paper presents the first work to solve the subfield scheduling problem while considering blocked regions by formulating the problem into a constrained maximum scatter traveling salesman problem (constrained MSTSP). To tackle the constrained MSTSP that can be shown to be NP-complete in general, we identify a special case thereof with points on two parallel lines and solve it optimally in linear time. We then decompose the constrained MSTSP into subproblems conforming to the special case, solve each subproblem optimally and efficiently by a graph-based algorithm, and then merge the subsolutions into a complete scheduling solution. We also extend our algorithm to handle the cases when the moving time of an e-beam writing head is comparable with the writing time of a subfield. Experimental results show that our algorithms are effective and efficient in finding good subfield scheduling solutions that can alleviate the successive heating problem (and thus reduce CD distortion) for e-beam photomask fabrication.

Index Terms—Design for manufacturability, electron beam lithography, photomask fabrication, subfield scheduling.

I. INTRODUCTION

AS INTEGRATED circuit (IC) process nodes continue to shrink to 22 nm and below, the IC industry will face severe manufacturing challenges with conventional optical lithography technologies. Electron beam lithography (EBL) is one of the most anticipated next-generation lithography (NGL) technologies, as the electron beam (e-beam) can be easily

Manuscript received June 13, 2012; revised August 27, 2012; accepted September 29, 2012. Date of current version January 18, 2013. This work was supported in part by IBM, SpringSoft, TSMC, and NSC of Taiwan, under Grant NSC 100-2221-E-002-088-MY3, Grant NSC 99-2221-E-002-207-MY3, Grant NSC 99-2221-E-002-210-MY3, and Grant NSC 98-2221-E-002-119-MY3. This paper was recommended by Associate Editor J. Hu.

S.-Y. Fang and W.-Y. Chen are with the Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 106, Taiwan (e-mail: yuko703@eda.ee.ntu.edu.tw; cweiyu@eda.ee.ntu.edu.tw).

Y.-W. Chang is with the Department of Electrical Engineering and the Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 106, Taiwan, and also with the Research Center for Information Technology Innovation, Academia Sinica, Taipei 115, Taiwan (e-mail: ywchang@cc.ee.ntu.edu.tw).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCAD.2013.2237947

Fig. 1. (a) Sequential e-beam writing process. (b) Writing process reordering with subfield scheduling to avoid successive writing of neighboring subfields. (c) Writing process of a subfield raises the temperature of neighboring regions.

focused into nanometer diameters by using electromagnetic or electrostatic lenses [9], [10]. In addition, in comparison with optical lithography, EBL is not limited by light diffraction [19]. As a result, the e-beam can define very fine, high-resolution patterns in a resist.

For high-volume manufacturing, massively parallel e-beam writing systems are under development but are not yet ready [5], [7], [8], [22]. On the other hand, since the dramatically decreasing minimum feature sizes of semiconductor devices have made mask making one of the most challenging tasks, the advantages of e-beam lithography have made high-voltage single e-beam systems popular in photomask fabrication. However, these high-voltage beams deposit a considerable amount of heat in a small area and result in critical dimension (CD) distortion [12], [14], [16], [18], [21].

In EBL, patterns are written by beam deflection and stage movement. Since the range of beam deflection is limited, a layout (main field) is split into subregions (subfields) and an e-beam writer writes patterns in a subfield at a time [20]. To maximize throughput, the writing process is usually performed in a sequential manner, as shown in Fig. 1(a). In this paper, we refer to this as contiguously sequential writing: where the writing proceeds in order from one subfield to the next adjacent subfield. However, the contiguously sequential writing process generates heat centralized in a region, which can aggravate the CD distortion problem. To solve this, Babin *et al.* [2], [3] proposed subfield scheduling, which reorders the sequence of the writing process to avoid the successive writing of subfields which are close to each other, as shown in Fig. 1(b).

Fig. 2 illustrates the effectiveness of subfield scheduling in alleviating the successive heating problem. Fig. 2(a) and (b)

TABLE I
IMPACT OF THE HEATING EFFECT ON LINE EDGE ROUGHNESS

Line spacing (nm)	Line edge roughness (nm)		Improvement
	Contiguously sequential	Noncontiguously-sequential	
120	11.28	3.51	68.88%
100	13.73	4.99	63.66%

Line edge roughness can be much improved with a noncontiguously-sequential writing process.

Fig. 2. SEM images of 1-D lines with different writing orders. SEM images with (a) contiguously sequential and (b) noncontiguously-sequential writing processes. (c), (d) Highlighted SEM images of the two corresponding writing processes. The resulting line end roughness is improved (much smaller) with a noncontinuously-sequential writing process.

shows two scanning electron micrograph (SEM) images of 1-D lines with contiguously sequential and noncontiguously-sequential writing processes, respectively. From the highlighted SEM images shown in Fig. 2(c) and (d), it can be seen that the resulting line end roughness is improved (much smaller) with the noncontiguously-sequential writing process. To further quantify the improvement in pattern quality with subfield scheduling, Table I lists the numerical data of line edge roughness obtained from the SEM images, which is computed by using the root mean square (RMS) method [13]. The results show that the noncontiguously-sequential writing process significantly reduces the line edge roughness by about 63% and 68% for 1-D line patterns with 100 and 120 nm spacings, respectively. Subfield scheduling is hence an effective technique to mitigate the successive heating problem in EBL and thus reduce CD distortion.

The objective of the subfield scheduling problem is to find a labeling (ordering) of subfields where the minimum distance between two subfields with successive labels is maximized. Babin *et al.* [2] first applied Lagarias scheduling to derive a well-spaced labeling, in which the resulting minimum Manhattan distance between subfields with successive labels is at most one unit length (edge length of a subfield) away from the optimal (maximum possible distance) [15]. However, using the Manhattan measure in this problem might not be appropriate, as the e-beam writing head moves freely (the Euclidean distance is a more appropriate metric in this case). Later, they proposed a greedy local improvement method for this problem [3], where a random subfield scheduling is

generated at the beginning and the scheduling is iteratively improved by swapping the orders of pairs of subfields. As the problem size grows, nevertheless, its running time grows prohibitively.

Another issue addressed in the previous work is the raised temperature of neighboring regions due to the writing process of a subfield [2]. As shown in Fig. 1(c), the writing process of the middle subfield significantly raises the temperature of neighboring subfields. Those subfields with temperatures higher than a threshold value are called blocked subfields for a certain processing duration; they should not be written in successive writing processes before the temperature drops below the threshold. However, no existing work has optimized the subfield scheduling problem while simultaneously considering blocked subfields during the writing process.

This paper presents the first work to solve the subfield scheduling problem considering blocked regions by formulating the problem as a constrained maximum scatter traveling salesman problem (constrained MSTSP for short). To tackle the constrained MSTSP that can be shown to be NP-complete in general, we identify a special case of the constrained MSTSP with points on two parallel lines and solve it optimally in linear time. We then decompose the constrained MSTSP into subproblems conforming to the special case, solve each subproblem optimally and efficiently by a graph-based algorithm, and then merge the subsolutions into a complete scheduling solution. In addition, to maximize the throughput for the cases when the moving time of an e-beam writing head is comparable with the writing time of a subfield, we further extend our algorithm considering the total traveling distance of an e-beam writing head. Experimental results show that our algorithms are effective and efficient in finding good subfield scheduling solutions that significantly alleviate the successive heating problem (and thus reduce CD distortion) for e-beam photomask fabrication.

The rest of this paper is organized as follows: Section II outlines the preliminaries and the problem formulation of this paper. Section III presents an exact algorithm that optimally solves the MSTSP for a special case; Section IV details the graph-based subfield scheduling algorithm. Experimental results are reported in Section V. Section VI concludes this paper.

II. PRELIMINARIES

In this section, the preliminaries of the subfield scheduling problem are given. First, a proven thermal model is introduced in Section II-A. Then, the blocked box of a subfield is defined in Section II-B. Section II-C introduces the traditional and constrained MSTSPs, and Section II-D gives the throughput

consideration in the subfield scheduling problem. Finally, the problem formulation is presented in Section II-E.

A. Thermal Model

Babin *et al.* [3] developed a greedy local improvement algorithm that yields better subfield scheduling results than any other methods. The algorithm is embedded with a model of temperature computation in which two basic principles are considered: 1) The writing process of a subfield causes temperature increases for all other subfields; these increases depend mainly on the amount of energy deposited through the writing process and the distance from the subfield being written. 2) The temperature of each subfield decays exponentially during the traveling time between two successive writing processes.

In the thermal model, $\pi = (\pi_1, \pi_2, \dots, \pi_n)$ denotes a scheduling order of subfields, and $T_{i,j}$ represents the temperature of a subfield π_i before the writing process of a subfield π_j . According to the first principle of the model, the amount of the raised temperature of a subfield π_i due to the writing process of a subfield π_j , denoted by $T_{i,j}^{\text{rise}}$, is proportional to the temperature difference and is inversely proportional to the squared Euclidean distance between the two subfields. We have

$$T_{i,j}^{\text{rise}} \propto \frac{T_{j,j} + T_{j,j}^{\text{rise}} - T_{i,j}}{\text{dist}(\pi_i, \pi_j)^2} \quad (1)$$

where $T_{j,j}^{\text{rise}}$ denotes the raised temperature of a subfield π_j due to its own writing process, and $\text{dist}(\pi_i, \pi_j)$ is the Euclidean distance between subfields π_i and π_j . As the temperature of a subfield π_i decays exponentially during the traveling time between two successive writing processes, the temperature of a subfield π_i right before the writing process of a subfield π_j can be formulated as

$$T_{i,j} = (T_{i,j-1} + T_{i,j-1}^{\text{rise}})f \quad (2)$$

where f is the decay factor which depending on the length of the traveling time between two successive writing processes; that is, f depends on the distance between two subfields with successive labels.

Equations (1) and (2) show that increasing the distance between each pair of subfields with successive labels may not only prevent the writing of a subfield with excessively high temperature but also allow for a longer time to cool down all subfields. In this paper, therefore, we formulate this problem as a constrained maximum scatter traveling salesman problem and propose an algorithm to maximize the distance between each pair of successive subfields.

B. Blocked Boxes of Subfields

First, we let $\mathcal{S}$ be the set of given subfields. As mentioned in Section I, the writing process of a subfield raises the temperature of neighboring subfields. Successive writing processes should not be performed on unwritten subfields whose temperatures are higher than a threshold value. As a result, we define a blocked box b_i for each subfield $S_i \in \mathcal{S}$ which indicates the area that is blocked because of the writing process

Fig. 3. Blocked boxes for two subfields. The size of a blocked box varies with the pattern density of the corresponding subfield.

of the subfield S_i , as illustrated in Fig. 3. Since the temperature increase depends on the energy deposited through the writing process, and the deposited energy is proportional to the volume of the patterns, the area of a blocked box varies with the pattern density of the corresponding subfield. In addition, each $S_i \in \mathcal{S}$ is associated with a blocking duration d_i . The subfields included in b_i cannot be written within the time from t_i to $t_i + w_i + d_i$, referred to as the blocked box constraint, where t_i is the writing start time of S_i and w_i is the time required for the writing process of S_i . In addition to satisfying the blocked box constraint, the time difference between the writing start times of two subfields with overlapped blocked boxes should be as large as possible to further control the temperature during photomask fabrication.

C. Constrained Maximum Scatter Traveling Salesman Problem

The maximum scatter traveling salesman problem (MSTSP), a variation of the classical traveling salesman problem (TSP), finds a Hamiltonian cycle or path that is most scattered. Its objective is to maximize the minimum edge weight in the cycle or the path. In an edge-weighted complete graph $G = (V, E)$, let $\mathbb{F}$ be the family of Hamiltonian cycles (or paths) in G . For each edge $e \in E$, the edge weight w_e is predefined. For a Hamiltonian cycle or path $\mathcal{H}$, we denote $c(\mathcal{H})$ as the cost of $\mathcal{H}$, which is the minimum edge weight in $\mathcal{H}$; i.e., $c(\mathcal{H}) = \min\{w_e : e \in \mathcal{H}\}$. Then, the MSTSP is to find a Hamiltonian cycle (or path) $\mathcal{H} \in \mathbb{F}$ such that $c(\mathcal{H})$ is maximized.

Observing that the objective of the subfield scheduling in e-beam photomask fabrication is similar to that of MSTSP, we transform the scheduling problem into an MSTSP. However, due to the blocked box consideration, an optimal solution derived from the corresponding MSTSP may not be feasible. Fig. 4 shows an example. For a case with seven subfields on a line, we assume that $w_i = 1$, $d_i = 2$, and that b_i only covers the adjacent subfields for each subfield S_i ; that is, if subfield S_i is written from time t to time $t + 1$, then subfields S_{i-1} and S_{i+1} are blocked from time $t + 1$ to time $t + 3$. If the blocked box constraint is not considered, an optimal scheduling solution for the corresponding MSTSP is given in Fig. 4(a), which is $\langle S_1, S_5, S_2, S_6, S_3, S_7, S_4 \rangle$. In this scheduling order, at time $t = 2$, the subfield S_2 is written while it is blocked; this solution is thus infeasible. Another scheduling solution $\langle S_1, S_3, S_5, S_7, S_2, S_4, S_6 \rangle$ is given in Fig. 4(b). Take subfield

Fig. 4. (a) Infeasible subfield scheduling with the shortest distance between successive subfields being 3. (b) Feasible subfield scheduling with the shortest distance between successive subfields being 2.

S_2 for example: it is blocked from $t = 2$ to $t = 4$ due to the writing process of subfield S_3 which finishes at $t = 2$. Thus, the writing process of S_2 starting at $t = 4$ is valid since S_2 is unblocked after $t = 4$. As shown in Fig. 4(b), the scheduling solution is feasible since no subfield is written while it is blocked. Although the shortest distance between any two successive subfields of the scheduling solution in Fig. 4(a) is longer than that in Fig. 4(b), the former solution does not satisfy the blocked box constraint and is thus not feasible. Therefore, we must consider the blocked box constraint while finding a subfield scheduling solution; we define the problem as the *constrained MSTSP*.

It has been proven that there exists no polynomial-time, constant-performance-bound approximation algorithm for the general MSTSP [1]. Even given a graph satisfying the triangle inequality, there exists no polynomial-time algorithm with a performance bound smaller than two [1]. Since the MSTSP can be reduced to the constrained MSTSP with blocked box constraints by restricting d_i to be zero for each S_i , the constrained MSTSP can be shown to be NP-complete.

Theorem 1: The constrained MSTSP is NP-complete.

D. Throughput Consideration in Subfield Scheduling

As shown in Section I, subfield scheduling can effectively mitigate the heating problem in EBL; however, it may increase the total writing time of a chip if the moving time of an e-beam writer is comparable with the writing time of a subfield. In this situation, subfield scheduling may reduce the throughput, and thus the total traveling distance of a subfield scheduling solution should be considered.

Fig. 5 shows that different subfield scheduling solutions can have different throughput results and heating mitigation effects. Note that in this example, the numbers denote the writing orders, and the blocked boxes are not considered. To

Fig. 5. (a) Writing in the contiguously sequential way to minimize the total traveling distance. (b) Writing with an optimal MSTSP solution to maximize the minimum distance. (c) Scheduling solution simultaneously considering the total traveling distance and the minimum distance.

maximize throughput, writing in the contiguously sequential way can get the minimum traveling distance of an e-beam writer, as illustrated in Fig. 5(a). However, the short distances between successive writing processes may cause a severe heating problem. On the other hand, as illustrated in Fig. 5(b), writing according to the order of an optimal MSTSP solution can maximize the minimum distance between any successive writing processes; nevertheless, the total traveling distance increases significantly at the same time. Therefore, a subfield scheduling solution that simultaneously mitigates the heating problem and considers the throughput in EBL is desired. As the scheduling solution shown in Fig. 5(c), the minimum distance between any successive writing processes is longer than that in Fig. 5(a), and the total traveling distance is shorter than that in Fig. 5(b).

E. Problem Formulation

We use a point to represent a subfield and the Euclidean measure to define the distances between subfields. We transform the subfield scheduling problem into a constrained MSTSP problem as follows:

Problem 1: Given a photomask layout and the predefined size of a subfield, find a Hamiltonian path containing all points such that the length of the shortest edge in the Hamiltonian path is maximized, and the blocked box constraint is satisfied.

By solving this constrained MSTSP problem, we can alleviate the problem with successive heating and thus also control CD distortion.

Note that Problem 1 is based on the assumption that the moving time of a writer is negligible compared to the writing time of a subfield; therefore, the moving of an e-beam writer will not harm the throughput. As a result, we prefer to maximize the minimum distance between two subfields with successive labels (orders) of subfields (i.e., the length of the shortest edge in the Hamiltonian path) to alleviate the problems with successive heating and CD distortions.

On the other hand, if the moving time of a writer is comparable with the writing time of a subfield, a solution

derived by solving Problem 1 may worsen the throughput due to the long total traveling distance. Therefore, another problem formulation which finds a subfield scheduling solution while considering the total traveling distance is given as follows:

Problem 2: Given a photomask layout and the predefined size of a subfield, find a Hamiltonian path containing all points such that the length of the shortest edge in the Hamiltonian path is maximized, the blocked box constraint is satisfied, and the total traveling distance is minimized.

III. EXACT MSTSP ALGORITHM FOR VERTICES ON TWO PARALLEL LINES

In this section, we develop an exact algorithm that optimally solves the MSTSP for a special case. In the case where points are on two parallel lines with an odd number of points on each line, and the points on different lines are aligned, the algorithm can find an optimal solution of the MSTSP under the Euclidean measure in linear time. Fig. 6(a) shows an example. There are an odd number of points on Lines a and b , and the two lines are parallel. We denote as a_i and b_i the i th points on Line a and Line b , respectively. For each point pair (a_i, b_i) , a_i and b_i are aligned; that is, the two points have the same x -coordinate.

In this special case, let V be the input points of size $|V| = 2n$, where $n = 2k + 1$. Then, an optimal Hamiltonian path can be derived by using the following algorithm.

- 1) For each point a_h , connect a_h to points b_i and b_j , where

$$i = \begin{cases} h + (1 + k), & \text{if } h + (1 + k) \leq n \\ h + (1 + k) - n, & \text{if } h + (1 + k) > n, \end{cases} \quad (3)$$

and

$$j = \begin{cases} h + k, & \text{if } h + k \leq n \\ h + k - n, & \text{if } h + k > n. \end{cases} \quad (4)$$

- 2) After connecting the edges, delete the shortest edge in the generated Hamiltonian cycle. The derived path is an optimal Hamiltonian path.

As illustrated in Fig. 6(b), a Hamiltonian cycle is first constructed by applying (3) and (4). After deleting the shortest edge, (a_5, b_3) , on the path, an optimal Hamiltonian path is then generated.

To prove the optimality of the algorithm, we first denote d^* as the length of the shortest edge of an optimal Hamiltonian path, and denote $\text{diam}(R)$ as the diameter of the set $R \subseteq V$. In addition, we say that an edge (u, v) spans n points if there are n points on the boundary of the bounding box of the point u and the point v . In Fig. 6(b), for example, the edge (a_1, b_3) spans six points and the edge (a_1, b_4) spans eight points. We utilize the following lemma and corollary stated in [1]:

Lemma 1: Let $R \subset V$ be a subset of points with $|R| > \lceil \frac{|V|}{2} \rceil$. Then, in any Hamiltonian path on V , there must exist an edge joining two points of R .

Corollary 1: Let $R \subset V$ be a subset of points with $|R| = \lceil \frac{|V|}{2} \rceil + 1$. Then, $d^* \leq \text{diam}(R)$.

These yield the following theorem:

Theorem 2: For a set of points V with $|V| = 2n$ ($n = 2k + 1$ is an odd integer), if the points are on two parallel lines and points on the two lines are aligned, the algorithm finds an

Fig. 6. (a) Special case in which points are on two parallel lines and points on different lines are aligned. (b) Optimal Hamiltonian path generated by using the proposed algorithm.

optimal Hamiltonian path for the MSTSP under the Euclidean measure in linear time.

Proof 1: It is easy to verify that the edges generated by using (3) and (4) consist of a Hamiltonian path. Thus, we only need to show that the upper bound of d^* stated in the corollary is achievable by applying the algorithm.

Let the subset of points spanned by an edge e_i be R_i . Each edge e_i constructed by using (3) or (4) spans at least $\lceil \frac{|V|}{2} \rceil + 1 = 2k + 2$ points, and the length of e_i is equal to $\text{diam}(R_i)$. With the corollary, we have

$$d^* \leq \text{diam}(R_i), \forall i. \quad (5)$$

Since each edge is an upper bound of d^* , the generated Hamiltonian path is optimal. Also, since an optimal Hamiltonian path can be specified in $O(1)$ time without knowing the positions of the points, and the path can be constructed in $O(n)$ time, an optimal Hamiltonian path can be found in linear time by using the proposed algorithm.

IV. GRAPH-BASED SUBFIELD SCHEDULING FOR E-BEAM PHOTOMASK FABRICATION

In this section, we introduce our graph-based algorithm for the subfield scheduling for e-beam photomask fabrication. In Section IV-A, we describe the two-stage algorithm, and in Sections IV-B and IV-C we detail the sub-MSTSP and the preprocessing stages, respectively.

A. Algorithm Flow

In this section, we present the overview of our subfield scheduling algorithm. Fig. 7 shows our subfield scheduling flow.

Given a photomask layout and the predefined size of a subfield, we first construct the blocked boxes for all subfields according to their pattern densities. Our subfield scheduling algorithm consists of two major stages: the sub-MSTSP stage followed by the postprocessing stage.

Fig. 7. Subfield scheduling flow.

In the sub-MSTSP stage, by utilizing the exact algorithm of MSTSP for points on two parallel lines presented in Section III, an optimal sub-Hamiltonian path for each subproblem can be constructed efficiently. First, for each row pair (r_i, r_j) of subfields, we apply a dynamic programming-based algorithm to find a set of subfields S_{ij} such that the arrangement of S_{ij} conforms to the special case mentioned in Section III. Then we compute in linear time the length of the shortest edge of an optimal sub-Hamiltonian path of the set S_{ij} . To simultaneously optimize all sub-Hamiltonian paths in an iteration, we propose a matching-based algorithm to find a best sub-MSTSP pairing result $\mathcal{M}$. Finally, for each $(r_i, r_j) \in \mathcal{M}$, an optimal Hamiltonian path of S_{ij} can be constructed by applying (3) and (4).

If the subproblem sizes are small and the subfields are close to each other, the solution quality of subproblems may not be sufficient; thus we add a postprocessing stage. In this stage, we first connect each isolated subfield to one of the existing Hamiltonian paths. While scattered subfields are eliminated, all sub-Hamiltonian paths are merged into one Hamiltonian path by iteratively using the matching-based method.

For both of the two stages, we also propose methodologies for throughput optimization. As we find a sub-MSTSP pairing in the first stage, we shall try to find a matching result such that the length of the shortest edge in sub-Hamiltonian paths is maximized and the path lengths are minimized. In addition, on the premise that no edge shorter than the existing shortest edge will be created, we propose approaches to minimize the total length of connecting edges in the post processing stage.

In the following, we detail the two stages in Sections IV-B and IV-C, respectively.

B. Sub-MSTSP Stage

Since the MSTSP is conjectured to be NP-hard for points in the plane with the Euclidean measure [1], an algorithm that can efficiently solve the MSTSP for all subfields of a photomask layout might not exist. In addition, since the problem we deal with is the constrained MSTSP, the blocked box constraint must also be taken into account. Thus, we propose

Fig. 8. Subproblem extraction. (a) Blocked boxes of a pair of chosen rows do not overlap vertically. (b) Blocked boxes of a maximum subfield subsequence do not overlap, and the total area of the boxes is maximized.

an algorithm that can solve the constrained MSTSP efficiently by decomposing the problem into a set of subproblems.

In the first stage of the algorithm, we first find the maximum subfield subsequence for each pair of subfield rows by using a dynamic programming-based algorithm. Then, we find a MSTSP pairing by applying one of the two matching-based methods, which are developed according to different objectives of Problem 1 and Problem 2. These processes are detailed in the following two sections.

1) *Maximum Subfield Subsequence Construction*: Due to the inefficiency of solving the MSTSP for all subfields in a photomask layout, we extract subproblems and find the corresponding optimal solutions by using the algorithm presented in Section III. To extract a subproblem conforming to the special case, two rows of subfields are selected first. In addition, since the problem we deal with is the constrained MSTSP, we expect that the blocked boxes of the subfields in an extracted subproblem are not overlapped, and thus the subproblem can be solved optimally with (3) and (4). Thus, the blocked boxes of the first subfield row cannot overlap with those of the second subfield row. As shown in Fig. 8(a), the blocked boxes of the chosen rows do not overlap vertically. The blocked boxes should not overlap horizontally either. We propose a dynamic programming-based algorithm to find a set of nonoverlapping subfields in which the subfields in the first row are aligned with those in the second row. Furthermore, to maximize the subproblem size and to achieve the maximum blocked box uniformity in a subproblem, we choose the subfields whose total area of blocked boxes is maximized: in Fig. 8(b), these subfields are rendered in red. We refer to the set of obtained subfields as the maximum subfield subsequence.

Let $A(a_i)$ denote the area of the blocked box of the i th subfield in row a , and $A(b_i)$ denote the area of the blocked box of the i th subfield in row b . In the dynamic programming formulation, we let $\text{Area}[i] = A(a_i) + A(b_i)$. The problem can thus be formulated as

$$C_0[i] = \begin{cases} 0, & \text{if } i = 1, \\ \max(C_1[i-1], C_0[i-1]), & \text{otherwise.} \end{cases} \quad (6)$$

$$C_1[i] = \begin{cases} \text{Area}[i], & \text{if } i = 1, \\ \text{Area}[i] + C_1[t] & \text{otherwise.} \end{cases} \quad (7)$$

$$\pi[i] = \begin{cases} \text{Nil}, & \text{if } i = 1, \\ t & \text{otherwise.} \end{cases} \quad (8)$$

If the blocked boxes of the i th subfields of the two rows are not chosen, $C_0[i]$ records the maximum total area of

Fig. 9. Length of the shortest edge in an optimal sub-Hamiltonian path can be computed by scanning $\lceil \frac{n}{2} \rceil$ -intervals in linear time.

Fig. 10. Best sub-MSTSP pairing deviation. (a) Four rows of subfields. (b) Compatible graph construction. (c) Modified compatible graph for use with maximum cardinality bottleneck matching algorithm.

non-overlapping blocked boxes (likewise, $C_1[i]$ records the maximum total area for those of the two rows that are chosen). $t < i$ is an index such that $C_1[t]$ is maximized and the blocked boxes of a_t and b_t do not overlap with those of a_i and b_i . The maximum subfield subsequence can be derived by finding the largest i such that $C_1[i] \geq C_0[i]$ and tracing back the indexes with (8). Note that if the number of selected subfields in a row is even, we omit subfields a_i and b_i with the minimum $Area[i]$.

2) *Best Sub-MSTSP Pairing Derivation*: After generating the maximum subfield subsequence of an extracted subproblem, we can compute the length of the shortest edge in an optimal sub-Hamiltonian path of the subproblem. Suppose that the size of a subproblem is $|V| = 2n$. Let the l -interval be an interval containing l subfields in a row. By scanning $\lceil \frac{n}{2} \rceil$ -intervals from left to right of a row, the length of the shortest $\lceil \frac{n}{2} \rceil$ -interval d_s can be found. Then, the length of the shortest edge l_s in an optimal sub-Hamiltonian path is computed as follows:

$$l_s = \sqrt{d_s^2 + d_r^2} \quad (9)$$

where d_r is the distance between the two subfield rows. As shown in Fig. 9, for a subproblem of size $|V| = 10$, the length of the shortest edge can be computed by scanning the three 3-intervals.

For each pair of subfield rows, we find the maximum subfield subsequence and compute the length of the shortest edge of an optimal sub-Hamiltonian path. To simultaneously optimize each subproblem, we need a pairing for all subfield rows that maximizes the length of the shortest edge among all sub-Hamiltonian paths. To solve this best sub-MSTSP pairing problem, we propose a matching-based algorithm.

First, we construct a compatible graph in which a point represents a subfield row, and two points are connected

Fig. 11. Iterative sub-Hamiltonian path construction. (a)–(c) First iteration of sub-Hamiltonian path construction. (d)–(f) Second iteration of sub-Hamiltonian path construction.

through an edge if the blocked boxes of the two rows are nonoverlapped vertically. Each edge is associated with an edge weight, which is the length of the shortest edge, that is, l_s , in the sub-Hamiltonian path of the corresponding maximum subfield subsequence. Fig. 10 illustrates an example. Four rows of subfields are shown in Fig. 10(a); suppose that the blocked boxes of each row only overlap vertically with those of adjacent rows. Thus, the corresponding compatible graph is shown in Fig. 10(b). By utilizing the compatible graph, the best sub-MSTSP pairing problem can be solved by finding a maximum matching $\mathcal{M}$ in which the smallest edge weight is maximized. We apply the algorithm proposed in [11] to solve the maximum cardinality bottleneck matching problem to find a matching solution by simply multiplying the edge weights by -1 and slightly modifying the algorithm. Hence, the best pairing of the example in Fig. 10 can be found by using the algorithm in [11] and the compatible graph in Fig. 10(c).

For each pair of subfield rows in $\mathcal{M}$, an optimal sub-Hamiltonian path is then constructed with (3) and (4) in linear time. We iteratively apply the above process for those subfields not contained in any sub-Hamiltonian paths constructed in previous iterations. Fig. 11 shows the iterative sub-Hamiltonian path construction. Fig. 11(a)–(c) shows the first iteration. The best sub-MSTSP pairing solution is $\{(r_1, r_3), (r_2, r_4)\}$, as shown in Fig. 11(a). Then, an optimal sub-Hamiltonian path according to the maximum subfield subsequence of each row pair is constructed, as illustrated in Fig. 11(b) and (c). Note that the maximum subfield subsequence of a row pair is found by using the dynamic programming-based algorithm in advance. In the second iteration [see Fig. 11(d)–(f)], the row pairing problem is then solved again for those unprocessed subfields, and other subpaths are constructed accordingly, as shown in Fig. 11(e) and (f).

3) *Sub-MSTSP Pairing Derivation Considering Throughput*: To consider throughput in the sub-MSTSP stage, we have to not only maximize the shortest edge length in all sub-Hamiltonian paths but also minimize the total path length. This can be achieved by modifying the matching-based algorithm proposed in Section IV-B2. First, similar to finding the shortest edge in an optimal sub-Hamiltonian path, all edge lengths in the path can be computed by scanning $\lceil \frac{n}{2} \rceil$ -intervals and $\lceil \frac{n}{2} \rceil + 1$ -intervals. As shown in Fig. 9, each edge in the path is either covered by an $\lceil \frac{n}{2} \rceil$ -interval or covered by an

$\lceil \frac{n}{2} \rceil + 1$ -interval. Thus, the total length L of an optimal sub-Hamiltonian path can be computed by using the following equation:

$$L = 2 \sum_i \sqrt{d_i^2 + d_r^2} + 2 \sum_j \sqrt{d_j'^2 + d_r^2} - l_s \quad (10)$$

where d_i is the length of an $\lceil \frac{n}{2} \rceil$ -interval and d_j' is the length of an $\lceil \frac{n}{2} \rceil + 1$ -interval. The equation computes the total length of an optimal Hamiltonian cycle, and then subtracts the length of the shortest edge to get the total length of an optimal Hamiltonian path. To simultaneously optimize the shortest edge length and the total path length, we change the weight of each edge in a compatible graph from l_s into the following equation:

$$\text{weight} = l_s - \alpha \cdot L. \quad (11)$$

Then, a sub-MSTSP pairing solution considering total path lengths can be generated by solving the maximum cardinality bottleneck matching problem [11]. Note that the user-specified parameter α may significantly affect the final scheduling solution; as α decreases, the shortest edge length and the total traveling distance tend to be larger.

C. Postprocessing Stage

If the subproblem sizes are small and the subfields are close to each other such that the length of the shortest $\lceil \frac{n}{2} \rceil$ -interval d_s is short, the solution quality of subproblems may not be sufficient; thus the subfield scheduling algorithm flow enters the second stage, the postprocessing stage. In this stage, the main objective is to maintain the good solution quality derived from the sub-MSTSP stage while constructing a complete subfield scheduling solution. We first eliminate scattered subfields by connecting each of them to an endpoint of existing sub-Hamiltonian paths. Then, we merge all sub-Hamiltonian paths into one Hamiltonian path, which is the final subfield scheduling solution. Also, to consider throughput in the postprocessing stage, we propose approaches to minimize the total length of connecting edges with the edges not shorter than the shortest edge length derived in the previous sub-MSTSP stage. We detail the scattered subfield elimination, the sub-Hamiltonian paths merging, and the extended methods for throughput consideration in the following subsections.

1) *Isolated Subfield Elimination*: At the beginning of this stage, each subfield is either contained in a sub-Hamiltonian path, or left as an isolated subfield. We sequentially connect each isolated subfield to an endpoint of existing sub-Hamiltonian paths such that the distance between the subfield and the endpoint is not shorter than the shortest edge in existing sub-Hamiltonian paths. Since the elimination sequence determines the final positions of sub-Hamiltonian paths, the sequence may greatly affect the solution quality in the postprocessing stage. Fig. 12 gives an example. There are two sub-Hamiltonian paths, path 1 and path 2, with their endpoints, 1a, 1b, 2a, and 2b, and four isolated subfields are left, S_1, S_2, S_3 , and S_4 . Fig. 12(a)–(d) shows the process in which the isolated subfields are eliminated from the margins to the center of the chip; that is, the isolated subfields are

Fig. 12. Scattered point elimination. (a)–(d) Eliminating the isolated points in the order $\langle S_1, S_2, S_3, S_4 \rangle$ (from the margins to the center). (e)–(h) Eliminating the isolated points in the order $\langle S_4, S_3, S_2, S_1 \rangle$ (from the center to the margins).

connected to endpoints of sub-Hamiltonian paths with the sequence $\langle S_1, S_2, S_3, S_4 \rangle$. In the example, we simply connect each isolated subfield to the farthest endpoint of path 1 and path 2. After connecting all isolated subfields, the four endpoints are close to each other, as shown in Fig. 12(d). Close endpoints may cause very short edges as we merge sub-Hamiltonian paths. It can easily be seen in Fig. 12(d) that the length of the edge connecting paths 1 and 2 must be one-unit long. On the other hand, if we eliminate the isolated subfields with the sequence $\langle S_4, S_3, S_2, S_1 \rangle$, as illustrated in Fig. 12(e)–(h), the endpoints of the final sub-Hamiltonian paths are much more scattered, as shown in Fig. 12(h). To merge paths 1 and 2, we can connect the endpoints 1b and 2a such that the length of the connecting edge is three units, which is much better than the previous one.

To maintain the solution quality of the sub-Hamiltonian paths merging, therefore, we eliminate isolated subfields from the center to the margins of a chip such that the final endpoints of sub-Hamiltonian paths tend to lie on the margins of the chip. This heuristic maintains the solution quality derived in the sub-MSTSP stage in most cases.

2) *Sub-Hamiltonian Paths Merging*: After eliminating all isolated points, the last step is to merge sub-Hamiltonian paths into one Hamiltonian path which contains all subfields in the given photomask layout. By observing that the operation of merging sub-Hamiltonian paths corresponds to connecting the endpoints of those paths, we transform the problem into a matching problem again.

Fig. 13 illustrates an example. There are four sub-Hamiltonian paths in Fig. 13(a), and the endpoints of sub-Hamiltonian path i are denoted as i_1 and i_2 . A compatible graph is then constructed, in which a vertex denotes an endpoint of a sub-Hamiltonian path and an edge between two vertices means that the two corresponding endpoints can be connected together (i.e., belonging to different sub-Hamiltonian paths), as shown in Fig. 13(b). The weight of edge (i, j) is set to be the distance between two corresponding endpoints i and j . To maximize the length of the shortest edge connecting

Fig. 13. Sub-Hamiltonian paths merging. (a) Four sub-Hamiltonian paths. (b) Compatible graph corresponding to the four sub-Hamiltonian paths. (c) Matching solution. (d) Resulting graph with two cycles. (e) Updated compatible graph after breaking the cycles.

sub-Hamiltonian paths, the maximum cardinality bottleneck matching algorithm is then applied again to obtain a matching solution $\mathcal{M}$. Connecting the endpoints of sub-Hamiltonian paths according to $\mathcal{M}$, the resulting graph is composed of cycles. As shown in Fig. 13(c) and (d), the graph corresponding to the matching result $\mathcal{M} = \{(a_1, b_1), (a_2, b_2), (c_1, d_2), (c_2, d_1)\}$ consists of two cycles. We break a cycle by deleting the shortest edge in the cycle, and thus the sub-Hamiltonian paths in the cycle are merged into one longer sub-Hamiltonian path. After deriving a set of longer sub-Hamiltonian paths, an updated compatible graph is constructed as illustrated in Fig. 13(e). The above merging process is performed iteratively until all sub-Hamiltonian paths are merged into one Hamiltonian path. Since during each iteration's merging process the number of sub-Hamiltonian paths is cut at least in half, the number of iterations required for the merging process is less than or equal to $\lg n$ times, where n is the number of sub-Hamiltonian paths constructed in the sub-MSTSP stage.

3) *Postprocessing with Throughput Consideration*: To optimize the throughput, in the postprocessing stage, we try to minimize both the lengths of connecting edges used to eliminate isolated subfields and the lengths of connecting edges used to merge sub-Hamiltonian paths. However, the objective of the postprocessing stage should not be violated as well; that is, the length of connecting edges should not be shorter than the shortest edge length derived in the sub-MSTSP stage to maintain the good solution qualities of existing sub-Hamiltonian paths.

As eliminating isolated subfields, we still use the heuristic proposed in Section IV-C1 to eliminate isolated subfields from the center to the margin of a chip. To minimize the total traveling distance, however, each isolated subfield is connected to an endpoint of existing sub-Hamiltonian paths such that the connecting edge length is the shortest one but is longer than the shortest edge in existing subpaths.

In the last step of paths merging, the objective is to find a set of connecting edges whose total edge length is minimized, but

Fig. 14. Sub-Hamiltonian paths merging with throughput consideration. (a) Compatible graph with edge weights. (b) Maximum bottleneck matching solution. (c) Updated compatible graph after deleting edges with weights smaller than the shortest length of existing edges. (d) Minimum-weight perfect matching solution.

the length of each connecting edge cannot be shorter than the shortest edge in existing subpaths as well. To achieve this goal, we use a two-pass method to derive a matching result in each matching iteration. First, given an input compatible graph with edge weights, each of which is set as the distance between two endpoints, maximum cardinality bottleneck matching is applied to derive a matching solution in which the minimum edge weight is maximized. If the shortest edge length in the corresponding matching solution is shorter than the shortest edge in existing subpaths, then the matching solution is adopted and the paths are merged accordingly. Otherwise, the compatible graph is updated by removing the edges with weights smaller than the shortest edge length in existing subpaths. Note that it can still be guaranteed that we can find a perfect matching solution in the updated compatible graph because the edges in the previous matching solution are not deleted. Then, to minimize the total length of connecting edges, we apply a minimum-weight maximum matching algorithm to find a matching solution. An example of the two-pass matching algorithm is shown in Fig. 14. In this example, we assume that the shortest edge length in existing subpaths is 2. Given a compatible graph shown in Fig. 14(a), a maximum bottleneck matching algorithm is first applied, as shown in Fig. 14(b). Since all the edges in the matching solution are not shorter than 2, the compatible graph is updated by deleting edges shorter than 2, as illustrated in Fig. 14(c). After that, a minimum-weight maximum matching algorithm is then applied as shown in Fig. 14(d). Compared to the matching solution in Fig. 14(b), the edge lengths in the matching solution in Fig. 14(d) are shorter, and thus the total traveling distance can be reduced.

V. EXPERIMENTAL RESULTS

Our algorithm was implemented in the C++ programming language on a 2.13 GHz Linux workstation with 16 GB mem-

TABLE II
STATISTICS OF THE BENCHMARK CIRCUITS

Circuit	Size (μm^2)	No. of Layers	No. of Nets	No. of Pins
Mcc1	$45\,000 \times 39\,000$	4	1694	3101
Mcc2	$152\,400 \times 152\,400$	4	7541	25\,024
Struct	4903×4904	3	3551	5471
Primary1	7522×4988	3	2037	2941
Primary2	$10\,438 \times 6488$	3	8197	11\,226
S5378	4330×2370	3	3124	4734
S9234	4020×2230	3	2774	4185
S13207	6590×3640	3	6995	10\,562
S15850	7040×3880	3	8321	12\,566
S38417	$11\,430 \times 6180$	3	21\,035	32\,210
S38584	$12\,940 \times 6710$	3	28\,177	42\,589

TABLE III
SUBFIELD SCHEDULING RESULTS OF OUR GRAPH-BASED SUBFIELD SCHEDULING ALGORITHM FLOW

Circuit	No. of subfields											
	32×32			64×64			128×128			256×256		
	Min.	Avg.	CPU (s)	Min.	Avg.	CPU (s)	Min.	Avg.	CPU (s)	Min.	Avg.	CPU (s)
Mcc1	18.868	21.845	0.02	32.016	42.679	0.46	64.070	83.236	15.99	128.004	165.815	33.90
Mcc2	12.166	22.072	0.01	23.770	43.722	0.68	51.865	86.652	24.08	128.082	168.988	414.73
Struct	19.209	22.373	0.02	32.203	43.648	0.29	64.885	84.741	4.35	128.144	167.653	8.58
Primary1	16.156	21.621	0.03	32.016	39.965	0.33	64.008	77.578	1.46	128.160	154.234	3.32
Primary2	19.417	22.459	0.03	35.903	43.627	0.24	64.078	83.525	7.36	128.156	164.165	50.46
S5378	19.209	22.454	0.03	33.051	43.432	0.51	64.031	83.050	2.93	128.141	162.937	12.58
S9234	18.439	21.952	0.02	32.203	42.843	0.38	64.031	82.430	2.62	128.818	162.078	6.32
S13207	21.260	22.666	0.03	33.838	45.028	0.38	64.073	88.778	4.27	128.550	169.550	27.37
S15850	19.209	22.519	0.02	33.838	44.807	0.49	64.031	88.053	5.56	128.144	170.343	30.92
S38417	17.889	22.701	0.03	32.016	45.022	0.43	64.141	89.445	7.05	134.380	175.806	43.07
S38584	21.260	22.656	0.03	33.242	45.151	0.42	64.070	89.375	4.99	128.160	174.855	104.01
Avg.	18.462	22.302	0.03	32.190	43.629	0.42	63.026	85.169	7.33	128.794	166.948	66.84

Min.: shortest length between two successive subfields in a subfield scheduling solution. Avg.: average length among all successive subfields.

TABLE IV
COMPARISON OF THE SUBFIELD SCHEDULING RESULTS BETWEEN GSS-M (ADOPTED FROM [3]) AND OUR GRAPH-BASED SUBFIELD SCHEDULING ALGORITHM FLOW

No. of subfields	GSS-m (adopted from [3])			Ours		
	Minimum	Average	CPU (s)	Minimum	Average	CPU (s)
32×32	3.16	24.36	33.56	18.46	22.30	0.03
64×64	3.61	48.84	> 1 h	32.19	43.63	0.42
128×128	2.00	96.43	> 1 h	63.03	85.17	7.33
256×256	1.00	164.92	> 1 h	128.79	166.95	66.84

ory. We adopted the LEDA package [17] for the minimum-weight maximum-matching algorithm. The experiment was based on the 11 benchmark circuits used in [4]. Table II lists the set of benchmark circuits. We routed the circuits by using a two-pass bottom-up gridless router, and we used the metal-1 of each circuit as the input photomask layout. A layout is first divided into 32×32 , 64×64 , 128×128 , and 256×256 subfields. The pattern density was analyzed, and the blocked box of each subfield was constructed. We let the blocked box of a subfield being written cover 15% of the total number of subfields in its neighboring region if the pattern density of the subfield was one, and let the area of a blocked box equal the area of the corresponding subfield if the pattern density was zero.

A. Comparison of Subfield Scheduling Results

The subfield scheduling results of the benchmark circuits with different numbers of subfields are listed in Table III.

In the table, “Min.” gives the shortest length between two successive subfields in a subfield scheduling solution, and “Avg.” gives the average length among all successive subfields. Note that the lengths are measured in unit subfield edge length.

We first compared our graph-based algorithm with the modified greedy algorithm for subfield scheduling (denoted as GSS) proposed in [3]. This algorithm first randomly generates a subfield scheduling order and then iteratively improves the solution by swapping pairs of subfields. We modified the cost function in [3] from temperature measurement to distance measurement and denoted GSS with the modified cost function as GSS-m. The cost function is given by

$$\alpha \cdot D_{\min} + \beta \cdot D_{\text{avg}} \quad (12)$$

where $D_{\min}$ is the minimum distance and D_{avg} is the average distance between two successive subfields. We set $\alpha \gg \beta$ to normalize the two values. Since GSS does not consider the blocked box constraint, we simply ignore whether the

TABLE V
COMPARISON OF THERMAL RESULTS BETWEEN GSS [3] AND OUR ALGORITHM FLOW FOR CASES OF SIZES 32×32 AND 64×64

Circuit	No. of subfields											
	32×32						64×64					
	Cost			Max. Temp.			Cost			Max. Temp.		
	GSS [3]	Ours	Improv. (%)	GSS [3]	Ours	Improv. (%)	GSS [3]	Ours	Improv. (%)	GSS [3]	Ours	Improv. (%)
Mcc1	103.00	89.20	13.41	104.66	97.11	7.22	124.02	70.36	43.27	132.81	73.16	44.92
Mcc2	119.43	94.75	20.67	123.96	100.09	19.26	145.26	83.05	42.83	158.38	88.46	44.14
Struct	98.34	95.07	3.33	99.76	99.48	0.28	125.97	91.82	27.11	133.11	95.98	27.89
Primary1	151.76	148.82	1.94	163.37	165.52	-1.31	160.75	132.15	17.79	177.39	150.08	15.39
Primary2	127.05	113.77	10.46	138.49	128.27	7.38	153.09	118.78	22.41	165.71	131.18	20.83
S5378	107.22	101.66	5.19	109.46	106.75	2.47	122.89	87.05	29.16	130.71	94.86	27.43
S9234	108.10	100.42	7.10	111.58	107.09	4.03	122.27	78.69	35.64	131.53	79.89	39.26
S13207	103.92	99.52	4.23	106.20	111.11	-4.62	125.13	99.09	20.81	133.75	122.97	15.54
S15850	115.85	109.99	5.06	123.31	124.89	-1.28	124.85	97.33	22.05	134.95	107.49	20.35
S38417	89.74	74.00	17.54	91.61	78.90	13.88	114.42	75.86	33.71	119.89	78.60	34.44
S38584	90.92	75.02	17.48	93.66	82.01	12.43	126.87	98.82	22.11	142.17	118.50	16.65
Avg.			9.67			5.43			28.81			27.89

Cost: the cost evaluated by (13). Max. Temp.: maximum critical temperature in a scheduling order. Improv.: improvement ratio.

TABLE VI
COMPARISON OF THERMAL RESULTS BETWEEN GSS [3] AND OUR ALGORITHM FLOW FOR CASES OF SIZES 128×128 AND 256×256

Circuit	No. of subfields											
	128×128						256×256					
	Cost			Max. Temp.			Cost			Max. Temp.		
	GSS [3]	Ours	Improv. (%)	GSS [3]	Ours	Improv. (%)	GSS [3]	Ours	Improv. (%)	GSS [3]	Ours	Improv. (%)
Mcc1	101.47	60.21	40.66	108.52	64.06	40.97	83.02	50.37	39.33	88.08	55.17	37.36
Mcc2	126.90	73.71	41.91	138.79	80.11	42.28	97.63	57.99	40.61	104.07	61.47	40.93
Struct	102.85	81.91	20.36	109.92	88.78	19.23	80.09	58.19	27.35	85.76	60.18	29.83
Primary1	135.65	98.88	27.11	150.15	108.43	27.79	105.67	76.90	27.23	117.17	84.45	27.93
Primary2	140.23	116.80	16.71	153.82	130.24	15.33	116.80	96.72	17.20	126.61	104.73	17.28
S5378	102.55	65.73	35.90	109.29	67.85	37.91	71.19	42.82	39.85	74.85	44.06	41.13
S9234	99.21	70.45	28.99	106.55	79.66	25.24	71.45	43.55	39.05	75.69	45.14	40.36
S13207	107.17	69.18	35.45	115.78	71.48	38.26	73.16	43.90	40.00	77.55	45.19	41.72
S15850	96.73	68.97	28.69	101.33	73.16	27.80	79.86	55.66	30.31	85.88	62.46	27.28
S38417	103.48	69.11	32.22	109.18	72.10	33.96	81.69	54.49	33.30	82.29	56.14	31.78
S38584	75.45	55.66	26.23	80.01	62.46	21.94	85.46	65.82	22.98	92.77	74.62	19.56
Avg.			30.48			30.07			32.47			32.29

scheduling order generated from GSS is feasible or not. In addition, since GSS does not consider the pattern density in each subfield, the scheduling results are independent of different circuit layouts. Thus, we compared their scheduling results with our average results derived from Table III.

Table IV shows the comparison results. Although the average lengths derived from our algorithm are slightly shorter than those derived from GSS-m, the minimum lengths obtained from our algorithms are much longer than those of GSS-m. This is because the objective of our algorithm is to maximize the minimum distance between any two successive subfields such that the problem with successive heating and thus CD distortion can be alleviated. Furthermore, the CPU time required by our algorithm is much less than that required by GSS-m. These results show that our algorithm is effective and efficient in finding a good subfield scheduling solution for e-beam photomask fabrication.

B. Comparison of Thermal Results

To verify that the subfield scheduling solution obtained from our algorithm flow indeed mitigates the heating problem effectively, we used the thermal model introduced in Section II-A to calculate the maximum temperature of each

subfield in a scheduling order, which is defined as the critical temperature of a subfield. Then, we computed the average critical temperature and the maximum critical temperature for each scheduling solution. Also, we implemented the original greedy subfield scheduling (GSS) algorithm developed in [3] to make a comparison. The cost function is

$$\alpha \cdot T_{\max} + (1 - \alpha) \cdot T_{\text{avg}} \quad (13)$$

where $T_{\max}$ is the maximum critical temperature and T_{avg} is the average critical temperature of all subfields ($\alpha = 0.5$, the same setting as in [3]). In each iteration, the greedy algorithm generates the next scheduling order by swapping the order of the subfield with the maximum critical temperature and some other subfield such that the cost in (13) is minimized. For this experiment, we set the initial temperature of each subfield to zero, and set the maximum running time to one hour. In addition, the maximum raised temperature of a subfield due to its own writing process was normalized with the average subfield density for each testcase.

The experimental results are shown in Tables V and VI. Column “Cost” lists the cost evaluated by (13), column “Max. Temp.” gives the maximum critical temperature in a scheduling order, and column “Improv.” shows the improvement ratio of

Fig. 15. Total traveling distance decreases as α increases.

Fig. 16. Minimum distance between any successive writing subfields decreases as α increases.

our results over those derived from the greedy algorithm. From the two tables, most of the scheduling solutions generated by our algorithm flow yield better solutions than those obtained from GSS for cases of size 32×32 . For other cases, moreover, our algorithm approximately achieves 30% temperature reductions on average compared with GSS. This substantial improvement may result from the expensive computation of the cost function which requires $O(|S|^2)$ time to evaluate, and thus it requires $O(|S|^3)$ time to perform each swap operation in GSS, where $|S|$ is the number of subfields. As a result, GSS can only perform a small number of iterations to improve the solution quality for the cases of sizes larger than 32×32 within an acceptable running time. These results show that our algorithm can effectively mitigate the heating problem and that it scales well with the number of subfields.

C. Tradeoff with Throughput

While considering throughput in our subfield scheduling flow, we used the modified cost function (11) for solving the best-MSTSP pairing problem. As mentioned in Section IV-B3, the parameter α may significantly affect the final solution quality: if α equals 0, our algorithm finds a subfield scheduling solution without the throughput consideration; that is, the algorithm maximizes the minimum distance between any successive writing subfields without considering the total traveling distance. On the other hand, if α is larger than zero, the cost function considers the total traveling distance in a sub-Hamiltonian path as a penalty. In addition, as α increases, the algorithm tends to reduce the total traveling distance to improve the throughput.

Fig. 17. Maximum temperature rises as α increases.

Figs. 15 and 16 show the variations of the total traveling distance and the minimum distance between any successive writing subfields with α for the circuit Primary1 of size 64×64 . As shown in the figures, as α increases, the total traveling distance and the minimum distance in a subfield scheduling solution decrease. The maximum temperature, however, tends to increase as the minimum distance decreases. As shown in Fig. 17, as α increases, the maximum temperature significantly rises due to the shortening of the minimum distance. These results show the tradeoff between the throughput and the heating effect in the subfield scheduling problem.

VI. CONCLUSION

This paper presented a graph-based algorithm that deals with the subfield scheduling problem by solving the constrained MSTSP. Compared with the prior work, our algorithm elegantly solved the problem by decomposing it into subproblems and solving the subproblems optimally. To optimize the throughput in EBL, we also considered the total traveling distance in our algorithm flow. Experimental results show that our algorithm was effective and efficient in finding good subfield scheduling solutions that alleviate the successive heating problem (and thus reduce CD distortion) for e-beam photomask fabrication.

ACKNOWLEDGMENTS

The authors would like to thank C.-W. Lee, Y.-H. You, and Prof. C.-H Kuan from the Thermal Image Lab of National Taiwan University for conducting the experiment of subfield scheduling on 1-D lines and providing the experimental data and the SEM images. They would also like to thank the anonymous TCAD reviewers for their constructive comments.

REFERENCES

- [1] E. M. Arkin, Y.-J. Chiang, J. S. B. Mitchell, S. S. Skiena, and T.-C. Yang, "On the maximum scatter travelling salesperson problem," *SIAM J. Comput.*, vol. 29, no. 2, pp. 515–544, 1999.
- [2] S. Babin, A. B. Kahng, I. I. Mandoiu, and S. Muddu, "Subfield scheduling for throughput maximization in electron-beam photomask fabrication," *Proc. SPIE*, vol. 5037, pp. 934–942, Jun. 2003.
- [3] S. Babin, A. B. Kahng, I. I. Mandoiu, and S. Muddu, "Resist heating dependence on subfield scheduling in 50 kV electron beam maskmaking," *Proc. SPIE*, vol. 5130, pp. 718–726, Aug. 2003.

- [4] J. Cong, J. Fang, M. Xie, and Y. Zhang, "MARS—a multilevel full-chip gridless routing system," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 24, no. 3, pp. 382–394, Mar. 2005.
- [5] E. A. Eakkennes, A. D. Wiersma, M. Hoving, N. Venema, S. Woutersen, T. van de Peut, M. Sanderse, and M. J. Wieland, "Demonstration of real time pattern correction for high throughput maskless lithography," *Proc. SPIE*, vol. 7970, p. 79701A, Mar. 2011.
- [6] S.-Y. Fang, W.-Y. Chen, and Y.-W. Chang, "Graph-based subfield scheduling for electron-beam photomask fabrication," in *Proc. Int. Symp. Physical Design*, Mar. 2012, pp. 9–16.
- [7] R. Freed, J. Sun, A. Brodie, M. McCord, K. Ronse, L. Haspeslagh, and B. Vereecke, "Demonstration of lithography patterns using reflective e-beam direct write," *Proc. SPIE*, vol. 7970, p. 79701T-1, Mar. 2011.
- [8] R. Freed, T. Gubiotti, J. Sun, F. Kidwingira, J. Yang, U. Ummethala, L. C. Hale, J. J. Hench, S. Kojima, W. D. Mieher, C. F. Bevis, S.-J. Lin, and W.-C. Wang, "Reflective electron-beam lithography: Progress toward high-throughput production capability," *Proc. SPIE*, vol. 8323, p. 83520H, Mar. 2012.
- [9] A. Fujimura, "Beyond light: The growing importance of e-beam," in *Proc. Tutorial Int. Conf. Comput.-Aided Des.*, Nov. 2009.
- [10] A. Fujimura, "Design for e-beam: Getting the best wafers without the exploding mask costs," in *Proc. Keynote Int. Symp. Quality Electronic Design*, Mar. 2010.
- [11] H. N. Gabow and R. E. Tarjan, "Algorithms for two bottleneck optimization problems," *J. Algorithms*, vol. 9, no. 3, pp. 411–417, 1988.
- [12] P. Hudek, "Fogging and heating effects in e-beam lithography," Invited Lecture at the BEA Meeting in Paris, 2011.
- [13] Z. Jiang, F. Zhao, W. Jing, P. D. Prewett, and K. Jiang, "Characterization of line edge roughness and line width roughness of nano-scale typical structures," in *Proc. Int. Conf. Nano/Micro Engineered Molecular Syst.*, 2009, pp. 299–303.
- [14] N. Kuwahara, H. Nakagawa, M. Kurihara, N. Hayashi, H. Sano, E. Murata, T. Takikawa, and S. Nohuchi, "Preliminary evaluation of proximity and resist heating effects observed in high acceleration voltage e-beam writing for 180 nm-and-beyond rule reticle fabrication," *Proc. SPIE*, vol. 3784, pp. 115–125, Sep. 1999.
- [15] J. C. Lagarias, "Well-spaced labelings of points in rectangular grids," *SIAM J. Discrete Math.*, vol. 13, no. 4, pp. 521–534, 2000.
- [16] D. K. Lam, E. D. Liu, M. C. Smayling, and T. Prescop, "E-beam to complement optical lithography for 1D layouts," *Proc. SPIE*, vol. 7970, p. 797011, Mar. 2011.
- [17] The LEDA Package [Online]. Available: <http://www.algorithmic-solutions.com/leda>
- [18] E. D. Liu and T. Prescop, "Optimization of e-beam landing energy for EBDW," *Proc. SPIE*, vol. 7970, p. 79701S, Mar. 2011.
- [19] H. C. Pfeiffer, "New prospects for electron beams as tools for semiconductor lithography," *Proc. SPIE*, vol. 7378, p. 737802, May 2009.
- [20] S. Rizvi, *Handbook of Photomask Manufacturing Technology*. New York: Taylor & Francis, 2005.
- [21] H. Sakurai, T. Abe, M. Itoh, A. Kumagai, H. Anze, and I. Higashikawa, "Resist heating effect on 50kV EB mask writing," *Proc. SPIE*, vol. 3748, pp. 126–136, Aug. 1999.
- [22] M. J. Wieland, G. de Boer, G. F. ten Berge, M. van Kervinck, R. Jager, J. J. M. Peijster, E. Slot, S. W. H. K. Steenbrink, T. F. Teepen, and B. J. Kampherbeek, "MAPPER: High throughput maskless lithography," *Proc. SPIE*, vol. 7637, p. 76370F, Mar. 2010.

Wei-Yu Chen received the B.S. degree in computer science from National Chiao Tung University, Hsinchu, Taiwan, in 2009 and the M.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 2011.

His current research interests include physical design and design for manufacturability.

Yao-Wen Chang (S'94–A'96–M'96–SM'12–F'13) received the B.S. degree from National Taiwan University (NTU), Taipei, Taiwan, in 1988, and the M.S. and Ph.D. degrees from the University of Texas, Austin, USA, in 1993 and 1996, respectively, all in computer science.

He is currently an Associate Dean of the College of Electrical Engineering and Computer Science, the Director of the Graduate Institute of Electronics Engineering, and a Distinguished Professor of the Department of Electrical Engineering, NTU. He has been working closely with the industries related to manufacturability and very large scale integration physical design. He has co-edited one textbook on electronic design automation and co-authored one book on routing and over 220 ACM/IEEE conference and journal papers in these areas. His current research interests include very large scale integration physical design and design for manufacturability.

Dr. Chang was a First Place winner of four recent contests: the 2012 ACM/IEEE DAC Placement Contest, the 2012 ACM ISPD Discrete Gate Sizing Contest, the 2011 IEEE CEDA PATMOS Timing Analysis Contest, and the 2009 ACM ISPD Clock Tree Synthesis Contest. He is a five time winner of the ACM ISPD contests on placement, global routing, clock network synthesis, and discrete gate sizing, and a recipient of six Best Paper Awards (ICCD, etc.) and 21 Best Paper Award nominations from DAC (five times), ICCAD (four times), ISPD (six times), etc. in the past ten years. He has received many research and teaching awards, such as the Distinguished Research Award (highest honor) from the National Science Council of Taiwan (twice), the IBM Faculty Awards (three times), CIEE Distinguished EE Professorship, MXIC Young Chair Professorship, and Excellent Teaching Awards from NTU (seven times). He is currently an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, and an editor of the IEEE DESIGN AND TEST OF COMPUTERS and the *Journal of Information Science and Engineering*. He has served as the General and Steering Committee Chairs of ISPD, and the Program Chair of ASP-DAC, FPT, ICCAD, and ISPD. He is on the IEEE CEDA Executive Committee, the ICCAD Executive Committee, the ASP-DAC Steering Committee, and the ACM/SIGDA Physical Design Technical Committee. He has served on technical program committees of major EDA conferences, including ASP-DAC, DAC, DATE, FPL, FPT, GLSVLSI, ICCAD, ICCD, ISPD, SLIP, SOCC, and VLSI-DAT. He has served as an independent Board Director of Genesys Logic, a Technical Consultant of Faraday, MediaTek, and RealTek, a Chair of the EDA Consortium of the Ministry of Education, Taiwan, and a member of the Board of Governors of the Taiwan IC Design Society.

Shao-Yun Fang received the B.S. degree in electronics engineering from National Taiwan University, Taipei, Taiwan, in 2008, where she is currently pursuing the Ph.D. degree from the Graduate Institute of Electronics Engineering.

Her current research interests focus on physical design and design for manufacturability.