

# Proximity Effect Aware Detailed Placement in Electron Beam Lithography

Yuhang Chen<sup>1</sup>, Zhipeng Huang<sup>2</sup>, Xiongfeng Chen<sup>3</sup>, Jianli Chen<sup>2</sup>, and Wenxing Zhu<sup>2</sup>

<sup>1</sup>College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China

<sup>2</sup>Center for Discrete Mathematics and Theoretical Computer Science, Fuzhou University, Fuzhou 350108, China

<sup>3</sup>Fujian Key Laboratory of Information Processing and Intelligent Control, Minjiang University, Fuzhou 350108, China

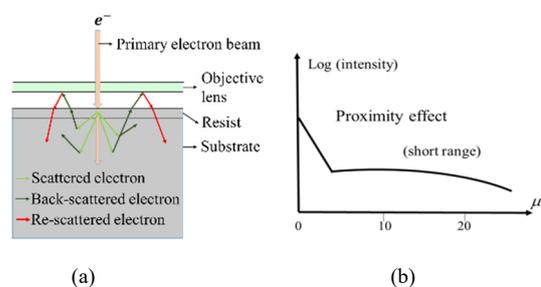
**Abstract.** Proximity effect is one of the most tremendous consequences that produces unacceptable exposures during electron beam lithography (EBL), and thus distorting the layout pattern. In this paper, we propose the first work which considers the proximity effect during layout stage. We first give an accurate evaluation scheme to estimate the proximity effect by fast Gauss transform. Then, we devote a proximity effect aware detailed placement objective function to simultaneously consider wavelength, density and proximity effect. Furthermore, cell swapping and cell matching based methods are used to optimize the objective function such that there is no overlap among cells. Compared with a state-of-the-art work, experimental result shows that our algorithm can efficiently reduce the proximity variations and maintain high wavelength quality at a reasonable runtime.

## 1 Introduction

As the feature sizes keep shrinking, the complexity of a circuit design increases dramatically. The resolution of traditional optical lithography is limited due to light diffraction, thus, next-generation lithography technologies are expected to have lower cost and higher resolution. Among them, electron-beam lithography (EBL) is a most promising one, which suits for the fabrication of integrated circuits (IC) with small dimensions and high packing densities.

Since EBL can print custom patterns directly without mask, it is capable to generate patterns accurately. However, there are still some critical challenges inherent in EBL to be settled before it is ready for mass production. In EBL manufacturing process, the electron gun emits electrons to write patterns on wafers directly through a set of lenses and apertures. As shown in Figure 1(a), when a primary electron beam emitted from the electron gun and hits the resist and substrate, the electrons may scattered and the scattered electrons produce backscattered electrons. These back-scattered electrons may hit the bottom of the objective lens and then produce rescattered electrons [6]. Thus, the electrons scattered within the resist and from the substrate cause inaccurate exposure of the resist in regions adjacent to those addressed by the electron beam. This effect is called “proximity effect”. More precisely, the scattered electrons (named forward-scattered electrons) incur forward-proximity effect, where electrons undergo small-angle deflection while going into the resist and the substrate. In contrast, the back-

scattered electrons (called backward-scattered electrons) generate



**Figure 1.** (a) Illustrations of the scattered, back-scattered, and rescattered electrons. These electrons result in undesired exposure which causes the proximity effect [6]. (b) The accumulated energy distribution can be modeled by the point spread function [6]. The proximity effect causes an unacceptable exposure of a narrow range, but high intensity.

backward-proximity effect, and they are usually deflected in a large angle.

Most of the published works have been proposed to handle the proximity effect during the post-layout stage. Ren [9] et al. simulated the electron scattering processes by Monte Carlo method. Hudek [7] et al. developed a software tool (called “PROX-In”) to determine the optimized control point spread function for proximity effect corrections in EBL. However, the post-layout dosage compensation and/or geometric modification inevitably incur significant computation time. Hence, it

<sup>a</sup> Corresponding author: [jlchen@fzu.edu.cn](mailto:jlchen@fzu.edu.cn)

N160320054@fzu.edu.cn; N155420004@fzu.edu.cn; chenxf2000126@126.com; wxzhu@fzu.edu.cn;

is desirable to design an effective and efficient algorithm for considering proximity effect as early as possible.

In this paper, we consider the EBL proximity effect during layout stage. During the detailed placement stage, we propose an algorithm to optimize the wirelength, displacement, and proximity variation simultaneously. The main contributions of our work are summarized as follows:

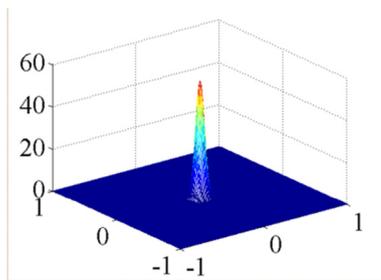
- We give an accurate evaluation scheme to estimate the proximity effect by fast Gauss transform, and devote a proximity effect aware detailed placement objective function to simultaneously consider wirelength, displacement and proximity effect.
- We present proximity aware cell swapping and cell matching technologies, which can make a good tradeoff between minimizing proximity variation and wirelength.
- Experimental results show that our algorithm is effective for the addressed problem. Compared with a state-of-the-art work, our algorithm can achieve 6.6% lower proximity variation.

The remainder of this paper is organized as follows. The next section describes preliminaries and the problem statement. Section III gives our proposed algorithm in detail. Experimental results are proposed in Section IV. Finally, Section V concludes this work.

## 2 Preliminaries

In this section, we first give the energy distribution model of proximity effect briefly. Then, we introduce the detailed placement problem considering the proximity effects in Section II-B.

### 2.1 Energy Distributions of the proximity effect



**Figure 2.** Deposited energy model of the proximity effect, which is a double Gaussian distribution centered on the primary electron beam.

In EBL, the proximity effect can be modeled as a double Gaussian function [2]. And the mathematical model of proximity effect  $f_{pro}(d)$  is:

$$f_{pro}(d) = \frac{1}{\pi(1+\eta)} \left( \frac{1}{\beta_f^2} e^{-\frac{d^2}{\beta_f^2}} + \frac{\eta}{\beta_b^2} e^{-\frac{d^2}{\beta_b^2}} \right) \quad (1)$$

where  $d = \sqrt{x^2 + y^2}$  is the distance from the incident point,  $\beta_f, \beta_b$  are the ranges of forward-proximity effect and backward-proximity effect, respectively;  $\eta$  is the ratio of the backward-scattered energy to the forward

one. Besides,  $\eta$  is 0.6 which is a suitable value for a 100KV e-beam lithography tool, a silicon substrate, and thin resist film. For correctable minimum sizes,  $\beta_f$  is from  $0.02\mu\text{m}$  to  $0.06\mu\text{m}$ , and  $\beta_b$  is  $30\mu\text{m}$  [2]. The proximity effect model are shown in Figure 2. It can be seen from Figure 2 that the proximity effect is a local effect.

### 2.2 Problem statement

In this paper, we consider the problem of minimizing the variation of proximity effect during detailed placement, therefore proximity effect can be corrected and unacceptable exposures induced by the effect can be decreased with dosage adjustment. Our proximity effect aware detailed placement problem is formulated as follows: The text of your paper should be formatted as follows:

- Given a netlist, a standard cell library and a layout after legalization, the objective of proximity effect aware detailed placement is to determine the position of each standard cell such that the total wirelength, displacement and the proximity effect are minimized and the layout is still legal.

## 3 Our algorithm

In this section, we first formulate the proximity effect aware detailed placement as an optimization problem. Then, we describe in detail the proximity aware cell swapping and cell matching techniques. Finally, the framework of our algorithm for proximity effect aware detailed placement is presented.

### 3.1 Proximity Variation Model

The model of the proximity effect is a Gaussian distribution, and this Gaussian distribution is convolved with the density map of a layout to simulate the proximity effect [4]. However, the convolution is a time-consuming process. In [6], the authors presented an efficient and accurate evaluation scheme to estimate the proximity effect by fast Gauss transform. In the evaluation scheme, each standard cell is treated as a source, and evaluation points which spread uniformly over the whole layout with their spacing between two adjacent evaluation points being a constant (e.g.,  $5\mu\text{m}$ ) are chosen to estimate the variation of fogging effect.

According to [6], the forward-proximity effect and backward-proximity effect can be calculated by

$$G_{t_i}^K(k) = \sum_{t=1}^{N_k} m_t e^{-\frac{|t_i - k_t|^2}{\delta_k}} \quad (2)$$

$$G_{t_i}^C(c) = \sum_{u=1}^{N_c} z_u e^{-\frac{|t_i - c_u|^2}{\delta_c}} \quad (3)$$

In Equations 2 and 3,  $K = \{k_1, k_2, \dots, k_{N_k}\}$  and  $C = \{c_1, c_2, \dots, c_{N_c}\}$  are the set of sources caused by the

forward-proximity effect and backward-proximity effect at the evaluation point  $t_i$ , respectively.

In this paper, we adopts this efficient and accurate evaluation scheme to estimate the proximity effect. Let evaluation points which spread uniformly over the whole layout  $T = \{t_1, t_2, \dots, t_{N_t}\}$ . The proximity effect variation can be calculated by

$$S_p(x, y) = \frac{1}{N_t} \sum_{t_i \in T^k} G_{t_i}^k(x, y)^2 - \left( \frac{\sum_{t_i \in T^k} G_{t_i}^k(x, y)}{N_t^k} \right)^2 + \frac{1}{N_t} \sum_{t_i \in T^c} G_{t_i}^c(x, y)^2 - \left( \frac{\sum_{t_i \in T^c} G_{t_i}^c(x, y)}{N_t^c} \right)^2 \quad (4)$$

where  $x$  and  $y$  are coordinates of the sources, e.g., the positions of standard cells.

### 3.2 Objective function

The goal of proximity effect aware detailed placement problem is to determine the best positions of standard cells such that the total wirelength, proximity variation are minimized, and the density constraint is satisfied. the objective function of our placement can be defined as:

$$F(x, y) = \lambda_1 W(x, y) + \lambda_2 S_p(x, y) + \lambda_3 D(x, y) \quad (5)$$

where  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$  are weights;  $W(x, y)$  is total wirelength,  $S_p(x, y)$  is proximity variation introduced in (4), and  $D(x, y)$  is total displacement of all cells, which is calculated by the Manhattan Distance between the initial positions and current positions of cells.

### 3.3 Cell swapping and cell matching

We optimize the objective function based on dividing windows. According to the objective function described in III-B, we try to reduce the proximity variation on the premise of increasing the wirelength as little as possible. We use two techniques named cell swapping and cell matching respectively to handle the problem.

1) Cell Swapping: We divide the layout region row by row such that each sub-region contains consecutive  $k$  cells in some row. In experiment  $k$  is a constant which is not greater than 6. We apply cell swapping method to each subregion. The specific optimization method is as follows.

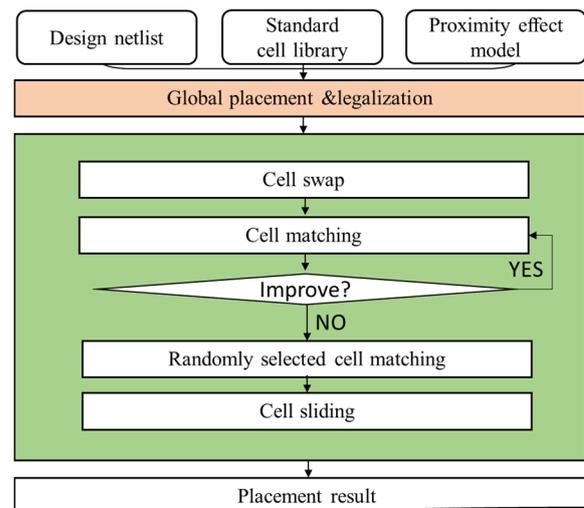
We use branch bounding method to find the optimal solution in all permutations of  $k$  cells. The cost of each solution is presented as an ordered pair  $(A, B)$ , where  $A = \lambda_1 \Delta w, B = \lambda_2 \Delta p$ . We minimize  $A$ , and if there is a tie, we minimize  $B$ . We search for the optimal solution in breadth first and the least cost-first way. Specifically, we maintain a minimum heap of currently-searched active nodes, in which nodes with smaller cost are considered first. Each node in the smallest heap is associate with a cost estimate value, which is estimated as a lower bound of the costs of the solutions that can be searched from that node. This value is calculated by our bounding function. We take out the lowest cost node

from the heap and denote its cost estimate value by  $p$ , and if it is already a solution and the minimum heap has no node or minimum cost estimate value from the heap is no less than  $p$ , we have found the optimal solution based on our search strategy. Otherwise, we remove the nodes with the cost estimate greater than  $p$  from the queue, and continue to search until we find the optimal solution.

2) Cell Matching: We extend the window-based detailed placement [8] algorithm to handle proximity effect. The algorithm selects a set of exchangeable cells and a set of free slots in a given window and constructs bipartite matching problem by assigning cells to empty slots in order to keep the layout result valid. A pair with a cell and a free slot is selected only if the width of the cell is not greater than the width of the empty slot.

In this paper, we implement the shortest augmenting path algorithm to solve the bipartite matching problem. Considering if two cells connected by some net are placed at the same time, it will cause errors in the computation of the objective value, so we select independent sets of cells to form the bipartite graph. Here, by independent cells, we mean for any pair of the selected cells there is no net connects them. Due to the excessive assignments of the cells, the execution time of the algorithm is too long. We sort all the assignments in increase order according to the cost of them and we greedily choose a certain amount of larger cost assignments such that the problem can be solved quickly. In addition to the local windows, we also take a certain number of cells randomly from the whole placement region and run the cell matching to optimize globally. Experiments show that our algorithm greatly optimizes the proximity effect.

### 3.4 Framework of our algorithm



**Figure 3.** Framework of our algorithm considering PE

Figure 3 shows that the overall flow of our FPE-aware placement algorithm. The input is design netlist, standard cell library and our FPE models. Given a legalization result, to handle proximity effect in detailed placement, we propose two methods named cell

swapping and cell matching respectively. Firstly, we run the cell swap algorithm which is described in III-C. In the cell matching phase, the layout region is divided into some sub-regions and the problem of each sub-region is reformulated as a bipartite graph matching problem. As

mentioned in III-C, we can only select an independent set of cells at a time, which makes the match insufficient. We apply an iterative

**Table 1.** Comparisons of the placement results among NTUplace4dr [5] and ours.

Benchmark	#blocks	#movable cells	#nets	util%	NTUplace4dr[5]			ours		
					Proximity variation	HPWL	Runtime(s)	Proximity variation	HPWL	Runtime(s)
mgc_des_perf_l	113018	112644	112878	90.6	288.78	1190228158	261	286.14	1202815958	290
mgc_des_perf_a	108666	108288	110281	56.7	338.89	2552767057	229	300.9	2568276264	281
mgc_des_perf_b	113018	112644	112878	56.3	118.79	2108584923	291	111.83	2107347723	355
mgc_edit_dist_a	129993	127431	131134	54.1	237.31	4837079730	252	214.19	4847102029	293
mgc_fft_1	35291	32281	33307	83.5	675.87	456958854	60	635.41	473312604	65
mgc_fft_2	35291	32281	33307	65.0	611.21	464176010	63	599.48	464268685	80
mgc_fft_a	33641	30625	32088	50.0	364.42	652007571	89	361.19	652247245	91
mgc_fft_b	33641	30625	32088	60.0	811.03	871813697	77	809.95	948662281	108
mgc_matrix_mult_l	160127	155325	158527	80.2	349.54	2231270016	251	347.24	2230231316	376
mgc_matrix_mult_a	154457	149650	154284	60.0	890.94	3371512632	239	842.19	3384684171	317
mgc_matrix_mult_b	151244	146435	151612	60.0	739.94	3702122881	201	738.47	3709503187	245
mgc_pci_bridge32_a	29882	29517	29985	64.0	380.26	488728110	53	284.04	494196710	62
mgc_pci_bridge32_b	29281	28914	29417	27.3	216.4	703911702	54	169.34	709508521	70
mgc_superblue11_a	984445	925616	935567	65.0	917.35	39092716888	15017	900.48	39606304266	17384
N.Average					1	1	1	0.934	1.013	1.226

approach that selects independent sets of cells in the sub-region that have not been processed before, until the results are no longer improved. Finally, we directly call cell sliding technology [3] to control the cell density of the layout region.

## 4 Experimental Results

Based on NTUplace4dr [5], we implemented our algorithm for proximity effect aware detailed placement in the C++ programming language, and tested it on the benchmarks provided by the authors[6]. These benchmarks were obtained by modifying the 2015 ISPD Blockage-Aware Detailed Routing-Driven Placement Contest [1], where the fence-region constraints were removed. Table I gives the circuit statistics of these benchmarks, for which the problem size range from 29k to 1292k. The numbers of blocks, movable cells, nets and the design utilization rates for the benchmark are denoted by “#blocks”, “#movable cells”, “#nets” and “util”, respectively.

In this experiment, we compare our proximity effect aware detailed algorithm (named “Ours”) with the original NTUplace4dr [5] (named “NTUplace4dr”) on the tested benchmarks. Table I lists the experimental results. In Table I, “proximity variation” gives the variation of the proximity effect, “HPWL” shows the total wirelength and “CPU(s)” denotes the total runtime in seconds, respectively. The bottom row gives the normalized proximity variation, wirelength and runtime ratios based on NTUplace4dr, and the best results among the two placers are marked in bold.

It can be seen from Table I that our algorithm can achieve the best proximity variation for the first fourteen benchmarks. On average, our algorithm can improve 6.6% proximity variation than “NTUplace4dr” with an

average wirelength overhead of 1.3%. Particularly, our algorithm can achieve the best proximity variation and HPWL on benchmark “mgc des perf b” and “mgc matrix mult 1”. Experimental results show that our algorithm is effective and efficient for handling the proximity effect aware detailed placement problem.

## 5 Conclusions

In this paper, we considered the proximity effect aware detailed placement problem. We propose proximity aware cell swapping and cell matching to solve the minimization problem with wirelength, displacement and proximity variation. Experimental results show that our algorithm is effective and efficient for the addressed problem.

## Acknowledgments

This work was supported in part by the NSFC under Grants 11501115 and 61672005, and in part by Fujian Provincial Natural Science Foundation under Grant 2017J01506.

## References

1. I. S. Bustany, D. Chinnery, J. R. Shinnerl, and V. Yutsis. Ispd 2015 benchmarks with fence regions and routing blockages for detailed routing-driven placement. In Symposium on International Symposium on Physical Design, pages 157–164, 2015.
2. Y. W. Chang, R. G. Liu, and S. Y. Fang. Euv and e-beam manufacturability: challenges and solutions. In Design Automation Conference, page 198, 2015.

3. T. C. Chen, Z. W. Jiang, T. C. Hsu, H. C. Chen, and Y. W. Chang. Ntuplace3: An analytical placer for large-scale mixed-size designs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27(7):1228–1240, 2008.
4. L. Greengard and J. Strain. The fast gauss transform. *Siam Journal on Scientific & Statistical Computing*, 12(1):79–94, 1991.
5. C. C. Huang, H. Y. Lee, B. Q. Lin, S. W. Yang, C. H. Chang, S. T. Chen, and Y. W. Chang. Detailed-routability-driven analytical placement for mixed-size designs with technology and region constraints. In *Ieee/acm International Conference on Computer-Aided Design*, pages 508–513, 2015.
6. Y. C. Huang and Y. W. Chang. Fogging effect aware placement in electron beam lithography. In *Design Automation Conference*, page 53, 2017.
7. P. Hudek and D. Beyer. *Exposure optimization in high-resolution e-beam lithography*. Elsevier Science Ltd., 2006.
8. Z. W. Jiang, T. C. Cheny, T. C. Hsuy, H. C. Chenz, and Y. W. Changyz. Ntuplace2: a hybrid placer using partitioning and analytical techniques. In *International Symposium on Physical Design*, pages 215–217, 2006.
9. L. Ren and B. Chen. Proximity effect in electron beam lithography. In *International Conference on Solid-State and Integrated Circuits Technology*, 2004. *Proceedings*, pages 579–582 vol.1, 2004.