

Mixed-Cell-Height Placement with Complex Minimum-Implant-Area Constraints*

Jianli Chen¹, Peng Yang¹, Xingquan Li¹, Wenxing Zhu¹, and Yao-Wen Chang^{2,3}

¹Center for Discrete Mathematics and Theoretical Computer Science, Fuzhou University, Fuzhou 350108, China

²Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 10617, Taiwan

³Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan

{jlchen, N165420006, N130320024, wxzhu}@fzu.edu.cn; ywchang@ntu.edu.tw

ABSTRACT

Mixed-cell-height standard cells are prevalingly used in advanced technologies to achieve better design trade-offs among timing, power, and routability. As feature size decreases, placement of cells with multiple threshold voltages may violate the complex minimum-implant-area (MIA) layer rule arising from the limitations of patterning technologies. Existing works consider the mixed-cell-height placement problem only during legalization, or handle the MIA constraints during detailed placement. In this paper, we address the mixed-cell-height placement problem with MIA constraints into two major stages: post global placement and MIA-aware legalization. In the post global placement stage, we first present a continuous and differentiable cost function to address the Vdd/Vss alignment constraints, and add weighted pseudo nets to MIA violation cells dynamically. Then, we propose a proximal optimization method based on the given global placement result to simultaneously consider Vdd/Vss alignment constraints, MIA constraints, cell distribution, cell displacement, and total wirelength. In the MIA-aware legalization stage, we develop a graph-based method to cluster cells of specific threshold voltages, and apply a strip-packing-based binary linear programming to reshape cells. Then, we propose a matching-based technique to resolve intra-row MIA violations and reduce filler insertion. Furthermore, we formulate inter-row MIA-aware legalization as a quadratic programming problem, which is efficiently solved by a modulus-based matrix splitting iteration method. Finally, MIA-aware cell allocation and refinement are performed to further improve the result. Experimental results show that, without any extra area overhead, our algorithm still can achieve 8.5% shorter final total wirelength than the state-of-the-art work.

1 INTRODUCTION

In traditional circuit designs, standard cells have the same height for easier design and optimization [17]. However, standard cells of different row heights are used in modern circuit designs to achieve

better design trade-offs among timing, power, and routability [12]. Specifically, higher cells give larger drive strengths and better pin accessibility and routability at the costs of larger area and power [7]. Such mixed-cell-height circuit designs are more challenging due to the heterogenous cell structures (and thus more global cell interferences and larger solution spaces). In such designs, further, power (Vdd) or ground (Vss) alignment is an additional placement constraint on multi-row-height cells [4]. The Vdd/Vss lines are interleaved through cell rows, and each cell must be aligned correctly such that its Vdd/Vss pins match the corresponding rows. So an even-row-height cell must be aligned to a placement row with the same type of power lines at the top (or bottom) boundary of the cell, while an odd-row-height cell can be aligned to any placement row directly or by flipping the cell vertically to match the Vdd/Vss pins.

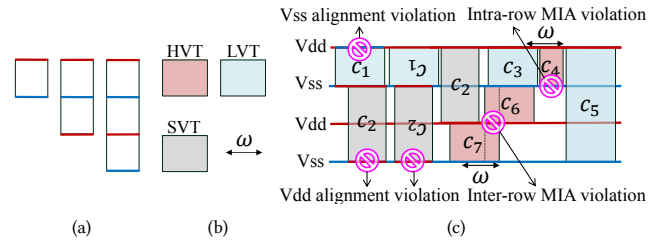


Figure 1: Mixed-cell-height structure with the Vdd/Vss and MIA constraints. (a) Different row-height standard cells. (b) Three types of VTs and its required minimum implant width ω . (c) A mixed-cell-height circuit design with the MIA constraints.

Simultaneous timing and power optimization is often a tough task in modern circuit designs [16]. A popular method to balance the two tasks is to apply multiple threshold voltages (multi-VTs) to reduce leakage power while maintaining circuit performance. There are three types of VTs: high threshold voltage (HVT), low threshold voltage (LVT), and standard threshold voltage (SVT). In a multi-VT design, LVT cells are used on critical paths to improve timing, while HVT cells on non-critical paths to reduce leakage power. As feature size decreases, however, placement with multi-VTs may violate the minimum-implant-area (MIA) layer rule arising from the limitations of patterning technologies [18]. The MIA constraints include two complex constraints: intra-row MIA violations and inter-row MIA violations [9]. The impact of the MIA constraints is critical because small-area cells are often employed for cost-driven and low-power designs [18].

Fig. 1 illustrates a mixed-cell-height structure with the Vdd/Vss alignment constraints and MIA constraints. For the Vdd/Vss alignment constraints, the odd-row-height cells c_1, c_3, c_4, c_5, c_6 and c_7

*This work was supported by Empyrean and the National Natural Science Foundation of China under Grants 11501115, 61672005 and 11331003, and by AnaGlobe, TSMC, MOST of Taiwan under Grant No.'s MOST 105-2221-E-002-190-MY3, MOST 106-2911-I-002-511, and MOST 107-2221-E-002-161-MY3.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICCAD '18, November 5–8, 2018, San Diego, CA, USA

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5950-4/18/11...\$15.00

<https://doi.org/10.1145/3240765.3240828>

can be placed to any row to match the correct Vdd/Vss power rails directly or by flipping the cells vertically, while the even-row-height cell c_2 must match a power rail of the same type because its bottom boundary is designed for a Vdd line. The MIA constraints include intra-row and inter-row MIA violations. For intra-row violations, since cell c_4 is sandwiched by different VT cells c_3 and c_5 on the same row and its width is less than the minimum implant width ω , cell c_4 causes an intra-row MIA violation. For inter-row violations, cells c_6 and c_7 incur an inter-row MIA violation because their overlap width (area) in the horizontal direction is less than ω .

Since multiple VTs and the mixed-cell-height structure are both important to modern circuit designs for simultaneous area, timing, and power optimization, it is desirable to consider the mixed-cell-height placement with the complex MIA constraints. Such mixed-cell-height circuit designs with multi-VTs are challenging mainly because of their heterogeneous cell structures and additional constraints.

1.1 Previous Work

Recently, several works addressed the mixed-cell-height circuit designs during legalization and detailed placement. The works [4, 6, 17] considered the general mixed-cell-height legalization problem with the Vdd/Vss alignment constraints, and found a legal placement with the minimum total cell displacement and wirelength. In [12], the authors developed a detailed placement method for mixed-cell-height circuit designs which considers wirelength, cell density, and pin density. However, all these previous works do not consider the crucial MIA constraints. As a result, MIA constraints only can be fixed by filler insertion, which may incur significant area and wirelength overheads.

On the other hand, the works [9, 11, 13] considered the single-row-height standard cell placement with MIA constraints. Kahng and Lee in [11] applied filler insertion and threshold voltage refinement during the post-placement stage to handle the MIA constraints, which incurs area overheads and cannot guarantee to fix all violations of timing constraints. The work [9] proposed a mixed integer linear program (MILP) to concurrently address the MIA constraints, minimum oxide diffusion jog length, and drain-drain abutment constraints during detailed placement. An integer linear programming (ILP) was used in [13] to handle detailed placement and threshold voltage refinement. However, in mixed-cell-height circuit designs, shifting a cell in one row may cause cell overlaps in another row. Due to the larger solution space and additional power-rail constraints, it is not easy to extend these works on single-row-height standard cell placement with MIA constraints to handle mixed-cell-height designs effectively.

More recently, Wu and Chang [18] first considered the MIA-aware mixed-cell-height detailed placement problem. For a given mixed-cell-height legal placement result and its MIA constraints, they applied clustering and reshaping techniques to fix intra-row violating cells, pushed violating cells to legal positions to eliminate inter-row MIA violations, and used a refinement operation to compact design area. Experimental results shown that their method achieved high solution quality. Detailed placement is regarded as a refinement stage in placement, in which cells are typically perturbed within a local region. Due to the heterogeneous cell structures, however, cell overlapping in mixed-cell-height placement is dependent among rows. The local operations may not be effective enough

to handle global cell interferences, which may incur significant area and wirelength overheads. In order to obtain a better mixed-cell-height placement result, it is desirable to consider the MIA constraints as early as possible. This consideration is beneficial not only for handling MIA constraints, but also for reducing the area and wirelength overheads.

1.2 Our Contributions

In this paper, we consider the mixed-cell-height placement problem with the Vdd/Vss constraints and MIA violations. In order to preserve the quality of a given global placement result and eliminate its MIA violations, our algorithm contains two major stages: post global placement and MIA-aware legalization. The main contributions of our work are summarized as follows:

- We present a continuous and differentiable cost function to address the Vdd/Vss alignment constraints, and dynamically add weighted pseudo nets on these cells of the same VT types, which reduces the difficulty of fixing the Vdd/Vss constraints and MIA constraints in the subsequent steps.
- We propose a proximal optimization method based on the given global placement result to simultaneously consider Vdd/Vss alignment constraints, MIA constraints, cell distribution, cell displacement, and total wirelength. Thus all cells are located to match or to be near the rows legally, which can be legalized more easily and effectively at the legalization step.
- We develop a graph-based algorithm to cluster cells of specific threshold voltages to resolve the intra-row MIA violations, and a strip-packing-based binary linear programming (BLP) to compact and reshape the total area of cells in a cluster.
- An inter-row MIA-aware quadratic programming (QP) problem is formulated to resolve inter-row MIA violations and cell overlaps. Then the QP is solved by a modulus-based matrix splitting iteration method (MMSIM) efficiently. To guarantee the convergence of the MMSIM, we construct a constraint matrix of full row rank by cell splitting and virtual cell insertion.
- Experimental results show that our algorithm can achieve 8.5% shorter total wirelength than a state-of-the-art work without extra area overhead. In particular, our MIA-aware legalization stage reduce 43.3% total cell displacement than the state-of-the-art work.

The remainder of this paper is organized as follows. Section 2 gives the problem statement and our algorithm framework. Section 3 presents our post global placement. Section 4 details our MIA-aware legalization. Section 5 gives the experimental results. Finally, conclusions are made in Section 6.

2 PRELIMINARIES

In this section, we first give the problem statement and then present our algorithm framework.

2.1 Problem Statement

Given a mixed-cell-height global placement result with a set $C = \{c_1, c_2, \dots, c_n\}$ of n standard cells and a set $E = \{e_1, e_2, \dots, e_m\}$ of m nets, the placement region is a rectangular sheet with $(0, 0)$ and (W, H) as the bottom-left and top-right corner coordinates,

respectively. Let (x_i, y_i) be the center coordinate of cell c_i , and (w_i, h_i) be the width and height of this cell. Each cell has a boundary type for Vdd/Vss. The site width $Site_w$ and site height $Site_h$ are two given constants, where $Site_h$ equals the row height h . For a multi-row-height cell, the height of this cell is a multiple of $Site_h$. Let C^H and C^L be the sets of HVT and LVT cells, respectively. The minimum implant width (or area) ω is a given constant.

In this paper, the mixed-cell-height circuit placement problem with the MIA constraints is to determine the position of each cell such that the total wirelength is minimized and the constraints are all satisfied. Based on the problem formulations in [3, 6, 18], we formally model the mixed-cell-height placement problem with the complex MIA constraints as follows:

$$\begin{aligned}
\min W(x, y) &= \sum_{e \in E} \left(\max_{c_i, c_j \in e} |x_i - x_j| + \max_{c_i, c_j \in e} |y_i - y_j| \right) \\
\text{s.t.} \\
1) & O_{ij}(x_i, y_i, x_j, y_j) = 0, \text{ for all } c_i, c_j \in C, i \neq j; \\
2) & 0 \leq x_i - \frac{w_i}{2}, x_i + \frac{w_i}{2} \leq W, \text{ for all } c_i \in C, \\
& 0 \leq y_i - \frac{h_i}{2}, y_i + \frac{h_i}{2} \leq H, \text{ for all } c_i \in C; \\
3) & \forall c_i \in C, \exists \alpha_i \in \{0, 1, 2, \dots\}, x_i - \frac{w_i}{2} = \alpha_i Site_w; \\
4) & \forall c_i \in C \text{ s.t. } h_i = 2l \times Site_h \text{ for some } l = 1, 2, 3, \dots \\
& y_i - \frac{h_i}{2} \in \begin{cases} \{0, 2, 4, \dots\} \times Site_h, & \text{if the first row} \\ & r_0 \text{ matches Vdd/Vss of } c_i, \\ \{1, 3, 5, \dots\} \times Site_h, & \text{otherwise;} \end{cases} \\
5) & O_{ij}^H = 0 \text{ or } O_{ij}^H \geq \omega, \text{ for all } c_i, c_j \in C^H \text{ (or } C^L), \\
& \text{and } c_i, c_j \text{ in the adjacent rows} \\
& w_{i_1} + w_{i_2} + \dots + w_{i_{u_l}} \geq \omega, c_{i_1}, c_{i_2}, \dots, c_{i_{u_l}} \in C^H \\
& \text{(or } C^L) \text{ are on the same row and adjacent to each other,}
\end{aligned} \tag{1}$$

where $O_{ij}(x_i, y_i, x_j, y_j)$ is the overlap function between cells c_i and c_j , O_{ij}^H is the overlap length between cells c_i and c_j in the horizontal direction. In Problem (1), Constraint (1) requires cells to be non-overlapping; Constraint (2) requires cells to be placed inside the placement region; Constraint (3) requires cells to be located at placement sites on rows; Constraint (4) requires cells to be aligned to correct Vdd/Vss; Constraint (5) requires cells to be MIA inter-row and intra-row violation-free.

2.2 Algorithm Framework

Our algorithm for the mixed-cell-height placement with the complex MIA constraints is summarized in Fig. 2. It consists of two major parts: post global placement and MIA-aware legalization. In post global placement stage, in order to draw HVT/LVT cells together, we add pseudo-nets among intra-row violation cells in a local window. With the addition pseudo nets, we present a continuous and differentiable cost function to address the Vdd/Vss alignment constraints, and optimize it by the conjugate gradient method. In clustering and reshaping stage, we cluster violation cells based on a graph algorithm, and minimize the bounding box area of a cluster by a strip packing based ILP. In MIA-aware legalization stage, an inter-row violation aware QP is formulated, which is solved by MMSIM solver. Finally, to obtain a better result, an inter-row MIA-aware cell allocation and cell refinement is performed. After these operations, we generate a MIA-violation-free placement with minimized wirelength and area. The details of each part are elaborated in the following sections.

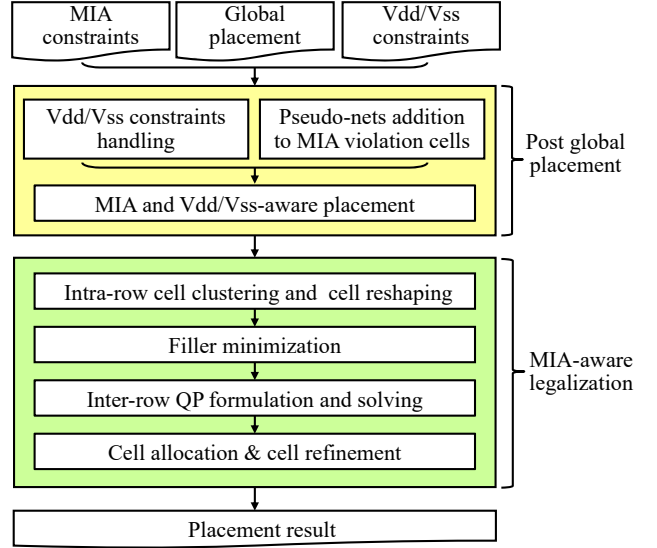


Figure 2: Framework of our algorithm.

3 POST GLOBAL PLACEMENT (POST-GP)

In this section, we introduce our post global placement, which consists of three stages: (1) Vdd/Vss constraints handling, (2) pseudo-nets addition to MIA-violation cells, and (3) MIA and Vdd/Vss-aware placement.

3.1 Vdd/Vss Constraints Handling

In Problem (1), if each cell is aligned to its correct rows (which meets the Vdd/Vss alignment constraint) as much as possible during post-GP, then it can be legalized more easily and effectively. As a result, the total wirelength and runtime can be reduced during legalization. Since the Vdd/Vss alignment constraint (4) in Problem (1) is in discrete form, it is hard to optimize directly using a continuous optimization method. Thus we present a continuous and differentiable cost function $Cost(c_i)$ to address the Vdd/Vss alignment constraints at the post-GP stage:

$$Cost(c_i) = \frac{1 + s_i}{2} \left(\sin \left(\frac{y_i - \frac{h_i}{2}}{(1 + s_i)h} \pi + \frac{s_i r_i}{2} \pi \right) \right)^2, \tag{2}$$

where

$$s_i = \begin{cases} 0, & \text{if cell } c_i \text{ is an odd-row-height cell;} \\ 1, & \text{if cell } c_i \text{ is an even-row-height cell,} \end{cases} \tag{3}$$

$$r_i = \begin{cases} 0, & \text{if the bottom boundaries of cell } v_i \text{ and} \\ & \text{the chip have the same type of Vdd/Vss;} \\ 1, & \text{otherwise.} \end{cases} \tag{4}$$

According to the definition of $Cost$, for an odd-row-height cell, if it is far away from a row (whether or not to meet the Vdd/Vss alignment requirement), then the cost value of the cell will be larger. In contrast, if it is close to a row, then the value of $Cost$ decreases. For an even-row-height cell, it must match a Vdd/Vss of the same type. If it is far away from a row that meets the Vdd/Vss requirement, the cost value of the cell will be larger.

For example, in Fig. 3, assume that the center y -coordinates of the cells are $y_1 = 3.3h$, $y_2 = 2.5h$, $y_3 = 2.6h$, $y_4 = 2h$, $y_5 = 1.8h$, and

$y_6 = 1.5h$, and the type of Vdd/Vss for the chip bottom boundary is Vdd. For cell c_1 , $y_1 - \frac{h_1}{2} = 2.8h$, $s_1 = 0$, $s_1 r_1 = 0$, and thus $Cost(c_1) = 0.173$. Similarly, we have $Cost(c_2) = 0$, $Cost(c_3) = 0.655$, $Cost(c_4) = 1$, $Cost(c_5) = 0.327$, and $Cost(c_6) = 0$.

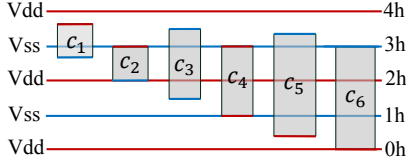


Figure 3: Example of the penalty term for the Vdd/Vss alignment constraints.

3.2 Pseudo-Nets Addition to MIA-Violation Cells

To reduce MIA violations, each HVT/LVT cell violating the MIA constraints is supposed to adjoin with cells of the same type. We add pseudo-nets among HVT/LVT cells and assign appropriate weights to these nets. Based on the weighted pseudo-nets, the HVT/LVT cells can be placed as close as possible to reduce MIA violations.

In addition, to avoid excessive cell displacement, we only add pseudo-nets among cells within a local region. Specifically, if cells c_i and c_j are of the same VT type, the Manhattan distance between them is not more than R_c , and at least one of their widths is less than the minimum implant width ω , then a pseudo-net e_{ij}^p is added between c_i and c_j . To differentiate pseudo-nets, we then assign a weight to pseudo-net e_{ij}^p dynamically. An intuitive weight assignment is that the two cells are more likely to adjoin together if the distance between c_i and c_j is smaller; further, if the two cells have the same height, their weight would be larger to encourage them to adjoin together because no white space will be introduced.

Hence, the dynamic pseudo-net weight w_{ij} between c_i and c_j in the R_c region is set as

$$w_{ij} = \frac{h_i + h_j}{2\max\{h_i, h_j\}} \cdot e^{\left\{ \frac{\frac{1}{n_p} \sum e_{ij}^p (|x_i - x_j| + |y_i - y_j|)}{|x_i - x_j| + |y_i - y_j| + \kappa} - 1 \right\}}, \quad (5)$$

where n_p is the number of pseudo-nets. The exponent term is used to scale the value of w_{ij} , where $|x_i - x_j| + |y_i - y_j|$ is the Manhattan distance between cells c_i and c_j . $\frac{1}{n_p} \sum e_{ij}^p (|x_i - x_j| + |y_i - y_j|)$ is the average Manhattan distance of all pseudo-nets e_{ij}^p , and κ ($\kappa > 0$) is a constant for keeping the exponent term finite.

3.3 MIA and Vdd/Vss-Aware Placement

In global placement, since both of the wirelength and density functions are non-smooth, the log-sum-exp wirelength model $\hat{W}(x, y)$ and the bell-shaped function $\hat{D}_b(x, y)$ are applied to approximate the total HPWL and smooth the density function [5], respectively. Then, the traditional global placement problem can be formulated as a constrained minimization problem as:

$$\begin{aligned} \min \quad & \hat{W}(x, y) \\ \text{s.t.} \quad & \hat{D}_b(x, y) = M_b, \text{ for each bin } b, \end{aligned} \quad (6)$$

where M_b is the maximum allowable area of movable cells in bin b [5].

After adding pseudo-nets among HVT/LVT cells in Section 3.2, the wirelength function $\hat{W}(x, y)$ is converted to a new wirelength

Algorithm 1 Clustering

Input: a post global placement result and a clustering spacing cs .

Output: clustering result.

```

1:  $V' = \emptyset$ ;
2: construct clustering graph  $G_C(V, E, \bar{W})$  according to  $cs$ , the weight  $\bar{w}_{uv}$  of each edge  $e_{uv}$  is calculated by Equation (9);
3: do
4:   find an edge  $e_{uv} \in E$  with the minimum value of  $\bar{w}_{uv}$ ;
5:   if at least one of HVT/LVT cells  $u$  and  $v$  is intra-row violating
   then
6:     cluster  $u$  and  $v$  to  $u_\ell$ ;
7:     find a minimum wirelength region for  $u_\ell$ ;
8:     add the edges connecting  $u$  or  $v$  to  $u_\ell$  (except  $e_{uv}$ );
9:     update the weight of edges connecting  $u_\ell$  by Equation (9);
10:     $V = V \cup \{u_\ell\}$ ;
11:   end if
12:    $V = V - \{u\} - \{v\}$ ,  $V' = V' \cup \{u\} \cup \{v\}$ ;
13:   delete the edges connecting  $u$  or  $v$ ;
14: until all intra-row MIA-violating cells (except isolated vertices) are visited
15: Return  $V, V'$ 

```

function $\bar{W}(x, y)$. We also introduce the constraints $\sum_{c_i \in C} Cost(c_i)$ to consider the Vdd/Vss constraints. Hence, our post global placement model considering Vdd/Vss constraints, MIA violations, and cell displacement can be reformulated as:

$$\begin{aligned} \min \quad & \bar{W}(x, y) \\ \text{s.t.} \quad & \hat{D}_b(x, y) = M_b, \text{ for each bin } b; \\ & \sum Cost(c_i) = 0, \text{ for all } c_i \in C. \end{aligned} \quad (7)$$

In Problem (7), the objective function and the constraints are continuous and differentiable. Using the penalty function method, we introduce factors λ_1 and λ_2 to the area constraint $\hat{D}_b(x, y) = M_b$ and the equality constraint $\sum cost(c_i) = 0$, respectively. Then, Problem (7) can be further reformulated as an unconstrained optimization problem as follows:

$$\min f(u) = \bar{W}(x, y) + \lambda_1 \sum_b (\hat{D}_b - M_b)^2 + \lambda_2 \sum_{c_i \in C} Cost(c_i), \quad (8)$$

where λ_1 and λ_2 are normalized factors based on the total wirelength. We use the conjugate gradient optimization method without exact line search to solve Problem (8).

4 MIA-AWARE LEGALIZATION

After the post global placement stage, these HVT/LVT cells are placed as close as possible, and all cells satisfy the Vdd/Vss constraints as much as possible. The purpose of our MIA-aware legalization is to eliminate MIA violations and place cells on rows legally. Our legalization algorithm contains four stages: (1) intra-row cell clustering and cell reshaping, (2) filler minimization, (3) inter-row QP formulation and solving, and (4) cell allocation and cell refinement.

4.1 Intra-Row Cell Clustering and Cell Reshaping

In order to fix intra-row violations, we propose a graph-based clustering algorithm to cluster intra-row violating cells. Then, we apply a strip-packing-based cells reshaping method to reshape these clusters, which can compact these clusters and reduce area overheads.

4.1.1 Graph-Based Clustering. In order to eliminate all the intra-row MIA violations, we propose a graph-based clustering method to cluster HVT/LVT cells together and find their desired positions in this stage.

First, given a clustering spacing cs , we construct a clustering graph $G_C(V, E, \bar{W})$ as follows.

Definition 4.1 (Clustering Graph). A clustering graph is an undirected graph $G_C(V, E, \bar{W})$, where vertex $v \in V$ denotes a HVT/LVT cell. If the distance between cells u and v with same VT types (HVT or LVT) are less than cs , then there exists an edge $e_{uv} \in E$ between u and v , and there is a weight $\bar{w}_{uv} \in \bar{W}$ of e_{uv} .

According to the clustering graph G_C , we propose Algorithm 1 as our clustering algorithm. In Line 2 of Algorithm 1, the weight $\bar{w}_{uv} \in \bar{W}$ of G_C is calculated by

$$\bar{w}_{uv} = \frac{2(|x_u - x_v| + |y_u - y_v|) \times \max\{h_u, h_v\}}{h_u + h_v}. \quad (9)$$

In Line 7, the optimal position (x_{u_ℓ}, y_{u_ℓ}) for cluster $u_\ell = \{c_1, c_2, \dots, c_{n_{u_\ell}}\}$ can be obtained by solving the following optimization problem:

$$(x_{u_\ell}, y_{u_\ell}) = \operatorname{argmin} \sum_{i=1}^{n_{u_\ell}} \sum_{j=1}^{m_i} |x_{u_\ell} - x_{i,j}| + \sum_{i=1}^{n_{u_\ell}} \sum_{j=1}^{m_i} |y_{u_\ell} - y_{i,j}|, \quad (10)$$

where n_{u_ℓ} is the number of cells in the cluster u_ℓ , m_i is the number of cells which connects cell c_i , $(x_{i,j}, y_{i,j})$ is the coordinate of the cell $c_{i,j}$. Since it is time-consuming to directly solve Problem (10), we apply the median formulation in [8] to obtain a minimum wirelength region for cluster u_ℓ .

In Algorithm 1, the final results of V and V' are the obtained clusters. For cluster V' , if the width of the cluster is larger than 2ω , we will check whether it can be divided into two or more intra-row violation-free clusters, and if so the positions of new clusters will be updated. In a cluster u_ℓ , all even-row-height cells of the same height satisfy the Vdd/Vss alignment constraints.

4.1.2 Strip-Packing-Based Cells Reshaping. After graph-based clustering, to minimize the chip design area, we compress each cluster to obtain a compact structure. Furthermore, in a cluster, if the highest cell is r -row-height, then it is assumed that the cluster is of r row heights. Correspondingly, the objective is transformed as minimizing the width of the cluster. This reshaping can be seen as a special strip packing problem as follows:

Problem 1 (Special Strip Packing Problem). Given a cluster of cells $u_\ell = \{c_1, c_2, \dots, c_l\}$ with Vdd/Vss, in which the respective width and height of each cell c_i are w_i and h_i , the height of each cell is the multiple of $Site_h$, and a strip with Vdd/Vss. The special strip packing problem is to pack all the cells in u_ℓ into a strip of height $r \times Site_h$ such that the non-overlapping and Vdd/Vss alignment constraints are met and the width of the strip is minimized.

We can obtain a compact cluster with the minimum width for all cells in a cluster by solving the above special strip packing problem. However, the special strip packing problem is NP-hard because its plain version without the Vdd/Vss alignment constraints is already NP-hard [15]. To address this problem, we first decide the row re-assignment, i.e., cells in u_ℓ are reassigned to the corresponding rows such that the maximum width of the strip is minimized. In this paper, we formulate the row re-assignment as a binary linear

programming (BLP) problem, which is solved by the branch-and-bound method.

Let $z_{i,j}$ be a binary variable, for which $z_{i,j} = 1$ denotes that the bottom-left coordinate of cell $c_i \in u_\ell$ is assigned to row j . And $r_i = h_i / Site_h$ denotes that the height of cell c_i is r_i times of $Site_h$. W_s is the width of the strip. Then the BLP for the row re-assignment is formulated as:

$$\begin{aligned} & \min W_s \\ & \text{s.t.} \\ & 1) \sum_{i=1}^l \sum_{k_i=1}^{r_i} z_{i,j-k_i+1} \cdot w_i \leq W_s, \quad j = 1, 2, \dots, r; \\ & 2) \sum_{j=1}^{r-r_i+1} z_{i,j} = 1, \quad i = 1, 2, \dots, l; \\ & 3) i = 1, 2, \dots, l, \quad \text{if } c_i \text{ is of even row height,} \\ & \quad \begin{cases} \sum_{j=1}^{r/2} z_{i,2j} = 0, & \text{if } c_i \text{ is aligned with } 2j-1 \text{ row,} \\ \sum_{j=1}^{r/2} z_{i,2j-1} = 0, & \text{if } c_i \text{ is aligned with } 2j \text{ row,} \end{cases} \\ & 4) z_{i,j} \in \{0, 1\}, \quad i = 1, 2, \dots, l, \text{ and } j = 1, 2, \dots, r. \end{aligned} \quad (11)$$

In BLP (11), Constraint (1) ensures that the total cell width of every row is not larger than W_s . Constraint (2) ensures that every r_i -row-height cell is assigned to one of the rows between row 1 and row $r-r_i+1$. Constraint (3) guarantees that an even-row-height cell can be aligned to a correct Vdd/Vss. Since the size of each cluster is small, the branch-and-bound method is used to solve the BLP Problem (11). After re-assigning cells into corresponding rows, it is easy to obtain a reshaping result.

4.2 Filler Minimization

After intra-row MIA-aware cells clustering and reshaping, most of the intra-row MIA violations are eliminated. For the remaining intra-row MIA-violating cells, fillers should be inserted such that the sum of the widths of HVT/LVT cells and the widths of abutting fillers is larger than ω .

For these intra-row violating cells, we first search all the cells of the same VT type and fillers around them. Then we propose a matching-based method to simultaneously cluster intra-row MIA-violating cells and minimize their filler insertion. To honor the initial solution quality and avoid excessive wirelength increase, we fix violating cells in a local region. Given a window, a bipartite graph is constructed according to the following rules:

- If both of the width and height of an intra-row MIA-violating cell c_i are smaller than a filler f_j , then an edge is built between cell c_i and filler f_j ;
- If an intra-row MIA-violating cell c_i and another MIA violation-free cell/cluster c_j have the same VT type, height, and Vdd/Vss, then an edge is built between cell c_i and cell/cluster c_j .

In the bipartite graph, the weight of each edge \hat{w}_{ij} is calculated by:

$$\hat{w}_{ij} = \frac{\theta_{ij}(h_i + h_j)}{2\max\{h_i, h_j\} \times (|x_i - x_j| + |y_i - y_j|)}, \quad (12)$$

where if either i or j is a filler, $\theta_{ij} = 2$; otherwise, $\theta_{ij} = 1$. Based on the weighted bipartite graph, the Kuhn-Munkres algorithm [14] is applied to obtain a maximum weight matching.

Fig. 4 shows an example of our matching-based method. After intra-row MIA-aware cells clustering and reshaping, in Fig. 4(a), cells c_1 , c_2 , and c_3 are isolated intra-row MIA-violating HVT cells,

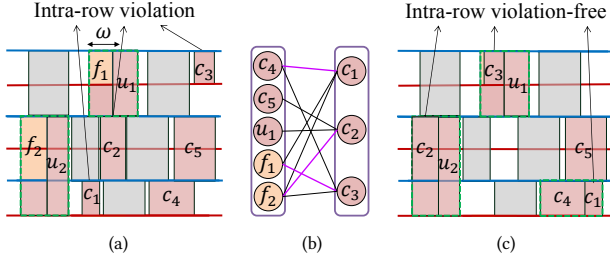


Figure 4: An example of our matching-based method. (a) Given a circuit with three intra-row violating cells. (b) Construction of the weighted bipartite graph. (c) A maximum weight matching results after Kuhn-Munkres algorithm.

cells c_4 and c_5 are intra-row MIA violation-free HVT cells, f_1 and f_2 are fillers. According to Fig. 4(a), a corresponding weighted bipartite graph is constructed as shown in Fig. 4(b). Then, the Kuhn-Munkres algorithm is applied to obtain a maximum weight matching $\{(c_1, c_4), (c_2, f_2), (c_3, f_1)\}$. Finally, as shown in Fig. 4(c), c_1 is moved to the position behind c_4 , and c_2 and c_3 are moved to the position of f_2 and f_1 , respectively. As a result, the intra-row MIA violations are eliminated, and the fillers are minimized.

4.3 Inter-Row QP Formulation and Solving

After the above processing, all intra-row violations are eliminated. In order to fix inter-row violation, we first model our legalization problem as a quadratic programming problem, and then we use a modified modulus-based matrix splitting iteration method (MMSIM) to solve the problem.

4.3.1 Inter-Row QP Formulation. In this subsection, we consider the inter-row violations during legalization. We first detect all possible inter-row violations in each two adjacent rows. For two cells c_i and c_j in adjacent rows, an inter-row violation is generated when $0 < O_{ij}^H < \omega$, where O_{ij}^H is the overlap length between cells c_i and c_j in the horizontal direction. There are two possible inter-row violation structures: $0 < O_{ij}^H < \omega/2$, and $\omega/2 \leq O_{ij}^H < \omega$. For the former case, the ideal violation-free result is $O_{ij}^H = 0$. And for the latter case, the ideal violation-free result is $O_{ij}^H \geq \omega$.

According to the above descriptions, we formulate the inter-row MIA-aware mixed-cell-height legalization problem as:

$$\begin{aligned}
 & \min \frac{1}{2} \sum_{i=1}^n (x_i - x'_i)^2 \\
 & \text{s.t.} \\
 & 1) x_i - x_j \geq \frac{w_i + w_j}{2}, \quad \text{if } x'_i \geq x'_j, \\
 & \quad \text{for all adjacent cells } i \text{ and } j \text{ on the same row;} \\
 & 2) \text{ if } c_i \text{ and } c_j \text{ are inter-row violating cells and } x'_i \geq x'_j \\
 & \quad \begin{cases} x_i - x_j \geq \frac{w_i + w_j}{2}, & \text{if } 0 < O_{ij}^H < \frac{\omega}{2}, \\ x_j - x_i \geq \omega - \frac{w_i + w_j}{2}, & \text{if } \frac{\omega}{2} < O_{ij}^H < \omega, \end{cases} \\
 & 3) x_i \geq 0, \text{ for each cell } c_i,
 \end{aligned} \tag{13}$$

where x'_i and x'_j are the x -coordinate of cells c_i and c_j before legalization. It can be rewritten as a quadratic program (QP):

$$\begin{aligned}
 & \min \frac{1}{2} x^T Q x + d^T x \\
 & \text{s.t.} \quad A x \geq a, \\
 & \quad B x \geq b, \\
 & \quad x \geq 0,
 \end{aligned} \tag{14}$$

where Q is an identity matrix, d is a vector with the i -th component $d_i = -x'_i$, and A and B are the corresponding constraint matrices for Constraints (1) and (2). In matrices A and B , there are two nonzero elements -1 and 1 in each row, in which the number of rows gives the number of constraints, and the number of columns equals the number of variables.

4.3.2 QP Solving by MMSIM. In [4], Chen et al. developed an effective and efficient MMSIM for solving the mixed-cell height legalization problem. In this work, the method requires that the objective matrix of a QP is symmetric positive definite and the constraint matrix is of full row rank to guarantee an optimal solution. Compared with Problem (5) in [4], our Problem (14) has an additional Constraint (2). As a result, we use the following two operations on cells to ensure that our constraint matrix is of full row rank:

- (1) For any two inter-row violating cells c_i and c_j , a virtual cell c'_i is inserted, where $x'_i = x_i$, $y'_i = y_j$, $w'_i = w_i$, and $h'_i = h_j$;
- (2) Each multi-row-height cell c_i (including virtual cells) is split into multiple single-row-height subcells, i.e., $x_{i1} = x_{i2} = \dots = x_{ir}$, where r is the number of subcells.

Based on the operations, we construct a new constraint matrix H with considering cell overlaps and inter-row violations. For Operation (2), we introduce a new constraint $Ex = 0$ to ensure that the variables for each multi-row-height standard cell are equal. Let matrix $h = \begin{pmatrix} a \\ b \end{pmatrix}$. Problem (14) can be reformulated as follows:

$$\begin{aligned}
 & \min \quad \frac{1}{2} x^T Q x + d^T x \\
 & \text{s.t.} \quad H x \geq h, \\
 & \quad Ex = 0, \\
 & \quad x \geq 0.
 \end{aligned} \tag{15}$$

We introduce a penalty factor ρ to add the equality constraint $Ex = 0$ into the objective function in Problem (15). Thus, Problem (15) is converted as

$$\begin{aligned}
 & \min \quad \frac{1}{2} x^T Q x + d^T x + \rho x^T E^T E x \\
 & \text{s.t.} \quad H x \geq h, \\
 & \quad x \geq 0.
 \end{aligned} \tag{16}$$

Similar to [4], it is easy to prove that $Q + \rho E^T E$ is a symmetric positive definite matrix, and we use the MMSIM solver to solve Problem (16). The time complexity of the MMSIM is only $O(n)$ [2], where n is the number of variables in Problem (16). In addition, according to Theorem 2 of [4], we also have the following theorem:

THEOREM 1. *The solution generated by the MMSIM solver gives the optimal solution of Problem (16), where the solver runs in linear time to the number of variables.*

4.4 Cell Allocation & Cell Refinement

4.4.1 Cell Allocation. Since we ignore the right-boundary constraint in Problem (14), there could exist illegal cells. The MIA-aware allocation aims at aligning cells to placement sites and place them within the chip boundary. We handle inter-row violations by introducing a *forbidden region* to MIA-violating cells. The width of a forbidden region of a HVT/LVT cell c_i is ω , and its bottom-right coordinate equals the top-right coordinate of c_i . The bottom-left coordinate of each HVT/LVT cell should not be placed into any

Table 1: Comparisons of the Placement Results Between ICCAD’17 [18] and Our MIA-aware Legalization (Our MIA-LG)

Benchmark	Statistics			Initial result		ICCAD’17 [18]				Our MIA-LG			
	#Cells	#Nets	Util.	HPWL (nm)	Area (um ²)	Disp. (sites)	HPWL (nm)	Area (um ²)	CPU (s)	Disp. (site)	HPWL (nm)	Area (um ²)	CPU (s)
mgc_des_perf_1	112644	112878	0.90	1292055150	198025	2061736	1421429000	199004	90.31	988607	1350602869	198025	72.38
mgc_des_perf_2	112644	112878	0.85	1306015910	210681	1296639	1384343050	210681	63.73	954151	1357262280	210681	60.47
mgc_edit_dist_1	130661	133223	0.40	3921430290	521284	4348328	3934779425	521284	110.12	4251859	3928640505	521284	101.37
mgc_edit_dist_2	130661	133223	0.43	3827588190	484416	5490582	3782877700	484416	130.02	4373716	3768117220	484416	112.21
mgc_pci_bridge32_1	30675	30835	0.84	257692710	60614	355899	265307825	60614	17.60	341852	257769404	60614	14.96
mgc_pci_bridge32_2	30675	30835	0.85	270781205	59536	384515	279348525	59536	18.91	345106	264706004	59536	13.46
mgc_fft	32281	33307	0.83	460900783	70225	515386	475652650	70225	18.22	372752	455615172	70225	16.14
mgc_matrix_mult	155325	158527	0.80	2624515107	302500	2734723	2573253600	306900	175.67	1242035	2383335992	302500	98.65
N.Average						1.433	1.036	1.002	1.255	1.000	1.000	1.000	1.000

Table 2: Comparisons of the Placement Results Between Our Algorithm without Post-GP (Our MIA-LG) and with Post-GP (Ours)

Benchmark	Global placement result		Our MIA-LG				Ours				
	HPWL (nm)	Area (um ²)	Disp. (sites)	HPWL (nm)	Area (um ²)	CPU (s)	Disp. (site)	HPWL (nm)	Area (um ²)	CPU1 (s)	CPU (s)
mgc_des_perf_1	1267096825	198025	1561953	1381654134	198025	118.49	1452104	1322557019	198025	45.54	138.03
mgc_des_perf_2	1300464025	210681	1192429	1353972640	210681	96.93	1080772	1333105820	210681	44.39	128.32
mgc_edit_dist_1	3953144775	521284	12044818	3907566285	521284	127.32	11384837	3873850040	521284	122.35	225.14
mgc_edit_dist_2	3773458450	484416	12756012	3759781290	484416	135.13	12257158	3724430080	484416	112.73	221.17
mgc_pci_bridge32_1	203779910	60614	394914	245146810	60614	32.12	279497	229708365	60614	13.98	39.09
mgc_pci_bridge32_2	205311825	59536	364577	239857294	59536	31.24	280492	232664789	59536	14.54	38.24
mgc_fft	405248992	70225	429776	454099242	70225	33.65	323816	445067902	70225	18.96	37.84
mgc_matrix_mult	2198918567	302500	1789008	2353929342	302500	130.48	1383955	2317969942	302500	64.89	157.38
N.Average			1.201	1.027	1.000	0.768	1.000	1.000	1.000	0.423	1.000

forbidden region of a HVT/LVT cell. All cells are sorted in nondecreasing order of their x -coordinates, and are placed into the nearest legal placement sites by this order one by one. In this process, placed cells should avoid overlapping with others, and all HVT/LVT cells should get around the forbidden regions of HVT/LVT cells.

4.4.2 Cell Refinement. After the cell allocation stage, all intra and inter-row violations and cell overlaps are eliminated. In order to further improve our placement quality, we first fix the location of all HVT/LVT cells in this stage to avoid destroying the intra/inter-row violation-free results, then we use cell matching, cell moving, and cell swapping to minimize wirelength and chip area [5]. In order to avoid excessive movement, these operations are processed in a window:

- For two cells of the same height (and the same Vdd/Vss for even-row-height cells), if swapping them can reduce wirelength, then the swapping operation is performed.
- There may exist some blank spaces around a cell c_i . For a blank space with the same height (and the same Vdd/Vss for even-row-height cells) as c_i ’s, if moving c_i into this blank space can reduce wirelength, then the moving operation is performed.

5 EXPERIMENTAL RESULTS

We implemented our algorithm for the mixed-cell-height placement with complex MIA constraints in the C++ programming language, and tested it on the benchmarks provided by the authors [18]. These benchmarks were obtained by modifying Benchmark Suit A of the 2014 ISPD Detailed-Routing-Driven Placement Contest [1]. There are 20% of HVT/LVT cells (10% for HVT and 10% for LVT), and 15% of cells are selected to double or triple their heights randomly, while making their widths half or one-third to keep the same areas accordingly. The columns of “Statistics” gives the benchmark statistics in Table 1, where “#Cells” gives the number of movable standard cells, “#Nets” the number of nets, “Util%” the design utilization rates. In our experiments, the parameters κ in Equation (5), the clustering space cs in Algorithm 1, the distance R_c in 3.2, and the parameter ρ in Problem (16) were set as 0.1, $6Site_h$, $4Site_h$ and

1000, respectively. We conducted three experiments to examine the performance of our algorithm.

Thanks to the benchmarks and the binary code provided by the authors in [18], all the experiments in this paper were run on the same platform with Intel Core 3.40GHz CPU and 16GB memory for fair comparison.

5.1 Performance of MIA-Aware Legalization

Since the objective and framework of our MIA-aware legalization stage is similar to that of [18], to evaluate the effectiveness of our MIA-aware legalization, named “MIA-LG”, we compared our algorithm with the state-of-the-art work [18], named “ICCAD’17”. The experimental results are listed in Table 1. In the Table, “HPWL (nm)” gives the total wirelength in nanometer, “Area (um²)” the total areas of circuits, “Disp. (sites)” the cell displacement in the number of the placement site width, “CPU (s)” the total runtime. We used the same initial placement results as those in [18], the column “Initial result” gives the initial total wirelength and total area for each benchmark.

Compared with the placement results in “ICCAD’17”, “Our MIA-LG” can significantly reduce displacement. The total displacement in “ICCAD’17” is 1.433× larger than “MIA-LG”. Moreover, “Our MIA-LG” reduces the average HPWL by 3.6%, and resolves all the MIA violations without any area overhead for all benchmarks. In contrast, the algorithm in [18] increases the total areas for two benchmarks, e.g., circuits `mgc_des_perf_1` and `mgc_matrix_mult`. The average runtime of “MIA-LG” is also smaller than “ICCAD’17” by 25.5%. These results show that our MIA-aware legalization is effective and efficient.

5.2 Performance of Post-Global Placement

To evaluate the effectiveness of our post global placement, we compared our algorithm without Post-GP (Our MIA-LG) and with Post-GP (Ours). In this experiment, we used the global placement results generated by NTUplace4dr [10] directly as our initial placement results. In Table 2, “CPU1 (s)” and “CPU(s)” are the runtime of post global placement and the total runtime of our algorithm, respectively.

The results of our algorithm with post global placement in the column “Ours” of Table 2 justify the effectiveness of our post global placement. Compared with the experimental results in “Our MIA-LG”, “Ours” achieves 20.1% smaller cell displacement and 2.7% smaller average HPWL increase rate. In addition, with the post-GP stage, the runtime of the legalization stage in “Ours” is 19.1% less than that of “Our MIA-LG”. Hence, our post global placement stage not only reduces the difficulty of fixing the Vdd/Vss constraints and MIA constraints, but also legalizes cells more easily and effectively at the legalization step.

5.3 Performance of Our Algorithm

Based on the same global placement results generated by NTU-place4dr, we listed the final wirelength and their corresponding ratio between “ICCAD’17” and “Ours” in Table 3. It can be seen from the table that, our algorithm (“Ours”) can reduce the average HPWL by 8.5%. In particular, our algorithm does not incur any area overhead for all the benchmarks. It must be noted that, since the utilization rates of benchmarks `mgc_edit_dist_1` and `mgc_edit_dist_2` are very low (0.40 and 0.43, respectively), these cells can be legalized easily and directly. As a result, we have a lower improvement for these two benchmarks. Overall, these improvements show that our proposed algorithm is effective and efficient.

Table 3: Comparisons of Final HPWL Between ICCAD’17 [18] and Ours

Benchmark	HPWL/(nm)		Ratio
	ICCAD’17 [18]	Ours	
<code>mgc_des_perf_1</code>	1421429000	1322557019	1.07
<code>mgc_des_perf_2</code>	1384343050	1333105820	1.04
<code>mgc_edit_dist_1</code>	3934779425	3873850040	1.02
<code>mgc_edit_dist_2</code>	3782877700	3724430080	1.02
<code>mgc_pci_bridge32_1</code>	265307825	229708365	1.15
<code>mgc_pci_bridge32_2</code>	279348525	232664789	1.20
<code>mgc_fft</code>	475652650	445067902	1.07
<code>mgc_matrix_mult</code>	2573253600	2317969942	1.11
N.Average		1.000	1.085

Fig. 5 shows the placement layout for the circuit `mac_pci_bridge32_1` generated from our algorithm. Figs. 5(a) and 5(b) show the respective layout and a partial layout.

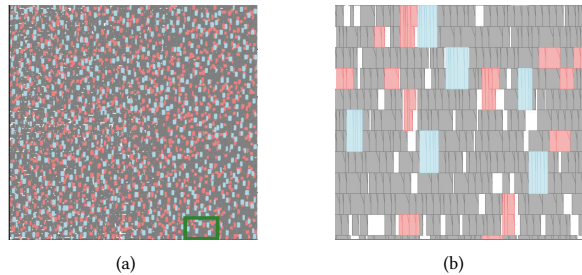


Figure 5: The layouts for the circuit `mgc_pci_bridge32_1`. (a) The layout of the final placement by our algorithm. (b) A partial layout of the final placement.

6 CONCLUSIONS

We have addressed the mixed-cell-height circuit designs with both MIA constraints and Vdd/Vss alignment constraints in the whole placement flow, and developed an effective and efficient algorithm

to solve the problem. We have presented a continuous and differentiable cost function to consider the mixed-cell-height standard cell Vdd/Vss alignment constraints, and added pseudo-nets to ensure cells of the same VT type to be placed as close as possible to reduce MIA violations. To minimize the wirelength, we have also developed a graph-based clustering algorithm and a special strip packing based BLP to eliminate intra-row MIA violations. We have transferred the inter-row MIA-aware legalization problem into a quadratic programming problem, and solved it by an MMSIM solver. Finally, we have also presented MIA-aware cell allocation and refinement to further improve our placement result. Experimental results have shown that our algorithm achieves better wirelength and runtime without any area overhead, compared with the state-of-the-art work.

REFERENCES

- [1] ISPD 2014 Detailed Routing-Driven Placement Contest. http://www.ispd2014_contest.html.
- [2] Z. Z. Bai. Modulus-based matrix splitting iteration methods for linear complementarity problems. *Numerical Linear Algebra with Applications*, 17(6):917–933, 2010.
- [3] J. Chen and W. Zhu. An analytical placer for VLSI standard cell placement. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 31(8):1208–1221, August 2012.
- [4] J. Chen, Z. Zhu, W. Zhu, and Y.-W. Chang. Toward optimal legalization for mixed-cell-height circuit designs. In *Proceedings of ACM/IEEE Design Automation Conference*, 2017.
- [5] 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 with preplaced blocks and density constraints. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27(7):1228–1240, July 2008.
- [6] W.-K. Chow, C.-W. Pui, and E. F. Y. Young. Legalization algorithm for multiple-row height standard cell design. In *Proceedings of ACM/IEEE Design Automation Conference*, 2016.
- [7] S. Dobre, A. B. Kahng, and J. Li. Mixed cell-height implementation for improved design quality in advanced nodes. In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pages 854–860, 2015.
- [8] S. Goto. An efficient algorithm for the two-dimensional placement problem in electrical circuit layout. *IEEE Transactions on Circuits and Systems*, 28(1):12–18, January 1981.
- [9] K. Han, A. B. Kahng, and H. Lee. Scalable detailed placement legalization for complex sub-14nm constraints. In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pages 867–873, 2015.
- [10] C.-C. Huang, H.-Y. Lee, B.-Q. Lin, S.-W. Yang, C.-H. Chang, S.-T. Chen, Y.-W. Chang, T.-C. Chen, and I. Bustany. NTUplace4dr: a detailed-routing-driven placer for mixed-size circuit designs with technology and region constraints. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, June 2017.
- [11] A. B. Kahng and H. Lee. Minimum implant area-aware gate sizing and placement. In *Proceedings of the Great Lakes Symposium on VLSI*, pages 57–62, 2014.
- [12] Y. Lin, B. Yu, X. Xu, J.-R. Gao, N. Viswanathan, W.-H. Liu, Z. Li, C. J. Alpert, and D. Z. Pan. MrDP: multiple-row detailed placement of heterogeneous-sized cells for advanced nodes. In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pages 7:1–7:8, 2016.
- [13] W.-K. Mak, W.-S. Kuo, S.-H. Zhang, S.-I. Lei, and C. Chu. Minimum implant area-aware placement and threshold voltage refinement. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 36(7):192–197, January 2017.
- [14] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, March 1957.
- [15] S. Martello, M. Monaci and D. Vigo. An exact approach to the strip-packing problem. *Inform Journal on Computing*, 15(3):310–319, August 2003.
- [16] K.-H. Tseng, Y.-W. Chang, and C. C. Liu. Minimum-implant-area-aware detailed placement with spacing constraints. In *Proceedings of ACM/IEEE Design Automation Conference*, pages 84:1–84:6, 2016.
- [17] C.-H. Wang, Y.-Y. Wu, J. Chen, Y.-W. Chang, S.-Y. Kuo, W. Zhu, and G. Fan. An effective legalization algorithm for mixed-cell-height standard cells. In *Proceedings of IEEE/ACM Asia and South Pacific Design Automation Conference*, pages 450–455, January 2017.
- [18] Y.-Y. Wu and Y.-W. Chang. Mixed-cell-height detailed placement considering complex minimum-implant-area constraints. In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, 2017.