



A Robust Modulus-Based Matrix Splitting Iteration Method for Mixed-Cell-Height Circuit Legalization

JIANLI CHEN, ZIRAN ZHU, and WENXING ZHU, Fuzhou University, China
YAO-WEN CHANG, National Taiwan University, Taiwan

Modern circuits often contain standard cells of different row heights to meet various design requirements. Taller cells give larger drive strengths and higher speed at the cost of larger areas and power. Multi-row height standard cells incur challenging issues for layout designs, especially the mixed-cell-height legalization problem with heterogeneous cell structures. Honoring the good cell positions from global placement, we present in this article a robust modulus-based matrix splitting iteration method (RMMSIM) to solve the mixed-cell-height legalization problem. Fixing the cell ordering from global placement and relaxing the right-boundary constraints, our proposed method first converts the problem into an equivalent linear complementarity problem (LCP), and then properly splits the matrices in the LCP so that the RMMSIM can solve the LCP optimally. The RMMSIM effectively explores the sparse characteristic of a circuit, and takes only linear time per iteration; as a result, it can solve the QP very efficiently. Finally, an allocation scheme for illegal cells is used to align such cells to placement sites on rows and fix the placement of out-of-right-boundary cells, if any. Experimental results show the effectiveness and efficiency of our proposed algorithm. In addition, the RMMSIM convergence and optimality are theoretically proved and empirically validated. In particular, this article provides a new RMMSIM formulation for various optimization problems that require solving large-scale convex quadratic programming problems efficiently.

CCS Concepts: • **Applied computing** → **Computer-aided design**;

Additional Key Words and Phrases: Physical design, placement, legalization, multi-row height cell, quadratic programming, linear complementarity problem, modulus-based matrix splitting iteration method

ACM Reference format:

Jianli Chen, Ziran Zhu, Wenxing Zhu, and Yao-Wen Chang. 2020. A Robust Modulus-Based Matrix Splitting Iteration Method for Mixed-Cell-Height Circuit Legalization. *ACM Trans. Des. Autom. Electron. Syst.* 26, 2, Article 15 (December 2020), 28 pages.
<https://doi.org/10.1145/3423326>

Preliminary version of this article was presented at the 2017 ACM/IEEE Design Automation Conference (DAC'17), Austin, TX, Jun 2017 [8], which received a best paper award from DAC'17.

This work was partially supported by the National Science Foundation of China under Grants 61672005 and 61977017, the Fujian Science Fund for Distinguished Young Scholars under Grant 2019J06010, and supported by AnaGlobe, IBM, MediaTek, TSMC, and MOST of Taiwan under Grant MOST 103-2221-E-002-259-MY3, MOST 104-2221-E-002-132-MY3, MOST 105-2221-E-002-190-MY3, MOST 106-2911-I-002-511, and MOST 106-2221-E-002-203-MY3, NTU under Grant NTU-ERP-104R8951, and NTU under Grant NTU-ERP-105R8951.

Authors' addresses: J. Chen, Z. Zhu, and W. Zhu, Fuzhou University, Fuzhou, China; emails: jlchen@fzu.edu.cn, ziranzhu24@gmail.com, wxzhu@fzu.edu.cn; Y.-W. Chang, National Taiwan University, Taipei, Taiwan; email: ywchang@ntu.edu.tw.

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.

© 2020 Association for Computing Machinery.

1084-4309/2020/12-ART15 \$15.00

<https://doi.org/10.1145/3423326>

1 INTRODUCTION

In traditional circuits, standard cells have the same height for easy design and optimization [27]. With the increasing complexity in modern circuit designs, however, standard cells often have different cell heights based on area, power, and speed characteristics. For example, taller cells give larger drive strengths, higher speed, and better pin accessibility and routability at the cost of larger areas and power, while shorter cells have smaller areas and power with weaker drive strengths, lower speed, and less pin accessibility and routability. As a result, mixed-cell-height standard cells become popular in addressing various design requirements, where simple standard cells are designed as the single-row height structure, while complex ones the multi-row height structure [1].

A modern placement flow typically consists of three major stages: (1) *Global Placement*: Ignoring possible cells overlaps, the goal of global placement intends to determine the best position for each cell without violating placement constraints (e.g., density) such that the target cost (e.g., wirelength, routability) is minimized; (2) *Legalization*: Align cells into rows and remove cell overlaps to minimize cell displacement (or wirelength increase); and (3) *Detailed Placement*: Refine the placement solution. Multi-row height standard cells incur challenging issues to placement, especially to mixed-cell-height legalization and detailed placement, due to the heterogeneous cell structures (which result in more global cell interferences and larger solution spaces) and additional power-rail constraints (as pointed out in [26]).

The general mixed-cell height legalization problem is NP-hard since it is a strip packing problem in nature [7]. Tetris [13], and Abacus [21] have been shown to be the most popular legalization methods for traditional single-row height standard cells. However, existing works reveal that they cannot be modified directly to handle multi-row height cells effectively. In single-row height standard-cell legalization, cell overlapping is independent among rows. With multi-row height cells, in contrast, shifting a cell in one row may cause cell overlaps in another row. Therefore, we need to consider cell overlapping in multiple rows when legalizing any multi-row height cell.

In a standard-cell design, further, power (VDD) and ground (VSS) lines are interleaved among cell rows, and each cell must be aligned correctly such that its power/ground pins match the corresponding rows. For an odd-row height cell (e.g., a single- or triple-row height cell), such alignment in a row can be achieved directly or by vertical cell flipping. In contrast, both sides of an even-row height standard cell (e.g., a double- or quadruple-row height cell) must be aligned with either power or ground pins, so it can only be aligned on every other row with a proper power rail to meet the power-rail alignment requirement. Figure 1 illustrates such mixed-cell-height standard-cell legalization considering the power-rail alignment issue (similar to that shown in [26]). As shown in this figure, the odd-row height cells *A* and *C* can be placed to any row by matching the correct VDD/VSS power rails directly or by flipping the cells vertically, while the even-row height cell *B* must match a power rail of the same type since its bottom boundary is designed for a VSS line. It is illegal that the bottom boundary of cell *B* is aligned to a VDD line, and further, such power-rail mismatch cannot be resolved by cell flipping.

1.1 Previous Work

For the mixed-cell-height legalization problem, Wu and Chu in [29] first handled designs with single- and double-row height standard cells. By cell pairing and cell expansion, they transformed all single-row height cells into double-row height cells. Thus, a traditional detailed placer could still be adopted. However, they did not consider the crucial power-rail alignment issue.

The mixed-cell-height standard-cell legalization works [9, 26] considered power-rail alignment. In [9], a cell is intended to be placed at the nearest site-aligned and power-rail-matched position from its global placement position. If the placement does not cause any overlap with any other

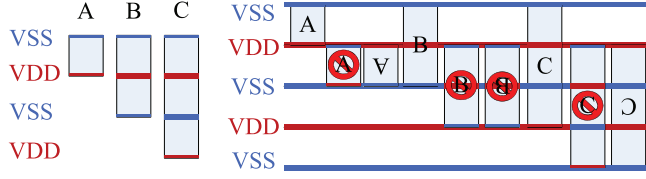


Fig. 1. Mixed-cell-height standard-cell legalization considering power-rail alignment.

cell, the cell is placed at the position directly; otherwise, a local region that can accommodate this cell is picked and the cell is placed into this region by a multi-row local legalization algorithm. This method is fast, but its solution quality may be limited because the selection of the region and legalization tend to be local. The recent work [18] first applied the legalizer in [9] to minimize the cell displacement from global placement and then developed a detailed placer for mixed-cell-height standard-cell designs that considers wirelength, cell density, and pin density.

The work [26] developed a new mixed-cell-height legalization algorithm that honors the good cell ordering from global placement. By theoretically analyzing the behaviors of Abacus for tackling the mixed-cell-height standard-cell legalization problem, this algorithm remedies Abacus's insufficiencies and extends its advantages, and is thus effective. To achieve high-quality legalization solutions, this work reveals the importance of preserving the original cell ordering from global placement.

The above works legalize cells or a cluster of cells one by one in a local region, which might limit the solution quality. To achieve better legalization solution, Hung et al. [14] first presented a parallel legalization method which uses integer linear programming (ILP) formulation to minimize total cell displacement. First, they partitioned a placement region into uniform bins, and spread cells from higher density bins to neighboring lower density ones. Then a legalization solution is obtained by using an ILP-based formulation. Finally, a linear-programming-based approach is performed to improve the solution quality. The proposed method can achieve high-quality solutions for the mixed-cell-height legalization problem. A major drawback is that, however, even with 28 threads, its runtime is almost 50× slower than the above works with only single thread.

In addition to the above legalizers [9, 14, 26] which focus on minimizing the total cell displacement, some recent mixed-cell-height legalization works optimized the maximum cell displacement and handled additional constraints such as minimum implant area [6, 30] and drain-to-drain abutment [23]. Specially, two recent works [16, 31] considered the 2017 CAD Contest at ICCAD on Multi-desk Standard Cell Legalization [11], which not only minimized the total cell displacement, but also the maximum cell displacement and detailed routing constraints.

Based on the multi-row local legalization algorithm in [9], Li et al. [16] presented the first work to consider the mixed-cell-height legalization problem with fence region and different routing constraints by a multi-row global legalization method and two post-processing network-flow-based techniques. Based on our conference version [8], Zhu et al. [31] developed a movement-aware cell reassignment method and a technology-aware legalization to legalize a placement, and proposed a technology-aware refinement to further reduce the total and maximum cell displacements. Note that both works are developed based on the previous works that focus on minimizing the total cell displacement, which is the most fundamental goal of the mixed-cell-height legalization problem. In this article, our main goal is to provide a new generic solution for various optimization problems that require solving large-scale convex quadratic programming (QP) problems efficiently by optimizing the fundamental goal of the legalization problem.

Obviously, the mixed-cell-height legalization problem is an extension of the strip packing problem, so it is NP-hard [7]. With proper preprocessing and relaxation, however, we can relax this

problem as a convex QP problem which can be solved by a QP solver [12]. There are two major types of methods for convex QP: primal-dual interior-point methods and active-set methods [19]. A primal-dual interior-point method maintains an approximate solution that lies strictly inside a feasible region defined by some inequality constraints. At each iteration, a Karush-Kuhn-Tucker (KKT) system involving all decision variables is solved, which requires $O((n + m)^3)$ time, where n and m denote the numbers of variables and constraints respectively [19]. An interior-point method generally needs only very few iterations regardless of its problem size. An active-set method distinguishes an active constraint from an inequality one. At each iteration, a KKT system defined by the active constraints is solved, which requires only $O((n + m)^2)$ time. Nevertheless, an active-set method typically needs many more iterations, compared with an interior-point one [19].

1.2 Our Contributions

Based on the modulus-based matrix splitting iteration method (MMSIM) [2], we convert the mixed-cell-height legalization problem to an equivalent linear complementarity problem (LCP), and develop a robust MMSIM (RMMSIM) to solve the corresponding LCP efficiently. The major contributions of our work are summarized below:

- Honoring the cell ordering from global placement and relaxing the right-boundary constraints, we formulate the fundamental mixed-cell-height legalization problem as a QP problem, and further convert it into an equivalent LCP.
- We split the matrices in the converted LCP in a proper manner, and use the RMMSIM to solve the converted LCP. The RMMSIM is an extension of the MMSIM, which only requires the system matrix in the converted LCP to be positive semi-definite.¹ We theoretically prove and empirically validate the convergence and optimality of our proposed RMMSIM.
- The RMMSIM effectively explores the sparse characteristic of a circuit and takes only $O(n + m)$ time per iteration, thus solving the LCP very efficiently. Unlike the existing methods that legalize cells one by one, further, the RMMSIM handles all mixed-row-height cells simultaneously, which provides a more global view of the legalization problem.
- Experimental results not only show the effectiveness and efficiency of our proposed algorithm, but also validate the convergence and optimality of our RMMSIM.
- In particular, this paper provides a new RMMSIM formulation for various optimization problems that require solving large-scale convex quadratic programs efficiently (e.g., global placement [25], buffering and wire sizing [10], dummy fill insertion [22], and analog circuit optimization [24].)

The remainder of this article is organized as follows. Section 2 formulates the legalization problem and introduces the LCP. Section 3 details the RMMSIM for the mixed-cell-height circuit legalization, and proves its convergence. Section 4 details the core techniques and framework of our legalization algorithm. Experimental results are shown in Section 5, and finally, the conclusion is given in Section 6.

2 PRELIMINARIES

In this section, we first formulate the mixed-cell-height standard-cell legalization problem and then introduce the linear complementarity problem.

¹A positive definite matrix is also positive semi-definite, but a positive semi-definite matrix is not necessarily positive definite.

2.1 Mixed-Cell-Height Circuit Legalization

In the mixed-cell-height legalization problem, we are given a chip with a global placement of n standard cells $C = \{c_1, \dots, c_n\}$, where each cell c_i has the respective height and width h_i and w_i , and the (bottom-left corner) coordinate (x'_i, y'_i) , $1 \leq i \leq n$. Each even-row height cell has a boundary power-rail type VDD or VSS. Like the works [8, 9, 14, 26], the fundamental goal of mixed-cell-height legalization in this paper is to place each cell c_i to a coordinate (x_i, y_i) , such that the total cell displacement is minimized and the following constraints are met:

- (1) Cells must be located at placement sites on rows (which are naturally inside the chip region);
- (2) Cells must be non-overlapping;
- (3) Cells must be aligned with correct power rails (and thus rows).

Similar to Abacus [21], the objective function is the quadratic cell displacement. Then, the mixed-cell-height legalization problem can be formulated as

$$\begin{aligned}
 \min \quad & \sum_{i=1}^n (x_i - x'_i)^2 + (y_i - y'_i)^2 \\
 \text{s.t.} \quad & (1) \text{ cells must be at placement sites on rows;} \\
 & (2) x_j - x_l \geq w_l, \text{ if } x_j \geq x_l \\
 & \quad \text{for all cells } j \text{ and } l \text{ in each row, for all rows;} \\
 & (3) \text{ cells must be aligned with correct power rails.}
 \end{aligned} \tag{1}$$

Constraint (3) requires that odd-row height cells be aligned in the rows (with vertical flipping to match the correct power/ground lines, if necessary), while even-row height cells additionally meet the power-rail requirement.

2.2 Linear Complementarity Problem

Given a large, sparse, and real matrix $A = (a_{ij})^{n \times n}$ and a real vector $q = (q_1, q_2, \dots, q_n)^T \in \mathbb{R}^n$, the goal of the linear complementarity problem [2], LCP(q, A) for short, is to find a pair of real vectors w and $z \in \mathbb{R}^n$ such that

$$w = Az + q \geq 0, \quad z \geq 0 \text{ and } z^T w = 0.$$

In this problem, the notation \geq means that a vector is greater than or equal to another one, and the superscript T denotes the transpose of a vector.

Many methods have been presented to solve the LCP, e.g., the projected successive overrelaxation iterations, the general fixed-point iterations, and the sequential and parallel splitting methods. A detailed comparison of these methods can be found in [2]. Among these methods, the modulus-based iteration method is considered the most effective and efficient [2].

The modulus-based iteration method for LCP(q, A) [2] is given as follows. Let $A = M - N$ be a splitting of the matrix A . Given an initial vector $s^{(0)} \in \mathbb{R}^n$, for $k = 0, 1, 2, \dots$, we compute $s^{(k+1)} \in \mathbb{R}^n$ by solving the linear system:

$$(M + \Omega)s^{(k+1)} = Ns^{(k)} + (\Omega - A)|s^{(k)}| - \gamma q, \tag{2}$$

and set

$$z^{(k+1)} = \frac{1}{\gamma} (|s^{(k+1)}| + s^{(k+1)}),$$

where Ω is an $n \times n$ positive diagonal matrix, and γ is a positive constant. Bai [2] has proven convergence of the modulus-based matrix splitting iteration method (MMSIM) when some conditions are satisfied, e.g., both of the matrices $A \in \mathbb{R}^{n \times n}$ and $M \in \mathbb{R}^{n \times n}$ are positive definite.

3 ROBUST MMSIM

The target of mixed-cell-height standard-cell legalization is to place cells into rows and remove cell overlaps such that the total cell displacement is minimized. The general mixed-cell-height legalization problem formulated in Equation (1) is NP-hard since it is a strip packing problem in nature [7]. Therefore, it is computationally prohibitive to solve Equation (1) directly. In order to build a mathematical formulation which handles all mixed-row-height cells simultaneously and solve the problem quickly, we first preprocess the macros and placement blockages. If there are movable macros in a circuit, we will legalize them one by one (find the nearest legal position) and then fix them. Standard cells overlapping with macros/blockages are then moved out of the macros/blockages with their shortest distances. Then, the position of each cell c_i is updated to (x_i^0, y_i^0) from (x_i', y_i') . Finally, we assign each cell to the correct row (to be elaborated in Section 4.4), and the position of each cell c_i is updated from (x_i^0, y_i^0) to (x_i^1, y_i^1) . For an odd-row height standard cell, the correct row can be any desired row; for an even-row height standard cell, the correct row must match the power rail constraint.

After the preprocessing, according to the cell positions x_i^1 , for $i = 1, 2, \dots, n$, we can obtain a cell ordering O by sorting them in non-decreasing order. Then, by relaxing the right-boundary, macros/blockages and placement-site constraints, the mixed-cell-height legalization problem in (1) can be relaxed as:

$$\begin{aligned} \min \quad & \frac{1}{2} \sum_{i=1}^n (x_i - x_i^0)^2 \\ \text{s.t.} \quad & x_j - x_l \geq w_l, \text{ if cell } c_j \text{ is on the right of cell } c_l \text{ in } O, \\ & \text{for all adjacent cells } c_j \text{ and } c_l \text{ in the same row;} \\ & x \geq 0. \end{aligned} \tag{3}$$

In the above formulation, since the objective is the sum of $(x_i - x_i^0)^2$, not the sum of $(x_i - x_i')^2$, it is hard for cells to move into the blockages again after solving this formulation. In addition, the constraints assure that there are no overlaps among the cells. It should be noted that a multi-row height standard cell could be considered multiple times in the constraints because it occupies several rows. In contrast, a single-row height standard cell occupies only one row and will be considered at most twice in the constraints.

In this section, we shall first convert the convex quadratic programming (QP) problem to an equivalent linear complementarity problem (LCP), and then develop an RMMSIM to solve the LCP. Finally, we will prove the convergence of our RMMSIM.

3.1 Convert QP into an Equivalent LCP

The legalization Problem (3) can be generalized as the following QP problem:

$$\begin{aligned} \min \quad & \frac{1}{2} x^T Q x + p^T x \\ \text{s.t.} \quad & Bx \geq b, \quad x \geq 0, \end{aligned} \tag{4}$$

where $Q \in \mathbb{R}^{n \times n}$ is an objective matrix, $B \in \mathbb{R}^{m \times n}$ is the constraint matrix, $p \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$ are two vectors.

According to the Karush-Kuhn-Tucker (KKT) conditions [4], if x is a global optimal solution of Problem (4), there exist vectors r and u such that the triple (x, r, u) satisfies the following KKT conditions:

$$\begin{cases} Qx + p - B^T r - u = 0, \\ v = Bx - b, \quad r^T v = 0, \\ u^T x = 0, \quad x, r, u, v \geq 0. \end{cases} \tag{5}$$

Equation (5) can be rewritten as the LCP(q, A) as follows:

$$w = Az + q \geq 0, \quad z \geq 0 \text{ and } z^T w = 0, \quad (6)$$

where $w = \begin{bmatrix} u \\ v \end{bmatrix}$, $A = \begin{bmatrix} Q & -B^T \\ B & 0 \end{bmatrix}$, $z = \begin{bmatrix} x \\ r \end{bmatrix}$, $q = \begin{bmatrix} p \\ -b \end{bmatrix}$.

We have the following theorem:

THEOREM 1. *An optimal solution of the convex QP problem (4) gives the KKT point of LCP(q, A) (6), and vice-versa.*

PROOF 1. According to the first-order optimality conditions [4], an optimal solution of Problem (4) satisfies the KKT conditions (5), which can be reformulated as LCP (6).

Conversely, suppose that $(w^*, z^*) = (\begin{bmatrix} u^* \\ v^* \end{bmatrix}, \begin{bmatrix} x^* \\ r^* \end{bmatrix})$ is a solution of Problem (6), and x^* is the KKT point. Then it satisfies

$$w^* = Az^* + q \geq 0, \quad z^* \geq 0 \text{ and } z^{*T} w^* = 0,$$

which is

$$u^* = Qx^* + p - B^T r^*, \quad (7a)$$

$$v^* = Bx^* - b, \quad (7b)$$

$$r^{*T} v^* + u^{*T} x^* = 0, \quad (7c)$$

$$x^*, r^*, u^*, v^* \geq 0. \quad (7d)$$

Since x^*, r^*, u^* , and v^* are all greater than or equal to zero, we have

$$r^{*T} v^* \geq 0, u^{*T} x^* \geq 0,$$

which together with Equation (7c) implies that

$$r^{*T} v^* = 0, u^{*T} x^* = 0.$$

With the above equations and (7a)–(7d), x^*, r^*, u^* , and v^* satisfy (5). According to Theorem 16.4 of [20], x^* is an optimal solution of the convex QP problem (4). \square

3.2 RMMSIM for LCP

In this subsection, we develop an RMMSIM for the converted LCP(q, A) which only requires a system matrix to be positive semi-definite, where the convergence and optimality of the RMMSIM can be proved relatively easily, and the parameter selection for RMMSIM is more flexible than that for the MMSIM.

A positive semi-definite matrix A can be approximated by a positive definite sequence $A(\epsilon)$, where ϵ is a sufficiently small constant. Then the LCP sequence $LCP(q, A(\epsilon))$ is closely related to the unperturbed $LCP(q, A)$, which is characterized by the following lemma in Proposition 7.5.9 of [28]:

LEMMA 1. *In LCP(q, A) (6), suppose $Q \in \mathbb{R}^{n \times n}$ is symmetric positive semi-definite. The following statements are equivalent [28]:*

- (1) *The QP problem (4) has an optimal solution, and the set of KKT pairs (z, w) is bounded.*
- (2) *There exist positive scalars $\hat{\epsilon}$ and L such that for every $(\hat{Q}, \hat{B}, \hat{p}, \hat{b})$ satisfying*

$$\|(\hat{Q}, \hat{B}, \hat{p}, \hat{b}) - (Q, B, p, b)\| \leq \hat{\epsilon},$$

ALGORITHM 1: RMMSIM for Problem (4)**Input:** matrices: $M(\varepsilon), N, A(\varepsilon), \Omega$ vectors: $q, \bar{s}^{(0)}, z^{(0)}$ constants: $\gamma, \tilde{\varepsilon}$ **Output:** $z^{(k)}$

```

1:  $k = 0$ ;
2: do
3:   solve  $[M(\varepsilon) + \Omega]\bar{s}^{(k+1)} = N\bar{s}^{(k)} + [\Omega - A(\varepsilon)]|\bar{s}^{(k)}| - \gamma q$ ;
4:    $z^{(k+1)} = \frac{1}{\gamma}(|\bar{s}^{(k+1)}| + \bar{s}^{(k+1)})$ ;
5:    $k + +$ ;
6: until  $|z^{(k)} - z^{(k-1)}| < \tilde{\varepsilon}$ 
7: return  $z^{(k)}$ .

```

the associated QP problem (4) with the data $(\widehat{Q}, \widehat{B}, \widehat{p}, \widehat{b})$ has the KKT pair $(\widehat{z}, \widehat{w})$; moreover, if \widehat{Q} is positive semi-definite, then for any KKT pair $(\widehat{z}, \widehat{w})$ of the perturbed QP, there exists a corresponding pair (z, w) of KKT vectors of the unperturbed program (4) such that

$$\|(\widehat{x}, \widehat{w}) - (x, w)\| \leq L \|(\widehat{Q}, \widehat{B}, \widehat{p}, \widehat{b}) - (Q, B, p, b)\|.$$

By Lemma 1, it is still possible to solve the LCP(q, A) (6) even if matrix A is positive semi-definite. In order to solve the LCP(q, A) with a positive semi-definite system matrix A robustly, we introduce a sufficiently small constant ε and set $A(\varepsilon) = A + \varepsilon I_{n+m}$, where $I_{n+m} \in \mathbb{R}^{(n+m) \times (n+m)}$ is an identity matrix. It is obvious that $A(\varepsilon)$ is positive definite, and LCP(q, A) can be converted to LCP($q, A(\varepsilon)$). We will prove that the solution of LCP($q, A(\varepsilon)$) approaches the solution of LCP(q, A) in Section 3.3.

Based on the MMSIM for LCP in [2], Algorithm 1 gives the RMMSIM for the QP problem (4). Let $A(\varepsilon) = M(\varepsilon) - N$ be a splitting of the matrix $A(\varepsilon)$. Given an initial vector $\bar{s}^{(0)} \in \mathbb{R}^{n+m}$, a positive diagonal matrix Ω , a positive constant γ , and a sufficiently small positive threshold $\tilde{\varepsilon}$, for $k = 0, 1, 2, \dots$, we compute $\bar{s}^{(k+1)}$ by solving the linear system in Line 3. That is, we solve

$$[M(\varepsilon) + \Omega]\bar{s}^{(k+1)} = N\bar{s}^{(k)} + [\Omega - A(\varepsilon)]|\bar{s}^{(k)}| - \gamma q, \quad (8)$$

and calculate $z^{(k+1)}$ in Line 4, until the iteration sequence $\{z^{(k)}\}$ converges. Then the optimal solution of Problem (4) is the first n components of the converged solution.

Let $I_n \in \mathbb{R}^{n \times n}$ and $I_m \in \mathbb{R}^{m \times m}$ be two identity matrices. For the linear system (8), similar to the Uzawa splitting in [2], we choose the splitting matrices $M(\varepsilon)$ and N as

$$M(\varepsilon) = \begin{bmatrix} \frac{1}{\beta^*}Q + \varepsilon I_n & 0 \\ B & \frac{1}{\theta^*}D + \varepsilon I_m \end{bmatrix}, N = \begin{bmatrix} (\frac{1}{\beta^*} - 1)Q & B^T \\ 0 & \frac{1}{\theta^*}D \end{bmatrix}, \quad (9)$$

where $D = \text{tridiag}(B(Q + \varepsilon I_n)^{-1}B^T)$ is a tridiagonal approximation to the Schur complement $B(Q + \varepsilon I_n)^{-1}B^T$ of the matrix $A(\varepsilon)$. Here B is required to be of full row rank [2]. Further, β^* and θ^* are two positive constants. According to the selection of the positive diagonal matrix Ω , we analyze in Appendix the ranges of the two parameters β^* and θ^* to guarantee the convergence of our RMMSIM.

3.3 Convergence Analysis of the RMMSIM

In this subsection, we prove the convergence of our RMMSIM with the positive semi-definite matrix A . Let $\{s^k\}$ denote the iteration sequence generated by the linear system (2) in Section 2.2, and s^* be the converged solution.

The key iteration of the RMMSIM can be characterized by the linear system (8). Let $\{\bar{s}^k\}$ denote the iteration sequence generated by the linear system (8), and \bar{s}^* be the converged solution.

For convenience, let

$$\begin{aligned}\delta(\Omega) &= \|(\Omega + M)^{-1}N\|_{\Omega^{\frac{1}{2}},2}, \\ \eta(\Omega) &= \|(\Omega + M)^{-1}(\Omega - M)\|_{\Omega^{\frac{1}{2}},2}, \\ \rho_2(\Omega) &= 2\delta(\Omega) + \eta(\Omega), \\ \delta(\Omega, \varepsilon) &= \|(\Omega + M(\varepsilon))^{-1}N\|_{\Omega^{\frac{1}{2}},2}, \\ \eta(\Omega, \varepsilon) &= \|(\Omega + M(\varepsilon))^{-1}(\Omega - M(\varepsilon))\|_{\Omega^{\frac{1}{2}},2}, \\ \rho_2(\Omega, \varepsilon) &= 2\delta(\Omega, \varepsilon) + \eta(\Omega, \varepsilon).\end{aligned}$$

For a non-singular matrix $P \in \mathbb{R}^{n \times n}$, let the matrix norm $\|X\|_{P,\#} = \|PXP^{-1}\|_{\#}$, the vector norm $\|x\|_{P,\#} = \|PX\|_{\#}$.

LEMMA 2. Suppose that Ω is a positive definite diagonal matrix. If $\rho_2(\Omega, \varepsilon) \leq 1$, then $\|\bar{s}^{k+1} - \bar{s}^*\|_{\Omega^{\frac{1}{2}}} \leq \|\bar{s}^k - \bar{s}^*\|_{\Omega^{\frac{1}{2}}}$.

PROOF 2. Similar to the proof of Theorem 4.1 in [2], let $N_{\Omega} = \Omega^{-\frac{1}{2}}N\Omega^{-\frac{1}{2}}$, $M_{\Omega}(\varepsilon) = \Omega^{-\frac{1}{2}}M(\varepsilon)\Omega^{-\frac{1}{2}}$, $\hat{s} = \Omega^{\frac{1}{2}}s$, and $\hat{\bar{s}} := \Omega^{\frac{1}{2}}\bar{s}$. As \bar{s}^* solves the linear system (8), we have

$$(M(\varepsilon) + \Omega)\bar{s}^* = N\bar{s}^* + (\Omega - A(\varepsilon))|\bar{s}^*| - \gamma q. \quad (10)$$

Subtracting Equation (10) from Equation (8), we get

$$(I + M_{\Omega}(\varepsilon))(\hat{\bar{s}}^{k+1} - \hat{\bar{s}}^*) = N_{\Omega}(\hat{\bar{s}}^k - \hat{\bar{s}}^*) + (I - M_{\Omega}(\varepsilon) + N_{\Omega})(|\hat{\bar{s}}^k| - |\hat{\bar{s}}^*|),$$

which implies $\|\hat{\bar{s}}^{k+1} - \hat{\bar{s}}^*\| \leq (2\delta(\Omega, \varepsilon) + \eta(\Omega, \varepsilon))\|\hat{\bar{s}}^k - \hat{\bar{s}}^*\| = \rho_2(\Omega, \varepsilon)\|\hat{\bar{s}}^k - \hat{\bar{s}}^*\| \leq \|\hat{\bar{s}}^k - \hat{\bar{s}}^*\|$. \square

LEMMA 3. If $\rho_2(\Omega, \varepsilon) < 1$, then $\{\bar{s}^k\}$ is bounded.

PROOF 3. According to Lemma 1, both the KKT pair (z, w) of the unperturbed QP (4) and the KKT pair (\hat{z}, \hat{w}) of the corresponding perturbed QP are bounded, so $\{\bar{s}^*\}$ is bounded. Furthermore, by Lemma 2, $\{\bar{s}^k\}$ is also bounded. \square

LEMMA 4. Let $A = M - N$ be a splitting of the positive semi-definite matrix A , where M is positive semi-definite, and Ω is a diagonal positive definite matrix. Suppose that A and M are perturbed as follows:

$$A(\varepsilon) = A + \varepsilon I, \quad M(\varepsilon) = M + \varepsilon I,$$

where I is an identity matrix. Suppose that $\rho_2(\Omega) < 1$, then the gap between the two solutions s^* and \bar{s}^* depends on the choice of ε , and will be sufficiently small.

PROOF 4. As s^* solves $LCP(q, A)$, we have

$$(M + \Omega)s^* = Ns^* + (\Omega - A)|s^*| - \gamma q. \quad (11)$$

On the other hand, the fixed-point iteration (8) can be rewritten as

$$(M + \Omega + \varepsilon I)\bar{s}^* = N\bar{s}^* + (\Omega - A - \varepsilon I)|\bar{s}^*| - \gamma q. \quad (12)$$

Deducing Equation (12) from Equation (11), we get

$$(M + \Omega)s^* - (M + \Omega + \varepsilon I)\bar{s}^* = Ns^* - N\bar{s}^* + (\Omega - A)|s^*| - (\Omega - A - \varepsilon I)|\bar{s}^*|.$$

Then we can further get the following equation

$$(\Omega + M)(s^* - \bar{s}^*) = N(s^* - \bar{s}^*) + (\Omega - A)(|s^*| - |\bar{s}^*|) + \varepsilon \bar{s}^* + \varepsilon |\bar{s}^*|,$$

and then we have

$$\begin{aligned}(s^* - \bar{s}^*) &= (\Omega + M)^{-1}N(s^* - \bar{s}^*) + (\Omega + M)^{-1}N(|s^*| - |\bar{s}^*|) \\ &\quad + (\Omega + M)^{-1}(\Omega - M)(|s^*| - |\bar{s}^*|) + \varepsilon(\Omega + M)^{-1}(\bar{s}^* + |s^*|).\end{aligned}$$

By Lemma 2, we can derive that

$$\begin{aligned}\|s^* - \bar{s}^*\|_2 &\leq \|(I + M_\Omega)^{-1}N_\Omega\|_2\|s^* - \bar{s}^*\|_2 + \|(I + M_\Omega)^{-1}N_\Omega\|_2\|s^* - \bar{s}^*\|_2 \\ &\quad + \|(I + M_\Omega)^{-1}(I - M_\Omega)\|_2\|s^* - \bar{s}^*\|_2 + 2\varepsilon\|(I + M_\Omega)^{-1}\|_2\|\bar{s}^*\|_2.\end{aligned}$$

The above inequality is equivalent to

$$\|s^* - \bar{s}^*\|_2 \leq \rho_2(\Omega)\|s^* - \bar{s}^*\|_2 + 2\varepsilon\|(I + M_\Omega)^{-1}\|_2\|\bar{s}^*\|_2,$$

where $\rho_2(\Omega) = 2\|(I + M_\Omega)^{-1}N_\Omega\|_2 + \|(I + M_\Omega)^{-1}(I - M_\Omega)\|_2$. Since $\rho_2(\Omega) < 1$, we can rearrange the inequality and get that

$$\|s^* - \bar{s}^*\|_2 \leq \frac{2\varepsilon\|(I + M_\Omega)^{-1}\|_2\|\bar{s}^*\|_2}{1 - \rho_2(\Omega)}. \quad (13)$$

By Lemma 3, the sequence $\{s^k\}$ is bounded, and $\|\bar{s}^*\|_2$ is also bounded. Therefore, Inequality (13) can be written as

$$\|s^* - \bar{s}^*\|_2 \leq \varepsilon H,$$

where H is a positive constant. The above inequality shows the gap. \square

Finally, we have the following theorem:

THEOREM 2. *Suppose that A is a positive semi-definite matrix, and Ω is a diagonal positive definite matrix. Let ε be a sufficiently small positive constant. Suppose that A and M are perturbed as follows:*

$$M(\varepsilon) = M + \varepsilon I, \quad A(\varepsilon) = A + \varepsilon I,$$

where I is an identity matrix, and thus matrices $A(\varepsilon)$ and $M(\varepsilon)$ are positive definite. Suppose that $\rho_2(\Omega, \varepsilon) < 1$ and $\rho_2(\Omega) < 1$. If $\varepsilon \rightarrow 0$, then $\bar{s}^ \rightarrow s^*$, which is the solution of the linear system (8).*

PROOF 5. Note that when $\rho_2(\Omega, \varepsilon) < 1$, the iteration sequence $\{\bar{s}^k\}$ of the linear system (8) converges to \bar{s}^* . From Lemma 4, the gap between \bar{s}^* and s^* is εH . So $\bar{s}^* \rightarrow s^*$ if $\varepsilon \rightarrow 0$. \square

4 OUR LEGALIZATION ALGORITHM

Since the single-row height standard cell legalization problem is much easier to be reformulated, we shall convert this problem to a linear complementary one, split the matrices in the LCP to meet the convergence requirement of the RMMSIM, and use the method to solve the problem. Then, we shall extend this method to solve the mixed-cell-height standard-cell legalization problem. Finally, we shall give the overall flow of our mixed-cell-height standard-cell legalization algorithm.

4.1 Single-Row Height Standard Cells Legalization

If all standard cells are of single-row height, we can rewrite the legalization problem (3) as follows:

$$\begin{aligned}\min \quad & \frac{1}{2}x^T \bar{Q}x + \bar{p}^T x \\ \text{s.t.} \quad & \bar{B}x \geq \bar{b}, \quad x \geq 0,\end{aligned} \quad (14)$$

where \bar{Q} is an identity matrix, \bar{p} is a vector with the i th component $\bar{p}_i = -x_i^0$, \bar{B} is the constraint matrix with only 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 that of variables.

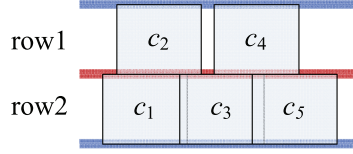


Fig. 2. A placement with single-row height cells.

PROPOSITION 1. In Problem (14), \bar{Q} is a symmetric positive definite matrix, and \bar{B} is of full row rank.

PROOF 6. Since \bar{Q} is an identity matrix, it is a symmetric positive definite matrix. Suppose \bar{B} is an $m \times n$ matrix, where m is the number of constraints and n is that of variables. If all standard cells are of single-row height and the right-boundary constraints are ignored, it is obvious that $m < n$. In addition, in matrix \bar{B} , there are only two nonzero elements -1 and 1 in each row and at most two nonzero elements in each column. We can choose the columns with element 1 in each row to form an m -order matrix, and the matrix can be further transformed to an identity matrix by elementary row transformation. Hence, the constraint matrix \bar{B} in Problem (14) is of full row rank. \square

For example, in Figure 2, cells c_2 and c_4 are aligned to row 1, and cells c_1 , c_3 , and c_5 to row 2. The constraint matrix is

$$\bar{B} = \begin{bmatrix} 0 & -1 & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix},$$

$$\bar{p} = [-x_1^0, -x_2^0, -x_3^0, -x_4^0, -x_5^0]^T, \bar{b} = [w_2, w_1, w_3]^T.$$

According to Section 3.1, Problem (14) can be converted into the following LCP via KKT conditions:

$$\bar{w} = \bar{A}\bar{z} + \bar{q} \geq 0, \quad \bar{z} \geq 0 \text{ and } \bar{z}^T \bar{w} = 0, \quad (15)$$

where $\bar{w} = [\frac{\bar{u}}{\bar{v}}]$, $\bar{A} = [\frac{\bar{Q}}{\bar{B}} \quad -\bar{B}^T]$, $\bar{z} = [\frac{x}{r}]$, $\bar{q} = [\frac{\bar{p}}{-\bar{b}}]$. Based on the discussion in Section 3.1, we have the following theorem:

THEOREM 3. An optimal solution of the convex QP problem (14) gives a solution of LCP (15), and vice-versa.

According to Theorem 3, Problem (14) is equivalent to the LCP(\bar{q}, \bar{A}) in Problem (15). Since \bar{Q} is a symmetric positive definite matrix and \bar{B} is of full row rank in Problem (14), we can perturb the positive semi-definite matrix \bar{A} to a positive definite matrix, and use the RMMSIM (Algorithm 1) in Section 3.2 to solve Problem (14).

4.2 Mixed-Cell-Height Standard Cells Legalization

In the mixed-cell-height legalization problem, a standard-cell library may contain cells of different cell heights, i.e., single-row height, double-row height, triple-row height, and so on. As mentioned above, to guarantee the RMMSIM convergence, the matrices in Problem (15) require \bar{Q} to be a symmetric positive semi-definite matrix and \bar{B} a rectangular matrix of full row rank. If there are multi-row height cells, however, the constraint matrix \bar{B} may not be of full row rank, and thus the RMMSIM convergence cannot be guaranteed.

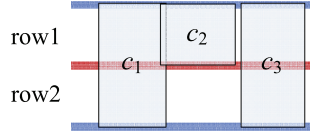


Fig. 3. A placement with mixed-cell-height cells.

For example, Figure 3 shows a placement with mixed-cell-height standard cells. By the definition in Problem (14), the constraint matrix $\bar{B} = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ -1 & 0 & 1 \end{bmatrix}$ is not of full row rank.

To guarantee the RMMSIM convergence for solving the mixed-cell-height standard-cell legalization problem, the corresponding legalization problem is reformulated as

$$\begin{aligned} \min \quad & \frac{1}{2}x^T \tilde{Q}x + \tilde{p}^T x \\ \text{s.t.} \quad & \tilde{B}x \geq \tilde{b}, \\ & Ex = 0, \\ & x \geq 0, \end{aligned} \quad (16)$$

where \tilde{Q} and \tilde{p} are the same as those in Problem (14). Matrix E is given in the following. If a cell c_i is of single-row height, we introduce the variable x_{i1} for this cell; otherwise, we introduce the variables $x_{i1}, x_{i2}, \dots, x_{id}$ for a multi-row height cell c_i , where d is equal to cell height h_i divided by the row height. The new constraint $Ex = 0$ ensures that the variables for each multi-row height standard cell are equal.

For the example shown in Figure 3 formulated as Problem (16), matrices

$$\tilde{B} = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 0 & 1 \end{bmatrix}, E = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix},$$

and the vectors $x = [x_{11}, x_{12}, x_{21}, x_{31}, x_{32}]^T$, $\tilde{p} = [-x_1^0, -x_1^0, -x_2^0, -x_3^0, -x_3^0]^T$, $\tilde{b} = [w_1, w_2, w_1]^T$. The matrix \tilde{B} is of full row rank.

For Problem (16), we introduce a penalty factor $\tilde{\lambda}$ to add the equality constraint into the objective function. Then Problem (16) can be reformulated as

$$\begin{aligned} \min \quad & \frac{1}{2}x^T \tilde{Q}x + \tilde{p}^T x + \tilde{\lambda}x^T E^T E x \\ \text{s.t.} \quad & \tilde{B}x \geq \tilde{b}, \quad x \geq 0. \end{aligned} \quad (17)$$

PROPOSITION 2. In Problem (17), $\tilde{Q} + \tilde{\lambda}E^T E$ is a symmetric positive definite matrix, and \tilde{B} is of full row rank.

PROOF 7. Since \tilde{Q} is an identity matrix and $E^T E$ is a positive semi-definite matrix, it is obvious that $\tilde{Q} + \tilde{\lambda}E^T E$ is a symmetric positive definite matrix. Suppose that \tilde{B} is an $m \times n$ matrix, where m is the number of constraints and n is the number of variables. With the newly added variables in multi-row cells, it is obvious that $m < n$, and there are only two nonzero elements -1 and 1 in each row and at most two nonzero elements in each column of the matrix \tilde{B} . Similar to the proof of Proposition 1, we can conclude that matrix \tilde{B} is of full row rank. \square

Since the matrix $\tilde{Q} + \tilde{\lambda}E^T E$ in Problem (17) is symmetric positive definite, x is the global optimal solution of Problem (17) if and only if there exist vectors \tilde{r} and \tilde{u} such that the triple $(x, \tilde{r}, \tilde{u})$ satisfies

the following KKT conditions:

$$\begin{cases} \tilde{Q}x + \tilde{p} + \tilde{\lambda}E^TEx - \tilde{B}^T\tilde{r} - \tilde{u} = 0, \\ \tilde{v} = \tilde{B}x - \tilde{b}, \quad \tilde{r}^T\tilde{v} = 0, \\ \tilde{u}^Tx = 0, \quad x, \tilde{r}, \tilde{u}, \tilde{v} \geq 0. \end{cases} \quad (18)$$

Similar to the single-row height legalization problem, we can rewrite Condition (18) as the following LCP:

$$\tilde{w} = \tilde{A}\tilde{z} + \tilde{q} \geq 0, \quad \tilde{z} \geq 0 \text{ and } \tilde{z}^T\tilde{w} = 0, \quad (19)$$

where $\tilde{w} = [\tilde{u}^T, \tilde{v}^T]^T$, $\tilde{A} = \begin{bmatrix} \tilde{Q} + \tilde{\lambda}E^TE & -\tilde{B}^T \\ \tilde{B} & 0 \end{bmatrix}$, $\tilde{z} = [\tilde{x}^T, \tilde{r}^T]^T$, $\tilde{q} = [\tilde{p}^T, -\tilde{b}^T]^T$. Similar to Theorem 1, we can also prove that the following theorem is correct.

THEOREM 4. *An optimal solution of the convex QP problem (17) gives a solution of LCP (19), and vice-versa.*

In LCP(\tilde{q}, \tilde{A}) (19), since $\tilde{Q} + \tilde{\lambda}E^TE$ is a symmetric positive definite matrix and \tilde{B} is of full row rank, we perturb the positive semi-definite matrix \tilde{A} to the positive definite matrix $\tilde{A}(\epsilon) = \begin{bmatrix} \tilde{Q} + \tilde{\lambda}E^TE + \epsilon I_n & -\tilde{B}^T \\ \tilde{B} & \epsilon I_m \end{bmatrix}$, and choose the splitting matrices $\tilde{M}(\epsilon)$ and \tilde{N} as

$$\tilde{M}(\epsilon) = \begin{bmatrix} \frac{1}{\beta^*}(\tilde{Q} + \tilde{\lambda}E^TE) + \epsilon I_n & 0 \\ \tilde{B} & \frac{1}{\theta^*}\tilde{D} + \epsilon I_m \end{bmatrix}, \quad \tilde{N} = \begin{bmatrix} (\frac{1}{\beta^*} - 1)(\tilde{Q} + \tilde{\lambda}E^TE) & \tilde{B}^T \\ 0 & \frac{1}{\theta^*}\tilde{D} \end{bmatrix}, \quad (20)$$

where $\tilde{D} = \text{tridiag}(\tilde{B}(\tilde{Q} + \tilde{\lambda}E^TE + \epsilon I_n)^{-1}\tilde{B}^T)$ is a tridiagonal approximation to the Schur complement $\tilde{B}(\tilde{Q} + \tilde{\lambda}E^TE + \epsilon I_n)^{-1}\tilde{B}^T$ of the matrix \tilde{A} , and β^* and θ^* are two positive constants determined by Theorem A.2 in the appendix. Then, the RMMSIM (Algorithm 1) in Section 3.2 can be used to solve Problem (17), and the convergence of the RMMSIM can be guaranteed.

In Equation (20), matrix \tilde{D} involves computation of the inverse of the matrix $\tilde{Q} + \tilde{\lambda}E^TE + \epsilon I_n$. In general, it is very time-consuming to calculate the inverse of a matrix. Fortunately, in our formulated problem, since \tilde{Q} and I_n are two identity matrix and E^TE exhibits good structures, the inverse of the matrix $\tilde{Q} + \tilde{\lambda}E^TE + \epsilon I_n$ can be obtained quickly by calculating the inverse of each block of $\tilde{Q} + \tilde{\lambda}E^TE + \epsilon I_n$.

We first consider the cases of the matrix E^TE for various row height cells in a circuit. Figures 4(a)-(d) show the matrix E^TE of a single-, double-, triple-, and quadruple-row height cell respectively. Figure 4(e) gives the general form of the matrix E^TE of a multi-row height cell, which is a tridiagonal matrix. The diagonal elements of the first row and last row are equal to 1, and the other diagonal elements are equal to 2. The elements on the tridiagonal but not on the diagonal are equal to -1.

In matrix E^TE , the component formed by each cell can be regarded as a block matrix of E^TE . For example, Figure 5(a) gives the placement with mixed-cell-height cells, and Figures 5(b) and 5(c) show the corresponding matrix E and matrix E^TE .

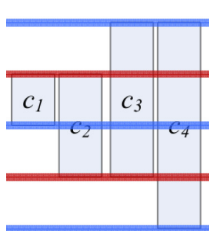
Since the cells of the same height correspond to the same block matrix, we can calculate the inverse of the block matrix corresponding to the cells with various heights in advance. Then we can directly write the inverse of matrix $\tilde{Q} + \tilde{\lambda}E^TE + \epsilon I_n$.

4.3 Complexity Analysis for the RMMSIM

In Algorithm 1, the major work is to solve the linear system in Line 3. With our splitting matrices $\tilde{M}(\epsilon)$ and \tilde{N} , the matrix $(\tilde{M}(\epsilon) + \tilde{N})$ in Line 3 has the following special structure (as shown in

$$\begin{aligned}
 (a) \quad E^T E &= [0] \\
 (b) \quad E^T E &= \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \\
 (c) \quad E^T E &= \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \\
 (d) \quad E^T E &= \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \\
 (e) \quad E^T E &= \begin{bmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & \dots & \dots & \dots & & \\ & & \dots & \dots & \dots & \\ & & & \dots & \dots & \dots \\ & & & & -1 & 2 & -1 \\ & & & & & -1 & 1 \end{bmatrix}
 \end{aligned}$$

Fig. 4. Illustration of Matrix $E^T E$. (a) Single-row height cell. (b) Double-row height cell. (c) Triple-row height cell. (d) Quadruple-row height cell. (e) Multiple-row height cell.



(a)

$$E = \begin{bmatrix} x_{11} & x_{21} & x_{22} & x_{31} & x_{32} & x_{33} & x_{41} & x_{42} & x_{43} & x_{44} \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

(b)

$$E^T E = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

(c)

Fig. 5. (a) A placement with mixed-cell-height cells. (b) The matrix E of the placement (a). (c) The matrix $E^T E$ of the placement (a).

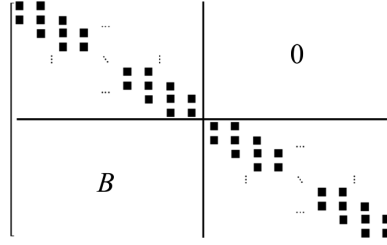


Fig. 6. The structure of Matrix $(M(\epsilon) + \Omega)$ in Algorithm 1, where the block matrices (1,1) and (2,2) are both tridiagonal, and Matrix B is a sparse matrix with only two nonzero elements -1 and 1 in each row.

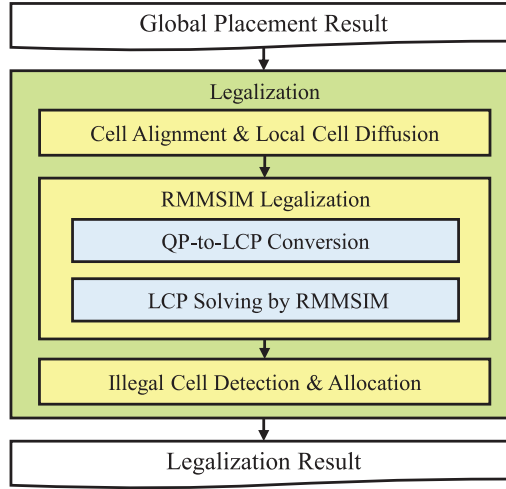


Fig. 7. Our legalization flow.

Figure (6): the blocks matrices (1,1) and (2,2) are both tridiagonal, and matrix B is a sparse matrix with only two nonzero elements -1 and 1 in each row.

Solving a linear system with tridiagonal matrices only needs linear time. To solve the linear system $(M(\epsilon) + \Omega)$, we can first solve the block matrix (1,1) in $O(n)$ time. After solving the block matrix (1,1), the first n variables are known. Then we substitute those variables in the block matrix B . Finally, we only need to solve the tridiagonal block matrix (2,2), which runs in $O(m)$ time. As a result, the total time complexity in each iteration is $O(n + m)$. It should be noted that our proposed method needs only few iterations regardless of the problem size, so the RMMSIM runs very fast.

4.4 Our Legalization Framework

Figure 7 shows the overall flow of our mixed-cell-height standard-cell legalization algorithm. The input is a global placement result, which computes the best position for each cell by ignoring overlaps among cells. Our legalization method is to align the cells into rows and remove cell overlaps such that the total cell displacement is minimized.

As shown in Figure 7, our legalization algorithm first aligns each cell to the nearest correct row. If there are movable macros in a circuit, we will legalize them one by one (find the nearest legal position) and then fix them. Standard cells overlapping with macros/blockages are also moved out of the macros/blockages with their shortest distances. After that, if the distribution of cells in a row is locally sparse, then the overlaps among these cells in the row can be desirably solved by

the RMMSIM legalization. However, if it is locally dense, then the final locations of some cells in the row may not be desirable. Therefore, to achieve a high-quality legalization solution, we further adopt local cell diffusion in dense regions to obtain a better cell assignment and a better cell ordering. Then, ignoring the right-boundary constraints, we convert the QP problem (17) into the equivalent LCP (19), and use the RMMSIM discussed in Section 3.2 to solve the LCP. Finally, we further align all the cells to placement sites on rows and use an illegal cell allocation approach to fix the illegal cells (including overlapping or out-of-right-boundary cells).

The local cell diffusion can be divided into three steps: (1) diffusion region identification, (2) budgeting, and (3) cell moving. In order to identify the local dense regions, we divide each row into smaller areas called bins, and the bins with density higher than 1 are treated as *overfilled bins* (i.e., dense regions). Then, for each overfilled bin, local cell diffusion is applied to spread the cells from overfilled bins to neighboring low-density bins.

For the diffusion region identification, we take the overfilled bin as the center and its row height as the basic expansion unit. The diffusion region r is gradually expanded from the overfilled bin until the density of r is equal to or smaller than $\min\{r_1, \max(r_2, \text{chipDensity} + r_3)\}$, where *chipDensity* represents the density of the chip, and r_1, r_2 and r_3 are user-defined parameters, which are set as 1, 0.8, and 0.05, respectively. In the worst case, the diffusion region r can be the entire chip region. Since the global placement stage aims at distributing cells evenly and finding a desired position for each cell to optimize the target cost (e.g., wirelength, routability), the region r would not be too large. Particularly, if the density of a circuit is large and the cells are not well spread, we can increase the values of r_1 and r_3 to control the region r .

For budgeting, we further divide the diffusion region r into an array of uniform bin grids, and calculate the density of each bin. Since most of the standard cells are of single row height, we set the height of each bin as one row height in this preprocessing step for fast processing. Then, a network flow model is adopted to determine the cell area for each bin to be moved out to the abutting bins. In the network flow model, we regard each bin as a node, and introduce a source node and a sink node (not a bin node). For each overfilled bin node (with its bin density greater than 1), it has an edge from the source node, the capacity and cost of the edge are set to the overfilled cell area in this bin and zero, respectively. For each underfilled bin node (with its bin density not more than 1), it has an edge to the sink node, the capacity and cost of the edge are set to the area in the free space in this bin and zero, respectively. For each bin node, it has up to four edges to the abutting bins with lower density, and the capacity and cost of the each edge are set to infinity and the Manhattan distance between the two bins. The network flow model is solved directly by [15]. Compared with the work [14] that constructs a network flow for the whole placement region with edges from an overfilled bins to the underfilled ones, our local cell diffusion is performed only for the identified locally dense regions and each bin node has up to only four edges to the abutting bins with lower density; as a result, our local cell diffusion can better preserve the quality of the given global placement within more reasonable computation time.

After budgeting, we obtain the budgeted amount of area for each bin to move from this bin to its abutting bins. In the local cell diffusion, each cell can only be moved from its original bin to an abutting bin and needs to meet the power-rail alignment constraints on rows. As mentioned above, we set the height of each bin to one row height for fast processing. If a multi-row height cell is moved, it could affect several bins, and thus makes the diffusion step much more difficult and may violate the power-rail alignment constraints. Hence, we only move single-row height cells in this step for simplification. For cell moving, we first define the moving cost for single-row height standard cell c_i as follows:

$$\text{cost}_i^d = \Delta c_{disp} + \alpha \Delta c_{overlap},$$

where $d \in \{L, R, B, T\}$ denotes the left, right, bottom, and top bins adjacent to the bin where cell c_i is located, Δc_{disp} represents the displacement change before and after the movement, and $\Delta c_{overlap}$ represents the overlap change between this cell and others cells before and after the movement. Then, we sort bins in non-ascending order according to their density. After that, for each bin, we consider the moving cells one by one from this bin to its abutting bins. For each cell c_i in a bin, we calculate the four costs (i.e., $cost_i^L, cost_i^R, cost_i^B$, and $cost_i^T$) of moving the cell from this bin to its abutting bins, and add these costs into a priority queue. Finally, we iteratively take out the least cost $cost_i^d$ (and thus the cell c_i and the moving direction d) from the priority queue. If moving the cell c_i to the target abutting bin will cause the budgeted amount of area from this bin to the target abutting bin to be exceeded, we will delete the cost from the priority queue and not move the cell; otherwise, we will delete all the costs associated with the cell c_i and move the cell to the target abutting bin.

The penalty factor $\tilde{\lambda}$ in Problem (17) affects the values of the variables for each multi-row height cell. Since the RMMSIM is a convergent method, if the value of $\tilde{\lambda}$ is large enough, the variables of each multi-row height cell will have equal values in theory. Due to the precision of computation, however, there may be differences in the values of the variables for each multi-row height cell. Hence, after solving Problem (17) by our RMMSIM, we first unify the values of the variables for each multi-row height cell. Then, we align each cell to the nearest placement site, and check the cells one by one for their legality. If a cell is overlapped with another cell/macro/blockage or out of the right boundary, it will be marked as an illegal cell and temporarily removed from the layout.

The illegal cell allocation approach first extracts all free spaces based on the placed cells positions, and place the illegal cells one by one. When placing an illegal cell, we check free spaces from the nearest one to the farthest distance from the cell. If the illegal cell is placed into a free space that does not cause any overlap with any other cell, then the cell is placed at the position directly. Otherwise, we allow the placed cells to move horizontally, i.e., we try to move the placed cells toward left or right to expand the free space on the same rows, and if the expanded free space can fully accommodate the cell, then the cell is placed at the expanded free space and the corresponding placed cells are updated to the new positions. However, if the expanded space cannot accommodate the cell, then we will check the next free space. In this way, all cells can be placed in the chip region legally, and we can prevent the illegal cell being moved too far away from its global placement position.

5 EXPERIMENTAL RESULTS

We implemented our proposed mixed-cell-height legalization algorithm in the C++ programming language and tested it on the benchmarks provided by the authors of [9]. The benchmarks were modified from the 2015 ISPD Detailed-Routing-Driven Placement Contest [5] by the authors of [9], which do not consider fence regions given in the original benchmarks, and 10% of the cells were randomly selected to double their heights and half their widths to form mixed-cell-height standard-cell benchmarks. This modification maintains the total cell area and ensures that each chip can accommodate all the cells. According to the work [9], the global placement results were obtained from the global placer provided by one of the top-3 winners of the ISPD 2015 contest [5]. We conducted five experiments and reported the results in the following subsections. In our algorithm, the parameter $\tilde{\lambda}$ in Problem (17) affects the precision of the variables in each multi-row height cell. Since the RMMSIM is a convergent method, if the value of $\tilde{\lambda}$ is large enough, the variables of each multi-row height cell will have equal values in theory. We set the value of $\tilde{\lambda}$ to 1,000 because the difference in the variable values for each multi-row height cell is very small (or even zero) by using this value. In addition, the values of β^* , θ^* , and ε in the splitting matrices $M(\varepsilon)$ and N were

Table 1. Statistics of the Benchmarks and Illegal Cells after the RMMSIM Legalization

Benchmark	#S. Cell	#D. Cell	Density	#I. Cell	%I. Cell
des_perf_1	103842	8802	0.91	416	0.369
des_perf_a	99775	8513	0.43	8	0.007
des_perf_b	103842	8802	0.50	0	0
edit_dist_a	121913	5500	0.46	32	0.025
fft_1	30297	1984	0.84	30	0.093
fft_2	30297	1984	0.50	0	0
fft_a	28718	1907	0.25	0	0
fft_b	28718	1907	0.28	8	0.026
matrix_mult_1	152427	2898	0.80	27	0.017
matrix_mult_2	152427	2898	0.79	17	0.011
matrix_mult_a	146837	2813	0.42	32	0.021
matrix_mult_b	143695	2740	0.31	17	0.012
matrix_mult_c	143695	2740	0.31	3	0.002
pci_bridge32_a	26268	3249	0.38	0	0
pci_bridge32_b	25734	3180	0.14	0	0
superblue11_a	861314	64302	0.43	734	0.079
superblue12	1172586	114362	0.45	771	0.060
superblue14	564769	47474	0.56	2716	0.444
superblue16_a	625419	55031	0.48	367	0.054
superblue19	478109	27988	0.52	588	0.116

respectively set to 0.5, 0.5, and 0.001, meeting the requirement of Theorem A.2 and also simplifying the calculation.

5.1 Illegal Cells after RMMSIM Legalization

Table 1 gives the characteristics of the benchmarks and the illegal cells after RMMSIM legalization (which need to be fixed by the illegal cell allocation). In this table, “#S. Cell” gives the total number of single-cell-height standard cells, “#D. Cell” the total number of double-cell-height standard cells, “Density” the density of a design, and “#I. Cell” and “%I. Cell” the total number and the percentage of illegal cells, respectively. It can be seen from Table 1 that, the ratios of illegal cells for most of the benchmarks are less than 0.1% (except benchmarks des_perf_1, superblue14, and superblue19). For the given cell assignment and cell ordering, since the RMMSIM can generate the minimum displacement for the QP problem (17), and only very few illegal cells, if not none, need to be fixed by the illegal cell allocation approach, our mixed-cell-height legalization algorithm can guarantee the optimal solutions for the benchmarks without illegal cells (such as the benchmarks des_perf_b, fft_2, fft_a, pci_bridge32_a, and pci_bridge32_b) and achieve near optimal solutions for other benchmarks.

5.2 Mixed-Cell-Height Legalization Comparisons

We compared our legalization algorithm with four state-of-the-art works [8, 9, 26], and [14]. All the four algorithms consider power-rail alignment. With the binary codes provided by the authors of [8, 26], and [9], we were able to run the three algorithms and our algorithm on the same PC with a 3.40-GHz Intel Core CPU and 16-GB memory, and the results of [14] were quoted from

Table 2. Experimental Results Compared with [9, 26, 14] and [8]

Benchmark	GP	Average Disp. (sites)					Δ HPWL (%)				Runtime (s)				
	HPWL	[9]	[26]	[14]	[8]	Ours	[9]	[26]	[8]	Ours	[9]	[26]	[14]	[8]	Ours
des_perf_1	1.43	2.48	4.21	1.44	1.54	1.44	1.77	0.99	0.87	0.74	6.05	7.53	1162	2.34	2.57
des_perf_a	2.57	0.75	0.67	0.65	0.67	0.65	0.16	0.12	0.07	0.07	2.47	3.83	27	2.27	1.51
des_perf_b	2.13	0.72	0.64	0.62	0.64	0.62	0.21	0.16	0.09	0.08	2.23	3.94	23	2.36	2.37
edit_dist_a	5.25	0.47	0.48	0.44	0.47	0.45	0.10	0.12	0.09	0.08	1.82	4.90	76	2.79	2.18
fft_1	0.46	1.69	1.65	1.06	1.18	1.07	1.47	0.89	0.93	0.86	0.95	1.27	215	0.63	0.49
fft_2	0.46	0.80	0.65	0.63	0.65	0.63	0.73	0.67	0.53	0.49	0.38	1.10	12	0.63	0.48
fft_a	0.75	0.64	0.59	0.56	0.58	0.57	0.33	0.29	0.24	0.23	0.23	1.18	17	0.58	0.40
fft_b	0.95	0.93	0.69	0.62	0.70	0.66	0.18	0.30	0.25	0.23	0.36	1.18	17	0.59	0.41
matrix_mult_1	2.39	0.52	0.47	0.38	0.41	0.39	0.27	0.21	0.22	0.19	3.97	5.40	148	3.48	2.72
matrix_mult_2	2.59	0.49	0.42	0.36	0.38	0.36	0.21	0.17	0.18	0.16	4.16	5.39	132	3.49	2.74
matrix_mult_a	3.77	0.31	0.27	0.25	0.28	0.26	0.11	0.09	0.09	0.08	1.62	5.73	55	3.24	2.03
matrix_mult_b	3.43	0.27	0.25	0.23	0.26	0.24	0.10	0.09	0.08	0.07	1.21	5.60	49	3.12	2.17
matrix_mult_c	3.29	0.29	0.28	0.27	0.28	0.26	0.11	0.11	0.10	0.09	1.39	5.61	58	3.15	2.14
pci_bridge32_a	0.46	0.94	0.90	0.86	0.89	0.88	0.57	0.63	0.45	0.46	0.28	1.22	10	0.57	0.31
pci_bridge32_b	0.98	0.96	0.90	0.90	0.91	0.90	0.13	0.06	0.05	0.05	0.17	1.01	9	0.58	0.30
superblue11_a	42.94	1.93	2.14	1.85	1.73	1.68	0.15	0.26	0.11	0.11	29.70	50.28	1798	25.21	18.64
superblue12	39.23	1.57	1.55	1.33	1.40	1.34	0.20	0.22	0.16	0.14	103.61	56.47	1487	37.69	27.18
superblue14	27.98	2.61	2.45	2.18	2.43	2.22	0.22	0.18	0.21	0.16	16.68	48.11	3436	16.99	14.75
superblue16_a	31.35	1.72	1.69	1.61	1.59	1.53	0.11	0.11	0.09	0.08	20.66	41.83	965	18.05	15.38
superblue19	20.76	1.59	1.60	1.48	1.44	1.37	0.14	0.13	0.11	0.10	10.50	29.59	929	12.71	10.72
Normalized		1.225	1.207	1.000	1.048	1.000	1.565	1.375	1.096	1.000	1.263	2.644	91.874	1.363	1.000

the publication directly because its binary code is not available to us. It should be noted that the binary code for [9] is an improved version after the conference, which can achieve better results than those reported in the DAC'16 paper.

The experimental results are shown in Table 2. In the table, “GP HPWL” gives the wirelength in unit meter of the global placement result, “Average Disp. (sites)” the cell displacement measured in the number of the placement site width, “ Δ HPWL(%)” the HPWL increase from the corresponding global placement, and “Runtime(s)” the running time in second excluding the file I/O operation. The last row of Table 2 shows the average normalized average displacement, Δ HPWL, and runtime ratios based on our results. Since the work [14] does not report its resulting HPWL, we omit [14] for the HPWL comparison.

As can be seen from Table 2, our legalization algorithm can achieve better solution quality and faster runtime than the works [9, 26, 8]. In addition, although the work [14] can achieve almost the same solution quality as our algorithm, the runtime of [14] is 91 \times slower than our algorithm, even with 28 threads.

Compared with the works [9] and [26], our legalization algorithm achieves 22.5% and 20.7% smaller cell displacement, 56.5% and 37.5% smaller HPWL increase rate, and 1.26 \times and 2.64 \times faster runtime, respectively. Both works in [9] and [26] optimized cells one by one in local regions, thus with only a local view of the total cell legalization. In contrast, our algorithm optimizes all cells simultaneously by using the RMMSIM legalization, leading to the best solution quality among the three algorithms and much stable results.

Compared with our earlier version [8] (which received the best paper award at DAC'17), our algorithm achieves 4.8% smaller cell displacement, 9.6% smaller HPWL increase rate and 1.36 \times

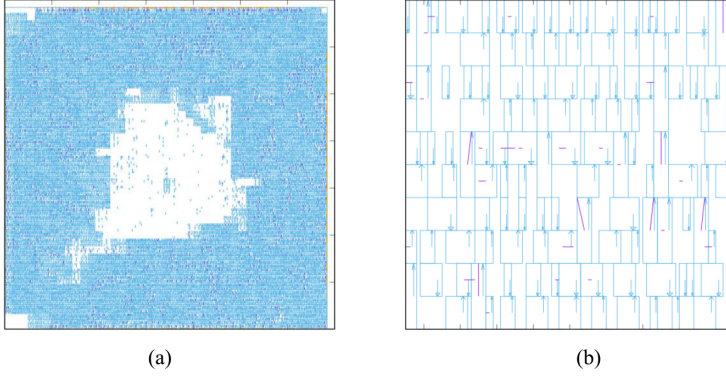


Fig. 8. (a) Legalization result of the benchmark `fft_2`. Cells are in blue, and displacement in purple. (b) Partial layout of (a).

faster runtime. We ascribe the improvements in solution quality to the following two aspects: (a) network-flow-based local cell diffusion in high-density regions, which gives the better cell assignment and ordering for RMMSIM legalization than directly aligning cells to the nearest correct rows used in the work [8]; (b) illegal cells allocation, which allows the legalized cell movement to create free space for illegal cells, and thus minimizes the distance of illegal cell movement, resulting in more effective legalization than the Tetris-like allocation used in [8]. In addition, our algorithm also runs faster than [8] because (1) we can directly compute the inverse of matrix $\tilde{Q} + \tilde{\lambda}E^TE + \epsilon I_n$, (2) the local cell diffusion gives better initial solutions for our RMMSIM legalization, (3) our RMMSIM can converge for the LCP much more quickly and thus save the computation time.

Compared with [14], our algorithm with single-thread achieves a $91.87\times$ speedup over the work [14] even with 28 threads, while the solution quality is about the same on average. The major reason is that the work [14] is based on an ILP formulation, while our legalization algorithm is based on QP and our RMMSIM effectively explores the sparse characteristic of a circuit and solve the QP very efficiently. In addition, both algorithms legalize multiple cells simultaneously, and thus can achieve high solution quality for the mixed-cell-height legalization problem.

Overall, the experimental results show that our proposed legalization algorithm is effective and efficient. Figure 8(a) shows the legalization result of the benchmark `fft_2` generated by our legalization algorithm, and Figure 8(b) shows a partial layout which justifies that the cell order is well preserved by our algorithm, a key to our superior results.

5.3 Effectiveness and Optimality of RMMSIM

In order to empirically validate the effectiveness and optimality of RMMSIM, we compared our RMMSIM-based algorithm with the CPLEX-based legalization algorithm and the MMSIM-based algorithm [8]. The only difference between the three algorithms lies in the method of solving the QP, which was formulated in Equation (17). Our legalization algorithm used the RMMSIM to solve Equation (17) by solving the equivalent converted LCP, the MMSIM-based legalization algorithm adopted the MMSIM solver presented in [8] to solve Equation (17), and the CPLEX-based one used the IBM ILOG CPLEX Optimizer (V12.7.1) to solve Equation (17) directly. The three algorithms were run on the same Linux machine with a 3.40GHz Intel Core CPU and 16 GB memory.

Table 3 shows the comparisons among our algorithm, the MMSIM-based algorithm, and the CPLEX-based one. In the table, the column “#Var.” gives the total number of variables in Equation (17), “#Cons.” the total number of constraints in Equation (17), “Average Disp. (sites)” the cell

Table 3. Comparisons among the RMMSIM-based, the MMSIM-based, and the CPLEX-based Legalization Algorithms

Benchmark			Average Disp. (sites)			Δ HPWL (%)			Eq. (17) Runtime (s)			#Iteration	
	#Var.	#Cons.	CPLEX	MMSIM	Ours	CPLEX	MMSIM	Ours	CPLEX	MMSIM	Ours	MMSIM	Ours
des_perf_1	121446	130026	1.467	1.461	1.44	0.73	0.74	0.74	9.67	1.26	0.83	63	56
des_perf_a	116801	124912	0.654	0.655	0.654	0.07	0.07	0.07	7.71	1.02	0.36	24	14
des_perf_b	121446	129948	0.623	0.623	0.623	0.08	0.09	0.08	7.09	1.31	0.83	51	46
edit_dist_a	132913	138013	0.448	0.448	0.448	0.08	0.08	0.08	7.31	1.52	1.03	52	48
fft_1	34265	36117	1.064	1.067	1.07	0.84	0.85	0.86	2	0.34	0.21	49	45
fft_2	34265	36078	0.628	0.625	0.625	0.49	0.49	0.49	1.63	0.29	0.21	47	44
fft_a	32532	34109	0.571	0.571	0.569	0.23	0.23	0.23	1.9	0.3	0.18	43	40
fft_b	32532	34275	0.66	0.658	0.658	0.23	0.23	0.23	2	0.31	0.18	46	42
matrix_mult_1	158223	160846	0.391	0.387	0.387	0.2	0.19	0.19	10.8	1.54	1.24	51	50
matrix_mult_2	158223	160844	0.365	0.364	0.363	0.17	0.16	0.16	11.75	1.56	1.25	55	47
matrix_mult_a	152463	155244	0.258	0.257	0.257	0.08	0.08	0.08	11.61	1.28	0.52	27	15
matrix_mult_b	149175	151462	0.243	0.244	0.242	0.07	0.07	0.07	10.9	1.42	0.82	39	31
matrix_mult_c	149175	151362	0.263	0.264	0.264	0.09	0.09	0.09	11.4	1.46	0.82	39	32
pci_bridge32_a	32766	35815	0.878	0.878	0.877	0.45	0.46	0.46	1.46	0.23	0.1	31	18
pci_bridge32_b	32094	35074	0.904	0.904	0.904	0.05	0.05	0.05	1.55	0.23	0.09	26	16
superblue11_a	989918	1051995	1.701	1.69	1.676	0.11	0.11	0.11	110.52	15.8	6.45	51	34
superblue12	1401310	1514267	1.344	1.341	1.342	0.14	0.14	0.14	142.39	14.34	6	29	17
superblue14	659717	705605	2.236	2.222	2.221	0.17	0.16	0.16	66.6	8.23	4.32	47	34
superblue16_a	735481	789231	1.542	1.524	1.525	0.08	0.08	0.08	68.69	8.1	6.38	62	48
superblue19	534085	560374	1.387	1.385	1.373	0.11	0.1	0.1	50.3	6.26	4.49	65	52
Normalized			1.004	1.002	1.000	1.003	1.002	1.000	13.268	1.826	1.000		

displacement measured in the number of the placement site in width, “ Δ HPWL” the HPWL increase from the corresponding global placement, “Equation (17) Runtime(s)” the running time in second for solving Equation (17), and “#Iteration” the number of iterations for the MMSIM and RMMSIM. The columns “CPLEX”, “MMSIM”, and “Ours” give the corresponding results generated by the CPLEX-based algorithm, the MMSIM-based algorithm, and our algorithm, respectively.

It can be seen from Table 3, the CPLEX-based legalization algorithm, the MMSIM based algorithm and our algorithm generated almost the same average cell displacements and HPWL increase rate for all the 20 benchmarks. Therefore, the RMMSIM optimality for mixed-cell-height circuit design is empirically validated. We ascribe the slight differences between the results of the two algorithms to the precision of computation and the operations after solving Equation (17). For runtime, our RMMSIM of solving the equivalent converted LCP is $13.27\times$ faster than the latest IBM ILOG CPLEX Optimizer of solving Equation (17) directly. Besides, the RMMSIM is $1.83\times$ faster than the MMSIM solver presented in [8] because the RMMSIM directly computes the inverse of matrix $\tilde{Q} + \tilde{\lambda}E^TE + \varepsilon I_n$ and requires fewer iterations. The results show the great effectiveness and efficiency of our RMMSIM algorithm.

5.4 Effectiveness of Individual Techniques

From Figure 7, our legalization algorithm consists of three steps: cell alignment and local cell diffusion, RMMSIM legalization, and illegal cell detection and allocation. Compared with our conference version [8], all the three steps have been improved. In this experiment, we integrated each

Table 4. Average Displacement and Runtime Impact of Local Cell Diffusion, RMMSIM, and Illegal Cell Detection & Allocation

Benchmark	Average Disp. (sites)				Runtime (s)			
	w. L.	w. R.	w. I.	[8]	w. L.	w. R.	w. I.	[8]
des_perf_1	1.466	1.529	1.496	1.537	2.62	2.08	2.52	2.34
des_perf_a	0.656	0.669	0.665	0.670	2.43	1.64	2.04	2.27
des_perf_b	0.623	0.637	0.637	0.637	2.54	1.92	2.28	2.36
edit_dist_a	0.453	0.468	0.461	0.468	2.82	2.25	2.73	2.79
fft_1	1.084	1.175	1.155	1.176	0.67	0.50	0.58	0.63
fft_2	0.625	0.646	0.645	0.645	0.66	0.48	0.62	0.63
fft_a	0.569	0.586	0.585	0.585	0.61	0.44	0.57	0.58
fft_b	0.663	0.695	0.689	0.696	0.62	0.46	0.58	0.59
matrix_mult_1	0.392	0.408	0.402	0.408	3.74	2.97	3.27	3.48
matrix_mult_2	0.367	0.381	0.378	0.382	3.73	3.06	3.32	3.49
matrix_mult_a	0.261	0.276	0.272	0.276	3.48	2.52	3.06	3.24
matrix_mult_b	0.245	0.258	0.255	0.258	3.28	2.50	2.99	3.12
matrix_mult_c	0.266	0.279	0.278	0.280	3.31	2.21	3.26	3.15
pci_bridge32_a	0.877	0.883	0.886	0.886	0.59	0.42	0.58	0.57
pci_bridge32_b	0.904	0.913	0.912	0.912	0.6	0.44	0.60	0.58
superblue11_a	1.691	1.728	1.715	1.734	27.23	15.78	23.74	25.21
superblue12	1.352	1.397	1.381	1.399	41.31	28.95	35.25	37.69
superblue14	2.289	2.420	2.336	2.425	18.25	13.18	16.06	16.99
superblue16_a	1.540	1.582	1.561	1.586	20.01	16.02	17.28	18.05
superblue19	1.395	1.439	1.415	1.441	13.97	10.76	11.89	12.71
Normalized	0.963	0.999	0.988	1.000	1.067	0.787	0.969	1.000

individual step into our conference version to verify the effectiveness of each step. Table 4 lists the experimental results. In the table, based on our conference version [8], the columns “w. L.” give the results generated by integrating the local cell diffusion into our conference version, the columns “w. R.” show the results generated by replacing the MMSIM in our conference version with the RMMSIM, and the columns “w. I.” give the results generated by replacing the Tetris-like allocation in our conference version with the illegal cell detection and allocation.

As shown in the table, based on our conference version [8], the local cell diffusion can achieve 3.7% smaller average displacement at the cost of 6.7% slower runtime. Although the local cell diffusion needs additional runtime, it gives better initial solutions for MMSIM/RMMSIM legalization, and thus the MMSIM/RMMSIM can converge for the LCP much more quickly. Besides, the local cell diffusion also gives better cell assignment and ordering and thus can improve the solution quality. Since the RMMSIM can directly compute the inverse of matrix $\bar{Q} + \bar{\lambda}E^TE + \varepsilon I_n$ and converge for the LCP more quickly, compared with our conference version with the MMSIM, the conference version with RMMSIM can achieve 21.3% faster runtime with almost the same solution quality. Further, the illegal cell detection and allocation can achieve 1.2% smaller average displacement in a comparable runtime than the Tetris-like allocation used in the conference version [8].

Figure 9 shows the runtime ratio for the three steps on five large benchmarks (i.e., superblue11_a, superblue12, superblue14, superblue16_a, and superblue19). Since the local cell diffusion is only performed in dense regions as described in Section 4.4, the runtime of this step depends mainly on the cell distribution of the given global placement results. Besides, the local cell

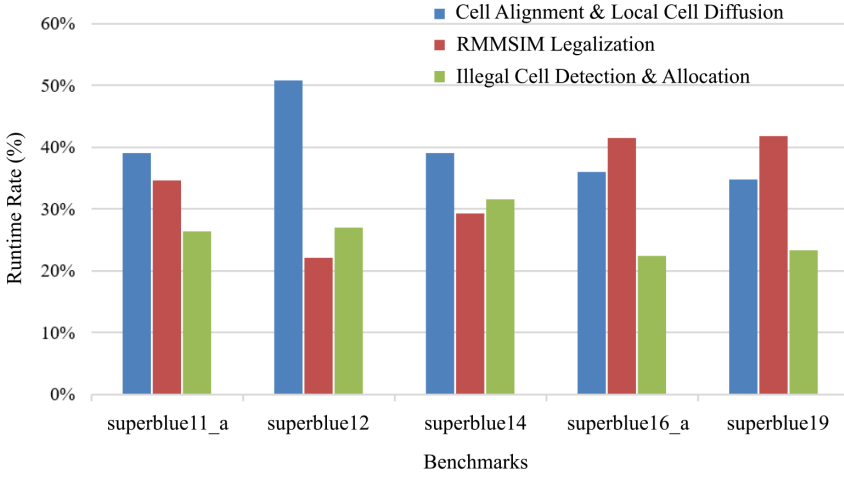


Fig. 9. The runtime ratio for the three steps (i.e., cell alignment and local cell diffusion, RMMSIM legalization, and illegal cell detection and allocation) on five large benchmarks.

diffusion provides better initial solutions for RMMSIM legalization, so that RMMSIM can converge faster.

5.5 Robustness of the Proposed Algorithm

In order to show the robustness of our proposed algorithm, we compared our algorithm with the work [17] based on the ICCAD 2017 CAD contest benchmarks [11]. For fair comparison, as in [17], we omit the fence region constraints and the soft constraints. Table 5 shows the statistics of the benchmarks and the comparison of our algorithm with the work [17]. In the table, “#S. C.,” “#D. C.,” “#T. C.,” and “#Q. C.” give the total numbers of single-, double-, triple-, and quadruple-row height cells, respectively. “Avg. Disp. (sites)” gives the average cell displacement measured in the number of the placement site width, and “CPU(s)” the runtimes in seconds. Note that the results of the work [17] are quoted from the publication directly, which were obtained on a PC with a 2.7 GHz CPU and 16 GB memory, while our algorithm was performed on a PC with a 3.40 GHz Intel Core CPU and 16 GB memory. As can be seen from the table, compared with [17], our algorithm can achieve 3.7% smaller average displacements with faster runtime. Overall, the experimental results show the robustness of our proposed algorithm.

6 CONCLUDING REMARKS

We have presented an RMMSIM to solve the mixed-cell-height standard-cell legalization problem. Since the RMMSIM effectively explores the sparse characteristic of a circuit, and takes only linear time per iteration, it can solve the legalization problem very efficiently. The RMMSIM convergence and optimality have been theoretically proved and empirically validated. Fixing the cell ordering from global placement and relaxing the right-boundary constraints, specifically, we first converts the legalization problem into an equivalent LCP, and then properly splits the matrices in the LCP so that the RMMSIM can solve the LCP optimally. Finally, an allocation scheme for illegal cells is used to align such cells to placement sites on rows and fix the placement of out-of-right-boundary cells, if any. Experimental results not only have shown the effectiveness and efficiency of our proposed algorithm, but also have validated the convergence and optimality of our RMMSIM. In particular, our formulation provides new generic solutions and research directions for various optimization problems that require solving large-scale QP problems efficiently.

Table 5. Experimental Results Compared with the work [17] Based on the ICCAD 2017 Contest Benchmarks

Benchmark	Statistics				Avg. Disp. (sites)		Runtime (s)	
	#S. Cell	#D. C.	#T. C.	#Q. C.	[17]	Ours	[17]	Ours
des_perf_1	112644	0	0	0	6.97	6.52	11.75	3.54
des_perf_a_md1	103589	4699	0	0	5.94	5.84	2.79	1.81
des_perf_a_md2	105030	1086	1086	1086	5.93	5.81	6.82	3.29
des_perf_b_md1	106782	5862	0	0	4.77	4.76	3.64	2.31
des_perf_b_md2	101908	6781	2260	1695	5.25	5.24	3.12	1.98
edit_dist_1_md1	118005	7994	2664	1998	5.79	5.62	5.19	1.83
edit_dist_a_md2	115066	7799	2599	1949	5.51	5.38	2.24	1.93
edit_dist_a_md3	119616	2599	2599	2599	7.08	7.09	15.68	5.96
fft_2_md2	28930	2117	705	529	7.54	6.92	2.89	0.4
fft_a_md2	27431	2018	672	504	4.86	4.43	0.60	0.35
fft_a_md3	28609	672	672	672	4.55	4.07	0.40	0.33
pci_bridge32_a_md1	26680	1792	597	448	5.64	5.51	2.29	0.33
pci_bridge32_a_md2	25239	2090	1194	994	7.14	7.09	3.34	0.96
pci_bridge32_b_md1	26134	1756	585	439	6.01	5.79	0.70	0.31
pci_bridge32_b_md2	28038	292	292	292	5.53	5.38	0.66	0.37
pci_bridge32_b_md3	27452	292	585	585	6.10	5.95	1.58	0.97
Normalized					1.037	1.000	2.685	1.000

(e.g., global placement [25], buffering and wire sizing [10], dummy fill insertion [22], analog circuit optimization [24], etc.), by applying our proposed linear complementarity relaxation and its fast RMMSIM.

APPENDIX

In this appendix, we analyze the range of the two parameters β^* and θ^* in the splitting matrices (9) to guarantee the convergence of our RMMSIM (Algorithm 1 in Section 3.2).

Let $I_n \in \mathbb{R}^{n \times n}$ and $I_m \in \mathbb{R}^{m \times m}$ be the two identity matrices. For the linear system (8) and the splitting matrices $M(\epsilon)$ and N in (9), we set the positive diagonal matrix Ω as

$$\Omega = \begin{bmatrix} \frac{1}{\beta^*} I_n & 0 \\ 0 & \frac{1}{\theta^*} I_m \end{bmatrix}.$$

Let $\bar{s}^k = \begin{pmatrix} \bar{s}_u^k \\ \bar{s}_d^k \end{pmatrix}$. Then the fixed-point equation (8) can be rewritten as

$$(M(\epsilon) + \Omega) \begin{pmatrix} \bar{s}_u^{k+1} \\ \bar{s}_d^{k+1} \end{pmatrix} = N \begin{pmatrix} \bar{s}_u^k \\ \bar{s}_d^k \end{pmatrix} + (\Omega - A(\epsilon)) \mid \begin{pmatrix} \bar{s}_u^k \\ \bar{s}_d^k \end{pmatrix} \mid -\gamma q.$$

Since Matrices $A(\epsilon)$ and $M(\epsilon)$ are positive definite, so the analysis is based on the validity of $LCP(q, A(\epsilon))$.

Although the initial point \bar{s}^0 and the solution \bar{s}^* are positive, some components of the internal iterative solution \bar{s}^k may not be positive. So we let $\bar{s}^k = \bar{s}_1^k + \bar{s}_2^k$, where \bar{s}_1^k, \bar{s}_2^k are defined as follows:

$$\bar{s}_1^k = \frac{1}{2}(\bar{s}^k + |\bar{s}^k|), \bar{s}_2^k = \frac{1}{2}(\bar{s}^k - |\bar{s}^k|).$$

The above notations mean that all positive components of \bar{s}^k are contained in \bar{s}_1^k , all negative ones are contained in \bar{s}_2^k , and the rest components of \bar{s}_i^k , $i = 1, 2$, are zero. By these notations, we can ignore the absolute values in the iterative equation:

$$\begin{aligned} (M(\varepsilon) + \Omega)\bar{s}^{k+1} &= N(\bar{s}_1^k + \bar{s}_2^k) + (\Omega - A(\varepsilon)) | \bar{s}_1^k + \bar{s}_2^k | - \gamma q \\ &= N(\bar{s}_1^k + \bar{s}_2^k) + (\Omega - A(\varepsilon))\bar{s}_1^k + (A(\varepsilon) - \Omega)\bar{s}_2^k - \gamma q \\ &= (\Omega - 2M(\varepsilon) + N)\bar{s}_1^k + (M(\varepsilon) - \Omega)\bar{s}_2^k - \gamma q. \end{aligned}$$

Then

$$\bar{s}^{k+1} = (M(\varepsilon) + \Omega)^{-1}(\Omega - 2M(\varepsilon) + N)\bar{s}_1^k + (M(\varepsilon) + \Omega)^{-1}(M(\varepsilon) - \Omega)\bar{s}_2^k - (M(\varepsilon) + \Omega)^{-1}\gamma q. \quad (21)$$

Let $\bar{s}^{k+1} = \bar{s}_1 + \bar{s}_2$, $(M(\varepsilon) + \Omega)^{-1}\gamma q = \Delta_1 + \Delta_2$, and by Equation (21), we let

$$\begin{cases} \bar{s}_1 = (M(\varepsilon) + \Omega)^{-1}(\Omega - 2M(\varepsilon) + N)\bar{s}_1^k - \Delta_1, \\ \bar{s}_2 = (M(\varepsilon) + \Omega)^{-1}(M(\varepsilon) - \Omega)\bar{s}_2^k - \Delta_2. \end{cases} \quad (22)$$

For convenience, let

$$\begin{aligned} \mathcal{H}(\beta^*, \theta^*) &= (M(\varepsilon) + \Omega)^{-1}(\Omega - 2M(\varepsilon) + N), \\ \mathcal{M}(\beta^*, \theta^*) &= M(\varepsilon) + \Omega, \\ \mathcal{J}(\beta^*, \theta^*) &= (M(\varepsilon) + \Omega)^{-1}(M(\varepsilon) - \Omega). \end{aligned}$$

Then

$$\bar{s}^{k+1} = \mathcal{H}\bar{s}_1^k + \mathcal{J}\bar{s}_2^k - \mathcal{M}\gamma q. \quad (23)$$

Since \bar{s}^* is the solution of the fixed-point equation (21), we have $\bar{s}^* = \mathcal{H}\bar{s}_1^* + \mathcal{J}\bar{s}_2^* - \mathcal{M}\gamma q$, and

$$\|\bar{s}^{k+1} - \bar{s}^*\| = \left\| \begin{bmatrix} \mathcal{H} & 0 \\ 0 & \mathcal{J} \end{bmatrix} \begin{pmatrix} \bar{s}_1^k \\ \bar{s}_2^k \end{pmatrix} - \begin{bmatrix} \mathcal{H} & 0 \\ 0 & \mathcal{J} \end{bmatrix} \begin{pmatrix} \bar{s}_1^* \\ \bar{s}_2^* \end{pmatrix} \right\| \leq \left\| \begin{bmatrix} \mathcal{H} & 0 \\ 0 & \mathcal{J} \end{bmatrix} \right\| \|\bar{s}^k - \bar{s}^*\|. \quad (24)$$

Thus, if we can prove that $|\rho(\mathcal{H}(\beta^*, \theta^*))| < 1$ and $|\rho(\mathcal{J}(\beta^*, \theta^*))| < 1$, then the RMMSIM is convergent. We divide the proof into the following two parts.

Part 1: We focus on the first equation of (22). Referring to [2], let $\mathcal{N}(\beta^*, \theta^*) = \mathcal{M}(\beta^*, \theta^*) - 2A(\varepsilon) = \Omega - M(\varepsilon) + 2N$, the following equation holds:

$$\mathcal{H}(\beta^*, \theta^*) = \mathcal{M}(\beta^*, \theta^*)^{-1}\mathcal{N}(\beta^*, \theta^*), \quad (25)$$

where

$$\begin{aligned} \mathcal{M}(\beta^*, \theta^*) &= \Omega + M(\varepsilon) = \begin{bmatrix} \frac{1}{\beta^*}(Q + I_n) + \varepsilon I_n & 0 \\ B & \frac{1}{\theta^*}(D + I_m) + \varepsilon I_m \end{bmatrix}, \\ \mathcal{N}(\beta^*, \theta^*) &= \Omega - M(\varepsilon) + 2N = \begin{bmatrix} \frac{1}{\beta^*}(Q + I_n) - 2Q - \varepsilon I_n & 2B^T \\ -B & \frac{1}{\theta^*}(D + I_m) - \varepsilon I_m \end{bmatrix}. \end{aligned} \quad (26)$$

To ensure that $\rho(\mathcal{H}(\beta^*, \theta^*)) < 1$, we consider the eigenpairs of Matrix $\mathcal{H}(\beta^*, \theta^*)$ similar to Theorem 4.2 of [2].

THEOREM A.1. *In our work, $(Q + \varepsilon I_n), (Q + I_n) \in \mathbb{R}^{n \times n}$, $(D + I_m) \in \mathbb{R}^{m \times m}$ are all symmetric positive definite, and $B \in \mathbb{R}^{m \times n}$ is of full row rank, with $m \leq n$. In this section, assume that λ is an eigenvalue of the iteration matrix $\mathcal{H}(\beta^*, \theta^*)$, and $\bar{s}_u, \bar{s}_d \in \mathbb{C}^m$ are two complex vectors. Let*

$$\omega = \omega(\bar{s}_u) = \frac{\bar{s}_u^* I_n \bar{s}_u}{\bar{s}_u^* (Q + \varepsilon I_n) \bar{s}_u}, \quad \mu = \mu(\bar{s}_u) = \frac{\bar{s}_u^* B(I_m + D)^{-1} B^T \bar{s}_u}{\bar{s}_u^* (Q + \varepsilon I_n) \bar{s}_u}.$$

Then λ satisfies the quadratic equation

$$(1 + \omega)\lambda^2 + 2(-1 - \omega + \beta^* + \beta^* \mu \theta^*)\lambda + 1 + \omega - 2\beta^* + 2\beta^* \mu \theta^* = 0. \quad (27)$$

PROOF 8. With the assumption of the theorem and (25), and by the first equation of (22), we have

$$\mathcal{N}(\beta^*, \theta^*) \begin{pmatrix} \bar{s}_u \\ \bar{s}_d \end{pmatrix} = \lambda \mathcal{M}(\beta^*, \theta^*) \begin{pmatrix} \bar{s}_u \\ \bar{s}_d \end{pmatrix}.$$

Then we can expand the above equation based on (26). Note that ε is very small, so we ignore the parts with ε to simplify the proof in the following discussion. We have

$$\begin{cases} \frac{\lambda}{\beta^*}(Q + I_n)\bar{s}_u = (\frac{1}{\beta^*}(Q + I_n) - 2Q)\bar{s}_u + 2B^T\bar{s}_d, \\ \lambda(B\bar{s}_u + \frac{1}{\theta^*}(D + I_m)\bar{s}_d) = -B\bar{s}_u + \frac{1}{\theta^*}(D + I_m)\bar{s}_d, \end{cases}$$

or equivalently,

$$\begin{cases} \frac{\lambda - 1}{\beta^*}(Q + I_n)\bar{s}_u + 2Q\bar{s}_u = 2B^T\bar{s}_d, \\ (1 + \lambda)B\bar{s}_u = \frac{1 - \lambda}{\theta^*}(I_m + D)\bar{s}_d. \end{cases} \quad (28)$$

To solve the two equations in (28), we first see that $\lambda \neq 1$ and $\bar{s}_u \neq 0$. Then, from the second equality, we get

$$\bar{s}_d = \frac{1 + \lambda}{1 - \lambda} \theta^* B(I_m + D)^{-1} \bar{s}_u.$$

By substituting this relationship into the first equality, we get

$$\frac{\lambda - 1}{\beta^*}(Q + I_n)\bar{s}_u + 2Q\bar{s}_u = 2B^T \frac{1 + \lambda}{1 - \lambda} \theta^* (I_m + D)^{-1} B\bar{s}_u,$$

which can be written as

$$(\lambda - 1)^2(Q + I_n)\bar{s}_u + 2(\lambda - 1)\beta^*Q\bar{s}_u = -2\beta^*B^T(1 + \lambda)\theta^*(I_m + D)^{-1}B\bar{s}_u.$$

Note that $\bar{s}_u \neq 0$ and $\bar{s}_u^*(Q + \varepsilon I_n)\bar{s}_u \neq 0$. After multiplying \bar{s}_u^* and then dividing $\bar{s}_u^*(Q + \varepsilon I_n)\bar{s}_u$ on both sides of the above equality, the theorem thus follows because ε is very small, and thus we take $\frac{\bar{s}_u^* I_n \bar{s}_u}{\bar{s}_u^*(Q + \varepsilon I_n)\bar{s}_u} = 1$. \square

Let ω_{min} and ω_{max} , and μ_{min} and μ_{max} be the smallest and the largest eigenvalues of the matrix $(Q + \varepsilon I_n)^{-1}$ and Matrix $(I_m + D)^{-1}B^T P^{-1}B$, respectively. Based on Theorem A.1, we can get the condition for the convergence of the RMMSIM method.

THEOREM A.2. *In our work, $(Q + \varepsilon I_n), (Q + I_n) \in \mathbb{R}^{n \times n}, (D + I_m) \in \mathbb{R}^{m \times m}$ are symmetric positive definite, and $B \in \mathbb{R}^{m \times n}$ is of full row rank. The RMMSIM method is convergent, provided that β^* satisfies $0 < \beta^* < 1 + \omega_{min}$ and θ^* satisfies $\theta^* > \frac{\mu_{max}\beta^*}{2(2 - \omega_{max}\beta^*)}$.*

PROOF 9. From (27), if $\mu = 0$, then $\lambda = \frac{2}{1 + \omega}\beta^* - 1$. It follows from the result in [3] that when $\lambda = \frac{2}{1 + \omega}\beta^* - 1$ and both roots λ satisfy $|\lambda| < 1$ if and only if

$$\begin{aligned} & \left| \frac{2}{1 + \omega}\beta^* - 1 \right| < 1, \\ & |2(-1 - \omega + \beta^* + \beta^*\mu\theta^*)| < 2(1 + \omega - \beta^* + \beta^*\mu\theta^*). \end{aligned}$$

By solving the inequality above, we get

$$0 < \frac{1}{1 + \omega}\beta^* < 1 \quad \text{and} \quad \beta^*\mu\theta^* > 0.$$

Note that $\theta^* > 0$ does not meet our requirement since it is a large interval. So we consider Theorem 2.1 in [3] and restrict the range by computing $0 < \frac{1}{\theta^*} < \frac{2(2 - \omega_{max}\beta^*)}{\mu_{max}\beta^*}$. Then, this theorem thus follows. \square

Part 2: Using the similar method of Part 1, we discuss the second equation of (22). Let

$$O(\beta^*, \theta^*) = \mathcal{M}(\varepsilon)(\beta^*, \theta^*) - 2\Omega = M(\varepsilon) - \Omega.$$

Then

$$O(\beta^*, \theta^*) = -\Omega + M(\varepsilon) = \begin{bmatrix} \frac{1}{\beta^*}(Q - I_n) + \varepsilon I_n & 0 \\ B & \frac{1}{\theta^*}(D - I_m) + \varepsilon I_m \end{bmatrix}.$$

We have $\mathcal{J}(\beta^*, \theta^*) = \mathcal{M}(\beta^*, \theta^*)^{-1}O(\beta^*, \theta^*)$. Then, using the same assumption of Theorem A.1, we get

$$O(\beta^*, \theta^*) \begin{pmatrix} \bar{s}_u \\ \bar{s}_d \end{pmatrix} = \lambda \mathcal{M}(\beta^*, \theta^*) \begin{pmatrix} \bar{s}_u \\ \bar{s}_d \end{pmatrix}.$$

Rearranging the equations leads to

$$\begin{cases} \frac{\lambda}{\beta^*}(Q + I_n)\bar{s}_u = \frac{1}{\beta^*}(Q - I_n)\bar{s}_u, \\ \lambda B\bar{s}_u + \frac{\lambda}{\theta^*}(D + I_m)\bar{s}_d = B\bar{s}_u + \frac{1}{\theta^*}(D - I_m)\bar{s}_d. \end{cases} \quad (29)$$

From the second equation of (29), we can represent \bar{s}_u by \bar{s}_d and derive the relationship that

$$\bar{s}_u = \frac{(1 - \lambda)D\bar{s}_d - (1 + \lambda)I_m\bar{s}_d}{B(\lambda - 1)\theta^*}.$$

Substituting the above expression into the first equality in (29), we get

$$\frac{\lambda}{\beta^*}(Q + I_n) \frac{(1 - \lambda)D\bar{s}_d - (1 + \lambda)I_m\bar{s}_d}{B(\lambda - 1)\theta^*} = \frac{1}{\beta^*}(Q - I_n) \frac{(1 - \lambda)D\bar{s}_d - (1 + \lambda)I_m\bar{s}_d}{B(\lambda - 1)\theta^*}.$$

Note that β^* and θ^* on both sides of the equation can be eliminated, so λ has nothing to do with the parameters β^* and θ^* .

By solving the system of Equations (29), we derive that

$$|\lambda| = \left| \frac{Q - I_n}{Q + I_n} \right| < 1.$$

Hence, for $\mathcal{J}(\beta^*, \theta^*)$, $|\lambda| < 1$ holds for all β^* and θ^* .

Based on the above discussions in the two parts, we have that the RMMSIM is convergent as long as the conditions of Theorem A.1 are satisfied.

REFERENCES

- [1] Sang-Hoon Baek, Ha-Young Kim, Young-Keun Lee, Duck-Yang Jin, Se-Chang Park, and Jun-Dong Cho. 2008. Ultra-high density standard cell library using multi-height cell structure. In *Proceedings of SPIE 7268*. 72680C–72680C.
- [2] Zhong-Zhi Bai. 2010. Modulus-based matrix splitting iteration methods for linear complementarity problems. *Numerical Linear Algebra with Applications* 17, 6 (2010), 917–933.
- [3] Zhong-Zhi Bai, Beresford N. Parlett, and Zeng-Qi Wang. 2005. On generalized successive overrelaxation methods for augmented linear systems. *Numer. Math.* 102, 1 (2005), 1–38.
- [4] Stephen Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press, Cambridge.
- [5] Ismail S. Bustany, David Chinnery, Joseph R. Shinnerl, and Vladimir Yutsis. 2015. ISPD 2015 benchmarks with fence regions and routing blockages for detailed-routing-driven placement. In *Proceedings of the ACM International Symposium on Physical Design*. 157–164.
- [6] Jianli Chen, Peng Yang, Xingquan Li, Wenxing Zhu, and Yao-Wen Chang. 2018. Mixed-cell-height placement with complex minimum-implant-area constraints. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*.
- [7] Jianli Chen, Wenxing Zhu, and Zheng Peng. 2012. A heuristic algorithm for the strip packing problem. *Journal of Heuristics* 18, 4 (2012), 677–697.
- [8] Jianli Chen, Ziran Zhu, Wenxing Zhu, and Yao-Wen Chang. 2017. Toward optimal legalization for mixed-cell-height circuit designs. In *Proceedings of the ACM/IEEE Design Automation Conference*.

- [9] Wing-Kai Chow, Chak-Wa Pui, and Evangeline F. Y. Young. 2016. Legalization algorithm for multiple-row height standard cell design. In *Proceedings of the ACM/IEEE Design Automation Conference*.
- [10] Chris C. N. Chu and D. F. Wong. 1999. A quadratic programming approach to simultaneous buffer insertion/sizing and wire sizing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 18, 6 (June 1999), 787–798.
- [11] Nima Karimpour Darav, Ismail S. Bustany, Andrew Kennings, and Ravi Mamidi. 2017. ICCAD-2017 CAD contest in multi-deck standard cell legalization and benchmarks. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*.
- [12] E. Michael Gertz and Stephen J. Wright. 2003. Object-oriented software for quadratic programming. *ACM Trans. Math. Softw.* 29, 1 (2003), 58–81.
- [13] D. Hill. 2002. Method and system for high speed detailed placement of cells within integrated circuit designs. In *U.S. Patent 6370673*.
- [14] Chung-Yao Hung, Peng-Yi Chou, and Wai-Kei Ma. 2017. Mixed-cell-height standard cell placement legalization. In *Proceedings of the on Great Lakes Symposium on VLSI 2017*. ACM, 149–154.
- [15] Zoltán Király and Péter Kovács. 2012. Efficient implementations of minimum-cost flow algorithms. *Arxiv Preprint Arxiv:1207.6381* (2012).
- [16] Haocheng Li, Wing-Kai Chow, Gengjie Chen, Evangeline F. Y. Young, and Bei Yu. 2018. Routability-driven and fence-aware legalization for mixed-cell-height circuits. In *Proceedings of the ACM/IEEE Design Automation Conference*.
- [17] Xingquan Li, Jianli Chen, Wenxing Zhu, and Yao-Wen Chang. 2019. Analytical mixed-cell-height legalization considering average and maximum movement minimization. In *Proceedings of the ACM International Symposium on Physical Design*. 27–34.
- [18] Yibo Lin, Bei Yu, Xiaoping Xu, Jih-Rong Gao, Natarajan Viswanathan, Wen-Hao Liu, Zhuo Li, Charles J. Alpert, and David Z. Pan. 2016. MrDP: Multiple-row detailed placement of heterogeneous-sized cells for advanced nodes. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*. 7:1–7:8.
- [19] Christopher M. Maes. 2010. *A Regularized Active-set Method for Sparse Convex Quadratic Programming*. Ph.D. Dissertation. Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA.
- [20] Jorge Nocedal and Stephen J. Wright. 2006. *Numerical Optimization*. Springer, New York.
- [21] Peter Spindler, Ulf Schlichtmann, and Frank M. Johannes. 2008. Abacus: Fast legalization of standard cell circuits with minimal movement. In *Proceedings of the ACM International Symposium on Physical Design*. 47–53.
- [22] Yudong Tao, Changhao Yan, Yibo Lin, Shengguo Wang, David Z. Pan, and Xuan Zeng. 2016. A novel unified dummy fill insertion framework with sqp-based optimization method. In *Proceeding of the IEEE/ACM International Conference on Computer-Aided Design*. 88:1–88:8.
- [23] Yu-Wei Tseng and Yao-Wen Chang. 2018. Mixed-cell-height placement considering drain-to-drain abutment. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*.
- [24] Sergey Vichik, Murat Arcak, and Francesco Borrelli. 2016. Stability of an analog optimization circuit for quadratic programming. *Systems & Control Letters* 88 (2016), 68–74.
- [25] Natarajan Viswanathan, Gi-Joon Nam, Charles J. Alpert, Paul Villarrubia, Haoxing Ren, and Chris C. N. Chu. 2007. RQL: Global placement via relaxed quadratic spreading and linearization. In *Proceedings of the ACM/IEEE Design Automation Conference*. 453–458.
- [26] Chao-Hung Wang, Yen-Yi Wu, Jianli Chen, Yao-Wen Chang, Sy-Yen Kuo, Wenxing Zhu, and Genghua Fan. 2017. An effective legalization algorithm for mixed-cell-height standard cells. In *Proceedings of the IEEE/ACM Asia and South Pacific Design Automation Conference*. 450–455.
- [27] Jun Wang, Alfred K. Wong, and Edmund Y. Lam. 2004. Standard cell layout with regular contact placement. *IEEE Transactions on Semiconductor Manufacturing* 17, 3 (August 2004), 375–383.
- [28] Richard W. Cottle, Jong-Shi Pang, and Richard E. Stone. 1992. *The Linear Complementarity Problem*. Academic Press.
- [29] Gang Wu and Chris C. N. Chu. 2016. Detailed placement algorithm for VLSI design with double-row height standard cells. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 35, 9 (2016), 1569–1573.
- [30] Yen-Yi Wu and Yao-Wen Chang. 2017. Mixed-cell-height detailed placement considering complex minimum-implant-area constraints. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*.
- [31] Ziran Zhu, Xingquan Li, Yuhang Chen, Jianli Chen, Wenxing Zhu, and Yao-Wen Chang. 2018. Mixed-cell-height legalization considering technology and region constraints. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*.

Received February 2020; revised September 2020; accepted September 2020