Toward Optimal Legalization for Mixed-Cell-Height Circuit Designs*

Jianli Chen[1], Ziran Zhu[1], Wenxing Zhu[1], and Yao-Wen Chang[2,3]
[1]Center for Discrete Mathematics and Theoretical Computer Science, Fuzhou University, Fuzhou 350108, China
[2]Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 10617, Taiwan
[3]Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan
jlchen@fzu.edu.cn; N150320066@fzu.edu.cn; wxzhu@fzu.edu.cn; ywchang@ntu.edu.tw

ABSTRACT

Modern circuits often contain standard cells of different row heights to meet various design requirements. Higher cells give larger drive strengths at the costs of larger areas and power. Multi-row-height standard cells incur challenging issues to layout designs, especially the mixed-cell-height legalization problem due to the heterogeneous cell structures. Honoring the good cell positions from global placement, we present in this paper a fast and near-optimal algorithm to solve the legalization problem. Fixing the cell ordering from global placement and relaxing the right boundary constraints, we first convert the problem into a linear complementarity problem (LCP). With the converted LCP, we split its matrices to meet the convergence requirement of a modulus-based matrix splitting iteration method (MMSIM), and then apply the MMSIM to solve the LCP. This MMSIM method guarantees the optimality if no cells are placed beyond the right boundary of a chip. Finally, a Tetris-like allocation approach is used to align cells to placement sites on rows and fix the placement of out-of-right-boundary cells, if any. Experimental results show that our proposed algorithm can achieve the best cell displacement and wirelength among all published methods in reasonable runtimes. The MMSIM optimality is theoretically proven and empirically validated. In particular, our formulation provides new generic solutions and research directions for various optimization problems that require solving large-scale quadratic programs efficiently.

CCS CONCEPTS

• **Hardware → Electronic design automation**;

KEYWORDS

Physical Design, Placement, Legalization, Multi-row-height cell, Quadratic programming

1 INTRODUCTION

In traditional circuits, standard cells have the same height for easy design and optimization [19]. With the increasing complexity in modern circuit designs, however, standard cells often have different cell heights based on the area, power, and speed characteristics. For example, higher cells give larger drive strengths and better pin accessibility and routability at the costs of larger areas and power, while shorter cells have smaller areas and power with weaker drive strengths and lower pin accessibility and routability. As a result, mixed-cell-height standard cells become popular to address various design requirements, where simple standard cells are designed as the single-row-height structure, while complex ones the multi-row-height structure [4].

A modern placement flow typically consists of three major stages: (1) global placement: finds the desired position for each cell, ignoring the non-overlapping constraint of cells; (2) legalization: aligns cells into rows and remove cell overlaps to minimize cell displacement (or wirelength increase), and (3) detailed placement: refines the placement solution. Multi-row-height standard cells incur challenging issues to placement, especially the mixed-cell-height legalization problem, due to the heterogenous cell structures (and thus more global cell interferences and larger solution spaces) and additional power-rail constraints, as pointed out in [18].

The general mixed-cell-height legalization problem is NP-hard because it is a strip packing problem in nature [6]. Tetris [11] and Abacus [14] 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. What is worse, shifting a cell may make cells illegal in a layout. 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) or 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-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 [18]). 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 rail directly or by flipping the cells vertically, while the even-row-height cell *B* must match a power rail of the same type because 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; further, such power-rail mismatch cannot be resolved by cell flipping.

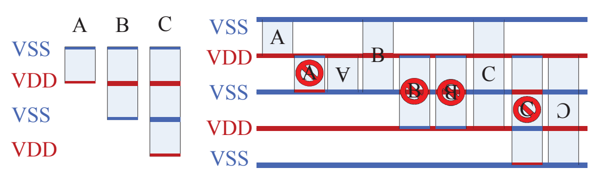


Figure 1: Mixed-cell-height standard-cell legalization considering power-rail alignment.

For the mixed-cell-height legalization problem, Wu and Chu in [21] first handled designs with single- and double-row-height standard cells, without considering the crucial power-rail alignment issue. Two recent mixed-cell-height standard-cell legalization works [7, 18] considered power-rail alignment. In [7], 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 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. Because the selection of the region and legalization tend to be local, the solution quality may be limited. The recent work [12] first applied the legalizer in [7] 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.

*This work was partially supported by the National Natural Science Foundation of China under Grants 61672005, 11501115 and 11331003, and by AnaGlobe, MediaTek, TSMC, MOST of Taiwan under Grant No's MOST 104-2221-E-002-132-MY3, MOST 105-2221-E-002-190-MY3, MOST 106-2911-I-002-511, and NTU under Grant No. NTU-ERP-104R8951.

DAC '17, June 18-22, 2017, Austin, TX, USA
 ISBN 978-1-4503-4927-7/17/06…$15.00
DOI: http://dx.doi.org/10.1145/3061639.3062330

The work [18] 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.

Honoring the good position for each cell and the cell ordering from global placement, we present in this paper a fast and near-optimal algorithm to solve the mixed-cell-height standard-cell legalization problem. The major contributions of our work are summarized below:

- Honoring the cell ordering from global placement and ignoring (relaxing) the right boundary constraints, we convert the mixed-cell-height standard-cell legalization problem to a linear complementarity problem (LCP), which can be solved by existing optimization methods effectively and efficiently.
- We split the matrices in the converted LCP in a proper manner, and use a modulus-based matrix splitting iteration method (MMSIM) to solve the converted LCP. The splitting process not only meets the convergence requirement of the MMSIM, but also significantly speeds up the computation. In particular, the MMSIM effectively explores the sparse characteristic of a circuit, and thus can solve the LCP very efficiently.
- Unlike the existing methods that legalize cells one by one, the MMSIM handles all mixed-row-height cells simultaneously, which provides a more global view of the legalization problem.
- If no cells are placed beyond the right boundary of a chip, our MMSIM method guarantees the solution optimality, and thus we only need to align cells to placement sites on rows. We use a Tetris-like allocation approach to place cells on rows and fix the placement of out-of-right-boundary cells, if any. Overall, our proposed legalization algorithm can achieve an optimal or near-optimal solution for the legalization problem. In particular, the MMSIM optimality is theoretically proven and empirically validated.
- Experimental results show that our proposed algorithm can achieve the best cell displacement and wirelength among all published methods in a reasonable runtime.
- In particular, our formulation provides new generic solutions and research directions for various optimization problems that require solving large-scale quadratic programs efficiently (e.g., global placement [17], buffering and wire sizing [8], dummy fill insertion [15], analog circuit optimization [16], etc.).

The remainder of this paper is organized as follows. Section 2 formulates the legalization problem and introduces the linear complementarity problem. Section 3 details the core techniques for reformulating the legalization problem as a linear complementarity one. Section 4 presents our algorithm. Section 5 shows the experimental results. 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.

2.1 Problem Statement

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 its (bottom-left corner) coordinate (x'_i, y'_i), $\forall i$, $1 \le i \le n$, and each even-row-height cell has a boundary power-rail type VDD or VSS. The goal of mixed-cell-height legalization 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 placed inside the chip region;
(2) Cells must be located at placement sites on rows;
(3) Cells must be non-overlapping;
(4) Cells must be aligned with correct power rails (and thus rows).

Similar to Abacus [14], the objective function is the quadratic cell displacement. Then, the mixed-cell-height legalization problem can be formulated as

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

Constraint (4) 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.

If the cell ordering is unknown, Problem (1) is a strip packing problem, which is NP-hard [6]; otherwise, with proper preprocessing and relaxation, we can relax Problem (1) as a convex quadratic programming (QP) problem which can be solved by a QP solver [10]. Because it is generally time-consuming to solve a large-scale quadratic programming problem with inequality constraints, we convert the quadratic problem equivalently into a linear complementarity problem, and solve the problem by the MMSIM [1]. The MMSIM explores the sparse characteristic of a circuit and is thus very fast. Consequently, the original quadratic programming problem can be solved much more efficiently.

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 a linear complementarity problem [1], LCP(q, A) for short, is to find a pair of real vectors w and $z \in \mathbb{R}^n$ such that

$$
w = Az + q \ge 0, \quad z \ge 0 \text{ and } z^T w = 0. \tag{2}
$$

In this problem, the notation $\ge$ 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 [1]. Among these methods, the modulus-based iteration method is considered the most effective and efficient.

The modulus-based iteration method for LCP(q, A) [1] 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 $k = 0, 1, 2, \dots$, until the iteration sequence $\{z^{(k)}\}_{k=0}^{+\infty} \subset \mathbb{R}^n$ is convergent, we compute $s^{(k+1)} \in \mathbb{R}^n$ by solving the linear system:

$$
(M + \Omega)s^{(k+1)} = Ns^{(k)} + (\Omega - A)|s^{(k)}| - \gamma q \tag{3}
$$

and set

$$
z^{(k+1)} = \frac{1}{\gamma}(|s^{(k+1)}| + s^{(k+1)}), \tag{4}
$$

where Ω is an $n \times n$ positive diagonal matrix and γ is a positive constant. The work [1] has proven the convergence of the modulus-based matrix splitting iteration method (MMSIM) when the system matrix is under some conditions, 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 OUR METHOD

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. To achieve a high-quality legalization solution, it is very important to preserve the original cell ordering from global placement. Relaxing the chip right-boundary constraints temporarily, we assume that each cell is aligned to the nearest correct row, and all cells in a row are sorted by their global x-positions, i.e., $x'_j \ge x'_l$ if cell j is on the right of cell l in the global placement. For an odd-row-height standard cell, the nearest correct row is the nearest row from its global y-position; for an even-row-height standard cell, the nearest correct row is the nearest row which matches the power rail from its global y-position.

If all cells are assigned to nearest correct rows, then the total displacement in the y-direction is minimal. Consequently, the mixed-cell-height legalization problem in (1) can be relaxed as:

$$\begin{array}{ll} \min & \frac{1}{2}\sum_{i=1}^{n}(x_i - x_i')^2 \\ \text{s.t.} & x_j - x_l \geq w_l, \qquad \text{if } x_j' \geq x_l', \\ & \text{for all adjacent cells } j \text{ and } l \text{ in the same row;} \\ & x \geq 0. \end{array} \tag{5}$$

In the above formulation, a multi-row-height standard cell may be considered several times in the constraints since it occupies several rows. In contrast, a single-row-height standard cell only occupies a row and will be considered at most twice in the constraints. Since the single-row-height standard-cell legalization problem is much easier to be reformulated, we shall convert this problem to a linear complementarity one, split the matrices in the LCP to meet the convergence requirement of the modulus-based matrix splitting iteration method (MMSIM), and use the method to solve the problem. Then, we shall extend this method to solve the mixed-cell-height standard-cell legalization problem.

3.1 Single-Row-Height Standard Cells

If all standard cells are of single-row height, we can rewrite the legalization problem (5) as the convex quadratic programming problem by relaxing the right boundary constraints as follows:

$$\begin{array}{ll} \min & \frac{1}{2}x^T Qx + p^T x \\ \text{s.t.} & Bx \geq b, \\ & x \geq 0, \end{array} \tag{6}$$

where Q is an identity matrix, p is a vector with the i-th component $p_i = -x_i'$, 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.

Proposition 1. *In Problem (6), Q is a symmetric positive definite matrix, and B is of full row rank.*

Proof: Since Q is an identity matrix, it is a symmetric positive definite matrix. Suppose 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 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 B in Problem (6) is of full row rank. □

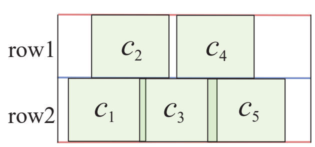


Figure 2: A placement with single-row-height cells.

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

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

$p = [-x_1', -x_2', -x_3', -x_4', -x_5']^T$, $b = [w_2, w_1, w_3]^T$.

In Problem (6), Q is a symmetric positive definite matrix. Then by the Karush-Kuhn-Tucker (KKT) conditions [3], if x is the global optimal solution of Problem (6), there exist vectors r and u such that the triple (x, r, u) satisfies the following KKT conditions:

$$\begin{cases} Qx + p - B^T r - u = 0, \\ v = Bx - b, \\ r^T v = 0, \\ u^T x = 0, \\ x, r, u, v \geq 0. \end{cases} \tag{7}$$

Equation (7) can be rewritten as the linear complementarity problem below:

$$w = Az + q \geq 0, \quad z \geq 0 \text{ and } z^T w = 0, \tag{8}$$

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

Theorem 1. *The optimal solution of the convex quadratic programming problem (6) gives the solution of LCP (8), and vice versa.*

Proof: According to the first-order optimality conditions [3], the optimal solution of Problem (6) satisfies the KKT conditions (7), which can be reformulated as LCP (8).

Conversely, suppose that $(w^*, z^*) = \left(\begin{bmatrix} u^* \\ v^* \end{bmatrix}, \begin{bmatrix} x^* \\ r^* \end{bmatrix}\right)$ is the solution of Problem (8). Then it satisfies

$$w^* = Az^* + q \geq 0, \quad z^* \geq 0 \text{ and } {z^*}^T w^* = 0, \tag{9}$$

which is

$$\begin{cases} u^* = Qx^* + p - B^T r^*, \\ v^* = Bx^* - b, \\ {r^*}^T v^* + {u^*}^T x^* = 0, \\ x^*, r^*, u^*, v^* \geq 0. \end{cases} \tag{10}$$

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 the third equation of (10) implies that

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

Considering the above equations and (10), we have that x^*, r^*, u^*, and v^* satisfy (7). According to Theorem 16.4 of [13], x^* is the optimal solution of the convex quadratic programming problem (6). □

According to Theorem 1, the quadratic programming problem (6) is equivalent to the LCP(q, A) in Problem (8). We use the MMSIM in [1] to solve the LCP(q, A). By [1], since matrix Q is a symmetric positive definite matrix and B is a rectangular matrix of full row rank, the convergence of the MMSIM for the LCP can be assured.

For the single-row-height standard-cell legalization problem, we choose the splitting matrices M, N in Equation (3) as

$$M = \begin{bmatrix} \frac{1}{\beta^*}Q & 0 \\ B & \frac{1}{\theta^*}D \end{bmatrix}, \quad N = \begin{bmatrix} (\frac{1}{\beta^*} - 1)Q & B^T \\ 0 & \frac{1}{\theta^*}D \end{bmatrix} \tag{11}$$

where $D = tridiag(BQ^{-1}B^T)$ is a tridiagonal approximation to the Schur complement $BQ^{-1}B^T$ of the matrix A, and β^* and θ^* are two positive constants determined by the formulas given in [2].

Based on the MMSIM for LCP(q, B) in [1], we can develop an algorithm for the single-row-height standard-cell legalization problem by solving the linear system shown in Equations (3) and (4).

3.2 Mixed-Cell-Height Standard Cells

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 MMSIM convergence, the matrices in Problem (8) require Q to be a symmetric positive definite matrix and B a rectangular matrix of full row rank. If there are multi-row-height cells, however, the constraint matrix B may not be of full row rank, and thus the MMSIM convergence cannot be guaranteed.

For example, Figure 3 shows a placement with mixed-cell-height standard cells. By the definition in Problem (6), the constraint matrix $B = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ -1 & 0 & 1 \end{bmatrix}$ is not of full row rank.

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

$$\begin{array}{ll} \min & \frac{1}{2}x^T Qx + p^T x \\ \text{s.t.} & Bx \geq b, \\ & Ex = 0, \\ & x \geq 0, \end{array} \tag{12}$$

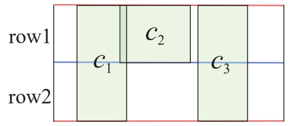


Figure 3: A placement with mixed-cell-height cells.

where Q and p are the same as those in Problem 6. Matrix E is given in the following. If a cell i is of single-row height, we introduce the variable x_{i1} for this cell; otherwise, we divide the multi-row-height cell into multiple single-row-height subcells, denoted by the variables $x_{i1}, x_{i2}, ..., x_{id}$, where d is the number of subcells. 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 (12), matrices

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

and the vectors $x = [x_{11}, x_{21}, x_{31}, x_{12}, x_{32}]^T$, $p = [-x_1', -x_2', -x_3', -x_1', -x_3']^T$, $b = [w_1, w_2, w_1]^T$. The matrix B is of full row rank.

For Problem (12), we introduce a penalty factor λ to add the equality constraint into the objective function. Then Problem (12) can be reformulated as

$$\begin{aligned} \min \quad & \tfrac{1}{2}x^T Q x + p^T x + \lambda x^T E^T E x \\ \text{s.t.} \quad & Bx \geq b, \\ & x \geq 0. \end{aligned} \tag{13}$$

Proposition 2. *In Problem (13), $Q + \lambda E^T E$ is a symmetric positive definite matrix, and B is of full row rank.*

Proof: Since Q is an identity matrix and $E^T E$ is a positive semi-definite matrix, it is obvious that $Q + \lambda E^T E$ is a symmetric positive definite matrix. Suppose that 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 B. Similar to the proof of Proposition 1, we can conclude that matrix B is of full row rank. □

Since the matrix $Q + \lambda E^T E$ in Problem (13) is symmetric positive definite, x is the global optimal solution of Problem (13) if and only if there exist vectors r and u such that the triple (x, r, u) satisfies the following KKT conditions:

$$\begin{cases} Qx + p + \lambda E^T E x - B^T y - u = 0, \\ v = Bx - b, \\ r^T v = 0, \\ u^T x = 0, \\ x, r, u, v \geq 0. \end{cases} \tag{14}$$

Similar to the single-row-height legalization problem, we can rewrite Condition (14) as the following LCP:

$$w = Az + q \geq 0, \quad z \geq 0 \text{ and } z^T w = 0, \tag{15}$$

where $w = \begin{bmatrix} u \\ v \end{bmatrix}$, $A = \begin{bmatrix} Q + \lambda E^T E & -B^T \\ B & 0 \end{bmatrix}$, $z = \begin{bmatrix} x \\ r \end{bmatrix}$, $q = \begin{bmatrix} p \\ -b \end{bmatrix}$. Similar to Theorem 1, we can also prove that the optimal solution of the convex quadratic programming problem (13) gives the solution of LCP (15), and vice versa.

In LCP (15), we choose the splitting matrices M and N as

$$\begin{aligned} M &= \begin{bmatrix} \frac{1}{\beta^*}(Q + \lambda E^T E) & 0 \\ B & \frac{1}{\theta^*} D \end{bmatrix}, \\ N &= \begin{bmatrix} (\frac{1}{\beta^*} - 1)(Q + \lambda E^T E) & B^T \\ 0 & \frac{1}{\theta^*} D \end{bmatrix}, \end{aligned} \tag{16}$$

where $D = tridiag(B(Q + \lambda E^T E)^{-1} B^T)$ is a tridiagonal approximation to the Schur complement $B(Q + \lambda E^T E)^{-1} B^T$ of the matrix A, and β^* and θ^* are two positive constants determined by the formulas given in [2]. Since the matrix $Q + \lambda E^T E$ is symmetric positive definite and B is a rectangular matrix of full row rank, the MMSIM convergence with the splitting matrices M and N for the LCP can be assured.

Algorithm 1 MMSIM Solver

Input: matrices: M, N, A, I
vectors: q, $s^{(0)}$, $z^{(0)}$
constants: γ, ε
Output: MMSIM legalization result

```
1: k = 0;
2: do
3:     solve (M + I)s^(k+1) = N s^(k) + (I − A)|s^(k)| − γ q;
4:     z^(k+1) = (1/γ)(|s^(k+1)| + s^(k+1));
5:     k + +;
6: until (|z^(k) − z^(k−1)| < ε)
7: return z^(k).
```

In Equation (16), matrix D involves computation of the inverse of the matrix $Q + \lambda E^T E$, which could be time-consuming. To speed up the computation, we use the Sherman-Morrison formula [20] to obtain the matrix D. Suppose that W is an $n \times n$ matrix, U is an $n \times k$ matrix, V is a $k \times n$ matrix, and $Y = W + UV$. Then, the Sherman-Morrison formula is

$$Y^{-1} = W^{-1} - W^{-1} U (I_k + V W^{-1} U)^{-1} V W^{-1}. \tag{17}$$

Using the Sherman-Morrison formula in Equation (17), we have

$$(Q + \lambda E^T E)^{-1} = Q^{-1} - Q^{-1} \lambda E^T (I_k + E Q^{-1} \lambda E^T)^{-1} E Q^{-1}.$$

Since matrix Q is an identity matrix, and EE^T is a diagonal matrix with all diagonal elements equal to 2, we have

$$\begin{aligned} (Q + \lambda E^T E)^{-1} &= I_n - \lambda E^T (I_k + \lambda E E^T)^{-1} E \\ &= I_n - \tfrac{\lambda}{2\lambda + 1} E^T E. \end{aligned}$$

Hence the matrix D in Equation (16) can be computed by

$$\begin{aligned} D &= tridiag(BB^T - \tfrac{\lambda}{2\lambda+1} BE^T E B^T) \\ &= tridiag(BB^T - \tfrac{\lambda}{2\lambda+1} BE^T (BE^T)^T). \end{aligned}$$

Basing on the MMSIM for LCP in [1], Algorithm 1 gives the algorithm for the quadratic programming problem (13). The MMSIM for the LCP in Equation (3) requires matrix Ω to be positive diagonal, which is an $n \times n$ identity matrix I in Algorithm 1 for easier computation. In line 3 of Algorithm 1, for a faster computation of the vector $s^{(k+1)}$, matrix $M + I$ is transformed to a lower triangular matrix using the Gaussian elimination method. We have the following theorem for the optimality of the MMSIM:

Theorem 2. *The iteration sequence $\{z^{(k)}\}_{k=0}^{+\infty} \subset \mathbb{R}_+^n$ generated by Algorithm 1 converges to the unique solution $z^\star \in \mathbb{R}_+^n$ of Problem (13) for any initial vector $s^{(0)} \in \mathbb{R}^n$.*

Proof: By [1], to guarantee the MMSIM convergence for LCP, the splitting matrices M, N in Equation (15) require $Q + \lambda E^T E$ to be symmetric positive definite and B to be a rectangular matrix of full row rank. In Proposition 2, we have proven that $Q + \lambda E^T E$ is a symmetric positive definite matrix, and the matrix B is of full row rank. By Theorem 2.1 in [2], if $0 < \beta^* < 2$ and θ^* satisfies

$$0 < \theta^* < \frac{2(2 - \beta^*)}{\beta^* \mu_{max}},$$

where μ_{max} is the largest eigenvalue of the matrix $\Gamma = D^{-1} B^T (Q + \lambda E^T E)^{-1} B$, the iteration method is convergent. In this way, the splitting matrices M and N do not lose the convergence property of the MMSIM in Algorithm 1. As a result, the convergence of Algorithm 1 can be guaranteed. □

From this theorem, for both the single-row-height and the mixed-row-height designs ignoring the right boundary constraints, the MMSIM solver guarantees the optimal solutions for Problem (5); Problem (5) assumes that all cells are sorted by their x-coordinates from global placement and are aligned to the nearest correct rows. The MMSIM optimality is empirically validated with experiments in Section 5.3.

4 OUR FRAMEWORK

Figure 4 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.

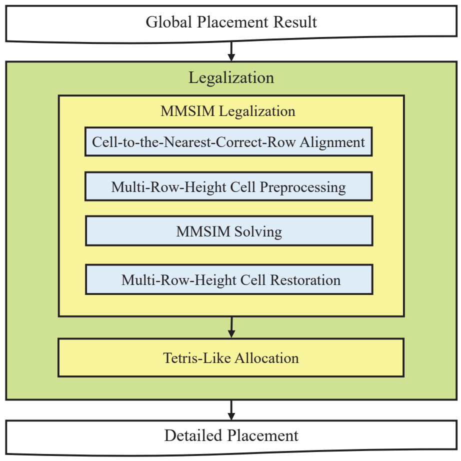


Figure 4: Our legalization flow.

Table 1: Statistics of the benchmarks and illegal cells after the MMSIM legalization.

Benchmark	#S. Cell	#D. Cell	Density	#I. Cell	%I. Cell
des_perf_1	103842	8802	0.91	902	0.80
des_perf_a	99775	8513	0.43	11	0.01
des_perf_b	103842	8802	0.50	6	< 0.01
edit_dist_a	121913	5500	0.46	20	0.02
fft_1	30297	1984	0.84	183	0.57
fft_2	30297	1984	0.50	2	< 0.01
fft_a	28718	1907	0.25	2	< 0.01
fft_b	28718	1907	0.28	10	0.03
matrix_mult_1	152427	2898	0.80	88	0.06
matrix_mult_2	152427	2898	0.79	62	0.04
matrix_mult_a	146837	2813	0.42	3	< 0.01
matrix_mult_b	143695	2740	0.31	7	< 0.01
matrix_mult_c	143695	2740	0.31	2	< 0.01
pci_bridge32_a	26268	3249	0.38	0	0
pci_bridge32_b	25734	3180	0.14	0	0
superblue11_a	861314	64302	0.43	40	< 0.01
superblue12	1172586	114362	0.45	89	< 0.01
superblue14	564769	47474	0.56	264	0.04
superblue16_a	625419	55031	0.48	42	< 0.01
superblue19	478109	27988	0.52	62	0.01
Average	252034	18454	0.49	90	0.03

As shown in Figure 4, ignoring the right boundary constraints, our legalization algorithm first aligns each cell to the nearest correct row. If a standard cell is of a multi-row height, it is divided into single-row-height subcells. After pre-processing the multi-row-height cells, the MMSIM discussed in Section 3.2 is used to solve the linear complementarity problem. Then, the multi-row-height cells are restored. Finally, a Tetris-like allocation approach is used to further align all the cells (including those violating the right boundary constraints) to placement sites on rows.

During pre-processing multi-row-height cells, each cell is divided into single-row-height subcells. After applying the MMSIM to solve the linear complementarity problem, we need to guarantee that the subcells of a multi-row-height cell take the same x-position. The penalty factor λ in Problem (13) affects the mismatch distances of the subcells of multi-row-height cells. Since the MMSIM is a convergent method, if the value of λ is large enough, there will be no mismatch distance for each multi-row-height cell in theory. Due to the precision of computation, however, there may still be overlaps among cells after the process of multi-row-height cell restoration. Further, since we ignore the right boundary constraints, some cells may be out of the right boundary. We resort to a Tetris-like allocation method to fix all these illegal placements.

The Tetris-like allocation method first aligns each cell to the nearest placement site. Then, for each row, we check the cells one by one for their legality. If a cell is overlapped with another cell or out of the right boundary, it will be marked as an illegal cell. Finally, for each illegal cell, we find the nearest free site to place it. It will be clear in the next section that the numbers of such illegal cells are very small or even zero for practical applications, so the Tetris-like method is empirically sufficient to resolve the illegal cells.

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 [7]. Their benchmarks were modified from the 2015 ISPD Detailed-Routing-Driven Placement Contest [5], 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. We conducted three experiments and reported the results in the following subsections. In our proposed algorithm, the parameter λ in Problem (12) was set to 1000, and β^* and θ^* were both set to 0.5 in the splitting matrices M and N.

5.1 Legalization on Mixed-Cell-Height Designs

Table 1 gives the characteristics of the benchmarks and the illegal cells after the MMSIM legalization (which need to be fixed by the Tetris-like 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 after the MMSIM legalization, 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 and fft_1), and the average ratio of illegal cells is only 0.03%. Since the MMSIM can generate the minimum displacement for the quadratic programming problem, and only very few cells, if not none, need to be moved to their nearest overlap-free locations, our mixed-cell-height legalization algorithm can guarantee the optimal solutions for the benchmarks without illegal cells after the MMSIM legalization (such as the benchmarks pci_bridge32_a and pci_bridge32_b) and achieve near optimal solutions for other benchmarks.

5.2 Mixed-Cell-Height Legalization Comparisons

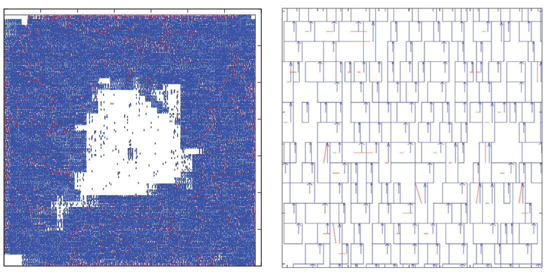
We compared our legalization algorithm with two state-of-the-art works [7] (a best paper nominee at DAC'16) and [18]. All the three algorithms consider power-rail alignment. With the binary codes provided by the authors of [7] and [18], we were able to run all the three algorithms on the same PC with a 3.40GHz Intel Core CPU and 16GB memory. It should be noted that the binary code for [7] 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, "Total Disp. (sites)" the cell displacement measured in the number of the placement site width, "ΔHPWL" the HPWL increase from the corresponding global placement, and "Runtime" the running time in second. The column "DAC'16" lists the experimental results reported in [7] directly. The columns "DAC'16-Imp", "ASP-DAC'17," and "Ours" give the corresponding results generated by the binary codes of the three algorithms. The last row of Table 2 shows the average normalized total displacement, ΔHPWL, and runtime ratios based on our results.

From Table 2, our legalization algorithm can achieve the best published quality. Compared with the work "DAC'16" [7], our legalization algorithm achieves 16% smaller cell displacement and 72% smaller HPWL increase rate. Compared with the work "DAC'16-Imp", our legalization algorithm achieves 10% smaller cell displacement and 41% smaller HPWL increase rate. Compared with the work "ASP-DAC'17" [18], our algorithm achieves 6% smaller cell displacement and 22% smaller HPWL increase rate. Both works in [7] and [18] optimize cells one by one in local regions, thus with a local view of the total cell legalization. In contrast, our algorithm optimizes all cells simultaneously by using the MMSIM method, leading to the best solution quality among the three algorithms (in fact, all published works) and much stable results. For the runtime, our legalization algorithm is 1.96× faster than "ASP-DAC'17", and it is comparable to "DAC'16" and "DAC'16-Imp". The experimental results show that our proposed legalization algorithm is effective and efficient. Figure 5(a) shows the legalization result of the benchmark fft_2 generated by our legalization algorithm, and Figure 5(b) shows a partial layout which justifies that the cell order is well preserved by our algorithm, a key to our superior results.

Table 2: Experimental results.

Benchmark	GP HPWL (m)	Total Disp. (sites) DAC'16	DAC'16-Imp	ASP-DAC'17	Ours	ΔHPWL DAC'16	DAC'16-Imp	ASP-DAC'17	Ours	Runtime (s) DAC'16	DAC'16-Imp	ASP-DAC'17	Ours
des_perf_1	1.43	373978	279545	474789	242622	2.85%	1.77%	0.99%	1.12%	7.2	6.1	7.5	2.4
des_perf_a	2.57	103956	81452	73057	72561	0.28%	0.16%	0.12%	0.07%	2.6	2.5	3.8	2.3
des_perf_b	2.13	95747	81540	72429	71888	0.31%	0.21%	0.16%	0.08%	2.4	2.2	3.9	2.3
edit_dist_a	5.25	59884	59814	60971	62961	0.10%	0.10%	0.12%	0.09%	1.9	1.8	4.9	2.8
fft_1	0.46	58429	54501	53389	46121	1.66%	1.47%	0.89%	0.87%	1.1	1.0	1.3	0.7
fft_2	0.46	27762	25697	21018	20979	0.87%	0.73%	0.67%	0.51%	0.4	0.4	1.1	0.6
fft_a	0.75	19600	19613	18150	18304	0.33%	0.33%	0.29%	0.24%	0.3	0.2	1.2	0.6
fft_b	0.95	24500	28461	21234	21671	0.33%	0.18%	0.30%	0.27%	0.4	0.4	1.2	0.6
matrix_mult_1	2.39	82322	80235	73682	71793	0.28%	0.27%	0.21%	0.21%	3.9	4.0	5.4	3.6
matrix_mult_2	2.59	76109	75810	65959	65876	0.22%	0.21%	0.17%	0.17%	4.0	4.2	5.4	3.7
matrix_mult_a	3.77	49385	46001	40736	40298	0.14%	0.11%	0.09%	0.08%	1.6	1.6	5.7	3.4
matrix_mult_b	3.43	43931	40059	37243	37215	0.13%	0.10%	0.09%	0.08%	1.3	1.2	5.6	3.2
matrix_mult_c	3.29	42466	42490	40942	40710	0.11%	0.11%	0.11%	0.09%	1.4	1.4	5.6	3.2
pci_bridge32_a	0.46	28041	27832	26674	26289	0.58%	0.57%	0.63%	0.45%	0.3	0.3	1.2	0.6
pci_bridge32_b	0.98	27757	27864	26160	26028	0.13%	0.13%	0.06%	0.05%	0.2	0.2	1.0	0.4
superblue11_a	42.94	1795695	1786342	1983090	1742941	0.15%	0.15%	0.26%	0.16%	23.4	29.7	50.3	26.3
superblue12	39.23	2097725	2015678	1995140	1963403	0.22%	0.20%	0.22%	0.21%	106.5	103.6	56.5	38.6
superblue14	27.98	1604077	1599810	1497490	1566966	0.22%	0.22%	0.18%	0.23%	17.1	16.7	48.1	17.7
superblue16_a	31.35	1177179	1173106	1147530	1135186	0.12%	0.11%	0.11%	0.11%	21.7	20.7	41.8	18.7
superblue19	20.76	809755	806529	808164	781928	0.14%	0.14%	0.13%	0.12%	10.9	10.5	29.6	13.2
N. Average		1.16	1.10	1.06	1.00	1.72	1.41	1.22	1.00	1.02	0.97	1.96	1.00



(a) (b)

Figure 5: (a) Legalization result of the benchmark fft_2. Cells are in blue, and displacement in red. (b) Partial layout of (a).

5.3 MMSIM Optimality on Single-Row-Height Designs

For the problem with single-row-height cells only, we tested our proposed legalization algorithm, and the legalization algorithm that replaces our MMSIM solver with the *PlaceRow* subroutine in Abacus [14] on the benchmarks without doubling the cell heights. The experimental results show that the two algorithms generate exactly the same total cell displacements for all the 20 benchmarks, and our MMSIM solver gives a 1.51X speedup over *Placerow*; for example, both methods give 58850, 1618580, and 2023 displacement sites for the benchmarks des_perf_1 (the 1st benchmark), superblue12 (the largest), and pci_bridge32_b (the smallest), respectively. Since *PlaceRow* in Abacus [14] achieves optimal solutions when cells are aligned to rows and the cell ordering is decided (same for the MMSIM), this experiment empirically validates the MMSIM optimality, as proven in Theorem 2.

Note that, for the legalization problem with mixed-cell-height circuit designs, the optimal substructure cannot be guaranteed because legalizing a multi-row-height cell may cause cells in another row illegal. Hence, the dynamic programming algorithm in Abacus is not applicable, while our proposed MMSIM can still guarantee the optimality for this problem without right boundary constraints.

6 CONCLUDING REMARKS

We have presented a near-optimal algorithm for the mixed-cell-height standard-cell legalization problem. Our algorithm converts this problem to a linear complementarity problem, splits the matrices in the problem to meet the convergence requirement of the MMSIM, and adopts a Tetris-like allocation approach to legalize out-of-the-right-boundary cells. Experimental results have shown that our algorithm can achieve the best cell displacement and wirelength among all published methods in reasonable runtime. The MMSIM optimality has been theoretically proven and empirically validated. In particular, our formulation provides new generic solutions and research directions for various optimization problems that require solving large-scale quadratic programs efficiently (e.g., global placement [17], buffering and wire sizing [8], dummy fill insertion [15], analog circuit optimization [16], etc.), by applying our proposed linear complementarity relaxation and its fast MMSIM.

REFERENCES

[1] Z. Z. Bai. Modulus-based matrix splitting iteration methods for linear complementarity problems. *Numerical Linear Algebra with Applications*, 17, pp. 917–933, 2010.

[2] Z. Z. Bai, B. Parlett, and Z. Q. Wang. On generalized successive overrelaxation methods for augmented linear systems. *Numerische Mathematik*, 102, pp. 1–38, 2005.

[3] S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge: Cambridge University Press, 2004.

[4] S. H. Baek, H. Y. Kim, Y. K. Lee, D. Y. Jin, S. C. Park, and J. D. Cho. Ultra high density standard cell library using multi-height cell structure. In *Proceedings of SPIE 7268*, pp. 72680C–72680C, 2008.

[5] I. S. Bustany, D. Chinnery, J. R. Shinnerl and V. Tutsi. ISPD 2015 benchmarks with fence regions and routing blockages for detailed-routing-driven placement. In *Proceedings of ACM International Symposium on Physical Design*, pp. 157–164, 2015.

[6] J. Chen, W. X. Zhu, and Z. Peng. A heuristic algorithm for the strip packing problem. *Journal of Heuristics*, 18(4), pp. 677–697, 2012.

[7] W. K. Chow, C. W. Pui, and F. Y. Young. Legalization algorithm for multiple-row height standard cell design. In *Proceedings of ACM/IEEE Design Automation Conference*, 2016.

[8] C. C. N. Chu and D. F. Wong. 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), pp. 787–798, 1999.

[9] S. Dobre, A. B. Kahng, and J. Li. Mixed cell-height implementation for improved design quality in advanced nodes. In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp. 854–860, 2015.

[10] E. M. Gertz and S. J. Wright. Object-oriented software for quadratic programming. *ACM Transactions on Mathematical Software*, 29(1), pp. 58–81, 2003.

[11] D. Hill. Method and system for high speed detailed placement of cells within integrated circuit designs. In *U.S. Patent 6370673*, 2002.

[12] Y. Lin, B. Yu, X. Xu, J. Gao, N. Viswanathan, W. Liu, Z. Li, C. J. Alpert, and D. Z. Pan. MrDP: multiple-row detailed placement of heterogeneous-sized cells for advanced nodes. In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp. 7:1–7:8, 2016.

[13] J. Nocedal, S. J. Wright. *Numerical optimization.* New York: Springer, 2006.

[14] P. Spindler, U. Schlichtmann, and F. M. Johannes. Abacus: fast legalization of standard cell Circuits with minimal movement. In *Proceedings of ACM International Symposium on Physical Design*, pp. 47–53, 2008.

[15] Y. Tao, C. Yan, Y. Lin, S. Wang, David Z. Pan, and X. Zeng. A Novel Unified Dummy Fill Insertion Framework with SQP-Based Optimization Method. In *Proceeding of IEEE/ACM International Conference on Computer-Aided Design*, pp. 88:1–88:8, 2016.

[16] S. Vichik, M. Arcak, and F. Borrelli. Stability of an analog optimization circuit for quadratic programming. *Systems&Control Letters*, 88, pp. 68–74, 2016.

[17] N. Viswanathan, G. J. Nam, C. J. Alpert, P. Villarrubia, H. Ren, and C. Chu. RQL: global placement via relaxed quadratic spreading and linearization. In *Proceedings of ACM/IEEE Design Automation Conference*, 2007.

[18] C. H. Wang, Y. Y. Wu, J. Chen, Y. W. Chang, S. Y. Kuo, W. X. Zhu, and Genghua Fan. An effective legalization algorithm for mixed-cell-height standard cells. In *Proceedings of IEEE/ACM Asia and South Pacific Design Automation Conference*, pp. 450–455, 2017.

[19] J. Wang, A. K. Wong, and E. Y. Lam. Standard cell layout with regular contact placement. *IEEE Transactions on Semiconductor Manufacturing*, 17(3), pp. 375–383, 2004.

[20] P. H. William, S. A. Teukolsky, V. T. William, and B. P. Flannery. *Numerical Recipes: The Art of Scientific Computing (3rd ed.), Section 2.7.1 Sherman-Morrison formula.* New York: Cambridge University Press, 2007.

[21] G. Wu and C. Chu. 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), pp. 1569–1573, 2015.