Linear-time Mixed-Cell-Height Legalization for Minimizing Maximum Displacement

Chung-Hsien Wu
Department of Computer Science,
National Tsing Hua University
Hsinchu, Taiwan
s109062624@m109.nthu.edu.tw

Wai-Kei Mak
Department of Computer Science,
National Tsing Hua University
Hsinchu, Taiwan
wkmak@cs.nthu.edu.tw

Chris Chu
Department of Electrical and
Computer Engineering,
Iowa State University
Ames, IA, USA
cnchu@iastate.edu

ABSTRACT

Due to the aggressive scaling of advanced technology nodes, multiple-row-height cells have become more and more common in VLSI design. Consequently, the placement of cells is no longer independent among different rows, which makes the traditional row-based legalization techniques obsolete. In this work, we present a highly efficient linear-time mixed-cell-height legalization approach that optimizes both the total cell displacement and the maximum cell displacement. First, a fast window-based cell insertion technique introduced in [4] is applied to obtain a feasible initial row assignment and cell ordering which is known to be good for total displacement consideration. In the second stage, we use an iterative cell swapping algorithm to change the row assignment and the cell order of the critical cells for maximum displacement reduction. Then we develop an optimal linear time DAG-based fixed row and fixed order legalization algorithm to minimize the maximum cell displacement. Finally, we propose a cell shifting heuristic to reduce the total cell displacement without increasing the maximum cell displacement. Using the proposed approach, the quality provided by the global placement can be preserved as much as possible. Compared with the state-of-the-art work [4], experimental results show that our proposed algorithm can reduce the maximum cell displacement by more than 11% on average with similar average cell displacement.

CCS CONCEPTS

• **Hardware** → **Placement**.

KEYWORDS

legalization, VLSI placement, mixed-cell-height circuits

ACM Reference Format:
Chung-Hsien Wu, Wai-Kei Mak, and Chris Chu. 2022. Linear-time Mixed-Cell-Height Legalization for Minimizing Maximum Displacement. In *Proceedings of the 2022 International Symposium on Physical Design (ISPD '22), March 27–30, 2022, Virtual Event, Canada.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3505170.3506724

This work was supported in part by the Ministry of Science and Technology under grant MOST 110-2221-E-007-122.

ISPD '22, March 27–30, 2022, Virtual Event, Canada

ACM ISBN 978-1-4503-9210-5/22/03…$15.00
https://doi.org/10.1145/3505170.3506724

1 INTRODUCTION

Standard cells used to have the same height for easier design and optimization in traditional VLSI design. However, with the fast-growing complexity and the shrinking number of routing tracks within standard cells, mixed-cell-height circuits are adopted for more advanced technology nodes. This allows cells with complex function or large driving power to be double or even multiple-row height, while simple cells are still designed as single-row-height structure.

A modern placement flow is comprised of three stages. (1) Global placement which targets at finding a position for each cell with various objectives (usually the wirelength and local cell density) while ignoring the non-overlapping constraint between cells. (2) Legalization which places cells into correct placement sites and removes overlaps while minimizing the total cell displacement or other metrics (e.g., wirelength). (3) Detailed placement which further improves the legalization result. The above mentioned multiple-row-height standard cells lead to more complicated challenges for the legalization problem, since each row of a design cannot be considered independently anymore, and we should also take care of the power-rail alignment as pointed out in [2]. Moreover, most of the published works considered average cell displacement only. But an ideal legalization algorithm should optimize both average displacement and maximum displacement in order to preserve the quality of the global placement [3].

Some recent works considered the mixed-cell-height standard cell legalization problem. Wu et al. [8] first solved the legalization problem with double-row-height cells by matching single-row-height cells together, then using traditional single-row-height techniques to solve the legalization problem. Chow et al. [2] proposed a multi-row local legalization algorithm by inserting each cell within a defined window for it, while minimizing the total cell displacement of all the cells in the window. In [6], a chain move legalization method followed by a network flow-based formulation for ordered multi-row placement for wirelength and density optimization was proposed. Wang et al. [7] extended Abacus to handle the mixed-cell-height standard cell legalization problem by mostly honoring the cell ordering from global placement. However, [7] also pointed out that completely following the cell ordering of global placement tended to create a lot of dead space which affected both placement feasibility and optimality. So, it also considered inserting a cell into an existing whitespace when it is placed.

[1], [5], [9] used a common method that formulated the mixed-cell-height legalization problem as a mixed integer quadratic programming problem (MIQP) by preserving the global placement cell ordering, then converted the MIQP to a quadratic programming problem (QP). They further reformulated the QP as a linear complementarity problem (LCP). Finally, they solved the LCP by a modulus-based matrix splitting iteration method (MMSIM). After MMSIM, a post-processing step was applied to legalize those cells placed beyond the chip boundary. Compared to [1], [9] added the fence region constraint and [5] considered maximum cell displacement as an extra weighted term in their objective function. The state-of-the-art work [4] proposed a fast window-based cell insertion technique followed by an iterative min-cost bipartite matching algorithm to reduce the maximum displacement and a min-cost flow formulation of the fixed-row-and-order legalization to optimize a weighted sum of the maximum and average displacement. In addition, [4] also considered pin accessibility and fence region constraints.

In this paper, we propose a highly efficient linear-time legalization approach to deal with the mixed-cell-height standard cell legalization problem. It targets at optimizing both the maximum cell displacement and the total cell displacement. The rest of this paper is organized as follows. In Section 2, we introduce the background of cell legalization problem and some extra rules regarding multiple-row-height cells. Then we give the problem formulation. Section 3 presents our approach in details. Experimental results are shown in Section 4. Finally, Section 5 gives the conclusion.

2 PRELIMINARIES

Legalization is the process of removing overlaps among cells and aligning them to the placement sites on rows, while preserving the quality provided by the global placement as much as possible. Legalizers usually target at minimizing the cell displacement. Additional considerations like wirelength or fence region are treated as secondary objectives in recent works. Placement legalization is an important step in the VLSI design flow. The result has a significant impact on later stages in the design flow.

In the past, when all of the standard cells were one-row-height, the power-rail alignment constraint could be trivially solved by flipping cells horizontally. As for the multiple-row-height standard cells used today, odd-row-height cells can utilize the same technique as single-row-height cells, while even-row-height cells should be placed in every other row with a proper power rail, in order to satisfy the power-rail alignment constraint.

2.1 Problem Formulation

We are given a global placement where there are cells of multiple-row height. The objective is to legalize the placement of the movable cells such that the maximum displacement and total displacement are minimized.

During this stage, the following constraints should be satisfied:

- Cells must be placed inside the layout region.
- Cells must not overlap with one another or with the pre-placed macros.
- Cells must be located at placement sites on the rows.
- Cells must be placed in rows with proper P/G rail alignment.

3 PROPOSED APPROACH

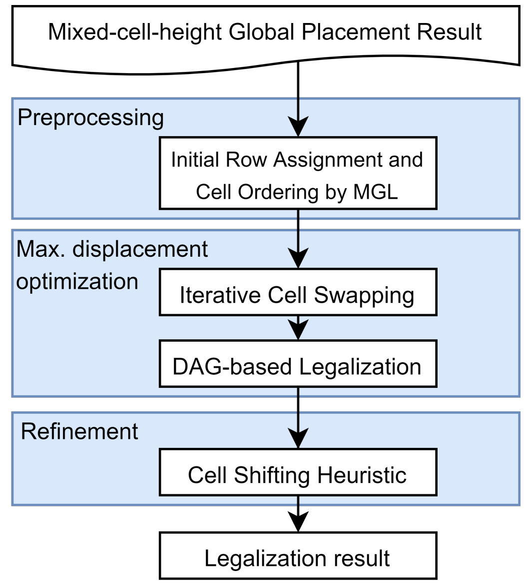


Figure 1: The overall flow of our method.

Fig. 1 illustrates the flow of our approach. There are three stages in our flow: (1) preprocessing, (2) maximum displacement optimization, and (3) refinement. In the preprocessing stage, we obtain a feasible initial row assignment and cell order by MGL in [4] which is also good for total displacement. In the second stage, we propose a cell swapping algorithm, which tries to minimize the maximum displacement by finding cells to swap with the maximum displacement cells iteratively. Then an optimal DAG-based algorithm is applied to further minimize the maximum displacement. In the refinement stage, we improve the total displacement with maximum displacement fixed by a cell shifting heuristic.

3.1 Initial Row Assignment and Cell Order

As mentioned before, assigning every cell to a row close to its global placement position and fixing the cell order in every row according to the order in global placement which lacks consideration of dead space will affect the placement feasibility and the solution quality. Instead, we obtain a good initial row assignment of cells and cell ordering using multi-row global legalization (MGL) in [4]. It ensures that placement legalization under this initial row assignment and cell ordering is feasible. Moreover, we know that there exists a high quality legalized placement in terms of total cell displacement under this initial row assignment and cell ordering.

3.2 Iterative Cell Swapping

After obtaining a good row assignment and cell order for total displacement, we use an iterative cell swapping algorithm to reduce the maximum displacement. Cell swapping changes the row assignment as well as the cell order in some rows. In each iteration, we first find the maximum displacement cell c_{max} and its displacement $disp_{max}$. Then we pick a target cell c_{target} within distance $disp_{max}$ from the global placement position of c_{max} such that swapping c_{max} and c_{target} will reduce the maximum displacement and will not increase the total displacement. We prefer a target cell with $width(c_{target}) + height(c_{target})$ as close to $width(c_{max}) + height(c_{max})$ as possible.

Note that swapping cells with different heights may potentially violate the power-rail alignment constraint. If an even-row-height cell violates the power-rail alignment constraint, it will be moved up or down a row depending on its global placement position to fix the violation. Moreover, we check if the new placement after each swapping operation is still legalizable. A necessary and sufficient condition for the existence of a feasible legalized placement is shown in Theorem 2 of section 3.3. So, subsequently the DAG-based legalization algorithm is guaranteed to have a feasible solution under the new row assignment and cell ordering.

3.3 DAG-Based Legalization

Suppose the row assignment of cells and the ordering of cells in each row have been fixed. We want to compute a legalized placement that minimizes the infinity-norm of the vector of displacement from the global placement. Therefore, our objective is to minimize the maximum displacement of any cell in the final legalized placement.

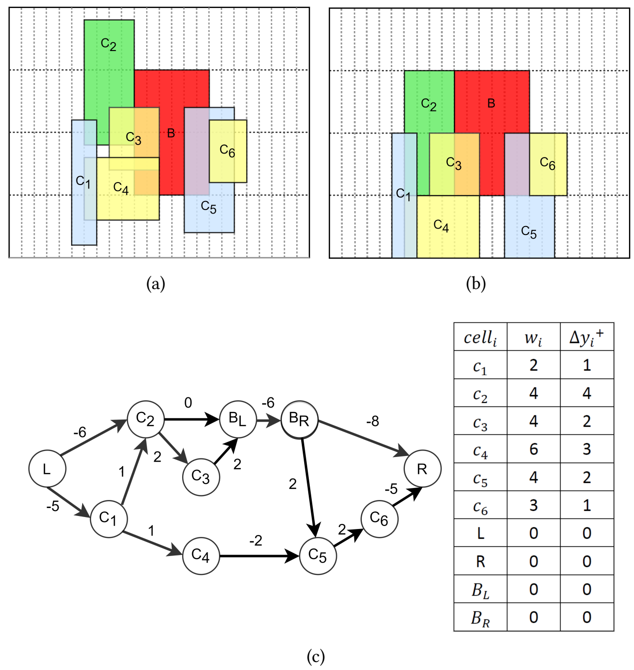
In our proposed algorithm, the smallest unit of measurement is the site width and everything is measured in terms of an integral number of site widths. Below are the inputs and outputs of our approach.

- Let $\widehat{x_i}(\widehat{y_i})$ be the $x(y)$-coordinate of the left(bottom) boundary of cell i in the given global placement.
- Let $x_i(y_i)$ be the $x(y)$-coordinate of the left(bottom) boundary of cell i in the legalized placement. As the row assignment is fixed, y_i is fixed for all i.
- Let $\Delta y_i^+ = |y_i - \widehat{y_i}|$. In other words, Δy_i^+ is the magnitude of the y-displacement of cell i which is fixed by the given row assignment.
- Let w_i and h_i be the width and height of cell i.

We define a *horizontal constraint graph* G as follows. G is a directed acyclic graph (DAG) in which all edges are pointing from left to right. There is an edge from cell i to cell j in G if cell j is immediately on the right of cell i in some row. For a unified treatment, the left and right boundaries of the layout region and obstacles are modeled as fixed cells of width 0. Let F be the set of fixed cells. For every edge $(i, j) \in G$, we set the edge length to $w_i - (\widehat{x_j} - \widehat{x_i})$.

An example is shown in Fig. 2. Fig. 2(a) shows a given placement of six cells c_1 to c_6 and an obstacle B. Suppose the desired row assignment is cells c_1 and c_5 to row pair $(1, 2)$, cell c_2 to row pair $(2, 3)$, cells c_3 and c_6 to row 2, and cell c_4 to row 1. Fig. 2(b) shows the perturbed placement under the above row assignment. The corresponding horizontal constraint graph is shown in Fig. 2(c). The left and right boundaries of the layout region are modelled by fixed cells L and R, respectively. The left and right ends of obstacle B are modelled by fixed cells B_L and B_R, respectively. All these fixed nodes have zero width.

We want to compute a legalized placement with no cell overlap that minimizes the maximum displacement of any cell from its global placement position. The mathematical formulation of our problem can be written as follows.



$cell_i$	w_i	Δy_i^+
c_1	2	1
c_2	4	4
c_3	4	2
c_4	6	3
c_5	4	2
c_6	3	1
L	0	0
R	0	0
B_L	0	0
B_R	0	0

Figure 2: (a) Cells at their global placement positions. (b) Cells assigned to rows. (c) The corresponding horizontal constraint graph of Fig. 2(a).

$$D = \text{Minimize} \quad \max_i (|x_i - \widehat{x_i}| + \Delta y_i^+) \tag{1}$$

$$\text{subject to} \quad x_i + w_i \le x_j \text{ for all } (i, j) \in G \tag{2}$$

Note that constraints related to the placement boundaries and obstacles are included implicitly in (2) as the constraint graph definition is extended to model them.

Next, we analyze the problem of placement legalization for maximum displacement minimization, which provides a mathematical foundation for our proposed algorithm.

Suppose p_{ij}^* is the directed path from cell i to cell j in G with the largest total cell width and w_{ij}^* is the total width of all cells in p_{ij}^*, i.e., $w_{ij}^* = \sum_{k \in p_{ij}^* \wedge k \neq j} w_k$. We define *deficit* s_{ij} by $s_{ij} = w_{ij}^* - (\widehat{x_j} - \widehat{x_i})$ for any cell j reachable from cell i. Note that if edge $(i, j) \in G$ and if there is no other path from i to j, then $s_{ij} = w_i - (\widehat{x_j} - \widehat{x_i})$, which is equal to the length of edge (i, j). It is easy to see that the deficit s_{ij} indicates the minimum relative right movement of cell j with respect to cell i in order to avoid any overlap of cells on p_{ij}^*. Note that s_{ij} can be negative. For example, if the minimum relative right movement of cell j with respect to cell i is -5 (i.e., $s_{ij} = -5$), then it means we can tolerate a relative left movement of cell j with respect to cell i of at most 5.

To analyze the problem further, we need to introduce some other useful terms first.

- Let $\rho_i = \max_k(\Delta y_k^+ + s_{ki})$ and $\theta_i = \max_j(s_{ij} + \Delta y_j^+)$ for any cell i.

- Let $\sigma = \max_{k,j}(\Delta y_k^+ + s_{kj} + \Delta y_j^+)$.
- Let $l_i = \max_{k \in F}(s_{ki} + \Delta y_i^+)$ and $r_i = \max_{j \in F}(\Delta y_i^+ + s_{ij})$ for any cell i. Recall that F is the set of fixed cells modelling the left/right boundaries of the layout region and the left/right ends of obstacles.
- Let $l = \max_i l_i$ and $r = \max_i r_i$.

It is not hard to see that

$$\sigma = \max_i(\rho_i + \theta_i) \tag{3}$$

$$l = \max_i l_i = \max_{k,i;k \in F}(s_{ki} + \Delta y_i^+) = \max_{k \in F} \theta_k \tag{4}$$

$$r = \max_i r_i = \max_{i,j;j \in F}(\Delta y_i^+ + s_{ij}) = \max_{j \in F} \rho_j \tag{5}$$

If there exists any cell i such that $l_i + r_i - 2\Delta y_i^+ > 0$, then the design cannot be legalized under the given row assignment and cell ordering. In other words, $l_i + r_i - 2\Delta y_i^+ \leq 0$ for all cell i is a necessary condition for the existence of a feasible legalized placement under the given row assignment and cell ordering. Note that $l_i + r_i - 2\Delta y_i^+ \leq 0$ for all i means all row segments can accommodate all cells assigned to them. Subsequently (in Theorem 2), we are going to prove that it is also a sufficient condition.

Below we introduce a few lemmas about the optimal maximum displacement D.

Lemma 1. *$D \geq \max_i \Delta y_i^+$.*

Lemma 2. *$D \geq l$.*

Lemma 3. *$D \geq r$.*

It is not difficult to see that Lemmas 1 to 3 are trivially true. In Lemma 4 below, we introduce another lower bound for the optimal maximum displacement D.

Lemma 4. *$D \geq \lceil \sigma/2 \rceil$.*

Proof. Consider the pair i, j the deficit of which is σ, i.e., $i, j = \arg\max_{i,j}(\Delta y_i^+ + s_{ij} + \Delta y_j^+)$. To legalize the cells along the path p_{ij}^*, it is impossible for both cells i and j to have a displacement less than $\sigma/2$. Hence, the optimal maximum displacement D is greater than or equal to $\lceil \sigma/2 \rceil$. □

By Lemmas 1 to 4, we have the following corollary.

Corollary 1. *$D \geq M$ where $M = \max\{\max_i \Delta y_i^+, l, r, \lceil \sigma/2 \rceil\}$.*

Next we are going to present two main theorems. Based on the theorems, we can derive an optimal linear time algorithm to compute a legalized placement with minimum maximum displacement.

Theorem 1. *For all i, $\max_i \Delta y_i^+$, ρ_i, θ_i, l_i, and r_i can be found in linear time.*

Proof. It is clear that $\max_i \Delta y_i^+$ can be computed in linear time for all i.
By traversing the cells in the horizontal constraint graph in topological order from left to right once, we can compute $\rho_i = \max_k(\Delta y_k^+ + s_{ki})$ and $l_i = \max_{k \in F}(s_{ki} + \Delta y_i^+)$ by dynamic programming for all i in linear time.
By traversing the cells in the horizontal constraint graph in reverse topological order from right to left once, we can compute $\theta_i = \max_j(s_{ij} + \Delta y_j^+)$ and $r_i = \max_{j \in F}(\Delta y_i^+ + s_{ij})$ by dynamic programming for all i in linear time. □

Theorem 2. *Assume that $l_i + r_i - 2\Delta y_i^+ \leq 0$ for all i. Let*

$$\Delta x_i = \begin{cases} l_i - \Delta y_i^+ & \text{if } l_i - \Delta y_i^+ > \mu_i \\ -r_i + \Delta y_i^+ & \text{if } -r_i + \Delta y_i^+ < \mu_i \\ \mu_i & \text{otherwise} \end{cases}$$

where $\mu_i = \lceil (\rho_i - \theta_i)/2 \rceil$. By moving cell i to the right by Δx_i sites, (a) the resulting placement is legal, and (b) all cells are displaced by at most $M = \max\{\max_i \Delta y_i^+, l, r, \lceil \sigma/2 \rceil\}$.

Proof. (a) We prove that the resulting placement is legal. In other words, we want to show that if there exists an edge (i, j) in G for cell i and cell j, then $\Delta x_j - \Delta x_i \geq s_{ij}$. Since there are three cases for Δx_i and three cases for Δx_j, there are totally nine cases to consider. Before considering the nine cases in turn, we first note that

$$\mu_j - \mu_i \geq s_{ij} \tag{6}$$

since

$$\begin{aligned} &\mu_j - \mu_i \\ =&\lceil (\rho_j - \theta_j)/2 \rceil - \lceil (\rho_i - \theta_i)/2 \rceil \\ \geq&\lceil ((\rho_i + s_{ij}) - (\theta_i - s_{ij}))/2 \rceil - \lceil (\rho_i - \theta_i)/2 \rceil \\ &\quad \because \rho_j \geq \rho_i + s_{ij} \text{ and } \theta_i \geq \theta_j + s_{ij} \text{ by def. of } \rho_j \text{ and } \theta_i \\ =&s_{ij} \quad \because \rho_i, \theta_i, s_{ij} \text{ are integers} \end{aligned}$$

Case 1 Suppose Δx_i is given by $l_i - \Delta y_i^+$ and Δx_j is given by $l_j - \Delta y_j^+$. Then,

$$\begin{aligned} &\Delta x_j - \Delta x_i \\ =&(l_j - \Delta y_j^+) - (l_i - \Delta y_i^+) \\ \geq&(l_i - \Delta y_i^+ + s_{ij}) - (l_i - \Delta y_i^+) \\ &\quad \because l_j - \Delta y_j^+ \geq l_i - \Delta y_i^+ + s_{ij} \text{ by def. of } l_j \\ =&s_{ij} \end{aligned}$$

Hence, in this case cells i and j will not overlap.

Case 2 Suppose Δx_i is given by $l_i - \Delta y_i^+$ and Δx_j is given by $-r_j + \Delta y_j^+$. Then,

$$\begin{aligned} &\Delta x_j - \Delta x_i \\ =&(-r_j + \Delta y_j^+) - (l_i - \Delta y_i^+) \\ \geq&(-r_i + \Delta y_i^+ + s_{ij}) - (l_i - \Delta y_i^+) \\ &\quad \because r_i - \Delta y_i^+ \geq r_j - \Delta y_j^+ + s_{ij} \text{ by def. of } r_i \\ =&-l_i - r_i + 2\Delta y_i^+ + s_{ij} \\ \geq&s_{ij} \quad \because l_i + r_i - 2\Delta y_i^+ \leq 0 \text{ by assumption} \end{aligned}$$

Hence, in this case cells i and j will not overlap.

Case 3 Suppose Δx_i is given by $l_i - \Delta y_i^+$ and Δx_j is given by μ_j. Assume to the contrary that $\Delta x_j - \Delta x_i < s_{ij}$, then

$$\begin{aligned} &\Delta x_j - \Delta x_i < s_{ij} \\ \Rightarrow&\mu_j - (l_i - \Delta y_i^+) < s_{ij} \\ \Rightarrow&\mu_j < l_i - \Delta y_i^+ + s_{ij} \\ \Rightarrow&\mu_j < l_j - \Delta y_j^+ \\ &\quad \because l_j - \Delta y_j^+ \geq l_i - \Delta y_i^+ + s_{ij} \text{ by def. of } l_j \end{aligned}$$

But if $\mu_j < l_j - \Delta y_j^+$, Δx_j would be equal to $l_j - \Delta y_j^+$ by definition of Δx_j, which contradicts the fact that Δx_j is μ_j.

Hence, we must have $\Delta x_j - \Delta x_i \geq s_{ij}$, which means cells i and j will not overlap.

Case 4 Suppose Δx_i is given by $-r_i + \Delta y_i^+$ and Δx_j is given by $l_j - \Delta y_j^+$. Then, by the definition of Δx_i, we must have $-r_i + \Delta y_i^+ < \mu_i$. Similarly, by the definition of Δx_j, we must have $l_j - \Delta y_j^+ > \mu_j$. Hence, $\Delta x_j - \Delta x_i > \mu_j - \mu_i$. Finally, by Inequality (6), we get $\Delta x_j - \Delta x_i > s_{ij}$, which means cells i and j will not overlap.

Case 5 Suppose Δx_i is given by $-r_i + \Delta y_i^+$ and Δx_j is given by $-r_j + \Delta y_j^+$. Then,

$$\begin{aligned}
&\Delta x_j - \Delta x_i \\
=&(-r_j + \Delta y_j^+) - (-r_i + \Delta y_i^+) \\
=&(-r_j + \Delta y_j^+) + (r_i - \Delta y_i^+) \\
\geq&(-r_j + \Delta y_j^+) + (r_j - \Delta y_j^+ + s_{ij}) \\
&\quad \because r_i - \Delta y_i^+ \geq r_j - \Delta y_j^+ + s_{ij} \text{ by def. of } r_i \\
=&s_{ij}
\end{aligned}$$

Hence, in this case cells i and j will not overlap.

Case 6 Suppose Δx_i is given by $-r_i + \Delta y_i^+$ and Δx_j is given by μ_j. Then, by the definition of Δx_i, we must have $-r_i + \Delta y_i^+ < \mu_i$. Hence, $\Delta x_j - \Delta x_i > \mu_j - \mu_i$. Finally, by Inequality (6), we get $\Delta x_j - \Delta x_i > s_{ij}$, which means cells i and j will not overlap.

Case 7 Suppose Δx_i is given by μ_i and Δx_j is given by $l_j - \Delta y_j^+$. Then, by the definition of Δx_j, we must have $l_j - \Delta y_j^+ > \mu_j$. Hence, $\Delta x_j - \Delta x_i > \mu_j - \mu_i$. Finally, by Inequality (6), we get $\Delta x_j - \Delta x_i > s_{ij}$, which means cells i and j will not overlap.

Case 8 Suppose Δx_i is given by μ_i and Δx_j is given by $-r_j + \Delta y_j^+$. As in Case 3, we can prove that $\Delta x_j - \Delta x_i < s_{ij}$ would lead to a contradiction. Hence, we must have $\Delta x_j - \Delta x_i \geq s_{ij}$, which means cells i and j will not overlap.

Case 9 Suppose Δx_i is given by μ_i and Δx_j is given by μ_j. Then, by Inequality (6), we have $\Delta x_j - \Delta x_i \geq s_{ij}$. Hence, in this case cells i and j will not overlap.

(b) It remains to prove that each cell is displaced by at most M, i.e., $|\Delta x_i| + \Delta y_i^+ \leq M$. We have three cases for $|\Delta x_i|$. Before considering the different cases in turn, we first note that

$$|\mu_i| + \Delta y_i^+ \leq M \tag{7}$$

Assume to the contrary that $|\mu_i| + \Delta y_i^+ > M$.

(i) If $\mu_i \geq 0$, then we would have

$$\begin{aligned}
&\lceil(\rho_i - \theta_i)/2\rceil + \Delta y_i^+ > M \\
\Rightarrow&\lceil(\rho_i - \theta_i)/2\rceil + \Delta y_i^+ > \lceil\sigma/2\rceil \quad \text{by def. of } M \\
\Rightarrow&\lceil(\rho_i - \theta_i)/2\rceil + \Delta y_i^+ > \lceil(\rho_i + \theta_i)/2\rceil \\
&\quad \because \sigma = \max_k(\rho_k + \theta_k) \text{ by (3)} \\
\Rightarrow&\rho_i - \theta_i + 2\Delta y_i^+ > \rho_i + \theta_i \\
&\quad \because \rho_i, \theta_i, \Delta y_i^+ \text{ are integers} \\
\Rightarrow&\Delta y_i^+ > \theta_i
\end{aligned}$$

which is impossible as we know that $\theta_i \geq \Delta y_i^+$ by the definition of θ_i.

(ii) Similar to (i), if $\mu_i < 0$, then $|\mu_i| + \Delta y_i^+ > M$ would imply $\Delta y_i^+ > \rho_i$, which is impossible as we know that $\rho_i \geq \Delta y_i^+$ by the definition of ρ_i.

Case 1 Suppose Δx_i is given by $l_i - \Delta y_i^+$.

(i) If $l_i - \Delta y_i^+ \geq 0$, then $|l_i - \Delta y_i^+| + \Delta y_i^+ = (l_i - \Delta y_i^+) + \Delta y_i^+ = l_i \leq M$ by the definition of M. Hence, $|\Delta x_i| + \Delta y_i^+ \leq M$.

(ii) Assume $l_i - \Delta y_i^+$ is negative. By the definition of Δx_i, Δx_i is given by $l_i - \Delta y_i^+$ only when $l_i - \Delta y_i^+ > \mu_i$. Now that $l_i - \Delta y_i^+$ is negative, we have $0 > l_i - \Delta y_i^+ > \mu_i$, which implies $|l_i - \Delta y_i^+| + \Delta y_i^+ < |\mu_i| + \Delta y_i^+$. Since $|\mu_i| + \Delta y_i^+ \leq M$ by Inequality (7), we have $|l_i - \Delta y_i^+| + \Delta y_i^+ < M$. Hence, $|\Delta x_i| + \Delta y_i^+ \leq M$.

Case 2 Suppose Δx_i is given by $-r_i + \Delta y_i^+$.

We can show that $|\Delta x_i| + \Delta y_i^+ \leq M$ similar to Case 1 above.

Case 3 Suppose Δx_i is given by μ_i.

Then, $|\Delta x_i| + \Delta y_i^+ \leq M$ by Inequality (7).

In all cases, we have $|\Delta x_i| + \Delta y_i^+ \leq M$. □

Theorem 2 and Corollary 1 together imply that there exists a feasible legalized placement if $l_i + r_i - 2\Delta y_i^+ \leq 0$ for all i and the optimal maximum displacement D is given by $M = \max\{\max_i \Delta y_i^+, l, r, \lceil\sigma/2\rceil\}$. Moreover, Theorems 1 and 2 suggest a simple linear-time algorithm to generate a legalized placement with optimal maximum displacement.

Algorithm 1

DAG-Based Maximum Displacement Minimization

Input: A global placement, the desired row assignment of cells and cell ordering.

Output: A legalized placement with optimal maximum displacement.

Build the horizontal constraint graph G.

From left-most node to right, compute ρ_i and l_i for each cell i.

From right-most node to left, compute θ_i and r_i for each cell i.

Compute Δx_i for each cell i.

Refer to the example in Fig. 2 again, we can now compute the values of $\rho_i, \theta_i, l_i, r_i, \mu_i$, and Δx_i for each cell as in Fig. 3(a). The legalized placement result after applying our algorithm is shown in Fig. 3(b).

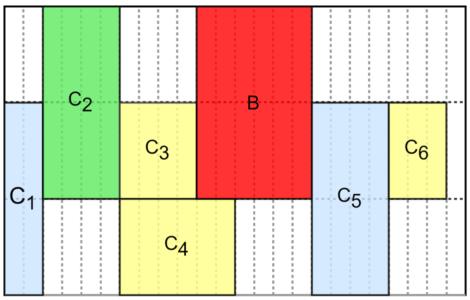
3.4 Cell Shifting Heuristic

After running the DAG-based linear-time legalization, we get the cell positions for optimal maximum displacement under the provided row assignment and cell ordering, but many cells are movable without harming the optimal maximum displacement. Thus, we propose an effective cell shifting heuristic to tune the cell positions for total displacement without increasing the maximum displacement as the last stage of our legalization. It is also linear in runtime.

The heuristic algorithm is shown in Algorithm 2. In lines 2-3, we will try to move a cell c to its MGL position if certain conditions are satisfied. The conditions are (i) the row assignment of c has not been changed by the iterative cell swapping algorithm in Section 3.2, (ii) c's MGL position is between the cells currently on its left and on its right, and (iii) moving c to its MGL position will not cause its displacement to exceed M. Empirically, about half of the cells can be placed at their MGL positions. Experiments show that performing

i	w_i	Δy_i^+	ρ_i	θ_i	l_i	r_i	$\lceil(\rho_i - \theta_i)/2\rceil$	$l_i - \Delta y_i^+$	$-r_i + \Delta y_i^+$
1	2	1	1	5	-4	6	-2	-5	-5*
2	4	4	4	4	0	8	0	-4	-4*
3	4	2	6	2	0	4	2	-2	-2*
4	6	3	3	3	-1	-2	0*	-4	5
5	4	2	4	3	4	-1	1	2*	3
6	3	1	6	1	5	-4	3	4*	5

(a)



(b)

Figure 3: (a) The computed values for each cell. The symbol "*" shows the term that determines Δx_i for cell i. (b) The legalization result after applying our algorithm.

lines 2-3 can improve the result and reduce the runtime of lines 12-17. Since the nodes connected by any edge on a directed path that determines the maximum displacement M in the horizontal constraint graph should be moved together, we merge them in lines 4-7. In addition, we also merge the nodes connected by any edge in the horizontal constraint graph with a non-negative edge length. In lines 7 and 10, we will relocate the nodes inside the same cluster together to keep the relative distances between nodes inside the same cluster. Each cluster is relocated to minimize the total displacement of the cells inside it such that no cell inside it will have a displacement exceeding M. Finally, we uncluster all cells and shift them individually to reduce the displacement of each cell.

4 EXPERIMENTAL RESULTS

We implemented the proposed algorithm in C++. The benchmarks are from ICCAD2017 contest problem C, and the experiments were performed on a 64-bit Ubuntu server with ThreadRipper 3970X and 256GB memory. To evaluate our method, we compare our results with [4] and [5]. We downloaded the source code of [4] from public repository, compiled and executed it on our machine. For fair comparison, all optional settings such as pin access optimization, fence region, and weighted displacement in [4] were disabled in our experiments. The code of [5] is not publicly available, so the reported results of [5] were directly taken from [5].

Table 1 shows the detail information for the benchmarks. "nxH" is the percentage of cells with height equal to n times of the design row-height. "#M" means the number of macros which cannot be moved and "Util." shows the utilization of a design.

Algorithm 2 Cell Shifting Heuristic

Input: A legalized placement with maximum cell displacement M.
Output: A legalized placement with maximum displacement M and improved total displacement.

1: Build horizontal constraint graph G
2: **for** each cell c in G in topological order **do**
3: Move c to its MGL position if it is feasible
4: **for** each edge $e = (c_i, c_j)$ in G **do**
5: **if** e is globally critical or $len(e) \geq 0$ **then**
6: Put c_i and c_j into same cluster
7: Horizontally shift each cluster to minimize total displacement while keeping displacement of any cell inside the cluster $\leq M$
8: **while** there are overlapping clusters **do**
9: Merge overlapping clusters to form a new cluster
10: Horizontally shift the new cluster formed to minimize total displacement while keeping displacement of any cell inside the cluster $\leq M$
11: Uncluster all cells
12: **repeat**
13: **for** each cell c in G in topological order **do**
14: Shift c to the left as much as possible to reduce its displacement
15: **for** each cell c in G in reverse topological order **do**
16: Shift c to the right as much as possible to reduce its displacement
17: **until** no more reduction on total displacement

Table 1: Benchmark Information.

Benchmarks	% cell types				#M	Util. (%)
	1xH	2xH	3xH	4xH		
des_perf_1	100.0	0.00	0.00	0.00	0	90.64
des_perf_a_md1	95.66	4.34	0.00	0.00	4	55.11
des_perf_a_md2	96.99	1.00	1.00	1.00	4	55.92
des_perf_b_md1	94.80	5.20	0.00	0.00	0	54.98
des_perf_b_md2	90.47	6.02	2.01	1.50	0	64.69
edit_dist_1_md1	90.31	6.12	2.04	1.53	0	67.47
edit_dist_a_md2	90.31	6.12	2.04	1.53	6	59.42
edit_dist_a_md3	93.88	2.04	2.04	2.04	6	57.22
fft_2_md2	89.62	6.56	2.18	1.64	0	83.12
fft_a_md2	89.57	6.59	2.19	1.65	6	32.41
fft_a_md3	93.42	2.19	2.19	2.19	6	31.24
pci_bridge32_a_md1	90.39	6.07	2.02	1.52	4	49.57
pci_bridge32_a_md2	85.51	7.08	4.04	3.37	4	57.72
pci_bridge32_b_md1	90.38	6.07	2.02	1.52	6	28.68
pci_bridge32_b_md2	97.97	1.01	1.01	1.01	6	19.72
pci_bridge32_b_md3	94.94	1.01	2.02	2.02	6	23.98

Table 2 shows the comparison of results by our method and those by [4] and [5]. "ΔHPWL" reports the increase in half-parameter wire length from the global placement after legalization. The column "Avg. Disp" reports the average cell displacement and "Max Disp." reports the maximum cell displacement for each design, both of them are measured in sites. Compared with [4], our algorithm

Table 2: Comparison with [4] and [5].

Benchmarks	ΔHPWL(%)			Avg. Disp. (sites)			Max Disp. (sites)		
	[4]	[5]	Ours	[4]	[5]	Ours	[4]	[5]	Ours
des_perf_1	5.86	6.66	6.01	6.52	6.97	6.81	42.07	48.95	38.49
des_perf_a_md1	1.65	2.48	1.62	5.60	5.94	5.61	607.30	607.30	607.30
des_perf_a_md2	1.55	2.51	1.53	5.48	5.93	5.50	480.55	403.86	480.55
des_perf_b_md1	1.72	1.52	1.69	4.55	4.77	4.58	42.53	38.45	30.27
des_perf_b_md2	1.51	1.72	1.47	4.95	5.25	4.97	59.39	39.76	30.62
edit_dist_1_md1	1.39	0.14	1.36	5.54	5.79	5.45	58.16	95.45	52.84
edit_dist_a_md2	0.80	1.01	0.79	5.18	5.51	5.16	164.00	164.00	164.00
edit_dist_a_md3	1.30	1.48	1.32	6.92	7.08	7.56	233.00	233.00	233.00
fft_2_md2	13.84	8.78	13.79	7.72	7.54	8.49	54.40	73.6	45.01
fft_a_md2	1.07	0.95	1.04	4.62	4.86	4.58	343.48	345.5	343.48
fft_a_md3	1.11	1.08	1.09	4.35	4.55	4.31	109.62	109.62	109.62
pci_bridge32_a_md1	2.49	3.38	2.45	5.31	5.64	5.30	63.76	63.76	63.76
pci_bridge32_a_md2	3.47	4.38	3.46	6.86	7.14	6.89	121.35	121.35	121.35
pci_bridge32_b_md1	1.73	2.26	1.64	5.58	6.01	5.54	340.30	332.71	332.72
pci_bridge32_b_md2	2.33	2.53	2.30	5.23	5.53	5.20	452.09	430.04	452.09
pci_bridge32_b_md3	2.57	3.17	2.54	5.69	6.1	5.68	476.91	398.57	476.91
Norm. Average	1.01	1.11	1	0.989	1.041	1	1.111	1.112	1

achieves 1% smaller ΔHPWL, 11.1% smaller maximum displacement, while having 1.1% larger average displacement. Compared with [5], our algorithm achieves 11% smaller ΔHPWL, 4.1% smaller average displacement, and 11.2% smaller maximum displacement. We note that [5] obtained a smaller maximum displacement than ours in a few designs due to the difference in row assignment and cell order. If we use the same row assignment and cell order as [5], then the maximum displacement obtained by our DAG-based legalization is guaranteed to be as good as [5].

Our proposed approach is highly efficient. We compare our runtimes with those by [4] on our machine. Both our approach and [4] use MGL as a preprocessing stage. Column 2 of Table 3 shows the runtime of MGL for each benchmark. Column 3 shows the runtime of the subsequent stages of [4]. Column 4 shows the runtime of the subsequent stages of our approach. Our subsequent stages are about 60.7% faster on average.

Table 3: Runtime comparison. The runtimes are measured in seconds.

Benchmarks	MGL	Remaining stages	
		[4]	Ours
des_perf_1	0.915	2.126	0.889
des_perf_a_md1	0.780	0.578	0.429
des_perf_a_md2	0.718	0.735	0.493
des_perf_b_md1	0.450	0.721	1.029
des_perf_b_md2	0.504	0.710	1.234
edit_dist_1_md1	0.495	0.787	1.624
edit_dist_a_md2	0.570	0.768	0.760
edit_dist_a_md3	0.873	1.654	1.310
fft_2_md2	0.133	0.318	0.287
fft_a_md2	0.089	0.122	0.114
fft_a_md3	0.073	0.113	0.107
pci_bridge32_a_md1	0.124	0.174	0.072
pci_bridge32_a_md2	0.144	0.196	0.106
pci_bridge32_b_md1	0.180	0.107	0.097
pci_bridge32_b_md2	0.125	0.193	0.049
pci_bridge32_b_md3	0.130	0.226	0.058
Norm. Average	1	1.607	1

5 CONCLUSION

In this paper, we proposed a highly efficient linear-time mixed-cell-height legalization approach that optimizes both the total cell displacement and the maximum cell displacement. Experimental results showed that our algorithm can produce high quality legalization solutions with 11% smaller maximum displacement on average than the state-of-the-art works with similar average displacement.

REFERENCES

[1] Jianli Chen, Ziran Zhu, Wenxing Zhu, and Yao-Wen Chang. 2017. Toward optimal legalization for mixed-cell-height circuit designs. In *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*. 1–6.

[2] Wing-Kai Chow, Chak-Wa Pui, and Evangeline F. Y. Young. 2016. Legalization algorithm for multiple-row height standard cell design. In *2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC)*. 1–6.

[3] Nima Karimpour Darav, Ismail S. Bustany, Andrew Kennings, and Laleh Behjat. 2017. A Fast, Robust Network Flow-Based Standard-Cell Legalization Method for Minimizing Maximum Movement. In *Proceedings of the 2017 ACM on International Symposium on Physical Design (ISPD '17)*. 141–148.

[4] Haocheng Li, Wing-Kai Chow, Gengjie Chen, Bei Yu, and Evangeline F.Y. Young. 2021. Pin-Accessible Legalization for Mixed-Cell-Height Circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2021), 1–1.

[5] 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 2019 International Symposium on Physical Design (ISPD '19)*. 27–34.

[6] Yibo Lin, Bei Yu, Xiaoqing Xu, Jhih-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 *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. 1–8.

[7] C. Wang, Y. Wu, J. Chen, Y.-W. Chang, S. Kuo, W. Zhu, and G. Fan. 2017. An Effective Legalization Algorithm for Mixed-Cell-Height Standard Cells. In *Proc. of Asia and South Pacific Design Automation Conference*. 450–455.

[8] G. Wu and C. Chu. 2015. 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 (2015).

[9] 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 *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. 1–8.