Detailed-Routability-Driven Analytical Placement for Mixed-Size Designs with Technology and Region Constraints

Chau-Chin Huang[1], Hsin-Ying Lee[1], Bo-Qiao Lin[1], Sheng-Wei Yang[1], Chin-Hao Chang[1], Szu-To Chen[1], and Yao-Wen Chang[1,2]
[1]Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 106, Taiwan
[2]Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan
{wlkb83, hylee, bqlin, swyang, chchang, stchen}@eda.ee.ntu.edu.tw; ywchang@ntu.edu.tw

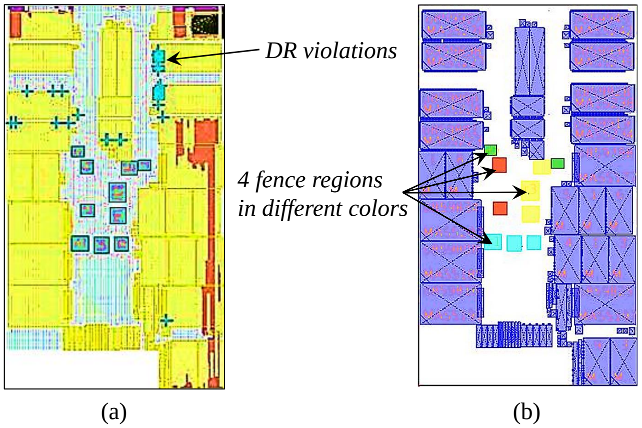
***Abstract*—A placer without considering modern technology and region constraints could generate solutions with irresolvable detailed-routing violations or even illegal solutions. This paper presents a high-quality placement algorithm to satisfy technology and region constraints and optimize detailed-routing routability with three major techniques: (1) a clustering algorithm followed by two-round quadratic placement to obtain an initial placement satisfying region constraints, (2) an analytical placement algorithm with new wirelength and density models to consider region constraints, and (3) a legalization algorithm that preserves the solution quality of global placement while satisfying technology/region constraints. Compared with the winning teams of the ISPD 2015 Blockage-Aware Detailed Routing-Driven Placement Contest, our placer achieves the best overall score and detailed-routing results.**

I. Introduction

With ever increasing requirements, many new rules are introduced to ensure printable GDSII masks for nanometer circuit designs. A placer without considering design rules may generate solutions with severe detailed-routing (DR) violations (as shown in Figure 1(b)). Fortunately, some design rules can be formulated as *technology constraints* and handled during the placement stage [14]. For example, an *edge-type* design rule describes a minimum spacing for two neighboring standard cells [1]. A violation of the edge-type rule may further cause detailed-routing violations [19]. Because these detailed-routing problems can be handled during placement, it is desirable to consider technology constraints for modern placers.

On the other hand, various predesigned features (e.g., power domains, datapaths) are introduced to achieve aggressive performance. These predesigned features impose additional constraints on modern placement. For example, designers may specify a region for placing cells belonging to a certain datapath to improve circuit performance [8]. A placement algorithm without considering these features may produce illegal placement or inferior solutions. As a result, it is important to handle design features at the placement stage. To encourage the research along this direction, ISPD 2015 held a placement contest [2] formulating these features as *(fence) region constraints*: a fence region is a placeable region, where cells assigned to a fence region (i.e., fence cells) must be placed inside this region, and cells not assigned to a fence region must be placed outside its boundary [4]. Moreover, a fence region may consist of multiple spatially disjoint rectangular subregions. Figure 1(b) shows a floorplan with four fence regions in different colors. The blue fence region consists of three disconnected subregions. Because the fence region constraint is a hard constraint, cells assigned to a disconnected fence region (e.g., the blue region) must be placed inside one of the corresponding subregions. Therefore, it is necessary to develop a placer to generate solutions satisfying region constraints.

As a complicated optimization problem, modern placement is often solved in a divide-and-conquer manner with three stages: (1) global placement, (2) legalization, and (3) detailed placement. Global placement distributes cells evenly and finds the desired position for each cell to optimize a predefined cost metric (e.g., wirelength, routability). Then legalization aligns cells to sites without overlap. Finally, detailed placement refines the placement solution. Because each stage is important in determining the final placement quality, detailed-routing routability (*detailed routability* for short) with technology and region constraints should be considered in the three stages.



(a) (b)

Fig. 1. (a) Routing result with detailed-routing (DR) violations; (b) floorplan with four disconnected fence regions (red, green, blue, and yellow) from the ISPD'15 contest [2].

Existing detailed-routability-driven placers considered technology constraints in various ways. Kennings et al. proposed a conservative approach in legalization to avoid detailed-routing violations [14]. By this conservative approach, pins and cells cannot be placed onto routing blockages. The conservative approach treats pre-routed wires as additional placement blockages. As a result, cells are prevented from being placed below such blockages, and the corresponding routing violations are strictly avoided. Huang et al. presented a whitespace allocation technique to minimize pin-short violations during legalization [12]. Recently, Wang et al. proposed a detailed placement algorithm to reduce cell-spacing violations [18]. The detailed placement algorithm consists of cell inflation followed by cell flipping.

Region constraints are important design features for modern placement. However, most of the published detailed-routability-driven works do not consider the region constraints. GORDIAN, a classic quadratic placer, could be extended to model region constraints into linear constraints of a quadratic programming problem and solve the problem optimally [15]. However, the solution quality of such a quadratic placer may be limited because of its less accurate quadratic wirelength metric and less effective partitioning-based framework.

This work was partially supported by Genesys Logic, IBM, MediaTek, TSMC, Academia Sinica, MOST of Taiwan under Grant NSC 102-2221-E-002-235-MY3, NSC102-2923-E-002-006-MY3, MOST 103-2221-E-002-259-MY3, and NTU under Grant NTU-ERP-104R8951

A. Our Contributions

In this paper, we propose an analytical placement framework to consider technology and region constraints while optimizing detailed routability. The main contributions of this paper are summarized as follows:

- A best-choice clustering algorithm is proposed to strictly avoid clustering cells belonging to different fence regions. We further show that the algorithm is effective by comparing it with another fence-region-aware clustering technique.
- A two-round quadratic placement approach is presented to generate an initial solution satisfying region constraints. The approach pulls cells toward the assigned regions effectively by introducing a limited number of anchors.
- A whitespace allocation technique for preplaced pins is presented to effectively improve detailed routability.
- A wirelength model and a density model are proposed to consider region constraints while minimizing cell overlaps. To the best of our knowledge, this is the first work to place all cells (including fence cells) simultaneously with an analytical framework.
- An Abacus-based legalization algorithm which integrates technology constraints into a quadratic programming is presented. We further prove that the optimality for the quadratic programming is preserved after the integration.
- Experimental results show that our algorithm is effective. Compared with the winning teams of the ISPD 2015 Blockage-Aware Detailed Routing-Driven Placement Contest, the proposed algorithm achieves the best overall score and detailed-routing results.

The remainder of this paper is organized as follows. Section II formulates the detailed-routability-driven placement problem and introduces the analytical placement framework. Section III details our algorithm. Section IV shows the experimental results. Section V concludes this paper.

II. Preliminaries

In this section, we describe the detailed-routability-driven placement problem and our analytical placement framework.

A. Problem Formulation

The circuit placement problem can be formulated as a hypergraph $H = (V, E)$ placement problem. Let vertices $V = \{v_1, v_2, ..., v_n\}$ represent cells and hyperedges $E = \{e_1, e_2, ..., e_m\}$ represent nets. Let x_i and y_i be the x and y coordinates of the center of the cell corresponding to vertex v_i, respectively. Blocks in a circuit can be categorized into two types: preplaced macros and movable cells. Preplaced macros have fixed x and y coordinates and cannot be moved. We intend to determine the desired position of each movable cell so that the total detailed-routing violations and detailed-routing wirelength are minimized, and technology and region constraints are satisfied.

B. Our Analytical Placement Framework

For a placement problem, global placement is considered the most critical stage because it has the most significant impact on the overall placement quality. We adopt a multilevel framework which consists of a coarsening stage and an uncoarsening stage for global placement to achieve a better trade-off between scalability and quality. During the coarsening stage, we cluster movable cells to reduce the placement problem size. After clustering, an initial placement for the coarsest level is generated by minimizing the quadratic wirelength. Then, we solve the placement problem from the coarsest level to the finest level. The placement for the current level provides an initial placement for the next level.

After the placement region is divided into uniform non-overlapping bins, the global placement problem can be formulated as a constrained minimization problem:

$$\begin{array}{ll} \min & W(\mathbf{x}, \mathbf{y}) \\ \text{s.t.} & D_b(\mathbf{x}, \mathbf{y}) \leq M_b, \quad \text{for each bin } b, \end{array} \tag{1}$$

where $W(\mathbf{x}, \mathbf{y})$ is the wirelength function, $D_b(\mathbf{x}, \mathbf{y})$ is the potential function that represents the total area of cells in bin b, and M_b is the maximum allowable area of circuit cells in bin b. M_b can be computed by $M_b = t_{density}(w_b h_b)$, where $t_{density}$ is a user-specified target density value for each bin, w_b (h_b) is the width (height) of bin b, and M_b is a fixed value as long as all preplaced macro positions are given and the bin size is determined.

During placement, the wirelength $W(\mathbf{x}, \mathbf{y})$ is usually defined as the total half-perimeter wirelength (HPWL). Because HPWL is neither smooth nor differentiable, it is hard to minimize HPWL directly with an analytical framework. As a result, we use the log-sum-exp (LSE) wirelength model [16] as follows:

$$\gamma \sum_{e \in E} \left(\begin{array}{c} \ln \sum_{v_i \in e} \exp\left(\frac{x_i}{\gamma}\right) + \ln \sum_{v_i \in e} \exp\left(\frac{-x_i}{\gamma}\right) \\ \ln \sum_{v_i \in e} \exp\left(\frac{y_i}{\gamma}\right) + \ln \sum_{v_i \in e} \exp\left(\frac{-y_i}{\gamma}\right) \end{array} \right). \tag{2}$$

When γ approaches zero, the LSE wirelength is close to the HPWL [16].

The potential function is defined as

$$D_b(\mathbf{x}, \mathbf{y}) = \sum_{v \in V} P_x(b, v) P_y(b, v), \tag{3}$$

where P_x and P_y are the overlap functions of bin b and cell v along the x and y directions. Because $D_b(\mathbf{x}, \mathbf{y})$ is neither smooth nor differentiable, the bell-shaped function [6], [7], [13] is used to smooth the density potential for each cell.

Equation (1) can be solved by the quadratic penalty method, implying that we solve a sequence of unconstrained minimization problems of the form

$$\min \quad \hat{W}(\mathbf{x}, \mathbf{y}) + \lambda \sum_b (\max(\hat{D}_b(\mathbf{x}, \mathbf{y}) - M_b, 0))^2, \tag{4}$$

with increasing λ's, where $\hat{W}(\mathbf{x}, \mathbf{y})$ and $\hat{D}_b(\mathbf{x}, \mathbf{y})$ are the smoothed wirelength and density functions, respectively. The solution of the previous problem is used as the initial solution for the next one. We solve the unconstrained problem in Equation (4) using a nonlinear conjugate gradient (CG) method.

III. The Proposed Algorithm

The proposed algorithm considers technology and region constraints while optimizing routability with the following techniques: (1) fence-region-aware clustering, (2) two-round quadratic placement to evenly distribute cells to corresponding fence regions, (3) whitespace allocation for fixed pins to improve detailed routability, (4) fence-region-aware wirelength and density models to simultaneously optimize routability and satisfy region constraints, (5) legalization and detailed placement considering technology constraints. Figure 2 summarizes the overall flow of our proposed algorithm. The techniques are detailed in the following subsections.

A. Fence-Region-Aware Clustering

Clustering is critical for most modern analytical placers to improve the final solution quality and speed up the overall running time [5], [6], [7]. Our clustering algorithm is based on the best-choice algorithm [3] to improve scalability. In the original best-choice clustering algorithm, the clustering score $d(u, v)$ for two cells u and v is as follows:

$$d(u, v) = \sum_{e \in E_{uv}} \frac{w_e}{a_u + a_v}, \tag{5}$$

where E_{uv} is the set of hyperedges incident on both u and v, w_e is the degree of hyperedge e, a_u is the area of u, and a_v is the area of v. By iteratively choosing the globally closest pair of cells with the highest clustering score, the best-choice algorithm can produce high-quality clustering results for subsequent placements.

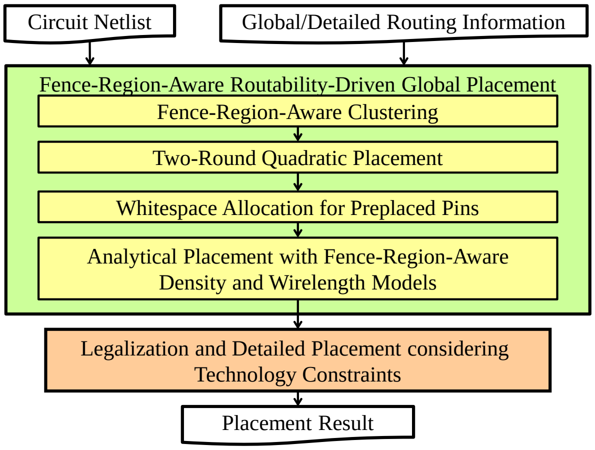

Circuit Netlist
Global/Detailed Routing Information
Fence-Region-Aware Routability-Driven Global Placement
Fence-Region-Aware Clustering
Two-Round Quadratic Placement
Whitespace Allocation for Preplaced Pins
Analytical Placement with Fence-Region-Aware Density and Wirelength Models
Legalization and Detailed Placement considering Technology Constraints
Placement Result

Fig. 2. Overview of our algorithm.

The formulation in Equation (5), however, does not consider region constraints, and thus may lead to inferior clustering solutions for the addressed problem in this paper. For example, two cells which are respectively assigned to different fence regions should not be clustered together because they must be placed inside two separate regions. Therefore, we modify the fence-region-aware clustering score $d_f(u, v)$ for two cells u and v as follows:

$$d_f(u,v) = \sum_{e \in E_{uv}} f_{uv} \cdot \frac{w_e}{a_u + a_v}, \tag{6}$$

where f_{uv} is a 0-1 function; $f_{uv} = 1$ when u and v are both assigned to the same fence region, and $f_{uv} = 0$ otherwise. As a result, clustering two cells assigned to different fence regions is strictly forbidden.

B. Two-Round Quadratic Placement

After clustering, two-round quadratic placement (QP) considering fence-region constraints is adopted to generate an initial placement solution. The objective function Φ of the quadratic placement is as follows:

$$\Phi(\mathbf{x},\mathbf{y}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + d_x^T\mathbf{x} + \frac{1}{2}\mathbf{y}^T Q\mathbf{y} + d_y^T\mathbf{y}, \tag{7}$$

where $\mathbf{x}$ and $\mathbf{y}$ are the coordinate vectors of movable cells, Q is the connectivity matrix, and d_x and d_y are the vectors originating from fixed pins to the relative pin coordinates. Φ can be optimally minimized by solving the equation $\nabla\Phi(x, y) = 0$.

Minimizing Equation (7) directly, however, may result in a placement solution in which fence cells are far away from their corresponding fence regions (see Figure 3(a)). Moreover, an initial solution without considering fence region constraints might lead to longer convergence time in the following analytical global placement stage. Therefore, we propose a two-round quadratic placement algorithm to spread fence cells to the corresponding regions more closely by the following three steps: (1) we first apply a first-round QP to obtain a solution (see Figure 3(a)). (2) According to the solution, we add anchors to each fence region and set pseudo nets for corresponding fence cells. As shown in Figure 3(b), two anchors (bottom-left and top-right) are added and two pseudo nets are connected to fence cell A. As a result, placing A into the region could result in a smaller quadratic wirelength. (3) We then apply a second-round QP with the pseudo nets to obtain a solution considering region constraints (see Figure 3(d)).

Slicing fence regions into small rectangular subregions and adding anchors to the subregions would significantly help spread fence cells toward the corresponding regions (see Figures 3(c) and (d)). We slice a fence region into subregions by slicing each rectangle of the fence region[1] as follows: For a rectangle, starting from the bottom-left corner, we draw horizontal and vertical cutlines subsequently for every fixed distance d_h (see Figure 3(d)). In our experiments, we set d_h to ten times of the row height to evenly distribute fence cells among the rectangle. Further, to avoid producing a subregion with an extremely small area, we discard the last vertical or horizontal cutlines if the width or height of a rectangle cannot be divided evenly by d_h. We then detail the adding process as follows: Based on the first-round QP placement result, sliced subregions are first sorted in non-increasing order by their distances to the gravity center of its corresponding fence cells. The distance between a subregion and a gravity center is defined as the minimum Euclidian distance for any points within the subregion to the gravity center. In this order, we iteratively identify a fixed number of closest cells for a subregion, and add anchors and pseudo nets to the cells and the subregion. For each subregion and cell, the assignment of anchors and pseudo nets are shown in Figure 3(b). Therefore, this adding process helps spread fence cells more evenly in each fence region (Figure 3(d)).

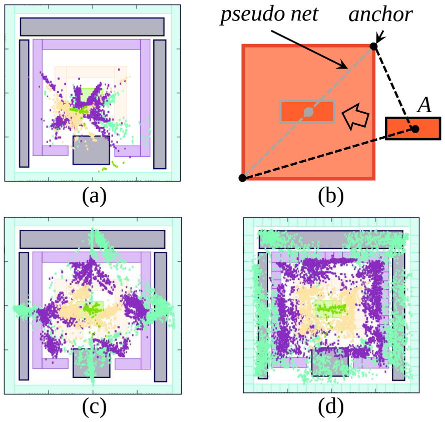


Fig. 3. Two-round quadratic placement. Fence cells and their corresponding fence regions share the same colors. (a) First quadratic placement result; (b) added pseudo nets pull cell A into the region; (c) and (d): respective second QP results without and with slicing fence regions.

C. Whitespace Allocation for Preplaced Pins

For designs with relatively few metal layers, previous routability-driven placers observed that preplaced pins significantly affect detailed routability. For example, a placement without considering fixed pins (Figure 4(a)) may lead to an inferior detailed-routing solution (see the detailed-routing violations in red crosses in Figure 4(c)). Consequently, we apply a simple, yet effective whitespace allocation technique for fixed pins to reduce detailed-routing violations: we inflate fixed pins with a ratio r and treat these pins as placement blockages during analytical global placement. The inflation ratio r is defined as

$$r = \begin{cases} \delta \cdot h_r, & \text{if } n \geq \epsilon, \\ h_r, & \text{otherwise,} \end{cases} \tag{8}$$

where δ and ϵ are two user-specified parameters, and n and h_r denote the number of movable cells and the placement row height, respectively. Our experiments showed that setting $\delta = 3$ and $\epsilon = 600000$ produces satisfactory results.[2] That is, the maximum allowable area of cells in each bin (i.e., M_b in Equation (4)) is reduced according to the corresponding fixed pin density in the bin for the optimization problem. As a result, cells

[1] Every fence region consists of rectangle(s) in the LEF/DEF format [1].

[2] The setting of ϵ distinguishes two suites of the ISPD 2015 contest [4]. The two suites are based on different technology nodes and thus possess a clear gap in the numbers of the corresponding movable cells (see Table I).

tend to leave bins with many fixed pins. On the other hand, because we do not actually remove placeable row sites for the whitespace allocation, cells are still allowed to be placed close to fixed pins during legalization and detailed placement if the cells have high connectivity with the fixed pins. Figures 4(b) and (d) show placement and routing results with our whitespace allocation, respectively.

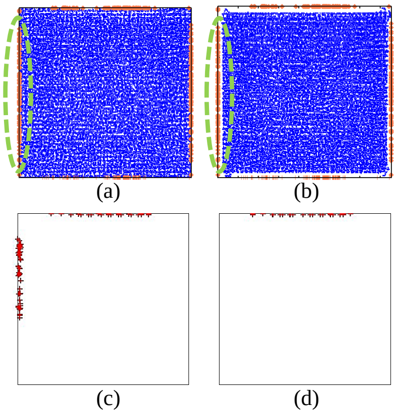


Fig. 4. Our whitespace allocation method for fixed pins for circuit mgc_fft_1 in [4]. (a) and (b): Respective placements without and with the whitespace allocations, where fixed pins along the left boundaries are in green circles; (c) and (d): Respective DR results for the placements in (a) and (b).

D. Fence-Region-Aware Wirelength Model

In global placement, we introduce anchors and pseudo nets for each fence region to pull fence cells to the region. However, an intuitive anchor assignment method may result in an inferior solution. As shown in Figures 5(a) and (c), a rectangular fence region ranges from $(-50,-50)$ to $(50,50)$ and an anchor is assigned to the center of this region. As a cell moves from the center to the corners of the region, the LSE wirelength W_{LSE} varies from 0 to approximately 100. Therefore, a cell assigned to this fence region tends to move toward the center even it is already in this region, which is not desirable.

On the other hand, assigning two anchors to the bottom-left and top-right corners for each subregion can avoid this problem. As shown in Figures 5(b) and (d), W_{LSE} is approximately 200 wherever a cell is located in this region. As the result, pseudo nets have no influence on cells when they are in this region. Both methods can pull cells toward their corresponding regions. However, when cells are already in their corresponding regions, only the method with two anchors can avoid further interference from pseudo nets. Therefore, we resort to the two-anchor method shown in Figure 5(b).

E. Fence-Region-Aware Density Model

Although the wirelength model can effectively pull a fence cell into a corresponding region, we still need a method to pull other cells out of a fence region. Rather than adopting an anchor assignment technique, we propose a new density model which considers fence region constraints and minimizes cell overlaps simultaneously. In the new density model, fence regions are regarded as preplaced macros for cells not belonging to the regions. Figure 6 gives the corresponding density map for a cell not belonging to fence regions. The corresponding unconstrained minimization problem is defined as

$$\min \quad \hat{W}(\mathbf{x},\mathbf{y}) + \lambda \sum_{k+1}\sum_{b}(\max(\hat{D}_{k,b}(\mathbf{x},\mathbf{y}) - M_{k,b},0))^2, \qquad (9)$$

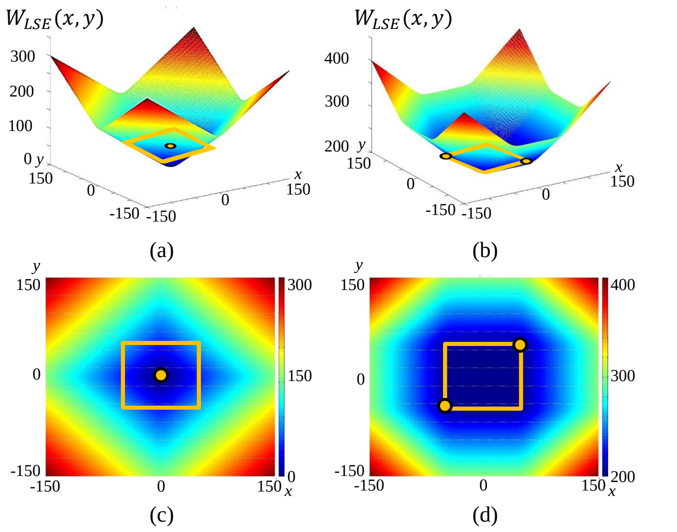


Fig. 5. Resulting LSE wirelength [16] with different anchor assignments for a fence region (ranging from $(-50,-50)$ to $(50,50)$). $W_{LSE}(x,y)$ in (a) and (b) denotes the wirelength of a net connecting to the anchor(s) and a cell at (x,y). (a) Assigning an anchor at the center of the region; (b) Assigning anchors at two corners of the region; (c) and (d): Planar maps for the anchor assignments in (a) and (b), respectively.

where k is the number of fence regions, and $D_{k,b}$ and $M_{k,b}$ are the potential and maximum allowable area for bin b of the k-th density map, respectively. D_b is defined as

$$D_{k,b}(\mathbf{x},\mathbf{y}) = \sum_{v\in V} f_{k,v}\cdot P_x(b,v)P_y(b,v), \qquad (10)$$

where $f_{k,v}$ is a 0-1 function which equals one only when cell v is assigned to the k-th fence region. This function implies that cells belonging to different fence regions consider different density maps. Further, because each cell belongs to either one fence region or does not belong to any fence region, for each cell, only one density cost needs to be computed. Therefore, the update for all the values and gradients of potential functions in Equation (9) is exactly the same as that in Equation (4); this is, no computation overhead is incurred.

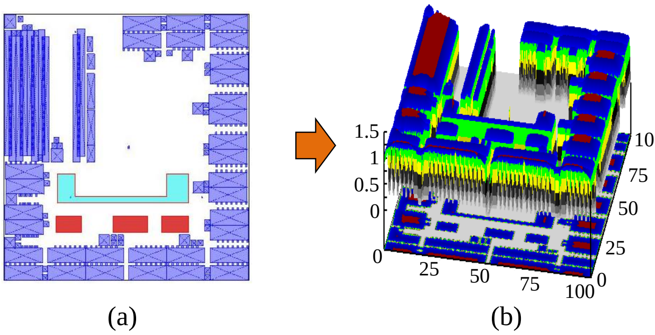


Fig. 6. A cell not belonging to a fence region regards the region as a preplaced macro. (a) Floorplan with preplaced macros and fence regions for the circuit mgc_superblue16_a; (b) The corresponding smoothed density map with macros and fence regions for cells not belonging to fence regions.

F. Legalization and Detailed Placement Considering Technology Constraints

This paper extends the Abacus legalization [17] to preserve the solution quality in global placement while satisfying technology constraints. In

Abacus, cells are first sorted by their x-positions and then legalized into row sites one by one. Given a row with N_r cells, which are already aligned in the row and are indexed from 1 to N_r (e.g., from left to right), the minimum displacement for the cells is guaranteed by solving the following quadratic programming optimally as follows:

$$\begin{aligned} \min \quad & \sum_{i=1}^{N_r} (x_i - x_i')^2 \\ \text{s.t.} \quad & x_i - x_{i-1} \geq w_{i,i-1}, \quad \text{for } i = 2, 3, \ldots, N_r, \end{aligned} \tag{11}$$

where x_i and x_i' are respectively the left-edge x-positions of cell v_i before and after the legalization, and $w_{i,i-1}$ is the sum of the width of v_{i-1} and the minimum spacing for two neighboring cells v_{i-1} and v_i considering edge-type spacing violations. Because every cell position for a set of abutting cells could be determined by the position of the left-most cell, Equation (11) can be solved optimally by replacing the inequalities with equalities and then setting the objective derivative to zero with respect to the left-most position x_1. As a result, optimal x-positions with a minimum displacement are determined. Note that a main difference of Equation (11) and the original formulation in Abacus [17] is the lower bound of each linear inequality constraint, where the value is set to the width of cell v_{i-1} in Abacus. Because of this difference, we can remove all cell overlaps with a minimal total displacement and satisfy the region constraints.

In our detailed placement, routability-driven cell matching and cell swapping in [11], [9], [10] are adopted to minimize routing congestions. The cell-matching technique selects only independent cells (i.e., cells with no connections to other selected cells) to ensure a correct wirelength evaluation. We further apply cell swapping to improve routability. The cell-swapping method enumerates all possible permutations for k adjacent cells in a single row to obtain minimized routing congestions. Here, k is usually a small constant because we refine our solution only locally during detailed placement.

IV. Experimental Results

To evaluate the proposed placement algorithm, we conducted experiments on the benchmarks (*including hidden cases*) of the ISPD 2015 Blockage-Aware Detailed Routing-Driven Placement Contest [4]. Table I gives the benchmark statistics. We implemented our algorithm in the C++ programming language, and performed all the experiments on the same Linux workstation with eight Intel Xeon 2.93GHz CPUs with 8GB memory. The Mentor Graphics Olympus global and detailed routers were used for routing congestion analysis. Note that *the reported routing work was performed by the contest organizers from Mentor Graphics*. Because no routing runtime was reported from the contest organizers, we can only report the placement runtime.

Table II summarizes the detailed-routing results of our algorithm and the top three teams (Team7, Team1, and Team4) of the contest [2]. The following results are presented in the table: (1) f_{of} (total density overflows), (2) DR WL (detailed routed wirelength), and (3) DRC Viols (detailed-routing DRC violations). All the three results are provided by the contest organizers from Mentor Graphics [4]. Table III compares normalized detailed-routing results of our algorithm with those of the top three teams based on the following data [4]: (1) SDP, (2) SWL, (3) SDR, and (4) FINAL. The SDP score gives the scaled displacement score performed by the Mentor Graphics Olympus-SoC legalizer to fix defects in all submitted placement solutions at the contest [2]. The SWL score is the scaled detailed-routed wirelength. The SDR score is the scaled number of detailed-routing violations. FINAL is the sum of SDP, SWL, and SDR. A lower score indicates a better placement result. An unroutable placement receives the worst score of FINAL = 50. Note that the calculations of SDP, SWL, and SDR in this paper follow the way in [4] and use the results of our algorithm and the top three teams for normalization. In summary, compared with Team1, Team4, and Team7, our algorithm obtained 57.6%, 93.3%, and 29.1% smaller total score, respectively. The experimental results show that we achieve the best detailed-routing results, compared with the top three teams of the ISPD 2015 Contest.

V. Conclusions

This paper has presented a detailed-routability-driven analytical placement algorithm for modern mixed-size designs with technology and region constraints. The algorithm consists of (1) a fence-region-aware clustering technique followed by two-round quadratic placement to generate an initial placement satisfying region constraints, (2) an analytical placer with new wirelength and density models to consider region constraints and remove overlaps simultaneously, and (3) an Abacus-based legalizer which integrates technology constraints into quadratic programming to strictly avoid standard-cell spacing violations. Compared with the winning teams of the ISPD 2015 Blockage-Aware Detailed Routing-Driven Placement Contest, experimental results have shown that our algorithm achieves the best overall score and detailed-routing results.

Acknowledgements

The authors would like to thank Dr. Tung-Chieh Chen for his valuable suggestions and comments.

References

[1] *Cadence LEF/DEF 5.3 to 5.7 Exchange Format 2009*. www.si2.org/openeda.si2.org/projects/lefdef.

[2] *ISPD 2015 Blockage-Aware Detailed Routing-Driven Placement Contest*. http://www.ispd.cc/contests/15/ispd2015contest.html.

[3] C. Alpert, A. Kahng, G.-J. Nam, S. Reda, and P. G. Villarrubia. A semi-persistent clustering technique for VLSI circuit placement. In *Proc. of ISPD*, pages 200–207, 2005.

[4] I. S. Bustany, D. Chinnery, J. R. Shinnerl, and V. Yutsis. ISPD 2015 benchmarks with fence regions and routing blockages for detailed-routing-driven placement. In *Proc. of ISPD*, pages 157–164, 2015.

[5] T. F. Chan, J. Cong, and K. Sze. Multilevel generalized force-directed method for circuit placement. In *Proc. of ISPD*, pages 185–192, 2005.

[6] T.-C. Chen, Z.-W. Jiang, T.-C. Hsu, H.-C. Chen, and Y.-W. Chang. A high-quality analytical placer considering preplaced blocks and density constraints. In *Proc. of ICCAD*, pages 187–192, 2006.

[7] T.-C. Chen, Z.-W. Jiang, T.-C. Hsu, H.-C. Chen, and Y.-W. Chang. NTUplace3: An analytical placer for large-scale mixed-size designs with preplaced blocks and density constraints. *IEEE Tran. on CAD*, 27(7):1228–1240, July 2008.

[8] S. Chou, M.-K. Hsu, and Y.-W. Chang. Structure-aware placement for datapath-intensive circuit designs. In *Proc. of DAC*, pages 762–767, 2012.

[9] M.-K. Hsu, Y.-F. Chen, C.-C. Huang, T.-C. Chen, and Y.-W. Chang. Routability-driven placement for hierarchical mixed-size circuit designs. In *Proc. of DAC*, pages 1–6, 2013.

[10] M.-K. Hsu, Y.-F. Chen, C.-C. Huang, S. Chou, T.-H. Lin, T.-C. Chen, and Y.-W. Chang. NTUplace4h: A novel routability-driven placement algorithm for hierarchical mixed-size circuit designs. *IEEE Tran. on CAD*, 33(12):19141927, December 2014.

[11] M.-K. Hsu, S. Chou, T.-H. Lin, and Y.-W. Chang. Routability-driven analytical placement for mixed-size circuit designs. In *Proc. of ICCAD*, pages 80–84, 2011.

[12] C.-C. Huang, C.-H. Chiou, K.-H. Tseng, and Y.-W. Chang. Detailed-routing-driven analytical standard-cell placement. In *Proc. of ASPDAC*, pages 378–383. IEEE, 2015.

[13] A. B. Kahng and Q. Wang. Implementation and extensibility of an analytic placer. *IEEE Tran. on CAD*, 24(5):734–747, May 2005.

[14] A. Kennings, N. K. Darav, and L. Behjat. Detailed placement accounting for technology constraints. In *International Conference on Very Large Scale Integration (VLSI-SoC)*, pages 1–6. IEEE, 2014.

[15] J. M. Kleinhans, G. Sigl, F. M. Johannes, and K. J. Antreich. GORDIAN: VLSI placement by quadratic programming and slicing optimization. *IEEE Tran. on CAD*, 10(3):356–365, March 1991.

[16] W. C. Naylor, R. Donelly, and L. Sha. Non-linear optimization system and method for wire length and delay optimization for an automatic electric circuit placer. 2001.

[17] P. Spindler, U. Schlichtmann, and F. M. Johannes. Abacus: Fast legalization of standard cell circuits with minimal movement. In *Proc. of ISPD*, pages 47–53, 2008.

[18] C.-K. Wang, C.-C. Huang, S. S.-Y. Liu, C.-Y. Chin, S.-T. Hu, W.-C. Wu, and H.-M. Chen. Closing the gap between global and detailed placement: Techniques for improving routability. In *Proc. of ISPD*, pages 149–156. ACM, 2015.

[19] V. Yutsis, I. S. Bustany, D. Chinnery, J. Shinnerl, and W.-H. Liu. ISPD 2014 benchmarks with sub-45nm technology rules for detailed-routing-driven placement. In *Proc. of ISPD*, pages 161–168, 2014.

TABLE I

Benchmark characteristics. #macros, #cells, #nets, #fence regions, #I/Os, $util_{std}$, $util$, and ω denote the numbers of fixed macros, movable standard cells, nets, fence regions, fixed I/O pins, design utilization rates for standard cells, design utilization rates for standard cells and macros, and density limit, respectively [4].

Circuit	#macros	#cells	#nets	#fence regions	#I/Os	$util_{std}$	$util$	ω
mgc_des_perf_1	0	112644	112880	0	374	0.91	0.91	0.91
mgc_des_perf_a	4	108292	115187	4	374	0.43	0.72	0.43
mgc_des_perf_b	0	112679	112951	12	374	0.50	0.50	0.50
mgc_edit_dist_a	6	127414	134051	1	2574	0.46	0.62	0.46
mgc_fft_1	0	32281	33307	0	3010	0.84	0.84	0.84
mgc_fft_2	0	32281	33307	0	3010	0.50	0.50	0.65
mgc_fft_a	6	30625	32090	0	3010	0.25	0.74	0.50
mgc_fft_b	6	30625	32090	0	3010	0.28	0.74	0.60
mgc_matrix_mult_1	0	155325	158529	0	4802	0.80	0.80	0.80
mgc_matrix_mult_2	0	155325	158529	0	4802	0.80	0.80	0.80
mgc_matrix_mult_a	5	149650	154284	0	4802	0.45	0.77	0.60
mgc_matrix_mult_b	7	146435	151612	3	4802	0.34	0.73	0.60
mgc_matrix_mult_c	7	146435	151612	3	4802	0.33	0.73	0.60
mgc_pci_bridge32_a	4	29533	34058	3	361	0.38	0.41	0.38
mgc_pci_bridge32_b	6	28932	32546	3	361	0.14	0.51	0.14
mgc_superblue11_a	1458	925616	935613	4	27371	0.35	0.73	0.65
mgc_superblue12	89	1286948	1293413	0	5908	0.44	0.57	0.65
mgc_superblue16_a	419	680450	697303	2	17498	0.50	0.74	0.55
mgc_superblue14	340	612243	619697	0	21078	0.55	0.77	0.56
mgc_superblue19	286	506097	511606	0	15422	0.52	0.81	0.53

TABLE II

Resulting total density overflow (f_{of}; the smaller, the better), detailed-routing wirelength (DR WL), detailed-routing DRC violations (DRC Viols) of ours and the top three teams of the contest [2].

	Team1			Team4			Team7			Ours		
Circuit	f_{of}	DR WL	DRC Viols	f_{of}	DR WL	DRC Viols	f_{of}	DR WL	DRC Viols	f_{of}	DR WL	DRC Viols
mgc_des_perf_1	0.024	1.8	5093.4	0.014	1.91	1682.8	0.011	1.79	11.2	0.011	1.79	11.2
mgc_des_perf_a	0.159	N/A	N/A	0.220	3.09	2371.2	0.237	3.21	2571.4	0.237	3.21	2571.4
mgc_des_perf_b	0.200	2.08	0	0.224	2.23	0	0.251	2.69	0	0.251	2.69	0
mgc_edit_dist_a	0.167	N/A	N/A	0.189	N/A	N/A	0.163	N/A	N/A	0.163	N/A	N/A
mgc_fft_1	0.027	0.63	146.8	0.015	0.66	532.8	0.054	0.65	22.4	0.054	0.65	22.4
mgc_fft_2	0.001	0.62	57.6	0.011	0.66	61.4	0.000	0.76	51.8	0.000	0.76	51.8
mgc_fft_a	0.016	1.41	1798.8	0.093	1.26	664.8	0.107	1.27	888.2	0.107	1.27	888.2
mgc_fft_b	0.009	1.47	4558.8	0.081	1.34	4201.4	0.121	1.3	5418.2	0.121	1.3	5418.2
mgc_matrix_mult_1	0.039	2.95	221.8	0.076	3.07	7603	0.059	2.87	5.4	0.059	2.87	5.4
mgc_matrix_mult_2	0.031	3	1693	0.070	3.13	7952.6	0.019	2.92	123.6	0.019	2.92	123.6
mgc_matrix_mult_a	0.009	5.2	111.8	0.150	4.93	1713.8	0.017	4.66	7	0.017	4.66	7
mgc_matrix_mult_b	0.045	5.48	5143.8	0.089	4.73	9019.6	0.050	4.54	4139.2	0.050	4.54	4139.2
mgc_matrix_mult_c	0.024	5.65	1770.4	0.088	5.36	5758.6	0.534	N/A	N/A	0.039	4.49	1690.6
mgc_pci_bridge32_a	0.341	0.86	2254.6	0.424	0.84	260.8	0.349	0.88	2052.8	0.349	0.88	2052.8
mgc_pci_bridge32_b	0.307	1.19	6.4	0.453	0.93	0	0.556	0.87	3.2	0.556	0.87	3.2
mgc_superblue11_a	0.114	46.83	62.4	0.129	44.38	78.2	0.103	48.44	94.4	0.103	48.44	94.4
mgc_superblue12	0.109	56.29	225.4	0.071	N/A	N/A	0.026	53.74	95.4	0.026	53.74	95.4
mgc_superblue16_a	0.145	N/A	N/A	0.139	N/A	N/A	0.141	36.75	177.6	0.141	36.75	177.6
mgc_superblue14	0.199	31.61	0.4	0.126	31.62	1.4	0.193	32.07	0.8	0.193	32.07	0.8
mgc_superblue19	0.177	22.74	678.8	0.149	N/A	N/A	0.133	N/A	N/A	0.164	24.36	1212.8

TABLE III

Resulting scaled displacement score (SDP), scaled detailed routed wirelength score (SWL), scaled detailed-routing violation score (SDR), total final scores (FINAL; the smaller, the better), and placement runtime (CPU_p) of ours and the top three teams of the contest [2].

	Team1				Team4				Team7				Ours				
Circuit	SDP	SWL	SDR	FINAL	SDP	SWL	SDR	FINAL	SDP	SWL	SDR	FINAL	SDP	SWL	SDR	FINAL	CPU_p (min)
mgc_des_perf_1	1.87	0.90	21.40	24.17	2.27	3.42	15.61	21.30	2.91	0.00	0.58	3.49	2.91	0.00	0.58	3.49	6.2
mgc_des_perf_a	0.00	NA	N/A	50.00	0.00	0.00	17.38	17.38	0.43	2.30	17.80	20.53	0.43	2.30	17.80	20.53	6.0
mgc_des_perf_b	0.00	0.00	0.00	0.00	0.00	2.81	0.00	2.81	0.00	10.47	0.00	10.47	0.00	10.47	0.00	10.47	7.7
mgc_edit_dist_a	0.00	25.00	25.00	50.00	0.00	25.00	25.00	50.00	0.51	25.00	25.00	50.00	0.51	25.00	25.00	50.00	6.8
mgc_fft_1	1.14	0.00	4.89	6.03	2.06	1.55	10.00	13.61	1.63	2.58	1.10	5.31	1.63	2.58	1.10	5.31	1.4
mgc_fft_2	0.00	0.00	2.46	2.46	1.26	2.59	2.59	6.44	0.00	7.75	2.26	10.01	0.00	7.75	2.26	10.01	1.8
mgc_fft_a	0.00	1.89	15.95	17.84	0.91	0.00	11.02	11.93	0.00	0.98	12.41	13.39	0.00	0.98	12.41	13.39	1.8
mgc_fft_b	0.00	1.18	20.81	21.99	1.02	0.00	20.38	21.40	0.00	0.30	21.73	22.03	0.00	0.30	21.73	22.03	1.8
mgc_matrix_mult_1	0.35	0.42	6.33	7.10	1.24	4.29	23.54	29.07	0.89	0.00	0.28	1.17	0.89	0.00	0.28	1.17	5.9
mgc_matrix_mult_2	0.32	1.86	15.64	17.82	1.24	5.93	23.78	30.95	1.17	0.00	4.36	5.53	1.17	0.00	4.36	5.53	6.2
mgc_matrix_mult_a	0.00	4.61	4.07	8.68	0.00	8.46	15.70	24.16	0.00	0.00	0.37	0.37	0.00	0.00	0.37	0.37	6.9
mgc_matrix_mult_b	0.00	8.98	21.45	30.43	0.00	3.59	24.45	28.04	0.00	0.00	20.30	20.30	0.00	0.00	20.30	20.30	5.5
mgc_matrix_mult_c	0.00	6.98	15.87	22.85	0.00	7.27	22.05	29.32	25.00	25.00	25.00	50.00	0.00	0.00	15.63	15.63	4.1
mgc_pci_bridge32_a	0.00	0.00	17.11	17.11	0.00	1.71	6.95	8.66	0.00	1.35	16.63	17.98	0.00	1.35	16.63	17.98	1.4
mgc_pci_bridge32_b	0.00	7.51	0.34	7.85	0.00	0.00	0.00	0.00	0.00	0.09	0.17	0.26	0.00	0.09	0.17	0.26	1.2
mgc_superblue11_a	0.00	1.77	2.63	4.40	0.00	0.00	3.13	3.13	0.00	2.86	3.60	6.46	0.00	2.86	3.60	6.46	293.4
mgc_superblue12	0.00	6.61	6.39	13.00	0.00	25.00	25.00	50.00	0.00	0.00	3.63	3.63	0.00	0.00	3.63	3.63	144.5
mgc_superblue16_a	0.00	25.00	25.00	50.00	0.00	25.00	25.00	50.00	0.00	0.00	5.53	5.53	0.00	0.00	5.53	5.53	85.7
mgc_superblue14	0.00	2.67	0.02	2.69	0.00	0.00	0.08	0.08	0.00	3.09	0.04	3.13	0.00	3.09	0.04	3.13	62.2
mgc_superblue19	0.00	0.00	11.12	11.12	0.00	25.00	25.00	50.00	0.00	25.00	25.00	50.00	0.00	2.73	13.95	16.68	68.3
Total Score	–	–	–	**365.55**	–	–	–	**448.27**	–	–	–	**299.57**	–	–	–	**231.88**	–