Master's Thesis

A Metaheuristic Method for Fast Multi-Deck Legalization

SangGi Do

Department of Electrical Engineering

Graduate School of UNIST

2018

A Metaheuristic Method for Fast Multi-Deck Legalization

SangGi Do

Department of Electrical Engineering

Graduate School of UNIST

A Metaheuristic Method for Fast Multi-Deck Legalization

A thesis
submitted to the Graduate School of UNIST
in partial fulfillment of the
requirements for the degree of
Master of Science

SangGi Do

12 / 14 / 2017

Approved by

Advisor
Seokhyeong Kang

A Metaheuristic Method for Fast Multi-Deck Legalization

SangGi Do

This certifies that the thesis of SangGi Do is approved.

12 / 14 / 2017

signature

Advisor: Seokhyeong Kang

signature

Jaehyouk Choi

signature

Seong-Jin Kim

Abstract

In the field of circuit design, decreasing the transistor size is getting harder and harder. Hence, improving the circuit performance also becoming difficult. For the better circuit performance, various technologies are being tired and multi-deck standard cell technology is one of them. The standard cell methodology is a fundamental structure of EDA (Electric Design Automation). Using the standard cell library, EDA tools can easily design, and optimize the physical design of chips.

In order to conventional standard cell, multi-deck standard cell occupies multiple rows on the chip. This multiple occupation increases complexity of the circuit physical design for EDA tools. Thus, legalization problem has become more challenging for the multi-deck standard cells. Recently, various multi-deck legalization methods are proposed because the conventional single-deck legalization method is not effective for multi-deck legalization. A state-of-the-arts legalization method is based on quadratic programming with the linear complementary problem(LCP). However, these previous researches can only cover the double-deck case because of runtime burden.

In this thesis, we propose the fast and enhanced the multi-deck standard cell legalization algorithm which can handle higher than double-deck standard cell cases. The proposed legalization method achieves the most fastest runtime result for the dominant number of benchmarks on ICCAD Contest 2017 [1] compared with Top 3 results.

Contents

List of Figures

Fig. 17 Raw score of ICCAD contest 2017 Benchmarks.

Fig. 18 Runtime of ICCAD contest 2017 Benchmarks.

Fig. 19 Score with runtime of ICCAD contest 2017 Benchmarks.

List of Tables

Acknowledgement

I would like to thank all of the people who helped make this thesis possible.

First, and foremost, I am thankful to my advisor, Professor Seokhyeong Kang, whose advice, teaching and support enabled me to complete my M.S. source and this thesis.

I certainly feel lucky to have worked with great students in Professor Kang's System-on-Chip Design Lab. I would like to thank current members Yesung Kang, Seungwon Kim, Jaewoo Kim, Sunmean Kim, Mingyu Woo, and Taeho Lim and former member Jaemin Lee. We had wonderful projects, and we spent great times together. In particular, Mingyu made visualization script for the contest result at Fig. 4, 11, 14 and appendix.

I would also like to thank my thesis committee members, Prof. Jaehyouk Choi and Prof. Seong-Jin Kim for taking time out of their busy schedules to review and evaluate my research. I am grateful for their valuable feedback.

Last, but not least, I would like to thank my parents Jaihong Do, Huijo Jung and the rest of my family for their endless love and support. This thesis is dedicated to them.

VITA

1992 Born, Changwon, South Korea

2016 B.S., Electrical and Computer Engineering
Ulsan National Institute of Science and Technology, Ulsan, South Korea

- Seungwon Kim, **SangGi Do** and Seokhyeong Kang, "Fast Predictive Useful Skew Methodology for Timing-Driven Placement Optimization", *Proc. ACM/IEEE Design Automation Conference*, 2017, pp. 18-2.

- **SangGi Do**, Seungwon Kim, and Seokhyeong Kang, "Skew control methodology for useful-skew implementation", *Proc. International SoC Design Conference*, 2016, pp. 221-222.

Chapter I

Introduction

For the circuit design, standard cell methodology is one of the most efficient and effective ways. Standard cell methodology encapsulates the large-scale integration (transistor-level) into an abstract logic representation (gate-level). This encapsulation decreases the physical design complexity for EDA. However, in advanced technology nodes, conventional standard cells have reached a limit of routing congestion for the complex design. In other words, single row height is not enough efficient in the complex standard cell design. Thus, multi-deck standard cell which occupies multiple rows has been proposed for the new technology. Fig. 1 shows single-deck and multi-deck standard cell structures with power (VDD) and ground (VSS) pins. This new technology can solve internal wire congestion, decreases the standard cell area, and provides low-power and high performance compared with the conventional single-deck standard cell. However, this new technology is not yet widely used due to the several disadvantages.

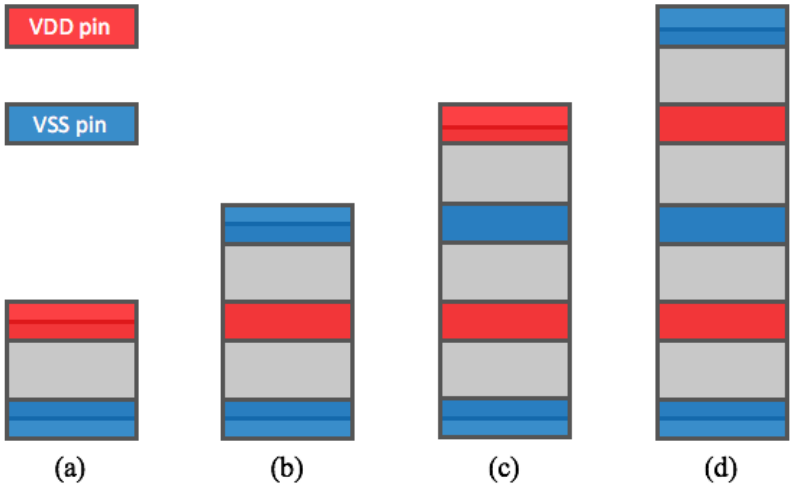


Fig. 1. Single-deck and Multi-deck standard cell with their vss and vdd pins. (a) a single row height; (b) a double row height; (c) a triple row height; and (d) a quadruple row height.

In the modern circuit design, however, the standard cell legalization problem has become more challenging because of complicated design rules and design utilization at the advanced technology nodes [1]. Furthermore, multi-deck standard cell technique makes this problem even worse. Multi-deck standard cell structure increases the physical design complexity of the legalization problem exponentially. Thus, multi-deck standard cell is not widely used.

The most important characteristic of the conventional standard cell is that every standard cell has the same height and this makes a lot easier to design a chip using EDA tools. However, multi-deck standard cell has various cell heights and this increases physical design complexity. Fig. 2 shows aligned multi-deck standard cells with power (VDD) and ground (VSS) rails.

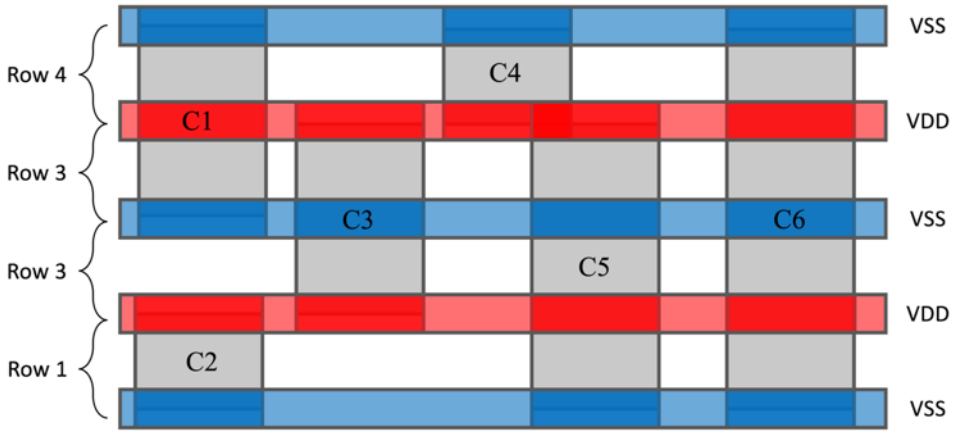


Fig. 2. An example of multi-deck standard cells aligned with the Power/Ground rail defined in the design. C2 and C4 are single row height cells, C1 and C3 are double row height cells, C5 is triple row height cell, and C6 is quadruple row height cell.

There are several recent multi-deck legalization methods, and these can be categorized as (i) heuristic algorithms [2, 3] and (ii) analytic methods [4, 5]. Each method has advantages and disadvantages. Heuristic algorithm can solve legalization problem in fast, but cannot guarantee the optimal solution. On the other hand, analytic method guarantees the optimal solution, but it takes long runtime.

In this thesis, we propose a fast legalization method for multi-deck standard cells based on a metaheuristic method using a simple nearest search and simulated annealing technique. The target objective of this work is not only minimizing the cell displacement sum, but also minimizing the maximum cell displacement. The main contributions of this work are summarized as bellows.

- Proposed legalization method can handle not only double-deck standard cell, but also multi-deck standard cell.
- Using multiple stage legalization, the extremely high utilization case also can be legalized without violation.
- Under the ICCAD 2017 contest benchmarks, our legalization method achieves fast runtime result compare with Top3 teams with comparable quality score.

Chapter II

Related Works

The standard cell legalization is one of the most important problems in the placement procedure during the circuit physical design. The placement procedure is divided in two steps, global placement and detailed placement. The global placement optimizes wire-length, routability, power and other stuff. On the other hands, the detailed placement is more focusing on the legalization. The objective of standard cell legalization is aligning the standard cells and remove overlapping to fit the design rules as well as maintaining the global placement result.

Since the legalization is essential for the physical design, various legalization algorithms are studied [6, 7]. By dividing the legalization algorithm into two categories, heuristic and analytic approach. Heuristic approach is traditional that seek optimal solutions by trials and errors. This can find the solution in practical time, but does not guarantee the optimal solution. The analytic approach builds mathematical models by objective functions and seeks the global optimal solution. This analytic approach guarantees the optimal solution, but it cannot solve the problem in practical time when the complexity of the problem is high.

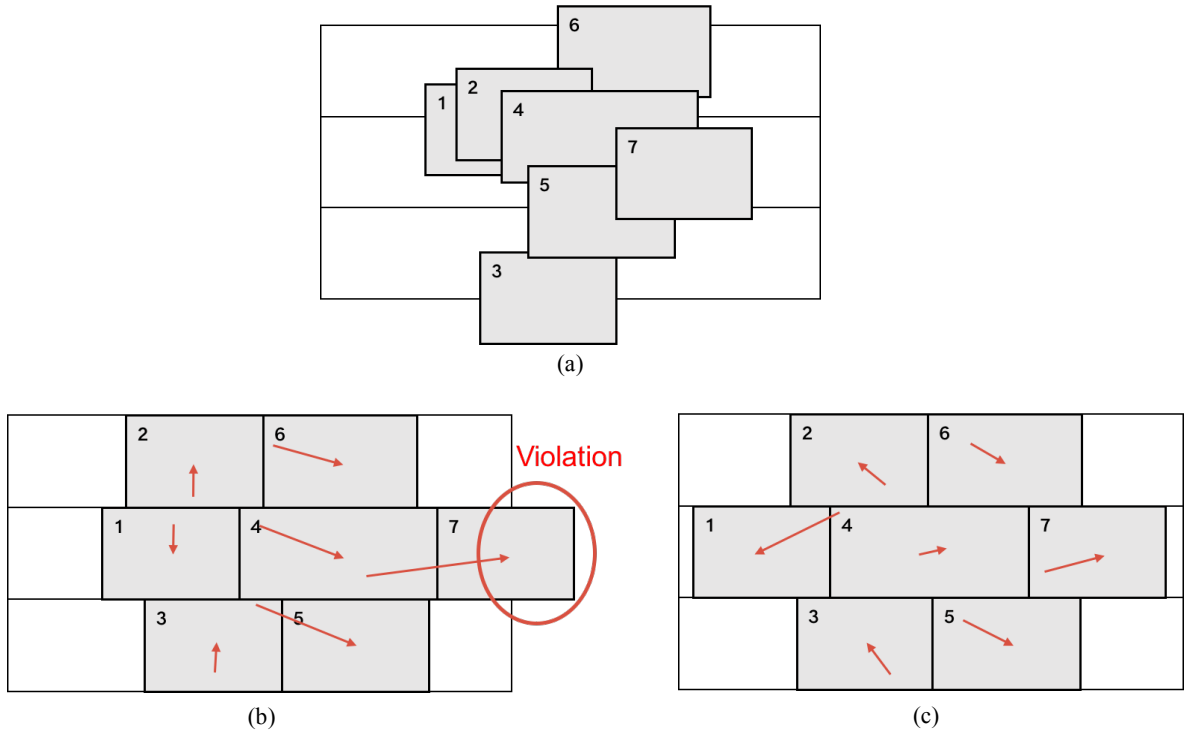


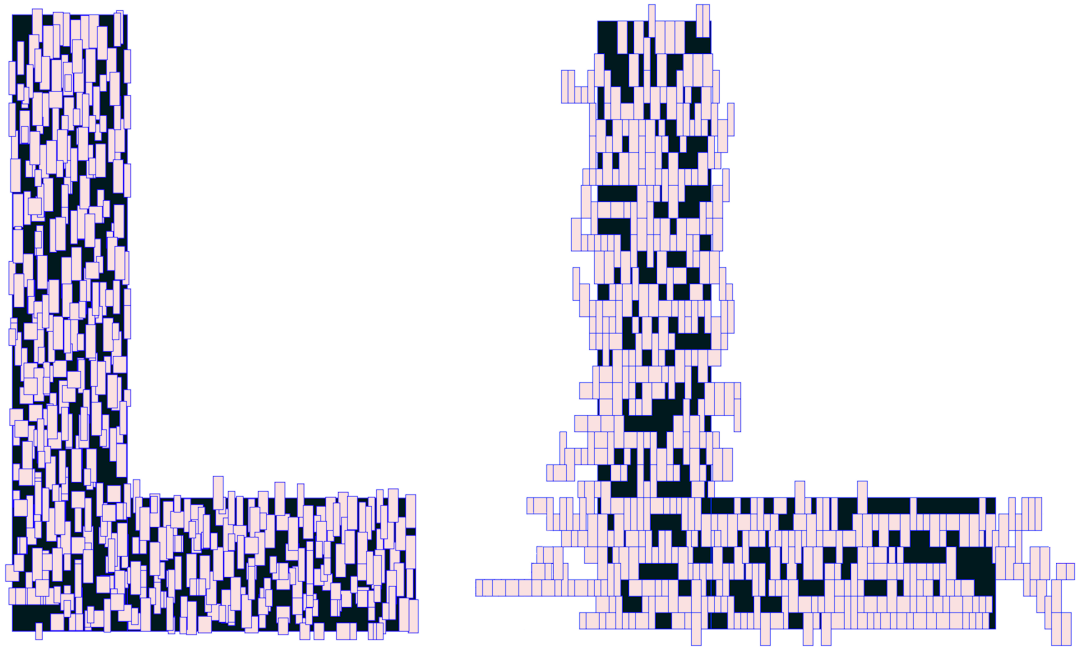
Fig. 3. Example of Tetris and Abacus legalization. (a) is global placement, (b) is Tetris based legalization, (c) is Abacus based legalization.

2.1 Single-deck Legalization

Single-deck legalization is a common legalization problem that every standard cell has a same height. This problem is quite easier than multi-deck legalization problem. Since the performance of a computer is not enough to solve analytical models, heuristic approaches are common for standard cell legalization problems. Tetris [6] and Abacus [7] are the famous single-deck standard legalization method. Fig. 3 shows an example of Tetris, and Abacus based legalization. Tetris based legalization use greedy heuristic and Abacus use dynamic programming. In terms of displacement performance, Abacus shows better performance compared with Tetris.

2.2 Multi-deck Legalization

Multi-deck legalization problem is much harder than a single-deck legalization, because shifting a cell may cause cell overlaps in other rows. Thus, heuristic approach was tried first [3, 8]. [3] Extend Abacus [7] algorithm for the multi-deck standard cell and [8] use dynamic programming. On the other hands, recent studies use analytic approaches [4, 5]. [5] Use dynamic programming with LP (Linear Programming) based refinement and [4] use LCP (Linear Complementarity Problem) solver. A state-of-the-art is LCP based legalization, but the LCP based approach has violation problem. Fig. 4 shows the implementation of LCP based legalization for the ICCAD contest 2017 benchmark. As shown in Fig. 4(b), LCP based legalization makes region violation at high utilization condition.



(a) (b)

Fig. 4 Example of LCP based legalization. (a) is global placement, (b) is LCP based legalization.

Chapter III

Proposed Algorithm

Proposed algorithm consists of three large steps; pre-legalization, multi-deck legalization, and quality refinement. Fig. 5 represent the overall flow of the proposed algorithm. First, pre-legalization step arranges the region violation cells. Second, multi-deck standard cell legalization step legalizes remaining cells using three small steps; nearest search, local shifting, and brick building. At last, improve the displacement quality using simulated annealing using cell moving and cell swapping.

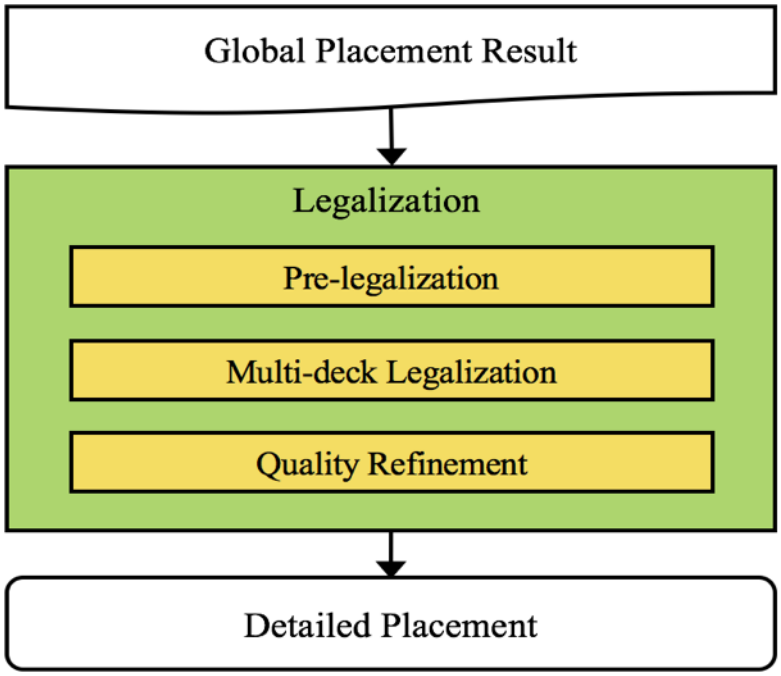


Fig. 5. Overall flow of proposed algorithm.

3.1 Pre-legalization

On the placement stage, hierarchical design has sub modules and their physical region can be assigned by the EDA tools. Not only for the hierarchical design, restricted placement region is frequently used for various reasons. Provided benchmarks from ICCAD contest 2017 also contain various numbers of fence region.

Pre-legalization step is similar with the next step; multi-deck legalization. However, pre-legalization only considers the region violation cells. Because displacement of the region violation cells determines the maximum cell displacement. Considering the region violation cells before the multi-deck legalization step, maximum cell displacement can be minimized. Below in Fig. 6 shows region convergence and region divergence. Region convergence converges the region violation cells into the

region boundary. Region divergence diverges the region violation cells out of the region boundary. Nearest search algorithm finds a nearest available position for each cell movement.

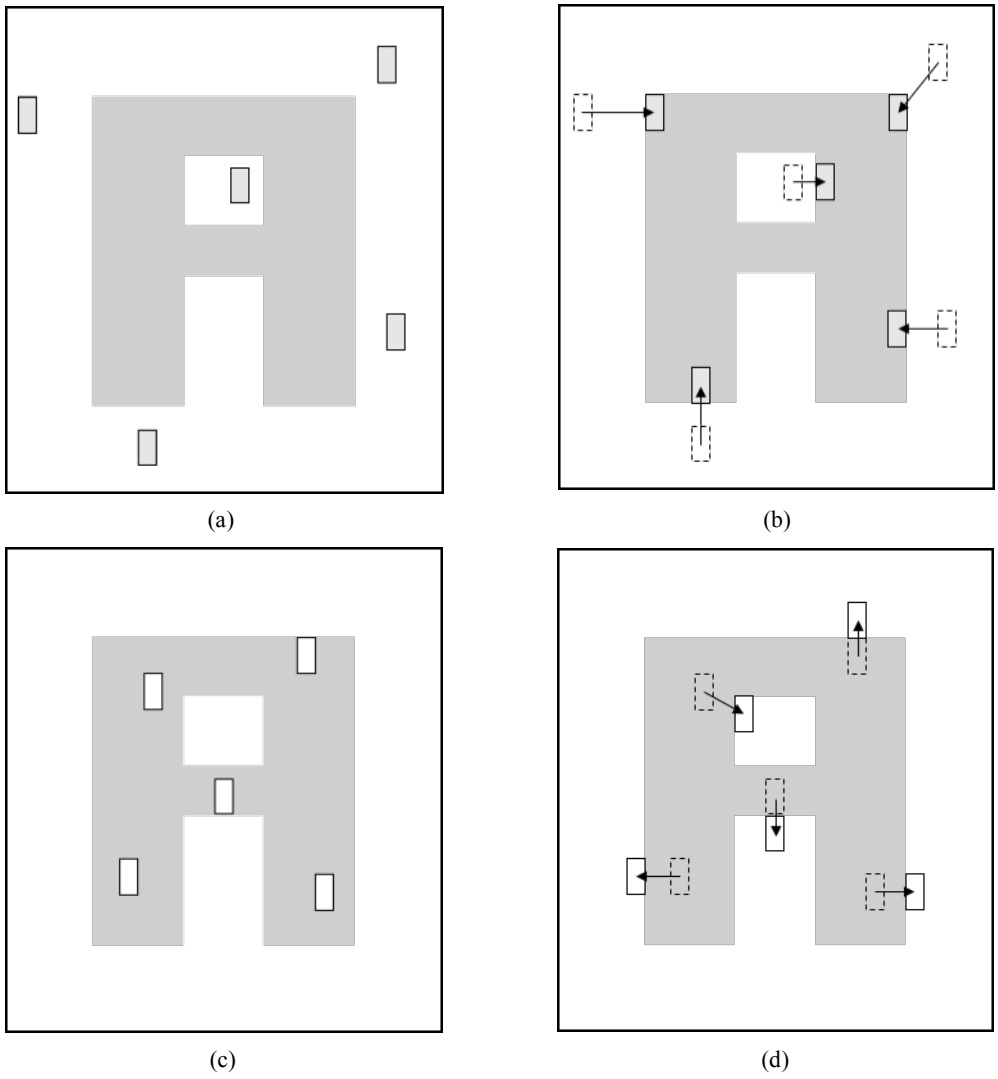


Fig. 6. Example of pre-legalization. (a) to (b) is region convergence, and (c) to (d) is region divergence.

3.2 Multi-deck Legalization

Multi-deck legalization consists of three steps; nearest search, local shifting, and brick building. First, the nearest search algorithm finds the nearest available position for the multi-deck standard cell. When the nearest search algorithm cannot find an available position, the local shifting algorithm trying to insert the multi-deck standard cell by shifting the neighbor cells. Also, when the local shifting cannot find an available insertion position, the brick building algorithm trying to legalize. Fig 7. shows the detail flow of the multi-deck legalization steps. The pre-legalized GP (global placement) result comes from the previous step. Without changing pre-legalized cells, the nearest search, local shifting, and brick building legalize the remaining cells. Finally, after the multi-deck legalization step, we can get the legalized result.

3.2.1 Nearest Search

Nearest search is a core algorithm of the proposed multi-deck standard cell legalization. This algorithm used not only this step, but also pre-legalization, local shifting and brick building. Despite the previous researches can only consider the movement of the x axis, our nearest search algorithm can find a nearest available position considering both x and y axis using 2-D grid structure. Also, using the encapsulated square grid, the nearest search algorithm achieves ultra-fast runtime not only for this step, but also the whole multi-deck standard cell legalization process.

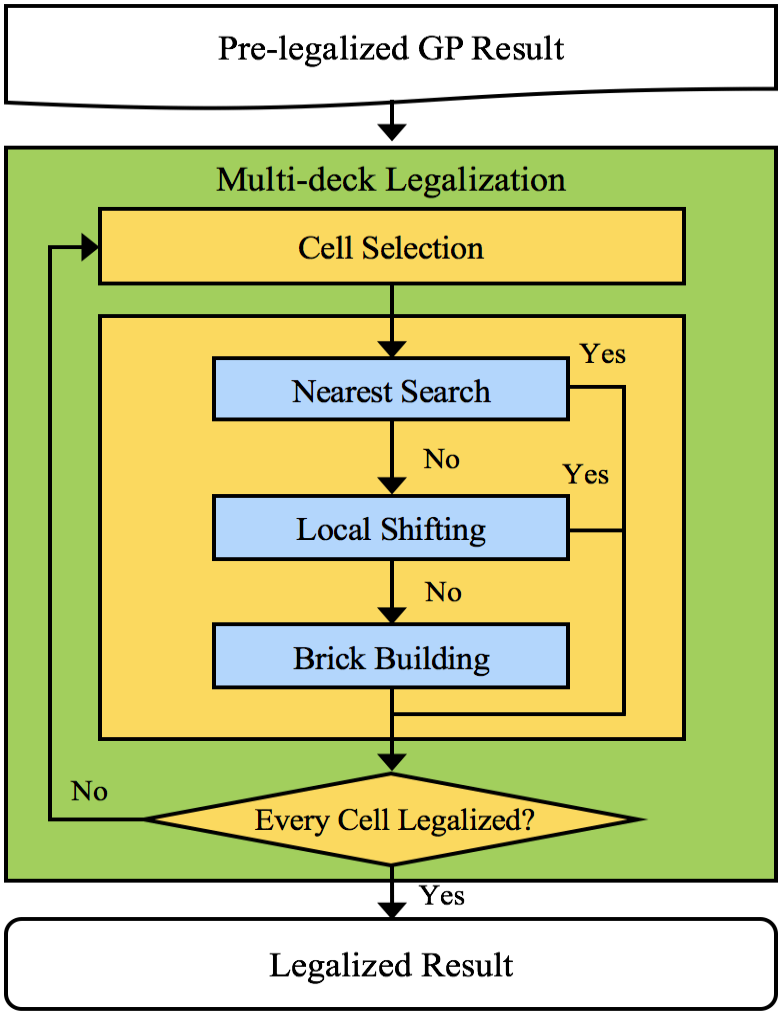


Fig. 7. Detail flow of the multi-deck legalization.

Algorithm 1 describes our nearest search procedure. From the provided benchmarks, row height is 10 times larger than site width. Thus, construct the encapsulated square grid *G'* using 2-D grid by rows & sites (Lines 2-3). Search the surrounding area from the encapsulated square grid *G'* by diamond shape until the available point is found (Lines 4-19). Each loop, search distance is increased when there is no available position (Line 18). If the search boundary is over the search limit *L*, stop searching and return false (Lines 5-7).

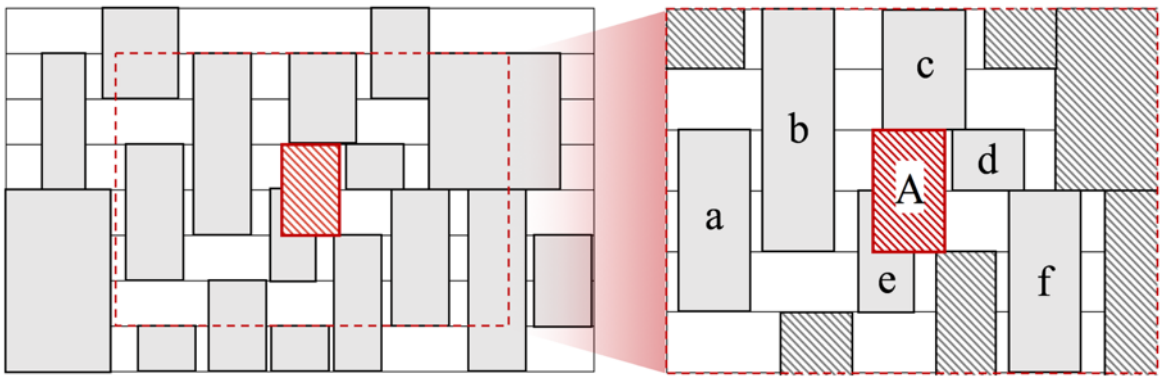
ALGORITHM 1: Nearest Search

input : $C \leftarrow$ Target cell
$P_{(x,y)} \leftarrow$ Search point coordinates
output: $T_{(x,y)} \leftarrow$ Nearest Available Position from 2-D **input** grid G

```
Define A as available position set and L as limitation of search boundary.
Define G as 2-D grid of circuit by rows & sites.
Define G'(s) as a set of encapsulated square region which distance is s
  from P_(x,y). (Initial s is zero.)
while A is empty do
    if Search boundary is over then L then
        return false.
    end
    Construct G'(s).
    for each square region in G'(s) do
        if square region has available position for C then
            Push available position to A.
        end
    end
    if A is not empty then
        Find nearest available point T_(x,y) from grid G.
        return T_(x,y).
    end
    Increase the search distance s.
end
```

3.2.2 Local Shifting

When nearest search cannot find an available position, local shifting start. The local shifting algorithm in this step is different with conventional shifting algorithms [6, 7, 9]. Conventional shifting algorithms shift the overlapping cells and/or clustering the overlapping cells on the x axis. However, multi-deck standard cell cannot apply conventional method because multi-deck standard cell occupying the multiple rows. Hence, an avalanche can happen with conventional method. Because, single shifting can cause the thousands of cell movement. To solve this problem, this local shifting algorithm reconstructs the neighbor cells using the nearest search algorithm. By reconstructing the neighbor cells, shifted result can be obtained.



(a) (b)

Fig. 8. Example of region selection. (a) is region clip. (b) is selected region and cells.

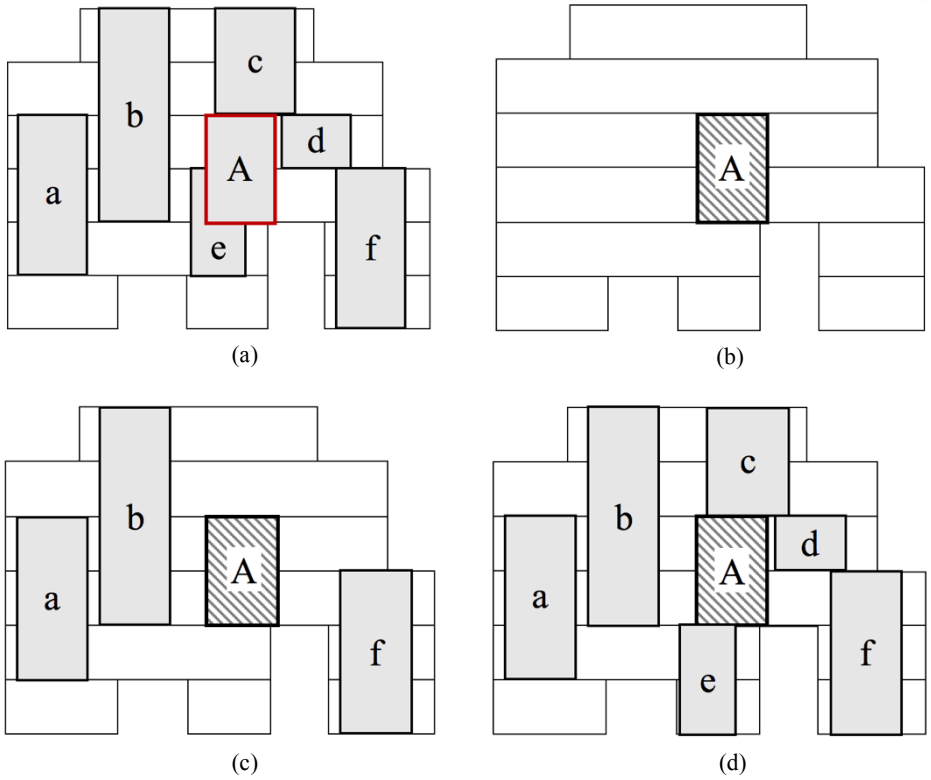


Fig. 9. Example of the local shifting. (a) is initial state and target cell A is overlapped. (b) is available area defined for neighbor cells (a to f) and target cell A. (c) place the bigger cells first. (d) place the remaining cells.

For the local shifting, region selection is required. Fig. 8 shows region selection. First, using region clip, set the around area. And obtain the neighbor cells and blocked area. Neighbor cells are *a* to *f* in Fig. 8(b) and blocked area is a black diagonal pattern from Fig. 8(b). Target cell *A* is represented as a red diagonal pattern from Fig. 8.

After region selection, reconstruction starts. Fig. 9 represent reconstruction steps and algorithm 2 shows a detail description of local shifting procedures. The initial state shown as Fig. 9(a). After overlapping cell collection, neighbor cells are ready for local shifting (Lines 3-7). Next, place the target cell *A* as Fig. 9(b) (Line 8). Due to the reason that smaller cells are easier to find the available position, sort the neighbor cells (Line 9) in ascending order. Fig. 9(c) shows the multi-deck standard cell placement for larger cells and Fig. 9(d) shows the multi-deck standard cell placement for smaller cells (Lines 10-19). However, if neighbor utilization is high, local shifting also cannot find legalization result. In this case, brick building step is required.

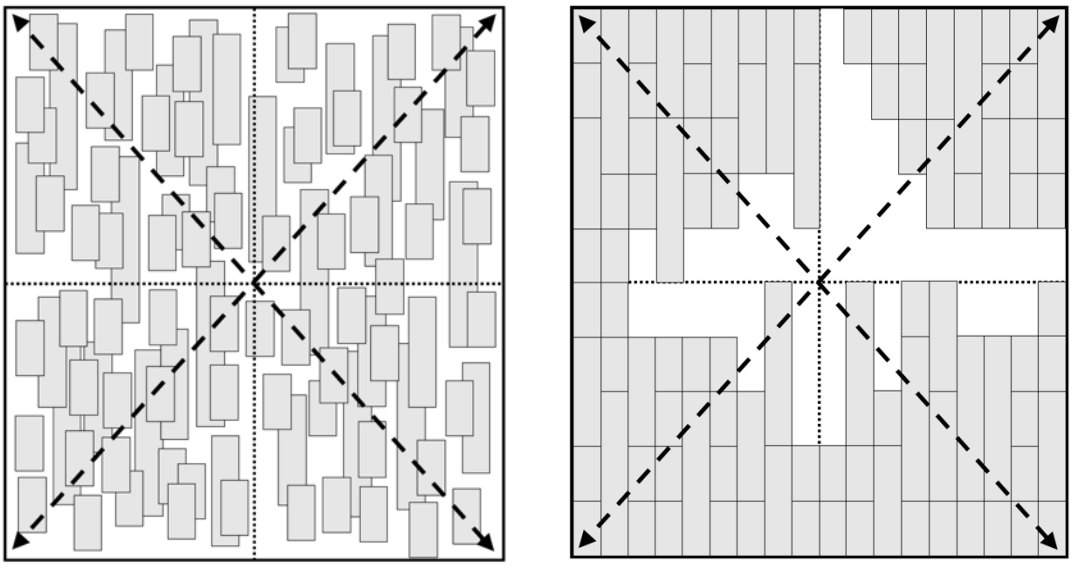
ALGORITHM 2: Local Shifting

input : $C \leftarrow$ Target cell
$T_{(x,y)} \leftarrow$ Target point coordinates
output: Placed result

```
Define G as 2-D grid of circuit by rows & sites.
Define Q as cell queue and R as region boundary.
for each cell overlap with R do
    if cell is inside the R then
        Push cell to Q and erase from G.
    end
end
Place the target cell C at target coordinates T_(x,y).
Sort Q by cell size in descending order.
while Q is not empty do
    Set F as front cell from Q.
    if Nearest_Search(F, GP_coordinates of F) is valid then
        Place the cell F.
    else
        return false.
    end
    Pop F from Q.
end
return placed result.
```

3.2.3 Brick Building

The brick building is the final legalization strategy that focus on the ability of legalization. Failing the nearest search and local shifting means that around the area is high utilization. In this case, the possible legalization strategy is minimizing the distance between the neighbor cells. Fig. 10 shows brick building. This algorithm only used the benchmarks 'des_perf_b_md1' which maximum fence utilization is 96.2%.



(a) (b)

Fig. 10. Example of brick building. (a) The global placement result. The target region is divide into four sub-regions. (b) All cells in each sub-region move toward each corner.

Fig. 11. Example of complicated fence region legalization (des_perf_b_md2). (a) is global placement result. (c) is legalized result. (b) is zoom in of (a) and (c).

3.3 Quality Refinement

After the multi-deck legalization step, quality refinement optimizes the legalized result. As our multi-deck legalization is heuristic method, legalized result is not an optimal result. In this step, we optimize the legalized result with cell moving and cell swapping techniques with a metaheuristic algorithm. Simulated annealing is used for metaheuristic algorithm to optimize the total and maximum displacement of legalized result. Each cell movement and cell swaps do not affect legality.

3.3.1 Cell Move

For the cell moving, the nearest search algorithm used to find the available position. However, cell position of legalized result has been already nearest available positioned from the global placement. To solve this limitation of heuristic algorithm, we adopt simulated annealing method. As we move the cells with displacement penalty, we can get a chance to improve the displacement quality that not only compensate but also improve the displacement penalty.

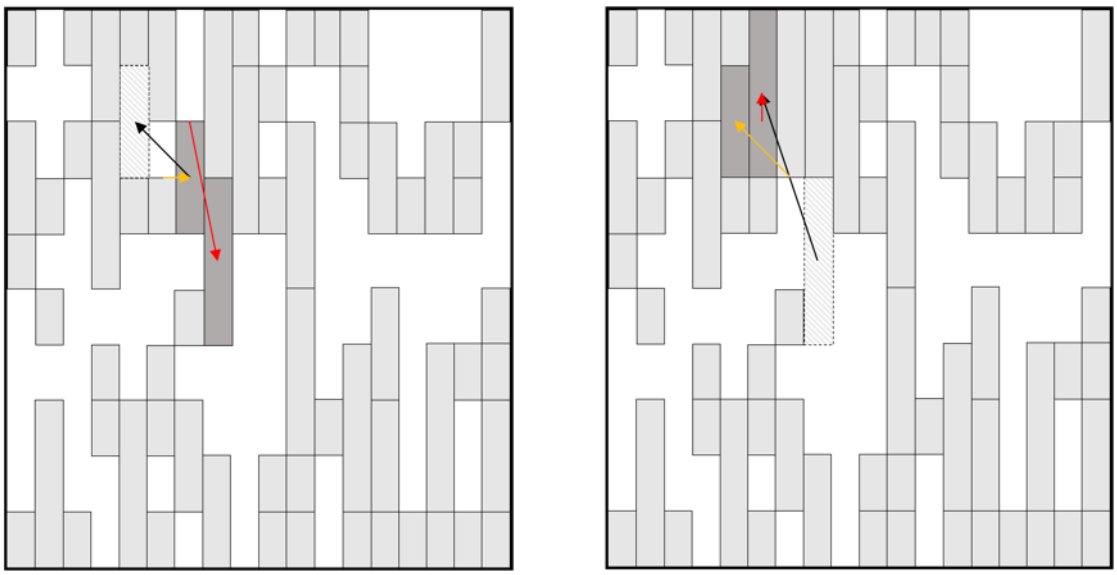
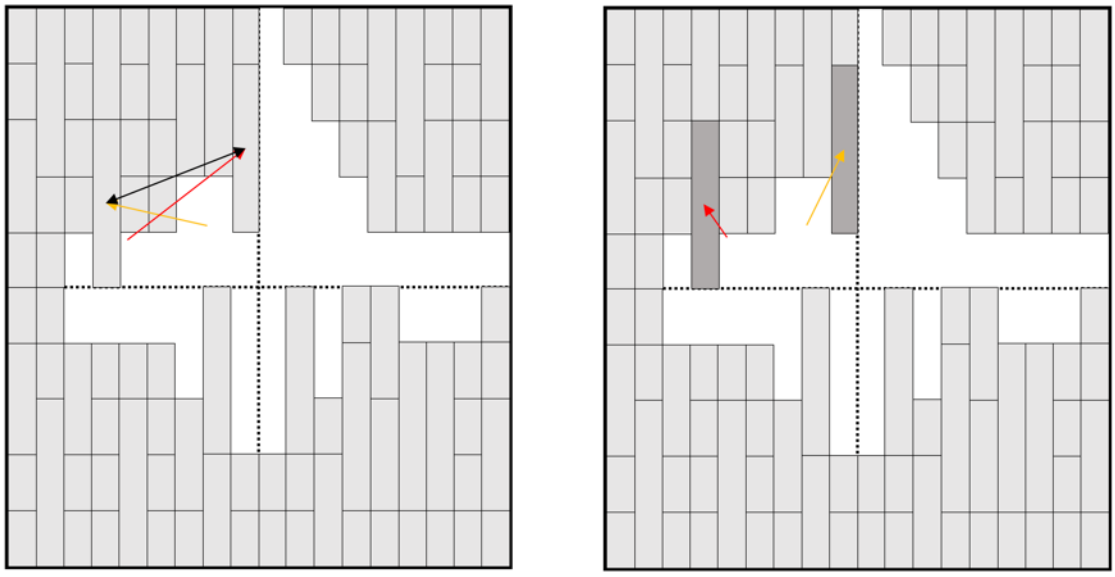


Fig. 12. Example of cell move. (a) is before moving, (b) is after moving. Black arrow is moving direction and orange & red line represent cell displacement.

Fig. 12 shows how cell movement improves the displacement quality. First, find the nearest available position of the target cell using nearest search. If the displacement penalty is less than the threshold value, move the cell. This movement might increase the cell displacement as shown in Fig. 12(a). On the other hands, other cell movement might decrease the cell displacement as shown in Fig. 12(b). From this two cell movement, previous cell displacement (orange arrow) is increased, but another cell movement (red arrow) is decreased. The summation of the two delta displacement is negative.

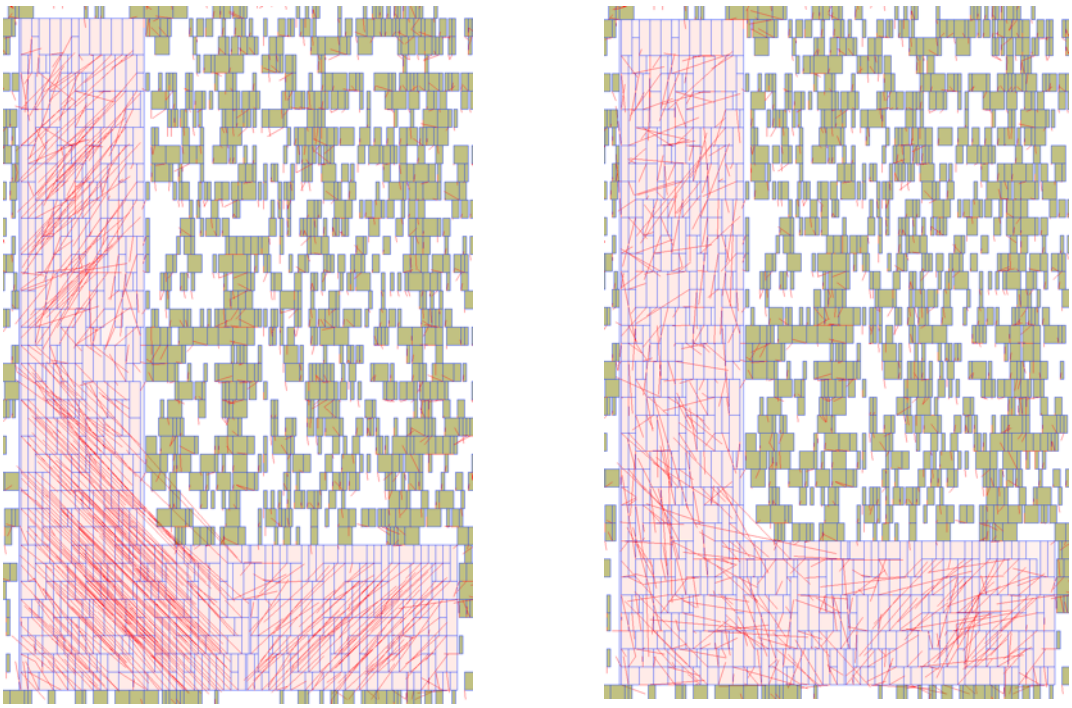
3.3.1 Cell Swap

Not only the cell movement, but also cell swap can improve the cell displacement. The principle is same as a previous cell movement. The difference is that only same size cells can be swapped. If we swap different sized cell, we have to consider overlapping or neighbor cells. Thus, we only consider same sized cell. Fig. 13 shows a cell swapping example.



(a) (b)

Fig. 13. Example of cell swap. (a) is before swapping, and (b) is after swapping. Black arrow designates swapping cells and orange & red line represent cell displacement.



(a) (b)

Fig. 14. Experiment result of cell swapping using 'des_perf_b_md1' benchmark.

The amount of displacement increase is less than displacement decrease from another cell. This cell swapping is effective when utilization is high. In case of benchmark 'des_perf_b_md1', maximum utilization of fence region is 96.2%. This means the fence region is almost full of standard cells. At this high utilization condition, cell swapping is much effective than cell movement. Fig. 14 shows a cell swapping experiment result using 'des_perf_b_md1' benchmark. The utilization of fence region in Fig. 14 is 96.2% and maximum height of multi-deck standard cell is double of single-deck standard cell. The red line represents the cell displacement. We can clearly identify that the cell displacement is improved using cell swapping.

3.3.2 Simulated Annealing

Simulated annealing is one of the most famous metaheuristic algorithm. This metaheuristic algorithm is a probabilistic technique for near global optimum for the given function. In this thesis, we use total displacement for the objective function as Equation 1.

$$min \sum_{i=1}^{n} |x_i' - x_i| + |y_i' - y_i| \qquad (1)$$

$$(x_i, y_i) = global\ placement\ position$$

$$(x_i', y_i') = \ legalized\ position$$

$$n = number\ of\ cells$$

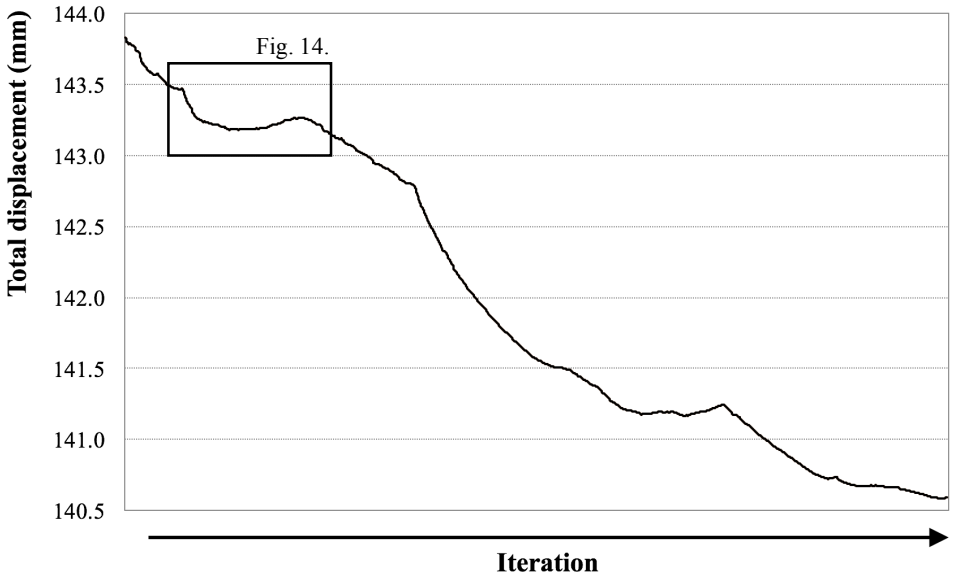


Fig. 15. Total displacement graph according to the iteration using simulated annealing. 'des_perf_b_md1' benchmark is used.

Simulated annealing achieves average 15% improvement from the previous legalized result. Fig. 15 shows the change of total cell displacement according to the iteration. We can see that the simulated annealing algorithm successfully escape the local optimal point and finally converge to near global optimal point.

Fig. 16 show detail view of Fig. 15. Every single edge of this graph is cell move or cell swap. As this graph, we can figure out that our metaheuristic algorithm can optimize multi-deck legalization cells in terms of the total cell displacement.

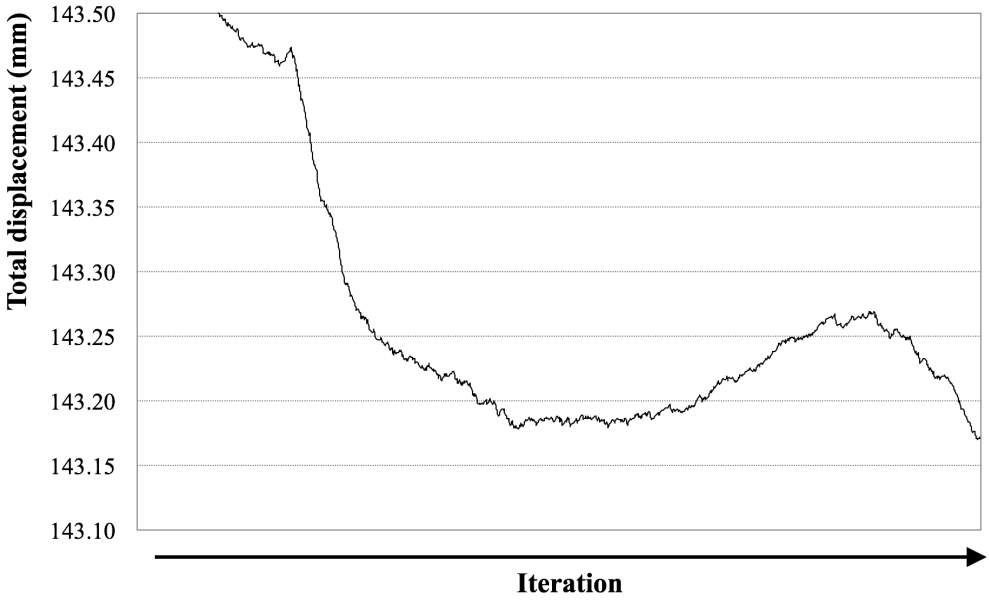


Fig. 16. Detail graph of total displacement according to the iteration using simulated annealing.

Chapter IV

Experiment

Experiment and evaluation are based on the ICCAD contest 2017 [1] environment. We use eight benchmarks from the contest and every experiment result was running on the provided contest CIC machine from ICCAD contest 2017. Evaluation scoring consider displacement quality, HPWL (Half-parameter Wire Length), utilization, runtime and constraint violation penalties. For the soft constraint violation, Mentor Graphics Olympus-SoC resolves the violations and give some penalties. On the other hands, experiment result treats as illegal when violation the hard constraints. The final evaluation score was generated from the ICCAD contest 2017.

4.1 Benchmarks

The benchmark information provided in the ICCAD contest 2017 is summarized in the Table 1. #Cells, #Nets means the number of cells and nets. 1xH, 2xH, 3xH and 4xH of Cell Type means the portion of single-deck, double-deck, triple-deck, and quadruple-deck standard cells. *Max F-Util*. and *Util.* means maximum utilization of fence region and utilization of entire benchmark.

TABLE I

BENCHMARK SUITE INFORMATION FROM ICCAD CONTEST 2017

Benchmark	#Cells	#Nets	#Fence Region	Cell Type (%)				Max F-Util.	Util.
				1xH	2xH	3xH	4xH		
des_perf_b_md1	112679	122951	12	94.8	5.2	0	0	96.2	54.98
des_perf_b_md2	112679	122951	12	90.47	6.02	2.01	1.5	71.74	64.69
edit_dist_1_md1	130661	133233	0	90.31	6.12	2.04	1.53	N/A	67.47
edit_dist_a_md2	127414	134051	1	90.31	6.12	2.04	1.53	9.73	59.42
fft_2_md2	32281	33307	0	89.62	6.56	2.18	1.64	N/A	83.12
fft_a_md2	30625	32090	0	89.57	6.59	2.19	1.65	N/A	32.41
fft_a_md3	30625	32090	0	93.42	2.19	2.19	2.19	N/A	31.24
pci_bridge32_a_md1	29533	34058	3	90.39	6.07	2.02	1.52	38.67	49.57

4.2 Evaluation Metric

Maximum cell movement, average cell movement, HPWL, target utilization and runtime are the component of evaluation metrics. Maximum cell movement is computed based on the maximum Manhattan distance of cells from their original positions divided by the height of a placement row. Average cell movement is the total Manhattan distance of movable cells divided by the number of movable cells and the height of the placement row. Target utilization provided for each design and runtime is directly affecting the evaluation. However, the maximum runtime limitation exists. The maximum number of threads allowed to run in parallel is 8. Following equations are calculation of each evaluation component.

$$S_{score} = S_{am} \times S_{mm} \times (1 + S_{hpwl} + S_v) \times (1 + S_t)$$

$$S_{am} = \frac{\sum_{k=1}^{4} M_{avg,k}}{4} \quad \leftarrow Average\ movement\ score$$

$$S_{mm} = 1 + \frac{M_{max}}{100} \times f_{mm} \quad \leftarrow Maximum\ movement\ score$$

$$S_{hpwl} = \max\left(\frac{hpwl_{legal} - hpwl_{gp}}{hpwl_{gp}}, 0\right) \times (1 + \max(\beta \times f_{of}, 0.2)) \quad \leftarrow HPWL\ score$$

$$S_v = min\left(0.2, \frac{N_v}{N_{cells}}\right) \quad \leftarrow Soft\ constraint\ score$$

$$S_t = max\left(-0.2, min\left(0.2, 0.05 \times log_2\left(\frac{t_{tool}}{t_{median}}\right)\right)\right) \quad \leftarrow Runtime\ score$$

$$f_{mm} = max\left(\frac{\sum_{c_i \in c_v} M_i}{M_x}, 1\right) \quad \leftarrow Maximum\ displacement\ penalty$$

$f_{of} =$ *utilization factor (same as ISPD 2015 contest[cite])*

$N_{cells} =$ *number of cells in the design*

$N_v =$ *number of soft constraint violations in the solution*

$N_{rows} =$ *number of rows in the design*

$t_{tool} =$ *the runtime on the design*

$t_{median} =$ *the mdeian runtime of all participants on the design*

$hpwl_{gp} =$ *the global placement HPWL of the design*

$hpwl_{legal} =$ *the HPWL of the legal design*

$M_{avg,k} =$ *average cell movement of all cells with a height equal to k rows*

$M_x =$ *the maximum cell movement constraints defined for the design*

$M_i =$ *the displacement of cell* c_i

$C_v =$ *a set of cells whose displacements are greater than* M_x

4.3 Constraints

Soft, hard, and netlist constraint should be considered for the multi-deck legalization. Target utilization, maximum cell movement, cell edge spacing, pin access and pin short are considered as soft constraints. If each of these constraints is unsatisfied, a penalty will be enforced for the evaluation scoring. Power and ground alignments of multi-deck standard cell, row and site alignments, and fence regions are defined as hard constraints. Violating these hard constraints is considered as illegal result. Each solution must satisfy these hard constraints. For the netlist constraints, netlist from the design must remain the same. Only cell flipping is allowed as long as no hard constraint violation.

TABLE II

EVALUATION SCORE COMPARISON FROM ICCAD CONTEST 2017 METRIC

Benchmark	Top1			Top2			Top3			Ours		
	Score	RT	Score \w RT	Score	RT	Score \w RT	Score	RT	Score /w RT	Score	RT	Score /w RT
des_perf_b_md1	0.63	6.59	0.61	1.27	10.85	1.27	Illegal	Illegal	Illegal	Illegal	Illegal	Illegal
des_perf_b_md2	0.72	6.26	0.68	0.92	18.01	0.94	0.88	4.8	0.82	0.75	3.93	0.69
edit_dist_1_md1	0.73	8.58	0.70	0.82	20.6	0.84	11.55	5.14	10.71	0.78	4.14	0.71
edit_dist_a_md2	0.53	7.59	0.50	0.78	31.5	0.83	4.42	4.79	4.07	4.35	4.46	3.98
fft_2_md2	0.79	1.6	0.74	1.47	30.64	1.68	1.06	1.08	0.95	1.01	0.93	0.90
fft_a_md2	0.53	1.41	0.50	0.53	2.5	0.52	0.63	1.13	0.58	0.56	11.53	0.62
fft_a_md3	0.42	1.36	0.39	0.41	2.45	0.41	0.5	1.11	0.47	0.43	10.39	0.47
pci_bridge32_a_md1	0.76	1.43	0.73	1.15	2.51	1.15	1.14	1.24	1.08	0.90	0.94	0.84
Avg.	0.64	4.35	0.61	0.92	14.88	0.96	2.88	2.76	2.67	1.25	5.19	1.17
Norm. Avg.	1	1	1	1.43	3.41	1.58	4.51	0.63	4.40	1.96	1.19	1.93

4.3 Result

Our multi-deck legalization binary is written in C++ and tested on Intel Xeon 2.20GHz 64G RAM CIC machine which was provided for the ICCAD 2017 contest. The experiments are conducted with ICCAD contest 2017 benchmarks and evaluated by a contest CIC machine with Mentor-graphics Olympus-SoC tool. Table 2 shows the contest result with a score and runtime. Fig. 17-19 shows raw score, runtime, and score with the runtime result of our experiment result compared with Top 3 contestants. We solve the illegal issue of ‘des_perf_b_md1’, but we can’t get the evaluation result due to the technical issue at the contest server.

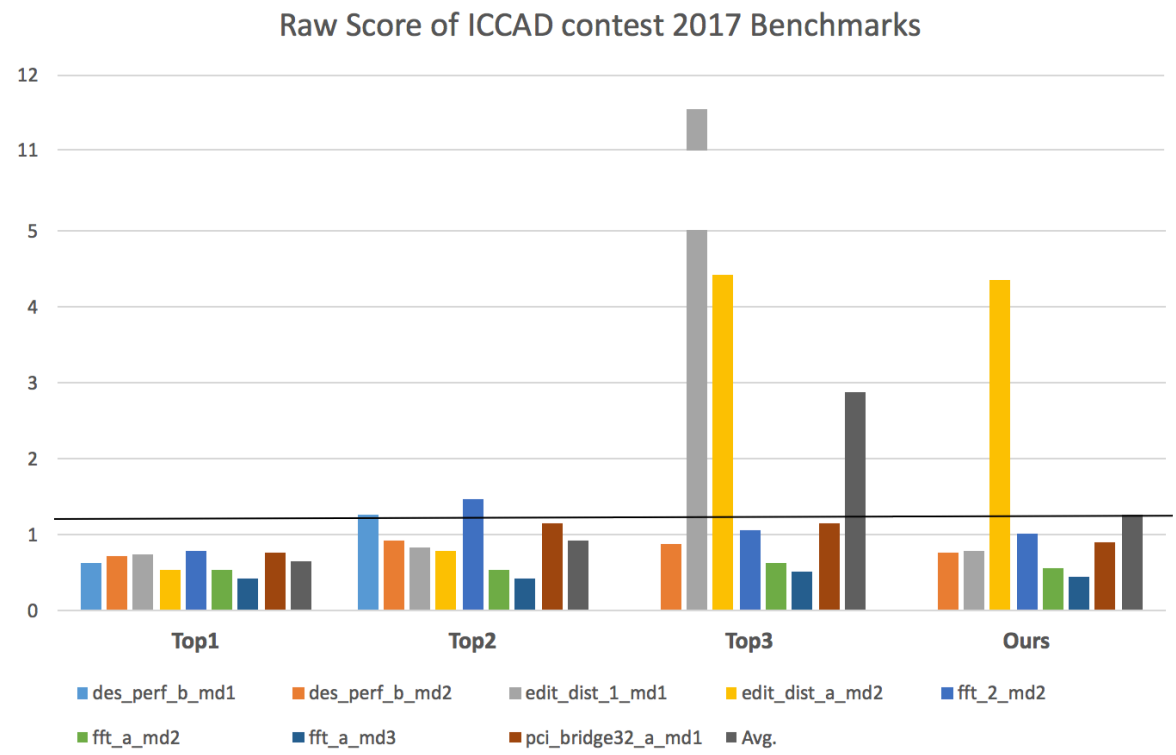


Fig. 17. Raw score of ICCAD contest 2017 Benchmarks.

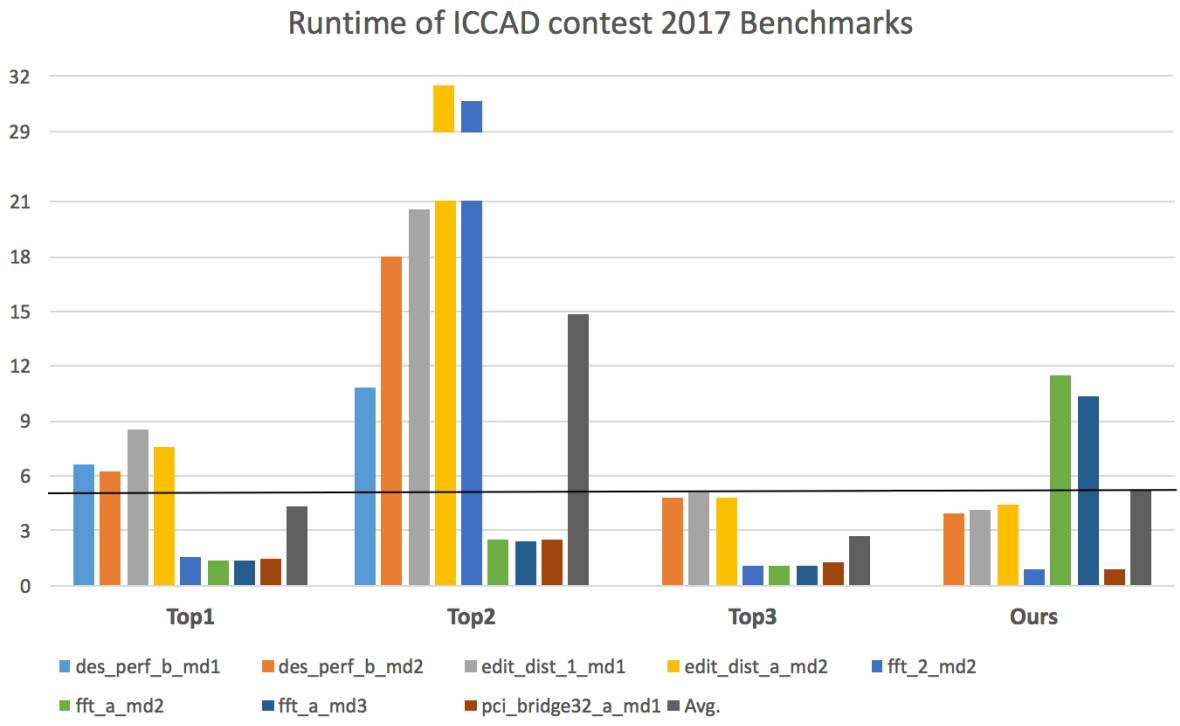


Fig. 18. Runtime of ICCAD contest 2017 Benchmarks.

Our experimental result of raw score is better than 3rd place of the ICCAD contest 2017. Without the largest raw score by 'edit_dist_a_md2' benchmark, our average raw score is better than 2nd place of the ICCAD contest 2017. In terms of runtime evaluation, our result is better than 2nd place of the ICCAD contest 2017. Without the two largest score by 'fft_2_md2' and 'fft_a_md2', our runtime result is better than 1st place of the ICCAD contest 2017.

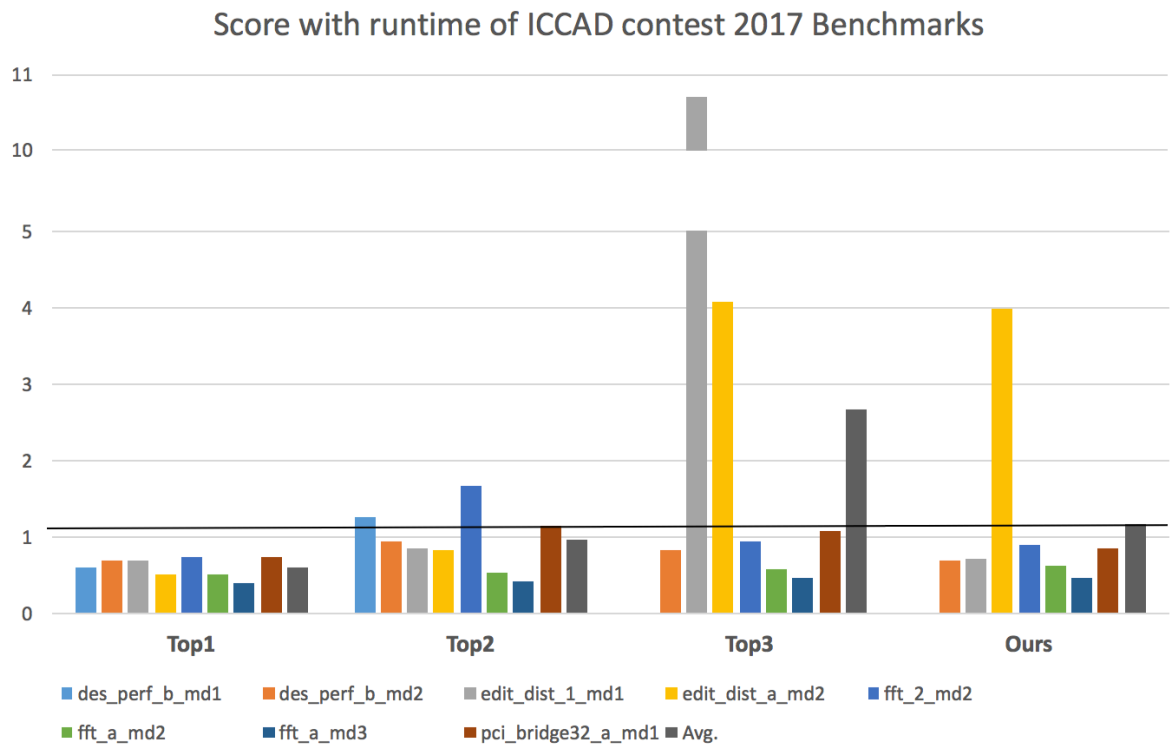


Fig. 19. Score with runtime of ICCAD contest 2017 Benchmarks.

Finally, our experiment result considering both score and runtime is better than 3rd place of the ICCAD contest 2017. However, at the 'des_perf_b_md2' and 'edit_dist_1_md1' benchmarks, our result got an almost identical score with 2x runtimes.

Chapter V

Conclusion

The multi-deck standard cell legalization problem is a new rising problem for the EDA research field. To get the optimal solution for the multi-deck legalization problem, several recent studies tried analytic approaches. However, analytic approach still cannot solve the problem in practical time when they consider both x axis and y axis. Thus, recent analytic approaches were considered only cell movement of the x axis.

In this study, we propose a metaheuristic method for fast multi-deck legalization. Since heuristic approaches has limitation of local optimal solution, metaheuristic overcome this problem. Using our ultra-fast nearest search algorithm, we can get the near optimal result of the multi-deck legalization. Our approach can consider the both x and y axis at the same time. Furthermore, our proposed multi-deck legalization can handle various heights of the multi-deck standard cell which height is more than double row heights. Our implementation gets better performance than 3rd place of ICCAD contest 2017 at the final score, and fastest runtime result in a dominant number of benchmarks.

REFERENCES

[1] N. K. Darav, I.S. Bustany, A. Kennings, and R. Mamidi, "ICCAD-2017 CAD Contest in Multi-Deck Standard Cell Legalization and Benshmarks", *Proc. ICCAD*, 2017.

[2] W. -K. Chow, C. -W. Pui, and E. F. Y. Young, "Legalization Algorithm for Multiple-Row Height Standard Cell Design", *Proc. DAC*, 2016, pp. 83:1-83:6.

[3] C. -H. Wang, Y. -Y Wu, J. Chen, Y. -W. Chang, S. -Y. Kuo, W. Zhu, and G. Fan, "An Effective Legalization ALgorithm for Mixed-Cell-Height Standard Cells", *Proc. ASP-DAC*, 2017, pp. 450-455.

[4] J. Chen, Z. Zhu, W. Zhu, and Y. -W. Chang, "Toward Optimal Legalization for Mixed-Cell-Height Circuit Designs", *Proc. DAC*, 2017, pp. 52:1-52:6.

[5] C. -Y. Hung, P. -Y. Chou, and W. -K. Mak, "Mixed- Cell-Height Standard Cell Placement Legalization", *Proc. GLSVLSI*, 2017, pp. 149-154.

[6] D. Hill, "Method and System for High Speed Detailed Place- ment of Cells within an Integrated Circuit Design", *US Patent* No. 6370673, 2002.

[7] P. Spindler, U. Schlichtmann and F. M. Johannes "Abacus: Fast Legalization of Standard Cell Circuits with Minimal Movement", *Proc. ISPD*, 2008, pp. 47-53.

[8] Y. Lin, B. Yu, X. Xu, J. -R. Gao, N. Viswanathan, and W. -H. Liu, "MrDP: Multiple-row Detailed Placement of Heterogeneous-sized Cells for Advanced Nodes", *Proc. ICCAD*, 2016, pp. 7:1-7:8.

[9] S. Kim, S. Do, and S. Kang, "Fast predictive useful skew methodology for timing-driven placement optimization", *Proc. DAC*, 2017, pp. 55:1-55:6.

[10] A. B. Kahng, J. Lienig, I. L. Markov and J. Hu, *VLSI Physical Design: From Graph Partitioning to Timing Closure*, Springer, 2011.

[11] V. Yutsis, I. S. Bustany, D. Chinnery, and J. R. Shinnerl, "ISPD 2014 Benchmarks with Sub-45nm Technology Rules for Detailed-Routing-Driven Placement", *Proc. ISPD*, 2014, pp. 161-168.

[12] 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", *Proc. ISPD*, 2015, pp. 157-164.

[13] S. -H. Beak, 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", *Proc. SPIE*, 2008, pp. 7268:1- 7268:8.

[14] S. Popovych, H. -H. Lai, C. -M. Wang, Y. -L. Li, W. -H. Liu, and T. -C. Wang, "Density-aware detailed placement with instant legalization", *Proc. DAC*, 2014, pp. 122:1-122:6.

[15] S. -Y. Lin, J. -Y. Chen, J. -C. Li, W. -Y. Wen, and S. -C. Chang, "Eh?Placer: A High-Performance Modern Technology- Driven Placer", *ACM Transactions on Design Automation of Electronic Systems*, 21(3)(2016), pp. 1-27.

[16] S. Dobre, A. B. Kahng, and J. Li, "Mixed cell-height imple- mentation for improved design quality in advanced nodes", *Proc. ICCAD*, 2015, pp. 854-860.

[17] N. K. Darav, I. S. Bustany, A. Kennings, and L. Behjat, "A Fast, Robust Network Flow-based Standard-Cell Legalization Method for Minimizing Maximum Movement", *Proc. ISPD*, 2017, pp. 141-148.

[18] 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)(2015), pp. 1569-1573.

[19] J. Lu, P. Chen, C. -C. Chang, L. Sha, D. J. -H. Huang, C. -C. Teng, and C. -K. Cheng, "ePlace: Electrostatics-Based Place- ment Using Fast Fourier Transform and Nesterovs Method", *ACM Transactions on Design Automation of Electronic Sys- tems*, 20(2)(2015), pp. 1-34.

[20] U. Brenner, "BonnPlace Legalization: Minimizing Movement by Iterative Augmentation", *IEEE Transactions on Computer- Aided Design of Integrated Circuits and Systems*, 32(8)(2013), pp. 1215-1227.

[21] T. R. Gheewala, M. J. Colwell, H. H. Yang, and D. G. Breid, "Dual-height cell with variable width power rail architecture", *US Patent* No. 6838713, 2005.

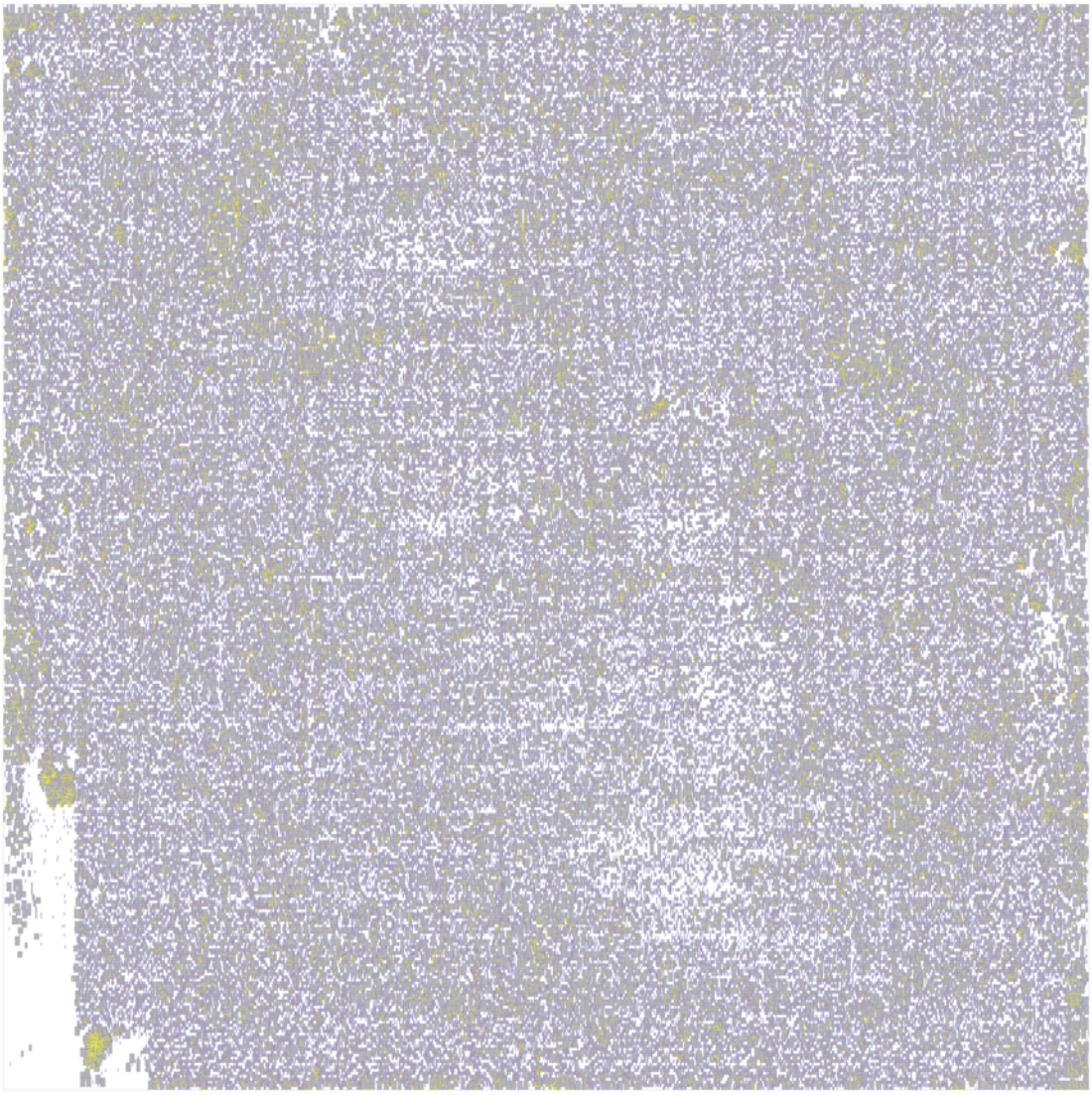
Appendix A

Legalization Result with ICCAD contest 2017 Benchmarks

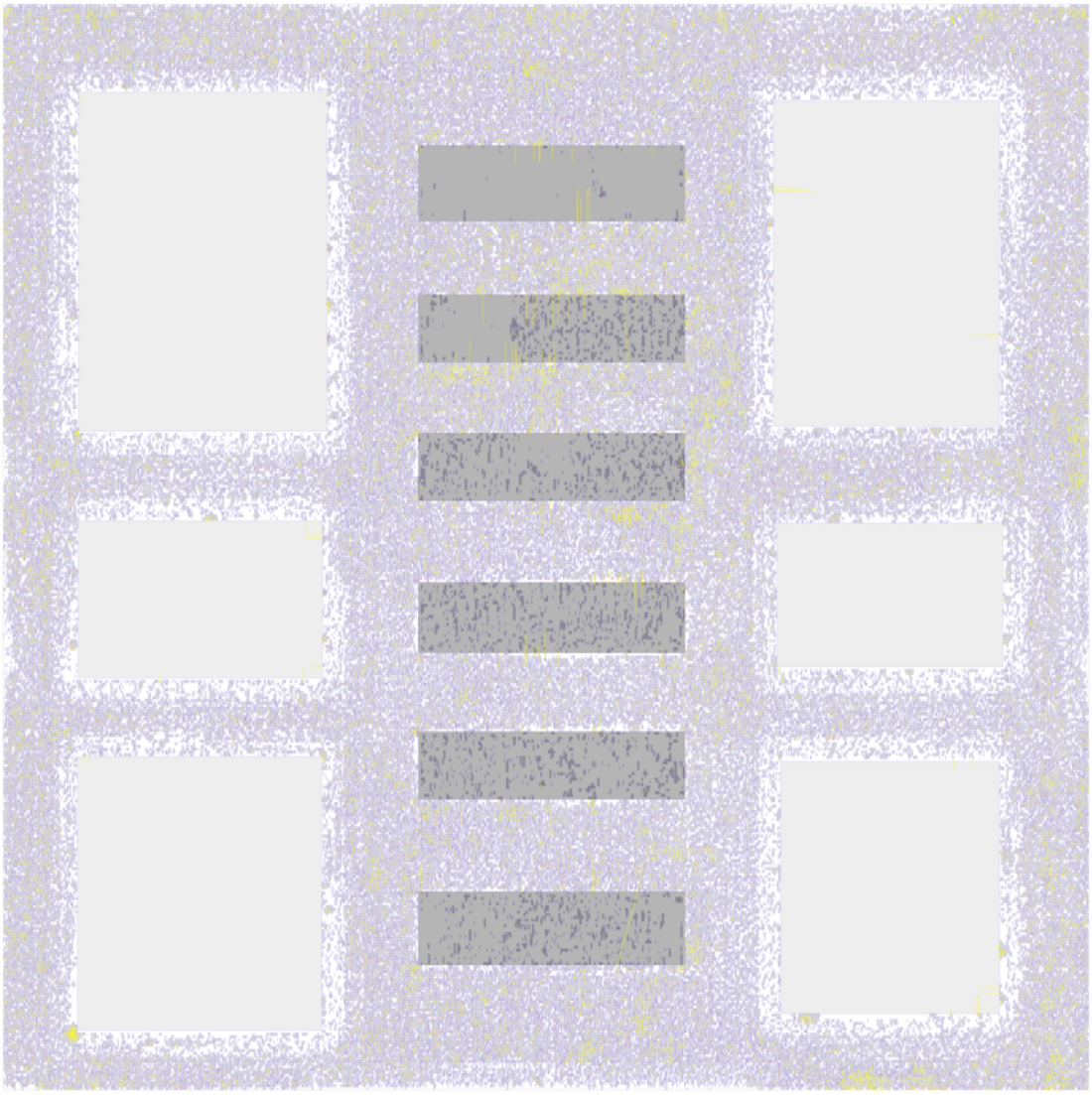
des_perf_b_md1

des_perf_b_md2

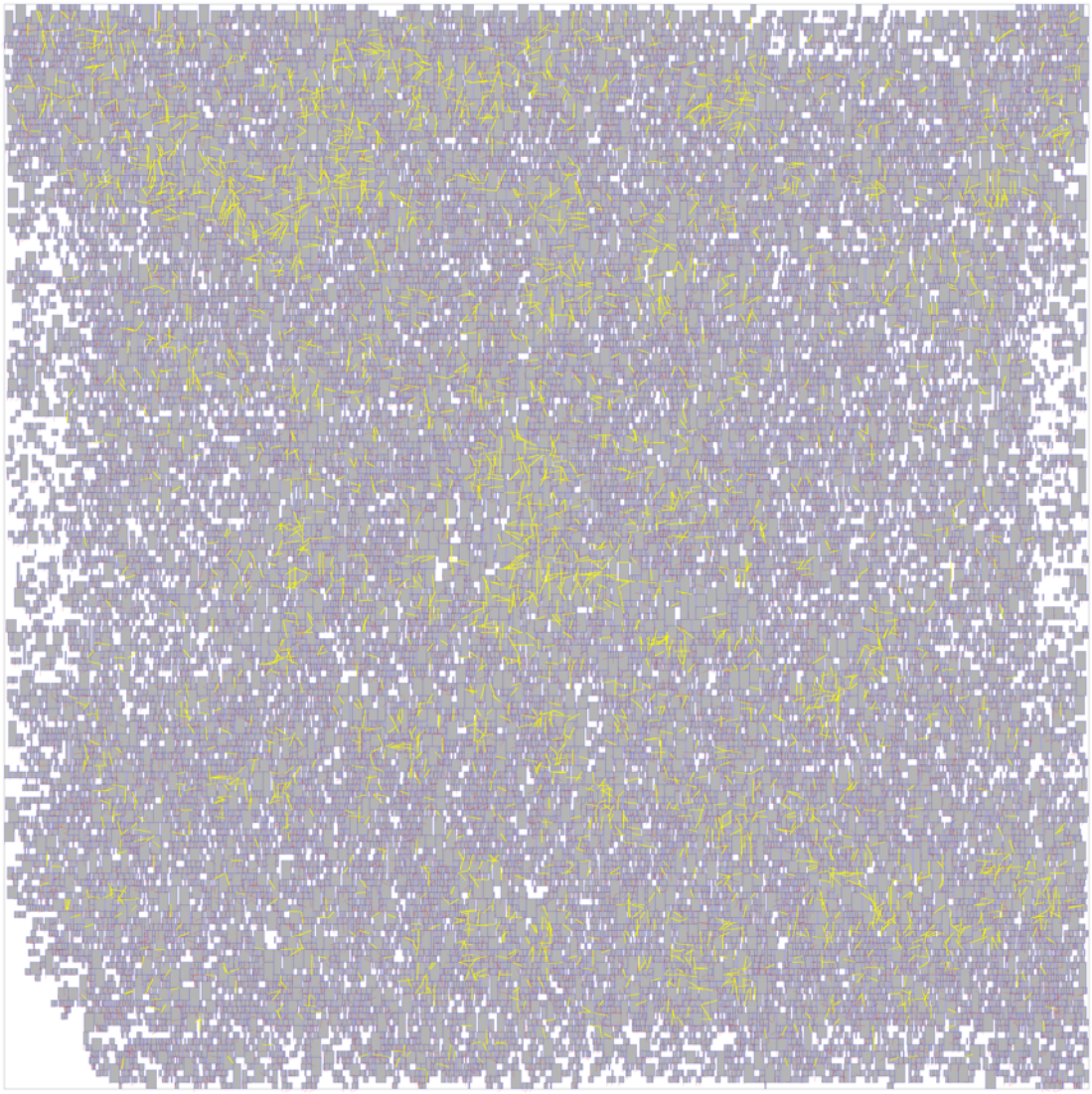
edit_dist_1_md1



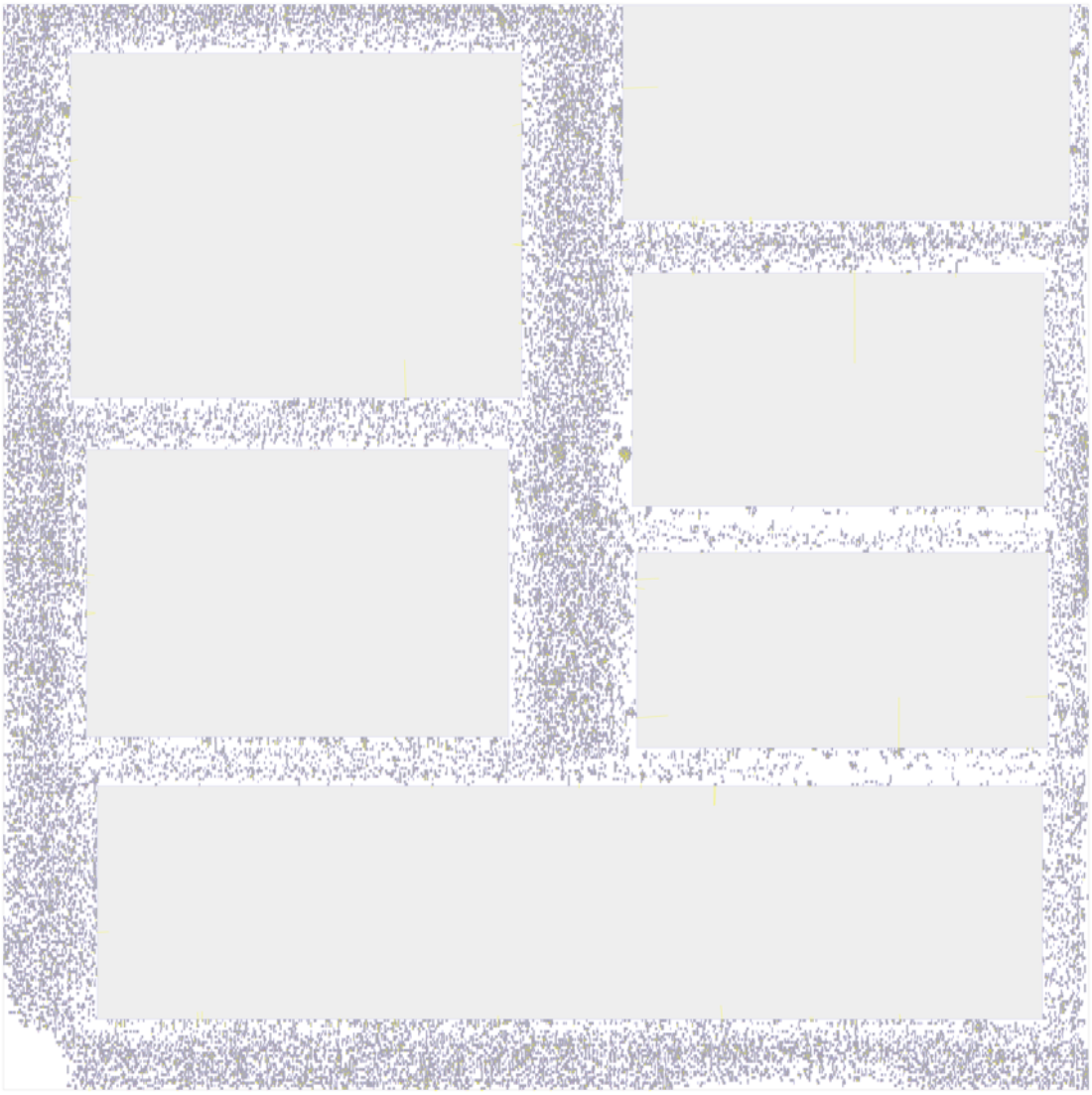
edit_dist_a_md2



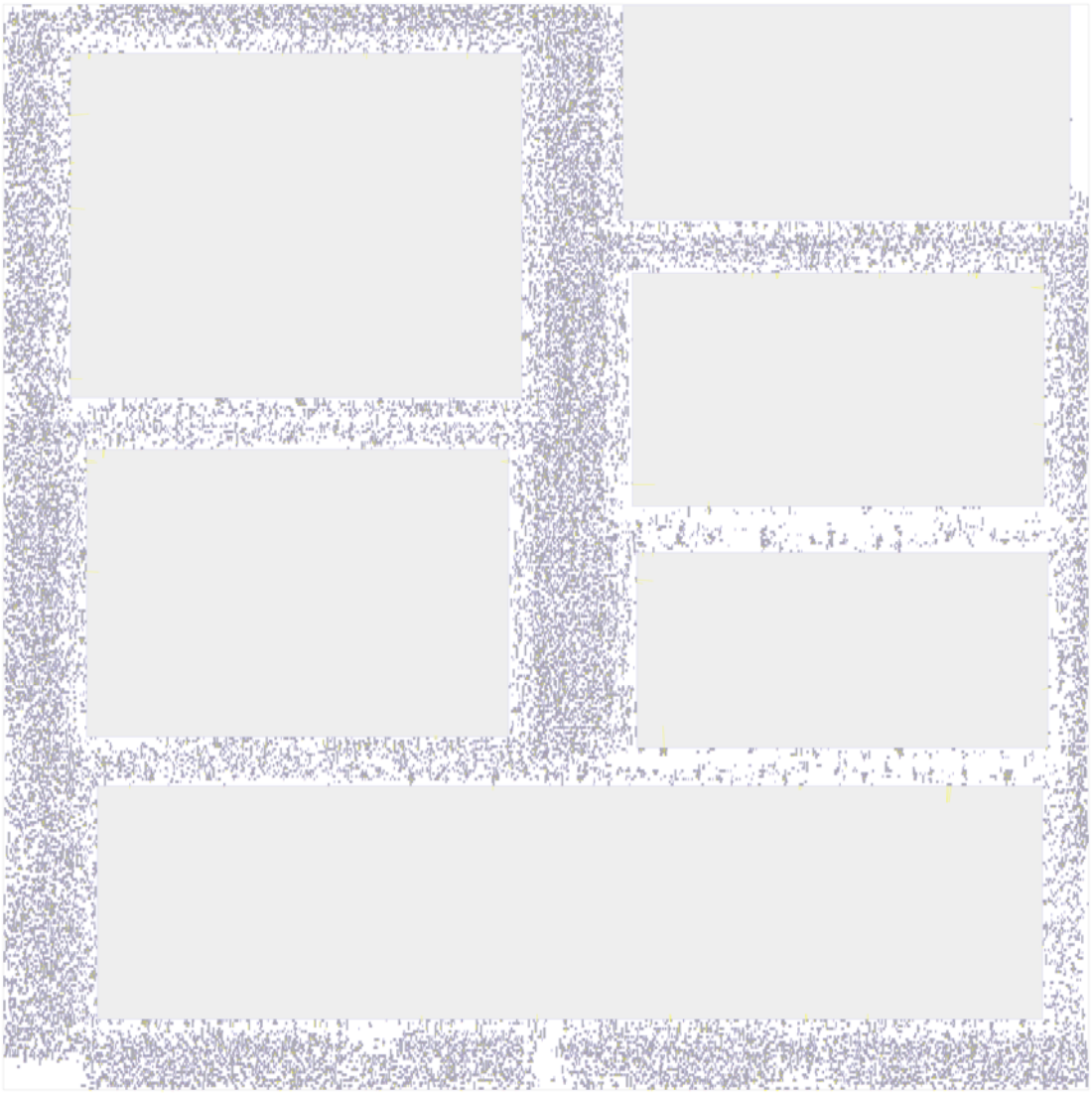
fft_2_md2



fft_a_md2



fft_a_md3



pci_bridge32_a_md1