TPL-Aware Displacement-driven Detailed Placement Refinement with Coloring Constraints

Tao Lin
Iowa State University
tlin@iastate.edu

Chris Chu
Iowa State University
cnchu@iastate.edu

ABSTRACT

To minimize the effect of process variation for a design in triple patterning lithography (TPL), it is beneficial for all standard cells of the same type to share a single coloring solution. In this paper, we investigate the TPL-aware detailed placement refinement problem under these coloring constraints. Given an initial detailed placement, the positions of standard cells are perturbed and a TPL solution complying with the coloring constraints is derived while minimizing cell displacement, lithography conflicts and stitches. We prove that this problem is NP-complete and show that it can be formulated as a mixed integer linear program. Since mixed integer linear programming is very time consuming, we propose an effective heuristic algorithm. In our approach, important adjacent pairs of standard cells are recognized firstly, since they have significant impact on cell displacement. Then a tree-based heuristic is applied to generate a good initial solution for our linear programming-based refinement. Experimental results show that compared with mixed integer linear programming, our heuristic approach is comparable in solution quality while using very short CPU runtime.

1. INTRODUCTION

With the technology node scaling to sub-16nm, electron beam (E-beam), extreme ultraviolet lithography (EUVL) and TPL are considered the most promising lithography technologies. In this paper, we are focusing on TPL.

There are many previous works on TPL optimization. The fundamental problem of TPL is to eliminate lithography conflicts while minimizing stitch count. [1–8] are related to TPL layout decomposition. [1–4] focus on 2-Dimension layout decomposition. [5, 6] focus on row-based 1-Dimension layout decomposition. [9, 10] consider TPL during detailed routing stage.

Recently, [11] presents a TPL aware detailed placement approach in which layout decomposition and placement are resolved simultaneously. The approach is effective in resolving lithography conflicts. However, the approach only considers the optimization of wirelength together with lithography conflicts and stitch number. It is not clear how to incorporate other placement objectives like timing and routability.

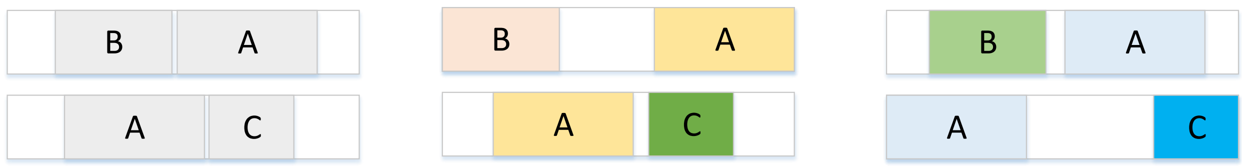
Besides, [6] points out the advantage of assigning the same lithography pattern for the same standard cell type during TPL layout decomposition. This would minimize the effect of process variation and best guarantee that those standard cells of the same type eventually have similar physical and electrical characteristics. However, [6] only considers the decomposition of a fixed layout, and hence often cannot completely satisfy these constraints.

In this paper, we investigate the TPL-aware detailed placement refinement problem under the coloring constraints that all standard cells of the same type should share the same TPL coloring solution. Given an initial detailed placement, the positions of standard cells are perturbed and a TPL solution complying with the coloring constraints is derived while minimizing total cell displacement, lithography conflicts and stitches simultaneously.

Different from [11], our approach is applied to an optimized detailed placement under any conventional placement metrics. By refining it with minimal perturbation, the quality of the detailed placement can be preserved. In addition, we consider the coloring constraints. Compared with [6], as placement perturbation is allowed, the coloring constraints are always satisfied in our approach. We prove that this problem is NP-complete and show that it can be formulated as a mixed integer linear program (MILP). Since the MILP is time consuming to solve, we propose an effective heuristic algorithm to solve it. In our algorithm, important adjacent pairs of standard cells are recognized firstly, since they have significant impact on cell displacement. Then a tree-based heuristic is applied to generate a good initial solution which is then refined by a linear programming (LP)-based technique. Experimental results show that compared with MILP solution, the heuristic method is comparable in solution quality while using very limited CPU runtime. The contributions of this paper are summarized as follows.

- We formulate a new TPL optimization problem considering TPL coloring constraints for standard cells during detailed placement.
- We prove that this new problem is NP-complete.
- We propose a MILP formulation for this new problem.
- Since MILP is very time consuming to solve, we propose an effective heuristic algorithm.

ISPD'15 March 29–April 1, 2015, Monterey, CA, USA.



(a) Given initial detailed placement. (b) One solution: try to optimize the displacement of the second row. (c) Another solution: try to optimize the displacement of the first row.

Figure 1: An instance of problem: choosing different coloring solutions for types A, B and C plus cell shifting.

The rest of paper is organized as follows. In Section 2, we gives the formal problem definition and its MILP formulation. In Section 3, we prove that this problem is NP-complete. In Section 4, we illustrate the heuristic algorithm. In Section 5, we present the experimental results. Finally, we make our conclusions in Section 6.

2. PROBLEM DEFINITION

Given a standard cell library, all feasible coloring solutions for each cell type are found out firstly. Since each cell contains only a small number of layout features, the enumerative approach proposed in [11] works well. Besides, this step is performed once per library. For the i-th type of cell denoted by t_i, there are n_i feasible coloring solutions $p_i^1, p_i^2, \cdots, p_i^{n_i}$. The corresponding stitch counts are $s_i^1, s_i^2, \ldots, s_i^{n_i}$. The width of t_i is w_i. There are k types of standard cells in the library. Given a detailed placement, which has n rows. For the j-th row, the types of standard cells ordered from left to right are $c_j^1, c_j^2, \cdots, c_j^{r_j}$, where r_j is the number of cells in the j-th row.

The TPL-aware displacement-driven detailed placement with coloring constraints is defined as follows.

Given a standard cell library with a set of feasible coloring solutions for each standard cell type, and an initial detailed placement, eliminate all lithography conflicts by choosing one coloring solution for each type of standard cell and shifting the standard cells without changing the cell ordering in each row. The objective is to minimize the total cell displacement and the number of stitches.

Fig. 1 gives an instance of this problem. By choosing coloring solutions for types A, B and C and shifting cells, conflicts are eliminated. In Fig. 1(a), an initial detailed placement with two rows is given. In Fig. 1(b), cell displacement of the second row is optimized well while that of the first row is not. On the contrary, in Fig. 1(c), cell displacement of the first row is optimized well while that of the second row is not. It shows that different TPL solutions may lead to significantly different cell distribution in each row.

2.1 MILP formulation

The above problem can be formulated as a MILP. We use a binary variable b_i^j to denote whether the coloring solution p_i^j is assigned to standard cell type t_i. In the i-th row, the original central x-coordinates of cells ordered from left to right are $o_i^1, o_i^2, \cdots, o_i^{r_i}$, their new central x-coordinates are $x_i^1, x_i^2, \cdots, x_i^{r_i}$, their displacement are $q_i^1, q_i^2, \cdots, q_i^{r_i}$. For any two adjacent cells, the type of left one is t_i and its coloring solution is p_i^u, the type of right one is t_j and its coloring solution is p_j^v. To avoid lithography conflict, the minimal distance between these two cells is a constant denoted by $d_{i,j}^{u,v}$. For any two adjacent cells in the row i, let x_i^{j-1} and x_i^j be their central x-coordiniates, their actual distance is denoted by z_i^j. Besides, the width W of placement region is also given. The problem can be formulated into the following mathematical programming. Note that in this paper, for any pair of adjacent cells, the distance is from the center of the left one to the center of the right one.

$$\text{Minimize: } \alpha \sum_{i=1}^{n} \sum_{j=1}^{r_i} \sum_{k=1}^{n_{c_i^j}} b_{c_i^j}^k \times s_{c_i^j}^k + \beta \sum_{i=1}^{n} \sum_{j=1}^{r_i} q_i^j$$

Subject to:

$$\sum_{j=1}^{n_i} b_i^j = 1, \forall 1 \le i \le k$$

$$x_i^j - x_i^{j-1} = z_i^j, \forall 1 \le i \le n \land 2 \le j \le r_i$$

$$z_i^j \ge \sum_{u=1}^{n_{c_i^{j-1}}} \sum_{v=1}^{n_{c_i^j}} b_{c_i^{j-1}}^u \times b_{c_i^j}^v \times d_{c_i^{j-1}, c_i^j}^{u,v}, \forall 1 \le i \le n \land 2 \le j \le r_i$$

$$x_i^j - o_i^j \le q_i^j, \forall 1 \le i \le n \land 1 \le j \le r_i$$

$$o_i^j - x_i^j \le q_i^j, \forall 1 \le i \le n \land 1 \le j \le r_i$$

$$x_i^j \ge \frac{w_{c_i^j}}{2}, \forall 1 \le i \le n \land 1 \le j \le r_i$$

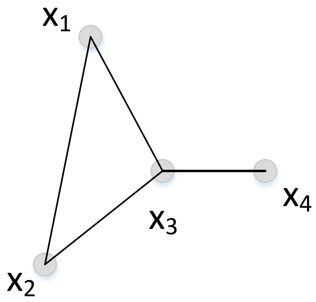
$$x_i^j \le W - \frac{w_{c_i^j}}{2}, \forall 1 \le i \le n \land 1 \le j \le r_i$$

$$b_i^j = 0 \quad or \quad 1, \forall 1 \le i \le k \land 1 \le j \le n_i$$

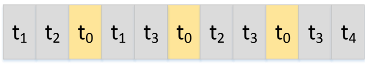
The objective is a weighted sum of total cell displacement and stitch count. The first constraint represents that standard cells of the same type should have the same coloring solution. The second and third constraints represent that for any two adjacent cells, there is enough distance to avoid lithography conflict. The fourth and fifth constraints represent cell displacement. Finally, the last two constraints mean that cells should be put inside of placement region. The product of two binary variables in the third constraint can be transformed into linear constraints as follows: $c = a * b \Leftrightarrow a + b - c \le 1 \land a - c \ge 0 \land b - c \ge 0$, where a, b, c are all binary variables. Therefore, the problem can be formulated as a MILP.

3. COMPLEXITY OF PROBLEM

To see the complexity of this problem, let us look at a special version of its decision problem firstly.



(a) An instance of 3-coloring problem. The three colors are RED, BLUE and GREEN.



(b) An instance of single-row version. The widths of cells are 1. The width of row is 11. For any type of standard cell t_i, it has three feasible coloring solutions (p_i^1, p_i^2, p_i^3). p_i^1, p_i^2 and p_i^3 are respectively corresponding to RED, BLUE and GREEN.

Figure 2: The reduction from 3-coloring problem to single-row version.

Definition 1 (Single-row version). *The given initial detailed placement has only one row. The problem is to decide whether there is a feasible solution to accommodate all cells without conflicts.*

Theorem 1. *The single-row version is NP-complete.*

Proof. It is easy to see that the single-row version is NP. We show that the 3-coloring problem can be reduced to single-row version. Since the 3-coloring is NP-complete [12], the single-row version is NP-complete.

Suppose in a 3-coloring problem instance, there are n nodes denoted by $x_1, x_2, \cdots, x_n$. There are m edges denoted by $e_1, e_2, \cdots, e_m$. We can construct the following single-row version instance.

Each node x_i is corresponding to one type of standard cell t_i, which has three feasible coloring solutions p_i^1, p_i^2, p_i^3. p_i^1, p_i^2 and p_i^3 are corresponding to RED, BLUE and GREEN respectively. There is a special type of standard cell t_0. The width of standard cells are all 1.

We define the minimal distance between t_i and t_j to eliminate conflict as follows.

$$d_{i,j}^{u,v} = \begin{cases} 1 & \text{if } u \neq v \text{ and } i \neq 0 \text{ and } j \neq 0 \\ 2 & \text{if } u = v \text{ and } i \neq 0 \text{ and } j \neq 0 \\ 1 & \text{if } i = 0 \text{ or } j = 0 \end{cases}$$

It means that for any pair of adjacent cells, if the type of either one is t_0, the minimal distance between these two cells to avoid conflict is 1 no matter what the final coloring solutions are. Otherwise, if the left one is assigned the coloring solution which is corresponding to p_i^k $(1 \leq k \leq 3)$ and the right one is assigned the coloring solution which is corresponding to p_j^k, the minimal distance between these two cells to avoid lithography conflict is 2. Otherwise the minimal distance is 1.

For any two nodes x_i and x_j, suppose $i < j$ without loss of generality. If there is an edge $e = (x_i, x_j)$, then we construct a pair of adjacent cells (t_i, t_j). Besides, we add a standard cell of type t_0 between any two pairs of constructed adjacent cells. And the width of row is defined as the number of constructed standard cells, i.e., 3m-1. Fig.2 (b) shows the corresponding single-row version instance of the 3-coloring problem instance in Fig.2 (a).

If the above 3-coloring problem instance is true, then in the constructed single-row version instance, for any two adjacent cells t_i and t_j $(i < j)$, we can choose the coloring solutions so that the minimal distance between these two cells to avoid lithography conflict is 1. Therefore, all the constructed standard cells can be put inside of the row. Similarly, if single-row version instance is true, then we can find a solution that satisfies the corresponding 3-coloring problem instance.

□

The displacement-driven TPL-aware detailed placement with ordering and coloring constraints is a generalization of the single-row version, so it is also NP-complete [12].

4. METHODOLOGY

Since the problem is NP-complete and MILP is very time consuming, we propose an effective heuristic algorithm to solve this problem. In this section, we firstly show the motivation of our approach. Next, we present its overview which is composed of three stages. Finally, we illustrate these three stages respectively.

4.1 Motivation

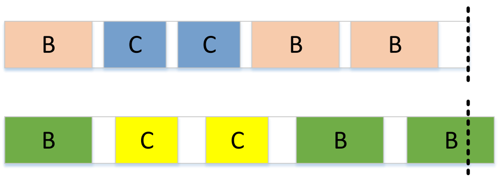
Since standard cells of the same type should have the same coloring solution, we define adjacent pair as follows.

Definition 2. *An adjacent pair is a pair of types of two adjacent standard cells.*

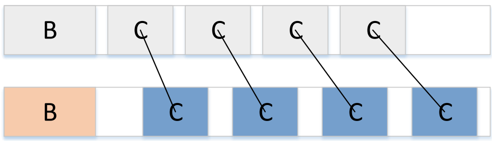
For example, if the type of left cell is t_i and the type of right one is t_j, the corresponding adjacent pair is (t_i, t_j). The minimal distances of adjacent pairs to avoid lithography conflicts have significant impact on solution quality of this problem. There are two reasons. Firstly, if these minimal distances are not optimized well, then it would be difficult to put all cells inside of the row region, as shown in Fig. 3(a). Secondly, different adjacent pairs have different impact on total cell displacement, as shown in Fig. 3(b). Therefore, our method tries to focus on the minimal distances of important adjacent pairs.

4.2 Overview

Our approach is composed of three stages. In the first stage, we propose a method to recognize the important adjacent pairs. In the second stage, we try to optimize minimal distances of important adjacent pairs and a tree-based



(a) The upper figure represents that all the cells are put inside of row if the minimal distances (to eliminate lithography conflicts) of adjacent pairs are optimized well. On the contrary, in the lower figure, the right most cell B is outside of row if those minimal distances of adjacent pairs are not optimized well.



(b) In the upper figure which represents the original placement, the left-most adjacent pair (cell B and cell C) is the most important one to optimize cell displacement. If the minimal distance of this pair to eliminate conflict is not optimized well, all the other cells on the right hand side would be shifted right as shown in lower figure.

Figure 3: The two examples reveal the motivation of our heuristic approach.

heuristic is applied to get a good initial solution. In the last stage, we apply LP-based method to refine the solution. The overview is presented in Fig. 4.

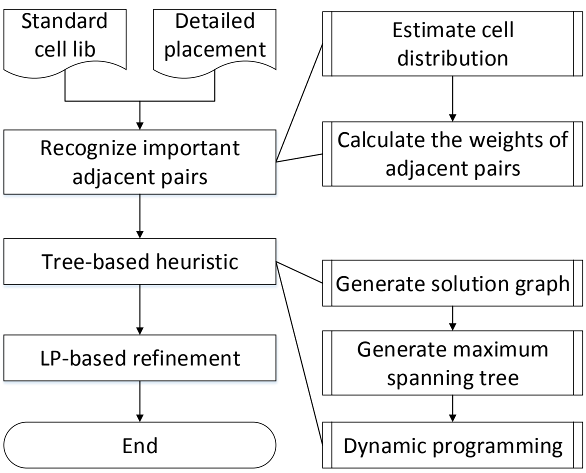


Figure 4: The overview of our heuristic approach.

4.3 Important adjacent pair recognition

We use a positive integer to represent how important an adjacent pair is. We call this integer the weight of adjacent pair. Higher weight means more important. For example, as shown in Fig. 3(b), apparently, the adjacent pair (B, C) should have the highest weight. We use $weight[i][j]$ to denote the weight of adjacent pair (t_i, t_j).

At this stage, we do not know what the final coloring is. Therefore, we propose a simple method to estimate the new cell distribution. For any adjacent pair (t_i, t_j), we calculate the average minimal distance $d_{i,j}^{ave}$ to avoid lithography conflict. This value is given by the following formula.

$$d_{i,j}^{ave} = \frac{\sum_{u=1}^{n_i} \sum_{v=1}^{n_j} d_{i,j}^{u,v}}{n_i * n_j}$$

The minimal total cell displacement can be achieved by LP as follows.

$$\text{Minimize: } \sum_{i=1}^{n} \sum_{j=1}^{r_i} q_i^j$$

Subject to:

$$x_i^j - x_i^{j-1} \geq d_{c_i^{j-1}, c_i^j}^{ave}, \forall 1 \leq i \leq n \wedge 2 \leq j \leq r_i$$
$$x_i^j - o_i^j \leq q_i^j, \forall 1 \leq i \leq n \wedge 1 \leq j \leq r_i$$
$$o_i^j - x_i^j \geq q_i^j, \forall 1 \leq i \leq n \wedge 1 \leq j \leq r_i$$
$$x_i^j \geq \frac{w_{c_i^j}}{2}, \forall 1 \leq i \leq n \wedge 1 \leq j \leq r_i$$
$$x_i^j \leq W - \frac{w_{c_i^j}}{2}, \forall 1 \leq i \leq n \wedge 1 \leq j \leq r_i$$

Then we define shifting direction of standard cell below.

DEFINITION 3. *For the j-th standard cell in row r_i, its shifting direction is left if $x_i^j < o_i^j$, and right if $x_i^j > o_i^j$, otherwise no shifting. We use ↰ to denote left shifting, ↱ for right shifting, and $=$ for no shifting.*

Algorithm 1 gives the method to calculate the weights of adjacent pairs. The idea is that for a pair of adjacent cells, if their minimal distance to eliminate conflict is increased, the weight of this pair would roughly reflect the increment of total cell displacement. Let us look at an example. A placement row contains six cells and five adjacent pairs. The shifting directions of these six cells are ↱, ↱, ↱, ↱, ↱, ↱. The five adjacent pairs' weights ordered from left to right are respectively 5, 4, 3, 2 and 1. The weight of the left-most one is 5, because if its minimal distance is increased by 1 unit, the total cell displacement would be increased by 5 units roughly.

4.4 Tree-based heuristic

After the weights of all adjacent pairs are computed, a solution graph can be constructed as follows. In the solution graph, each node represents a standard cell type. The edge between two nodes represents an adjacent pair.

Let f_i be the coloring solution that standard cell type t_i uses. The cost $cost_i$ of node t_i and the cost $cost_{i,j}$ of edge connecting t_i and t_j in the solution graph are defined as follows.

$$cost_i[f_i] = \beta * weight[i][i] * d_{i,i}^{f_i, f_i} + \alpha * s_i^{f_i}$$
$$cost_{i,j}[f_i, f_j] = \beta * [weight[i][j] * d_{i,j}^{f_i, f_j} + weight[j][i] * d_{j,i}^{f_j, f_i}]$$

The purpose of our tree-based heuristic is to find the coloring solution for each standard cell type, so that the total cost including cost of nodes and edges in the solution graph is minimized. It is not hard to see that if solution graph is

Algorithm 1 Method to calculate the weights of adjacent pairs

Calculate $d_{i,j}^{ave}$ for each pair of adjacent pair (t_i, t_j);
Solve the LP to get the shifting direction of each standard cell;
for each placement row **do**
  [start, end] is the index range of cells (in ascending order of their x-coordinate) in this row;
  for any adjacent pair $P = (t_i, t_j)$ in the row **do**
    ll and rr are the indexes of t_i and t_j in the row;
    if the left cell is $\Lsh$ **then**
      for k from ll to $start$ **do**
        if the cell whose order is k is $\Lsh$ or $=$ **then**
          $weight[i][j] += 1$;
        else
          break;
        end if
      end for
    end if
    if the right cell is $\Rsh$ **then**
      for k from $ll + 1$ to end **do**
        if the cell whose order is k is $\Rsh$ or $=$ **then**
          $weight[i][j] += 1$;
        else
          break;
        end if
      end for
    end if
  end for
end for

of a tree structure, then dynamic programming can be applied to get the optimal coloring solution. Fortunately, it is observed that solution graphs for industrial benchmarks are sparse graphs. Next, we propose a method to leverage this observation.

4.4.1 Maximum spanning tree generation

The basic idea to leverage the observation is to ignore some relatively less important adjacent pairs and turn the solution graph into a tree. The cost of each edge connecting t_i and t_j in solution graph is replaced by $cost'_{i,j} = \alpha * [weight[i][j] * (d_{i,j}^{max} - d_{i,j}^{min}) + weight[j][i] * (d_{j,i}^{max} - d_{j,i}^{min})]$, where $d_{i,j}^{max}$ and $d_{i,j}^{min}$ are defined as follows.

$$d_{i,j}^{max} = \max_{1 \le u \le n_i} \max_{1 \le v \le n_j} d_{i,j}^{u,v}$$
$$d_{i,j}^{min} = \min_{1 \le u \le n_i} \min_{1 \le v \le n_j} d_{i,j}^{u,v}$$

It is easy to see that for any edge connecting t_i and t_j, if $cost'_{i,j}$ is small, then no matter what the final coloring solutions for t_i and t_j are, the cost of this edge in the solution graph is similar. Therefore, we use maximum spanning tree to replace the original solution graph. Note that, $cost'_{i,j}$ is only used during generating maximum spanning tree rather than the following dynamic programming.

4.4.2 Dynamic programming solution

After maximum spanning tree is generated, dynamic programming could be applied to find an initial coloring solution. We use the node which has maximal out-degree as the root to generate the tree topology. Then bottom-up method is adopted to construct optimal solutions in the tree. For any node t_i, we maintain a vector $Best[i]$. The entry $Best[i][j]$ stores the best cost over all possible coloring solutions for the sub-tree rooted at node t_i if t_i is choosing coloring solution p_i^j. Suppose it has m children $(x_1, x_2, \cdots, x_m)$, and the vectors for these m children have already been constructed. The vector for t_i can be constructed by the following formula. The final total cost is the minimal element of $Best[i]$ if t_i is the root of the tree.

$$Best[i][j] = cost_i[p_i^j] + \sum_{1 \le p \le m} \min_{1 \le z \le n_{x_p}} \left(Best[x_p][z] + cost_{i,x_p}[p_i^j, p_{x_p}^z] \right)$$

4.5 LP-based refinement

The LP-based refinement technique is presented in Algorithm 2. The idea is that we enumerate all the coloring solutions for one standard cell type while others are fixated. The node whose associated edges' costs are larger is given a higher priority. In Line 4 of Algorithm 2, once the coloring solutions for all the cells are fixed, it is easy to see that minimal cell displacement can be achieved by solving the following LP, where $d_{c_i^{j-1}, c_i^j}$ is the minimal distance to eliminate conflict for adjacent cells c_i^{j-1} and c_i^j in the i-th row.

Minimize: $\sum_{i=1}^{n} \sum_{j=1}^{r_i} q_i^j$

Subject to:

$$x_i^j - x_i^{j-1} \ge d_{c_i^{j-1}, c_i^j}, \forall 1 \le i \le n \wedge 2 \le j \le r_i$$
$$x_i^j - o_i^j \le q_i^j, \forall 1 \le i \le n \wedge 1 \le j \le r_i$$
$$o_i^j - x_i^j \ge q_i^j, \forall 1 \le i \le n \wedge 1 \le j \le r_i$$
$$x_i^j \ge \frac{w_{c_i^j}}{2}, \forall 1 \le i \le n \wedge 1 \le j \le r_i$$
$$x_i^j \le W - \frac{w_{c_i^j}}{2}, \forall 1 \le i \le n \wedge 1 \le j \le r_i$$

Algorithm 2 LP-based refinement

1: Calculate the associated edges' costs of each node;
2: **for** each node in descending order of associated edges' costs **do**
3:   **for** each coloring solution for this node **do**
4:     Minimize the total cell displacement by solving the LP in Section 4.5;
5:     **if** the value of cost function is better than the current best **then**
6:       Update the current best;
7:       Update the coloring solution for this node.
8:     **end if**
9:   **end for**
10: **end for**

5. EXPERIMENTAL RESULTS

Our approach is implemented in C++ on a Linux server with Intel Xeon X5550 2.67GHz CPU, 94GB main memory. The benchmarks are derived from [11]'s. Gurobi [13] is used to solve MILP and LP. Since the problem is NP-complete and it cannot be expected to get the optimal solutions for some benchmarks within limited CPU runtime. We limit the MILP solver to run 7200s and report the best solutions within the time limit of MILP solver.

The experimental results are shown in Table I. Compare with MILP solutions, our heuristic approach achieves the same number of stitches. For total cell displacement, the heuristic method is only 2.9% worse than that of MILP solutions on average. However, the heuristic method gets 207× speed up on average. Besides, our method only increases wirelength by less 1% over the initial detailed placement.

Table 1: Experiment results: MILP V.S. Heuristic.

benchmark	MILP				Heuristic				
	displacement	# of conflicts	# of stitches	runtime(s)	displacement	# of conflicts	# of stitches	WL increase	runtime(s)
alu-70	2.88E+05	0	610	1245	2.94E+05	0	610	0.6%	12
alu-80	6.76E+05	0	610	7200	6.87E+05	0	610	1.4%	14
alu-90	1.94E+06	0	610	7200	1.97E+06	0	610	4.0%	15
byp-70	1.04E+05	0	1134	739	1.04E+05	0	1134	0.0%	21
byp-80	3.85E+05	0	1134	7200	3.68E+05	0	1134	0.1%	28
byp-90	1.54E+06	0	1134	7200	1.60E+06	0	1134	0.7%	31
div-70	1.60E+05	0	1316	3042	1.60E+05	0	1316	0.1%	28
div-80	3.53E+05	0	1316	7200	3.64E+05	0	1316	1.7%	35
div-90	3.62E+06	0	1316	7200	3.61E+06	0	1316	3.8%	32
ecc-70	2.76E+04	0	258	13	2.90E+04	0	258	0.0%	4
ecc-80	8.91E+04	0	258	11	1.09E+05	0	258	0.1%	5
ecc-90	3.55E+05	0	258	23	3.55E+05	0	258	0.9%	6
efc-70	2.84E+04	0	671	420	3.15E+04	0	671	0.0%	6
efc-80	1.14E+05	0	671	4127	1.16E+05	0	671	0.3%	8
efc-90	5.95E+05	0	671	4800	6.00E+05	0	671	2.4%	8
ctl-70	4.55E+04	0	275	351	4.89E+04	0	275	0.0%	10
ctl-80	1.38E+05	0	275	4345	1.40E+05	0	275	0.0%	12
ctl-90	3.49E+05	0	275	7200	3.50E+05	0	275	0.6%	13
top-70	4.95E+05	0	4731	3165	5.12E+05	0	4731	0.0%	326
top-80	1.48E+06	0	4731	7200	1.51E+06	0	4731	0.2%	391
top-90	7.36E+05	0	4731	7200	7.19E+05	0	4731	0.1%	482
Norm.	0.971	1	1	207	1.000	1	1	0.8%	1

6. CONCLUSIONS

In this paper, we are focusing on displacement-driven TPL optimization in detailed placement stage under coloring constraints. We recognize this problem as NP-complete, then propose two solutions. The first one is MILP, the other is heuristic approach. We show that the heuristic approach is very efficient compared with MILP by experiment. The proposed heuristic method can produce competitive solution quality within very limited CPU runtime.

References

[1] B. Yu, K. Yuan, B. Zhang, D. Ding, and D. Z. Pan, "Layout decomposition for triple patterning lithography," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 1–8, 2011.

[2] S.-Y. Fang, Y.-W. Chang, and W.-Y. Chen, "A novel layout decomposition algorithm for triple patterning lithography," in *Proceedings of the 49th Annual Design Automation Conference*, pp. 1185–1190, 2012.

[3] J. Kuang and E. F. Y. Young, "An efficient layout decomposition approach for triple patterning lithography," in *Proceedings of the 50th Annual Design Automation Conference*, pp. 69–75, 2013.

[4] Y. Zhang, W.-S. Luk, H. Zhou, C. Yan, and X. Zeng, "Layout decomposition with pairwise coloring for multiple patterning lithography," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 170–177, 2013.

[5] H. Tian, H. Zhang, Q. Ma, Z. Xiao, and M. D. F. Wong, "A polynomial time triple patterning algorithm for cell based row-structure layout," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 57–64, 2012.

[6] H. Tian, Y. Du, H. Zhang, Z. Xiao, and M. D. F. Wong, "Constrained pattern assignment for standard cell based triple patterning lithography," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 178–185, 2013.

[7] B. Yu, Y.-H. Lin, G. Luk-Pat, D. Ding, K. Lucas, and D. Z. Pan, "A high-performance triple patterning layout decomposer with balanced density," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 163–169, 2013.

[8] Z. Chen, H. Yao, and Y. Cai, "Suald: Spacing uniformity-aware layout decomposition in triple patterning lithography.," in *ISQED*, pp. 566–571, 2013.

[9] Q. Ma, H. Zhang, and M. D. F. Wong, "Triple patterning aware routing and its comparison with double patterning aware routing in 14nm technology," in *Proceedings of the 49th Annual Design Automation Conference*, pp. 591–596, 2012.

[10] Y.-H. Lin, B. Yu, D. Z. Pan, and Y.-L. Li, "Triad: A triple patterning lithography aware detailed router," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 123–129, 2012.

[11] B. Yu, X. Xu, J.-R. Gao, and D. Z. Pan, "Methodology for standard cell compliance and detailed placement for triple patterning lithography," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 349–356, 2013.

[12] M. R. Garey and D. S. Johnson, *A Guide to the Theory of NP-Completeness*. Macmillan Higher Education, 1979.

[13] "Gurobi." `http://www.gurobi.com`.