

Pin Accessibility-Driven Detailed Placement Refinement

Yixiao Ding
Department of Electrical and
Computer Engineering
Iowa State University
Ames, IA, 50011, USA
yxding@cadence.com

Chris Chu
Department of Electrical and
Computer Engineering
Iowa State University
Ames, IA, 50011, USA
cnchu@iastate.edu

Wai-Kei Mak
Department of Computer
Science
National Tsing Hua University
Hsinchu, Taiwan 30013
wkmak@cs.nthu.edu.tw

ABSTRACT

The significantly increased number of routing design rules at sub-20nm nodes has made pin access one of the most critical challenges in detailed routing. Resolving pin access issues in detailed routing stage may be too late due to the fixed pin locations, especially in the area with high pin density. In placement stage when cell movement is allowed, the consideration of pin access has more flexibility. We propose a refinement stage after detailed placement to improve pin access. To respect the given placement solution, the refinement techniques are restricted to cell flipping, same-row adjacent cell swap, and cell shifting. A cost function is presented to model pin access for each pin-to-pin connection. Based on the cost function, two phases are proposed to improve pin access for all the connections simultaneously. In the first phase, we refine the placement by cell flipping and same-row adjacent cell swap. The problem is solved by dynamic programming row by row. In the second phase, only cell shifting is used, and a linear program is formulated to further refine the placement. Experimental results demonstrate that the proposed detailed placement refinement can improve pin access and reduce unroutable nets by about 33% in the detailed routing stage.

1. INTRODUCTION

With increasing number of design rules in advanced technology nodes, detailed routing (DR) is becoming more and more difficult. Pin access is one of the most critical problems [1, 2]. To alleviate pin access difficulty, the choice of tapping point is important during DR. Fig. 1 shows the pin access issue for a standard cell containing three pins a , b , and c . Each pin has several tapping points for via insertion to connect to a metal 2 wire segment. Suppose pins a , b , and c belong to nets A , B , and C respectively, and connection direction for each pin is pre-determined by global routing (GR) or steiner tree. DR is performed in the order of $A \rightarrow B \rightarrow C$. In Fig. 1(b), after routing nets A and B , all tapping points of pin c are blocked by other routes. There is no way to connect pin c to a metal 2 wire segment, which

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ISPD '17, March 19-22, 2017, Portland, OR, USA

© 2017 ACM. ISBN 978-1-4503-4696-2/17/03...\$15.00

DOI: <http://dx.doi.org/10.1145/3036669.3036679>

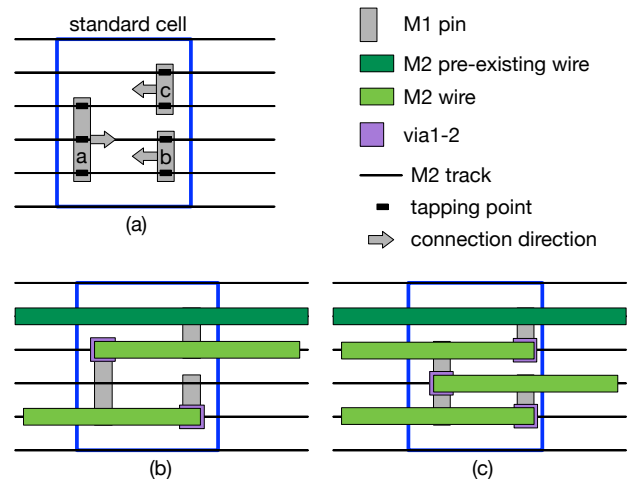


Figure 1: Detailed routing around pins in a standard cell. (a) A standard cell with three pins. (b) Pin C cannot be accessed because all its tapping points are blocked. (c) A wise choice of tapping points makes all pins accessible.

leads to a pin access failure. With a wise choice of tapping points as shown in Fig. 1(c), all three pins can be accessed in DR.

However, choosing tapping points wisely during DR is not always sufficient due to the fixed cell placement. In the area with high pin density, pin access may still be impossible even with a careful choice of tapping points. As shown in Fig. 2(a), the standard cell SC_a is placed abutting to the left boundary of standard cell SC_b . Thus, pins are very close to each other, especially for pins D and E , which potentially increase pin access difficulty. Fig. 2(b) shows DR around the two standard cells. Both of pin E 's tapping points are blocked by metal 2 wire segments, which used to access pin D and pin E . As a result, pin access for pin E is not possible.

To further improve pin accessibility, we propose a refinement stage after the detailed placement (DP) stage, in which small perturbation of a given DP is allowed. Fig. 3 shows the three possible options of refining the placement to improve pin access. In Fig. 3(a), cell SC_b shifts to the right to make some space between SC_a and SC_b . Then, a via can be inserted at the top tapping point of pin D , and there is enough space to form a metal 2 wire segment from the tapping point. By utilizing both metal 2 and metal 3 wire segments, pin access for pin D is not the bottleneck

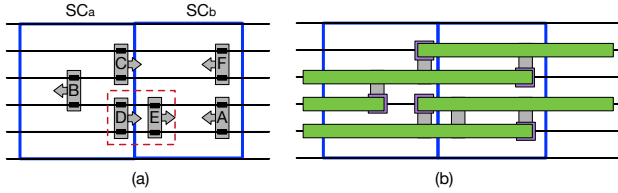


Figure 2: Enhance pin accessibility in DR. (a) Pin access becomes harder within area with high pin density. (b) Pin E cannot be accessed even with careful choice of tapping points

in DR anymore. In Fig. 3(b), cell SC_b is flipped, and all pins can be easily accessed during DR. Finally, in Fig. 3(c), two cells are swapped, and pins are better distributed with more space in between than in Fig. 2(a). Thus, pin access becomes easier and DR can be completed with less efforts. From the examples above, we observe pin access can be effectively improved by cell shifting, cell flipping, and adjacent cell swap. Thus, we propose a detailed placement refinement stage which directly targets at pin accessibility enhancement using these three approaches.

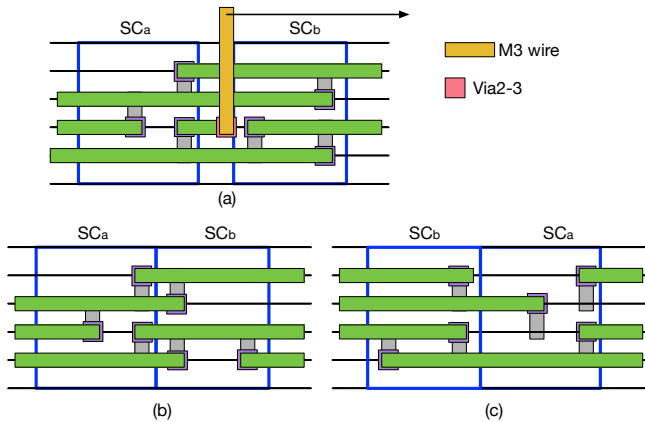


Figure 3: Three approaches to enhance pin accessibility in DP (a) Cell shifting. (b) Cell flipping. (c) Adjacent cell swap.

Pin access is considered in DR stage [3, 4, 5, 6, 7, 8]. [3, 4, 5, 6] proposed standard cell-level pin access planning under SADP lithography constraints. [7] addressed offgrid or gridless pin access while [8] tackled escape routing for a dense pin cluster. Pin access is also considered in the GR stage in the form of local routing congestion estimation [9, 10, 11]. However, they only took into account pin information, like pin count, pin shape, and pin's Steiner tree length, which fails to capture the real pin access scenario during DR. Furthermore, in both GR and DR stages, cell placement cannot be changed and all pin locations are fixed. Even with the proposed techniques in works above, pin access may still be impossible which leads to failed connection. To overcome such limitation, detailed routing is considered during placement. The recent ISPD placement contest [12] demonstrates that physical data, like pin geometries, is important, and needs to be considered in placement to improve routability in DR. Furthermore, routing congestion is con-

sidered during DP [13, 14, 15]. The models to estimate congestion in [13, 14] are very rough, especially for local routing congestion, and pin access problem is not directly addressed. [15] included pin information in the cost function to guide detailed placement. However, the pin information are not enough to model pin access in DR accurately.

Our major contributions are summarized as follows:

- It is the first work to directly consider pin access issue in detailed placement stage.
- We propose an accurate model to capture the pin access scenario during detailed routing. A cost function is used to guide the detailed placement refinement.
- The placement refinement operations are limited to cell flipping, adjacent cell swap, and cell shifting. Meanwhile, the proposed algorithm is dynamic programming and linear programming based. Thus, our DP refinement ensures a fast runtime without a big perturbation of the given legalized placement.
- The experimental results demonstrate that by applying the DP refinement, unroutable nets can be significantly reduced in DR. Moreover, the total cell displacement is kept in control and overheads in total wirelength, via count and runtime are low.

The rest of the paper is organized as follows. Section 2 presents some preliminaries. Section 3 is the problem formulation. The overall flow and details of our proposed algorithm are presented in Section 4. Section 5 shows our experimental results, and finally Section 6 concludes the paper.

2. PRELIMINARIES

2.1 Assumptions

In the following, we present our DP refinement approach under a few assumptions. Note that our approach is general and is not limited to these assumptions. It can be easily extended to handle different assumptions.

In typical designs, the majority of pins occur on metal 1 layer, where the available routing resources are extremely limited. Meanwhile, 1D gridded design has become the mainstream in advanced technology nodes [16]. Each metal layer has a routing direction, either horizontal or vertical. Without loss of generality, we assume metal 1 layer is not allowed for routing, metal 2 has horizontal routing direction, metal 3 has vertical routing direction, and so on. We also assume each pin in the cell is either a rectangle strip or a rectilinear shape spanning one or more metal 2 tracks. A tapping point (TP) is defined as the overlap of a metal 2 track and the pin shape, where a via can be inserted to connect the pin to a metal 2 wire segment.

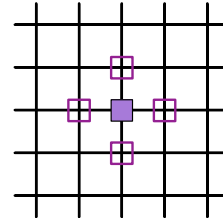


Figure 4: The minimum center-to-center spacing rule in via design rules.

In 1D gridded design, changing the routing direction means switching to another metal layer and via insertion between the layers. Thus, maintaining via design rules in DR is critical to both routing solution quality and layout manufacturing. For example, the minimum center-to-center via spacing is one of the major via design rules [17]. In this paper, the minimum center-to-center via spacing is assumed to be more than one routing pitch, and it is enforced for every via layer. As shown in Fig. 4, an via is inserted on the via layer between metal 1 and metal 2. Thus, the four via locations on the same via layer represented by purple empty squares are forbidden. Note that the assumed via design rules can be easily extended to other complex via design rules, e.g., multiple patterning constraints on via layer pattern [18].

2.2 Pin access region

Pin access is to select a tapping point as the via insertion location to connect the pin to a wire segment on metal 2. The wire segment is preferred to extend in the connection direction in order to move closer to the other pin of this connection. We define a pin access region (PAR) for each pin-to-pin connection of each pin. It is a bounding box, whose height is same as the pin shape and width is same as the horizontal distance between the two connected pins. PAR_{AB} denotes the PAR of connection AB of pin A , and $w_{PAR_{AB}}$ is the width of the PAR_{AB} . Fig. 5 shows totally four PARs of two pin-to-pin connections. The connection AB connects two pins from cells in a same standard cell row, we call it same-row pin-to-pin connection. On the other hand, connection CD is a different-row pin-to-pin connection since the two pins are from cells in different standard cell rows.

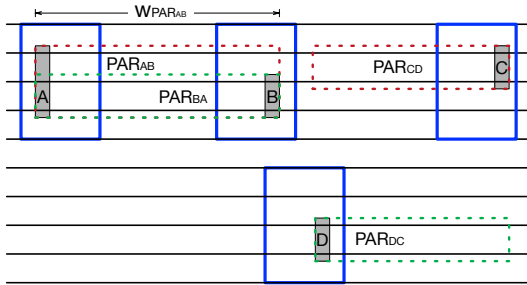


Figure 5: A PAR is defined for each pin-to-pin connection of each pin. Connection AB is a same-row connection while connection CD is a different-row connection.

2.3 Pin access penalty

If the PAR of a connection of a pin is obstructed by an object (e.g., blockage or metal 2 wire segment), the pin access to the pin in this connection is affected negatively. We use a penalty function to quantify the impact on the pin access, which is shown in Fig. 6. In penalty function $f_w(dist)$, input $dist$ is the horizontal distance between the pin and the object, and parameter w is the width of the PAR. For a same-row connection, e.g., connection AB in Fig. 5, it is desirable to have a single metal 2 wire segment to connect the two pins in DR. Fig. 6(a) shows the penalty function for this case. It always outputs the maximum penalty, namely 1, to penalize any object within the PAR, i.e., when $dist$ value

is smaller than w . When $dist$ is larger than w , i.e., the PAR does not intersect with the object, the output value of penalty function is zero. For a different-row connection, e.g., connection CD in Fig. 5, a router has the flexibility to choose a turning point within PAR to switch to metal 3 in DR. Fig. 6(b) shows the corresponding penalty function where min_i is the minimum length value for a metal 2 wire segment. When $dist$ is less than min_i , the space to form a metal 2 wire segment for pin access is occupied by the object. Thus, the penalty function outputs the maximum penalty 1. The penalty decreases with increasing $dist$ since the DR will then has more flexibility in pin access. Mathematically, when $dist \leq w$, $f_w(dist)$ is in the form of $\frac{\alpha}{dist} + \beta$, where parameters α and β are set so that $f_w(min_i) = 1$ and $f_w(w) = 0$. When $dist$ increases such that it is bigger than w , the PAR is not occupied by the object anymore. Thus, $f_w(dist) = 0$ in this case.

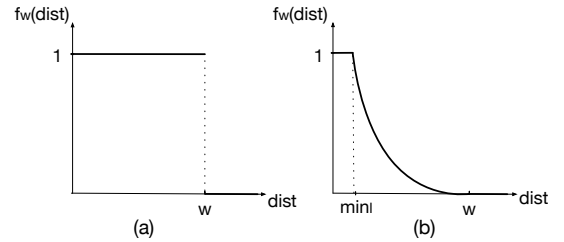


Figure 6: Penalty function $f_w(dist)$ for (a) same-row connection, and (b) different-row connection.

Given a connection of a pin, the pin access penalty (PAP) is a cost imposed on the connection to reflect the impact of an object on the pin access of the connection. The computation of PAP depends on the type of object and how the PAR of the connection intersects with the object. As shown in Fig. 7, there are totally four scenarios. Let's firstly consider the simplest scenario which is shown in Fig. 7(a). The $PAR_{AA'}$ of a connection AA' of pin A is intersected with a metal 2 blockage B . We say that connection AA' is in conflict with blockage B . $dist_{A,B}$ denotes the horizontal distance between pin A and blockage B . A conflict tapping point (CTP) is a TP on a metal 2 track which is obstructed by the blockage B . TP_A denotes the number of TPs of pin A . $CTP_{AA',B}$ denotes the number of CTPs of pin A when connection AA' is in conflict with blockage B . In Fig. 7(a), $CTP_{AA',B} = 2$, and they are highlighted with red color. In the first scenario, the pin access to pin A becomes harder since the routing resources used for pin access are occupied by blockage B . $PAP_{AA'}^B$ is the PAP cost imposed on connection AA' to model the increase in the hardness to access pin A .

$$PAP_{AA'}^B = \frac{CTP_{AA',B}}{TP_A} \times f_{w_{PAR_{AA'}}}(dist_{A,B}) \quad (1)$$

$PAP_{AA'}^B = 0$ if access to pin A in the connection AA' is completely free from the impact of blockage B . $PAP_{AA'}^B = 1$ if the pin access is impossible through all pin A 's TPs. $PAP_{AA'}^B$ consists of two components. One is the probability of the occupied routing resource will actually affect pin access, which is $\frac{CTP_{AA',B}}{TP_A}$. The other is the penalty function to determine the negative impact on pin access, which is $f_{w_{PAR_{AA'}}}(dist_{A,B})$. The penalty function used in the com-

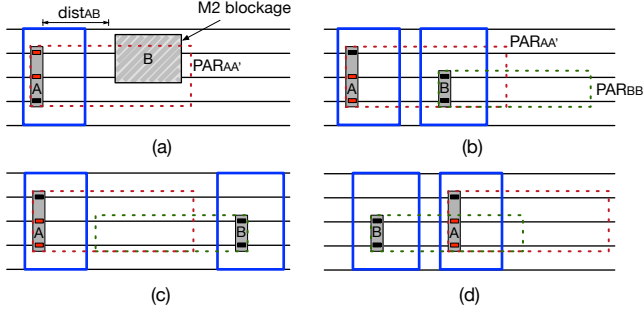


Figure 7: PAP in four scenarios. (a) 1st scenario. (b) 2nd scenario. (c) 3rd scenario. (d) 4th scenario.

putation of $PAP_{AA'}^B$ depends on whether connection AA' is a same-row connection or a different-row connection as described earlier.

In other scenarios, the PAP cost is imposed on the connection AA' when its PAR intersects with the PAR of another connection. As shown in Fig. 7(b)(c)(d), there are three kinds of intersection which lead to the other three scenarios. Suppose the $PAR_{AA'}$ of connection AA' of pin A is intersected with the PAR of connection BB' of pin B . We say that connection AA' is in conflict with the connection BB' . The second scenario is when the connection directions of pin A and pin B are right and pin A is on the left side of pin B , or connection direction of pin A and pin B are left and pin A is on the right side of pin B . The third scenario is when the connection directions of pin A and pin B are different. The fourth scenario is when the connection directions of pin A and pin B are right and pin A is on the right side of pin B , or connection directions of pin A and pin B are left and pin A is on the left side of pin B . $PAP_{AA'}^{BB'}$ is the PAP cost imposed on connection AA' due to conflicting connection BB' . $PAP_{AA'}^{BB'}$ for the three scenarios are shown as follows.

$$PAP_{AA'}^{BB'} = \frac{CTP_{AA'.BB'}}{TP_A \times TP_B} f_{w_{PAR_{AA'}}}(dist_{A,B}) \quad (2)$$

$$PAP_{AA'}^{BB'} = \frac{CTP_{AA'.BB'}}{TP_A \times TP_B} f_{w_{PAR_{AA'}}} \left(\frac{w_{PAR_{AA'}}}{w_{PAR_{AA'}} + w_{PAR_{BB'}}} dist_{A,B} \right) \quad (3)$$

$$PAP_{AA'}^{BB'} = \frac{CTP_{AA'.BB'}}{TP_A \times TP_B} f_{w_{PAR_{BB'}}}(dist_{A,B}) \quad (4)$$

Similar to $PAP_{AA'}^B$, $PAP_{AA'}^{BB'}$ is a product of a probability term and a penalty function. The penalty function used in the computation also depends on the type of connection AA' .

The PAP function computes the PAP cost imposed on a connection due to a conflicting blockage or connection. For each connection AA' , we can compute its total PAP cost $PAP_{AA'}$ imposed by all conflicting blockages and connections as follows.

$$PAP_{AA'} = \sum_{block \in CB} PAP_{AA'}^{block} + \sum_{conn \in CC} PAP_{AA'}^{conn} \quad (5)$$

where CB is the set of conflicting blockages with AA' and CC is the set of conflicting connections with AA' . Furthermore, we define $CPAP_c$ for each cell c as the total PAP cost

imposed on all the connections of all the pins in c .

$$CPAP_c = \sum_{A \in Pin_c} \sum_{AA' \in Conn_A} PAP_{AA'} \quad (6)$$

where $Conn_A$ denotes all the connections of pin A and Pin_c denotes all the pins in c . $CPAP_c$ reflects the pin accessibility for all the connections of all the pins in the cell c . It can be used to evaluate if the cell placement is good in terms of pin access. We are trying to refine the cell placement to improve pin access for all the connections. Thus, we define total cell pin access penalty (TCPAP) as the total PAP cost computed for all the connections in the placement.

$$TCPAP = \sum_{c \in All_Cells} CPAP_c \quad (7)$$

3. PROBLEM FORMULATION

The TCPAP can be used to evaluate the pin accessibility of a detailed placement. Our proposed DP refinement is targeted to improve pin access by minimizing the TCPAP. Below is our formal problem statement.

Given an legalized placement solution, we try to refine the placement solution to enhance pin accessibility and improve routability during detailed routing stage. The refinement techniques are limited to cell flipping, cell shifting, and adjacent cell swap. The objective is to minimize TCPAP while DP perturbation during refinement is kept minimal. Furthermore, the quality of detailed routing solution in terms of total wirelength and via count for refined placement solution should be good. The constraint is the placement solution should still be legal after refinement.

4. PROPOSED REFINEMENT ALGORITHM

4.1 Algorithm framework

Fig. 8 shows the framework of our pin accessibility-driven detailed placement refinement. The input is a legalized detailed placement solution, we target to improve pin access in DR by refining the placement. To keep the detailed placement perturbation minimal, we limit our refinement techniques to cell flipping, same-row adjacent cell swap, and cell shifting. Our DP refinement has two phases. In the first phase, we refine the placement by cell flipping and same-row adjacent cell swap. It is solved by dynamic programming row by row. In the second phase, we further refine placement by cell shifting. It is solved by linear programming. The output is the refined and legalized placement. It is ready for routing, and expected to have better pin access in DR.

4.2 Phase 1: Cell flipping and adjacent cell swap

In this phase, we try to refine an initial placement by cell flipping and adjacent cell swap to minimize TCPAP. To compute $PAP_{AA'}$ for a connection AA' of a pin A from cell c , we make an assumption that the PAP due to the connections conflicting with AA' but not containing a pin in cell c or a cell adjacent to c is not significantly affected by cell flipping and adjacent cell swap. By ignoring those conflicting connections, $PAP_{AA'}$ can be approximated as follows:

$$PAP_{AA'} \approx \sum_{block \in CB} PAP_{AA'}^{block} + \sum_{conn \in CC'} PAP_{AA'}^{conn} \quad (8)$$

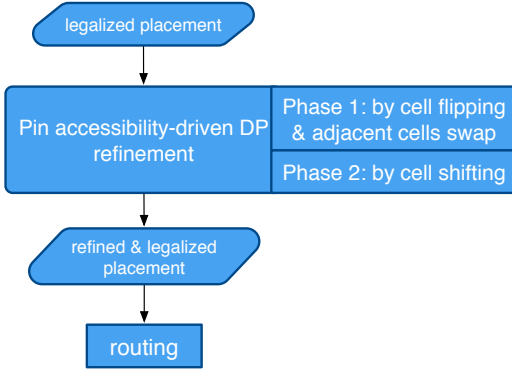


Figure 8: Our framework

where CB is the set of the conflicting blockages with AA' , and CC' is the set of conflicting connections with AA' which contain a pin in cell c or a cell adjacent to c . Based on the above approximation, we can solve the problem by dynamic programming row by row. For each row, the dynamic programming helps to find the optimal cell placement to minimize $\sum_{c \in C_{row}} CPAP_c$ where C_{row} denotes the set of all cells in the row. Given $TCPAP = \sum_{row \in All_Rows} \sum_{c \in C_{row}} CPAP_c$, $TCPAP$ can be minimized. In the following paragraph, we will discuss how an optimal cell placement for a given row of cells is found by dynamic programming.

Given a placement row with n placed cells, we denote the cells as c_1 to c_n according to their order in the original placement. We use c'_k to denote the flipped version of c_k . We use dynamic programming to compute the optimal refined prefix placement of length k for $k = 1, 2, \dots, n$ where refinement by cell flipping and adjacent cell swap is allowed. Let $sol_k(\gamma)$ denotes the optimal refined prefix placement of length k when the cell in the k -th position is γ . Since we only allow cell flipping and adjacent cell swap in this phase, the cell γ in $sol_k(\gamma)$ is either $c_k, c'_k, c_{k-1}, c'_{k-1}, c_{k+1}$, or c'_{k+1} . Let $P_k(\gamma)$ denotes the total CPAP of cells in $sol_k(\gamma)$. To construct $sol_{k+1}(\theta)$, we append a cell θ to the end of $sol_k(\gamma)$, and use $\Delta P(\gamma, \theta)$ to denote the change of total CPAP. Observe that to construct all $sol_{k+1}(\theta)$ where θ could be $c_{k+1}, c'_{k+1}, c_k, c'_k, c_{k+2}$, and c'_{k+2} , six solutions are required including $sol_k(c_k), sol_k(c'_k), sol_k(c_{k-1}), sol_k(c'_{k-1}), sol_k(c_{k+1}),$ and $sol_k(c'_{k+1})$. Specifically, for each case of $sol_{k+1}(\theta)$, we find the corresponding solution above and compute total CPAP after appending θ to the end of solution. The newly constructed solution with minimum total CPAP value is $sol_{k+1}(\theta)$.

Here is an example of how to construct intermediate solutions containing refined prefix placement of length k when $k = 5$. Firstly, to construct $sol_5(c_5)$, we append c_5 at the end of $sol_4(c_4), sol_4(c'_4), sol_4(c_3)$, and $sol_4(c'_3)$, respectively. Then, we compute $P_4(c_4) + \Delta P(c_4, c_5), P_4(c'_4) + \Delta P(c'_4, c_5), P_4(c_3) + \Delta P(c_3, c_5)$, and $P_4(c'_3) + \Delta P(c'_3, c_5)$ for all the newly constructed solutions. The $P_5(c_5)$ is the minimum of the above values, and the newly constructed solution with the minimum of the above values is $sol_5(c_5)$. Similarly, other refined prefix placement of length 5, which could be $sol_5(c'_5), sol_5(c_4), sol_5(c'_4), sol_5(c_6)$, and $sol_5(c'_6)$, can be constructed.

The base cases and recursive formulas of our dynamic program are shown as follows.

Base cases:

$$\{P_1(c_1), P_1(c'_1), P_1(c_2), P_1(c'_2)\}$$

where each case is obtained by computing total CPAP for initial solutions $sol_1(c_1), sol_1(c'_1), sol_1(c_2)$, and $sol_1(c'_2)$.

Recursive formulas:

$$P_k(c_k) = \min \{ P_{k-1}(c_{k-1}) + \Delta P(c_{k-1}, c_k), \\ P_{k-1}(c'_{k-1}) + \Delta P(c'_{k-1}, c_k), \\ P_{k-1}(c_{k-2}) + \Delta P(c_{k-2}, c_k), \\ P_{k-1}(c'_{k-2}) + \Delta P(c'_{k-2}, c_k) \}$$

$$P_k(c'_k) = \min \{ P_{k-1}(c_{k-1}) + \Delta P(c_{k-1}, c'_k), \\ P_{k-1}(c'_{k-1}) + \Delta P(c'_{k-1}, c'_k), \\ P_{k-1}(c_{k-2}) + \Delta P(c_{k-2}, c'_k), \\ P_{k-1}(c'_{k-2}) + \Delta P(c'_{k-2}, c'_k) \}$$

$$P_k(c_{k-1}) = \min \{ P_{k-1}(c_k) + \Delta P(c_k, c_{k-1}), \\ P_{k-1}(c'_k) + \Delta P(c'_k, c_{k-1}) \}$$

$$P_k(c'_{k-1}) = \min \{ P_{k-1}(c_k) + \Delta P(c_k, c'_{k-1}), \\ P_{k-1}(c'_k) + \Delta P(c'_k, c'_{k-1}) \}$$

$$P_k(c_{k+1}) = \min \{ P_{k-1}(c_{k-1}) + \Delta P(c_{k-1}, c_{k+1}), \\ P_{k-1}(c'_{k-1}) + \Delta P(c'_{k-1}, c_{k+1}), \\ P_{k-1}(c_{k-2}) + \Delta P(c_{k-2}, c_{k+1}), \\ P_{k-1}(c'_{k-2}) + \Delta P(c'_{k-2}, c_{k+1}) \}$$

$$P_k(c'_{k+1}) = \min \{ P_{k-1}(c_{k-1}) + \Delta P(c_{k-1}, c'_{k+1}), \\ P_{k-1}(c'_{k-1}) + \Delta P(c'_{k-1}, c'_{k+1}), \\ P_{k-1}(c_{k-2}) + \Delta P(c_{k-2}, c'_{k+1}), \\ P_{k-1}(c'_{k-2}) + \Delta P(c'_{k-2}, c'_{k+1}) \}$$

Finally, we construct $sol_n(c_n), sol_n(c'_n), sol_n(c_{n-1})$, and $sol_n(c'_{n-1})$, and compute their corresponding $P_n(c_n), P_n(c'_n), P_n(c_{n-1})$, and $P_n(c'_{n-1})$ for the given row. The minimum value of $\sum_{c \in C_{row}} CPAP_c$ is equal to $\min\{P_n(c_n), P_n(c'_n), P_n(c_{n-1}), P_n(c'_{n-1})\}$. The optimal refined cell placement for the given row is the solution with the minimum value of $\sum_{c \in C_{row}} CPAP_c$.

4.3 Phase 2: Cell shifting

In this phase, we try to further refine the placement by cell shifting. As mentioned before, cell shifting helps to redistribute the space between the cells, which will potentially improve pin access in DR. We continue to use the proposed PAP function to guide the refinement. However, we need to keep the cell displacement in control to avoid big perturbation of the given legalized placement. We solve this problem by formulating a linear program (LP). Given the refined placement after phase 1, we label all the cells in the i -th row from left to right as $cell_{i1}, cell_{i2}, \dots, cell_{in}$. For $cell_{ij}$, let L_{ij} denote the x coordinate of its bottom left corner, and W_{ij} denote its width. In addition, let LL and RR be the x coordinates of the left and right boundaries of the placement region, respectively. Suppose a pin A in $cell_{ij}$ has a connection AA' . Let bk denote a block which is in conflict with connection AA' . $dist_{A,bk}$ is the distance between A and bk . Similarly, let BB' denote a connection of pin B which is in conflict with connection AA' . $dist_{A,B}$ is the distance between A and B . We use a continuous variable δ_{ij} to represent the shift amount for $cell_{ij}$. To avoid a big amount of shift, we set ΔS as the maximum shift distance for every cell. For the connection AA' , the width of its PAR changes by δ_{ij} during the cell shifting. Meanwhile, both

$dist_{A,bk}$ and $dist_{A,B}$ also change by δ_{ij} . Other parameters in the equations (1)(2)(3)(4) in Section 2.3 to compute PAP remain the same. Hence, the equations to compute PAP become functions of δ_{ij} during cell shifting.

However, the PAP function is non-linear when AA' is a different-row connection. To formulate the problem as a LP, we use linear approximation to approximate the new penalty cost computed by the PAP functions after cell shifting. The linear functions of δ_{ij} used to approximate the PAP functions for connection AA' during cell shifting are shown as follows.

$$\widetilde{PAP}_{AA'}^{bk}(\delta_{ij}) = \alpha \times \delta_{ij} + \beta \quad (9)$$

where $\alpha = \frac{\partial \widetilde{PAP}_{AA'}^{bk}}{\partial \delta_{ij}}|_{\delta_{ij}=0}$ and $\beta = \widetilde{PAP}_{AA'}^{bk}|_{\delta_{ij}=0}$.

$$\widetilde{PAP}_{AA'}^{BB'}(\delta_{ij}) = \alpha' \times \delta_{ij} + \beta' \quad (10)$$

where $\alpha' = \frac{\partial \widetilde{PAP}_{AA'}^{BB'}}{\partial \delta_{ij}}|_{\delta_{ij}=0}$ and $\beta' = \widetilde{PAP}_{AA'}^{BB'}|_{\delta_{ij}=0}$. The δ_{ij} is controlled by ΔS which is usually small in practice (several routing pitches). The approximation is usually accurate enough to be used to compute PAP during cell shifting. Thus, we can approximate TCPAP by a linear function $\sum_{AA' \in C} (\sum_{bk \in CB} \widetilde{PAP}_{AA'}^{bk} + \sum_{BB' \in CC} \widetilde{PAP}_{AA'}^{BB'})$, where CB is a set of blockages in conflict with AA' , CC is a set of connections in conflict with AA' , and C contains all the connections of all the pins in all the cells in the placement. Below is the LP formulation for refinement phase 2.

Objective:

$$Min \sum_{AA' \in C} (\sum_{bk \in CB} \widetilde{PAP}_{AA'}^{bk} + \sum_{BB' \in CC} \widetilde{PAP}_{AA'}^{BB'})$$

Constraints:

C1: For the leftmost cell c_{i1} in row_i ,

$$L_{i1} + \delta_{i1} \geq LL$$

C2: For the rightmost cell c_{in} in row_i ,

$$L_{in} + \delta_{in} + W_{in} \leq RR$$

C3: For two adjacent cells c_{ij} and $c_{i(j+1)}$ in row_i ,

$$L_{ij} + \delta_{ij} + W_{ij} \leq L_{i(j+1)} + \delta_{i(j+1)}$$

C4: For each c_{ij} ,

$$\delta_{ij} \leq \Delta S \text{ and } \delta_{ij} \geq -\Delta S$$

C1 (C2) ensures that the leftmost (rightmost) cell in each row is not shifted out of left (right) boundary. C3 ensures that no two adjacent cells in the same row are overlapped after cell shifting. Finally, C4 makes sure each cell cannot shift more than the pre-set threshold.

5. EXPERIMENTAL RESULTS

We implemented our pin accessibility-driven detailed placement refinement by C++ programming language. All experiments are performed on a machine with 2.4 GHz Intel Core i5 and 8GB memory. Gurobi 6.0.5 is called to solve the LP in refinement phase 2. As mentioned before, this is the first work to directly consider pin access in DP stage. We derive our benchmarks based on the netlist and placement information that the authors of [5] used to construct their detailed routing benchmarks. Every net with m pins is decomposed

into $m - 1$ pin-to-pin connections. Meanwhile, the connection direction of the pin in each connection are determined. In a few standard cells, we found that some pins from metal 1 have only a single TP, and the distance between these TPs is less than the required assumed minimum center-to-center via spacing. Thus, pin access is impossible for these pins due to the constraint of via design rules. Thus, we elongate these pins to increase their TP counts. Table I shows the statistics for each benchmark, including cell count, the number of pin-to-pin connections, and average TP count of all the pins.

Table 1: Statistics of benchmarks

Benchmark	ecc	efc	ctl	alu	div	top
#Cells	1302	1197	1715	1802	3260	12576
#Connections	1615	2872	3308	3261	5847	18618
ave. #TPs	3.02	3.21	3.39	3.28	3.27	3.03

Table II shows the results of our pin accessibility-driven detailed placement refinement. TCPAP is computed for the given legalized placement, placement after refinement phase 1, and placement after refinement phases 1 and 2 respectively. Note that only one iteration of dynamic programming is performed in phase 1 and one iteration of LP is performed in phase 2. Compared with the given placement, the refinement phase 1 can reduce TCPAP by 15% on average over all the benchmarks. In refinement phase 2, we set ΔS to $3 \times p$ in the LP formulation, where p is the pitch size of metal 1 vertical tracks. It can further reduce the TCPAP by another 3%. One of the objectives for our DP refinement is to keep the change of the given placement small. To measure the difference between refined placement and given placement, we also report the average cell displacement and the flipped cell count, which are represented as ‘‘Ave. Disp.’’ and ‘‘#FCs’’ in Table II. The average cell displacement is defined as $\frac{\sum_{c_i \in Cells} |x_i - x'_i|}{Cell \times p}$, where $Cells$ denotes the set of all the cells in the given placement, x'_i denotes c_i 's x coordinate in the given placement, x_i is c_i 's x coordinate after refinement. In phase 1, the cell displacement is due to adjacent cell swap, and on average cells are displaced from their original location by 5.04 metal 1 pitch size. Meanwhile, on average 33.56% of all the cells are flipped after refinement phase 1. After refinement phases 1 and 2, on average, cells are displaced from the original placement by 6.73 metal 1 pitch size. As shown in Table II, our DP refinement is very fast and takes only 13.14 seconds on average over all the benchmarks.

Next we will demonstrate that our pin accessibility-driven DP refinement really improves pin access and reduces unroutable nets in DR. Table III compares the detailed routing solutions for the given placement, the placement after refinement phase 1, and the placement after refinement phases 1 and 2. The state-of-the-art SID-type SADP-aware detailed router from [19] is applied to route each placement. There are totally four layers, where metal 1 layer is not allowed for routing, metal 2 and metal 4 has horizontal routing direction, metal 3 has vertical routing direction. We report in Table III total wirelength, via count, the number of unroutable nets (#UNs), and detailed routing runtime. Compared with the routing for the given placement, the number of unroutable nets can be reduced by 20% on average in DR for placement with refinement phase 1. Meanwhile, the

Table 2: Comparison between the given placement and the placements after refinement

Benchmark	Given placement	After refinement phase 1				After refinement phases 1 and 2		
	TCPAP	TCPAP	Ave. Disp.	#FCs (pct %)	CPU(s)	TCPAP	Ave. Disp.	CPU(s)
ecc	2469.41	2108.51	5.65	500 (38.40)	4.25	2070.43	7.40	5.01
efc	3926.39	3472.32	5.20	438 (36.59)	5.66	3419.43	7.04	6.56
ctl	3327.58	2912.31	5.33	557 (32.48)	6.04	2874.78	7.18	7.00
alu	3409.91	2848.64	3.59	549 (30.47)	5.55	2849.36	5.14	5.80
div	7044.60	6659.68	5.40	1059 (32.48)	11.13	6119.21	6.99	12.48
top	18909.70	15236.80	5.08	3892 (30.95)	39.68	14823.90	6.61	41.96
Ave.	6514.60	5539.71	5.04	1165.83 (33.56)	12.05	5359.52	6.73	13.14
Nor.	1.00	0.85	1.00		1.00	0.82	1.33	1.09

Table 3: Comparison of the detailed routing results for the given and the refined placements.

Benchmark	Given placement				Placement after refinement phase 1				Placement after refinement phases 1 and 2			
	WL	#Vias	#UNs	CPU(s)	WL	#Vias	#UNs	CPU(s)	WL	#Vias	#UNs	CPU(s)
ecc	104016	10710	158	109.67	113015	11041	120	163.09	118717	11305	84	123.99
efc	85030	11446	162	106.26	95256	11719	102	87.09	101314	11852	92	84.17
ctl	111746	12936	139	109.39	121619	13501	104	134.08	127986	13668	89	98.58
alu	92117	12807	177	91.47	96823	12854	141	107.31	103084	13318	150	149.16
div	180061	23865	258	261.12	199196	24573	218	251.41	209648	25056	202	360.94
top	808978	73058	899	1181.68	855856	74521	743	1220.30	892590	76171	588	1314.92
Ave.	230324.67	24137.00	298.83	309.93	246960.83	24701.50	238.00	327.21	25889.83	25228.33	200.83	355.29
Nor.	1.00	1.00	1.00	1.00	1.07	1.02	0.80	1.06	1.12	1.05	0.67	1.15

wirelength, via count, and runtime are increased by 7%, 2%, and 6%, respectively. Note that part of the increase is due to the increase in number of routable nets in DR for placement with refinement phase 1. With our DP refinement phases 1 and 2, the number of unroutable nets can be reduced by 33% on average in DR. The wirelength, via count, and runtime are increased by 12%, 5%, and 15%, respectively. In conclusion, the TCPAP used to evaluate the pin accessibility in DP stage is accurate. Our dynamic programming and linear programming based refinement techniques can effectively improve pin access and reduce the number of unroutable nets in DR.

6. CONCLUSION

In this paper, we propose a detailed placement refinement stage after detailed placement. It directly targets to improve pin accessibility during detailed routing stage. One of the future works is to extend our pin accessibility-driven detailed placement refinement to handle standard cells with different row heights for wider industrial applications.

7. REFERENCES

- [1] Xiang Qiu and Malgorzata Marek-Sadowska. Can pin access limit the footprint scaling. In *Proc. of DAC*, June 2012.
- [2] M. Hsu, N. Katta, H. Lin, K. Lin, K. Tam, and K. Wang. Design and manufacturing process co-optimization in nano-technology. In *Proc. of ICCAD*, November 2014.
- [3] X. Xu, G. Yeric B. Cline, B. Yu, and D. Pan. Self-aligned double patterning aware pin access and standard cell layout co-optimization. In *Proc. of ISPD*, March 2014.
- [4] X. Xu, G. Yeric B. Cline, B. Yu, and D. Pan. Self-aligned double patterning aware pin access and standard cell layout co-optimization. *IEEE Trans. Computer-Aided Design Integrated Circuits Systems*, 34, 2015.
- [5] X. Xu, B. Yu, J. Gao, C. Hsu, and D. Pan. PARR: Pin access planning and regular routing for self-aligned double patterning. In *Proc. of DAC*, June 2015.
- [6] X. Xu, B. Yu, J. Gao, C. Hsu, and D. Pan. PARR: Pin access planning and regular routing for self-aligned double patterning. *ACM Transactions on Design Automation of Electronic Systems*, 21, 2016.
- [7] Tim Nieberg. Gridless pin access in detailed routing. In *Proc. of DAC*, June 2011.
- [8] Muhammet Ozdal. Detailed-routing algorithms for dense pin clusters in integrated circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 28, 2009.
- [9] C.J. Alpert, Z. Li, C.N. Sze, and Y. Wei. Consideration of local routing and pin access during vlsi global routing, April 9 2013. US Patent 8,418,113.
- [10] Z. Qi, Y. Cai, and Q. Zhou. Accurate prediction of detailed routing congestion using supervised data learning. In *Proc. of ICCAD*, March 2014.
- [11] Y. Wei, C. Sze, N.Viswanathan, Z. Li, C. Alpert, L. Reddy, A. Huber, G. Tellez, D. Keller, and S.Sapatnekar. GLARE: Global and local wiring aware routability evaluation. In *Proc. of DAC*, Jun 2012.
- [12] V. Yutsis, I. Bustany, D. Chinnery, J. Shinnerl, and W. Liu. ISPD 2014 benchmarks with sub-45nm technology rules for detailed-routing-driven placement. In *Proc. of ISPD*, March 2014.
- [13] Yanheng Zhang and Chris Chu. CROP: Fast and effective congestion refinement of placement. In *Proc. of ICCAD*, March 2009.
- [14] W.H. Liu, C. Koh, and Y. Li. Optimization of placement solutions for routability. In *Proc. of DAC*, June 2013.
- [15] T. Taghavi, C. Alpert, A.Huber, G. Nam Z. Li, and S. Ramji. New placement prediction and mitigation

- techniques for local routing congestion. In *Proc. of ICCAD*, March 2010.
- [16] Y. Ding, C. Chu, and W.K. Mak. Throughput optimization for SADP and e-beam based manufacturing of 1d layout. In *Proc. of DAC*, June 2014.
- [17] J. Cong, J. Fang, and K.Y. Khoo. Via design rule consideration in multilayer maze routing algorithms. *ACM Transactions on Design Automation of Electronic Systems*, 9, 2000.
- [18] Y. Ding, C. Chu, and W.K. Mak. Self-aligned double patterning-aware detailed routing with double via insertion and via manufacturability consideration. In *Proc. of DAC*, June 2016.
- [19] Y. Ding, C. Chu, and W. K. Mak. Self-aligned double patterning lithography aware detailed routing with color pre-assignment. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2016.