Optimizing Pin Assignment and Escape Routing for Blind-via-Based PCBs

Seong-I Lei and Wai-Kei Mak

***Abstract*—Pin assignment and escape routing are two closely related problems and it is desired to consider routability during pin assignment for package-board co-design. During pin assignment and escape routing, differential pairs and blind-via usage are two major factors to be considered in modern printed circuit board designs. However, most previous works only target on either differential pairs or blind-via usage but cannot handle both of them simultaneously. In this paper, we propose the first package-board co-design method to optimize the pin assignment and escape routing in the presence of differential pairs and blind-via usage for grid pin array. Experimental results show that our package-board co-design flow can achieve wirelength improvement and reduce the number of layers required compared with competitive baseline flows.**

***Index Terms*—Blind vias, differential pairs, escape routing, package-board co-design, pin assignment, single-ended signals.**

I. Introduction

GRID pin array is a very popular pin array structure and has widely been used. In recent years, the pin assignment and escape routing for a grid pin array have become greatly difficult due to the increasing pin count. For a package on a printed circuit board (PCB), the presence of differential pairs and blind-via usage in PCB increase the complexity of pin assignment and escape routing. As the pin assignment problem directly affects the escape routing problem, a good pin assignment can result in higher routability, less number of routing layers and shorter wirelength. In order to achieve better quality in escape routing, it is desired to consider routability during pin assignment for package-board co-design.

Escape routing is to route signals from all terminal pins inside a pin array to the boundaries of the pin array and it is a key problem in PCB routing. Many studies have focused on free escape routing for grid pin array [1]–[7]. For free escape routing, the pins inside an array are to be routed to the array boundaries without any constraint on the pin order. Network flow is commonly used to model this problem [2]–[7]. However, none of these works consider differential pairs in escape routing.

For many designs, it is necessary to consider differential pairs during pin assignment and escape routing. In modern high-speed PCB designs, differential pairs are usually used to transmit critical high-frequency signals. A differential pair consists of two complementary signals for transmitting one signal. The advantages of using differential pairs include better noise immunity, higher reduction of electromagnetic interference, and ground bounce insensitivity. However, the presence of differential pairs and single-ended signals will increase the complexity of pin assignment and escape routing. Fig. 1 shows examples of pin assignment and escape routing considering differential pairs. We assume that the escape boundary is on the bottom and we use solid black circles to denote the non-user I/O pin such as power/ground pins. In Fig. 1(a), two differential pairs (A, A') and (B, B') are considered in the pin assignment and escape routing. As we can determine the pin assignment of a differential pair, we prefer the two pins of a differential pair to be assigned to adjacent pins. In this way, the two routing wires from the differential pair can be merged immediately and the parallel routing can cancel most part of electromagnetic interference. It can also simplify the escape routing of differential pairs. The pin assignment and escape routing will become more difficult when single-ended signals are also present as shown in Fig. 1(b). Four single-ended signals (denoted by solid red circles) are assigned and the escape routing of differential pair (B, B') is forced to detour for a successful escape.

Several previous works considered escape routing of differential pairs [8]–[12]. Yan *et al.* [8] proposed a two-stage routing scheme combining network flow, rip-up and reroute for multiple differential pairs. Li *et al.* [9] considered length matching besides wirelength minimization in escape routing. However, single-ended signals are ignored. In the routing approach for chip-package-board co-design in [10], they assumed each differential pair must be routed within a specified region of the pin array to simplify the problem which may result in sub-optimal solution. Lei and Mak [11] proposed a simultaneous pin assignment and escape routing approach for field-programmable gate array (FPGA)-PCB co-design under FPGA-specific pin assignment constraints. Wang *et al.* [12] proposed an ILP based escape routing for staggered pin array. Single-layer escape routing are assumed in these works except [10] and [11]. Moreover, all these works assume through vias are used in escape routing and neither of them considers the usage of blind vias.

Manuscript received October 30, 2014; revised April 27, 2015; accepted June 24, 2015. Date of publication July 23, 2015; date of current version January 19, 2016. This work was supported by the Ministry of Science and Technology under Grant MOST 103-2220-E-007-002 and Grant MOST 104-2628-E-007-003-MY3. This paper was recommended by Associate Editor Z. Li.

The authors are with the Department of Computer Science, National Tsing Hua University, Hsinchu 300, Taiwan (e-mail: d9762804@oz.nthu.edu.tw; wkmak@cs.nthu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2015.2460458

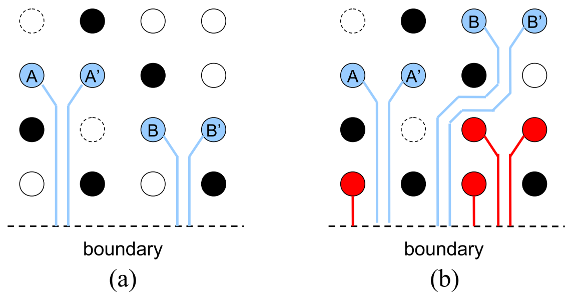


Fig. 1. Examples of pin assignment and escape routing considering differential pairs. (a) Only differential pairs. (b) Single-ended signals and differential pairs. Non-user I/O pin is denoted by a solid black circle.

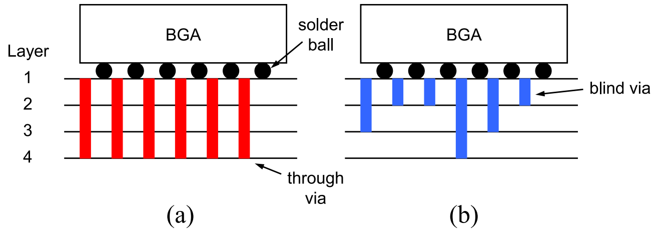


Fig. 2. Cross section views of a BGA on a PCB using two types of via. (a) Through vias. (b) Blind vias.

Through vias and blind vias are commonly used in escape routing. A through via is a via that goes from the top layer to the bottom layer while a blind via only goes from the top layer to an inner layer. Fig. 2 shows the difference between the through-via technology and the blind-via technology. The cross section views of a ball grid array (BGA) on a PCB using though-via and blind-via technology are shown in Fig. 2(a) and (b), respectively. For the through-via technology, the via for each pin connects from the top layer to the bottom layer. In this way, the via will become an obstacle in each layer and the escape routing of the nets needs to avoid these obstacles in each layer. However, for the blind-via technology, the via for each pin only connects from the top layer to the current layer in which the escape routing of this pin is performed. Different from the through-via technology, the area below a blind via in the other layers will become a missing pin and extra routing resources can be obtained.

Using blind vias can benefit the multilayer escape routing in PCB. As shown in Fig. 3, we illustrate the advantage of using the blind-via technology with an example. We assume that the escape boundary is on the left and all the pins need to escape to the escape boundary. In Fig. 3(a), the number of layers used in the escape routing solution using through vias is three. However, Fig. 3(b) shows another escape routing solution for the same number of pins using blind vias and the number of required layers is two. It is because many through vias become routing obstacles in lower layers and the amount of routing resource is the same for each layer. However, when blind vias are used, many blind vias become missing pins and the routing resource is increased for lower layers. So, it is possible to use less number of layers for escape routing using the blind-via technology.

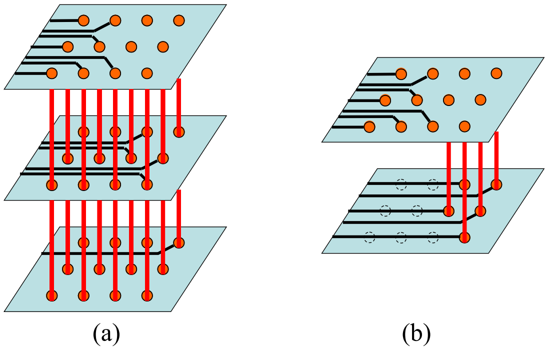


Fig. 3. (a) Escape routing solution using the through-via technology. The number of required layers is three. (b) Escape routing solution using the blind-via technology. The number of required layers is two.

Unfortunately, only a few works consider the escape routing problem using blind vias [2], [13], [14]. Wang *et al.* [2] proposed a network flow based algorithm to solve the escape routing for grid pin array. The pins are selected to escape in each layer by central triangular pattern. Shi and Cheng [13] and Ho *et al.* [14] addressed the escape routing problem in staggered pin array. In [13], a column-by-column horizontal escape routing strategy, a two-sided horizontal/vertical escape routing strategy and a multidirection hybrid channel escape routing strategy are proposed. In [14], a heuristic method is proposed to determine the selection of escaped pin in each layer. In the escaped pin selection, the maximum number of escaped pins for one layer is first determined. Critical pins are first selected and then some evenly distributed pins are selected. Finally, escape routing for the selected escape pins is performed layer by layer until all the pins are escaped. However, differential pairs and power/ground pins are ignored in these works. They assume that all the pins in the array are signal pins that have to be escaped without pin assignment.

In this paper, we address the problem of pin assignment and escape routing under the blind-via technology. Our contributions are summarized as follows.

1) To the best of our knowledge, this is the first work that specifically addresses the pin assignment and escape routing in the presence of differential pairs and blind-via usage for grid pin array.
2) We propose a pin assignment and blind-via usage co-optimization algorithm and a simultaneous multilayer escape routing algorithm considering the blind-via technology. Single-ended signals and differential pairs are handled in both algorithms.
3) Experimental results show that using our proposed co-design flow leads to wirelength and layer reduction compared with competitive baseline flows.

The rest of this paper is organized as follows. In Section II, we discuss the preliminaries and formulate the problem of pin assignment and escape routing under blind-via technology. In Section III, we propose our pin assignment and blind-via usage co-optimization algorithm and simultaneous multilayer escape routing algorithm for single-ended signals and differential pairs. Section IV reports the experimental results. Section V concludes the entire work.

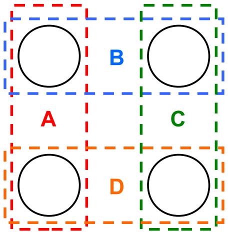


Fig. 4. Four pin-pair candidates A, B, C, and D in a 2×2 pin array.

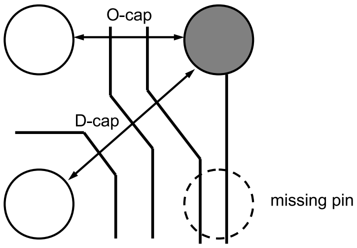


Fig. 5. Legal escape routing solution under O-cap = 2, D-cap = 3, and $\Delta = 2$.

II. Preliminaries and Problem Formulation

A. Pin-Pair Candidates for Differential Signals

In the pin assignment for a differential signal, we prefer the two pins assigned to a differential signal be adjacent. Every two horizontal (or vertical) adjacent pins in the pin array will form a pin-pair candidate for assigning a differential signal. For example, Fig. 4 shows four pin-pair candidates in a 2×2 pin array. There are several advantages of assigning a differential signal to two adjacent pins. First, the two wires from a differential pair can be merged immediately and the rest of the wires can be routed side by side as soon as possible. The parallel routing of a differential pair results in a high tolerance of crosstalk or electromagnetic interference. Second, the two wires from a differential pair will have similar wirelength and therefore the signal skew of a differential pair can be minimized. Moreover, it can simplify the escape routing of a differential pair.

B. Capacity Constraint for Escape Routing

In a pin array, a tile is a square formed by four adjacent pins.[1] The separation between adjacent pins in a tile limits the number of wires that can pass through two orthogonally or diagonally adjacent pins. The capacities of the four sides of a tile are equal to the orthogonal capacity (O-cap), and the capacities of its two diagonals are equal to the diagonal capacity (D-cap). The extra routing resource due to a missing pin is modeled by the extra capacity Δ. For a legal escape routing solution, these capacity constraints cannot be violated. Fig. 5 shows a legal escape routing solution under O-cap = 2, D-cap = 3, and $\Delta = 2$.

C. Problem Formulation

Given a grid pin array, the physical locations of user I/O pins and nonuser I/O pins, a set of single-ended signals and differential signals to be assigned, the capacity constraints from PCB design rules, our objective is to obtain a pin assignment and a legal escape routing solution satisfying the capacity constraints such that the total wirelength and the number of layers required are minimized. We assume the blind-via technology is used in the PCB.

III. Proposed Method

A. Overview of Our Package-Board Co-Design Flow

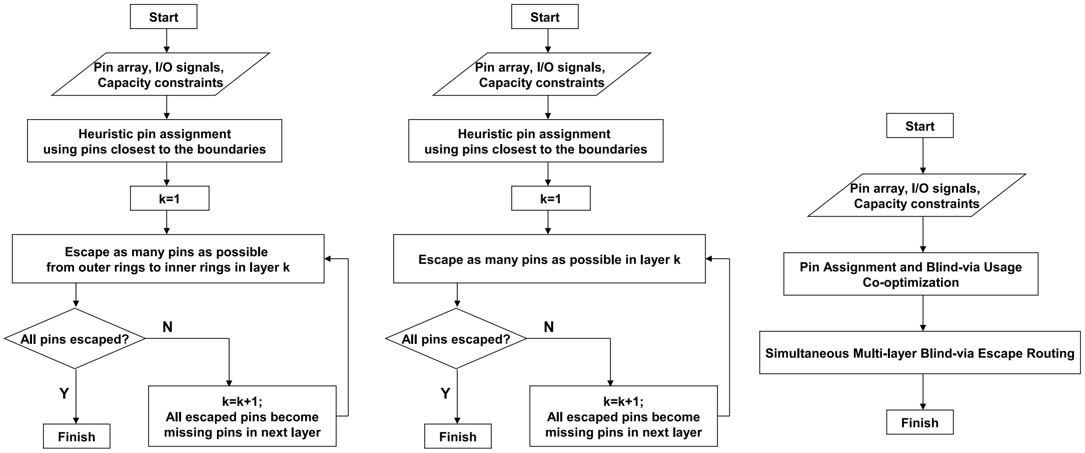
Traditional strategy to the problem of pin assignment and escape routing is to treat the pin assignment and escape routing as two independent stages. Fig. 6 shows two intuitive flows to the problem. A heuristic pin assignment is obtained first by assigning the signals to the pins closest to the boundaries. For the first flow shown in Fig. 6(a), escape routing is performed layer by layer after the pin assignment. Starting from the top layer, pins are escaped row-by-row/column-by-column from outer to inner rings [13]. Each escaped pin will become a missing pin in the subsequent layers. For each layer, we maximize the number of escaped pins as the primary objective and minimize the total wirelength as the secondary objective subject to the constraint that a pin cannot be escaped unless all pins in its outer rings have been escaped. We call the first flow as many as possible using outer rings first (AMAP-OUT). Escape routing is finished when all the assigned pins have successfully escaped to the boundary, otherwise a new layer is added and the escape routing is repeated for the new layer. The second flow shown in Fig. 6(b) is similar to the first one except that it does not require the pins in outer rings to be escaped first. It simply escapes as many pins as possible starting from the top layer till all pins are escaped. We call the second flow as many as possible (AMAP).

Here we introduce a package-board co-design flow as shown in Fig. 6(c). Different from the baseline flows, we propose a pin assignment and blind-via usage co-optimization algorithm to simultaneously map the user signals to appropriate I/O pins of the BGA package and determine the appropriate depth of the corresponding blind-via required on the PCB. After the pin assignment and blind-via usage co-optimization, the routing of each layer can be performed independently and missing pins in each layer are captured. The details of our co-design flow are presented in the following sections.

B. Pin Assignment and Blind-via Usage Co-Optimization

The pin assignment and escape routing problem for single-ended signals using through vias can be solved optimally by network-flow approach [7]. However, the problem becomes difficult when blind vias are used instead and it cannot be solved optimally by network-flow approach anymore as the structure of the routing network for a lower layer will vary according to the routing results of upper layers. In addition, when differential pairs are present, there are two independent signal sources in escape routing, one for single-ended signal and one for differential signal. The routing resources required by these two kinds of signals are different since the resources taken by a differential signal are twice as a single-ended signal. The problem is similar to a two-commodity flow

[1] Note that any one of the four adjacent pins can be replaced by a missing pin.


Start
Pin array, I/O signals, Capacity constraints
Heuristic pin assignment using pins closest to the boundaries
k=1
Escape as many pins as possible from outer rings to inner rings in layer k
All pins escaped?
N
Y
k=k+1; All escaped pins become missing pins in next layer
Finish
Start
Pin array, I/O signals, Capacity constraints
Heuristic pin assignment using pins closest to the boundaries
k=1
Escape as many pins as possible in layer k
All pins escaped?
N
Y
k=k+1; All escaped pins become missing pins in next layer
Finish
Start
Pin array, I/O signals, Capacity constraints
Pin Assignment and Blind-via Usage Co-optimization
Simultaneous Multi-layer Blind-via Escape Routing
Finish

(a) (b) (c)

Fig. 6. (a) Baseline flow AMAP-OUT that escapes as many pins as possible from outer to inner rings. (b) Baseline flow AMAP that escapes as many pins as possible. (c) Our package-board co-design flow.

TABLE I
PARAMETERS AND VARIABLES USED IN THE ILP FORMULATION OF PIN ASSIGNMENT AND BLIND-VIA USAGE CO-OPTIMIZATION

Parameters	
D_i	distance from pin i to the nearest boundary
D'_{ij}	distance from the middle of pins i and j to the nearest boundary
N_S	total number of single-ended signals
N_D	total number of differential signals
U_r^f	number of user I/O pins in row (column) r in region f
E_r^f	number of tile intervals in row (column) r in region f
R	total number of rings
L	total number of layers
A	set of horizontal or vertical adjacent pins that form a pin-pair candidate
P_r^f	set of pins in row (column) r in region f
$P_{1,r-1}^f$	set of pins from row (column) 1 to row (column) $r-1$ in region f
Δ	extra capacity of a missing pin
Variables	
s_i^t	0-1 integer variable that $s_i^t = 1$ if pin i is assigned to a single-ended signal and it is escaped on layer t, and $s_i^t = 0$ otherwise
d_{ij}^t	0-1 integer variable that $d_{ij}^t = 1$ if pins i and j $(i, j \in A$ and $i < j)$ are assigned to a differential signal and it is escaped on layer t, and $d_{ij}^t = 0$ otherwise

problem which has been proven to be NP-complete [15]. In order to maintain optimality, one of the approaches is to use integer linear programming (ILP) to simultaneously solve the pin assignment and escape routing on PCBs with blind vias. However, the complexity of the ILP is too high and it cannot be solved within a few days. But it is affordable to solve the problem if we first determine the pin assignment and the appropriate depths of the corresponding blind vias required on the PCB.

We propose an iterative method to perform pin assignment and blind-via usage co-optimization considering both single-ended signals and differential pairs. We want to optimize the pin assignment and determine the appropriate depths of the corresponding blind vias required on the PCB simultaneously to facilitate PCB escape routing that minimizes the expected total wirelength. The co-optimization is guided by estimating the routing detours under different configurations of pin assignment and blind-via usage. The notations used are defined in Table I. In the first iteration, we formulate an ILP as follows:

$$\min \quad \sum_{k=1}^{L}\sum_{i} D_i s_i^k + \sum_{k=1}^{L}\sum_{(i,j)\in A} 2D'_{ij} d_{ij}^k$$

$$\text{s.t.} \quad \sum_{k=1}^{L} s_i^k \le 1 \quad \forall i \tag{1}$$

$$\sum_{k=1}^{L}\sum_{i} s_i^k = N_S \tag{2}$$

$$\sum_{k=1}^{L} d_{ij}^k \le 1 \quad \forall (i,j) \in A \tag{3}$$

$$\sum_{k=1}^{L}\sum_{(i,j)\in A} d_{ij}^k = N_D \tag{4}$$

$$\sum_{k=1}^{L} s_i^k + \sum_{k=1}^{L}\sum_{(i,j)\in A} d_{ij}^k + \sum_{k=1}^{L}\sum_{(g,i)\in A} d_{gi}^k \le 1 \quad \forall i \tag{5}$$

$$\begin{aligned} &\Bigg(U_r^f - \sum_{k=t}^{L}\sum_{i\in P_r^f} s_i^k - \sum_{k=t}^{L}\sum_{i\in P_r^f, j\notin P_r^f; (i,j)\in A} d_{ij}^k \\ &\quad - 2\sum_{k=t}^{L}\sum_{i,j\in P_r^f; (i,j)\in A} d_{ij}^k \Bigg)\Delta + \text{O-cap} \times E_r^f \ge \sum_{i\in P_{1,r-1}^f} s_i^t \\ &+ \sum_{i\in P_{1,r-1}^f, j\notin P_{1,r-1}^f; (i,j)\in A} d_{ij}^t + 2\sum_{i,j\in P_{1,r-1}^f; (i,j)\in A} d_{ij}^t \end{aligned} \tag{6}$$

$$\text{for} \quad r = 2, \ldots, R, \quad t = 1, \ldots, L, \quad f = 1, \ldots, 4$$

$$s_i^t, d_{ij}^t \in \{0, 1\}. \tag{7}$$

Constraint (1) [(3)] guarantees that only one layer can be chosen for routing a single-ended signal (a differential signal).

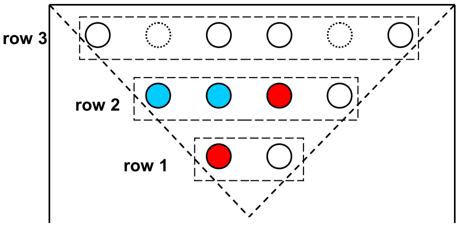


Fig. 7. Necessary condition holds for row 3. A solid red circle denotes a pin assigned to a single-ended signal and a solid blue circle denotes a pin assigned to a differential signal.

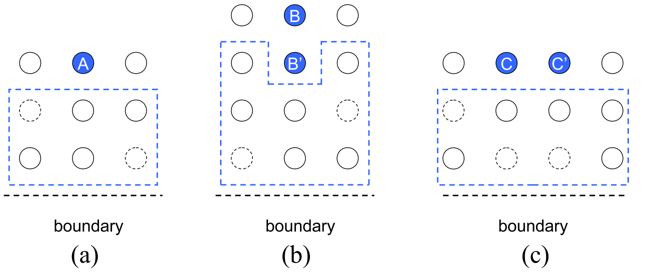


Fig. 8. Critical region (denoted by the dotted line bounding box) of (a) a pin A, (b) two vertical adjacent pins B and B', and (c) two horizontal adjacent pins C and C'.

Constraint (2) ensures that the number of single-ended signals assigned must be equal to the total number of single-ended signals and constraint (4) is for differential signals. We use constraint (5) to prevent assigning more than one signal to a pin. If two adjacent pins i and j are assigned to a differential signal, neither pin i nor j can be assigned to a single-ended signal again. On the other hand, if a pin i is assigned to a single-ended signal, any pin-pair candidate formed by pin i cannot be assigned to a differential signal. Constraint (6) is the routability constraint. For each ring, if its capacity is less than the number of signals assigned to its inner rings in the same PCB layer, it is impossible for all these signals to escape successfully. Hence, we impose constraint (6) to provide the necessary condition for routability. We partition the package into four regions and constraint (6) guarantees that for each region the capacity of a row (column) must be greater than or equal to the number of pins assigned to signals in all its inner rows (columns). For example, Fig. 7 shows a region with three rows. Assume that O-cap $= 2$ and $\Delta = 2$, the necessary condition holds for row 3 as the capacity of row 3 is $5 \times 2 + 2 \times 2 = 14$ and the total number of pins assigned to single-ended and differential signals in rows 2 and 1 is 4.

In the first iteration, we estimate the wirelength of a signal by the distance from its assigned pin to the nearest boundary. But usually it would be too optimistic since there are routing detours in practice. In subsequent iterations, we incorporate a penalty cost to model the estimated routing detour for a pin assignment.

For estimating the routing detour of a signal, we propose to examine the critical region of a pin (two adjacent pins) assigned to a single-ended signal (differential signal). The critical region of a pin assigned to a signal is the region which is most likely to cause the signal to detour if the region is congested. We define the critical region as follows.

1) Critical Region of Pin: The critical region of pin A includes the pins within the column (row) from pin A to the nearest boundary and the adjacent two columns (rows) with the same length if the nearest boundary is the top or bottom (left or right) boundary. For example, Fig. 8(a) shows the critical region of pin A. We denote the critical region of a pin i in layer k by CR_i^k.

2) Critical Region of Two Adjacent Pins: If two adjacent pins B and B' are vertical (horizontal) adjacent and the nearest boundary is the top or bottom (left or right) boundary, the critical region of pins B and B' is the critical region of pin B except for pin B'. For example, Fig. 8(b) shows the critical region of pins B and B'. On the other hand, if two adjacent pins C and C' are horizontal (vertical) adjacent and the nearest boundary is the top or bottom (left or right) boundary, the critical region of pins C and C' includes the pins within the columns (rows) from the pins C and C' to the nearest boundary and the adjacent two columns (rows) with the same length. For example, Fig. 8(c) shows the critical region of pins C and C'. We denote the critical region of pins i and j in layer k by CR_{ij}^k.

If a single-ended signal is assigned to a pin i in layer k, the penalty cost in iteration g ($\mathrm{penalty}_{i,k}^g$) which takes into account the utilization of the critical region of pin i is calculated

$$\mathrm{penalty}_{i,k}^g = \Big[\alpha\ P\_\mathrm{ratio}\Big(\mathrm{CR}_i^k\Big) + \beta\ S\_\mathrm{ratio}\Big(\mathrm{CR}_i^k\Big) + \gamma\ \mathrm{DP\_ratio}\Big(\mathrm{CR}_i^k\Big)\Big]D_i \quad (8)$$

where $P\_\mathrm{ratio}(\mathrm{CR}_i^k)$ is the ratio of the number of nonmissing pins to the total number of pins in CR_i^k, $S\_\mathrm{ratio}(\mathrm{CR}_i^k)$ is the ratio of the number of pins assigned to single-ended signals to the total number of pins in CR_i^k, and $\mathrm{DP\_ratio}(\mathrm{CR}_i^k)$ is the ratio of the number of pins assigned to differential signals to the total number of pins in CR_i^k. α, β, and γ are empirical parameters. The motivation of the penalty cost is to estimate the routing congestion of a critical region using the terms in (8). Note that the terms calculated in (8) are based on the information in the previous iteration. If the routing resource of a critical region is enough, the escape routing of this pin will always lie within the critical region and no long routing detour is produced. However, if the routing resource of a critical region is highly utilized, the pin needs to route beyond the critical region and long routing detour may happen. We estimate the routing congestion of a critical region based on the ratio of nonmissing pins and the ratio of pins assigned to other signals in the critical region. If the ratio of nonmissing pins is large for a critical region, the critical region is crowded with pins which are routing blockages. Besides, if the ratio of pins assigned to other signals is large for a critical region, it means the demand of the critical region is high and many other signals will compete for the routing resource of the critical region. Therefore, a larger penalty cost should be set for this critical region.

For pins i and j assigned to a differential signal in layer k, the $\mathrm{penalty}_{ij,k}^g$ can be calculated similarly by (9) except CR_i^k and D_i are replaced by CR_{ij}^k and D'_{ij}, respectively. The escape routing of a differential signal is more difficult than a single-ended signal as it requires sufficient space for two wires

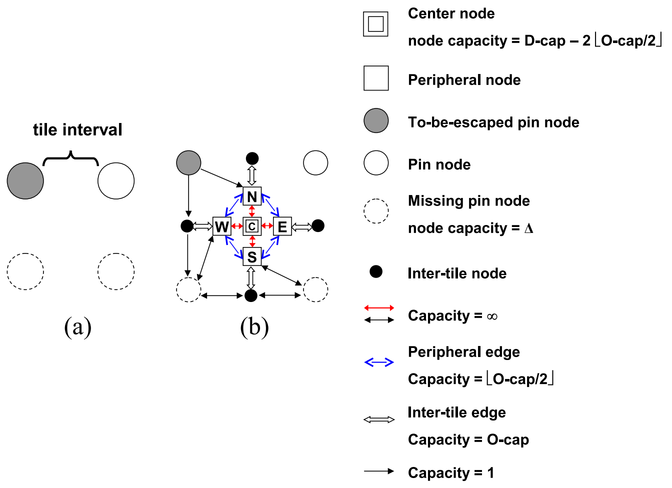


Fig. 9. (a) Tile. (b) Single-ended signal routing network for this tile.

to be routed side-by-side. Hence, we adjust the penalty cost for a differential signal by multiplying a constant $c > 1$

$$\text{penalty}^{g}_{ij,k} = c\Big[\alpha\ P\_\text{ratio}\Big(\text{CR}^{k}_{ij}\Big) + \beta\ S\_\text{ratio}\Big(\text{CR}^{k}_{ij}\Big) + \gamma\ \text{DP\_ratio}\Big(\text{CR}^{k}_{ij}\Big)\Big]D'_{ij}. \quad (9)$$

The objective function of the ILP at iteration n is defined as

$$\sum_{k=1}^{L}\sum_{i}\left(D_i + \frac{1}{n-1}\sum_{g=1}^{n-1}\text{penalty}^{g}_{i,k}\right)s^{k}_{i} + \sum_{k=1}^{L}\sum_{i,j}2\left(D'_{ij} + \frac{1}{n-1}\sum_{g=1}^{n-1}\text{penalty}^{g}_{ij,k}\right)d^{k}_{ij} \quad \text{for} \quad n > 1. \quad (10)$$

The stopping criterion of our pin assignment and blind-via usage co-optimization is met when the objective values converge.

C. Simultaneous Multilayer Escape Routing for Single-Ended Signals and Differential Pairs

For escape routing considering blind vias in a routing layer, we need to consider missing pins in our routing networks. We first introduce two routing network models considering missing pins for single-ended signals and differential pairs, respectively.

1) Network for Single-Ended Signal Escape Routing: Even though the network model in [5] can handle single-ended signal escape routing with missing pins, it cannot accurately capture the wirelength difference between an orthogonal wire and a diagonal wire in the area of missing pins. Hence, we propose a routing network model which is slightly different from the one in [5]. For each pin (missing pin) in the pin array, there is a corresponding pin node (missing pin node) in our routing network. The node capacity of a missing pin node is Δ. If a pin is assigned to a single-ended signal in the current layer, we call the corresponding pin node a to-be-escaped pin node. The area surrounded by four neighboring pin (missing pin) nodes is a tile. There are four peripheral nodes E, S, W, and N, as well as a center node C in each tile. The center node C has capacity D-cap−2⌊O-cap/2⌋. There is an intertile node in the middle of each tile interval [see Fig. 9(b)].

There is a bidirectional edge between every peripheral node and the center node with infinite capacity. Furthermore, there is a bidirectional edge called peripheral edge between peripheral nodes N and E, E and S, S and W, and W and N, respectively. Each peripheral edge has capacity ⌊O-cap/2⌋. Intertile edges are bidirectional edges that connect the intertile node to the nearest peripheral node. The capacity of an intertile edge is O-cap. For a to-be-escaped pin node, there is a directed edge from the pin node at the NW/NE/SE/SW corner to the peripheral node N/E/S/W. For an intertile node in a tile interval, if the tile interval is formed by one to-be-escaped pin node and one missing pin node, there are directed edges from the pin node to the intertile node and from the intertile node to the missing pin node. These directed edges have capacity 1. If the tile interval is formed by two missing pin nodes, there are bidirectional edges between the intertile node and the missing pin nodes. For a missing pin node, there is one bidirectional edge between the missing pin node and a peripheral node.[2] These bidirectional edges have infinite capacity. If a pin is not assigned to any signal in the current layer, there will not exist any incoming edge and outgoing edge for the corresponding pin node. For example, Fig. 9(a) shows a tile where the top-left pin is a to-be-escaped pin, the top-right pin is a pin and the bottom two pins are missing pins. Fig. 9(b) shows the corresponding routing network for this tile.

We illustrate the difference between the network model in [5] and our network model with an example shown in Fig. 10. For the escape routing instance in Fig. 10(a), Fig. 10(b) shows the corresponding flow solution in the network model in [5] while Fig. 10(c) shows the corresponding flow solution in our network model. In order to simplify the discussion, we refer to the edge between a missing pin node and a peripheral node as a diagonal edge and the edge between a missing pin node and an intertile node as an orthogonal edge. Note that there is no orthogonal edge in the network model in [5]. In the network model in [5], the cost of a missing pin node is 1.4. In our network model, the cost of an intertile edge is 0.5. We set the cost of an orthogonal edge and a diagonal edge to 0.5 and 0.7, respectively. Note that the cost of a missing pin node and the cost of an intertile node in our network model are zero. The total wirelength calculated by the network model in [5] for the instance in Fig. 10(a) is 5.2 while the total wirelength calculated by our network model is 4.4. It is because the wirelength of the left vertical wire passing through three consecutive missing pins calculated by the network model in [5] is 2.8, but it should be 2 as it is calculated in our network model. This shows that the network model in [5] overestimates the wirelength of orthogonal wire in the area of missing pins.

[2] The rule of connection between a missing pin node and a peripheral node is the same as a to-be-escaped pin node to a peripheral node but a directed edge is replaced by a bidirectional edge.

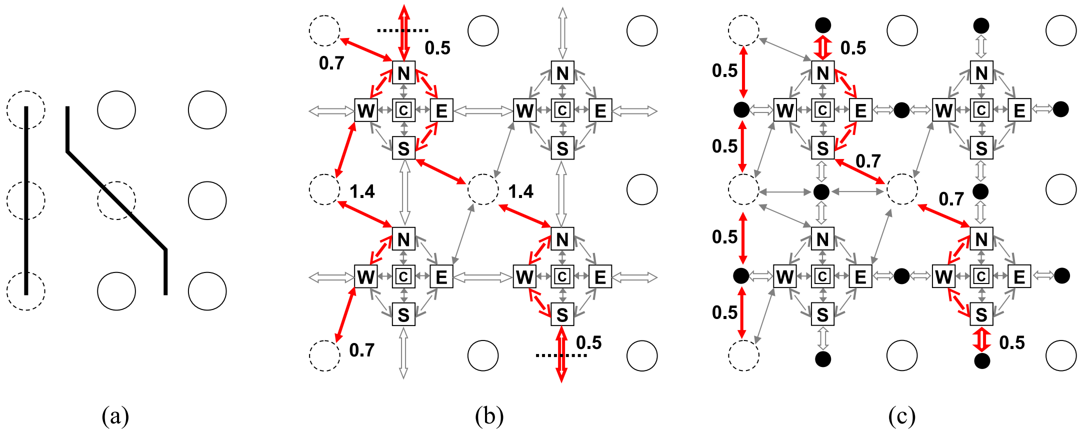


Fig. 10. (a) Escape routing solution in the presence of some missing pins. (b) Corresponding flow solution in the network model in [5]. (c) Corresponding flow solution in our network model. The edge with one unit flow is in red.

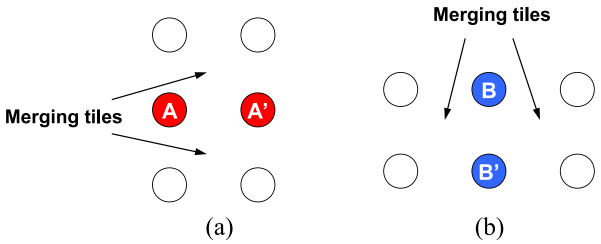


Fig. 11. Merging tiles of a differential pair (a) (A, A') with horizontal pin-pair arrangement and (b) (B, B') with vertical pin-pair arrangement.

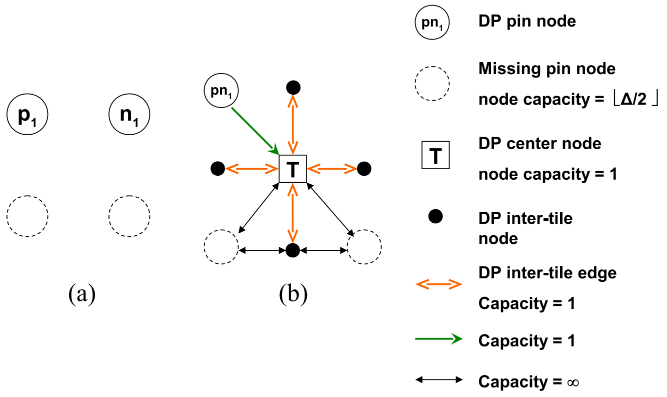


Fig. 12. (a) Tile. (b) Differential-pair routing network for this tile.

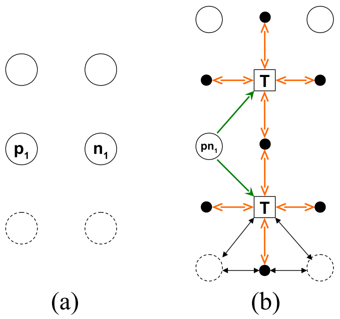


Fig. 13. (a) Merging tiles of a differential pair with horizontal pin-pair arrangement. (b) Corresponding routing network in the merging tiles of the differential pair.

2) Network for Differential-Pair Escape Routing: For the differential-pair routing network, we propose another network model and only use one DP pin node to represent two pins assigned to a differential pair. There is a DP center node in each tile. The node capacity of the DP center node is 1. Similar to the single-ended signal escape routing network, there is a missing pin node for each missing pin in the pin array. The node capacity of a missing pin node is $\lfloor \Delta/2 \rfloor$. There is a DP intertile node in the middle of each tile interval.

For a DP pin node, there are directed edges from a DP pin node of a differential pair to the DP center nodes in the merging tiles of this differential pair, these directed edges have capacity 1. For example, Fig. 11 shows the merging tiles of a differential pair. DP intertile edges are bidirectional edges that connect the DP center node to the DP intertile node. The capacity of a DP intertile edge is 1. For a DP intertile node in a tile interval, if the tile interval is formed by two missing pin nodes, there are bidirectional edges between the DP intertile node and the missing pin nodes. For a missing pin node, there is one bidirectional edge between a missing pin node and the DP center node. These bidirectional edges have infinite capacity. Recall that if a pin is not assigned to any signal in the current layer, there will not exist any incoming edge and outgoing edge for the corresponding pin node. For example, Fig. 12(a) shows a tile which has the top two pins assigned to a differential pairs and the bottom two pins are missing pins. Fig. 12(b) shows the corresponding routing network for this tile. Fig. 13(a) shows the merging tiles of a differential pair with horizontal pin-pair arrangement and Fig. 13(b) shows the corresponding routing network in the merging tiles of the differential pair. Note that one unit of flow represents two routing wires in our differential-pair routing network.

3) ILP Formulation for Simultaneous Multilayer Escape Routing: We propose an ILP formulation to solve the multilayer escape routing for single-ended signals and differential pairs simultaneously. We construct both single-ended signal routing network and differential-pair routing network for each layer and determine the flow of the two routing

TABLE II
SETS AND VARIABLES USED IN THE ILP FORMULATION OF ESCAPE ROUTING

Sets	
$\dot{E}_O$	set of inter-tile edges, edges between a to-be-escaped pin node and an inter-tile node and edges between a missing pin node and an inter-tile node
$\dot{E}_D$	set of edges between a to-be-escaped pin node and a peripheral node and edges between a missing pin node and a peripheral node
$\ddot{E}_O$	set of DP inter-tile edges and edges between a missing pin node and a DP inter-tile node
$\ddot{E}_D$	set of edges between a DP pin node and a DP center node and edges between a missing pin node and a DP center node
E_v^{in}	set of incoming edges of node v
E_v^{out}	set of outgoing edges of node v
$\dot{E}_b$	set of edges that escape to the boundary in single-ended signal routing network
$\ddot{E}_b$	set of edges that escape to the boundary in differential-pair routing network
$\dot{E}_{inter}$	set of inter-tile edges
$\dot{E}_{peri}$	set of peripheral edges
$\dot{E}_{C'C''}$	set of edges that connect nodes C' and C'' for any center node C
$\ddot{E}_{T'T''}$	set of edges that connect nodes T' and T'' for any DP center node T
$\dot{E}_{P'P''}$	set of edges that connect nodes P' and P'' for any missing pin node P in single-ended signal routing network
$\ddot{E}_{P'P''}$	set of edges that connect nodes P' and P'' for any missing pin node P in differential-pair routing network
Z	set of integers
Variables	
$f(e^k)$	number of signals passing through edge e in layer k
h_t^k	0-1 integer variable that $h_t^k = 1$ if tile t in layer k is reserved for a differential pair and $h_t^k = 0$ otherwise
m_p^k	0-1 integer variable that $m_p^k = 1$ if the area of missing pin p in layer k is reserved for a differential pair and $m_p^k = 0$ otherwise

networks simultaneously. In Table II, we define the notations used in the ILP formulation of escape routing. $\dot{E}$ denotes the set of edges in the single-ended signal routing network while $\ddot{E}$ denotes the set of edges in the differential-pair routing network. Our ILP formulation is as follows:

$$\min \quad 0.5 \sum_{e^k \in \dot{E}_O} f\left(e^k\right) + 0.7 \sum_{e^k \in \dot{E}_D} f\left(e^k\right) + \sum_{e^k \in \ddot{E}_O} f\left(e^k\right) + 1.4 \sum_{e^k \in \ddot{E}_D} f\left(e^k\right)$$

$$\text{s.t.} \quad \sum_{e^k \in E_v^{\text{out}}} f\left(e^k\right) = 1 \quad \forall \text{ to-be-escaped pin node } v \text{ in layer } k \tag{11}$$

$$\sum_{e^k \in E_v^{\text{out}}} f\left(e^k\right) = 1 \; \forall \text{ DP pin node } v \text{ in layer } k \tag{12}$$

$$\sum_{e^k \in E_v^{\text{in}}} f\left(e^k\right) = \sum_{e^k \in E_v^{\text{out}}} f\left(e^k\right) \quad \forall \text{ node } v \text{ except pin node and DP pin node} \tag{13}$$

$$\sum_{e^k \in \dot{E}_b} f\left(e^k\right) = N_S \tag{14}$$

$$\sum_{e^k \in \ddot{E}_b} f\left(e^k\right) = N_D \tag{15}$$

$$f\left(e^k\right) \le \text{D-cap} - 2\left\lfloor \frac{\text{O-cap}}{2} \right\rfloor \quad \forall e^k \in \dot{E}_{C'C''} \tag{16}$$

$$f\left(e^k\right) \le \text{O-cap} \quad \forall e^k \in \dot{E}_{\text{inter}} \tag{17}$$

$$f\left(e^k\right) \le \left\lfloor \frac{\text{O-cap}}{2} \right\rfloor \quad \forall e^k \in \dot{E}_{\text{peri}} \tag{18}$$

$$f\left(e^k\right) \le \left(1 - h_t^k\right)\text{O-cap} \quad \forall \text{ tile } t, \; \forall e^k \in \dot{E}_{\text{inter}} \text{ s.t. } e^k \text{ is in } t \tag{19}$$

$$f\left(e^k\right) \le h_t^k \quad \forall \text{ tile } t, \forall e^k \in \ddot{E}_{T'T''} \text{ s.t. } e^k \text{ is in } t \tag{20}$$

$$f\left(e^k\right) \le \left(1 - h_t^k\right)\Delta \quad \forall \text{ tile } t, \forall e^k \in E_v^{\text{out}} \text{ s.t. } v \text{ is a to-be-escaped pin node or missing pin node and } e^k \text{ is in } t \tag{21}$$

$$f\left(e^k\right) \le \left(1 - m_p^k\right)\Delta \quad \forall e^k \in \dot{E}_{P'P''} \tag{22}$$

$$f\left(e^k\right) \le \left\lfloor \frac{\Delta}{2} \right\rfloor m_p^k \quad \forall e^k \in \ddot{E}_{P'P''} \tag{23}$$

$$f\left(e^k\right) \ge 0 \quad \forall e^k \tag{24}$$

$$h_t^k, m_p^k \in \{0, 1\}. \tag{25}$$

Constraint (11) is the flow supply constraint for a single-ended signal. If a pin is assigned to a single-ended signal in layer k, one of the outgoing edges in layer k of the corresponding pin node will have one unit flow. Constraint (12) is the flow supply constraint for a differential signal. If two adjacent pins are assigned to a differential signal in layer k, one of the outgoing edges in layer k of the corresponding DP pin node will have one unit flow. Constraint (13) is the flow conversation constraint for all the nodes except pin node and DP pin node. The amount of incoming flow must equal to the amount of outgoing flow for each node. Constraint (14) [(15)] guarantees that the number of nets that escape to the boundaries in single-ended signal (differential-pair) routing network must be equal to the total number of single-ended signals (differential signals). Constraint (16) is the node capacity constraint for the center node C. We split the center node into nodes C' and C'' and the node capacity of C is equal to the capacity of edge $C'C''$. Constraints (17) and (18) are the capacity constraints for intertile edge and peripheral edge, respectively.

In order to prevent routing conflict between single-ended signals and differential pairs within a tile, we assume that a tile can be used by either a differential pair or single-ended signals.

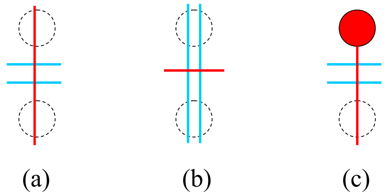


Fig. 14. Illegal escape routing cases. (a) Routing of a single-ended signal (denoted by a red line) passes through the area of two consecutive missing pins and crosses with the routing of a differential pair (denoted by two light blue lines). (b) Routing of a differential pair passes through the area of two consecutive missing pins and crosses with the routing of a single-ended signal. (c) Routing of a single-ended signal escapes from a pin and crosses with the routing of a differential pair.

We introduce an 0-1 integer variable h_t^k for each tile t in layer k. If tile t in layer k is reserved for a differential pair ($h_t^k = 1$), the capacity of the intertile edges in tile t in layer k of the single-ended signal routing network will be forced to be zero by constraint (19). Similarly, we can split the DP center node T into nodes T' and T''. The capacity of the edge $T'T''$ in tile t in layer k of the differential-pair routing network becomes one by constraint (20). Besides, if tile t in layer k is reserved for a single-ended signal ($h_t^k = 0$), the capacity of the corresponding intertile edges of the single-ended signal routing network will become O-cap and the capacity of the corresponding edge $T'T''$ in the differential-pair routing network will be forced to be zero. If a tile is reserved for a differential pair, constraint (21) will immediately forbid any single-ended signal to escape in this tile from any to-be-escaped pin or missing pin that forms the tile.

Although we assume that a tile can only be passed through by either a differential pair or single-ended signals in escape routing, routing conflict may still appear in the area of a missing pin. Similar to a tile, we assume that the area of a missing pin can be used by either a differential pair or single-ended signals. We introduce another 0-1 integer variable m_p^k for each missing pin p in layer k. Similar to the center node and the DP center node, we split the missing pin node P into nodes P' and P'' for missing pin p in both single-ended signal routing network and differential-pair routing network. If the area of a missing pin p is reserved for a differential pair ($m_p^k = 1$), constraint (22) will force the capacity of edge $P'P''$ in single-ended signal routing network to be zero but constraint (23) will provide an extra capacity $\lfloor(\Delta/2)\rfloor$ for edge $P'P''$ in differential-pair routing network. Moreover, if the area of a missing pin p is reserved for a single-ended signal ($m_p^k = 0$), constraint (22) will provide an extra capacity Δ for edge $P'P''$ in single-ended signal routing network while constraint (23) will force the capacity of this edge in the differential-pair routing network to be zero. Finally, constraint (24) ensures the number of signals passing through each edge in routing networks is non-negative. In the objective function, we minimize the total wirelength of single-ended signals and differential pairs.

It is not enough to eliminate all the illegal escape routing cases by constraints (19)–(23) since single-ended signals travel in one network and differential pairs travel in another.

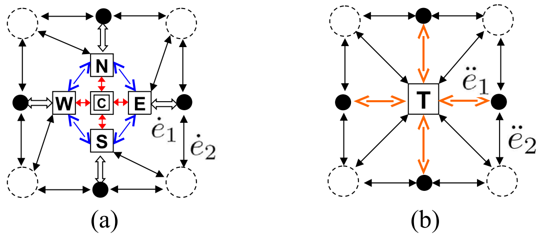


Fig. 15. (a) Single-ended signal routing network with labeled edges. (b) Differential-pair routing network with labeled edges.

Fig. 14 shows the illegal escape routing cases caused by a single-ended signal and a differential pair in a tile interval. Without loss of generality, we only show one orientation for each illegal case with one single-ended signal and one differential pair. It is not hard to see that other illegal cases with different orientations and different number of signals can be handled in the same way. We can add the following constraints to eliminate the illegal escape routing cases shown in Fig. 14:

$$f(\dot{e}_2) \leq (1 - f(\ddot{e}_1))\Delta \tag{26}$$

$$f(\dot{e}_1) \leq (1 - f(\ddot{e}_2))\text{O-cap} \tag{27}$$

$$f(\dot{e}_1), f(\dot{e}_2), f(\ddot{e}_1), f(\ddot{e}_2) \in Z. \tag{28}$$

We only construct the single-ended signal routing network shown in Fig. 15(a) and the differential-pair routing network shown in Fig. 15(b) according to the illegal cases shown in Fig. 14(a) and (b). Constraint (26) eliminates the illegal cases shown in Fig. 14(a) and (c) while constraint (27) eliminates the illegal case shown in Fig. 14(b). For example, suppose we want to eliminate the illegal cases shown in Fig. 14(a) and (b). If $f(\ddot{e}_1) = 1$, constraint (26) will force $f(\dot{e}_2)$ to be zero. Any single-ended signal cannot go through the area of two consecutive missing pins vertically if a differential pair goes through the interval of these two missing pins horizontally. Similarly, if $f(\ddot{e}_2) = 1$, constraint (27) will force $f(\dot{e}_1)$ to be zero. If a differential pair goes through the area of two consecutive missing pins vertically, any single-ended signal cannot go through the interval of these two missing pins horizontally.

For multilayer escape routing, the lower bound of the number of layers required can be calculated as follows:

$$L = \begin{cases} 1 & \text{if } n \leq p + t \times \text{O-cap} \\ \left\lceil \dfrac{n - p - t \times \text{O-cap}}{t \times \text{O-cap} + p \times \Delta} \right\rceil + 1 & \text{otherwise} \end{cases}$$

where L is the lower bound of the number of layers required, n is the number of pins used by the I/O signals, p is the number of user I/O pins on the boundaries of the package, t is the number of tile intervals on the boundaries of the package and Δ is the extra capacity of a missing pin.

The complexity of the proposed ILP formulation depends on the number of variables and the number of constraints. For the ILP formulation of the pin assignment and blind-via usage co-optimization, we assume that U is the number of user I/O pins, $|A|$ is the number of pin-pair candidates, R is

TABLE III
INFORMATION OF TESTCASES WITH ALL USER I/O PINS UTILIZED

testcase	#single-ended signals	#DPs	#pins used
T1	601	45	691
T2	583	54	691
T3	527	82	691
T4	491	100	691
V1	438	0	438
V2	418	10	438
V3	394	22	438
V4	382	28	438
X1	427	8	443
X2	417	13	443
X3	403	20	443
X4	383	30	443

the number of rings in the package, and L is the number of layers used. The number of variables is $(U + |A|)L$ and the number of constraints is $2U + |A| + 4(R - 1)L + 2$. For the ILP formulation of the multilayer escape routing, we assume that N_S is the number of single-ended signals, N_D is the number of differential signals, and P is the number of missing pins in all routing layers and the package has m rows and n columns. The number of edges for each tile in the single-ended signal routing network and the differential-pair routing network are no more than 49 and 33, respectively. The number of variables is no more than $83(m-1)(n-1)L + 3P$. The number of constraints is no more than $2N_S + N_D + 7P + 116$ $(m-1)(n-1)L + 2$.

IV. Experimental Results

We implemented our package-board co-design flow in the C++ programming language on a 3 GHz Linux machine with 64 GB memory. The Gurobi optimizer [16] is employed as our ILP solver. Three packages are used in our experiments. One is a package used by a commercial FPGA with 956 pins in total (691 user I/O pins). Another package is a flip chip land grid array (FCLGA) 1155 with 1155 pins in total (438 user I/O pins). It is used by Intel third generation Core processors with socket type LGA 1155 (e.g., Intel Core i7-3770). The last package is a FCLGA 1150 with 1150 pins in total (443 user I/O pins). It is used by Intel fourth generation Core processors with socket type LGA 1150 (e.g., Intel Core i7-4790). There are two sets of testcases. The first set of testcases assume that all user I/O pins are utilized which is common for microprocessor and application-specified integrated circuit designs. The number of single-ended signals, the number of differential pairs and the number of pins used for these testcases are listed in Table III. The second set of testcases assume that not all user I/O pins are utilized which is common for FPGA-based designs. Table IV shows the information of these testcases. We run testcases T1–T7 on the FPGA package, testcases V1–V7 on the package FCLGA 1155 and testcases X1–X7 on the package FCLGA 1150. We use O-cap = 2, D-cap = 3, and $\Delta = 2$ as our PCB capacity constraints. We set $\alpha = 0.1$, $\beta = 0.32$, $\gamma = 0.76$, and $c = 1.1$ in our experiments.

TABLE IV
INFORMATION OF TESTCASES WHERE NOT ALL USER I/O PINS ARE UTILIZED

testcase	#single-ended signals	#DPs	#pins used
T5	564	28	620
T6	571	34	639
T7	560	52	664
V5	375	10	395
V6	376	18	412
V7	387	20	427
X5	388	15	418
X6	379	25	429
X7	343	45	433

We implemented the baseline flows shown in Fig. 6(a) and (b) and compare them with our co-design flow. For these two baseline flows, the pin assignment is obtained by ILP to minimize the total distance from each assigned pin to its nearest boundary. The proposed routing networks are used in all baseline flows as well as our co-design flow. In baseline flow AMAP-OUT, each pin is escaped from outer to inner rings as many as possible in upper layers while the wirelength is minimized. Baseline flow AMAP is similar to AMAP-OUT except that each pin is escaped as many as possible in upper layers. For all baseline flows, we partition the package into four sectors and forbid signals in one sector to escape to the boundary in other sector in order to prevent long routing detour [1].

Table V shows the experimental results of the two baseline flows and our co-design flow using the testcases with all user I/O pins utilized. The number of layers required, wirelength, objective value and the number of iterations for our pin assignment and blind-via usage co-optimization (Obj. and #iter.), runtime for our pin assignment and blind-via usage co-optimization (PBC runtime), runtime for the escape routing in our co-design flow, and total runtime are reported. Comparing with baseline flows AMAP-OUT and AMAP, our co-design flow can achieve 11.5% and 9.7% wirelength improvement on average, respectively. Among the 12 testcases, eight testcases in baseline flow AMAP-OUT and five testcases in baseline flow AMAP require one more routing layer than our co-design flow.

We further compare our co-design flow with baseline flows AMAP-OUT and AMAP using testcases where not all user I/O pins are utilized in Table VI. Our co-design flow can reduce 13.6% and 10.4% wirelength on average compared with baseline flows AMAP-OUT and AMAP, respectively. Baseline flows AMAP-OUT and AMAP both require more routing layers on average than our co-design flow. Besides, we modify baseline flow AMAP as baseline flow 3 by adopting the pin positions chosen by our co-design flow. As baseline flow 3 escapes pins as many as possible in upper layers, the blind-via depth of each signal is determined in the escape routing stage. By adopting the pin positions chosen by our co-design flow, the number of routing layers required for baseline flow 3 can be reduced to the minimum as our co-design flow. However, our co-design flow can reduce 13.8% wirelength on average compared with baseline flow 3.

TABLE V
COMPARISON OF THE RESULTS BETWEEN THE BASELINE FLOWS AND OUR CO-DESIGN FLOW USING TESTCASES WITH ALL USER I/O PINS UTILIZED

	Baseline flow AMAP-OUT			Baseline flow AMAP			Ours						
testcase	#layers	WL	total runtime(s)	#layers	WL	total runtime(s)	#layers	WL	Obj.	#iter.	PBC runtime(s)	routing runtime(s)	total runtime(s)
T1	3	2547.8	25.60	3	2537.4	7.15	3	2496.4	2499.3	52	13.49	34.49	47.98
T2	3	2606.6	33.04	3	2555.2	51.31	3	2544.0	2504.8	108	28.32	23.19	51.51
T3	3	2635.8	37.95	3	2650.6	330.53	3	2552.2	2523.4	141	36.97	33.80	70.77
T4	3	2744.8	2931.73	3	2712.4	2440.77	3	2593.2	2536.3	90	23.73	67.48	91.21
V1	3	1980.4	1.64	3	1940.4	1.56	2	1718.0	1724.7	28	1.04	1.39	2.43
V2	3	2070.1	190.30	3	1967.4	49.89	2	1734.6	1730.7	37	1.42	5.41	6.83
V3	3	2076.4	473.47	3	1995.3	48.51	2	1793.6	1737.9	26	0.96	56.66	57.62
V4	3	2017.8	68.22	3	2013.8	84.45	2	1811.0	1740.4	58	2.23	72.09	74.32
X1	3	2021.1	60.41	3	1925.2	62.04	2	1664.4	1658.3	24	0.94	5.47	6.41
X2	3	2062.5	39.33	2	1928.5	25.54	2	1679.6	1661.7	57	2.26	20.37	22.63
X3	3	2140.8	41.26	2	2200.0	21.20	2	1692.2	1667.5	15	0.57	23.47	24.04
X4	3	2186.8	683.86	2	2143.0	62.71	2	1705.8	1673.5	26	1.07	47.33	48.40
average	3.0	2257.6	382.23	2.75	2214.1	265.47	2.33	1998.8	1971.5	55.2	9.42	32.60	42.01
ratio	1.29	1.13	9.10	1.18	1.11	6.32	1	1					1

TABLE VI
COMPARISON OF THE RESULTS BETWEEN THE BASELINE FLOWS AND OUR CO-DESIGN FLOW USING TESTCASES WHERE NOT ALL USER I/O PINS ARE UTILIZED

	Baseline flow AMAP-OUT			Baseline flow AMAP			Baseline flow 3			Ours						
testcase	#layers	WL	total runtime(s)	#layers	WL	total runtime(s)	#layers	WL	total runtime(s)	#layers	WL	Obj.	#iter.	PBC runtime(s)	routing runtime(s)	total runtime(s)
T5	3	1911.8	30.21	3	1931.8	19.16	3	1976.2	71.35	3	1879.6	1885.1	136	42.94	8.18	51.12
T6	3	2061.0	20.23	3	2065.2	27.72	3	2136.6	53.20	3	2047.0	2038.0	119	37.52	19.04	56.56
T7	3	2318.0	376.88	3	2318.6	562.29	3	2360.0	31.27	3	2272.8	2242.1	73	22.85	24.09	46.94
V5	2	1566.4	80.37	2	1539.9	18.20	2	1533.0	18.92	2	1300.6	1323.0	36	1.53	5.91	7.44
V6	2	1742.2	61.85	2	1701.4	18.08	2	1715.4	22.98	2	1449.0	1473.5	40	1.71	8.70	10.41
V7	3	1912.6	2921.41	3	1895.8	693.75	2	1889.0	38.41	2	1593.8	1616.6	45	1.91	9.73	11.64
X5	2	1795.2	41.74	2	1641.2	30.87	2	1760.4	6.28	2	1399.6	1417.7	26	1.04	3.72	4.76
X6	2	1975.3	67.77	2	1798.2	49.62	2	1956.3	19.93	2	1510.8	1527.2	44	1.77	31.60	33.37
X7	3	2130.2	354.86	2	1889.4	41.85	2	2115.2	41.04	2	1583.2	1581.8	79	3.23	46.63	49.86
average	2.56	1934.7	439.48	2.44	1864.6	162.39	2.33	1938.0	33.71	2.33	1670.7	1678.3	66.4	12.72	17.51	30.23
ratio	1.10	1.16	14.54	1.05	1.12	5.37	1	1.16	1.12	1	1					1

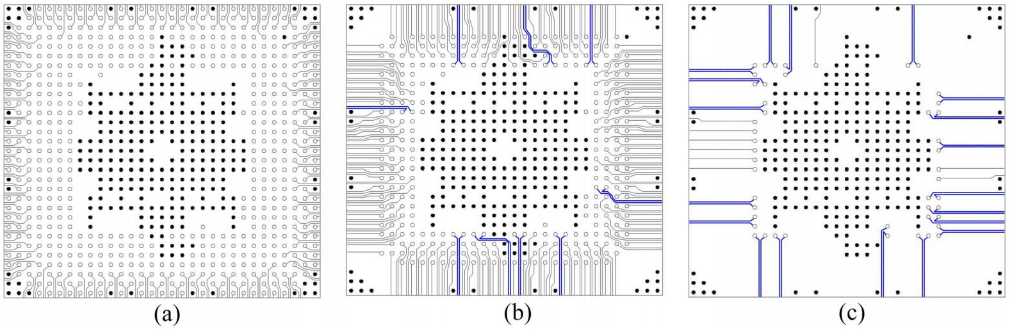


Fig. 16. Routing results for testcase T5 by baseline flow AMAP-OUT. (a) Layer 1. (b) Layer 2. (c) Layer 3. The routing wires of differential pairs are denoted by bold blue lines. P/G pins are denoted by solid black circles.

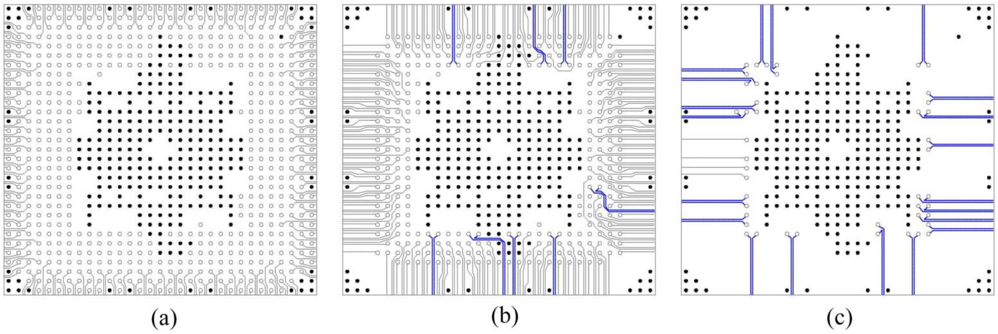


Fig. 17. Routing results for testcase T5 by baseline flow AMAP. (a) Layer 1. (b) Layer 2. (c) Layer 3. The routing wires of differential pairs are denoted by bold blue lines. P/G pins are denoted by solid black circles.

It shows that an appropriate blind-via depth for each signal can result in shorter wirelength in escape routing. Overall, the objective value of our pin assignment and blind-via usage co-optimization is very close to the wirelength in escape routing. For the FPGA package, Figs. 16 and 17 show the routing results for testcase T5 by baseline flows AMAP-OUT and

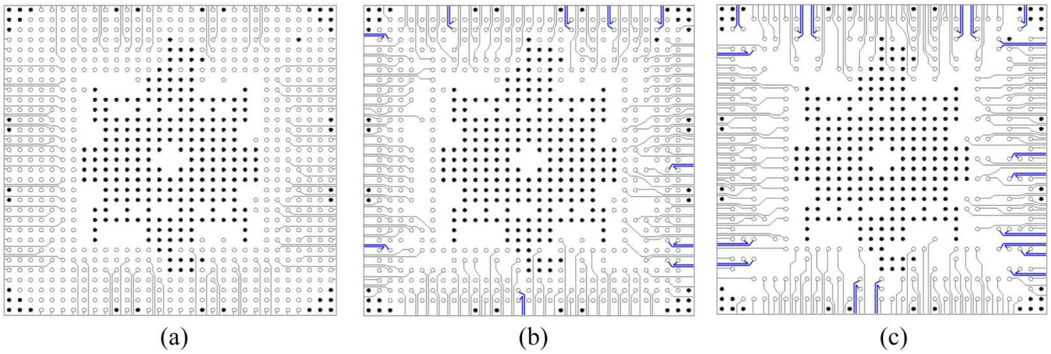


Fig. 18. Routing results for testcase T5 by our co-design flow. (a) Layer 1. (b) Layer 2. (c) Layer 3. The routing wires of differential pairs are denoted by bold blue lines. P/G pins are denoted by solid black circles.

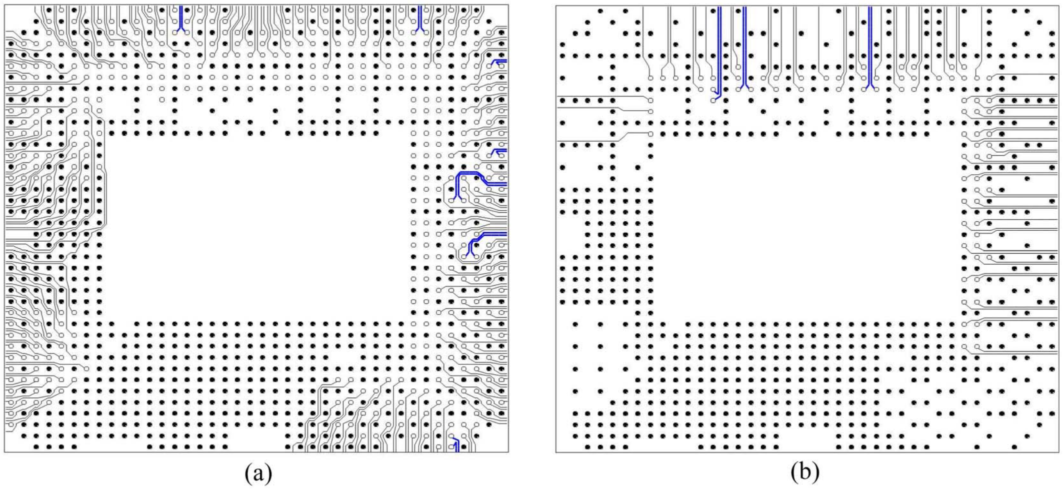


Fig. 19. Routing results for testcase V5 by baseline flow AMAP-OUT. (a) Layer 1. (b) Layer 2. The routing wires of differential pairs are denoted by bold blue lines. P/G pins are denoted by solid black circles.

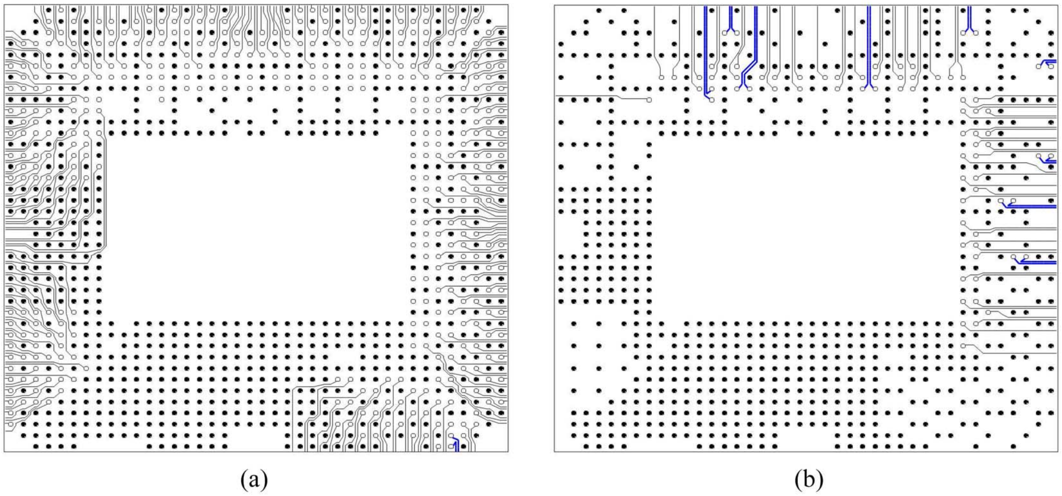


Fig. 20. Routing results for testcase V5 by baseline flow AMAP. (a) Layer 1. (b) Layer 2. The routing wires of differential pairs are denoted by bold blue lines. P/G pins are denoted by solid black circles.

AMAP, respectively. Fig. 18 shows the routing results for testcase T5 by our co-design flow. For the FCLGA 1155 package, Figs. 19 and 20 show the routing results for testcase V5 by baseline flows AMAP-OUT and AMAP, respectively. Fig. 21 shows the routing results for testcase V5 by our co-design flow.

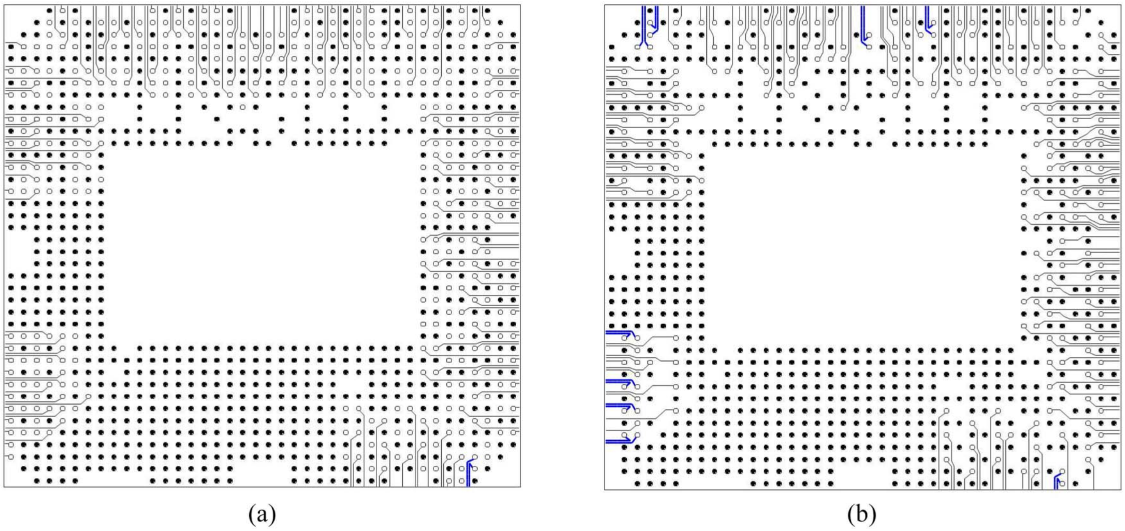


Fig. 21. Routing results for testcase V5 by our co-design flow. (a) Layer 1. (b) Layer 2. The routing wires of differential pairs are denoted by bold blue lines. P/G pins are denoted by solid black circles.

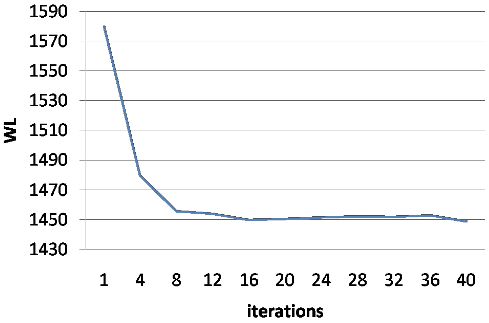


Fig. 22. Wirelength versus iterations for testcase V6 during the pin assignment and blind-via usage co-optimization stage.

The escape routing wirelength versus iterations for testcase V6 during the pin assignment and blind-via usage co-optimization stage in our co-design flow is shown in Fig. 22. The wirelength decreases substantially in early iterations and then converges to the minimum value. It shows that our iterative pin assignment and blind-via usage co-optimization is effective for wirelength reduction.

V. Conclusion

In this paper, we proposed a package-board co-design method to solve the problem of pin assignment and escape routing considering differential pairs and blind-via usage in PCB. To the best of our knowledge, this is the first work that considers differential pairs and blind-via usage in both pin assignment and escape routing for grid pin array. Experimental results demonstrate that our package-board co-design flow can achieve shorter wirelength and reduce the number of layers required compared with competitive baseline flows. For future work, we may target staggered pin array in our package-board co-design flow.

References

[1] M. F. Yu and W. W. Dai, "Single-layer fanout routing and routability analysis for ball grid arrays," in *Proc. ICCAD*, San Jose, CA, USA, 1995, pp. 581–586.

[2] R. Wang, R. Shi, and C. K. Cheng, "Layer minimization of escape routing in area array packaging," in *Proc. ICCAD*, San Jose, CA, USA, 2006, pp. 815–819.

[3] J. W. Fang, I. J. Lin, Y. W. Chang, and J. H. Wang, "A network-flow-based RDL routing algorithm for flip-chip design," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 8, pp. 1417–1429, Aug. 2007.

[4] X. Liu *et al.*, "Global routing and track assignment for flip-chip design," in *Proc. DAC*, Anaheim, CA, USA, 2010, pp. 90–93.

[5] T. Yan and M. D. F. Wong, "A correct network flow model for escape routing," in *Proc. DAC*, San Francisco, CA, USA, 2009, pp. 332–335.

[6] R. Wang and C. K. Cheng, "Octilinear redistributive routing in bump arrays," in *Proc. GLSVLSI*, Boston, MA, USA, 2009, pp. 191–196.

[7] H. Kong, T. Yan, and M. D. F. Wong, "Optimal simultaneous pin assignment and escape routing for dense PCBs," in *Proc. ASPDAC*, Taipei, Taiwan, 2010, pp. 275–280.

[8] T. Yan, P. C. Wu, Q. Ma, and M. D. F. Wong, "On the escape routing of differential pairs," in *Proc. ICCAD*, San Jose, CA, USA, 2010, pp. 614–620.

[9] T. H. Li, W. C. Chen, X. T. Cai, and T. C. Chen, "Escape routing of differential pairs considering length matching," in *Proc. ASPDAC*, Sydney, NSW, Australia, 2012, pp. 139–144.

[10] J. W. Fang, K. H. Ho, and Y. W. Chang, "Routing for chip-package-board co-design considering differential pairs," in *Proc. ICCAD*, San Jose, CA, USA, 2008, pp. 512–517.

[11] S. I. Lei and W. K. Mak, "Simultaneous constrained pin assignment and escape routing considering differential pairs for FPGA-PCB co-design," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 12, pp. 1866–1878, Dec. 2013.

[12] K. Wang, H. Wang, and S. Dong, "Escape routing of mixed-pattern signals based on staggered-pin-array PCBs," in *Proc. ISPD*, Stateline, NV, USA, 2013, pp. 93–100.

[13] R. Shi and C. K. Cheng, "Efficient escape routing for hexagonal array of high density I/Os," in *Proc. DAC*, San Francisco, CA, USA, 2006, pp. 1003–1008.

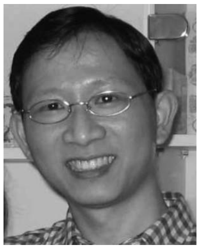
[14] Y. K. Ho, X. W. Shih, Y. W. Chang, and C. K. Cheng, "Layer minimization in escape routing for staggered-pin-array PCBs," in *Proc. ASPDAC*, Yokohama, Japan, 2013, pp. 187–192.

[15] S. Even, A. Itai, and A. Shamir, "On the complexity of timetable and multicommodity flow problems," *SIAM J. Comput.*, vol. 5, no. 4, pp. 691–703, 1976.

[16] *Gurobi Optimizer*. [Online]. Available: http://www.gurobi.com

Seong-I Lei received the B.S., M.S., and Ph.D. degrees in computer science from National Tsing Hua University, Hsinchu, Taiwan, in 2006, 2008, and 2015, respectively.

His current research interests include printed circuit board escape routing and related very large-scale integration physical design automation.



Wai-Kei Mak received the B.S. degree from the University of Hong Kong, Hong Kong, in 1993, and the M.S. and Ph.D. degrees from the University of Texas at Austin, Austin, TX, USA, in 1995 and 1998, respectively, all in computer science.

He is currently a Professor with the Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan. He was an Assistant Professor with the Department of Computer Science and Engineering, University of South Florida, Tampa, FL, USA, from 1999 to 2003. His current research interests include very large-scale integration physical design automation, and computer-aided design for field-programmable technologies.

Dr. Mak was a recipient of the IEEE/ACM Asia and South Pacific Design Automation Conference 2014 Best Paper Award for his work in e-beam lithography throughput optimization, the First Place at the International Conference on Field-Programmable Technology (FPT) 2008 Logic Block Clustering Contest, the Third Place at the IEEE CEDA PATMOS 2011 Timing Analysis Contest, and the Second Place at the TAU 2013 Variation-Aware Timing Analysis Contest. He has served on the Program and/or the Organizing Committee of Asia South Pacific Design Automation Conference, the International Conference on Field Programmable Logic and Applications, and the International Conference on FPT. He was the Technical Program Chair and the General Chair of FPT in 2006 and 2008, respectively. Since 2009, he has been a Steering Committee Member of the International Conference on FPT.