Escape Routing of Mixed-Pattern Signals Based on Staggered-Pin-Array PCBs

Kan Wang
Tsinghua University

Escape Routing of Mixed-Pattern Signals Based on Staggered-Pin-Array PCBs [*]

Kan Wang, Huaxi Wang, Sheqin Dong
Tsinghua National Laboratory for Information Science and Technology(TNList)
Department of Computer Science & Technology
Tsinghua University, Beijing, China 100084
wangkan09@mails.tsinghua.edu.cn, thomas.kasim@gmail.com,
dongsq@mail.tsinghua.edu.cn

ABSTRACT

Escape routing has become a critical issue in high-speed PCB routing. Most of previous work paid attention to either differential-pair escape routing or single-signal escape routing but few considered them together. In this paper, a unified ILP model is used to formulate the problem of escape routing of differential pairs together with single signals (mixed-pattern signals) on staggered pin array. A mixed-pattern escape routing algorithm is proposed to solve the problem and a slice-based heuristic method is presented to speed up the algorithm. Experimental results show that the proposed method is very efficient. It can solve all the test cases in short time and improve wire length and chip area by 13.8% and 13.4% respectively compared to traditional pin array. At the same time, the method can increase routability by 16.3% and reduce the wire length by 9.3% compared to a two-stage method.

Categories and Subject Descriptors

B.7.2 [**Integrated Circuits**]: Design Aids

General Terms

Algorithm, Design, Performance

Keywords

PCB Routing, Escape Routing, Differential Pair, Staggered Pin Array, Mixed-Pattern Signals

1. INTRODUCTION

High speed printed circuit board (PCB) routing has become more and more difficult for manual design due to increased pin count and dwindling routing resource. To address the problem, many methods were proposed, including PCB pin array structure [1]-[2] and escape routing algorithms [3]-[12].

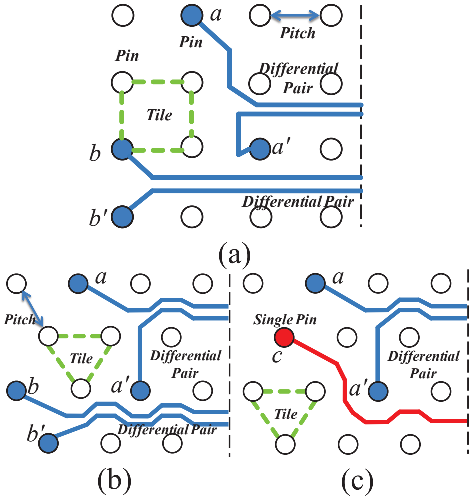


Figure 1: Escape routing pattern: (a) single-pattern escape routing on grid pin array (b) single-pattern escape routing on staggered pin array (c) mixed-pattern escape routing on staggered pin array

For pin array structure, although *grid pin array*(GPA)[3] has been widely used, it cannot satisfy the demands of the ever-increasing pin number on PCB. Then another pin array structure, *staggered pin array*(SPA) was formed [1]. Compared to GPA, the SPA can increase pin density greatly under the similar number of pins and same area constraints [2]. Due to the different routing resources and constrains, the routing on SPA will be much different.

For escape routing, differential-pair escape routing (two nets for each signal) [4]-[7] and single-signal escape routing (one net for each signal) [8]-[12] have been developed on GPA or SPA in recent years. However, there are still some disadvantages. For GPA, the previous works focused on only differential-pair escape routing or only single-signal escape routing(as shown in Fig.1(a)) and did not consider them together. For SPA, only single-signal escape routing was developed and there is no work on differential-pair escape routing on SPA, which is illustrated in Fig.1(b). Furthermore, to the best of our knowledge, there is still no work to consider the escape routing of both differential pairs and single sig-

[*] This paper is supported by MOST of China project 2011DFA60290 and NSFC 61176022

ISPD'13, March 24–27, 2013, Stateline, Nevada, USA.
Copyright 2013 ACM 978-1-4503-1954-6/13/03 ...$10.00.

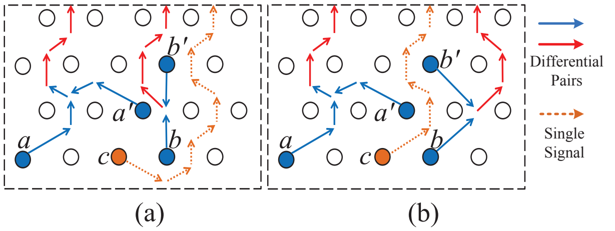


Figure 2: Routing examples (a)single-pattern escape routing will greedily optimize the routing of pin *a* and *a'* which can lead to congestion of single signal *c* while (b)mixed-pattern escape routing will optimize both signals

nals simultaneously on neither GPA nor SPA. Because of the high noise immunity and low electromagnetic interference, differential pairs are always used for the high-speed signal transmission on a PCB. However, due to the limitation of resources, not all signals will be transmitted by differential pairs. As a result, the signals of differential pairs and single signals will coexist on board and hence the research on escape routing of both differential-pair and single signals simultaneously will be quite valuable. For convenience, the escape routing only for differential pairs or only for single signals is referred as *single-pattern escape routing* and the escape routing of both differential-pair and single signals simultaneously is referred as *mixed-pattern escape routing.* Note that the *single-pattern escape routing* is just a special case of *mixed-pattern escape routing.* In this paper, the mixed-pattern escape routing problem on SPA is considered, as shown in Fig.1(c).

Although network flow algorithm [4]-[7] has been successfully applied to single-pattern escape routing on GPA, however, it is not suitable for mixed-pattern escape routing. Because on one hand, the two kinds of signals take different routing resource: differential pair takes two units of resource while single signal takes one, and on the other hand, they have different constraints. For example, differential pairs should satisfy the length-matching constraint [5] while single signals need not.

One feasible solution for mixed-pattern escape routing is to solve differential-pair routing and single-signal routing separately. However, the routing result cannot be optimized as the routing of the two kinds of signals will influence each other. As a result, the routing of one signal will probably fail if the routing resource is occupied by the other. This is hard to avoid even though congestion-aware method is used. Fig.2(a) shows an example where the optimization of differential pairs will lead to congestion of routing resource and longer routing path of single-signals. On this occasion, a mixed-pattern escape routing, which can optimize both differential-pair routing and single-signal routing as shown in Fig.2(b), is required.

Another feasible solution is to use unified model such as network flows to solve the problem. However, there are some challenges. First, the two kinds of signals are required to satisfy different constraints, which should be formulated in the unified model. Second, as the two signals are independent sources and take different routing resources, the mixed-pattern escape routing problem becomes multi-commodity flow problem which has been proven to be NP-hard. To solve the problem, some efficient methods are required. Besides, some rules such as non-crossing rule should also be satisfied. Since it is more likely to obtain the best solution, in this paper, the unified modeling method based on ILP formulation is adopted, and a slice-based algorithm is presented to solve it.

The major contributions of this paper are summarized as follows:

1. A mixed-pattern escape routing algorithm on staggered pin array is proposed. To the best of our knowledge, this is the first work for mixed-pattern escape routing. Experimental results show that it is very efficient to solve both mixed-pattern routing and single-pattern routing.
2. A unified ILP model is formulated for mixed-pattern escape routing problem. Design rules such as non-crossing rule and length-matching rule[5] are also considered.
3. A slice-based heuristic method is presented to prune the variables of ILP and speed up the solving. The proposed approach can achieve 100% routability for all of ten test cases in reasonable running time.

The rest of the paper is organized as follows. Section II introduces the related work and Section III introduces the routing style in SPA and gives the formulation of mixed-pattern escape routing. In Section IV, the detailed routing algorithm is described and experimental results are shown in Section V. Conclusions are provided in Section VI.

2. RELATED WORK

The related work can be classified into two parts: one is SPA based modeling and escape routing algorithms and the other is GPA based escape routing including differential-pair escape routing and single-signal escape routing.

On one hand, [1] analyzed the properties of the SPA and proposed three escape routing strategies for SPA. [2] proposed an escape routing algorithm based on SPA and showed the advantages of SPA compared to GPA. However, they did not consider the differential-pair routing on SPA.

On the other hand, [8]-[12] proposed network flow based escape routing algorithms on GPA. However, they only focused on single-signal escape routing problem. A work on chip-package-board co-design [7] considered differential pairs, but it paid more attention to co-design instead of the optimization of differential pair escape routing. [4] proposed a negotiated congestion-based differential-pair routing but it did not take length-matching rule into account. A recent work [5] proposed a five-stage algorithm for escape routing of differential pairs considering length matching. However, all of them are based on GPA and cannot be applied to SPA directly. Furthermore, all previous work is for single-pattern escape routing and an algorithm for mixed-pattern escape routing is required.

3. PROBLEM DEFINITION

3.1 Preliminaries

For SPA, triangular tiles are usually modeled to stand for routing resource, as shown in Fig.3. Similarly to [5], in this paper, tile node is used as intermediary and wires are routed via tile nodes. Before we present the definition of the problem, some definitions for routing network are introduced first.

Definition 1 (Staggered Pin Array). *A* $m \times n$ *stag-*

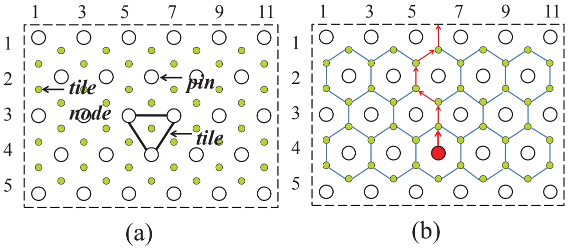


Figure 3: Tile network (a) staggered pin array (b) example in tile routing

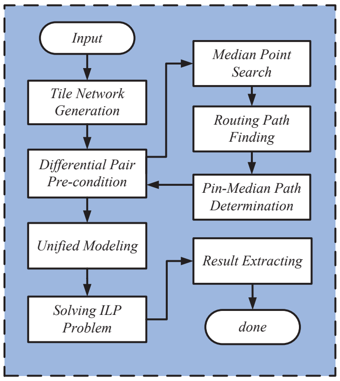


Figure 4: Overview flow of MPERA

gered pin array (SPA) is composed of n rows, and in each row there are m (in odd rows) or $m-1$ pins (in even rows). A triangular tile is composed of three adjacent pins and in each tile there is a tile node, as shown in Fig.3(a).

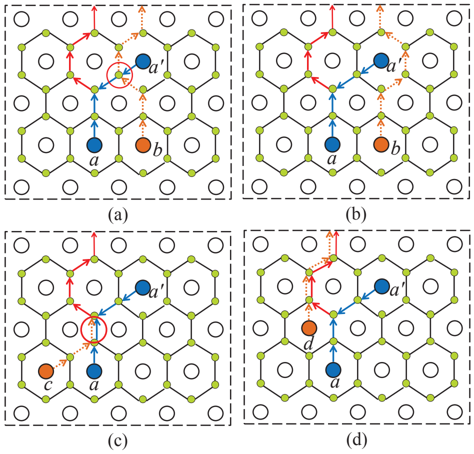
Definition 2 (Tile Network). *The tile network is generated by connecting triangular tiles with each other in the form of hexagons, as shown in Fig.3(b). The edges of tile hexagons are the channels for escape routing and the angle between the routing channels is 120-degree. An example of routing path is also shown in Fig.3(b).*

3.2 Problem Formulation

Based on the definitions, the problem of mixed-pattern escape routing (MPER) can be defined as follows: Given (1) an $m \times n$ staggered pin array; (2) a differential pairs and b single signals to be routed to the boundary; (3) design rules including wire length matching of differential pairs, non-crossing rules and (4) constraints such as routing resource and wire width constraint, the objective is to escape all marked pins to the array boundary with minimized total wire length via the tile network. At the same time, no design rule is violated and 100% routability is guaranteed. In the following section, a mixed-pattern escape routing algorithm (MPERA) is proposed to solve the problem.

4. THE MIXED-PATTERN ESCAPE ROUTING ALGORITHM

As mentioned in Section 1, MPER can be solved by a unified model. However, it is very difficult to take the wire length matching rule of differential pairs into unified model since it is only for differential pairs. To address it, the wire length matching rule will be solved separately. Consider that the wire length matching rule only work at the beginning of differential-pair escape routing, which is before two signals of differential pair meet with each other at median point[5], the escape routing process can be divided into two stages: pin-median-point routing of differential pairs and the mixed-pattern escape routing containing both differential pairs and single signals. The overview flow of MPERA is shown in Fig.4.



(a) (b) (c) (d)

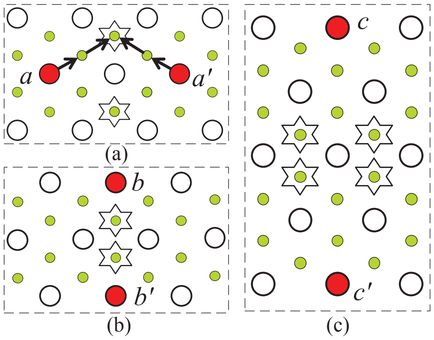
Figure 5: Routing Constraints in MPER: (a) is not allowed due to the non-crossing rule; (c) is not allowed due to differential pair protection constraint; while (b) and (d) are both legal examples

Given the routing requirement, tile network is generated first. Then the differential pairs are preconditioned to reduce the complexity of the problem. Median point searching and pin-median-point path determination algorithms are used for differential-pair escape routing and some strategies guarantee the differential pairs to satisfy the length-matching rule. After that, a unified ILP modeling is proposed to formulate the mixed-pattern escape routing problem and a heuristic algorithm is performed to solve the ILP in reasonable time. If the problem cannot be solved as a result of congestion, then go back to differential-pair precondition. The details of MPERA are described in the following sections.

4.1 Differential Pair Pre-conditioning

Different from single signals, the pins of differential pairs have more constraints. First, the routing of differential pairs should satisfy the length-matching rule. Without considering this, large signal skews will be created, which can lead to degradation of performance. Second, in order to avoid signal crosstalk, before the two signals of differential pair meet with each other, no other signal is allowed to be close to. That is, the routing path between differential-pair pins and median points do not allow other signals to cross or go along, as shown in Fig.5. In this paper, the constraint is named as *differential pair protection constraint.*

It is hard to take these constraints into a unified modeling. To solve it, the pin-median-point routing is handled separately. The median point routing, which routes median point to boundary, is solved with single-signal escape routing together through a unified modeling. Since the pin-median-



(a)

(b)

(c)

Figure 6: Simple cases for median points searching

point paths occupies routing resource, it will directly impact solution quality of mixed-pattern escape routing. Therefore, it is quite necessary to reduce the total length of pin-median-point paths. In this section, a two-stage pin-median-point routing is proposed, including median point searching and pin-median-point path determination. The detailed algorithms will be introduced as follows.

4.1.1 Median point searching algorithm

An effective method was proposed[5] to find median point candidates for each pair. However, it is based on GPA. For SPA, it would be different and the previous method cannot be adopted directly due to different structure and different constraints. In this section, a median point searching method for staggered pin array is proposed.

Let (${x_a}^p$,${y_a}^p$) and (${x_b}^p, {y_b}^p$) be the coordinates of two pins of differential pair. Based on the features of the staggered pin array, the median point searching can be solved according to the following six cases:

Case 1: ${y_a}^p = {y_b}^p$: there are two min-cost median point candidates, which lie on the midperpendicular between two pins, as shown in Fig.6(a).

Case 2: ${x_a}^p = {x_b}^p$ and $|{y_a}^p = {y_b}^p|$ is not multiple of 4: There are two min-cost median point candidates, which lie on the line between two pins, as shown in Fig.6(b).

Case 3: ${x_a}^p = {x_b}^p$ and $|{y_a}^p = {y_b}^p|$ is multiple of 4: There are four min-cost median point candidates, which lie around the middle pin of the two pins, as shown in Fig.6(c).

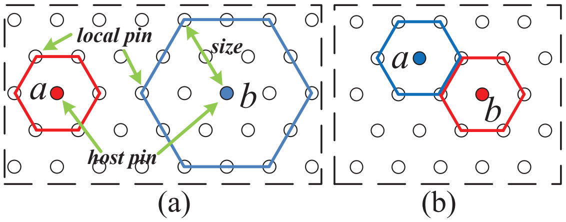
Before we formally investigate the following three cases, the *pin hexagon* and *minimum intersecting hexagon* are first defined:

Definition 3 (Pin Hexagon). *One pin hexagon is composed of certain pins which have the same distance to this pin. The pin in the center is called as host pin while the pins around are called as local pin, as shown in Fig.7(a). The size of pin hexagon is determined by the distance between local pin and host pin.*

Definition 4 (The Minimum Intersecting Hexagon). *Given a differential pair, the intersecting hexagons of the pair are two intersecting or adjacent hexagons with the same size and the minimum intersecting hexagons are those with the minimum size, as shown in Fig.7(b).*

Based on the definitions, the following three cases can be formulated:

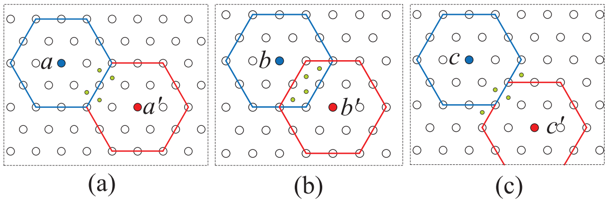
Case 4: ${x_a}^p \neq {x_b}^p$, ${y_a}^p \neq {y_b}^p$ and minimum intersection hexagons are adjacent: the candidates lie in each hexagon and beside the adjacent line of the hexagons, as shown in Fig.8(a).



(a)

(b)

Figure 7: Pin hexagon (a) examples of pin hexagon (b) minimum intersection hexagons



(a)

(b)

(c)

Figure 8: Complicated cases for median points searching

Case 5: ${x_a}^p \neq {x_b}^p$, ${y_a}^p \neq {y_b}^p$ and minimum intersection hexagons are intersectional to each other: The candidates lie in the intersection region of minimum intersection hexagons of two pins, as shown in Fig.8(b).

Case 6: ${x_a}^p \neq {x_b}^p$, ${y_a}^p \neq {y_b}^p$ and minimum intersection hexagons are partly adjacent: The candidates lie around the middle point of two pins and beside the adjacent line of the hexagons, as shown in Fig.8(c).

Based on the above cases, several median point candidates can be obtained, and then the corresponding pin-median paths can be found according to the relative positions between pins and median points. A dynamic programming algorithm can easily solve the pin-median paths finding for each differential pair with length-matching constraint. Then the obtained paths will be treated as path candidates and the following sections will introduce how to select the final paths from all candidates.

4.1.2 Acute-angle Avoidance for Differential Pairs

Though a lot of path candidates are found in the previous section, some of them are not legal. It's easy to find that based on SPA, the acute-angle(60-degree) paths are possibly generated, especially for the differential pairs whose pins are close to each other. The acute-angle routing results will reduce the strength of signals and even cause undercutting or over etching of the circuitry. Therefore, it is necessary to reduce the numbers of acute-angle routing. In this paper, we bring in *path priority* for the selection of path candidates. The paths without acute-angles will be assigned high priorities. And the higher priority a path possesses, the higher possibility it will be selected as a routing solution. This method can remove the acute-angles by more than 50%. For the rest unavoidable cases, the 60-degree angle can be split into two 120-degree angles by adding an additional segment with a little wire length sacrificed. We don't discuss this in detail as the acute-angle issues are not emphasis of this paper.

4.1.3 *Simultaneously Median Point and Shortest Pin-median-point Path Determination*

After the acute-angles removal, the paths with high priority will be considered as the final candidates. Then differential pairs can be classified into K groups according to the crossing possibility of path candidates while K is the maximum value that makes the paths in groups without crossing with each other. A similar method as [5] can be used to solve it.

After that an ILP formulation is used to determine median points and pin-median-point paths for all differential pairs in each group.

Assume that each group G_k contains a_k differential pairs. For each pair i, there are n_{ki} path candidates. l_p denotes the sum of the length of path p from differential pair pin to median point and the distance of median point to the nearest boundary.

For group G_k, binary decision variables x_{ip} ($1 \leq i \leq a_k$, $1 \leq p \leq n_{ki}$) are defined such that x_{ip} is 1 if path p is selected for differential pair i, and x_{ip} is 0 otherwise. For each differential pair, one single path is assigned, so:

$$\sum_{p=1}^{n_{ki}} x_{ip} = 1 \ s.t. \ x_{ip} = 0 \ or \ 1, \forall 1 \leq i \leq N \tag{1}$$

Let PCC_k stand for the path crossing cluster for group G_k. For each crossing pair of $path_{ip}$ and $path_{jq}$ in G_k, x_{ip} and x_{jp} should satisfy:

$$x_{ip} + x_{jq} \leq 1, (ip, jq) \in PCC_k \tag{2}$$

Based on Eq.(1)-Eq.(2), the objective can be written by:

$$Min \ \sum_{i=1}^{a_k} \sum_{p=1}^{n_{ki}} x_{ip} \cdot l_p \tag{3}$$

4.2 Unified Modeling for MPER

By solving the previous problem, the median points for differential pairs and the shortest paths for pins to median points can be determined. However, the paths of median points to boundary cannot be guaranteed as all pairs compete for the limited routing resource, especially with single signals considered. Therefore, it is quite necessary to consider them together in a unified model.

The network flow based ILP formulation method [2] has succeeded on the problem of single-pattern routing problem. However, when considering mixed-pattern routing, the situation will be different. First, there are two kinds of input sources including both differential pairs and single signals. Second, as the two kinds of signals take different network resources, more constraints will be brought in to distinguish them. As a result, the problem will become a multi-commodity problem which has been proven to be NP-hard. In this section, we focus on mixed-pattern escape routing problem and propose a unified ILP modeling to formulate it. A heuristic algorithm is also performed to solve the ILP in reasonable time. Some notations are defined for the routing network, as shown in Table 1.

Thereafter, the problem can be formulated as the following ILP objective function:

$$Min \ \alpha \times \sum_{e_{i,j} \in E} l(e_{i,j}) \times f_d(e_{i,j}) + \sum_{e_{i,j} \in E} l(e_{i,j}) \times f_s(e_{i,j}) \tag{4}$$

Table 1: notations of MPER

Parameters of nodes on network graph	
T_D	The set of determined median point of differential pairs
P_S	The set of escaped single signals
T_S	The set of tiles of escaped single signal pins
$Tile$	The set of routing tiles
S_0	The source node of total network
S_D	The source node for differential pair
S_S	The source node for single signal pins
S_t	The sink node of total network
$Path_{DP}$	The set of tiles which has been occupied by pin-median routing of Differential Pair
Parameters of edges on network graph	
E	The set of all edges
e_{S_0,S_D}	The directed edge from S_0 to S_D
e_{S_0,S_S}	The directed edge from S_0 to S_S
e_{S_D,t_i}	Directed edge from S_D to a median point t_i
e_{S_S,p_i}	Directed edge from S_S to a single signal pin p_i
e_{p_i,t_i}	Directed edge from signal pin p_i to nearby tile t_i
e_{t_i,t_j}	Directed edge from an tile node t_i to another t_j
e_{t_i,S_t}	Directed edge from boundary tile t_i to sink node
Parameters of constraints	
$l(e_{i,j})$	The length of edge $e_{i,j}$
$f_d(e_{i,j})$	The flow of differential pair on edge $e_{i,j}$
$f_s(e_{i,j})$	The flow of single signals on edge $e_{i,j}$
$f(e_{i,j})$	The total flow on edge $e_{i,j}$
$c(e_{i,j})$	The capacity of edge $e_{i,j}$
$Integer$	The set of non-negative integers

subject to:

$$\sum_{e_{i,j} \in E} f(e_{i,j}) = 1, \ \forall i \in T_D \bigcup P_S \tag{5}$$

$$\sum_{e_{i,j} \in E} f(e_{i,j}) = |T_D|, \ \forall i \in S_D \tag{6}$$

$$\sum_{e_{i,j} \in E} f(e_{i,j}) = |P_S|, \ \forall i \in S_S \tag{7}$$

$$\sum_{e_{i,j} \in E} f(e_{i,j}) = |T_D| + |P_S|, \ \forall i \in S_0 \tag{8}$$

$$\sum_{e_{i,j} \in E} f(e_{i,j}) = |T_D| + |P_S|, \ \forall j \in S_t \tag{9}$$

$$\sum_{e_{i,j} \in E} f_d(e_{i,j}) = \sum_{e_{j,k} \in E} f_d(e_{j,k}), \ \forall j \in Tile \tag{10}$$

$$\sum_{e_{i,j} \in E} f_s(e_{i,j}) = \sum_{e_{j,k} \in E} f_s(e_{j,k}), \ \forall j \in T_S \bigcup Tile \tag{11}$$

$$2 \times f_d(e_{i,j}) + f_s(e_{i,j}) \leq c(e_{i,j}) \tag{12}$$

$$\sum_{e_{i,j} \in Path_{DP}} f_s(e_{i,j}) \leq 0 \tag{13}$$

$$f_d(e_{i,j}) \geq 0,\ f_s(e_{i,j}) \geq 0 \tag{14}$$

$$f_d(e_{i,j}),\ f_s(e_{i,j}) \in Integer \tag{15}$$

The objective of Eq.(4) is to minimize the total wire-length including both differential pairs and single signals. As we regard the wire-length as criterion of delay, The wire length of differential pairs is not doubled one because we regard the wire-length as criterion of signal delay instead of the cost of routing resource. α is used to adjust the relative weighting between differential pairs and single signals, and in this paper, α is set to 1. Three source nodes are defined to drive the flows. The S_0 node is used for source of all flows and two sub-source nodes S_D, S_S are used for differential pairs and single signals respectively. For each edge, there are two kinds of flows, $f_d(e_{i,j})$ for differential pairs and $f_s(e_{i,j})$ for single signals. Constraints (5)-(9) ensure the 100% routability of each flow by forcing all escaped pins to be routed to the sink and constraints (10)-(11) ensure that each flow satisfies the flow conservation constraint. Constraint (12)-(13) guarantees the capacity constraint and differential pair protection constraint. Constraint (14)-(15) makes the flow on all edges non-negative integers.

4.3 Slice-based MPER Algorithms

However, directly solving the ILP problem is difficult due to the nature of NP-hard. To address the problem, some heuristic strategies are proposed. Considering that the multi-commodity problem can be modified to single commodity problem by adding some constraints, the problem can be divided to three cases and solved respectively as follows.

4.3.1 Modified MPERA for single-pattern escape routing

Considering the single-pattern escape routing for either differential pairs or single signals, the problem will be transformed to the LP problem in [2] and can be solved in polynomial time.

4.3.2 MPERA considering crosstalk between single signals

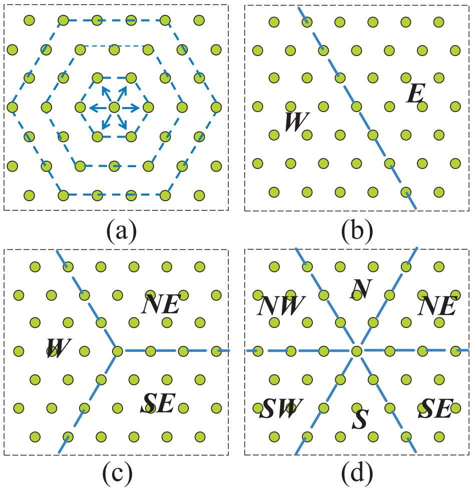
Considering the crosstalk issues between single signals, the single signals cannot share the routing resources as the close distance will lead to crosstalk. While on the other hand, differential pairs can enjoy the same routing resource as the signals of differential pair will be protected by themselves. Especially for the tile capacity of 2, the constraint of (12) will be transformed into:

$$f_d(e_{i,j}) + f_s(e_{i,j}) \leq 1 \tag{16}$$

Then the problem will satisfy the unimodularity property [13] and can be transformed to LP formulation without integer constraints (15) as it is guaranteed to be an integer solution according to unimodularity property. The LP problem can be solved in polynomial time.

4.3.3 Slice-based MPERA without considering the crosstalk between single signals

As complement to the cases of 4.3.1 and 4.3.2, if the crosstalk issues between single signals are not considered, the problem will not satisfy the unimodularity property any more. Then a slice-based heuristic method is proposed to speed-up the algorithm.



(a) (b) (c) (d)

Figure 9: Region division: according to (a) the symmetrical characteristic of SPA, the chip can be partitioned into (b) two regions; (c) three regions; (d) six regions

As it is well known, when a water-drop falls down to the ground, the droplets will spread to all around and if the ground is smooth, the spread will be even enough. Inspired by this physical phenomenon, if the pins of differential pairs and single signals are uniform distributed, the routing paths will be evenly sent out to the boundary, that is, the escape routing satisfies divergence, which can be observed from existing work such as [2]-[5]. In this paper, we refer the property as *divergence property* of escape routing. Though it cannot satisfy all cases, it is still suitable for most cases. For the special cases, we just need some special treatments to handle them.

Algorithm 1 Variable Pruning for ILP

```
Require: $width$, $height$, $sinpins$, $dpairs$, $apartNum$;
$generatePinArray()$;
$generateTileArray()$;
$createRegions$(apartNum);
$isdone = false$;
while $isdone$ do
    for each $i \in [1, apartNum]$ do
        $initialization(i)$;
        $generateDpair(i)$;
        $generaterSinpins(i)$;
        $IDCoorMapping(i)$;
        $createEdges(i)$;
        $ILPEscapeRouting(i)$;
        $resultsExtracting(i)$;
    end for
    if ILP is optimized solved for all regions then
        $isdone = true$;
    else
        $redistributeFailurePoints()$;
    end if
end while
$mergeResults()$;
$outputAllPaths()$;
```

Table 2: Effect of escape routing algorithm on staggered pin array

Benchmark information			[5]			Ours			
Benchmark	Dpair num	Array size	Avg. len.	Run-time (s)	Area	Pin Array len.	Avg. len.	Run-time (s)	Area
case1	10	11x8	2.7	<1	70	2.67	2.31	<1	60.62
case2	18	22x7	4.0	<1	126	4.00	3.46	<1	109.12
case3	18	18x12	4.8	<1	187	4.74	4.10	<1	161.95
case4	8	11x3	2.0	<1	20	1.33	1.15	<1	17.32
case5	11	14x3	2.6	<1	26	1.76	1.52	<1	22.52
case6	11	17x6	4.1	<1	80	4.18	3.62	<1	69.28
case7	18	17x6	3.0	<1	80	3.04	2.63	<1	69.28
case8	20	9x16	3.5	<1	120	3.50	3.03	<1	103.92
case9	20	8x15	3.3	<1	98	3.30	2.86	<1	84.87
case10	60	35x35	12.2	<1	1156	13.50	11.68	2	1001.13
Ratio			1	0.91	1	0.996	0.862	1	0.866

Considering *divergence property*, the problem can be divided into several subproblems and solved individually. Specifically, for the case of SPA, the whole region can be divided into 2, 3 or 6 sub-regions according to the characteristics of hexagons, as illustrated in Fig.9. Due to this, for each subproblem, the variables of ILP can be reduced by at least 50% to 87%.

The detailed algorithm is shown in Algorithm 1. Initially, the pin array and tile network are generated according to the input information(line 1-2). The chip is partitioned into $apartNum$ regions(line 3), each of which will solve the ILP independently. Some initialization is done to store the information of pin array and tile array in each region(line 7). Then differential pairs and single signals are also classified into each region according to coordinates of median point and pin respectively(line 8-9). Correspondingly, the IDs of pins are also re-mapped to a new set(line 10). After that, edges are generated for the network flow based ILP formulation(line 11). Then an ILP solver is performed to solve the problem(line 12). If the ILP is optimally solved for all regions, then go to end. However, the region is not always equally partitioned. As a result, some regions may be congested and not able to get a feasible solution. To address it, the failed signals in congested regions will be redistributed heuristically into nearby region which has the most routing resource, until the problem is solved successfully(line 15-19). Finally, the results in each region are merged and final routing results are stored(line 21-22).

5. EXPERIMENTAL RESULTS

In this section, two experiments are conducted to show the efficiency of the proposed algorithms on benchmarks generated according to [5]. All experiments are implemented in C++ on a workstation with Intel Xeon 2.40GHz CPU and 12GB physical memory. The capacity of edge on tile network is set to 2 and only one routing layer is considered. *lp_solve*[14] is used for the ILP and LP solving. The wire lengths of experimental results are all normalized values.

5.1 Effect of MPERA on Staggered Pin Array

In this section, we show the effect of the proposed escape routing algorithm on single-pattern escape routing based on staggered pin array.

In Table 2, *Dpair num* denotes the number of differential pairs, while *Sin.num* denotes the number of single signals, *Array size* shows the size of the staggered pin array and *Avg.len* shows the average number of tiles from two pins to the boundary. To make the comparison more fair, we create ten similar grid pin arrays according to the test cases in [5] and generate the ten benchmarks by shifting specific columns of the grid pin arrays. Table 2 shows that the error of our generated grid pin array compared to [5] is only 0.4%. *Run-time*(s) gives the running time for routing in second and *Area* gives the total chip area based on the pin array pattern.

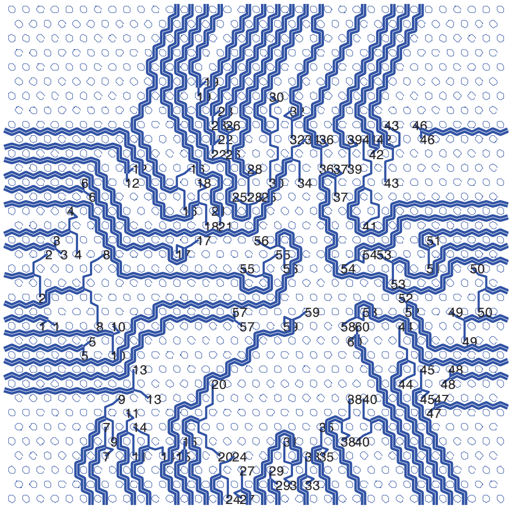


Figure 10: Escape routing results of case10

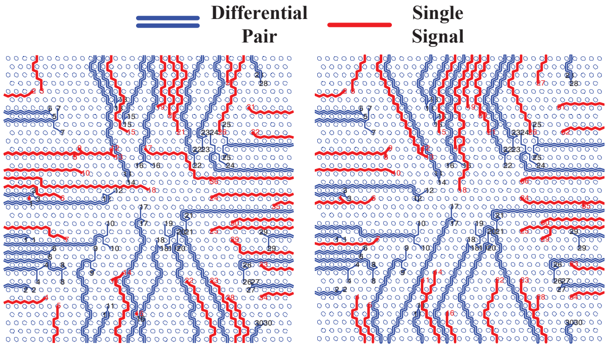
The proposed method is very efficient on differential-pair escape routing. It can solve the routing problem in quite short time. Furthermore, compared to traditional pin array, the staggered pin array can reduce wire length by about 13.8% on average and chip area by 13.4%. Fig.10 shows the routing result of case 10.

5.2 Effect of MPERA on Mixed-Pattern Escape Routing

In this section, the effect of the proposed mixed-pattern escape routing is shown. To make the comparison more clear, we also implement a two-stage network flow based routing algorithm named *TS* to be a competitor to show

Table 3: Effect of the proposed mixed-pattern escape routing

Benchmark information				Two-stage method				Ours			
Benchmark	Dpair num	Sin. num	array size	DPair length	Sin. length	Routability (%)	Runtime (s)	DPair length	Sin. length	Routability (%)	Runtime (s)
case1-2	5	5	11x8	8.8	8.5	80%	<1	8.8	8.2	100%	<1
case2-2	9	9	22x7	5.78	6.1	88%	<1	5.78	5.67	100%	<1
case3-2	9	9	18x12	10.44	7.5	88%	<1	10.44	7.33	100%	<1
case4-2	5	5	11x3	2.4	2.6	90%	<1	2.4	2.2	100%	<1
case5-2	6	5	14x3	3.67	2	100%	<1	3.67	2	100%	<1
case6-2	6	7	17x6	7.17	5.67	85%	<1	7.67	4.43	100%	<1
case7-2	9	9	17x6	4.67	4.88	88%	<1	4.78	4.55	100%	<1
case8-2	10	10	9x16	6.4	7	90%	<1	6.4	6.5	100%	<1
case9-2	10	10	8x15	5.7	6.63	80%	<1	5.8	6.1	100%	<1
case10-2	30	36	35x35	18.33	16.12	88%	2	18.4	14.52	100%	3
case11-2	50	66	35x35	15.92	12.61	69%	2	16.0	11.36	100%	13
Ratio				1	1.093	1	-	1.010	1	1.163	-



(a) (b)

Figure 11: Escape routing results of case 10-2 (a) two-stage network algorithm (b) the proposed algorithm

the advantage of our method. Specifically, *TS* is to solve differential-pair routing and single-signal routing respectively.

The results are shown as Table 3, where we can see that *TS* cannot solve all test cases, while the proposed method can solve all single-signal escape routing. The routability is increased by about 16.3%. Furthermore, for the routed pins, the proposed method can also reduce the average wire length of single signals by 9.3% on average and 22.0% at most. At the same time, compared to the pure ILP solving which cannot converge even during thousands of seconds for large cases, the proposed method can simultaneously solve both differential pairs and single signals in short time.

Fig. 10 shows the escape routing result of case 10-2 using two-stage network algorithm(Fig.11 (1)) and the proposed algorithm based on 3 sub-regions(Fig.11 (2)).

6. CONCLUSION

Staggered pin array has become more and more popular for printed circuit board (PCB), as it has higher pin density than grid pin array. At the same time, differential-pair is a good method to increase noise immunity of signal for high-speed signal transmission on. In this paper, an algorithm for escape routing of simultaneously differential-pair and single signals is proposed on staggered pin array based PCB. Experimental results show that the proposed method can solve both single-pattern and mixed-pattern effectively.

7. REFERENCES

[1] R. Shi and C.-K. Cheng. Efficient escape routing for hexagonal array of high density I/Os. *DAC*, 2006, pp. 1003-1008.

[2] Y. Ho, H. Lee, and Y. Chang. Escape Routing for Staggered-Pin-Array PCBs. *DAC*, 2011, pp. 306-309.

[3] T. Yan and M. D.-F. Wong. Recent research development in PCB layout. *ICCAD*, 2010, pp. 398-403.

[4] T. Yan, Pei-Ci Wu, Qiang Ma, Martin D. F. Wong. On the Escape Routing of Differential Pairs. *ICCAD*,2010, pp. 614-620.

[5] T. Li, W. Chen, X. Cai, T. Chen. Escape Routing of Differential Pairs Considering Length Matching. *ASP-DAC*, 2012, pp 139-144.

[6] T. Yan, Martin D. F. Wong. A Correct Network Flow Model for Escape Routing. *DAC*,2009, pp. 332-335.

[7] J.-W. Fang, K.-H. Ho, and Y.-W. Chang. Routing for chip-package-board co-design considering differential pairs. *ICCAD*,2008, pp. 512-517.

[8] 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. Computer-Aided Design*, vol. 26, no. 8, Aug. 2007.

[9] J.-W. Fang and Y.-W. Chang. Area-I/O flip-chip routing for chip-package co-design. *ICCAD*, pp. 518-522, 2008.

[10] M.-F. Yu, J. Darnauer, and W. W.-M. Dai. Interchangeable pin routing with application to package layout. *ICCAD*, 1996, pp. 668ÍC673.

[11] R. Wang, R. Shi, and C.-K. Cheng. Layer minimization of escape routing in area array packaging. *ICCAD*, 2006, pp. 815ÍC819.

[12] M.-F. Yu and W. W.-M. Dai. Single-layer fanout routing and routability analysis for ball grid arrays. *ICCAD*, 1995, pp. 581ÍC586.

[13] R. J. Vanderbei. Linear Programming: Foundations and Extensions. *Springer*, 2007.

[14] "Lp_solve". http://www.lpsolve.sourceforge.net/5.5/