Escape Routing of Differential Pairs Considering Length Matching

Tai-Hung Li, Wan-Chun Chen, Xian-Ting Cai, and Tai-Chen Chen
Department of Electrical Engineering, National Central University, Taoyuan, Taiwan
{975401006, 995201126, 100521026}@cc.ncu.edu.tw, tcchen@ee.ncu.edu.tw

***Abstract*—** The escape routing problem for PCB designs has been extensively studied in literature. Although industrial tools and few studies have worked on the escape routing of differentials pairs, the routing solutions are not good enough by previous methods. In this paper, we propose an escape routing approach of differential pairs considering length matching. The approach includes two stages. The first stage is to find min-cost median points which connect two pins by shortest and equal wire lengths while the second stage is adopted a network-flow approach with min-cost max-flow to simultaneously route all differential pairs. Experimental results show that our approach can efficiently and effectively obtain length-matching differential pairs with significant reduction in maximum and average differential-pair skews.

I. INTRODUCTION

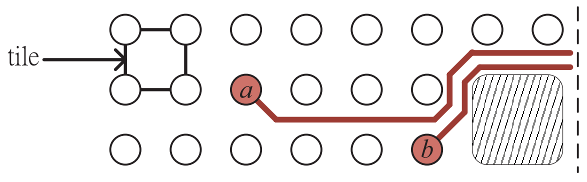
Printed circuit board (PCB) routing becomes more and more complicated and cannot be accelerated effectively without EDA (Electronic Design Automation) tools. The escape routing problem, an important issue in PCB routing, is to route specific pins inside a pin grid to its grid boundary. Many studies handle the escape routing problem and can be classified into ordered escape routing [1]-[5] and unordered escape routing [6]-[13]. The escaped wires around the grid boundary of the ordered escape routing are required to follow some ordering constraints while those of the unordered escape routing are not. In this paper, we focus on the unordered escape routing problem and "escape routing" denotes unordered escape routing hereafter.

For high-speed PCB designs, differential-pair routing [14] is a popular technique because differential pairs have high noise immunity, electromagnetic interference reduction, and ground bounce insensitivity, which are all critical issues in high-speed signal transmission on a PCB. Each differential pair consists of two complementary signals used to transmit one signal. These two signals are transmitted in close proximity along a routing channel. Since the signal wires are routed close to each other, the noises on the channel can be simultaneously absorbed by these two signals. In order to utilize the advantages of differential pairs, the two nets of a differential pair shall be routed parallel to each other with similar wire lengths. Therefore, it is desirable to consider differential pairs with length matching to improve the performance and signal integrity during PCB routing.

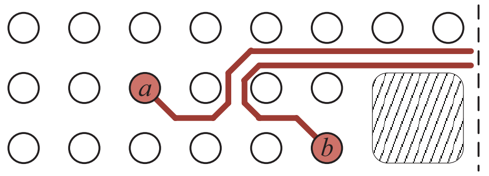
If the two pins of a differential pair are located on adjacent pin grids, routing the two pins to the grid boundary with equal wire length would be easily realized. However, a large portion of differential pairs have significant distance between their two pins in practice designs [15]. Therefore, routing differential pairs considering length matching (equal wire length) is an important problem.

Industrial tools for escape routing propose a solution to handle differential pairs. However, they cannot guarantee that the two nets (from one pin to the grid boundary) of a differential pair have similar wire lengths, as shown in Fig. 1. Fig. 1(a) shows a differential-pair routing result by an industrial tool. As shown in Fig. 1(b), the routing result needs extra routing resource and manually rerouting by an empirical engineer to achieve length matching. In Fig. 1, the hollow and full circles are pins which compose a pin grid. The dotted line at the right side is the grid boundary. A square surrounded by four adjacent pins is called a tile, as shown at the left side of Fig. 1(a). Table I shows the wire lengths and differential-pair skews of Fig. 1. The differential-pair skews are simulated by HSPICE with industrial design specification. The first column denotes two routing types (non-length-matching and length-matching) of differential pairs. The second column shows the wire length of each net. The third column shows the differential-pair skew. As shown in Table I, the length-matching type can obtain a smaller differential-pair skew than the non-length-matching type can. Since a smaller differential-pair skew is one major factor to achieve better performance of differential pairs, we must try to make two nets have similar wire lengths.

This work was partially supported by National Science Council of Taiwan under Grant No's NSC 99-2221-E-008-106 and NSC 100-2221-E-008-069.



(a)



(b)

Fig. 1. Escape routing of differential pairs. The dotted line at the right side is the grid boundary. (a) Routing result obtained by an industrial PCB router. Lengths of the two nets are not matching. (b) Manually rerouting by an empirical engineer for length matching.

TABLE I
SIMULATED RESULTS OF FIG. 1.

	Wire Length of *a*/*b* (mm)	Differential-Pair Skew (ps)
Non-Length-Matching (Fig. 1(a))	22.1/10.9	106.6
Length-Matching (Fig. 1(b))	21.8/21.3	5.9

Fang et al. [16] and Yan et al. [15] also aim the escape routing of differential pairs. However, [16] assumes a monotonic routing style which may away from many practical designs. [15] adopts a negotiated congestion-based routing [17] to find a better location of a meeting point, which connects two pins and becomes the start point of a differential pair to the grid boundary. However, the escape routing of differential pairs considering length matching is not accomplished by [15]. As shown in Fig. 2, the escape routing result of differential pairs obtained by [15] is similar to that obtained by industrial tools (non-length-matching result).

Since industrial tools and previous works cannot solve the escape routing problem of differential pairs considering length matching, we propose an approach to solve the problem automatically. Our approach includes two stages. The first stage is to find all *min-cost median points* which can connect two pins by *shortest* and equal wire lengths. We define a *median point*, which has *equal* wire lengths from this point to the two pins of a differential pair, to distinguish it from a meeting point, which is defined in [15] and just a point between the two pins of a differential pair. The locations of min-cost median points can be computed by an algorithm which will be introduced in section III. After finding min-cost median points, overlapped shortest pin-to-pin paths of multiple differential pairs from one pin to another through one min-cost median point would be considered and be solved by integer linear programming (ILP). The second stage is to route a differential pair from its median point to the grid boundary. We adopt a network-flow approach with min-cost max-flow [15] to simultaneously route all differential pairs and obtain an optimal solution. Experimental results show that our escape routing architecture guarantees that all differential pairs are length-matching. Therefore, the differential-pair skew of each differential pair is minimum.

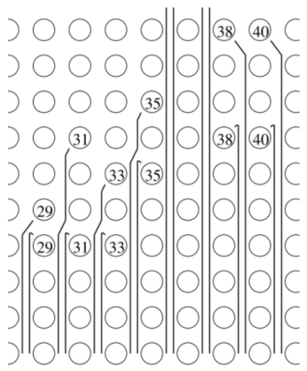


Fig. 2. Partial escape routing result (from [15]) of differential pairs without considering length matching.

The rest of this paper is organized as follows: Section II introduces the problem formulation. Detail algorithms are presented in section III. Experimental results are shown in section IV. Finally, conclusions are then given in section V.

II. PROBLEM FORMULATION

Since differential pairs should be designed carefully to maintain the performance, length matching on differential pairs becomes the constraint to satisfy the design specification including differential-pair skew.

Problem 1: Given *p* differential pairs with pins {(*1a*, *1b*), . . . , (*pa*, *pb*)} in a *r* row by *c* column pin grid, the problem of the escape routing of differential pairs considering length matching is to find a routing path from the two pins to the gird boundary for each differential pair. All differential pairs are length-matching and the total wire length of all routing paths is minimized.

III. ROUTING ALGORITHM

In order to find length-matching differential pairs, we propose a routing algorithm to handle the escape routing problem. The routing algorithm includes five steps: min-cost median point finding, shortest pin-to-pin paths through min-cost median points enumerating, grouping and large group dividing, simultaneously median point and shortest pin-to-pin path determination, and simultaneously median-point-to-grid-boundary path determination. Each step will be introduced in details as follows.

A. Min-Cost Median Point Finding

The first step of our routing algorithm is to find all min-cost median points for each differential pair. A *min-cost* median point for a differential pair is a median point which has the shortest and equal Manhattan distances from the median point to the two pins of the differential pair. Maze routing algorithm with double fan-out reduction [17] can be used in this step. However, maze routing is a time-consuming process. Thus, we propose an efficient algorithm called MCMPF (Min-Cost Median Point Finding) to find all min-cost median points for each differential pair. Fig. 3 shows the algorithm of MCMPF. We define $(x_a^p,\ y_a^p)$ and $(x_b^p,\ y_b^p)$ as coordinates of pins *a* and *b* for differential pair *p*, respectively. Since the wires are routed via passing tiles, we use *tile nodes* to record paths conveniently. According to the relative locations of two pins, four cases are classified in the following.

```
Algorithm: MCMPF ($x_x^p,\ y_a^p,\ x_b^p,\ y_b^p,$)
Input: $x_a^p$ /* x-coordinate of pin a of differential pair p */
       $y_a^p$ /* y-coordinate of pin a of differential pair p */
       $x_b^p$ /* x-coordinate of pin b of differential pair p */
       $y_b^p$ /* y-coordinate of pin b of differential pair p */
1    if $x_a^p = x_b^p$
2        if $|y_a^p - y_b^p|$ is odd
3            2 adjacent tiles in the middle of $y_a^p$ and $y_b^p$;
4        else
5            4 adjacent tiles in the middle of $y_a^p$ and $y_b^p$;
7    else if $y_a^p = y_b^p$
8        if $|x_a^p - x_b^p|$ is odd
9            2 adjacent tiles in the middle of $x_a^p$ and $x_b^p$;
10       else
11           4 adjacent tiles in the middle of $x_a^p$ and $x_b^p$;
12   else
13       if $|x_a^p - x_b^p| + |y_a^p - y_b^p|$ is odd
14           $2 \times (1 + \min\{|x_a^p - x_b^p|,\ |y_a^p - y_b^p|\})$
             adjacent tiles in the middle of two pins;
15       else
16           $\min\{|x_a^p - x_b^p|,\ |y_a^p - y_b^p|\}$
             adjacent tiles in the middle of two pins;
```

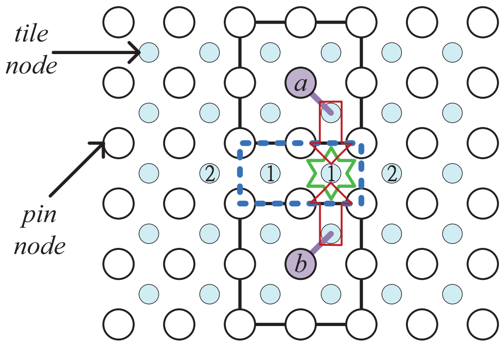
Fig. 3. Algorithm of min-cost median point finding (MCMPF).

Case 1. x_a^p (y_a^p) is equal to x_b^p (y_b^p) and the Manhattan distance between pins *a* and *b* is ***odd***: The number of min-cost median points is two. These two min-cost median points are located in the two adjacent tiles in the middle of the two pins, as shown in Fig. 4(a). The tile nodes surrounded by dotted line are the min-cost median points. An example for connecting a min-cost median point surrounded by the star to the adjacent tile nodes via hollow arrows is also shown in Fig. 4(a).

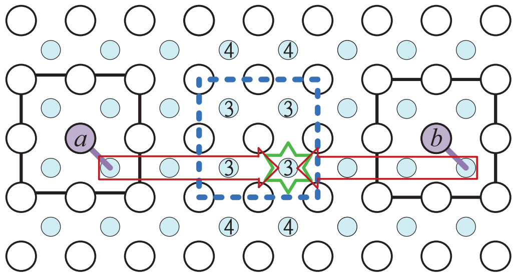
Case 2. x_a^p (y_a^p) is equal to x_b^p (y_b^p) and the Manhattan distance between pins *a* and *b* is ***even***: The number of min-cost median points is four. These four min-cost median points are located in the four adjacent tiles in the middle of the two pins, as shown in Fig. 4(b).

Case 3. x_a^p is different to x_b^p, y_a^p is different to y_b^p, and the Manhattan distance between pins *a* and *b* is ***odd***: The number of min-cost median points is $2 \times \left(1 + \min\{|x_a^p - x_b^p|,\ |y_a^p - y_b^p|\}\right)$. These min-cost median points are located in the $2 \times \left(1 + \min\{|x_a^p - x_b^p|,\ |y_a^p - y_b^p|\}\right)$ adjacent tiles in the middle of the two pins, as shown in Fig. 4(c).

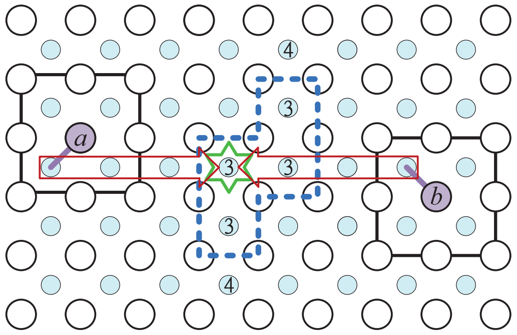
Case 4. x_a^p is different to x_b^p, y_a^p is different to y_b^p, and the Manhattan distance between pins *a* and *b* is ***even***: The number of min-cost median points is $\min\{|x_a^p - x_b^p|,\ |y_a^p - y_b^p|\}$. These min-cost median points are located in the $\min\{|x_a^p - x_b^p|,\ |y_a^p - y_b^p|\}$ adjacent tiles in the middle of the two pins, as shown in Fig. 4(d).



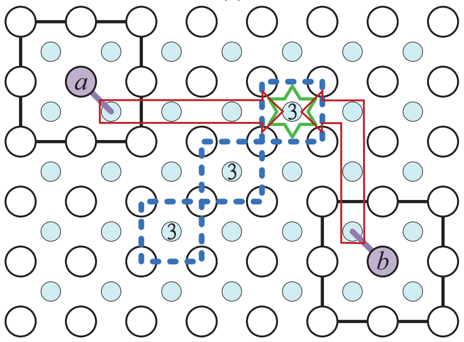
(a)



(b)



(c)



(d)

Fig. 4. Four cases for different locations of two pins. (a) x_a^p (y_a^p) is equal to x_b^p (y_b^p) and the Manhattan distance between pins *a* and *b* is odd. (b) x_a^p (y_a^p) is equal to x_b^p (y_b^p) and the Manhattan distance between pins *a* and *b* is even. (c) x_a^p is different to x_b^p, y_a^p is different to y_b^p, and the Manhattan distance between pins *a* and *b* is odd. (d) x_a^p is different to x_b^p, y_a^p is different to y_b^p, and the Manhattan distance between pins *a* and *b* is even.

For finding equal wire lengths from two pins to median points, the following claim should be satisfied:

Theorem 1: *Algorithm MCMPF guarantees that median points for each differential pair can be found to route equal length wires.*

B. Shortest Pin-to-Pin Paths through Min-Cost Median Points Enumerating

After calculating all min-cost median points of a differential pair, we can enumerate all *shortest pin-to-pin paths*, which are from one pin to another pin through one min-cost median point, of this differential pair. According to enumerating shortest pin-to-pin paths of all differential pairs, respective min-cost median point for each differential pair can be determined by ILP to avoid crossing problems between pin-to-pin paths of any two differential pairs.

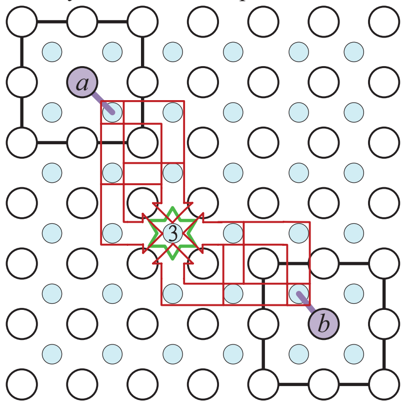


Fig. 5. Existing nine path of one min-cost median point.

For each min-cost median point of a differential pair, it may exist more than one shortest pin-to-pin paths to connect two pins through the min-cost median point. As shown in Fig. 5, pin *a* and the median point surrounded by the star can be connected by three different shortest paths. Pin *b* and the median point surrounded by the star can be connected by three different shortest paths, too. Therefore, there are nine different shortest pin-to-pin paths from pin *a* to pin *b* through the min-cost median point.

Calculating the total number of shortest pin-to-pin paths of a differential pair can be formulated as a permutation combination C_β^α problem. The parameter α denotes the Manhattan distance from the start tile node to the median point tile node. The parameter β denotes the Manhattan distance of x-coordinates (or y-coordinate). All listed paths are the candidate paths for connecting two pins and their median points.

C. Grouping and Large Group Dividing

Since long Manhattan distance of two pins may incur large ILP formulation, we propose two reduced methods, grouping and large group dividing, to reduce the ILP variables and constraints.

C.1. Grouping Method

Since crossing problems between two differential pairs are only needed to be considered when their shortest pin-to-pin paths have probabilities of crossing, we group differential pairs which are overlapped in their possible routing regions. The possible routing region of a differential pair can be calculated by enumerating all shortest pin-to-pin paths of the differential pair. However, the possible routing region of a differential pair can be calculated quickly according to its Manhattan distance of two pins. We define (x_a^p, y_a^p) and (x_b^p, y_b^p) as coordinates of pins *a* and *b* for differential pair *p*, respectively. We also define region R_p as the largest rectangle decided by the coordinate of the adjacent tile nodes of two pins in differential pair *p* (see the dotted line rectangle R_1 in Fig. 6). According to the relative locations of the two pins, three cases are classified in the following.

Case 1. x_a^p (y_a^p) is equal to x_b^p (y_b^p) and the Manhattan distance between pins *a* and *b* is ***even***: The possible routing region is R_p. See the differential pair {2*a*, 2*b*} shown in Fig. 6.

Case 2. Manhattan distance between pins *a* and *b* is ***odd***: The possible routing region is R_p except four tile nodes in the corners. See the differential pair $\{1a, 1b\}$ shown in Fig. 6.

Case 3. x_a^p is different to x_b^p, y_a^p is different to y_b^p, and the Manhattan distance between pins a and b is ***even***: The possible routing region is R_p except tile nodes in the outermost ring. See the differential pair $\{3a, 3b\}$ shown in Fig. 6.

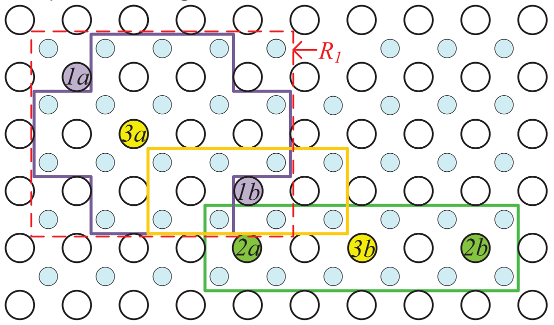


Fig. 6. An example of grouping.

Fig. 6 is an example for grouping differential pairs. There are three differential pairs $\{1a, 1b\}$, $\{2a, 2b\}$, and $\{3a, 3b\}$. The possible routing regions of $\{1a, 1b\}$, $\{2a, 2b\}$, and $\{3a, 3b\}$ can be calculated by the above three cases. The upper-left cross area, the lower-right rectangular area, and the middle rectangular area are possible routing regions of $\{1a, 1b\}$, $\{2a, 2b\}$, and $\{3a, 3b\}$, respectively. Since routing regions of $\{1a, 1b\}$, $\{2a, 2b\}$, and $\{3a, 3b\}$, are overlapped, $\{1a, 1b\}$, $\{2a, 2b\}$, and $\{3a, 3b\}$ are in the same group. Therefore, their median points are needed to be determined simultaneously by ILP.

C.2. Large Group Dividing Method

```
Algorithm: LGD ($g$, $num_g$, $N_{max}$, $r_p$)
Input:  $g$  /* a differential pair group */
        $num_g$  /* the number of differential pairs in group $g$ */
        $N_{max}$  /* the criterion that defines a group as a large one */
        $r_p$  /* the set of shortest pin-to-pin paths of differential pair $p$*/
1 do
2    for each differential pair $p \in g$
3      if existing at least one shortest pin-to-pin path which does not
       overlap with other possible routing regions
4        divide $p$ with related shortest pin-to-pin paths which do not
         overlap with other possible routing regions into a new group;
5        remove $p$ from $g$;
6        update $num_g$;
7 while $num_g > N_{max}$ and at least one $p$ is removed from $g$;
```

Fig. 7. Algorithm of large group dividing (LGD).

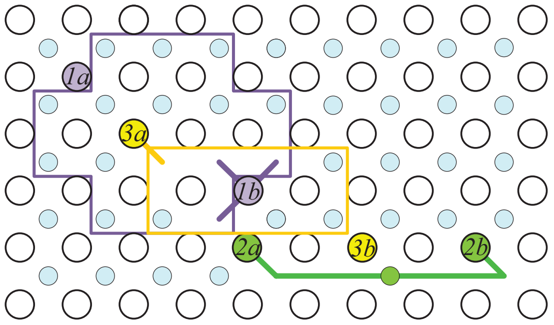


Fig. 8. Example in Fig. 6 using LGD.

Fig. 7 reveals the algorithm of LGD (Large Group Dividing) to divide a differential-pair group, whose number of differential pairs is larger than N_{max}, into smaller subgroups. N_{max} is the criterion that defines a group as a large one. For each differential pair p in group g, if p has at least one shortest pin-to-pin path which does not overlap with other possible routing regions, p will be removed from g accordingly. Besides, p and its shortest pin-to-pin paths which do not overlap with other possible routing regions will be divided into a new group. The above procedure will re-run until the remaining number of differential pairs in g is smaller than than N_{max} or no more differential pair can be removed from g.

Fig. 8 shows the example in Fig. 6 using LGD. Since differential pair $\{2a, 2b\}$ has a shortest pin-to-pin path which does not overlap with other possible routing regions, LGD removes differential pair $\{2a, 2b\}$ from its original group and creates a new group for $\{2a, 2b\}$. Because the group of $\{1a, 1b\}$ and $\{3a, 3b\}$ cannot be divided anymore, all shortest pin-to-pin paths of $\{1a, 1b\}$ and $\{3a, 3b\}$ will be assigned variables by ILP to find solutions without crossing problems. Although LGD can reduce the number of ILP variables, some possible paths may be ignored after LGD.

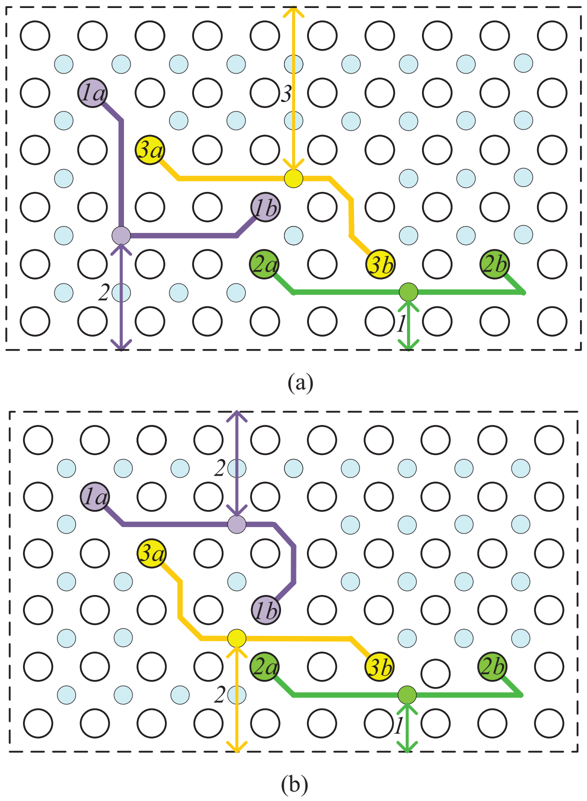


Fig. 9. Two possible pin-to-pin paths for the example in Fig. 8. (a) Solution with longer distance (2+3+1=6) from median points to the grid boundary. (b) Solution with shorter distance (2+2+1=5) from median points to the grid boundary.

D. Simultaneously Median Point and Shortest Pin-to-Pin Path Determination

Since there are several min-cost median points and several shortest pin-to-pin paths can be chosen, we use ILP to find median points and shortest pin-to-pin paths for all differential pairs considering shortest total length from median points to the grid boundary. Fig. 9 reveals two possible pin-to-pin paths for the example in Fig. 8. Fig. 9(a) is one solution with longer distance from median points to the grid boundary. The distance between the grid boundary and median points of differential pairs $\{1a, 1b\}$, $\{2a, 2b\}$, and $\{3a, 3b\}$ are 2, 3, and 1, respectively. Total distance is 6 $(2 + 3 + 1)$. A better solution is shown in Fig. 9(b). The total distance between differential pairs to the gird boundary is 5 $(2 + 2 + 1)$. The configuration in Fig. 9(b) will be selected by ILP due to the shorter distance.

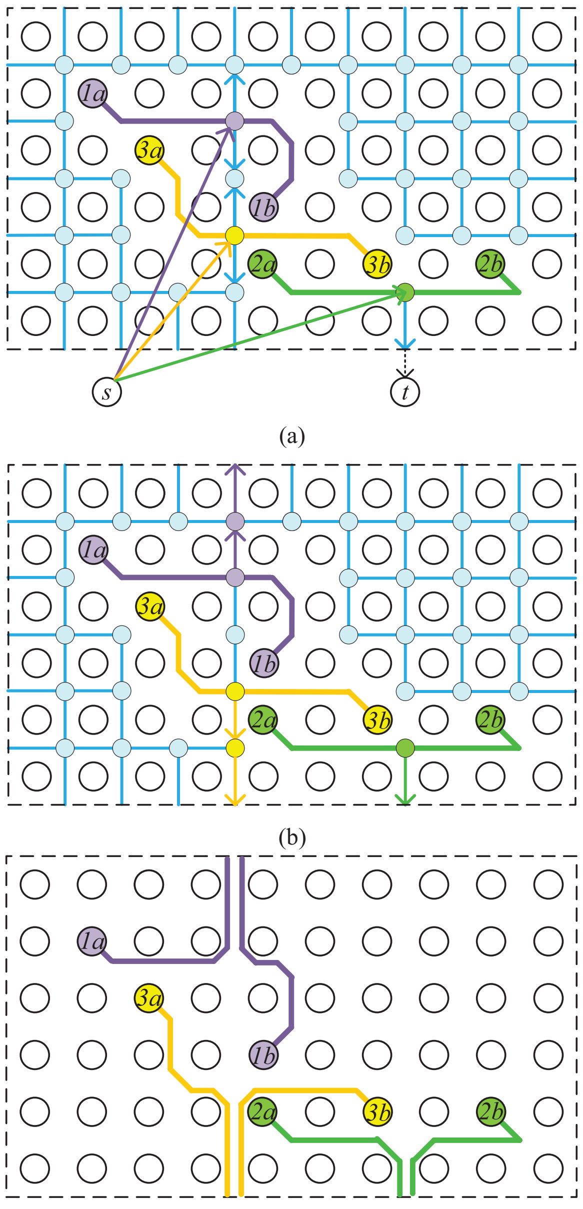


Fig. 10. Simultaneously median-point-to-grid-boundary path determination by network-flow routing. (a) Network-flow construction. (b) Result after running network-flow. (c) Final routing solution.

E. Simultaneously Median-Point-to-Grid-Boundary Path Determination

After determining median points and shortest pin-to-pin paths for all differential pairs, network-flow approach is used to connect median points to the grid boundary. To tally with network-flow operating format, the first action is using *tile nodes* to denote tiles which are not occupied by shortest pin-to-pin paths. Undirected edges with one capacity and one cost are used to connect two adjacent tile nodes. An undirected edge allows flow passing both directions. The second action is to assign *median nodes*, which are different to the tile nodes, to all median points. Directed edges, which are flow out of median nodes, are used to connect median nodes and adjacent tile nodes. A super source s is connected to all median nodes by directed edges with one capacity and zero cost and a super sink t is connected to the grid boundary.

As shown in Fig. 10(a), differential pairs {*1a*, *1b*}, {*2a*, *2b*}, and {*3a*, *3b*} connect to their median nodes by equal length wires. The super source s is connected to all median nodes and the super sink t is connected to the grid boundary. Network-flow approach will start from super source s with flows and end to super sink t with minimum cost. Fig. 10(b) shows the result of network-flow. The median nodes of differential pairs {*1a*, *1b*}, {*2a*, *2b*}, and {*3a*, *3b*} are connected to the grid boundary by directed edges with costs 2, 2, and 1, respectively. After finding the optimal paths from median nodes to the grid boundary, we should route double wires according to the optimal paths. As shown in Fig. 10(c), the routing result of escape routing is finished. We now obtain totally length-matching differential pairs and maintain better performance of differential pairs at the escape routing stage.

IV. EXPERIMENTAL RESULTS

The experiments are performed on a Linux workstation with a 2.4GHz Intel Xeon CPU and 8GB memory. We implement our work in C++ and test it on 10 test cases. Due to the Intellectual Property limitation, we cannot obtain the execution code and benchmarks from [15]. To compare with [15], we re-implement the negotiation congestion-based performance-driven router referring to [15] and [18]. The test case ex10 is rebuilt from the layout figure of [15]. The other 9 test cases are generated according to the information of the number of differential pairs, the size of pin grid, and the distribution of the distances between the two pins of a differential pair from [15]. The benchmark circuits are listed in TABLE II. In this table, "*#diff. pair*" gives the number of differential pair, and "*pin grid (#row×#col)*" gives the size of the pin grid.

Table III gives the comparison of [15] and our method. In this table, "*max.*" in the "*DP skew* (*ps*)" gives the maximum differential-pair skew in pico-second, "*avg.*" in the "*DP skew* (*ps*)" gives the average differential-pair skew in pico-second, "*avg. len* (*tile*)." gives the average number of tiles from two pins to the grid boundary, "*equal len. rate*" gives the rate of length-matching differential pairs, and "*runtime* (s)" gives the running time for routing in second. Maximum and average differential-pair skews are simulated by HSPICE with industrial design specification. As shown in the table, our escape routing of differential pair considering length matching reduces the maximum and average differential-pair skews about 68% and 74%, respectively. With the significant reduction in maximum and average differential-pair skews, only 9% average length increases. Besides, our method can guarantee that all differential pairs are length-matching while [15] can achieve only 57% length-matching result. To demonstrate the efficiency of LGD (Section 3.C), we also compare our algorithm without/with LGD in TABLE III. Our algorithm with LGD can significantly reduce the running time in most cases especially for the largest case, ex10.

Fig. 11 shows the routing results of ex10 obtained by our method. The experiment shows that the effectiveness and efficiency of our escape routing of differential pairs considering length matching with significant reduction in maximum and average differential-pair skews.

V. CONCLUSIONS

In this paper, we proposed an escape routing approach of differential pairs considering length matching. The approach includes two stages. The first stage is to find min-cost median points which connect two pins by minimum equal wire length while the second stage is adopted a network-flow approach with min-cost max-flow to simultaneously route all differential pairs. Experimental results showed that our approach can efficiently and effectively obtain length-matching differential pairs with significant reduction in maximum and average differential-pair skews.

TABLE II
BENCHMARK INFORMATION

test cases	#diff. pair	pin grid (#row×#col)
rcase1	10	11×8
rcase2	18	22×7
rcase3	18	18×12
rcase4	8	11×3
rcase5	11	14×3
rcase6	11	17×6
rcase7	18	17×6
rcase8	20	9×16
rcase9	20	8×15
ex10	60	35×35

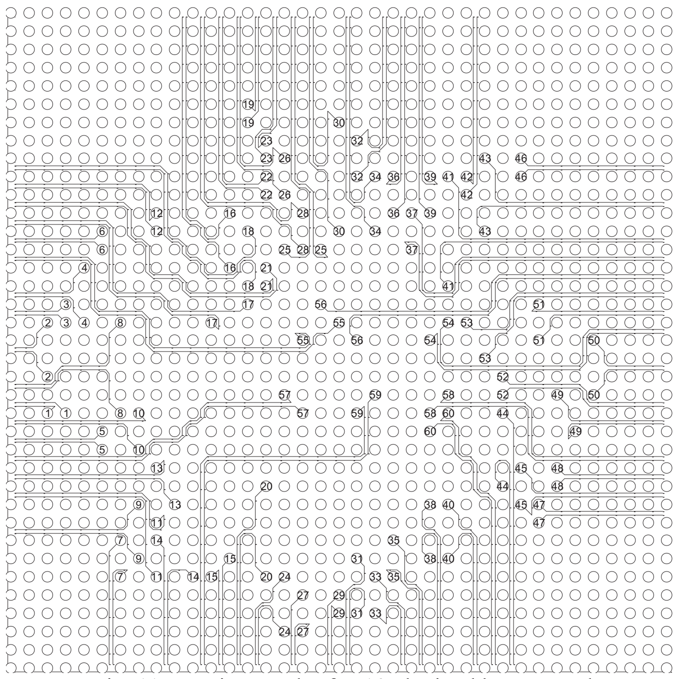


Fig. 11. Routing result of ex10 obtained by our work

REFERENCES

[1] J.-W. Fang, C.-H. Hsu, and Y.-W. Chang, "An integer linear programming based routing algorithm for flip-chip design," in *Proc. Design Automation Conf.*, pp. 606–611, 2007.

[2] Y. Kubo and A. Takahashi, "Global routing by iterative improvements for two-layer ball grid array packages," *IEEE Trans. Computer-Aided Design*, vol. 25, no. 4, Apr. 2006.

[3] Y. Kubo and A. Takahashi, "A global routing method for 2-layer ball grid array packages," in *Proc. Int. Symp. on Physical Design*, pp. 36–43, 2005.

[4] Y. Tomioka and A. Takahashi, "Monotonic parallel and orthogonal routing for single-layer ball grid array packages," in *Proc. Asia and South Pacific Design Automation Conf.*, pp. 642–647, 2006.

[5] L. Luo and M. D. F. Wong, "Ordered escape routing based on Boolean satisfiability," in *Proc. Asia and South Pacific Design Automation Conf.*, pp. 244–249, 2008.

[6] 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.

[7] J.-W. Fang and Y.-W. Chang, "Area-I/O flip-chip routing for chip-package co-design," in *Proc. Int. Conf. on Computer-Aided Design*, pp. 518–522, 2008.

[8] M.-F. Yu and W. W.-M. Dai, "Single-layer fanout routing and routability analysis for ball grid arrays," in *Proc. Int. Conf. on Computer-Aided Design*, pp. 581–586, 1995.

[9] R. Wang, R. Shi, and C.-K. Cheng, "Layer minimization of escape routing in area array packaging," in *Proc. Int. Conf. on Computer-Aided Design*, pp. 815–819, 2006.

[10] D. Wang, P. Zhang, C.-K. Cheng, and A. Sen, "A performance-driven I/O pin routing algorithm," in *Proc. Asia and South Pacific Design Automation Conf.*, pp. 129–132, 1999.

[11] M.-F. Yu, J. Darnauer, and W. W.-M. Dai, "Interchangeable pin routing with application to package layout," in *Proc. Int. Conf. on Computer-Aided Design*, pp. 668–673, 1996.

[12] M.-F. Yu and W. W.-M. Dai, "Pin assignment and routing on a single-layer pin grid array," in *Proc. Asia and South Pacific Design Automation Conf.*, pp. 203–208, 1995.

[13] T. Yan and M. D. F. Wong, "A correct network flow model for escape routing," in *Proc. Design Automation Conf.*, pp. 332–335, 2009.

[14] C. T. Robertson, *Printed Circuit Board*, Prentice Hall, 2004.

[15] T. Yan, P. C. Wu, Q. Ma, and Wong, M.D.F., "On the escape routing of differential pairs," in *Proc. Int. Conf. on Computer-Aided Design*, pp. 614-620, 2010.

[16] J.-W. Fang, K.-H. Ho, and Y.-W. Chang, "Routing for chip-package-board co-design considering differential pairs," in *Proc. Int. Conf. on Computer-Aided Design*, pp. 512–517, 2008.

[17] L. McMurchie and C. Ebeling, "Pathfinder: a negotiation-based performance-driven router for FPGAs," in *Proc. Int. Symp. on Field-Programmable Gate Arrays*. ACM, pp. 111–117, 1995.

[18] Q. Ma; T. Yan; Wong, M.D.F., "A negotiated congestion based router for simultaneous escape routing," *Int. Symp. on Quality Electronic Design*, pp. 606-610, 2010.

TABLE III
EXPERIMENTAL RESULTS

	[15]					Ours									
						without LGD					with LGD				
test	DP skew (ps)		avg. len.	equal	runtime	DP skew (ps)		avg. len.	equal	runtime	DP skew (ps)		avg. len.	equal	runtime
cases	max.	avg.	(tile)	len. rate	(s)	max.	avg.	(tile)	len. rate	(s)	max.	avg.	(tile)	len. rate	(s)
rcase1	9.9	2.6	2.4	60.0%	<1	1.4	0.7	2.7	100%	<1	1.4	0.7	2.7	100%	<1
rcase2	15.3	5.9	3.6	44.4%	<1	3.9	0.7	4.0	100%	2.1	3.9	0.7	4.0	100%	<1
rcase3	20.0	6.5	4.2	27.8%	<1	3.7	0.7	4.6	100%	1.4	3.5	0.7	4.8	100%	<1
rcase4	2.8	0.8	2.0	87.5%	<1	1.4	0.6	2.0	100%	<1	1.4	0.6	2.0	100%	<1
rcase5	8.6	1.8	2.2	63.6%	<1	1.5	0.2	2.6	100%	1.0	1.5	0.2	2.6	100%	<1
rcase6	26.0	7.9	4.1	36.4%	<1	3.2	1.0	4.0	100%	2.1	3.3	1.0	4.1	100%	<1
rcase7	6.5	2.4	2.9	66.7%	<1	5.6	1.0	2.9	100%	4.1	5.8	1.0	3.0	100%	<1
rcase8	8.2	2.6	3.3	70.0%	<1	2.4	0.6	3.3	100%	6.3	2.6	0.6	3.5	100%	<1
rcase9	12.0	2.8	3.0	75.0%	<1	5.2	1.0	3.3	100%	4.7	5.2	0.9	3.3	100%	<1
ex10	80.8	16.4	10.2	36.7%	<1	16.2	3.2	11.9	100%	1237	16.4	3.3	12.2	100%	<1
comp.	1	1	0.91	0.57		0.31	0.27	0.97	1		0.32	0.26	1	1	