Ordered Escape Routing for Grid Pin Array Based on Min-cost Multi-commodity Flow

Fengxian Jiao
Department of Computer Science & Technology
Tsinghua University
jfx14@mails.tsinghua.edu.cn

Sheqin Dong *
Department of Computer Science & Technology
Tsinghua University
dongsq@tsinghua.edu.cn

Abstract— Ordered Escape routing is a critical issue in high-speed PCB routing. In this paper, for the first time, a Min-cost Multi-commodity Flow (MMCF) approach is proposed to solve the ordered escape routing. The characteristic of grid pin array is analyzed and then a basic network model is used to convert ordered escape routing to MMCF model. To satisfy the constraints of ordered escape routing, three novel transformations, such as non-crossing transformation, ordering transformation and capacity transformation, are used to convert the basic network model to the final correct MMCF model. Experimental results show that our method achieves 100% routability for all the test cases. The method can get both a feasible solution and an optimal solution. Compared to published approaches, our method improves in both wire length and CPU time remarkably.

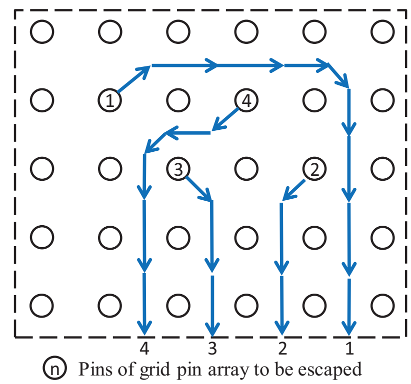

Fig. 1. An example of the ordered escape routing on grid pin array.

I. Introduction

Escape routing, which is one of the most important issues in PCB design, is classified into *unordered escape routing* and *ordered escape routing* [1]. For unordered escape routing, network-flow formulations are pervasively used to solve the problem [2] [3] [4]. However, network-flow cannot deal with the ordered escape routing because it does not carry the ordering information. In [5], Luo and Wong convert the ordered escape routing into Boolean Satisfiability (SAT) problem and solve it by SAT solver. Without any assumption on the routing style, the approach can exactly complete the routing. However, the approach can hardly deal with capacity problem of escape routing. Besides that, the SAT-based approaches take much CPU time and the time complexity grows very fast when the grid size increases. In a later work [6], Luo and Wong extended their router to handle cyclic ordering and pin clustering. A recent research [7] solves the problem based on the optimality of hierarchical bubble sorting. Although their approach can achieve high routability in reasonable CPU time for some cases, it reduces the size of solution space and will lose many solutions of the problem.

Furthermore, none of these researches consider minimizing the total wire length for manufacturing cost reduction.

The pin array of PCB design is usually divided into grid pin array (GPA) and staggered pin array (SPA). In this paper, we focus on the ordered escape routing problem for GPA. Fig.1 shows an example of the problem.

We summarize our contributions as follows:

- For the first time, we apply min-cost multi-commodity flow to ordered escape routing and propose an MMCF model to solve the problem.
- We are the first to optimize the total wire length of ordered escape routing, while the published approaches only consider the routability.
- Three novel transformations, such as non-crossing transformation, ordering transformation and capacity transformation, are proposed to meet the constraints of ordered escape routing.

The rest of this paper is organized as follows: Section II defines the problem of orderd escape routing and the formulation. Section III presents our approach to convert the ordered escape routing to MMCF at first, and then discuss the transformations we proposed to meet the constraints. Experimental results are shown in Section IV and conclusions are provided in Section V.

II. Problem Definition

Before we present the problem, the definitions for routing network are introduced first.

Definition 1 *(Grid Pin Array): An $m \times n$ grid pin array (GPA) is composed of m rows and n columns. The $m \times n$ pins "arrayed" in a series of square "grids". In other word, each row*

*This paper is supported by NSFC 61176022.

978-1-4673-9569-4/16/$31.00 ©2016 IEEE

and each column of pins are aligned. A square tile is composed of four adjacent pins and we assume that there is a tile node at the center of each tile, as shown in Fig.2 (a).

Definition 2 *(Tile Node Network): The adjacent tile nodes can be connected to each other to form the tile node network, as shown in Fig.2(b). The edges of tile node network are the channels for escape routing. Fig.2(b) illustrates an example of routing via tile node network.*

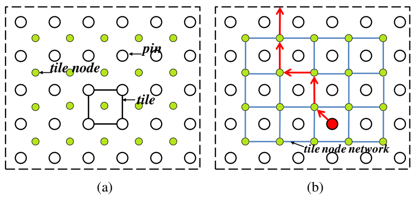

Fig. 2. The grid pin array and a routing example via tile node network. (a) grid pin array (b) the routing example.

The ordered escape routing problem can be defined as follows:

Given an $m \times n$ grid pin array with a set of marked pins $P = \{P_1, P_2, ..., P_r\}$. Route the pins to the boundary without crossing and exceeding capacity of each signal path. Besides, the escaped wires around the boundary are required to follow some ordering constraints. At the same time, the total wire length needs to be minimized under 100% routability guarantee.

If the destinations of all the escaped nets are required to be escaped on a single side of the pin array, the ordered escape routing problem can be defined as a 1-sided ordered escape routing problem. Similarly, we can define the 2-sided, 3-sided and 4-sided ordered escape routing problem.

III. Min-cost Multi-commodity Flow Model

In this section, we propose Min-cost Multi-commodity Flow Model (MMCF model) to formulate the ordered escape routing, and then present three transformations to meet the constraints.

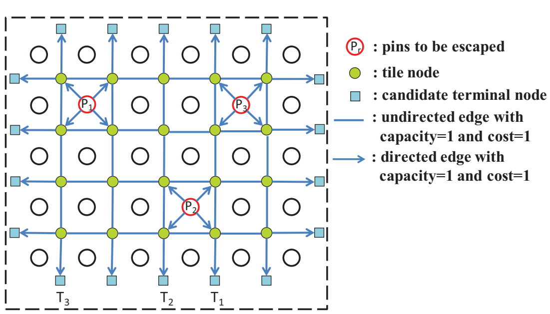

Fig. 3. The basic MMCF model for grid pin array.

The basic MMCF model is illustrated in Fig.3. We place additional cells called *candidate terminal nodes* at the boundary. Each pin to be routed is a source node in the model, which will ship 1 unit of commodity to the boundary. Note that these pins ship different types of commodities. Choosing the same number of terminal nodes as the destination of the transportation from candidate terminal nodes, with satisfying the ordering constraint clockwise (in this example, we choose T_1, T_2, T_3). The capacity and cost of each edge is 1. Finally, we get a MMCF problem and the total wire length equals to the total cost of the problem.

However, there are some errors in the model. Firstly, the route paths are used to transmit signals on PCB, but the nets of two signals may be crossed at a tile node. Secondly, there are too many combinations of selecting the destinations from terminal nodes. Thirdly, there are both undirected edges and directed edges in the network which must be unified before using the MMCF solver to solve, but it's a little difficult to satisfy the capacity constraint when we convert the undirected edge to directed edge. The details of the transformations to avoid the errors will be introduced in the following sections.

A. Modeling the Non-crossing Constraint

As shown in Fig.4 (a), the nets of two signals may be crossed at a tile node. In unordered escape routing, the problem can be solved easily by swapping the paths after the crossed node, as illustrated in Fig.4(b) and Fig.4(c). However, it's much more difficult in ordered escape routing to avoid the crossing since swapping the paths is not allowed, which doesn't meet the ordering constraint.

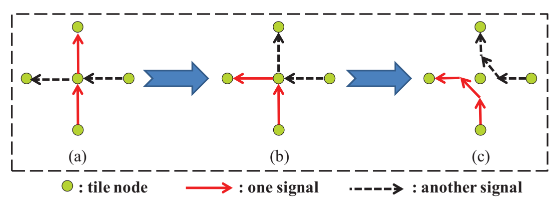

Fig. 4. The method to avoid crossing in unordered escape routing.

First, we enumerate all possible crossing cases in grid pin array (shown in Fig.5) and define two types of crossing as follows:

Definition 3 *(TYPE 1 CROSSING): A TYPE 1 CROSSING is the crossing between signals from two tile nodes, as shown in Fig.5 (a).*

Definition 4 *(TYPE 2 CROSSING): A TYPE 2 CROSSING is the crossing between signals from two pins (Fig.5(b)) or the crossing between signals from one pin and one tile node (Fig.5(c))*

A.1 Modeling TYPE 1 CROSSING

To avoid TYPE 1 CROSSING, we must transfer the basic tile model to a new tile model as follows:

Definition 5 *(NON-CROSSING TRANSFORMATION TYPE 1): For each tile, create four imaginary tile nodes, which are*

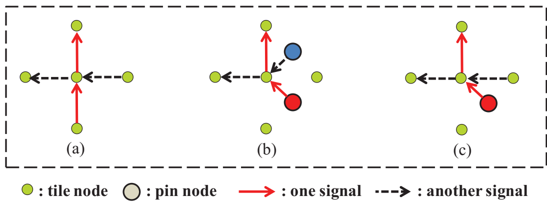

Fig. 5. Three kinds of crossing in grid pin array.

N-tilenode, E-tilenode, S-tilenode, and W-tilenode, respectively (see Fig.6(b)). Undirected edges are created between the node pairs (N, E), (E, S), (S, W), (W, N) with capacity 1 and cost 0, respectively. Call such edges in-tile edges. The N-tilenode connects to the S-tilenode of the north neighboring tile with undirected edge of which the capacity and cost is 1. Call such edges inter-tile edges. Other three imaginary tile nodes connect to the neighboring nodes with inter-tile edges too.

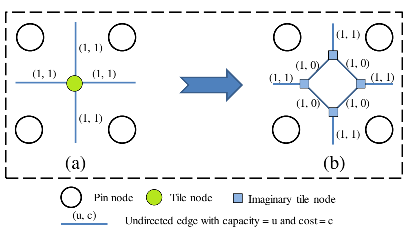

Fig. 6. Transformation of avoiding TYPE 1 CROSSING (a) the basic tile model (b) the transferred tile model.

Theorem 1 *Using NON-CROSSING TRANSFORMATION TYPE 1, there exists no TYPE 1 CROSSING in the MMCF model.*

Proof. To prove Theorem 1, we have to demonstrate that our network model could avoid TYPE 1 CROSSING and wouldn't influence the legal routing cases. We enumerate all possible cases in Fig.7, in which case 1 to case 3 is legal and case 4 is illegal.

It is clear that for the legal cases, the signals can pass the tile node correctly and for the illegal case, the signal cannot pass as a result of the capacity constraint. The details of the proof is omitted due to the limited space.

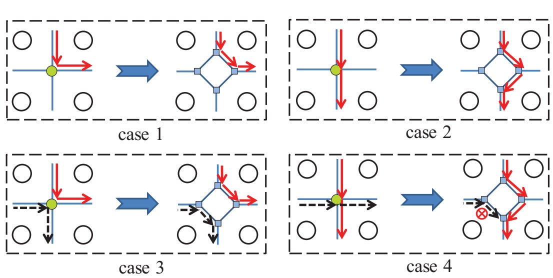

Fig. 7. All possible cases of signals passing a tile node, where case 1, case 2, case 3 are legal and case 4 is illegal.

A.2 Modeling TYPE 2 CROSSING

TYPE 2 CROSSING is caused by the pin nodes. The transformation to avoid them is as follows:

Definition 6 *(NON-CROSSING TRANSFORMATION TYPE 2): For each pin to be escaped, create four imaginary pin nodes in the center of four edges around the pin, which are N-pinnode, E-pinnode, S-pinnode, and W-pinnode, respectively (see Fig.8 (b)). The imaginary pin nodes divide the edges into 8 edges, which are called around-box edges. Directed edges are created between the pin and imaginary pin nodes with capacity 1, cost 1. Call such edges in-box edges.*

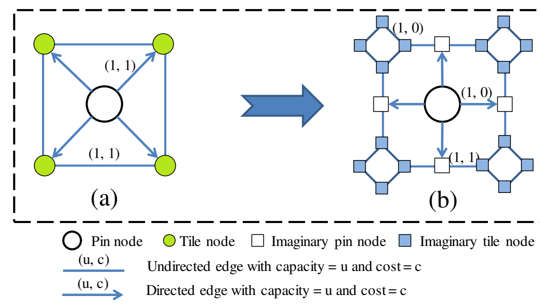

Fig. 8. Transformation of avoiding TYPE 2 CROSSING on grid pin array.

Theorem 2 *Using NON-CROSSING TRANSFORMATION TYPE 2, there exists no TYPE 2 CROSSING in the MMCF model.*

Proof. The direct connections between pin node and tile node are replaced by the connections passing by imaginary pin nodes. Thus, we transfer TYPE 2 CROSSING to TYPE 1 CROSSING.

Similarly, by enumerating all the possible cases we find that for the legal cases, our model can guarantee the correctness and for all the illegal cases, our model can convert them to TYPE 1 CROSSING.

In conclusion, our network model can avoid the TYPE 2 CROSSING correctly.

B. Modeling the Ordering Constraint

In ordered escape routing, the destinations of escaped pins must at the boundary of the pin array. In other word, we needn't determine which destinations they are at the very beginning. However, in MMCF problem, we must choose the destination of every commodity at first.

The method to decide the destinations of each commodity is defined as follows:

Definition 7 *(ORDERING TRANSFORMATION): Assume that there are r pins and $k (k \geq r)$ candidate terminals (see Fig.9 row 1). Create $k - r$ rows of imaginary terminal nodes and the number of nodes in the next row is one less than the previous. In other words, the number of nodes in each* *row forms a tolerance of 1 arithmetic progression. Note that the number of nodes in the last row is r, equaling to the number of pins to be routed. Let $T_{i,j}$ denotes the jth node in row i. Directed edges are created between the node pairs $(T_{i,j}, T_{i+1,j})$ and $(T_{i,j+1}, T_{i+1,j})$ for all the legal i,j , as shown in Fig.9. The capacity of each edge is 1 and the cost is 0. Then, the last row of imaginary nodes is the terminals of pins to be escaped. Call these nodes destination nodes. In other word, the destination nodes are the destinations of our MMCF model.*

To guarantee the ordering constraint, at most 1 signal could pass through each imaginary terminal node in our network model. For this, we could replace the imaginary terminal node (except the last row) with an imaginary edge, of which the capacity is 1 and cost is 0, as shown in Fig.10.

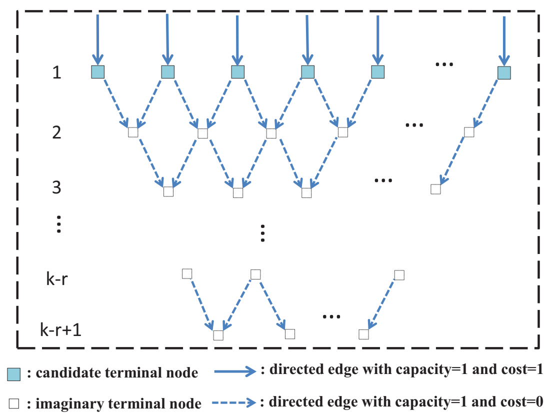

Fig. 9. Transformation to construct the terminals of pins to be escaped.

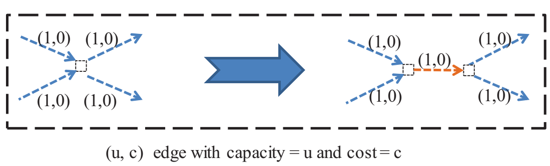

Fig. 10. The method to guarantee there is at most 1 signal passing through each imaginary terminal node.

Let us call the escape routing is legal if all the pins to be escaped are routed to the boundary with correct ordering. A flow of our MMCF is legal if all the commodities from source nodes are transported to the correct destination nodes. Theorem 3 guarantees the equivalence of the two legalities.

Theorem 3 *Given a pin array with r to-be-escaped pins, there exists a legal routing of r pins if and only if its corresponding MMCF model has a legal flow with r commodities.*

In order to prove theorem 3, we must show the MMCF model can correctly capture all the legal cases of escape routing and avoid all the illegal cases. Therefore, we must prove the following two lemmas, Lemma 1 and Lemma 2, are both correct.

Lemma 1 *If a legal routing of r pins exists, there must exists a legal flow of MMCF model with r commodities.*

Proof. Lemma 1 means for all the legal combinations of terminals at row 1 in Fig.9, the signals can be transported to the destinations (row $k - r + 1$ in Fig.9) in correct order. Thus, we only need to prove that the signals in row $i(1 \leq i \leq k - r)$ can be routed to row $i + 1$ legally.

The number of nodes in row i is $k - i + 1$ and there are $k - i$ signals at most, which means $r \leq k - i$.

If $r = k - i$, there is only 1 empty node in row i. For the nodes at the right of the empty node, choose the edges from top right to lower left and for the nodes at the left of the empty node, choose the edges from top left to lower right. Through these edges, we can route the signal to the row $i + 1$ in correct order, as shown in Fig.11.

If $r < k - i$, there is more than 1 empty node in row i . Choose 1 empty node randomly and the correctness can be proved with the similar method.

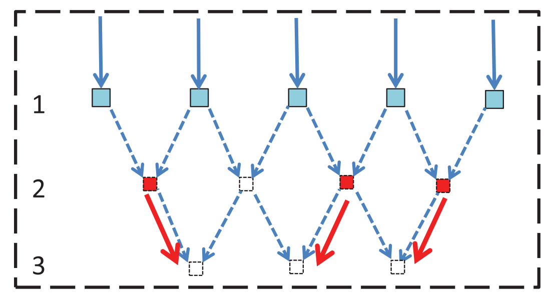

Fig. 11. An example of routing the signals to the next row with correct order.

Lemma 2 *If a routing of r pins is illegal, there must have no legal flow of MMCF model with r commodities.*

Proof. Lemma 2 means for the illegal escape routing cases, there exists no solution for our MMCF model.

To prove Lemma 2, we could show the signals which is in error order at row 1 in Fig.9, will be in the same error order at the last row. In other word, if the order of signal terminals is illegal, the order of destinations of our MMCF problem is illegal too.

Without loss of generality, let us assume two signals are in error order at row i , one passing T_{i,j_1} and another passing T_{i,j_2} , where $j_1 < j_2$. We need only prove that if the two signals can arrive at row $i + 1$, their order remains the same.

As mentioned above, T_{i,j_1} connects to T_{i+1,j_1-1} and T_{i+1,j_1} , T_{i,j_2} connects to T_{i+1,j_2-1} and T_{i+1,j_2} . Since $j_1 < j_2$, we can get that $j_1 - 1 < j_1 \leq j_2 - 1 < j_2$, which means the order of two signals has no change if they can be routed to row $i + 1$.

In conclusion, theorem 3 is correct.

C. Modeling the Capacity Constraint

As mentioned above, in this paper, we only consider the capacity of each edge is 1. Therefore, we can make sure that the flow solution is corresponding to a valid routing solution only if the multi-commodity flow is an integral flow. However, there are both directed edges and undirected edges in the MMCF model. The undirected edges with capacity 1 means only one unit of commodity can be transformed on this edge, with either of forward and backward directions. Before resolving the network model with MMCF solver, we have to convert the undirected edges to directed edges.

In unordered escape routing, a forward edge and a backward edge are created to replace an undirected edge with the same capacity u, as shown in Fig.12(a). If one signal occupy the forward edge and another signal occupy the backward edge, the capacity of the original undirected edge is exceeded. Fortunately, we can remove both of the signals on the forward edge and backward edge and swap the paths to solve the problem easily, as is shown in Fig.13(a). However, in ordered escape routing, the method is illegal as a result of fail to satisfy the ordering constraint. The transformation to solve the problem is as follows[8]:

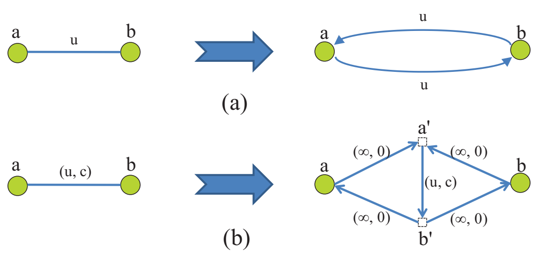

Fig. 12. The method to transfer undirected edges to directed edges in (a) unordered escape routing (b) ordered escape routing.

Definition 8 *(CAPACITY TRANSFORMATION): For each undirected edge, create two imaginary nodes. Then, one directed edge with capacity u and cost c is created to connect them. After that, create four directed edge with capacity ∞ and cost 0 to connect the adjacent nodes, as illustrated in Fig.12 (b).*

Theorem 4 *Using the CAPACITY TRANSFORMATION, the MMCF model can replace an undirected edge by a set of directed edges with satisfying the same capacity constraint.*

Proof. Every unit of flow between nodes a and b in the undirected network must flow on the paths $a - a^{'} - b^{'} - b$ or $b - a^{'} - b^{'} - a$ in the directed figure. The arc $(a^{'}, b^{'})$ implies that this flow cannot exceed u; moreover, each unit incurs a cost of c. Thus, theorem 4 is correct.

In our work, the capacity u=1 and the cost c=1.

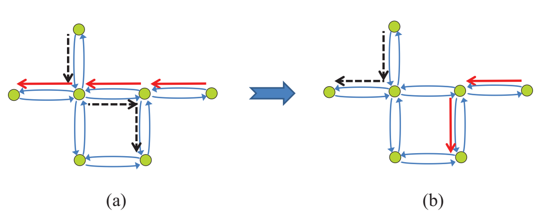

Fig. 13. An example of how to satisfy the capacity in unordered escape routing.

Through the transformations introduced in section V.A, V.B, V.C, we convert the basic MMCF model to the final MMCF model, as shown in Fig.14, with capturing the constraints in ordered escape routing correctly.

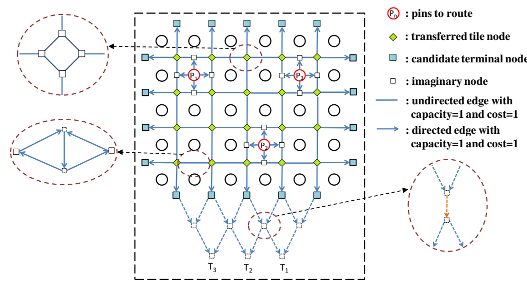

Fig. 14. The final MMCF model in grid pin array and partial enlargement view.

IV. Experimental Results

We implement our ordered escape routing algorithm in C++ and test it on a workstation with Intel Xeon 2.4GHz CPU and 12GB physical memory. Due to the Intellectual Property limitation, we cannot obtain the execution code and benchmarks of published research. We re-implement their methods for the comparison of the final routing result. Gurobi optimizer solver [9] is used as our MMCF solver. The capacity of each edge in our experiments is 1 and the wire-lengths of results are all normalized values.

Eight cases are tested for all the three methods and the characteristics of them are shown in Table 1. These cases cover 1-side, 2-side, 3-side, and also 4-side escape situations.

In Table 1, "#Col" and "#Row" denote the number of columns and rows in a given pin array, respectively. "#Pin" is the pin terminals which need to be escaped and "#Type" is the number of sides on the pin array that can be escaped through. Wire-length and CPU time of all the experimental results are measured and shown in the table. As for our algorithm, we measure the wire-length and CPU time for both feasible solution and optimal solution. "Length1" and "Time1" belong to feasible solution; "Length2" and "Time2" belong to optimal solution. Besides, the symbol "-" means cannot get a solution.

Table 1 shows that our algorithm is very effective on ordered escape routing, compared to published methods. For small-scale cases, the solver can get an optimal solution with minimal wire-length and slight long but reasonable CPU time. On the other hand, it can also obtain a feasible solution with extremely short CPU time and little long wire-length. Fig.15 is the final optimal routing result of case4. For large-scale cases, it is too hard for the solver to get an optimal solution, but it can still get an feasible solution with reasonable CPU time. Fig.16 is the final feasible routing result of case8. For case 6, our method get a feasible solution with wire-length equal to 37 in 5s, however, the solver cannot guarantee the solution is optimal. After 20s, the solver proves that the solution is optimal.

For case7 and case8, the approaches mentioned above cannot get a feasible solution.

TABLE I
EXPERIMENTAL RESULTS FOR ORDERED ESCAPE ROUTING

Benchmark	#Col	#Row	#Pin	#Type	SAT-based[5]		Routablility-Driven[7]		Our Approach			
					Length	Time(s)	Length	Time(s)	Length1	Time1(s)	Length2	Time2(s)
case1	8	6	6	1-side	20	0.21	25	0.05	17	0.05	17	0.05
case2	8	6	6	2-side	28	0.26	35	0.10	20	0.09	14	0.16
case3	8	6	11	2-side	104	2.21	121	1.21	84	1.22	72	8.41
case4	10	6	25	3-side	77	1.13	92	0.66	65	0.58	61	1.15
case5	9	6	11	2-side	45	0.32	59	0.22	37	0.24	36	0.36
case6	8	8	8	1-side	46	10.22	-	-	37	5.02	37	20.26
case7	20	15	45	2-side	-	-	-	-	352	10.13	-	-
case8	30	30	70	4-side	-	-	-	-	530	30.15	-	-

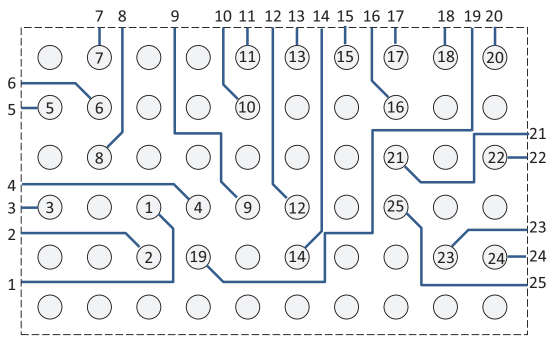

Fig. 15. The final routing result of case4

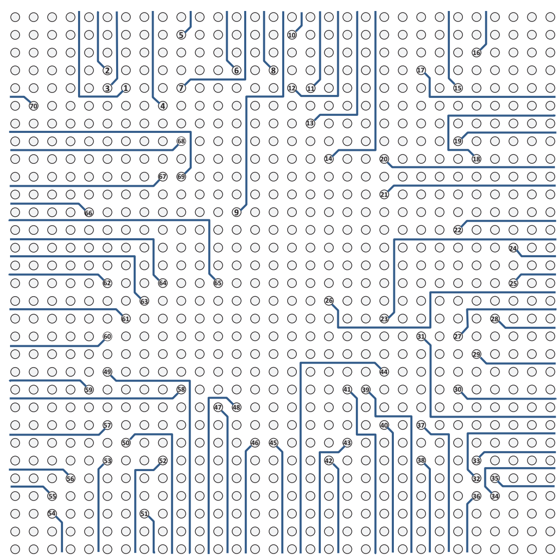

Fig. 16. The final routing result of case8

V. CONCLUSIONS

In this paper, for the first time, we propose an algorithm based on Min-cost Multi-commodity Flow for ordered escape routing. Furthermore, three transformations are proposed to satisfy the constraints. The experimental results show that our method is more efficient in CPU time and optimization of wire length compared with the published approaches.

Our future work would focus on how to improve the efficiency for large-scale application and increase the capacity of edges.

ACKNOWLEDGMENTS

This paper was first submit to DAC 2015 and then ICCAD 2015, we would like to thank all the reviewers of this paper and their helpful comments.

REFERENCES

[1] Tan Yan and Martin D. F. Wong. Recent research development in pcb layout. In *Computer-Aided Design (ICCAD), 2011 IEEE/ACM International Conference on*, pages 398 – 403. IEEE, 2010.

[2] Jia-Wei Fang, I-Jye Lin, Yao-Wen Chang, and Jyh-Herng Wang. A network-flow-based rdl routing algorithmz for flip-chip design. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 26(8):1417–1429, 2007.

[3] Qiang Ma, Evangeline FY Young, and Martin DF Wong. An optimal algorithm for layer assignment of bus escape routing on pcbs. In *Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE*, pages 176–181. IEEE, 2011.

[4] Kan Wang, Huaxi Wang, and Sheqin Dong. Escape routing of mixed-pattern signals based on staggered-pin-array pcbs. In *Proceedings of the 2013 ACM international symposium on International symposium on physical design*, pages 93–100. ACM, 2013.

[5] Lijuan Luo and Martin DF Wong. Ordered escape routing based on boolean satisfiability. In *Proceedings of the 2008 Asia and South Pacific Design Automation Conference*, pages 244–249. IEEE Computer Society Press, 2008.

[6] Lijuan Luo and Martin D.F. Wong. On using sat to ordered escape problems. In *Proceedings of the 2008 Asia and South Pacific Design Automation Conference*, pages 594 – 599. IEEE Computer Society Press, 2009.

[7] Jin-Tai Yan, Chung-Wei Ke, and Zhi-Wei Chen. Ordered escape routing via routability-driven pin assignment. In *Proceedings of the 20th symposium on Great lakes symposium on VLSI*, pages 417–422. ACM, 2010.

[8] Ravindra K Ahuja, Thomas L Magnanti, and James B Orlin. Network flows: theory, algorithms, and applications. 1993.

[9] Gurobi optimizer. `http://www.gurobi.com`.