MC-MCF: A Multi-Capacity Model for Ordered Escape Routing

Zhenyi Gao, Sheqin Dong, Zhicong Tang and Wenjian Yu
Department of Computer Science & Technology, Tsinghua University, Beijing
E-mail: gzy22@mails.tsinghua.edu.cn, dongsq@tsinghua.edu.cn,
tzc21@mails.tsinghua.edu.cn, yu-wj@tsinghua.edu.cn

Abstract

Ordered escape routing (OER) is an important research topic in PCB design, which means the wires need to be routed in a given order at the boundary of pin array. Although OER has been widely investigated, most works assume the routing capacity between two adjacent pins is just 1. In this paper, we focus on multi-capacity OER (MC-OER), which means multiple wires are allowed to pass through between two adjacent pins. We first analyze the limitation of existing model based on min-cost multi-commodity flow graph, i.e. MMCF model, and point out the reason why it cannot support multi-capacity. Based on the MMCF model, a multi-capacity multi-commodity flow (MC-MCF) model is proposed for the MC-OER problem. To accelerate the solution based on MC-MCF model, a wiring resource driven partition strategy is further proposed which results in an accelerated MC-MCF algorithm for the MC-OER problem with objective of minimizing wiring length. Experiments on various grid pin array cases (with up to 308 pins) show that the proposed method achieves 100% routability within reasonable time. And, it performs similarly well or better than existing methods when solving single-capacity OER problem.

Keywords

Ordered escape routing, multi-capacity, multi-commodity flow

1. Introduction

Escape routing, which can be classified into unordered escape routing (UER) and ordered escape routing (OER) [1], is an important research topic in PCB design. To accommodate the fast growing number of pins in modern PCB designs, efficient and automatic routing methods are required. For the UER problem, rule-based approach [2], [3] and network flow-based approach [4]–[6] are commonly used. For the OER problem, Tomioka et al. proposed a method that can solve it while satisfying monotonicity [7]. Fang et al. proposed a method that can solve the cyclic OER problem based on the integer linear programming (ILP) [8]. In 2008, Luo et al. converted the OER problem into boolean satisfiability (SAT) problem and solved it using a SAT solver [9]. Later on, a simple partitioning method was proposed to improve the efficiency of the SAT method by exploiting the characteristics of the OER in industry [10]. In 2010, Yan et al. proposed a more efficient method based on hierarchical bubble sorting [11]. However, it reduces the size of solution space and loses many solutions of the problem. It should be pointed out that, none of these works can solve the OER with optimization objective.

In 2015, Satter and Naveed proposed a network flow method that works to maximize the number of escape nets [12]. Then, Jiao and Dong proposed the min-cost multi-commodity flow (MMCF) model [13], which is the first model that minimizes the total wiring length for OER problem. In a recent study [14], Liao and Dong proposed a compact MMCF model with a partitioning strategy to improve the efficiency and achieved more than 100X speedup over the method of [13]. With the development of machine learning, neural network is gradually applied to the field of PCB designs. Chen et al. proposed a machine learning based method [15], which reduces the solution space of ordered escape problem and achieves an acceleration of 4~370X over the method of [9], [13]. However, these works assume the routing capacity between two adjacent pins is just 1 and none of them considers the multi-capacity ordered escape routing (MC-OER) problem with optimization objective.

Usually, the pin array of PCB design is classified into grid pin array (GPA) and staggered pin array (SPA) [16]. In this work, we focus on the MC-OER problem for GPA (see Fig. 1(a) as an example) and consider minimizing the total wiring length as the objective. Our contributions are as follows.

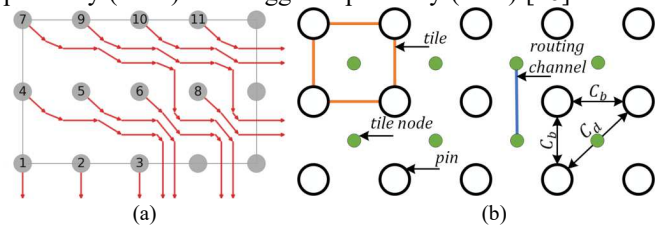


Figure 1: (a) MC-OER on GPA. (b) Some concepts for MC-OER.

1) We propose a model supporting MC-OER, i.e. multi-capacity multi-commodity flow (MC-MCF) model, which is based on stackable reconstructions of three main structures in the MMCF model and apply it to the MC-OER problem with objective of minimizing the wiring length.

2) A wiring resource driven partition strategy (WRDPS) is proposed to reduce the problem size, which considers candidate terminal nodes in MC-MCF graph as a basis for partition. Based on it, an accelerated MC-MCF algorithm is presented to speed up model solving where fine-grained partition strategy is applied. These approaches largely accelerate the MC-MCF model based method with less sacrifice on wiring length.

3) Experimental results show that the proposed method achieves 100% routability in various test cases and can solve large-scale MC-OER problem (>300 pins) in reasonable time (<38 minutes). The accelerated MC-MCF algorithm achieves an average speedup of over 231X. Compared to previous works [13], [14], our method costs similar or less time for single-capacity OER problem.

2. Background

2.1. Problem formulation

An $m \times n$ grid pin array (GPA) consists of m rows and n columns. Each row and each column of pins are aligned. Every four adjacent pins form a square, called *tile*, and we assume that there is a *tile node* at the center of each tile, as shown in Fig. 1(b). We call the edge between adjacent tile nodes *routing channel*.

There are two kinds of routing capacities in GPA: C_b refers to the number of wires that can pass through the routing channel and C_d refers to the number of wires that can pass through the tile node in diagonal direction. In this work, we assume that $C_d = 2C_b$. The ordered escape routing (OER) problem is: Given an $m \times n$ GPA with a sequence of pins $P = \{p_1, p_2, \dots, p_n\}$, route all the pins to the boundary of GPA without wire crossing (*non-crossing constraint*) and violating the two routing capacities (*capacity constraint*), while satisfying the given order at the boundary of GPA (*ordering constraint*). When C_b is greater than 1, we call the problem a *multi-capacity ordered escape routing* (MC-OER) problem. For the example in Fig. 1(a), C_b=2, which means the routing capacity for passing through the routing channel is 2.

If all pins are required to be escaped on a single side of GPA, the problem is called a 1-side OER problem. Similarly, we can define the 2-side, 3-side and 4-side OER problems. The optimization objective can be minimizing the wiring length, minimizing the number of wiring layers, or maximizing the number of escape pins. In this work, we focus on minimizing the wiring length.

2.2. MMCF model for OER

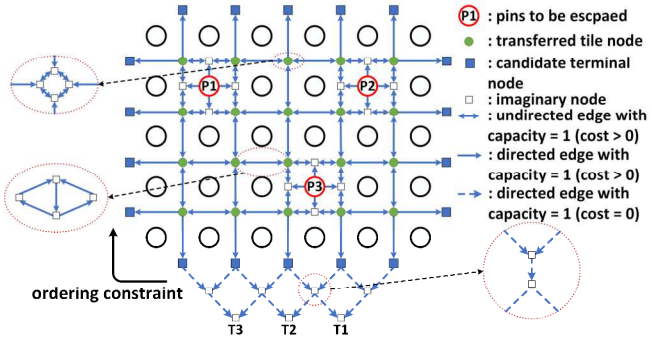


Figure 2: Illustration of the MMCF model for OER with C_b=1, C_d=2 [13].

The min-cost multi-commodity flow (MMCF) graph [13] is illustrated in Fig. 2. Each pin to be routed is a source node in the model, which ships 1 unit of commodities to the boundary. The model encodes the non-crossing constraint, capacity constraint and ordering constraint within a commodity flow graph, so that the solution space of the min-cost multi-commodity flow problem equals to that of OER problem. As shown in Fig. 2, the non-crossing constraint is satisfied by special transformations of tile nodes and undirected edges, while the capacity constraint is naturally satisfied. The ordering constraint can be satisfied by setting the destination node of commodity in a given order. For the example in Fig. 2, we set T1, T2 and T3 as the destination nodes of pins P1, P2 and P3, respectively. Since each node in GPA can find the corresponding node in the commodity flow graph (for example, tile node in GPA ↔ transferred tile node in MMCF graph), it is easy to find out the correspondence between the OER solution and the MMCF graph solution, as shown in Fig. 3. Therefore, the OER with objective of minimizing wiring length is equivalent to solving the MMCF

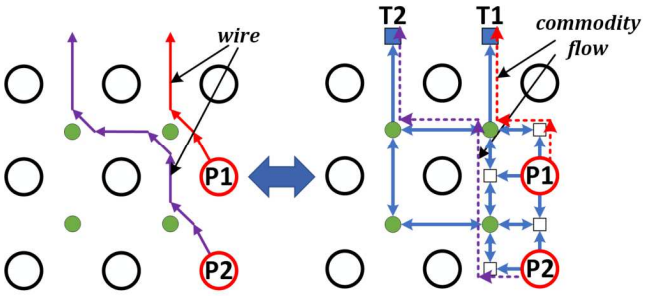


Figure 3: Correspondence between OER solution and MMCF solution.

problem.

When capacity $C_b \geq 2$, this MMCF model does not guarantee that the non-crossing constraint is satisfied. In the MMCF model, each undirected edge is converted to directed edges by capacity transformation (illustrated by Fig. 4). When C_b=1, there is a mutually exclusive relationship between the two commodity flows passing through between two adjacent tile nodes. Under multi-capacity condition, the simultaneous existence of the two commodity flows is permitted, which results in wire crossing as shown in Fig. 4. This reveals that non-crossing constraint in MMCF model is only satisfied for

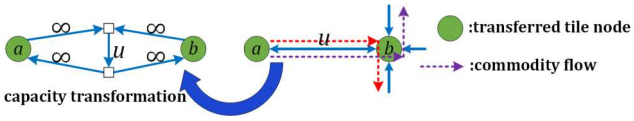


Figure 4: The capacity transformation eliminating undirected edges and the wire crossing issue of MMCF for C_b>1. The edge capacities are labelled (u>1).

3. MC-MCF model for MC-OER

In this section, we propose the multi-capacity multi-commodity flow (MC-MCF) model for solving the MC-OER problem. To overcome the drawback of the MMCF model [13], we first present a non-crossing determination theorem. Based on it, the method of constructing a non-crossing multi-capacity commodity flow graph is then presented. Finally, we discuss the computational cost of the proposed method.

3.1. Non-crossing determination theorem

For a commodity flow graph G, we use $F = \{f_1, f_2, \dots\}$ to denote the set of all commodity flow graph solutions. For any $f \in F$, there is a corresponding solution r of the OER problem (illustrated by Fig. 3). $R = \{r_1, r_2, \dots\}$ denotes the set of all corresponding solutions. $N = \{n_1, n_2, \dots\}$ and $E = \{e_1, e_2, \dots\}$ denote the set of all its nodes and the set of all edges in G before the capacity transformation shown in Fig. 4 respectively. If there is no crossing wire in any $r \in R$, the commodity flow graph G is called *non-crossing*

commodityflow graph. Before stating the theorem, we first give two lemmas for the node non-crossing and edge non-crossing conditions.

Lemma 1 (Node non-crossing). For any node $n \in N$ in a commodity flow graph G, if its degree is not larger than 3, then for any corresponding OER solution $r \in R$ there is no crossing wire at n.

Proof. For any $n \in N$ and $r \in R$, when the degree of n is 0, no commodity flow passes through n, then no wires pass through n in r, so there is no crossing wire.

When the degree of n is 1, at most one commodity passes though n. In this case, one wire passes through n in r at most, so there is no crossing wire.

When the degree of n is 2, n_1, n_2 denote the adjacent nodes of n. In the case where n is source (or destination) and we denote the commodity flow out (or in) as $n \rightarrow n_1$ (or $n_1 \rightarrow n$), another commodity flow through edge $n \leftrightarrow n_2$ flows in but cannot flow out. Therefore, no commodity flow passes through edge $n \leftrightarrow n_2$. So, only one wire passes through n in r. In the case where n is not source or destination, the commodity flow passes through n is $n_1 \rightarrow n \rightarrow n_2$ or $n_2 \rightarrow n \rightarrow n_1$, which means only one wire passes through n in r. So, there is no crossing wire when the degree of n is 2.

When the degree of n is 3, n_1, n_2, n_3 denote the adjacent nodes of n. In the case where n is source (or destination) and we denote the commodity flow out (or in) as $n \rightarrow n_1$ (or $n_1 \rightarrow n$), the route $n_2 \leftrightarrow n \leftrightarrow n_3$ can be simplified to $n_2 \leftrightarrow n_3$. Therefore, no crossing wire is generated whether or not there is commodity flow passing through $n_2 \leftrightarrow n_3$. In the case where n is not source or destination, if there is a legal commodity flow pass throw n, it will occupy 2 edges (take $n \leftrightarrow n_1$, $n \leftrightarrow n_2$ as an example), another commodity flow through edge $n \leftrightarrow n_3$ flows in but cannot flow out. So, there is only one wire pass through n in r. Therefore, there is no crossing wire. □

Lemma 2 (Edge non-crossing). For any edge $e \in E$ in a commodity flow graph G, if its capacity is 1, then for any corresponding OER solution $r \in R$, there is no crossing wire at e.

Proof. For any $e \in E$ and $r \in R$, since the capacity of e is 1, at most one commodity flow passes through e, which means at most one wire passes through e in r. Therefore, there is no crossing wire at e. □

Theorem 1. For a commodity flow graph G, if the degree of any $n \in N$ is less than or equal to 3 and the capacity of any $e \in E$ is 1, the commodity flow graph is a non-crossing commodity flow graph.

Proof. Combining the statements of Lemma 1 and 2, and according to the definition of non-crossing commodity flow graph, we can derive this theorem. □

3.2. Non-crossing commodity flow graph for MC-OER

In this subsection, based on Theorem 1, we will extend and generalize for the three main structures (*routing channel, candidate terminal node and tile node*) in MMCF model [13] to construct non-crossing commodity flow graph supporting multi-capacity.

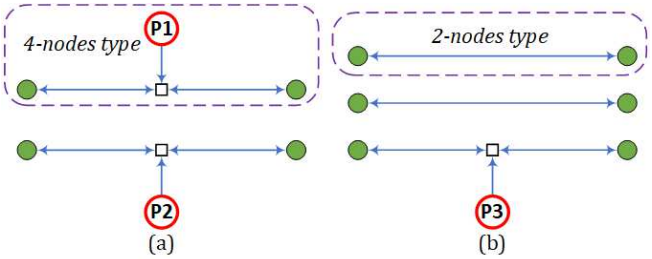


Figure 5: Routing channel for C_b=2(a), C_b=3(b).

As shown in Fig. 5, the structure of multi-capacity routing channel is composed of multiple sub-routing channels for single-capacity. Each sub-routing channel follows the design of routing channel in the MMCF [13]. Remarkably, the construction method is stackable and it is easy to construct routing channel with $C_b > 2$, as shown in Fig. 5(b). In order to reduce the number of nodes and edges in the commodity flow graph, we use 4-nodes type shown in Fig. 5(a) if the adjacent pins (for example, p1, p2, p3) need to be escaped. Otherwise, 2-nodes type shown in Fig. 5(b) is used.

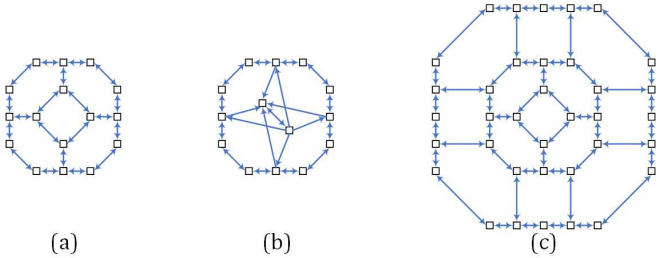


Figure 6: Tile node with C_d=4(a), C_d=3(b), C_d=6(c).

For the construction of candidate terminal nodes, a candidate terminal node is added for each sub-routing channel, which is similar to that in the MMCF [13]. Taking the case of C_d=4 as an example, Fig. 6(a) shows a new tile node design in which the capacity of each undirected edge is 1. And Fig. 6(b) shows the construction with C_d=3. Notice that the design of tile node is also stackable. It is easy to obtain a design with C_d=n by adding nested nodes around the design with C_d=n-2, as shown in Fig. 6(a)(c) (For example, the design with C_d=4 is nested inside the design with C_d=6). Based on the above three extension structures, we propose multi-capacity multi-commodity flow (MC-MCF) model shown in Fig. 7.

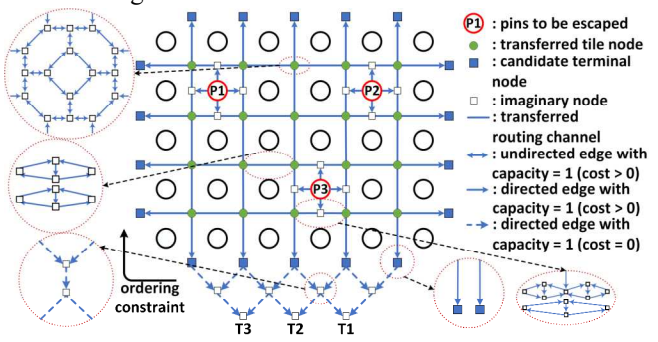


Figure 7: Illustration of the MC-MCF model for MC-OER with C_b=2, C_d=4.

Using the MC-MCF model to solve the OER problem with minimizing the wiring length, we first construct the commodity flow graph and convert it into an integer linear programming (ILP) problem which aims to minimize the total cost. Then, we use the ILP solver (such as Gurobi [17]) to solve it, and finally convert it into the solution of the OER problem. The formulation of the ILP problem for MC-MCF is as follows:

$$\min \sum_{(i,j)\in E_d, p\in P} F_{p,i,j} C_{i,j}$$

$$s.t. \sum_{(i,j)\in E_d} F_{p,i,j} + I_{p,j} = \sum_{(j,k)\in E_d} F_{p,j,k}, \forall p \in P, \forall j \in V, \quad (1)$$

$$\sum_{p\in P} F_{p,i,j} \le 1, \forall (i,j) \in E_d, \quad (2)$$

where P, V and E_d denote the sets of commodities (or pins), nodes and directed edges in MC-MCF graph, respectively. $C_{i,j}$ denotes the cost of directed edge $(i,j) \in E_d$. $F_{p,i,j}$ indicates whether the commodity p passes through edge (i,j). If node j is the source (or destination) node of commodity p, $I_{p,j}$=1 (or -1). Otherwise, $I_{p,j}$=0. Notice that non-crossing constraint is satisfied by the construction of MC-MCF graph and capacity constraint is satisfied by Eq. (2). Ordering constraint is satisfied by the setting of I.

3.3. Discussion on computational cost

Since the ILP and the OER problem have been proved to be NP-complete [18], we discuss the complexity of MC-MCF model by analyzing the number of nodes and edges in the commodity flow graph. A GPA with m rows, n columns and $C_d = 2C_b = 2c$ contains $(m-1)(n-1)$ tile nodes and $[m(n-1) + n(m-1)]$ routing channels. From the stackable structure of tile node, it's easy to find that the relationship between the number of nodes, directed edges and C_b is squared,

$$N_{node} = \sum_{i=1}^{c} 4(2i-1) + 2 \times 4[(i-1) + (2i-1)] = 16c^2 - 4c, \quad (3)$$

$$N_{edge} = 5\sum_{i=1}^{c} 4[(i-1) + 2(i-1)] = 30c^2 - 10c, \quad (4)$$

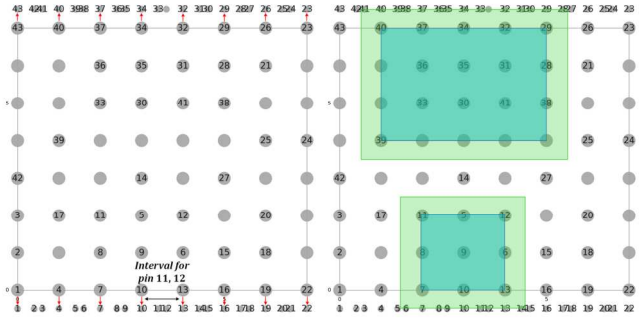
where N_{node}, N_{edge} imply the number of nodes, directed edges respectively. Similarly, the relationship between the number of nodes, directed edges and C_b in routing channel is linear. Therefore, the number of nodes, directed edges and C_b in the commodity flow graph is squared. With the increasing GPA scale and C_b, the number of nodes and edges in commodity flow graph increases rapidly, leading to unacceptable computational cost for solving the problem.

4. Accelerated MC-MCF model for OER

In this section, we propose a wiring resource driven partition strategy (WRDPS) and an accelerated MC-MCF algorithm to accelerate the solution of OER problem based on the MC-MCF model. In previous works, like [10], [14] only the clustering characteristics of pins are considered. In [15], neural network is used to extract high-dimensional routing features. However, the method in [15] can only be applied to the GPA in a fixed scale. In our method, the candidate terminal nodes are also considered in the partition step, as they are important routing resource in the MC-MCF model. Then, according to the arrangement characteristics of pins, we apply the fine-grained partition strategy to the accelerated MC-MCF algorithm, which further reduces the size of the sub-problem for OER.

4.1. Wiring resource driven partition strategy

The WRDPS considers the candidate terminal nodes as a basis for partition and contains three steps: *allocating candidate terminal nodes, constructing regions and selecting regions*. The allocation step focuses on assigning the closest possible candidate terminal node to the pin without violating the ordering constraint, which is based on *direct terminal node*. In an MC-MCF graph G, for a pin node p, if there is a candidate terminal node t directly connected to it, t is called a direct terminal node of p, and p is called a direct pin of t. Firstly, we assign candidate terminal nodes to pins that have direct terminal nodes, as shown by the red arrow in Fig. 8(a). Then, the illegal allocation pairs are removed (e.g. if the ordering constraint is not satisfied). The above allocation divides the sequence of pins and candidate terminal nodes into some intervals corresponding to each other. For each corresponding interval, the allocation is executed according to the principle of minimizing the total distance between pins and candidate terminal nodes. For the example in Fig. 8(a), the allocation for pin 10 and 13 generates an interval for pin 11 and 12. Then, the candidate terminal nodes numbered 11, 12 will then be assigned to pins 11, 12. We use A to denote the result of candidate terminal node allocation and $A[p]$ to denote the candidate terminal node allocated for pin p.



(a) (b)

Figure 8: Allocating terminal nodes (a) and constructing regions (b) in WRDPS.

Base on the result of the allocation step, the approach of constructing regions is as follows (each pin corresponds to a region). 1) Initialize region (R_i^k): R_i^k is the rectangular area containing exactly the pins numbered in $[k-w_1, k+w_2]$, $0 \le w_1, w_2 \le w_{max} = 2, w_1 = \max\{w_1 |$ pins numbered in $[k-w_1, k]$ satisfy distance constraint$\}, w_2 = \max\{w_2 |$ pins numbered in $[k, k+w_2]$ satisfy distance constraint$\}$. Pin s satisfies distance constraint if and only if $|x_k - x_s| \le l \wedge |y_k - y_s| \le l$ ((x_k, y_k) implies the coordinates of pin k). 2) Expand region (R_e^k): R_i^k is expanded in all directions by $h = 0.5$ unit lengths to obtain R_e^k. And if R_e^k does not include $A[p_k]$, expand it to include $A[p_k]$ exactly. This step is to provide sufficient wiring resources for the pins. As shown in Fig. 8(b), the dark green areas indicate R_i^7, R_i^{37} and the light green areas indicate R_e^7, R_e^{37}.

After region construction step, we need to select some regions as the partition result (R_f). These regions need to satisfy certain properties: pin continuity, mutual exclusivity, optimality. Pin continuity means that regions include continuous pin labels. Mutual exclusivity indicates that there is no intersection between the selected regions, and optimality means that the selected regions need to include as many pins as possible. We denote $R_{select} = \{R_1, R_2, \dots, R_L\}$ as regions that satisfy pin continuity property and $\mathrm{Sol}(a,b)$ as the sub-partition result while R_{select} is $\{R_a, \dots, R_b\}(1 \le a \le b \le L)$. $\mathrm{Sol}(a,b)$ satisfies:

$$\mathrm{Sol}(a,b) = [\mathrm{Sol}(a, m-1) - B(m)] \cup R_m \cup [\mathrm{Sol}(m+1, b) - B(m)], \quad (5)$$

$$m = arg\max_{a \le j \le b} \{\mathrm{PN}(\mathrm{Sol}(a, m-1) - B(j)) + \mathrm{PN}(R_j) + \mathrm{PN}(\mathrm{Sol}(j+1, b) - B(j))\}, \quad (6)$$

where $B(m)$ is the set of regions in R_{select} that do not satisfy mutual exclusivity with R_m and $\mathrm{PN}(R)$ is the number of pins in $R(R \subseteq R_{select})$. Finally, we regard $\mathrm{Sol}(1, L)$ as R_f.

4.2. OER based on MC-MCF and WRDPS

Based on the result of WRDPS, we can construct a MC-MCF model for each region $\in R_f$ and solve the problems in the regions respectively. However, routing conflicts may occur among adjacent regions, as the routing resource at the junction of the partitioned regions is shared by adjacent regions. The approach of removing conflicts is as follow: Firstly, we count the conflicts for each sub-model. Then, the sub-model with the most conflicts is selected to re-solve with taking the routing results of other sub-models as constraints. The process is repeated until there is no conflict. For the MC-OER problem with C_b=2 shown in Fig. 9(a), two adjacent regions share the routing resource at the junction. During the solution process, there are no restrictions on each other, so that the routing results of pin 2 and pin 3 conflict. To remove the conflict between the two regions, we re-solve region 2 with taking the routing results of region 1 as constraints, as shown in Fig. 9(b).

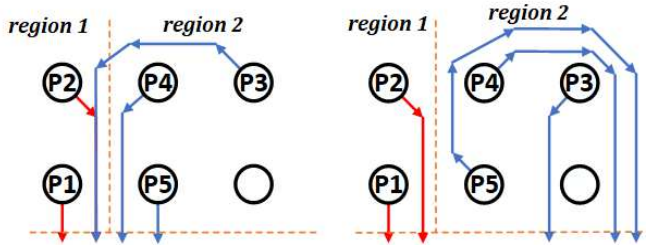


Figure 9: Illustration of routing conflicts (C_b=2).

It should be pointed out that there are pins out of the partitioned region. During the solution of each sub-model (for a region), it will cost unbearable time when there are too many pins in the region or the distribution of pins is complex. The pins generated by the above two conditions are called *non-escaped pins* and they are usually distributed complicatedly. For non-escaped pins, we propose a *fine-grained partition strategy* based on *ordering metrics* of pins and *unordered pin pair*. For an MC-MCF sub-graph, imaginary boundary nodes are added at the escape boundary, which are aligned with the pin nodes. These imaginary boundary nodes are then labelled according to the ordering constraint. The label of the nearest imaginary boundary node to pin node p is noted as the ordering metric for p which is denoted as $\mathrm{OD}(p) =$ label. As shown in Fig. 10, $\mathrm{OD}(p_3) = 5$. For a pin pair (p_i, p_j) in an MC-MCF subgraph, if $i < j$ and $\mathrm{OD}(i) > \mathrm{OD}(j)$, then (p_i, p_j) is said to be an unordered pin pair. In an MC-MCF sub-model, unordered pin pairs imply wire wrapping and increase the complexity of solving. Therefore, we can use unordered pin pairs to divide the pin sequence into a number of sub-sequences to improve the efficiency of the solution. The approach is divided into three sub-steps. Firstly, we find out all the unordered pin pairs and denote them as Pair . Secondly, for any two pairs $(p_{i1}, p_{j1}), (p_{i2}, p_{j2}) \in$ Pair, if $i_1 \le i_2$ and $j_1 \ge i_2$, then we combine both as $(p_{i1}, p_{max(j1,j2)})$. Finally, each unordered pin pair in Pair forms a sub-sequence, and the remaining pins form sub-sequences respectively. As shown in Fig. 10, after finding and combining, Pair $= \{(p_3, p_7)\}$ and we can easily divide pin sequence $[p_1, p_9]$ into three sub-sequences, i.e. $[p_1, p_2], [p_3, p_7]$ and $[p_8, p_9]$. Based on the three sub-sequences, the graph can be divided into three sub-graph (orange dot lines in Fig. 10).

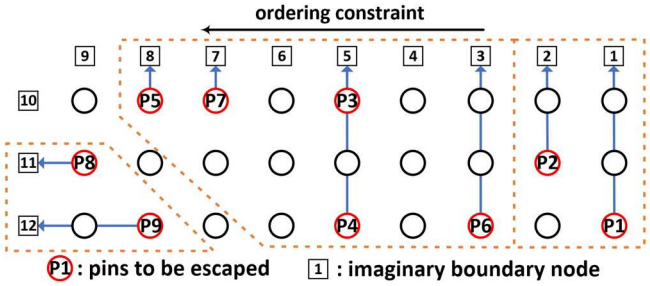


Figure 10: Illustration of the ordering metric.

We combine the techniques proposed above to obtain the accelerated MC-MCF algorithm presented in Alg. 1. In Alg. 1, we first construct sub-models based on the results of the WRDPS and solve them separately, then remove the routing conflicts among the sub-models and finally route the non-escaped pins with fine-grained partition strategy.

Algorithm 1: Accelerated MC-MCF algorithm

Input: An OER problem.

Output: The solution of OER.

1: Construct an MC-MCF model for the OER problem and partition the graph with the WRDPS to obtain R;

2: **for** $region \in R$ **do**

3: Using nodes and edges in $region$ to construct an MC-MCF graph and convert it into ILP problem;

4: Call an ILP solver to solve it;

5: end for

6: Remove conflicts among the results of ILP problems;

7: Route non-escaped pins with fine-grained partition strategy;

8: Convert the results of ILP problem into the solution of OER;

9: **return** the solution;

5. Experimental results

We have implemented the proposed algorithms and the MMCF [13] and ConDri(P) algorithm [14] in Python. To demonstrate the effectiveness of proposed algorithms, the cases of single-capacity OER in [14] are tested for the comparison among the proposed MC-MCF algorithm and those in [13] and [14]. Then, we test some cases for MC-OER. They are generated randomly or from industrial PCB design. Then, Gurobi optimizer solver is used as ILP solver [17]. All experiments are carried out on a machine with Intel Xeon E5-2630 CPU@2.40GHz and 512GB memory.

The comparison results of our MC-MCF algorithm and those in [13] and [14] are listed in Table 1. MC-MCF denotes the MC-MCF model with C_b=1. Size(#pin) indicates the grid number of GPA and the number of pins to be escaped. WL denotes the total wiring length, which is calculated by the same criteria as [14]. Notice that the test cases are originally for single-capacity OER, so that our method and method in [14] have achieved the solution close to optimal. From the table we see that, although the proposed MC-MCF algorithm has the unique advantage of solving multi-capacity OER problem, it performs similarly well as the algorithm in [14] for single-capacity OER problem. The runtime is similar, while the total wiring length of result is similar or even shorter. This experiment verified the effectiveness of the proposed method.

The results of multi-capacity OER with the proposed algorithms are listed in Table 2, where MC-MCF$_0$ denotes a direct implementation of MC-MCF model in Section 3 and MC-MCF denotes the accelerated MC-MCF algorithm in Section 4. #node denotes the number of nodes in the direct implementation of MC-MCF graph. #node(#sub) denotes the total number of nodes and the number of sub-graphs in the accelerated MC-MCF algorithm. From the table we see that, for the larger case the runtime of the direct MC-MCF algorithm is unacceptable (exceeding one day), while the accelerating approach in Section 4 brings several tens to over hundreds times speedup (see the last column). The number of nodes in the MC-MCF graph validate the complexity analysis in Section 3.3. The accelerating approach largely reduces the number of nodes in single MC-MCF graph. The test cases cover much diversity, where b1∼b3 and b9 are 2-side OER problems, b8 is a 1-side problem, and b4∼b7 and b10 are 4-side problems. For all of them, the proposed accelerated MC-MCF algorithm achieves 100% routability in reasonable time. For the first two cases, the accelerated algorithm still reaches the theoretically optimal wiring length. And, the results show that the increase of the complexity of problem does not lead to significant increase in time. Even for the largest test cases (b6∼b7, more than 300 pins), the proposed algorithm solves them in 38 minutes. The routing results for the largest case (b7) and the industrial case (b10) are shown in Fig. 11. Notice that these cases have no solution in the case of single-capacity. In other words, method in [13] and [14] cannot solve these OER problems.

6. Conclusion

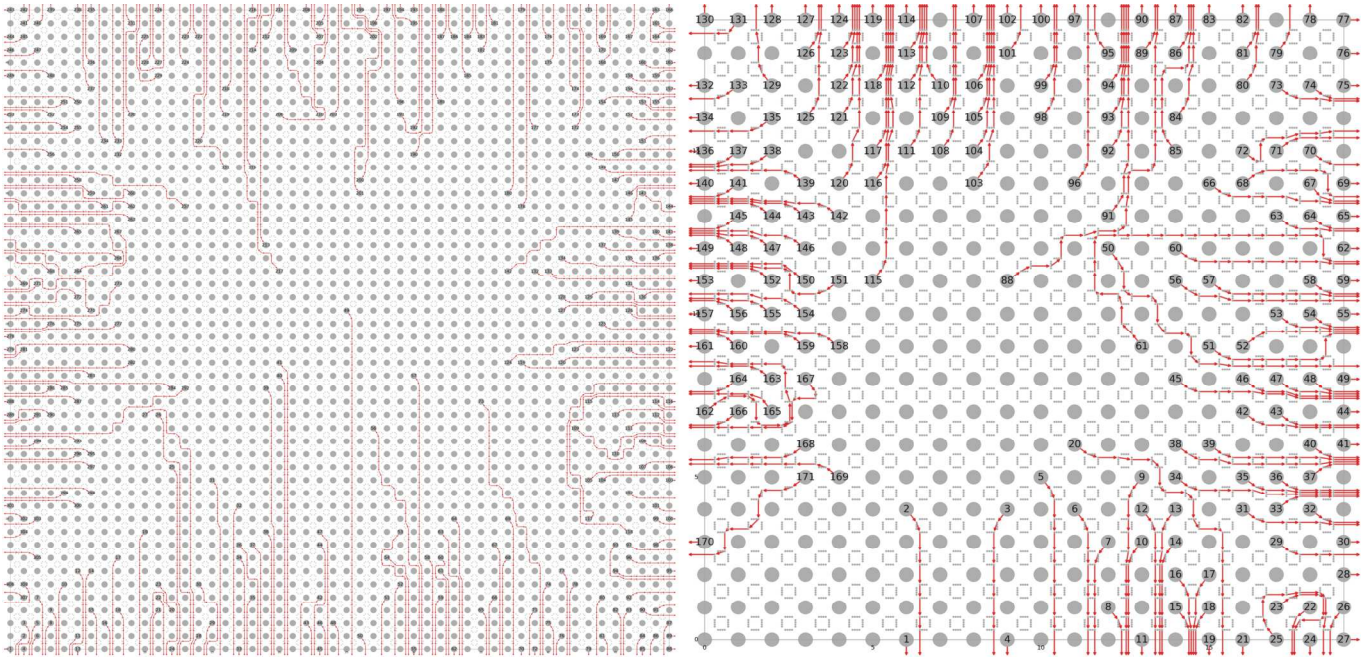
For the first time, a multi-capacity multi-commodity flow (MC-MCF) model is proposed for the multi-capacity ordered escape routing (MC-OER) problem with optimization objective of minimizing the wiring length. A wiring resource driven partition strategy (WRDPS) is proposed. Based on it, an accelerated MC-MCF algorithm was proposed to speed up the model solving. Experimental results show that the proposed method achieves 100% routability and the accelerated MC-MCF algorithm brings an average speedup of more than 231X with less sacrifice on wiring length. Compared to the methods in [13], [14], the accelerated MC-MCF algorithm performs similarly well or better for single-capacity OER. And, the proposed method is able to solve large-scale MC-OER problems (up to 308 pins in 50×50 GPA) in reasonable time.

Table 1: Comparison of our algorithm and those in [13], [14] (Time in unit of second).

Case	Size(#pin)	MMCF [13]		ConDri(P) [14]		MC-MCF	
		WL	Time	WL	Time	WL	Time
p1	10×6(25)	53.5	9.69	53.5	0.34	53.5	0.42
p2	20×21(42)	-	>86400	168	4.85	165	5.54
p3	25×26(60)	-	>86400	363	23.9	302	22.7
p4	50×50(130)	-	>86400	1160	80.7	1167	57.6

Table 2: Results of our algorithm for MC-OER.

Case	Size(#pin)	C_b	MC-MCF$_0$			MC-MCF			
			WL	Time	#node	WL	Time	#node(#sub)	Sp.
b1	10×5(28)	2	37	20.04	2537	37	0.76	1544(4)	26.4X
b2	8×8(43)	2	98.5	45895	3734	98.5	183	7693(5)	251X
b3	10×10(32)	2	-	>86400	7427	90	158	10217(4)	>547X
b4	23×24(93)	2	-	>86400	63326	338.5	206	50430(21)	>419X
b5	24×24(93)	2	-	>86400	66676	412.5	1171	104287(22)	>74X
b6	50×50(300)	2	-	>86400	270697	1617	2032	191174(73)	>43X
b7	50×50(308)	2	-	>86400	265945	1724	1388	213045(67)	>62X
b8	30×14(90)	3	-	>86400	57499	413	160	117343(20)	>540X
b9	30×30(158)	3	-	>86400	139287	641	360	200245(36)	>240X
b10	20×20(171)	4	-	>86400	191138	469.5	746	214450(33)	>116X



(a) Case b7 (b) Case b10

Figure 11: The routing results produced by the accelerated MC-MCF algorithm.

7. References

[1] T. Yan and M. Wong, "Recent research development in pcb layout," in *Proc. ICCAD*, 2010, pp. 398–403.

[2] Y. Takeuchi, et al., "Escape routing design to reduce the number of layers in area array packaging," *IEEE Trans. on Advanced Packaging*, pp. 686–691, 2000.

[3] R. Shi and C. Cheng, "Efficient escape routing for hexagonal array of high density i/os," in *Proc. DAC*, 2006, pp. 1003–1008.

[4] T. Yan and M. Wong, "A correct network flow model for escape routing," in *Proc. DAC*, 2009, pp. 332–335.

[5] Y. K. Ho et al., "Escape routing for staggered-pin-array pcbs," IEEE Trans. Computer-Aided Design, vol. 32, no. 9, pp. 1347–1356, 2013.

[6] P. Wu and M. Wong, "Network flow modeling for escape routing on staggered pin arrays," in *Proc. ASP-DAC*, 2013, pp. 193–198.

[7] Y. Tomioka et al., "Monotonic parallel and orthogonal routing for single-layer ball grid array packages," in *Proc ASP-DAC*, 2006, pp. 6–pp.

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

[9] L. Luo et al., "Ordered escape routing based on boolean satisfiability," in *Proc. ASP-DAC*, 2008, pp. 244–249.

[10] L. Luo and M. Wong, "On using sat to ordered escape problems," in *Proc. ASP-DAC*, 2009, pp. 594–599.

[11] J. Yan, C. Ke, and Z. Chen, "Ordered escape routing via routability-driven pin assignment," in *Proc. GLSVLSI*, 2010, pp. 417–422.

[12] K. Sattar and A. Naveed, "Ordered escape routing using network flow and optimization model," in *Proc. ICARA*, 2015, pp. 563–568.

[13] F. Jiao and S. Dong, "Ordered escape routing for grid pin array based on min-cost multi-commodity flow," in *Proc. ASP-DAC*, 2016, pp. 384–389.

[14] Z. Liao et al., "A constraint-driven compact model with partition strategy for ordered escape routing," in *Proc. GLSVLSI*, 2020, pp. 393–398.

[15] Z. Chen et al., "Machine learning based acceleration method for ordered escape routing," in *Proc. GLSVLSI*, 2021, pp. 365–370.

[16] K. Wang et al., "Escape routing of mixed-pattern signals based on staggered-pin-array pcbs," in *Proc. ISPD*, 2013, pp. 93–100.

[17] Gurobi Optimization, https://www.gurobi.com.

[18] Q. Ma and M. Wong, "Np-completeness and an approximation algorithm for rectangle escape problem with application to pcb routing," *IEEE Trans. Computer-Aided Design*, 9, pp. 1356–1365, 2012.