Target Pin-Pair Selection Algorithm
Using Minimum Maximum-Edge-Weight Matching for Set-Pair Routing

Kano Akagi
Department of Information and Communications Engineering, Tokyo Institute of Technology, Japan
akagi@eda.ict.e.titech.ac.jp

Shimpei Sato
satos@ict.e.titech.ac.jp

Atsushi Takahashi
atsushi@ict.e.titech.ac.jp

Abstract—Routing problems derived from silicon-interposer and etc. are often formulated as a set-pair routing problem where the combination of pin-pairs to be connected is flexible. In this paper, we propose an algorithm that gives a target pin-pair set which is used to efficiently generate a length matched routing pattern in set-pair routing. In our proposal algorithm, a target pin-pair set is formulated as a perfect matching in a complete bipartite graph. In order to obtain a better target pin-pair set efficiently, we propose an algorithm that obtains a perfect matching whose maximum-edge-weight is minimum. By using our algorithm, a target pin-pair set where distant pin-pairs are excluded is obtained and lower bounds of the maximum of connection lengths and the total connection length in set-pair routing are prevented to become large. The effectiveness of our algorithm is discussed by using several small illustrative instances.

I. Introduction

In routing pattern generation, a routing pattern that realizes not only connection requirements but also short signal propagation delays, short signal propagation delay difference, and etc. is required. Roughly speaking, the signal propagation delay can be considered to be proportional to the length of a connection. Therefore, a routing pattern that has small length and small length difference is often pursued.

Routing problems for silicon-interposer [1, 2], printed circuit board [3, 4], FPGA [5], and etc. are often defined as a set-pair routing problem where the combination of pin-pairs to be connected is flexible. In set-pair routing problem, a routing pattern that has the minimum total length can be obtained in polynomial time. On the other hand, a routing pattern that achieves length matching, that is, small length and small length difference is not easy to obtain [6].

In this paper, we propose an algorithm that gives a better target pin-pair set. A target pin-pair set is used to generate a length matched routing pattern in [7].

In order to obtain a routing pattern that achieves length matching as much as possible, heuristic algorithms that are based on minimum cost flow had been proposed [6, 8]. In these algorithms, a routing pattern that corresponds to a minimum cost flow is used as an initial solution. A routing pattern used as the initial solution is obtained in polynomial time. Then, the initial solution is modified so that the length difference becomes smaller. These algorithms improve length matching iteratively by checking the effect of candidate modifications one by one. However, no good ways to find modifications that improve length matching are known. These algorithms consume much time to find modifications and do not necessarily improve the length matching well.

In a set-pair routing problem, no pin-pair to be connected is specified. However, pin-pairs to be connected in good solutions could be specified. If an initial solution that connects these pin-pairs is obtained efficiently, a length matched routing pattern could be obtained with small modifications.

In order to obtain a length matched routing pattern efficiently, we consider a routing design flow in which (1) target pin-pairs are selected, (2) a routing pattern that connects the target pin-pairs as much as possible is obtained as an initial solution, and (3) the initial solution is modified according to the analysis of it. A routing algorithm for (2) has been proposed in our previous work [7].

In this paper, we propose an algorithm for (1) that selects target pin-pairs that include no distant pin-pairs. Target pin-pair selection problem is formulated as a minimization problem of the maximum-edge-weight of perfect matching in a bipartite graph. Our algorithm obtains such a perfect matching, and outputs a pin-pair set that corresponds to the matching as the target pin-pair set. As an analysis, an effectiveness of our algorithm is discussed. By setting an edge weight as the distance between a pin-pair, a target pin-pair set in which the maximum of the distance between pin-pairs is minimized is obtained.

The existence of a routing pattern connecting all pin-pairs in the target pin-pair set is not necessarily be guaranteed. However, since a target pin-pair set obtained by our algorithm includes no distant pin-pairs, if all target pin-pairs are connected, a length matched routing pattern is expected to be obtained. In experiments, it is confirmed that our algorithm is useful to obtain a length matched routing pattern.

II. Set-Pair Routing Problem

In the routing design such as silicon-interposer [1, 2], printed circuit board [3, 4], FPGA [5], and etc., the combination of pin-pairs to be connected is often flexible when connection requirements are given between passive elements, I/O pins of reconfigurable chip or etc. We call such routing problem as set-pair routing problem. In these routing problems, a routing pattern that realizes short signal propagation delays, short signal propagation delay difference and etc. is required.
In the set-pair routing problem, the connection requirements are given between the source-pin set and the sink-pin set. As an input of the problem, the locations of source-pins and sink-pins are given. A pin in the source-pin set and a pin in the sink-pin set are connected in the given routing area. All the pins should be connected without intersecting each other, and the combination of a pin-pair is flexible.

In this paper, a single layer routing area with an orthogonal routing grid with obstacles is assumed. Source-pins and sink-pins are placed on grid points of the routing grid. The routing area is modeled by an undirected grid graph, called routing graph, where a vertex corresponds to a grid point of the routing grid and an edge corresponds to the vertical or horizontal segment between adjacent grid points. A connection between pin-pair is a path that connects corresponding vertices in the routing graph.

A routing pattern on routing grid is said to be feasible in set-pair routing if each connection in the routing pattern connects source-pin and sink-pin without using grid points with obstacles, and the number of connections is equal to that of source-pin set, and connections are mutually disjoint. The set-pair routing problem can be regarded as a disjoint paths problem between set-pairs. In the disjoint paths problem, network flow algorithms [9, 10] are useful to obtain solutions. In set-pair routing, the existence of a feasible routing pattern is easily checked by a maximum flow algorithm. Therefore, in the following discussion, we focus on the problem instances that have a feasible routing pattern.

In network flow problems, a minimum cost flow is obtained in polynomial time. In algorithms proposed in [6, 8], a minimum cost flow algorithm is used to obtain a length matched set-pair routing whose total length is minimum.

An example of set-pair routing problem instance on single routing layer is shown in Fig. 1. The red rectangles and the blue rectangles represent source-pins and sink-pins, respectively. The bold lines which connect a source-pin and a sink-pin represent connections. Two routing patterns are shown in Fig. 1(a) and in Fig. 1(b). The total length of them is minimum. The length difference of the former is larger than that of the latter. Therefore, the routing pattern in Fig. 1(b) is more preferred. A routing pattern as shown in Fig. 1(a) is obtained by typical minimum cost flow based algorithms. While, a routing pattern as shown in Fig. 1(b) is the target of our design flow.

III. Nakatani method

In order to obtain a routing pattern that achieves length matching as much as possible in a set-pair routing problem, a heuristic algorithm, called Nakatani method [6], had been proposed. Nakatani method utilizes network flow and consists of three steps. The outline of Nakatani method is shown in Fig. 2(b).

1. **Total Length Minimization**: Obtain a routing pattern that has minimum total length by finding a minimum cost maximum flow. The routing pattern obtained in this step is used as the initial solution.

2. **Length Difference Reduction**: Modify the initial solution so that the length matching is improved while keeping the total length minimum. The initial solution is modified by using zero-cost cycles in the flow-graph. The modifications are iterated, and the initial solution is gradually improved.

3. **Minimum Length Increase**: Lengthen a connection of the minimum length as much as possible by using R-Flip [3] iteratively while the maximum length is kept.

In Nakatani method, several ideas to find a good solution effectively were introduced. However, Nakatani method is time consuming and does not always find a length matching pattern. The initial solution tends to have large length difference since pins that are connected last in Step 1 are often distant. No good ways to find modifications that improve length matching are known. In Step 2, the length matching is iteratively improved by checking the effect of candidate modifications one by one. There are a lot of candidates, but an effective candidate is a few. The number of attempts tends to be much, so Step 2

![Fig. 1. Example of Set-pair routing.](image)

![Fig. 2. The outline of design flow for length matching in set-pair routing.](image)
tends to be a time consuming. Furthermore, a routing pattern that has the minimum total length does not always have the smallest length difference. If the routing patterns whose total length is minimum are explored, it is impossible to obtain an optimal solution if the total length of it is not minimum.

IV. TARGET PIN-PAIR SELECTIONS

Target pin-pair selection problem is formulated as a minimization problem of maximum-edge-weight of perfect matching in a bipartite graph. We propose an algorithm in order to obtain a perfect matching whose maximum-edge-weight is minimum in $G$.

A. Overview

In this paper, we assume the design flow that utilizes target pin-pairs in order to obtain a length matched routing pattern efficiently in set-pair routing. A target pin-pair is a pin-pair that is expected to obtain a length matched routing pattern by connecting them.

Our design flow for length matching in set-pair routing consists of three steps. The design flow is shown in Fig. 2(a). Our algorithm proposed in this paper is assumed to be used in Step 1.

As an input of the problem, source-pins and sink-pins are given, each consisting of $n$ pins. Target pin-pair candidates are pairs of pins that belong to different pin sets. Our algorithm selects, disjoint $n$ pin-pairs as a target pin-pair set. The problem finding disjoint path is NP-hard when two or more two-pin nets are given on planner grid [11]. It is not easy to determine whether a target pin-pair set has a feasible routing pattern connecting all of them. Therefore, a target pin-pair set is allowed to be selected as a target even if no feasible routing pattern exists when these pairs are connected.

The length of a connection is always equal to or larger than the distance of pins to be connected. Therefore, the connection of the distant pins is long. A routing pattern with large length connections would be obtained if a distant pin-pair is selected as a target pin-pair and is connected.

In our algorithm, a bipartite graph $G$ whose vertex, edge, and edge weight correspond to pin, target pin-pair candidate, distance between pin-pairs is used to obtain a target pin-pair set. A set of disjoint pin-pairs corresponds to a matching, that is, a set of edges that have no common end vertices in $G$. A target pin-pair set corresponds to a perfect matching in $G$.

In this paper, a perfect matching where the sum of edge weights is minimum is called as min-total matching. A perfect matching where the maximum-edge-weight is minimum is called as min-max matching.

A min-total matching is not necessarily equivalent to a min-max matching. A min-total matching might include a large weighted edge. We propose an algorithm that obtains not a min-total matching but a min-max matching in a bipartite graph.

An instance of a set-pair routing is shown in Fig. 3(a). A bipartite graph $G'$ is shown in Fig. 3(b). A min-max matching $\{(A, X), (B, Y), (C, Z)\}$ is shown in Fig. 4(a). A min-total matching $\{(A, Z), (B, Y), (C, X)\}$ is shown in Fig. 4(b). Edges in matching are represented by thick lines, and the other edges are represented by dotted lines. In Fig. 4(a), the maximum weight is 8 and the maximum weight is minimum among all perfect matchings, and the total weight is 24. In Fig. 4(b), the maximum weight is 10, and the total weight is 22 and the total weight is minimum among all perfect matchings. The total weight of the former is larger than that of the latter. On the other hand, the maximum weight of the former is smaller than that of the latter. By our proposed algorithm, the former is obtained since the maximum weight is minimum among all perfect matchings.

An algorithm to find a min-max matching in a bipartite graph was proposed [12]. In the algorithm, the existence of a perfect matching in the graph obtained by removing edges whose weight are larger than the threshold is repeatedly checked. A min-max matching is obtained by finding the minimum threshold by binary search. The time complexity of the algorithm in [12] is $O(n^{2.5} \log n)$. This algorithm works well but the number of repetitions in binary search is relatively large for larger instance. Our proposed algorithm reduces the number of repetition by finding min-total matching and by checking the maximum weight in the obtained matching.

B. Our proposed min-max matching algorithm

The flowchart of our proposed min-max matching algorithm is shown in Fig. 5. Our algorithm gradually shrinks solution space in which finding a min-total matching and removing large weight edges is repeated. In the complete bipartite graph
where the maximum edge weight is finite, the existence of a perfect matching is guaranteed since we assume the existence of a feasible routing pattern. Our algorithm always outputs a perfect matching with finite maximum-edge-weight.

The algorithm consists of 4 steps shown in Fig. 5. The detail of each step is as follows.

(i) Construct the complete bipartite graph \( G \) whose vertex, edge, and edge weight correspond to pin, target pin-pair candidate, and distance between pin-pairs, respectively.

(ii) Find a min-total matching \( M \) in bipartite graph \( G \). In this step, a perfect matching with finite maximum-edge-weight always exist in \( G \). A min-total matching can be found by applying Hungarian algorithm [13].

(iii) Let \( w \) be the maximum-edge-weight in \( M \). Remove all edges from \( G \) except edges whose weight are less than \( w \), and let resultant graph be \( G' \).

(iv) If there is no perfect matching in \( G' \), then output \( M' \), otherwise return to (ii).

The validity of our proposed algorithm is as follows. Let \( M \) and \( G \) be the matching outputted by our algorithm and the bipartite graph when \( M \) is outputted. Also, let \( w \) be the maximum-edge-weight in \( M \). If \( M \) is not a min-max matching, that is, if \( M \) is not optimal, then there is an optimal perfect matching \( M^* \), and all edge weights in \( M^* \) are less than \( w \). \( G \) contains a perfect matching \( M^* \) since \( G \) keeps all the edges whose weight are less than \( w \), and the algorithm does not output \( M \) since there is a perfect matching \( M^* \) in \( G \).

Let \( G' \) be a complete bipartite graph corresponding to Instance-1 shown in Fig. 3(a) and let \( G'_{e} \) be a graph consisting of all edges whose weight is less than or equal to \( w \) in \( G' \). Also, let \( M'_{e} \) be a min-total matching in \( G'_{e} \). \( G' \) is shown in Fig. 3(b).

- \( M'_{e} \) in \( G'_{e} (= G') \) is shown in Fig. 6(ii-1). The maximum-edge-weight in \( M'_{e} \) is 10.
- Since the maximum-edge-weight in \( M'_{e} \) is 10, our algorithm generates \( G'_{i0} \) from \( G'_{e} \) by removing \((C, X), (C, Y)\) whose edge weight is 10 or more. \( G'_{i0} \) is shown in Fig. 6(ii-2). \( M'_{i0} \) is shown in Fig. 6(ii-3). In \( G'_{i0} \), a perfect matching exists.
- \( M'_{i0} \) in \( G'_{i0} \) is shown in Fig. 6(ii-3). The maximum-edge-weight in \( M'_{i0} \) is 9, and it is less than that in \( M'_{e} \).

Our algorithm iterates steps from (ii) to (iv) until no perfect matching exists in \( G'_{e} \). As for Instance-1 shown in Fig. 3(a), our algorithm obtains \( M'_{e} \) as shown in Fig. 6(ii-3), then generates \( G'_{i3} \) as shown in Fig. 6(ii-3). In \( G'_{i3} \), no perfect matching exists since vertex \( C \) and \( X \) are isolated. \( M'_{e} \) is outputted and our algorithm terminates.

V. Evaluation

A. Computational Complexity

Let \( P \) be the number of target pin-pairs, \( V \) be the number of the vertices in a routing graph, and \( E \) be the number of edges....
in a routing graph. The time complexity of our algorithm is $O(P^2 V^2 + P^3)$. In the rest of this section, the computational complexity of each step is discussed.

The number of iterations for each step is as follows. Step (i) is executed once, and Step (ii) - (iv) are executed up to $P^2$ times.

The distances from one source-pin in $S$ to all sink-pin in $T$ are obtained by applying Dijkstra algorithm [14]. The edge weights in complete bipartite graph used in our algorithm is set by applying the Dijkstra algorithm to all the pins of $S$. The time complexity of Step (i) is $O(P^2 V^2)$ since Dijkstra algorithm runs in $O(V^2)$ and the number pins of $S$ is $P$. The Hungarian algorithm finds a min-total matching in $O(P^3)$. The time complexity of Step (ii) is $O(P^3)$. The maximum-edge-weight in a perfect matching is found in $O(P)$. The time complexity of finding the maximum-edge-weight is $O(P^3)$ and that of removing the edges from $G$ is $O(P^2)$. Therefore, the time complexity of Step (iii) is $O(P^3)$. In Step (iv), whether perfect matching exists is determined by obtaining the min-total matching. Therefore, the time complexity of Step (iv) is at most that of Step (ii) and can be ignored.

B. Experiments

In this section, we discuss the effectiveness of target pin-pair set obtained by our algorithm. The effectiveness is evaluated by comparing routing patterns generated by our design flow in which target pin-pair connection algorithm [7] is used to connect target pin-pairs obtained by our algorithm and routing patterns generated by Nakatani method [6]. In routing pattern generation, the implementation uses the same one used in [6] and [7].

The instances used in evaluation are shown in Fig. 3(a) and Fig. 7(a). The min-max matchings are shown in Fig. 4(a) and Fig. 7(b). In both instances, target pin-pair set obtained by our algorithm are $\{(A, X), (B, Y), (C, Z)\}$. For both instances, the maximum-edge-weight in min-max matching is 8. That is, the maximum of connection lengths in any routing pattern is 8 or more.

Hereinafter, routing patterns generated in Step 2 of our design flow, Step 3 of our design flow, Step 1 of Nakatani method, Step 2 of Nakatani method, and Step 3 of Nakatani method are represented by A pattern, AD pattern, B pattern, BC pattern, and BCD pattern, respectively.

TABLE I and TABLE II show the statistics of routing patterns generated by our design flow and Nakatani method, respectively. The total length, the maximum length, the minimum length, and the maximum length difference are shown in $L_{\text{total}}$, $L_{\text{max}}$, $L_{\text{min}}$, and $\delta_{\text{max}}$, respectively. The maximum length $L_{\text{max}}$ and the maximum length difference $\delta_{\text{max}}$ of A pattern is smaller than those of B pattern in both instances.

A patterns of Instance-1 and Instance-2 are shown in Fig. 8(A) and Fig. 9(B). The maximum length in A patterns are 8 and 9. In A pattern of Instance-1, all of target pin-pairs are connected. On the other hand, in A pattern of Instance-2, pins different from the target pin-pair are connected. A routing pattern that connects all of target pin-pairs in Instance-2 are shown in Fig. 10. The maximum distance between pin-pair in the target pin-pair set is 8, but this routing pattern includes a connection of length 10. In Instance-2, it is impossible to achieve maximum length 8, and any routing pattern that connects all the target pin-pairs include a connection of the length 10 or more. On the other hand, the maximum length of A pattern is 9. Although a target pin-pair is not always connected in our design flow, a routing pattern with a small maximum length may be generated as a result.

B patterns of Instance-1 and Instance-2 are shown in Fig. 8(B) and Fig. 9(B). In both B pattern, pins different from
the target pin-pair are connected. The maximum length of each pattern is 10. In order to reduce the maximum length, it is necessary to modify the routing pattern. However, in length difference reduction (BC) of the Nakatani method, it is impossible to achieve maximum length 8 since the maximum length 8 is only achieved when total length is not minimum in these instances. The maximum length of BC patterns are 9 and 10. Both of them are larger than those of each A pattern.

It is confirmed that a better routing pattern compared to Nakatani method is obtained according to our design flow.

VI. Conclusion

In this paper, a target pin-pair selection algorithm was proposed in order to obtain a length matched routing pattern in set-pair routing problem. A target pin-pair is a pair of pins belonging to different pin-set. A target pin-pair set is a set of disjoint pin-pairs. A target pin-pair selection problem was formulated as a minimization problem of the maximum-edge-weight of perfect matching in a bipartite graph, and an algorithm that gets such matching was proposed.

By applying to our algorithm and a target pin-pair connections algorithm, it is confirmed that a length matched routing pattern was obtained. Implementing our algorithm and evaluating the effectiveness of our algorithm by using practical problem instances are in our future tasks.

ACKNOWLEDGEMENTS

We would like to thank Prof. Tomomi Matsui and Prof. Shmuel Wimer for their helpful discussions.

REFERENCES


