

A Routing Method based on Nearest Via Assignment for 2-Layer Ball Grid Array Packages

Yoshiaki KURATA[†] Yoichi TOMIOKA[†] Yukihide KOHIRA[†] Atsushi TAKAHASHI[†]

[†]Tokyo Institute of Technology

Dept. of Communications and Integrated Systems
2-12-1-S3-58 Ookayama, Meguro-ku, Tokyo 152-8550, Japan
{kurata, tomioka, kohira, atsushi}@lab.ss.titech.ac.jp

Abstract— In this paper, we propose a routing method for 2-layer ball grid array packages that generates a routing pattern satisfying the constraint of wire congestions. In the proposed method, the via of a net is restricted to be placed near the ball of the net, and a routing pattern that satisfies the constraints is formulated as a mixed integer programming. In experiments with several data, we obtain a routing pattern that satisfies the constraints of wire congestion within a practical time by using a mixed integer programming solver.

I. INTRODUCTION

Ball Grid Array (BGA) packages, which have hundreds of I/O pins of grid pattern, can realize a number of connections between chip and PCB. However, since the routing area of a BGA package is very small and has a number of obstacles, it is difficult to realize all connections under such severe circumstance. So currently package routing is done manually and needs a lot of time, and the development of auto routing is desired.

A routing method for BGA packages with single layer was proposed in [1] and was improved in [2]. In these methods, a netlist that achieves low wire congestion and the corresponding global routing are generated. For flip chips, a method based on mixed integer programming was proposed in [3]. Although the problem formulation of flip chip routing is similar to that of BGA routing, the method in [3] cannot be used for the multi-layer BGA packages as it is since it assumes a single routing layer.

Routing algorithms for multi-layer BGA and pin grid array (PGA) have been proposed in [4] and [5], respectively. These algorithms first assign each net to a layer and then generate routes in each layer. However, in these methods for multi-layer, the assignment of vias to the grid area where solder balls are placed is not considered. Generally, in the BGA packages, most vias are needed to be assigned to the grid area.

On the other hand, a via assignment method to the grid area in 2-layer BGA package was proposed in [6]. In this method, the total wire length and the wire congestion on layer-1 are decreased by repeatedly adjusting the via assignment. This method was improved in [7]. In the method in [7], the execution time is reduced and the routing ratio of layer-2 has been improved. However, a

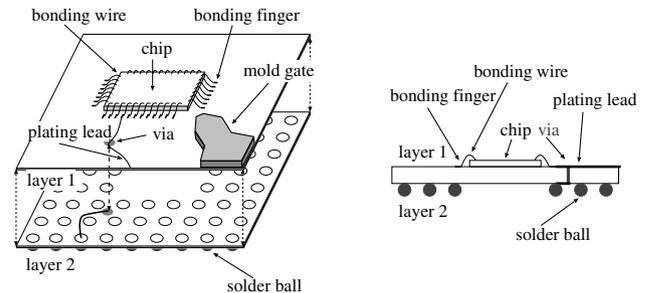


Fig. 1. 2-layer BGA package

routing pattern that satisfies the constraints of wire congestion is not necessarily obtained by the method in [7]. Moreover, a routing pattern obtained by the method in [7] consumes the routing resource of layer-2 much, since the via of a net is allowed to be assigned to a grid node which is far from the ball of the net. If an enough routing resource on layer-2 is secured, some plating leads of signal nets are routed on layer-2 to reduce the wire congestion of layer-1 by using a method proposed in [8]. Moreover, in actual design, the routing resource of layer-2 is requested to be reserved for the plating leads for power supply nets.

In this paper, we propose a routing method for 2-layer ball grid array packages that generates a routing pattern satisfying the constraint of wire congestions. In the proposed method, the via of a net is restricted to be placed near the ball of the net, and a routing pattern that satisfies the constraints is formulated as a mixed integer programming. In experiments with several data, we obtain a routing pattern that satisfies the constraints of wire congestion within a practical time by using a mixed integer programming solver.

II. PRELIMINARY

A. 2-layer BGA package

A model of BGA package that contains a single chip and a package substrate with two routing layers is shown in Fig. 1. A bonding finger, which is referred to as a *finger*, is placed on the perimeter of a rectangle enclosing the chip on layer-1, and is connected to the chip by a bonding wire. A solder ball, which is referred to as a *ball*,

is an I/O terminal of the package, and is placed in a grid array pattern on layer-2. A *mold gate* which is used to pour resin into the package is placed in a corner of the substrate on layer-1.

In package routing design, a connection requirement is given as a net which consists of fingers and balls. In this paper, we assume that a net consists of one finger and one ball. A net is realized by using wires on each layer and vias which connect wires on different layers. However, there are several special constraints. In this paper, two types of constraints are considered. The first one is related to mold gate. In the region at which a mold gate is placed, a wire on layer-1 is not allowed, but a wire on layer-2 is allowed. The second one is related to electric plating to protect wires. In order to enable electric plating, a net is requested to extend its wire to the substrate boundary on either layer. The extra connection is called a *plating lead*.

B. Problem Definition

The package substrate is divided into four sectors, and the nets are divided accordingly. In the following, we consider the bottom sector as shown in Fig. 2. Nets are labeled according to the order of fingers on the perimeter from the left to the right as $\Omega = \{1, 2, \dots, n\}$, where n is the number of nets.

The grid array pattern of balls is modeled by a ball grid array. The set of balls is denoted by B . According to the ball grid array, a sector is divided into unit squares surrounded by four adjacent balls.

The candidate positions of vias are restricted since the size of a via is relatively large even though it is smaller than the size of a ball. In our model, the number of vias to be placed in a unit square is at most one, and the via is placed to the center of a unit square. The center of a unit square is called a *grid node*. For ease of explanation, *dummy grid nodes* corresponding to the boundary of the routing region of layer-1 are introduced. A dummy grid node corresponding to the left boundary of a row is placed to the left of the leftmost grid node of the row. Similarly, dummy grid nodes corresponding to the right boundary are generated.

The set of grid nodes which are not dummy grid nodes is denoted by U . Note that a via can be assigned to the grid nodes. The set of dummy grid nodes is denoted by D . Note that no via is assigned to the dummy grid nodes. The grid array pattern of $U \cup D$ is called a *via grid array*. In Fig. 2, balls are shown as gray circles, grid nodes are shown as white circles and dummy grid nodes are shown as small gray circles.

Let u and v be a pair of adjacent grid nodes in $U \cup D$. If u and v are adjacent horizontally, the interval (u, v) between u and v is called a *horizontal interval*. Otherwise, the interval is called a *vertical interval*. Let G_h and G_v be the set of horizontal intervals and vertical intervals, respectively. The length of an interval (u, v) in $G_h \cup G_v$ is one if u and v are in U . Otherwise, the length is the distance between the corresponding grid node and the corresponding boundary of the routing region of layer-1.

The condition of the wire congestion is given depending on the packaging technology. In order to get a smaller

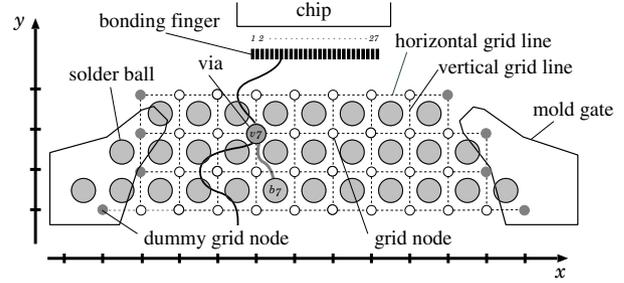


Fig. 2. Fingers and balls, a via grid array in the bottom sector of a package

package, the interval is set to be short as long as the connection requirements are satisfied. In our model, the length of an interval is set so that one wire can go through between adjacent balls on layer-2. It is the minimum except trivial cases. While, the number of wires that can intersect the interval on layer-1 is larger than one if there is no obstacle. The number of wires that can intersect interval (u, v) on layer-1 is denoted by $capacity(u, v)$. In experiments, if both of u and v are not dummy grid nodes, $capacity(u, v) = 4$ is used.

Since the routing on layer-2 is tight, we restrict the structure of each net so that it has just one via, and that its plating lead is routed on layer-1. Let b_i and v_i be the ball and the via of net i , respectively. In Fig. 2, a global routing of net 7 is illustrated. The route on layer-1 is shown as black line and the route on layer-2 is shown as gray line.

A via assignment is an assignment of vias to grid nodes which are not dummy. The global routing problem for a 2-layer BGA package is defined as follows:

<p>Routing problem for the 2-layer BGA Inputs: Finger, ball, and netlist Outputs: A via assignment and global routing on both layers Objective: Minimize the total wire length on both layers Constraints: The condition of the wire congestion</p>

C. Monotonic routing

Let (x_i, y_i) be the position of via v_i in a via assignment. It is clear that a monotonic routing is possible for the via assignment if and only if $x_i < x_j$ is satisfied for any pair of nets i and j ($i < j$) such that $y_i = y_j$. A via assignment is said to be *monotonic* if a monotonic routing of layer-1 is possible [1],[6].

Given a monotonic via assignment, monotonic routing on layer-1 is uniquely determined. An example of a monotonic via assignment and the monotonic routing for the via assignment are shown in Fig. 3. The routes of nets 10, 11, ..., and n pass to the right of via v_9 , and the routes of nets 1, 2, ..., and 12 pass to the left of via v_{13} . Therefore,

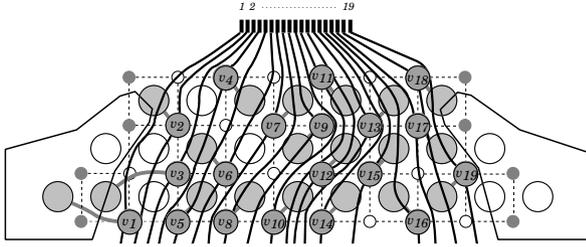


Fig. 3. An example of a global routing

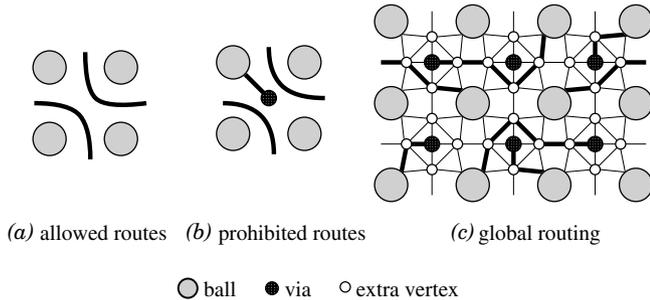


Fig. 4. The structure of a routing graph on layer-2

the routes of nets 10, 11, and 12 pass between vias v_9 and v_{13} .

In this paper, via assignment and global routing of layer-1 are restricted to be monotonic. By definition, a monotonic route intersects a horizontal grid line of the via grid array just once. While, if a monotonic route snakes, then the length of the route becomes longer. The number of routes between via v_i and the via above v_i is denoted by $cut_a(v_i)$. The total wire length of layer-1 is evaluated by $\sum_{i \in \Omega} cut_a(v_i)$.

D. Routing graph

On layer-2, the number of routes between adjacent balls is at most 1. Two routes can pass through a unit square if a via is not placed in the unit square as shown in Fig. 4(a). While, two routes cannot pass if a via is placed in the square as shown in Fig. 4(b).

We use the routing graph $G = (V, E)$ proposed in [8] to represent a global routing on layer-2. The routing graph has vertices corresponding to balls, vias and extra vertices. In Fig. 4, balls are shown as gray circles, vias are shown as black circles and extra vertices are shown as small white circles. Extra vertices are introduced in the routing graph to generate routes that satisfy the design rule. The set of extra vertices is denoted by S .

A subgraph of the routing graph and an example of global routing on it is shown in Fig. 4(c).

III. MIXED INTEGER PROGRAMMING FORMULATION

A. Outline of the method

In our method, two kinds of restrictions are imposed to reduce the variety of the routing patterns while keeping most of feasible routing patterns and to effectively obtain a feasible routing pattern. First, via assignment and global routing of layer-1 are restricted to be monotonic. Second, the via of a net is restricted to be placed to grid nodes which are neighborhood of the ball of a net. The routing problem with the restrictions is formulated as a mixed integer programming.

In the formulation, two kinds of variables are used. First one corresponding to a grid node is real variable called *grid variable*. For each grid node, one grid variable is generated. The grid variable of a grid node is mainly used to formulate routing of layer-1. Second one corresponding to the route of a net is binary variable called *segment variable*. For a net, some segment variables are generated. The segment variables are mainly used to formulate routing of layer-2.

In the following, a set of the grid nodes which is called a near grid node set is explained in Sec. III.B. In Sec. III.C. and Sec. III.D. variables and constraints for routing on layer-2 and layer-1 are explained, respectively.

B. Near grid node set

In this paper, via v_i of net i is restricted to be assigned to a grid node near ball b_i .

The Manhattan distance between two vertices u and v on the routing graph is denoted by $d(u, v)$. The bound distance D_i is given to net i . It is used to restrict grid nodes to which via v_i can be placed. The set of grid nodes to which the distance from b_i are at most D_i is called *near grid node set* and is denoted by U_i . It is defined as follows: $U_i = \{u \in U \mid d(b_i, u) \leq D_i\}$. Similarly, the near extra vertex set S_i of net i is defined as $S_i = \{s \in S \mid d(b_i, s) \leq D_i\}$.

The bound distance of a net whose ball exists near mold gates is set to larger than one. The bound distance of a rest net is set to one. In Fig. 5(a), a grid node in U_i where $D_i = 2$ is shown as a gray circle.

C. Routing on layer-2

If routes are generated without intersecting the different routes on a routing graph described in Sec. II.D., then the routing pattern on layer-2 satisfies the design rule.

In this section, variables and constraints which are used to formulate a routing problem on layer-2 are explained.

The *divergence graph* $G_i = (U_i \cup S_i \cup \{b_i\}, E_i)$ of net i is a directed graph which represents the candidates of routes from ball b_i to grid nodes in near grid node set U_i , where $E_i = \{(u, v)_i \mid u \in S_i \cup \{b_i\}, v \in U_i \cup S_i, d(b_i, u) \leq d(b_i, v), \text{ either } u \text{ or } v \text{ is not an extra vertex which is connected the ball } b_i\}$. The divergence graph G_i where $D_i = 2$ is illustrated in Fig. 5(b). Note that we distinguish between arc $(u, v)_i$ in E_i and arc $(u, v)_j$ in E_j if $i \neq j$. The set of divergence graphs is denoted by $G = \bigcup_{i \in \Omega} G_i$.

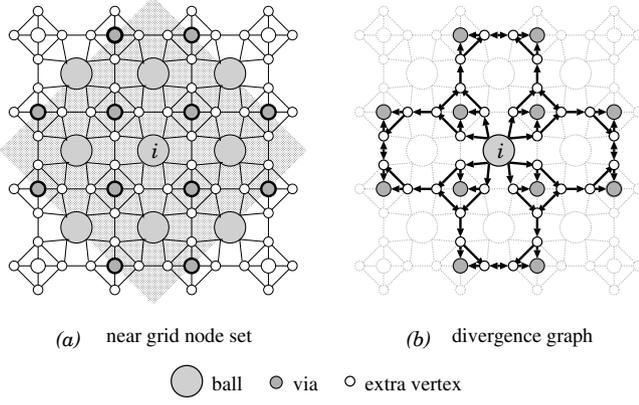


Fig. 5. Near grid node set U_i and divergence graph G_i , where $D_i = 2$

A segment variable is generated for an arc in divergence graph G_i . Let β^e be the segment variable corresponding to arc e in G_i . If the route of net i passes through arc e in G_i , then $\beta^e = 1$. Otherwise, $\beta^e = 0$. Let $l(e)$ be i if $e \in G_i$.

Let $O(v, i)$ and $I(v, i)$ be the sets of arcs in E_i which are incident from and to vertex v in G_i , respectively. Let $I(v)$ and $O(v)$ be $\bigcup_{i \in \Omega} I(v, i)$ and $\bigcup_{i \in \Omega} O(v, i)$, respectively. Let $A(v)$ be $I(v) \cup O(v)$.

In order to connect ball b_i to one grid node in U_i , three kinds of constraints should be satisfied.

$$\sum_{e \in A(b)} \beta^e = 1 \quad \forall b \in B \quad (1)$$

$$\sum_{u \in U_i} \sum_{e \in I(u, i)} \beta^e \leq 1 \quad \forall i \in \Omega \quad (2)$$

$$\sum_{e \in I(s, i)} \beta^e = \sum_{e \in O(s, i)} \beta^e \quad \forall s \in S \quad (3)$$

Constraint (1) is used to guarantee that just one arc incident from a ball is used. Constraint (2) is used to guarantee that just one grid node is used for each net. Constraint (3) is used for the flow conservation. Even if constraint (2) is not used, constraints (1) and (3) guarantee that a ball and a grid node are connected if it is possible. Therefore, constraint (2) can be pruned. In experiments, we use the constraints (1) and (3).

In order to prevent routes of different nets from intersecting each other, two kinds of constraints are used.

$$\sum_{e \in A(u)} \beta^e \leq 1 \quad \forall u \in U \quad (4)$$

$$\sum_{i \in \Omega} \sum_{e \in I(s, i)} \beta^e \leq 1 \quad \forall s \in S \quad (5)$$

Constraint (4) is used to guarantee that a grid node is used by just one net. Constraint (5) is used to forbid intersecting routes of different nets at an extra vertex.

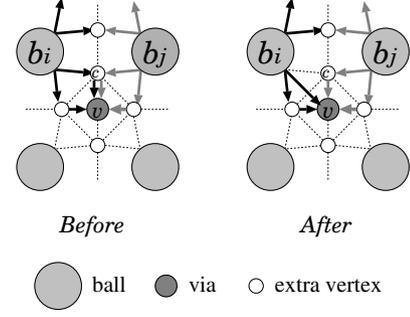


Fig. 6. Modification of a routing graph

Segment variables correspond one-to-one with arcs in the divergence graphs. If the number of arcs in the divergence graphs is reduced, then the number of segment variables is reduced and the number of constraints is accordingly reduced. Therefore, the modification of the divergence graphs to reduce the number of arcs without losing the feasible routing patterns is performed before generating constraints.

Let c be an extra vertex on the routing graph. Let b_i and b_j be balls which are adjacent to c . Let v be the grid node which is adjacent to c as shown in Fig. 6. If $A(c) = \{(b_i, c)_i, (c, v)_i\}$ or $A(c) = \{(b_i, c)_i, (c, v)_i, (b_j, c)_j, (c, v)_j\}$, then arcs $(b_i, c)_i$ and $(c, v)_i$ are replaced by $(b_i, v)_i$ as shown in Fig. 6. In former case, c can be used only by the route of net i , and b_i and v can be immediately connected without any intersections. In latter case, constraint (5) of c is guaranteed by constraint (4) of v . Therefore, $(b_i, c)_i, (b_j, c)_j, (c, v)_i$, and $(c, v)_j$ can be replaced by $(b_i, v)_i$ and $(b_j, v)_j$.

In experiments, the modification is iteratively performed until no modification is possible.

D. Routing on layer-1

A grid variable L_u is a real variable, and is generated for each grid node u . The grid variable represents that the route of a net on layer-1 whose net number is less or more than the value of the grid variable passes to the left or right of the grid node, respectively. If grid node u is connected to ball b_i on layer-2, then L_u is set to i since via v_i is assigned to grid node u . Otherwise, if u is not dummy, L_u is a value between 1 and N . Constraint (6) is used to satisfy above conditions.

$$1 + \sum_{e \in A(u)} \beta^e (l(e) - 1) \leq L_u \leq N - \sum_{e \in A(u)} \beta^e (N - l(e)) \quad \forall u \in U \quad (6)$$

If u is a dummy grid node corresponding to the left and right boundary, then $L_u = 0$ and $L_u = N+1$, respectively.

In order to satisfy the monotonic condition and the condition of the wire congestion, three kinds of constraints are used.

$$L_u \leq L_v \quad \forall (u, v) \in G_h \quad (7)$$

$$L_v - L_u - 1 \leq \text{capacity}(u, v) \quad \forall (u, v) \in G_h \quad (8)$$

$$|L_u - L_w| - 1 \leq \text{capacity}(u, w) \quad \forall (u, w) \in G_v \quad (9)$$

Constraint (7) corresponds to the monotonic condition. Under the monotonic condition, the number of routes intersecting interval (u, v) on layer-1 is $|L_v - L_u|$, and is denoted by $\text{cut}(u, v)$. Especially if (u, v) is a horizontal interval, then $\text{cut}(u, v) = L_v - L_u$ since $L_u \leq L_v$. Constraints (8) and (9) are used to guarantee that the number of routes intersecting horizontal and vertical intervals is less than or equal to the capacity.

E. Formulation of objective function

The objective function is to minimize $n + \sum_{i \in \Omega} \text{cut}_a(v_i)$. It can be calculated by $\sum_{(u,v) \in G_v} \text{cut}(u, v)$.

In order to calculate the objective function, a *temporal variable* $C_{u,v}$ and two constraints are generated for a vertical interval (u, v) .

$$L_v - L_u \leq C_{u,v} \quad (10)$$

$$L_u - L_v \leq C_{u,v} \quad (11)$$

Constraints (10) and (11) correspond to $C_{u,v} \geq \text{cut}(u, v)$. The objective function is given as follows:

$$\text{Minimize} \quad \sum_{(u,v) \in G_v} C_{u,v} \quad (12)$$

In our proposed method, a feasible routing pattern is obtained if it exists.

IV. EXPERIMENTS

We implemented the proposed method in C++ language and applied it to six test cases which are artificially generated. The program ran on a personal computer with a 2.93-GHz CPU and 2 GB of memory. CPLEX of the ILOG Co. is used as the solver of mixed integer programming.

The input data is shown in TABLE I. The name of a data in which a mold gate exists has suffix "-mold". #net is the number of nets. #row and #col are the number of rows and columns of the ball grid array, respectively. D_{max} is the maximum bound distance, and it is given as an input in our experiments. In all data, the bound distance of nets such that their balls are in the left side of ball grid array and on k -th column from the left is set to $\max\{D_{max} + 1 - k, 1\}$. Similarly, the bound distance of nets such that their balls are in the right side are set.

First, we compared the proposed method (PM) and the sequential search algorithm [7] (SSA). Experimental results are shown in TABLE II. CL1 and LL2 are $\sum_{(u,v) \in G_v} \text{cut}(u, v)$ and $\sum_{i \in \Omega} d(b_i, v_i)$, respectively. #vio is the number of intervals such that the number of routes intersecting the interval exceeds the capacity. Although the execution time of PM is longer than SSA, some data to which the violation has occurred exist in SSA. If an obtained routing pattern has some violations, then it can not

TABLE I
EXPERIMENTAL DATA

data	#net	#row	#col	D_{max}
<i>data1</i>	40	4	10	1
<i>data2</i>	72	4	18	1
<i>data3-mold</i>	41	3	17	3
<i>data4-mold</i>	41	3	17	4
<i>data5-mold</i>	41	3	17	5
<i>data6-mold</i>	46	3	19	5

be used in practical design. On the other hand, PM obtains feasible routing patterns where the total wire length is smaller than SSA.

Second, we checked the effectiveness of the modification of divergence graphs. Experimental results are shown in TABLE III. #var and #cons are the number of variables and constraints with modifications, respectively. The modification of arcs reduces the execution time by 33.4% on an average and up to 71.6%. The execution time of the modification of arcs is very small compared with the execution time of solver.

Our method obtains feasible routing patterns with shorter total wire length of layer-1 within endurable time in all data. Moreover, most of vias are placed near their balls in routing patterns, and it is likely that that many plating leads of the power supply nets can be realized.

The via assignments and the global routing of *data2* and *data6-mold* are shown in Fig. 7 and Fig. 8, respectively.

V. CONCLUSION

In this paper, we propose a routing method for 2-layer ball grid array packages that satisfies the constraints of wire congestion in which the via of a net is placed near to the ball of the net. In experiments with several data, we obtain a routing pattern that satisfies the constraints of wire congestion within a practical time by using a mixed integer programming solver.

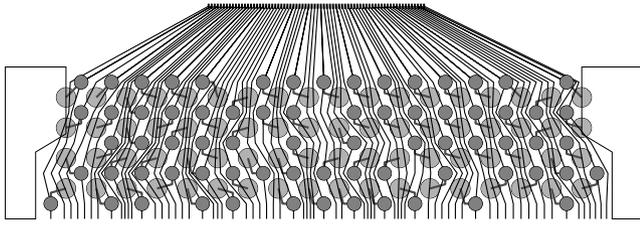
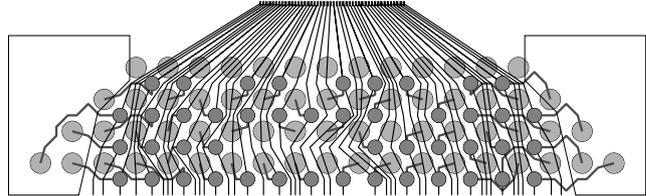
In our future works, the reduction of the number of variable and the partition of problem will be investigated.

ACKNOWLEDGMENTS

This research was partially supported by Grant-in-Aid for JSPS Fellows (19-9340).

REFERENCES

- [1] M.-F. Yu and W. W.-M. Dai: "Single-Layer Fanout Routing and Routability Analysis for Ball Grid Arrays", Proceedings of International Conference Computer-Aided Design, pp. 581–586 (1995).
- [2] S. Shibata, K. Ukai, N. Togawa, M. Sato and T. Ohtsuki: "A BGA Package Routing Algorithm on Sketch Layout System", The journal of Japan Institute for Interconnecting and Packaging Electronic Circuits, **12**, 4, pp. 241–246 (1997). (In Japanese).

Fig. 7. The output routes of *data2*Fig. 8. The output routes of *data6-mold*TABLE II
EXPERIMENTAL RESULTS

data	method	C.L1	L.L2	#vio	CPU time[sec]
<i>data1</i>	<i>PM</i>	17	40	0	0.19
	<i>SSA</i>	27	65	2	0.24
<i>data2</i>	<i>PM</i>	29	72	0	30.13
	<i>SSA</i>	31	81	1	0.48
<i>data3-mold</i>	<i>PM</i>	10	59	0	6.53
	<i>SSA</i>	12	62	1	0.2
<i>data4-mold</i>	<i>PM</i>	9	60	0	3.22
	<i>SSA</i>	12	62	1	0.2
<i>data5-mold</i>	<i>PM</i>	9	60	0	36.37
	<i>SSA</i>	12	62	1	0.2
<i>data6-mold</i>	<i>PM</i>	10	65	0	865.15
	<i>SSA</i>	13	64	0	0.7

TABLE III
EXPERIMENTAL RESULTS

data	CPU time[sec]					
	#var(Imp.)		#cons(Imp.)		without Mod.	with Mod.
<i>data3-mold</i>	1711	(10.52%)	1407	(13.94%)	6.77	6.53
<i>data4-mold</i>	1778	(10.11%)	1443	(13.64%)	6.57	3.22
<i>data5-mold</i>	1994	(9.12%)	1553	(12.8%)	39.32	36.37
<i>data6-mold</i>	2030	(11.28%)	1624	(15.24%)	3152.99	896.15

Imp. is the reduction rate of variables(constraints).

Mod. is the modification of arcs.

- [3] J.-W. Fang, C.-H. Hsu and Y.-W. Chang: "An integer linear programming based routing algorithm for flip-chip design", DAC '07: Proceedings of the 44th annual conference on Design automation, New York, NY, USA, ACM, pp. 606–611 (2007).
- [4] C.-C. Tsai, C.-M. Wang and S.-J. Chen: "NEWS: A Net-Even-Wiring System for the Routing on a Multilayer PGA Package", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, **17**, 2, pp. 182–189 (1998).
- [5] S.-S. Chen, J.-J. Chen, C.-C. Tsai and S.-J. Chen: "An Even Wiring Approach to the Ball Grid Array Package Routing", Proceedings of International Conference on Computer Design, pp. 303–306 (1999).
- [6] Y. Kubo and A. Takahashi: "Global Routing by Iterative Improvements for 2-Layer Ball Grid Array Packages", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), **25**, 4, pp. 725–733 (2006).
- [7] Y. Tomioka and A. Takahashi: "Routability driven modification method of monotonic via assignment for 2-layer ball grid array packages", Proceedings Asia and South Pacific Design Automation Conference 2008 (ASP-DAC 2008), pp. 238–243 (2008).
- [8] N. Sato, Y. Tomioka and A. Takahashi: "Global Routing Method of Plating Lead for 2-Layer BGA Packages", IEICE Technical Report (VLD2008-154), Vol. 107, pp. 61–66 (2008). (In Japanese).