Article Information

Title	MILP-Based Efficient Routing Method with Restricted Route Structure for 2-Layer Ball Grid Array Packages
Authors	Yoichi TOMIOKA, Yoshiaki KURATA, Yukihide KOHIRA and Atsushi TAKAHASHI
Citation	IEICE transactions on fundamentals of electronics, communications and computer sciences 92.12 (2009): 2998-3006.
Copyright	copyright@2009 IEICE
IEICE Transactions Online URL	https://search.ieice.org/

IEICE TRANS. FUNDAMENTALS, VOL.E92–A, NO.12 DECEMBER 2009

PAPER Special Section on VLSI Design and CAD Algorithms

MILP-Based Efficient Routing Method with Restricted Route Structure for 2-Layer Ball Grid Array Packages**

Yoichi TOMIOKA[†a)], ***Member***, **Yoshiaki KURATA**[†*], ***Nonmember***, **Yukihide KOHIRA**[††b)], ***and*** **Atsushi TAKAHASHI**[†††c)], ***Members***

SUMMARY In this paper, we propose a routing method for 2-layer ball grid array packages that generates a routing pattern satisfying a design rule. In our proposed method, the routing structure on each layer is restricted while keeping most of feasible patterns to efficiently obtain a feasible routing pattern. A routing pattern that satisfies the design rule is formulated as a mixed integer linear programming. In experiments with seven data, we obtain a routing pattern such that satisfies the design rule within a practical time by using a mixed integer linear programming solver.

key words: *ball grid array, monotonic, nearest via assignment, package routing, radiate*

1. 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.

There are several planar routing methods related to BGA package routing. A routing method for BGA packages with single layer was proposed in [2], and was improved in [3]. The methods in [2], [3] generate a netlist that achieves low wire congestion with even wire distribution and generate the corresponding routing. Routing methods for flip chips were proposed in [4], [5]. The method in [4] generates a netlist which avoids net intersections and generates the corresponding routing. The method in [5] is an exact routing method based on integer linear programming. Although these methods are related to BGA package routing, they cannot be used for multi-layer BGA as they are since they assume a single routing layer.

Routing algorithms for multi-layer BGA and pin grid array (PGA) have been proposed in [6] and [7], 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 ball-region where solder balls are placed on layer 2 is not considered. Generally, in the BGA packages, most of vias need to be assigned to the ball-region.

On the other hand, a via assignment method to the ball-region in 2-layer BGA package was proposed in [8]. 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 [9]. In the improved method, the execution time is reduced and the routing ratio of layer 2 has been improved. However, a routing pattern that satisfies the design rule is not necessarily obtained by the method in [9]. The method in [9] does not prohibit a routing pattern that has wire congestion which is higher than the limit on layer 1. Moreover, the method consumes the routing resource of layer 2 much since it sometimes assigns the via of a net to a grid which is away from the ball of the net. It is better to secure the routing resource of layer 2 as much as possible since the routing resource of layer 2 is used for the routing of the plating leads of power nets and signal nets to reduce the wire congestion of layer 1 [10].

In this paper, we propose a mixed integer linear programming (MILP) based routing method for 2-layer BGA packages that generates a routing pattern satisfying the design rule. In BGA package routing, there exist a number of routing patterns including infeasible routing patterns. However, the structure of feasible routing pattern on each layer has several properties. In order to efficiently obtain a feasible routing pattern, we restrict the structures of routes while keeping most of feasible routing patterns. In our proposed method, the routing on layer 1 and layer 2 is restricted to be monotonic and radiate, respectively. Moreover, via assignment is restricted so that all routes on layer 1 can be monotonic, and that the via of a net is placed near the ball of the net. In experiments with seven data, we obtain a routing pattern that satisfies the design rule within a practical time by using a MILP solver.

Manuscript received March 23, 2009.
Manuscript revised June 22, 2009.

[†]The authors are with the Department of Communications and Integrated Systems, Tokyo Institute of Technology, Tokyo, 152-8552 Japan.
[††]The author is with the School of Computer Science and Engineering, the University of Aizu, Aizuwakamatsu-shi, 965-8580 Japan.
[†††]The author is with the Division of Electrical, Electronic and Information Engineering, Osaka University, Suita-shi, 565-0871 Japan.
[*]Presently, with Nippon Life Insurance Company.
[**]The preliminary vesion was presented at [1].
a) E-mail: yoichi@lab.ss.titech.ac.jp
b) E-mail: kohira@u-aizu.ac.jp
c) E-mail: atsushi@si.eei.eng.osaka-u.ac.jp
DOI: 10.1587/transfun.E92.A.2998

Copyright © 2009 The Institute of Electronics, Information and Communication Engineers

2. Preliminary

2.1 2-Layer BGA Package

There exist various kinds of BGA packages. In this paper, we consider a fanout type BGA package which consists of a single chip and a package substrate with two routing layers. A model of the package is shown in Fig. 1.

A bonding finger, which is referred to as a *finger*, is connected to the chip by a bonding wire. Fingers are placed on an interior region of package substrate in fanout type, while fingers are placed on the perimeter of package substrate in fanin type. Our model is fanout type and fingers are placed on the perimeter of a rectangle enclosing the chip on layer 1. A solder ball, which is referred to as a *ball*, is an I/O terminal of the package, and it is placed in a grid array pattern on layer 2. In our model, balls are not placed under the region where fingers and chip are placed. The interior region where no ball is placed is called *free-region*, while the exterior region where balls are placed in a grid array pattern is called *ball-region*. A *mold gate* which is used to pour resin into the package is placed in a corner of the package 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 as route by using wires on each layer and vias which connect wires on different layers. The route of a net must not intersect any routes of other nets. Moreover, 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 electrical plating to protect wires. In order to enable electrical plating, a net is requested to extend its wire to the package substrate boundary on either layer. The extra connection is called a *plating lead*.

2.2 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.

The grid array pattern of balls is modeled by a ball grid array. The set of balls is denoted by $\mathcal{B}$.

There are two types of nets, called signal net and power supply net. The routing of a signal net is done within ball-region. The most part of routing of a power supply net is done within free-region, but its plating lead passes ball-region. Our concern is the routing of a signal net within ball-region, but it is requested to reserve the routing resource of layer 2 as much as possible to enable the routing of the plating leads of power supply nets.

The candidate positions of a via in ball-region are restricted since the via cannot be placed on a ball and the size of the via is relatively large even though it is smaller than the size of a ball. According to the ball grid array, the routing region is divided into unit squares surrounded by four adjacent balls. 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 nodes* corresponding to the boundary of the routing region of layer 1 are introduced. A dummy node corresponding to the left boundary of a row is placed at the intersection of the row and the left boundary of the routing region. Similarly, dummy nodes corresponding to the right boundary are generated.

The sets of grid nodes and dummy nodes are denoted by $\mathcal{N}_G$ and $\mathcal{N}_D$, respectively. The set of nodes is denoted by $\mathcal{N}$. That is, $\mathcal{N} = \mathcal{N}_G \cup \mathcal{N}_D$.

Note that a via cannot be assigned to the dummy nodes but the grid nodes. The grid array pattern on $\mathcal{N}$ is called a *via grid array*. In Fig. 2, balls are shown as big gray circles, grid nodes are shown as small white circles and dummy nodes are shown as small black circles.

Let u be a ball, a via or a node. The coordinate of u is denoted by $(x(u), y(u))$.

Let u and v be two of balls, vias and nodes. The distance between u and v is the Manhattan distance between them, and is calculated as $|x(u) - x(v)| + |y(u) - y(v)|$.

Let u_l and u_r be nodes in $\mathcal{N}$, where they are adjacent in the via grid array and u_r is on the right of u_l in a row. The interval between u_l and u_r is called *horizontal interval*, and

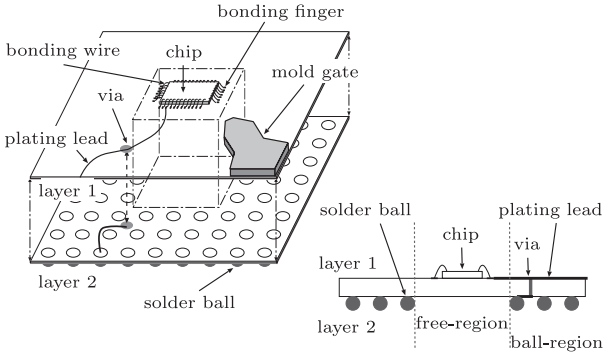


Fig. 1 A model of 2-layer BGA package.

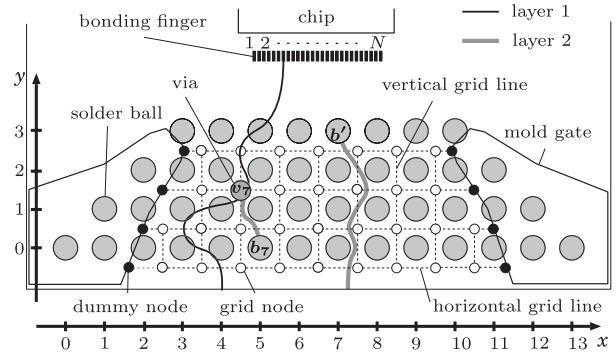


Fig. 2 A model of the bottom sector of BGA package.

denoted by an ordered pair (u_l, u_r). Similarly, let u_a and u_b be nodes in $\mathcal{N}$, where they are adjacent in the via grid array and u_a is above u_b in a column. The interval between u_b and u_a is called *vertical interval*, and denoted by an ordered pair (u_b, u_a). In our definition, a dummy node is not vertically adjacent to any other grid node in the via grid array. Let $\mathcal{I}_h$ and $\mathcal{I}_v$ be the set of horizontal intervals and vertical intervals, respectively. The set of horizontal intervals and vertical intervals is denoted by $\mathcal{I}$. The length of an interval (u, v) in $\mathcal{I}$ is the distance between u and v, and denoted by $\texttt{length}(u, v)$.

The condition of wire clearance is given depending on the packaging technology. In order to get a smaller package, the length of an interval between adjacent grid nodes is usually set to be short as long as the connection requirements are satisfied. In our model, the length of the 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 maximum number of possible wires that intersect the interval on layer 1 is larger than one if there is no obstacle. Let the length of the interval be unit length of our coordinate system. Let c be the maximum number of possible wires on layer 1 per unit length. In experiments, $c = 7$ is used. Note that the length of the interval related to a dummy node may not be unit length.

A signal net consists of a finger and a ball. The set of signal nets is denoted by Ω_S, and they are labeled according to the order of fingers from the left to the right as $\Omega_S = \{1, 2, \ldots, N\}$, where N is the number of nets. The route of a net consumes the routing resource of layer 2 much if the net has more than one via since vias are connected on layer 2. Since the routing of layer 2 is tight, we restrict that each net in Ω_S has just one via. Let b_i and v_i be the ball and the via of net i, respectively. Moreover, we restrict the plating lead of a signal net to be routed from the via of the net on layer 1. The routing resource of layer 2 is reserved for the plating leads of power supply nets as much as possible.

In Fig. 2, the route of a signal net 7 including its plating lead and the plating lead of a power supply net from ball b' are shown. The via of the signal net is illustrated by small gray circle. The routes of the nets on layer 1 and layer 2 are shown as black line and as gray thick line, respectively.

A via assignment is an assignment of the vias of nets in Ω_S to nodes. The global routing problem for a 2-layer BGA package is defined as follows:

Ball-region routing problem for 2-layer BGA
Inputs: The set of signal nets Ω_S
Outputs: A via assignment and routing for Ω_S
Objective: Minimize the total wire length of both layers
Constraint: satisfy the design rule

2.3 Monotonic Routing on Layer 1

If the route of each net on layer 1 from its finger to the package boundary intersects any horizontal grid line at most once, then the route is said to be *monotonic*.

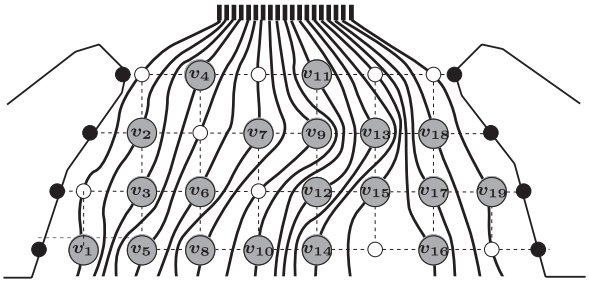


Fig. 3 Monotonic via assignment and the corresponding monotonic routing on layer 1.

(a) Feasible routing pattern. (b) Infeasible routing pattern.

Fig. 4 Feasible and infeasible routing patterns where two routes pass through a square surrounded by adjacent balls on layer 2.

It is clear that a monotonic routing is possible for the via assignment if and only if $x(v_i) < x(v_j)$ is satisfied for any pair of nets i and j $(i < j)$ such that $y(v_i) = y(v_j)$. A via assignment is said to be *monotonic* if a monotonic routing of layer 1 is possible [2], [8].

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

Since the capacity of a horizontal line is close to the number of nets, via assignment and global routing of layer 1 in actual design are usually monotonic. By definition, a monotonic route intersects a horizontal grid line of the via grid array just once. While, the number of intersections of vertical grid lines and routes increases according to route snaking. We evaluate the total wire length of layer 1 by the number of intersections.

2.4 Radiate Routing on Layer 2

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) since enough space does not exist around the via.

We use the routing graph $G = (V, E)$ proposed in [10] to represent a global routing on layer 2. The routing graph has vertices which are balls in $\mathcal{B}$, grid nodes in $\mathcal{N}_G$, and ex-

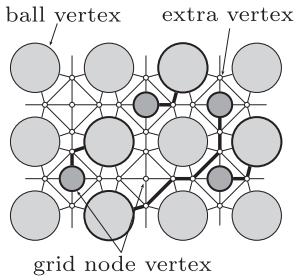


Fig. 5 Subgraph of routing graph to represent feasible routing patterns on layer 2.

tra vertices. The set of extra vertices is denoted by V_e. Note that $V = \mathcal{B} \cup \mathcal{N}_G \cup V_e$. Ball vertices and grid node vertices are embedded in a plane according to the bottom sector of BGA package. An extra vertex is placed at the center of triangle consisting of adjacent two ball vertices and grid node vertex. The ball vertices and the grid node vertex are connected via the extra vertex. Moreover, adjacent extra vertices are connected. A routing pattern satisfying the design rule is obtained by generating routes on the routing graph without intersecting each other. A subgraph of the routing graph and an example of global routing on it are shown in Fig. 5.

The Manhattan distance between vertices $u, v \in V$ is denoted by $\mathtt{dist}(u, v)$. Let $(u_1, u_2, \ldots, u_m)$ be a route of net k from ball b_k to grid node u_m on routing graph G, where $u_i \in V$ $(1 \le i \le m)$, $u_1 = b_k$ and $u_m \in \mathcal{N}_G$. Route $(u_1, u_2, \ldots, u_m)$ is said to be *radiate* if $\mathtt{dist}(b_k, u_i) \le \mathtt{dist}(b_k, u_j)$ for any u_i and u_j $(1 \le i < j \le m)$.

The via and ball of a net is better to be connected by a route with shorter wire length to save the routing resource on layer 2. A radiate route connects the via and ball of a net with shorter wire. In our method, the routing on layer 2 is restricted so that all routes of nets in Ω_S on layer 2 are radiate.

3. Mixed Integer Programming Formulation

3.1 Outline of the Method

In our method, in order to effectively obtain a feasible routing pattern, via assignment and routing for each layer are restricted so that the variety of the routing patterns is reduced while keeping most of feasible routing patterns. Via assignment is restricted to be monotonic via assignment for layer 1, and to be nearest via assignment for layer 2 such that the via of each net is placed to grid nodes which are neighborhood of the ball of a net. Moreover, routing on layer 1 is restricted to be monotonic, and routing on layer 2 is restricted to be radiate. In order to obtain a routing pattern with total wire length on both layers small, the total wire length on layer 1 is minimized while keeping the total wire length on layer 2 small. The routing problem with the restrictions is formulated as a mixed integer linear 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 our proposed method, a feasible routing pattern is obtained if it exists.

In the following, in Sect. 3.2 and Sect. 3.3, the formulation of routing on layer 1 and our objective function are explained, respectively. In Sect. 3.4 and Sect. 3.5, nearest via assignment and the formulation of routing on layer 2 are explained, respectively.

3.2 Formulation of Monotonic Routing on Layer 1

In routing on layer 1, a via assignment and the corresponding routing pattern satisfying the design rule, are required.

If a monotonic via assignment is given, which side of the via of net i each route passes is determined in the corresponding monotonic routing. In other words, some net numbers are assigned to some of grid nodes, the corresponding monotonic routing pattern is determined. We represent a monotonic routing pattern with real variables assigned to grid nodes.

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

The via of net i is placed to just one of grid nodes, and the grid variable of the grid node to which the via is placed is set to i. A grid variable is set to an appropriate value by using segment variables. The formulation is given in constraint (11) in Sect. 3.5. Moreover, let $\mathtt{via}(u)$ be whether a via is placed to grid node u or not. That is, if a via is placed to u, then $\mathtt{via}(u) = 1$. Otherwise, $\mathtt{via}(u) = 0$. $\mathtt{via}(u)$ is set to an appropriate value by using segment variables. The formulation is given in Sect. 3.5.

In order to satisfy the monotonic condition, constraint (1) is used.

$$L_u \le L_v \quad \forall (u, v) \in \mathcal{I}_h \tag{1}$$

Under the monotonic condition, the number of intersections of routes and interval (u, v) on layer 1 is $|L_v - L_u|$, and is denoted by $\mathtt{cut}(u, v)$. Let (u_i, u_{i+1}) $(1 \le i \le m-1)$ be a series of intervals, where vias are placed to u_1 and u_m and no via is placed to the others. The number of intersections of routes and the interval between u_1 and u_m can be calculated by $\sum_{1 \le i \le m-1} \mathtt{cut}(u_i, u_{i+1})$. Note that the route passing through the via placed to u_m is counted in it. Especially if (u, v) is a horizontal interval, then $\mathtt{cut}(u, v) = L_v - L_u$ since $L_u \le L_v$.

The maximum number of possible routes intersecting interval (u, v) on layer 1 is denoted by $\texttt{capacity}(u, v)$. If no via is placed at the both ends of the interval, then $\texttt{capacity}(u, v) = c{\cdot}\texttt{length}(u, v)$. Otherwise, it is decreased according to the radius of a via. Let r be the radius of via. In the case that a via is placed to u, the capacity is decreased by cr. While, in the case that a via is placed to v, the capacity is decreased by $cr - 1$ since the route passing through the via is counted. Formally, $\texttt{capacity}(u, v)$ is defined as follows:

$$\begin{aligned}&\texttt{capacity}(u, v)\\ &\quad = c \cdot \texttt{length}(u, v) - cr \cdot \texttt{via}(u) - (cr - 1) \cdot \texttt{via}(v).\end{aligned}$$

In experiments, $cr = 1.5$ is used.

Constraints (2) and (3) are used to guarantee that the number of routes intersecting horizontal and vertical intervals is less than or equal to the capacity.

$$L_v - L_u \leq \texttt{capacity}(u, v) \quad \forall (u, v) \in \mathcal{I}_h \tag{2}$$

$$|L_u - L_v| \leq \texttt{capacity}(u, v) \quad \forall (u, v) \in \mathcal{I}_v \tag{3}$$

3.3 Formulation of Objective Function

Our objective is to reduce the total wire length on both layers. The total wire length on layer 1 is evaluated by the number of intersections of vertical grid lines and routes. In order to calculate the objective function, a temporal variable $C_{u,v}$ and two constraints are generated for a vertical interval (u, v).

$$L_v - L_u - \texttt{via}(v) \leq C_{u,v} \quad \forall (u, v) \in \mathcal{I}_v \tag{4}$$

$$L_u - L_v - \texttt{via}(v) \leq C_{u,v} \quad \forall (u, v) \in \mathcal{I}_v \tag{5}$$

Constraints (4) and (5) correspond to $C_{u,v} \geq \texttt{cut}(u, v) - \texttt{via}(v)$. The objective function is given as follows:

$$\textit{Minimize} \quad \sum_{(u,v)\in\mathcal{I}_v} C_{u,v} \tag{6}$$

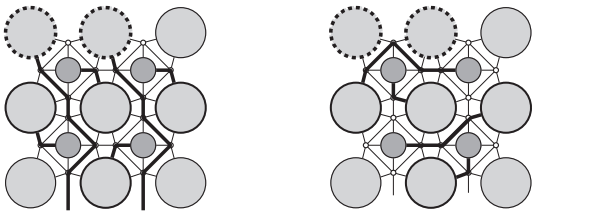
It is equivalent to minimize the number of intersections of routes and vertical grid lines.

On the other hand, the total wire length on layer 2 is kept to be small by nearest via assignment and radiate routing. In the following sections, the formulation of nearest via assignment and radiate routing are explained.

3.4 Nearest via Assignment

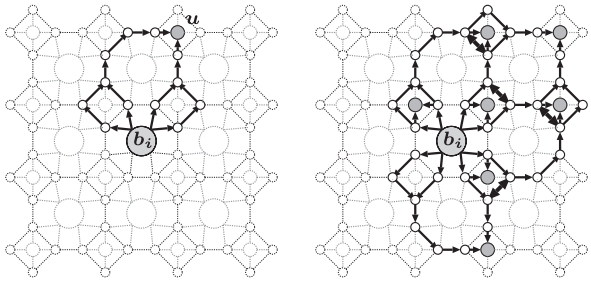
The long route between the ball and the via of a signal net blocks the plating leads of power supply nets. For example, in Fig. 6, let balls indicated by thick dotted line be the balls of power supply nets. In Fig. 6(a), each via is placed to a nearest grid node of its ball, and two plating leads of power supply nets are generated. On the other hand, in Fig. 6(b), two vias are not placed to a nearest grid node of their balls, and no plating leads can be generated.

In our method, a via is placed to the grid node near its ball in order to keep the routability of plating leads of power supply nets.



(a) Nearest via assignment. (b) Non nearest via assignment.

Fig. 6 Blocking plating leads due to non-nearest via assignment.



(a) $|\mathcal{N}_i| = 1$ (b) $|\mathcal{N}_i| = 7$

Fig. 7 Examples of radiate graph of net i.

The via of a net whose ball is near a mold gate is restricted to be assigned to the grid nodes near the mold gate. A via of the rest nets is restricted to be assigned to one of the nearest grid nodes of its ball. The candidate set of net i is a set of grid nodes to which via v_i can be assigned, and is denoted by $\mathcal{N}_i$.

3.5 Formulation of Radiate Routing on Layer 2

If routes are generated on the routing graph described in Sect. 2.4 without intersecting each other, then the obtained 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.

Let $D = (V, A)$ be the directed graph obtained from a routing graph $G = (V, E)$ by replacing each edge with two arcs of opposite direction. Let $A_i(u)$ be the set of arcs in A each of which is on a radiate route from ball b_i to grid node u. The radiate graph $D_i = (V_i, A_i)$ of net i is a subgraph of D where $V_i = \mathcal{N}_i \cup V_e \cup \{b_i\}$ and $A_i = \bigcup_{u\in\mathcal{N}_i} A_i(u)$. The radiate graph D_i represents the set of radiate routes from ball b_i to a grid node in N_i. Note that we distinguish between arc $(s, t)_i$ in A_i and arc $(s, t)_j$ in A_j if $i \neq j$. Examples of radiate graph D_i in the cases that $|\mathcal{N}_i| = 1$ and $|\mathcal{N}_i| = 7$ are illustrated in Figs. 7(a) and (b), respectively.

A segment variable is generated for an arc in radiate graph D_i. Let β^e be the segment variable corresponding to arc e in A_i. If the route of net i passes through arc e in A_i, then $\beta^e = 1$. Otherwise, $\beta^e = 0$.

The owner of arc $e \in A_i$ is net number such that the radiate graph of the net includes the arc, is denoted by $l(e)$. That is, $l(e) = i$ if $e \in A_i$.

Let $O(v, i)$ and $I(v, i)$ be the sets of arcs in A_i which are incident from and to vertex v in D_i, respectively. Let $I(v)$ and $O(v)$ be $\bigcup_{i\in\Omega_S} I(v, i)$ and $\bigcup_{i\in\Omega_S} O(v, i)$, respectively.

In order to connect ball b_i to just one grid node in $\mathcal{N}_i$, three kinds of constraints should be satisfied.

$$\sum_{e\in O(b_i,i)} \beta^e = 1 \qquad \forall i \in \Omega_S \tag{7}$$

$$\sum_{u\in\mathcal{N}_i} \sum_{e\in I(u,i)} \beta^e = 1 \qquad \forall i \in \Omega_S \tag{8}$$

$$\sum_{e\in I(v,i)} \beta^e = \sum_{e\in O(v,i)} \beta^e \quad \forall v \in V_e, \forall i \in \Omega_S \tag{9}$$

Constraint (7) is used to guarantee that just one arc incident from a ball is used. Constraint (8) is used to guarantee that just one grid node is used for each net. Constraint (9) is used for the flow conservation. The source of radiate graph D_i is ball b_i, and the sinks of D_i are only grid nodes in $\mathcal{N}_i$. Therefore, even if constraint (8) is not used, constraints (7) and (9) guarantee that ball b_i and a grid node in $\mathcal{N}_i$ are connected if it is possible. Therefore, constraint (8) can be pruned. In experiments, we use the constraints (7) and (9).

In order to forbid intersections between any routes, each vertex must be used by at most one net. Constraint (10) is used to forbid intersecting routes of different nets at extra vertices and grid nodes since it restricts each vertex so that at most one arc incident to the vertex can be used.

$$\sum_{e\in I(v)} \beta^e \le 1 \qquad \forall v \in \mathcal{N}_G \cup V_e \tag{10}$$

Let u be a grid node in $\mathcal{N}_G$. Again, L_u is the grid variable of grid node u. If u is connected to ball b_i on layer 2, then L_u is set to i since via v_i is assigned to u. Otherwise, L_u is a real value between 1 and N. Constraint (11) is used to satisfy above conditions.

$$1 + \sum_{e\in I(u)} \beta^e (l(e) - 1) \le L_u \le N - \sum_{e\in I(u)} \beta^e (N - l(e)) \qquad \forall u \in \mathcal{N}_G \tag{11}$$

Moreover, $\texttt{via}(u)$ is whether a via is placed to grid node u or not, and formulated with segment variables as follows:

$$\texttt{via}(u) = \sum_{e\in I(u)} \beta^e$$

It is substituted in constraints explained in Sect. 3.2 and Sect. 3.3.

4. Arc Reduction Technique

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

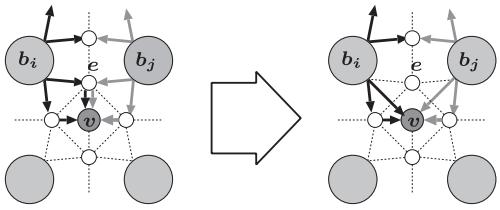


Fig. 8 Arc replacements related to extra node e.

4.1 Candidate Reduction

By focusing the design rule on layer 1, redundant grid nodes in $\mathcal{N}_i$ can be removed without losing any feasible routing patterns. Accordingly, arcs in D_i can be reduced since the redundant candidates of radiate routes on layer 2 for net i are removed.

$\texttt{lower}(u)$ and $\texttt{upper}(u)$ are a lower bound and an upper bound of net number such that the via of the net can be placed to grid node u under the wire clearance constraint, respectively. In our method, grid node u is removed from $\mathcal{N}_i$ before generating radiate graph D_i if $i < \texttt{lower}(u)$ or $\texttt{upper}(u) < i$. The modification is called *Candidate Reduction (CR)*.

Again, let r be the radius of a via. c is the maximum number of possible wires on layer 1 per unit length. The capacity of an interval is the maximum number of possible routes intersecting the interval.

The capacity of a row is decreased by $2cr - 1$ for each via placed on the row. It must be more than or equal to the number of nets since routes of all nets pass through each row on layer 1. The maximum number of vias placed on a row to keep the capacity of the row enough is calculated by $\lfloor \frac{\texttt{cdist}(d_l,d_r)-N}{(2cr-1)} \rfloor$, where d_l and d_r are left and right dummy nodes on the row, respectively. While, the number of grid nodes is limited, and a lower bound of the number of vias placed on a row is defined by using the maximum number of vias placed on each row. Similarly, a lower bound of the number of vias placed on an interval on a row is defined. Let u be a grid node on the row, and d_l be the left dummy node on the row. An upper bound of the capacity of the interval between u and d_l, denoted by $\texttt{ucap}(u, d_l)$, is defined by using the lower bound of the number of vias placed on the interval. If the via of a net whose net number is more than $\texttt{ucap}(d_l, u)$ is placed at u, then the design rule is violated at the interval between u and d_l. Therefore, $\texttt{upper}(u)$ is set to $\texttt{ucap}(d_l, u)$.

4.2 Arc Replacement

Let e be any extra vertex on the routing graph, and let s and t be any adjacent vertices of e. If $I(e)$ consists of only arcs incident from vertex s, then a pair of arcs $(s, e)_i$ and $(e, t)_i$ for any net i can be replaced an arc $(s, t)_i$ since constraint (10) of e is equivalent to constraint (10) of s. Similarly, if $O(e)$ consists of only arcs incident to vertex t, then a pair of arcs $(s, e)_i$ and $(e, t)_i$ for any net i can be replaced an arc $(s, t)_i$.

In Fig. 8 $I(e) = \{(b_i, e)_i, (b_j, e)_j\}$ and $O(e) = \{(e, v)_i, (e, v)_j\}$. $(b_i, e)_i, (b_j, e)_j, (e, v)_i$, and $(e, v)_j$ can be replaced by $(b_i, v)_i$ and $(b_j, v)_j$ since constraint (10) of e is guaranteed by constraint (10) of v.

In experiments, the replacement is iteratively performed until no modification is possible. The modification is called *Arc Replacement (AR)*.

Table 1 Input data.

data	$\|\Omega_S\|$	$\|\mathcal{B}_P\|$	#row	#column
data1	45	2	5	16
data2	51	4	5	18
data3	61	6	5	20
data4	70	8	5	22
data5	78	10	5	24
data6	86	11	5	26
data7	94	12	5	28

Table 2 The comparison of PM with SSA.

data		wire length				vio.		time
		L1	L2	SUM	([%])	V	U	[h:m:s]
data1	PM	355.2	76.9	432.1	(99.8%)	0	0	00:00:02
	SSA	358.5	74.3	432.9	(100.0%)	2	0	00:00:03
data2	PM	402.0	95.7	497.7	(101.7%)	0	0	00:00:10
	SSA	393.6	95.6	489.2	(100.0%)	3	3	00:00:00
data3	PM	480.2	116.0	596.2	(99.2%)	0	0	00:00:25
	SSA	484.4	116.5	600.9	(100.0%)	3	0	00:00:17
data4	PM	572.6	141.6	714.3	(96.6%)	0	0	00:02:32
	SSA	564.0	173.2	737.2	(100.0%)	4	8	00:00:06
data5	PM	619.3	152.4	771.7	(96.9%)	0	0	00:16:10
	SSA	621.2	175.4	796.6	(100.0%)	0	2	00:00:02
data6	PM	691.1	178.3	869.4	(102.0%)	0	0	00:04:55
	SSA	678.6	173.5	852.1	(100.0%)	0	1	00:00:19
data7	PM	754.1	183.4	937.5	(98.2%)	0	0	11:02:37
	SSA	744.3	210.3	954.6	(100.0%)	2	4	00:00:06

5. Experiments

We implemented the proposed method in C++ language and applied it to seven 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 11.0.0 of the ILOG Co. [11] is used as the solver of mixed integer linear programming.

The input data is shown in Table 1. $|\Omega_S|$ and $|\mathcal{B}_P|$ are the number of signal nets and power supply nets, respectively. #row and #column are the number of rows and columns of the ball grid array, respectively.

The candidate set of net i is set to

$$\mathcal{N}_i = \begin{cases} \{u \in \mathcal{N}_G \mid x(u) \le l + 2\} & \text{if } x(b_i) \le l + 2, \\ \{u \in \mathcal{N}_G \mid x(u) \ge r - 2\} & \text{if } x(b_i) \ge r - 2, \\ \{u \in \mathcal{N}_G \mid d(b_i, u) = 1\} & \text{otherwise,} \end{cases}$$

where the x-coordinates of the lower-left grid node and the lower-right grid node are l and r, respectively.

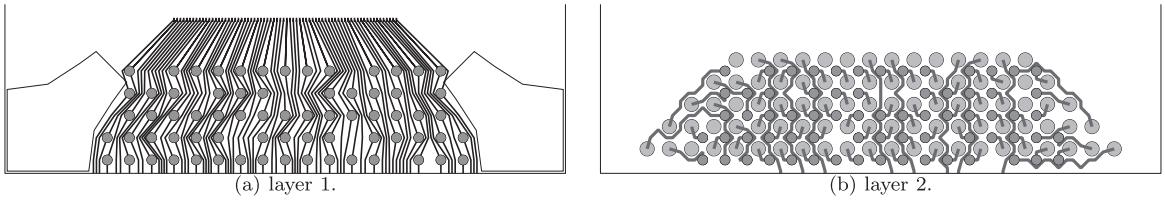
First, we compared the proposed method (PM) and the sequential search algorithm [9] (SSA). In both of the methods, the plating leads of power supply nets are routed by ripup and reroute technique on the routing graph described in Sect. 2.4 after generating via assignment and routing of nets in Ω_S.

Experimental results are shown in Table 2. L1 and L2 are the total wire length on layer 1 and on layer 2, respectively. V is the number of intervals such that the wire clearance constraint is violated. U is the number of unconnected nets in signal nets and power supply nets. Although the execution time of PM is longer than SSA, some data where violations of wire clearance constraint and/or unconnected nets have occurred exist in SSA. If an obtained routing pattern has some violations, then it cannot be immediately used in practical design. On the other hand, PM obtains feasible

Table 3 The effectiveness of each arc reduction technique.

Data	CR+AR (PM)			CR			AR		
	#var	#cons	time	#var	#cons	time	#var	#cons	time
data1	2221 (64.9%)	1578 (69.4%)	(15.0%)	(73.2%)	(83.7%)	(18.3%)	(92.0%)	(86.5%)	(105.3%)
data2	2612 (78.7%)	1798 (77.0%)	(20.0%)	(90.8%)	(96.3%)	(28.4%)	(89.3%)	(83.1%)	(35.6%)
data3	2818 (76.2%)	1967 (74.2%)	(11.5%)	(90.2%)	(95.6%)	(10.5%)	(87.5%)	(81.0%)	(54.3%)
data4	2943 (74.4%)	2090 (72.3%)	(10.4%)	(90.3%)	(95.8%)	(29.5%)	(85.8%)	(79.1%)	(59.2%)
data5	2931 (72.3%)	2160 (70.6%)	(2.9%)	(89.7%)	(95.6%)	(3.0%)	(84.0%)	(77.2%)	(23.8%)
data6	3093 (72.4%)	2300 (70.1%)	(8.5%)	(91.8%)	(97.1%)	(14.9%)	(81.7%)	(74.5%)	(112.8%)
data7	3086 (71.2%)	2381 (69.4%)	(17.4%)	(91.4%)	(96.9%)	(14.4%)	(80.4%)	(73.4%)	(126.4%)
ave.	(72.9%)	(71.8%)	(12.2%)	(88.2%)	(94.4%)	(17.0%)	(85.8%)	(79.3%)	(73.9%)

The percentages of #var, #cons, and exe. time of PM without CR and AR are shown.



(a) layer 1. (b) layer 2.

Fig. 9 The feasible routing pattern of *data4* obtained by PM.

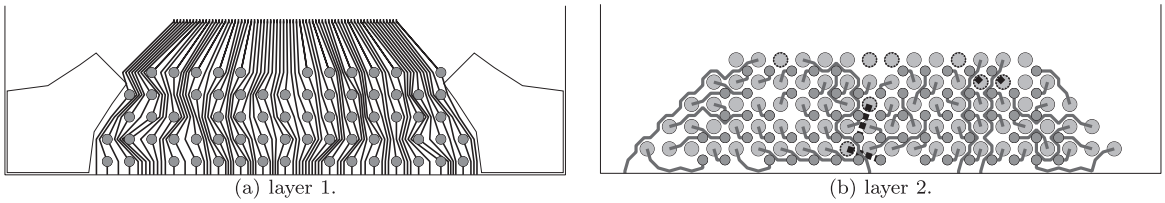

(a) layer 1. (b) layer 2.

Fig. 10 The infeasible routing pattern of *data4* obtained by SSA. (Unconnected nets are indicated by dotted lines).

routing patterns where the total wire length is as small as that of SSA. The execution time of *CR* and *AR* is within 1 sec. in all data, and it is included in Table 2.

Second, we checked the effectiveness of the modifications of radiate graphs by *CR* and *AR*. Experimental results are shown in Table 3. #var and #cons are the number of variables and constraints with modifications, respectively. Compared to the case that neither *CR* nor *AR* is used, *CR* and *AR* improve the execution time to 17.0% and 73.9% on average, respectively. Moreover, the execution time is improved to 12.2% on average by using *CR* and *AR* at the same time.

The via assignments and the global routing of *data4* obtained by PM and SSA, are shown in Fig. 9 and Fig. 10, respectively. In Fig. 10, unconnected signal nets and unconnected balls of power supply nets are indicated by dotted lines.

6. Conclusion

In this paper, we propose a routing method for 2-layer ball grid array packages satisfying design rule, 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 design rule within a practical time by using a mixed integer linear 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] Y. Kurata, Y. Tomioka, Y. Kohira, and A. Takahashi, "A routing method based on nearest via assignment for 2-layer ball grid array packages," Proc. 15th Workshop on Synthesis And System Integration of Mixed Information Technologies, pp.307–312, March 2009.
[2] M.F. Yu and W.W.M. Dai, "Single-layer fanout routing and routability analysis for ball grid arrays," Proc. International Conference Computer-Aided Design, pp.581–586, 1995.
[3] S. Shibata, K. Ukai, N. Togawa, M. Sato, and T. Ohtsuki, "A BGA package routing algorithm on sketch layout system," J. Japan Institute for Interconnecting and Packaging Electronic Circuits, vol.12, no.4, pp.241–246, 1997.
[4] J.T. Yan and Z.W. Chen, "IO connection assignment and RDL routing for flip-chip designs," Proc. 2009 Conference on Asia and South Pacific Design Automation Conference 2009, pp.588–593, IEEE Press, Piscataway, NJ, USA, 2009.
[5] J.W. Fang, C.H. Hsu, and Y.W. Chang, "An integer linear programming based routing algorithm for flip-chip design," Proc. 44th Annual Conference on Design Automation, pp.606–611, ACM, New York, NY, USA, 2007.
[6] 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 Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.17, no.2, pp.182–189, 1998.
[7] S.S. Chen, J.J. Chen, C.C. Tsai, and S.J. Chen, "An even wiring approach to the ball grid array package routing," Proc. International Conference on Computer Design, pp.303–306, 1999.
[8] Y. Kubo and A. Takahashi, "Global routing by iterative improvements for 2-layer ball grid array packages," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.25, no.4, pp.725–733, 2006.
[9] Y. Tomioka and A. Takahashi, "Routability driven via assignment method for 2-layer ball grid array packages," IEICE Trans. Fundamentals, vol.E92-A, no.6, pp.1433–1441, June 2009.
[10] N. Sato, Y. Tomioka, and A. Takahashi, "Global routing method of plating lead for 2-layer BGA packages," IEICE Technical Report, VLD2008-154, 2008.
[11] ILOG, CPLEX, http://www.ilog.com/

Yoichi Tomioka received his B.E., M.E., and D.E. degrees from Tokyo Institute of Technology, Tokyo, Japan, in 2005, 2006, and 2009, respectively. He is currently a researcher of Department of Communications and Integrated Systems in Tokyo Institute of Technology. His research interests are in VLSI package design automation and combinational algorithms. He is a member of IPSJ.

Yoshiaki Kurata received his B.E. and M.E. degrees from Tokyo Institute of Technology, Tokyo, Japan, in 2007 and 2009, respectively. He is currently with Nippon Life Insurance Company. His research interests are in VLSI package design automation and combinational algorithms.

Yukihide Kohira received his B.E., M.E., and D.E. degrees from Tokyo Institute of Technology, Tokyo, Japan, in 2003, 2005, and 2007, respectively. He had been a researcher of Department of Communications and Integrated Systems in Tokyo Institute of Technology from 2007 to 2009. He is currently with the School of Computer Science and Engineering, the University of Aizu, as an assistant professor since 2009. His research interests are in VLSI design automation and combinational algorithms. He is a member of IEEE and IPSJ.

Atsushi Takahashi received his B.E., M.E., and D.E. degrees in electrical and electronic engineering from Tokyo Institute of Technology, Tokyo, Japan, in 1989, 1991, and 1996, respectively. He had been with the Tokyo Institute of Technology as a research associate from 1991 to 1997 and had been an associate professor from 1997 to 2009. He visited University of California, Los Angeles, U.S.A., as a visiting scholar from 2001 to 2002. He is currently with Division of Electrical, Electronic and Information Engineering, Graduate School of Engineering, Osaka University, as an associate professor since 2009. His research interests are in VLSI layout design and combinational algorithms. He is a member of IEEE and IPSJ.