Single-Layer Trunk Routing Using 45-Degree Lines within Critical Areas for PCB Routing

Kyosuke SHINODA[†] Yukihide KOHIRA[††] Atsushi TAKAHASHI[†††]

[†] Tokyo Institute of Technology
Dept. of Communications and Integrated Systems
2–12–1–S3–58 Ookayama, Meguro–ku, Tokyo 152–8552, Japan
shinoda@lab.ss.titech.ac.jp

[††] The University of Aizu
School of Computer Science and Engineering,
Tsuruga, Ikki-machi, Aizu-Wakamatsu, Fukushima 965-8580, Japan
kohira@u-aizu.ac.jp

[†††] Osaka University
Division of Electrical, Electronic and Information Engineering
2–1 Yamadaoka, Suita, Osaka 565–0871, Japan
atsushi@si.eei.eng.osaka-u.ac.jp

Abstract— In recent Printed Circuit Boards (PCB), the design size and density have increased, and the improvement of routing tools for PCB is required. Although there are several routing tools that generate high-quality global routings when only horizontal and vertical segments are used, PCB designers are not satisfied with these tools because high-density PCBs require segments that have arbitrary directions. In this paper, we propose a routing method that maintains the advantages of tools that use horizontal and vertical segments only, while handling higher density designs by using 45-degree segments to locally relax the routing density.

I. Introduction

Printed circuit board routing is necessary in order to realize connections between elements, and to achieve the specifications related to delay and noise, for example. Since the quality of the routing obtained by automatic routing tools for PCBs is inferior to the routing achieved by designers, high-density PCBs are still designed by hand. However, the number of nets on a PCB has increased due to the emergence of BGA packages, which have numerous I/O pins, and the design scale approaches the limit of manual design. As the design scale increases, the time required to complete PCB design by hand increases. Therefore, the quality enhancement of automatic PCB routing tools is needed in order to shorten the design period.

In a typical PCB routing design flow, first, in the global routing phase, the global structure of the route is determined for each net group, which consists of related nets as the result of layer assignment, area assignment, etc. Then, in the detailed routing phase, the detailed structure of route of each net group is determined within the assigned routing area.

In the detailed routing phase, the detailed routing problem of a net group is usually divided into two types of subproblems, namely, escape routing and river routing.

Escape routing is primarily used to find an escape route for each net from highly congested areas where I/O pins are closely arranged [1–5]. On the other hand, river routing is used to connect the terminals of each net, which are arranged on the boundary of a less congested routing area of a single layer, as well as to achieve various specifications.

A typical river routing problem is modeled as a trunk routing problem in which the routing area can be bifurcated such that each area contains one terminal of a net [6]. The trunk routing problem is a key problem because this problem corresponds to a subproblem for a typical net group that consists of nets connecting two modules. The formulation of the trunk routing problem is simple, but an inferior routing pattern, as compared to manual design, is often obtained by current automatic PCB tools. The quality improvement in the trunk routing problem is needed in order to improve the quality of automatic PCB routing tools.

In the trunk routing problem, a routing pattern should achieve various specifications and realize connections [6–12]. There are several routing tools that generate high-quality global routings when only horizontal and vertical segments are used. However, non-orthogonal segments are often essential in order to realize the connection requirements in a trunk routing problem. Tools that handle only horizontal and vertical segments cannot be used for actual PCB routing design.

There exist several routing methods that can handle non-orthogonal segments in multi-layer VLSI routing [13–15]. However, in the routing methods proposed in [13–15], non-orthogonal segments are primarily used to shorten the wire length. These methods do not effectively use non-orthogonal segments to realize connection requirements when the routing resources are limited.

In the trunk routing problem, a feasible routing pattern that realizes the connection requirement will be obtained if the route of a net is determined along the boundary of the routing area, including non-orthogonal segments, so that the design rule is satisfied. However, the obtained routing pattern will contain a number of unnecessary bends, and the total length will be large. The quality might be improved by post-processing, but this would require too much time to obtain a satisfactory routing pattern if the initial routing pattern is too poor.

In this work, we propose a routing method for a trunk routing problem in which the connection requirements are realized in one routing layer using 45-degree segments as well as horizontal and vertical segments. In the proposed routing method, 45-degree segments are used only when they are essential to realize the connection requirements.

In the trunk routing problem, the routing area is divided into critical zones and non-critical zones, where critical zones require 45-degree segments in order to realize the connection requirements, but non-critical zones do not. The proposed method identifies the critical zones by flow and determines the routing pattern in these zones using 45-degree segments as well as horizontal and vertical segments without loss of connectivity. The connection requirements are then realized by generating a routing pattern of the remaining zones, i.e., non-critical zones, using only horizontal and vertical segments.

The proposed method efficiently generates a feasible routing pattern. Although the achievement of various specifications is beyond the scope of this paper, the proposed method will be helpful for improving various indices. Although the 45-degree segments in the critical zone are invariant in most cases, the segments in a non-critical zone are not invariant. The routing pattern in a non-critical zone has large flexibility. The improvement of various indices will be achieved by applying routing tools that effectively into account take various indices in non-critical zones. The routing tools that handle horizontal and vertical segments only are applicable because 45-degree segments are not required. Of course, the routing tools that handle 45-degree segments can be used. These routing tools can be used to realize various specifications without worrying about connectivity issues.

II. Preliminaries

In this paper, a single-layer routing area that contains several obstacles is considered. In the problem formulation, the width of each wire is considered to be 0, and the minimum distance between adjacent wires is set to one unit length. In addition, the minimum distance between a wire and an obstacle is set to one unit length.

In the routing area, the tracks on which wires run are defined. All wires run only on these tracks. No wire runs on a part of a track on which an obstacle exists. Three types of tracks, namely, horizontal tracks (H tracks), vertical tracks (V tracks), and ±45-degree tracks (X tracks) are defined. Both H tracks and V tracks are defined as having a separation of one unit length and are defined over the entire routing area. The intersection of an H track and a V track is referred to as a grid-point. Both H tracks and V tracks are referred to as HV tracks, and X tracks are defined locally with the same separation, which is at least one unit length. In other words, an HV track always ends at the boundary of the routing area but an X track usually ends inside the routing area. The X tracks are defined in detail in Section III.

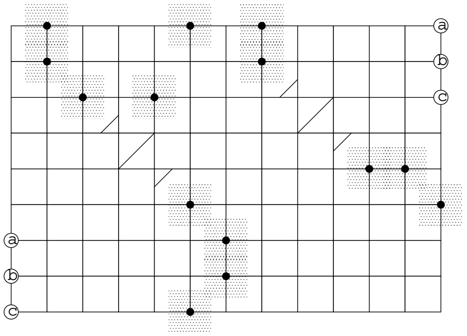

Fig. 1. Single-layer multiple-nets routing problem

An obstacle is assigned a set of grid-points such that no wire can separate the grid-points without violating the design rule. A grid-point is called a *off-point* if it is contained in an obstacle and is otherwise called an *on-point.* A net consists of two terminals referred to as a source terminal and a sink terminal.

An example of a routing area is shown in Fig. 1. In this example, H tracks, V tracks, and X tracks are represented by the horizontal, vertical, and 45-degree lines, respectively. An off-point that corresponds to an obstacle is represented by a black dot. A wire cannot use a track if the track is in a gray region that surrounds a black dot. A terminal of a net is represented by a white circle in which the name of the net is written.

Flow graphs are defined to help analyze a given routing problem. The flow graph G with edge capacity is defined as follows. The vertex set of G consists of the primary source, the primary sink, and vertices that correspond to on-points. For each on-point in the routing area, two vertices called the *in-vertex* and the *out-vertex* are included in the vertex set of G. The edge set of G consists of the following four types of directed edges:

type-1) a directed edge from the in-vertex of an on-point to the out-vertex of the on-point;

type-2) a directed edge from the out-vertex of an on-point to the in-vertex of an adjacent on-point;

type-3) a directed edge from the primary source to the in-vertex of the on-point to which a source terminal is assigned;

type-4) a directed edge from the out-vertex of the on-point to which the sink terminal is assigned to the primary sink.

The capacity of an edge is one if the edge is a type-1 edge, and is infinite otherwise. Unit flows in G do not intersect because the capacity of each type-1 edge is set to one. For example, the flow graph with edge capacity corresponding to the routing area shown in Fig. 1 is shown in Fig. 2.

A partition (V_s, V_t) of the edge set of V is called a cut if V_s and V_t contain the primary source and the primary sink, respectively. The capacity of the cut (V_s, V_t) is the

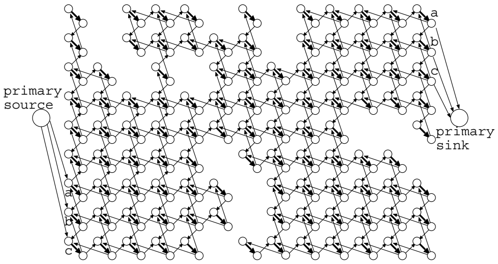

Fig. 2. Flow graph with edge capacities

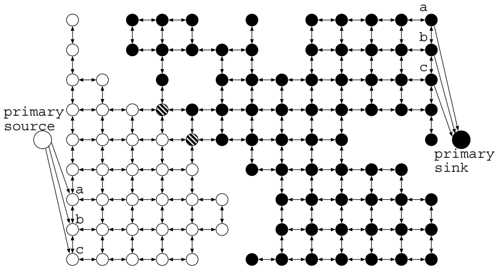

Fig. 3. Flow graph with vertex capacities

sum of the capacities of the edges that connect from a vertex in V_s to a vertex in V_t.

The flow graph G' with vertex capacity that corresponds to G is defined in order to simplify the graph drawing. The flow graph G' is obtained from G by contracting type-1 edges. The capacity of a vertex is one if the vertex corresponds to a type-1 edge, and is infinite otherwise. A cut (V_s, V_t) of G, having a finite capacity is represented by the set of vertices of G' that correspond to the edges of G that connect from a vertex in V_s to a vertex in V_t. The set of such vertices of G' is called a cut of G'. For example, the flow graph with vertex capacity corresponding to the flow graph with edge capacity shown in Fig. 2 is shown in Fig. 3.

The *trunk routing topology condition* is defined in [6] as follows:

Trunk routing topology condition

1. Each net consists of two terminals, namely, a source terminal and a sink terminal.
2. All terminals of nets are set on the boundary of the routing area.
3. The sequence of terminals along the boundary consists of the sequence S of source terminals of all nets and the sequence T of sink terminals of all nets, where T is the reverse of S, and vice versa.

A routing problem that satisfies this condition is called a trunk routing problem. The trunk routing problem is defined as follows:

Trunk routing problem

Input: Routing area with obstacles.
A set of nets that satisfies the trunk routing topology condition.

Output: Routing pattern that connects all nets without violating the design rule (minimum distance between wires and obstacles).

For example, a routing pattern that satisfies the design rule is shown in Fig. 4. In this routing pattern, all nets pass along the H, V, and X tracks only and do not intersect. This routing pattern is feasible.

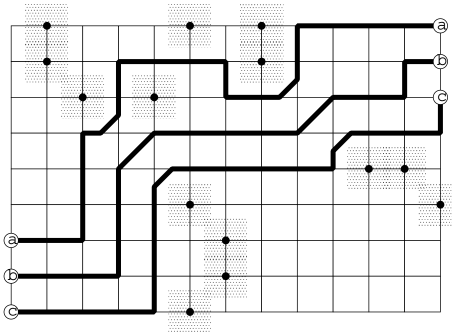

Fig. 4. An example of HVX routing for the problem of Fig. 1

In the trunk routing problem, each unit flow on the flow graph G corresponds to the route from a source terminal to a sink terminal using only H and V tracks. The maximum number of routes using only H and V tracks can be obtained using a set of unit flows with the maximum number.

III. X track

The X tracks defined in part of the routing area are specified by the pitch and the offset. The routing area is modeled on the XY coordinate plane. The pitch of the X tracks is the horizontal distance between X tracks, and the offset of the X tracks is the reminder of the y-intercepts of the X tracks divided by the pitch of the X tracks, as if they are extended to intersect to y-axis. The pitch and offset of the X tracks should be defined carefully because they affect the connectivity.

In multiple layer routing, an X track is often defined such that it intersects an HV track at a grid-point. It is easy to place vias that connect wires on different layers if tracks intersect at grid-points. However, in single layer routing, an X track does not necessarily intersect an HV track at a grid-point. The pitch must be at least $\sqrt{2} \simeq 1.41$, so that the distance between X tracks is at least one. Although the connectivity becomes maximum when the pitch is set to $\sqrt{2}$, the routing tool may cause a malfunction due to rounding error, for example. In the proposed method, the pitch is set to 1.5 so that an X track intersects an HV track either at a grid-point or at the middle point of adjacent grid-points. Even though the distance between X tracks is greater than one, in most cases, maximum connectivity is achieved.

The offset of the X tracks also affects the connectivity. Let us consider the number of wires that runs between the two obstacles shown in Fig. 5 for the case in which the pitch of the X tracks is 1.5. The routing results for the cases in which the offset of the X tracks is set to 0.0, 0.5, and 1.0 are shown in Fig. 6, 7, and 8, respectively. The connection requirements are realized only for the case in which the offset is set to 0.0, in which case, three wires run between obstacles. In general, the number of wires between obstacles at (x, y) and $(x+w+1, y+h+1)(w, h \geq 1)$ is maximized if the offset is set to $\frac{(3-(x+y))\text{mod}3}{2}$. The number of wires that can run between the obstacles using the X tracks when the offset is set appropriately is

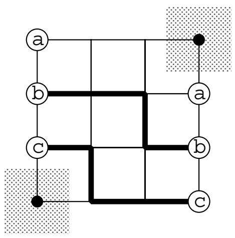

Fig. 5. HV routing. #net=2

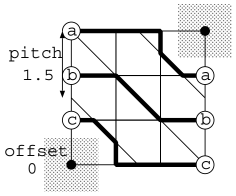

Fig. 6. HVX routing. offset=0.0, #net=3

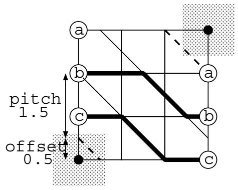

Fig. 7. HVX routing. offset=0.5, #net=2

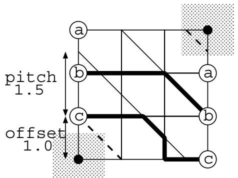

Fig. 8. HVX routing. offset=1.0, #net=2

$\left\lceil\frac{2(w+h)-1}{3}\right\rceil$, whereas the number of wires that can run between the obstacles using HV tracks is $\max(w, h)$. The connectivity between these obstacles is improved when $\left\lceil\frac{2(w+h)-1}{3}\right\rceil > \max(w, h)$ if the offset of the X track is set appropriately.

An appropriate offset of X tracks depends on the pair of obstacles. An offset of X tracks that maximizes the number of wires between a pair of obstacles does not necessarily maximize the number of wires between another pair of obstacles. The offset of X track should be defined locally in order to achieve the maximum connectivity. In the proposed method, a sub-area, referred to as the critical zone, in which X tracks are essential in order to realize the connection requirements, is identified, and the offset of the X track is defined appropriately for each sub-area. The identification of the critical zone is discussed in Section IV.

IV. Critical zone

In the following, we consider a routing problem that satisfies the trunk routing topology condition.

A line that divides the routing area into two sub-areas such that one area contains all source terminals and the other area contains all sink terminals is referred to as a dividing line. A dividing line corresponds to a cut of flow graph G'. In a feasible routing pattern, the wire of any net passes any dividing line. The number of wires that can pass a dividing line depends on the design style. The maximum numbers of wires that can pass a dividing line L using HV tracks and HVX tracks are referred to as the capacities of L in the HV and HVX design styles, respectively. Let $C_{HV}(L)$ and $C_{HVX}(L)$ be the capacities of a dividing line L in the HV and HVX design style, respectively.

Let n be the number of nets. If $C_{HV}(L) \geq n$ for all dividing lines L, then all nets can be connected using only HV tracks. Whereas, if there exists a dividing line L, the capacity of which in the HV design style is less than n ($C_{HV}(L) < n$), then not all nets can be connected using only HV tracks. Even though not all nets can be connected using only HV tracks, all nets can be connected using HVX tracks if $C_{HVX}(L) \geq n$ for all dividing lines L.

A dividing line L is said to be critical if $C_{HV}(L) < n$. Not all nets can pass a dividing line L using only HV tracks if L is critical. In order to obtain a feasible routing pattern, the capacities of all critical dividing lines should be increased. However, the number of critical dividing lines would become exponential. Thus, it is impractical to increase the capacities of all critical dividing lines individually.

A dividing line may pass over obstacles. A dividing interval is the interval of a dividing line such that both ends are located in an obstacle or on the boundary of the routing area, which does not pass over any obstacle. A dividing line consists of several dividing intervals. The capacity of a dividing interval is defined analogously. Let $C_{HV}(I)$ and $C_{HVX}(I)$ be the capacity of a dividing interval I in the HV and HVX design styles, respectively.

Let L' be a critical dividing line that consists of dividing intervals $I'_0, I'_1, \ldots, I'_m$, where I'_i is an interval between obstacles B_i and B_{i+1} ($0 \leq i \leq m$). The sequence of obstacles $B_0, B_1, \ldots, B_{m+1}$ is derived from Lf. There may exist several critical dividing lines, where the sequence of obstacles derived is the same as L'. The capacities of critical dividing lines from which the sequence of obstacles $B_0, B_1, \ldots, B_{m+1}$ is derived are increased simultaneously. Let L be a critical dividing line in which the order that passes obstacles is the same as L' and whose capacity in the HV design style is minimum among them. By adopting the HVX design style, if all nets can pass L, then all nets can pass dividing lines from which the sequence of obstacles $B_0, B_1, \ldots, B_{m+1}$ is derived. Let $I_0, I_1, \ldots, I_m$ be the dividing intervals of L.

Assume that $n > C_{HV}(L)$ and $n \leq C_{HVX}(L)$, and assume that the number of X tracks used at L is minimized.

The number of nets that can pass I_i in the HVX design style is at most $C_{HVX}(I_i)$. If the maximum number of nets pass dividing intervals of L, except for I_i in the HV design style, then the number of nets that must pass I_i is $n - C_{HV}(L) + C_{HV}(I_i)$. Let $K(I_i) = \max(C_{HVX}(I_i), n - C_{HV}(L) + C_{HV}(I_i))$. The number of nets that pass I_i is at most $K(I_i)$.

Let $d_{HV}(g, B_i)$ be the maximum of x-distance and y-distance between grid-point g and obstacle B_i. Note that $d_{HV}(g, B_i)$ corresponds to the number of wires that can pass between g and B_i in the HV design style. Let the capacity $C_{HV}(B_i, g, B_{i+1})$ of grid-point g in terms of obstacles B_i and B_{i+1} in the HV design style be the maximum number of wires that can pass any dividing interval between B_i and B_{i+1} that passes g in the HV design style. Note that $C_{HV}(B_i, g, B_{i+1}) = d_{HV}(g, B_i) + d_{HV}(g, B_{i+1}) - 1$.

If $C_{HV}(B_i, g, B_{i+1}) \leq K(I_i)$, then all nets can pass any dividing interval between B_i and B_{i+1} that passes g without an X track. Otherwise, there exists a feasible routing pattern that uses an X track in order to pass the dividing interval between B_i and B_{i+1} that passes g.

The set of grid-points g having a capacity of $C_{HV}(B_i, g, B_{i+1})$ is less than or equal to $K(I_i)$ is defined as the critical zone between B_i and B_{i+1}. If X tracks are appropriately defined in the critical zone between B_i and B_{i+1} ($0 \leq i \leq m$), then all of the nets can pass the critical dividing lines that pass over

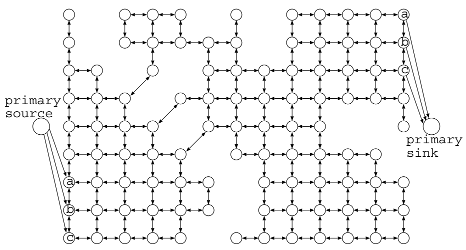

Fig. 9. Flow graph modified from the flow graph shown in Fig. 3

$B_0, B_i, \ldots, B_{m+1}$.

V. Proposed HVX routing

The proposed routing method consists of three phases. In the first phase, the proposed method analyzes the given trunk routing problem and extracts critical zones from the routing area. In the second phase, a routing pattern in each extracted critical zone is determined using H, V, and X tracks. In the third phase, a routing pattern in the remaining routing area is determined using only H and V tracks.

A. *Extracting critical zones*

In the first phase, the proposed method analyzes the given trunk routing problem to determine whether a feasible routing pattern can be obtained. If a feasible routing pattern can be obtained, the proposed method extracts critical zones from the routing area.

First, a flow graph with vertex capacity is constructed from a given routing area. Then, a minimum cut is obtained using a flow method[16]. If the capacity of the obtained minimum cut is larger than or equal to the number of nets, then the proposed method proceeds to the second phase. Otherwise, the capacity of the corresponding dividing line in the HVX design style is calculated. If the calculated capacity is less than the number of nets, then the proposed method stops and fails to obtain a feasible routing pattern. Otherwise, critical zones in terms of the corresponding dividing line are extracted. For example, the minimum cut obtained by the flow method that consists of vertices indicated by striped lines is shown in Fig. 3. In this example, the two vertices are extracted as a critical zone.

The capacity of the obtained minimum cut is then increased by modifying the flow graph to find other critical dividing lines. In order to increase the capacity of the obtained minimum cut, the flow graph is modified as follows.

1. The vertices corresponding to critical zones and the edges incident to the vertices are deleted.
2. Edges with infinite capacity are inserted between vertices that are adjacent to the deleted vertices.

As an example, the modified flow graph obtained from the flow graph in Fig. 3 is shown in Fig. 9.

Then, a minimum cut in the modified flow graph is obtained. This procedure is repeated until the capacity of a minimum cut of the flow graph becomes at least equal to the number of nets or a dividing line that cannot be passed is found.

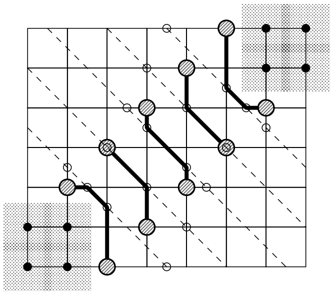

Fig. 10. An example of critical zone routing

B. *Critical zone routing*

In the second phase, a routing pattern is generated for each critical zone. In each critical zone, X tracks are first generated so that the number of wires that can pass the critical zone is maximized. Let us consider a critical zone in which k nets pass ($1 \leq k \leq n$). The boundary of the critical zone is divided into the source terminal side boundary, the sink terminal side boundary, and the remaining boundary, which corresponds to the boundary of the routing area or obstacles. The source terminal side boundary has k on-points. The number of H tracks, V tracks, and X tracks that pass the source (sink) terminal side boundary is k. On-points in the source terminal side boundary are labeled $s_1, s_2, \ldots, s_k$ along the boundary. Similarly, the i-th H track, i-th V track, and i-th X track that pass the source terminal side boundary are labeled h_i^s, v_i^s, and x_i^s, respectively.

The route of a net that passes the s_i is then determined so that the route connects s_i and t_i using h_i^s, v_i^s, x_i, h_i^t, and v_i^t. The wire of each net uses part of an X track in the zone and uses HV tracks if necessary. An example of a routing pattern in a critical zone is shown in Fig. 10.

In the first phase, each critical zone corresponding to dividing interval I_i is defined assuming that $K(I_i)$ nets pass I_i. However, the number of wires that pass I_i, which corresponds to the amount of flow that passes I_i in the final flow graph, might be less than $K(I_i)$. If the number of wires that pass I_i is less than $K(I_i)$, then each critical zone is redefined using the number of wires that pass I_i instead of $K(I_i)$.

C. *Non-critical zone routing*

In the third phase, the routing pattern outside the critical zones is determined. In this phase, the routing problem is divided into several subproblems by regarding critical zones as obstacles and by arranging terminals on the boundaries of critical zones. Each subproblem satisfies the trunk routing topology condition and the connection requirement can be realized using only HV tracks. Although the routing method used in this phase is not specified, any routing method can be applied, even methods that cannot handle X tracks appropriately.

Finally, the results for the routing patterns are merged. The obtained routing pattern does not take various indices into account. Therefore, it might be necessary to modify the obtained routing pattern in post processing, but this is beyond the scope of this paper.

TABLE I
Experimental results

Ratio	Design style	d1	d2	d3	d4	d5
0.05	HV	18	19	18	18	19
	HVX	18	**20**	**19**	**19**	**20**
0.10	HV	15	14	13	13	15
	HVX	15	**16**	13	**14**	**16**
0.15	HV	12	14	13	13	14
	HVX	**13**	14	**14**	**14**	14
0.20	HV	10	11	10	10	11
	HVX	10	11	10	10	11

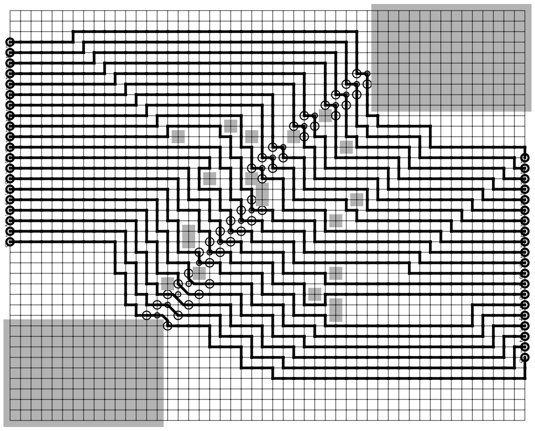

Fig. 11. An example of the routing pattern

VI. Experimental results

The proposed routing method is implemented in C++, which is compiled with gcc4.2.4 and executed on a PC having a 2.66-GHz Intel Core2 CPU and 4 GB of RAM.

The proposed method is applied to 20 sample data sets in which obstacles are randomly generated in the middle of the routing area. The obstacle ratio in the middle of the routing area is changed from 0.05 to 0.20 in 0.05 increments. The number of nets in each sample data set is 20. In Table I, for each sample data set, the numbers of nets that are completed in HV design style and HVX design style by the proposed method are shown. Bold font is used if the number of nets that are completed in the HVX design style is larger than that in the HV design style. Figure 11 shows the routing pattern of d2, the obstacle ratio of which is 0.05, obtained by the proposed method. In the figure, grid-points that correspond to the minimum cut and grid-points that correspond to the boundary of the corresponding critical zone are represented by a small circle and a large circle, respectively.

If the obstacle ratio is between 0.05 and 0.15, then the number of nets that can connect often increase with the use of X tracks. The effect of the X track increases when obstacles are placed diagonally and when the distance between the obstacles is relatively large. In this experiment, small obstacles are randomly arranged in a small area. The condition of this experiment appears to be relatively severe compared to the requirements in actual PCB design. These results indicate that the proposed method is promising for application in actual PCB design.

VII. Conclusion and future research

In this paper, we proposes a routing method for a trunk routing problem in which the connection requirements are realized in one routing layer using 45-degree segments as well as horizontal and vertical segments. In the proposed routing method, 45-degree segments are used in critical zones only. The proposed method extracts every critical zone efficiently and then obtains a route in the critical zone using horizontal segments, vertical segments, and 45-degree segments. In the non-critical zone, any routing tool that can handle horizontal and vertical segments only can be used to improve various specifications because the proposed method guarantees the connectivity.

In the future, we intend to enhance the proposed method to enable various evaluations, such as length, bends, and delay, to be taken into account. In addition, we will develop a PCB routing system that uses the proposed method as a subroutine.

References

[1] Y. Kajitani, “The Potential Router,” Technical Report of IEICE, VLD2006–129, vol.106, no.548, pp.61–66, 2007. (In Japanese).

[2] M. Inagi, Y. Takashima, and Y. Kajitani, “Escape Fitting between a Pair of Pin-Sets,” Technical Report of IEICE, VLD2006–130, vol.106, no.548, pp.67–72, 2007. (In Japanese).

[3] T. Yan and M. Wong, “Untangling Twisted Nets for Bus Routing,” Proceedings of International Conference Computer-Aided Design, pp.396–400, 2007.

[4] L. Luo and M. Wong, “Ordered Escape Routing Based on Boolean Satisfiability,” Proceedings of Asia and South Pacific Design Automation Conference (ASP-DAC), pp.244–249, 2008.

[5] L. Luo and M. Wong, “On using sat to ordered escape problems,” ASP-DAC ’09: Proceedings of the 2009 Asia and South Pacific Design Automation Conference, Piscataway, NJ, USA, pp.594–599, IEEE Press, 2009.

[6] Y. Kohira and A. Takahashi, “CAFE router: A Fast Connectivity Aware Mutiple Nets Routing Algorithm for Routing Grid with Obstacles,” Proceedings of Asia and South Pacific Design Automation Conference(ASP-DAC), pp.281–286, 2010.

[7] R. Pinter, “On Routing Two-Point Nets Across a Channel,” Proceedings of Design Automation Conference, pp.894–902, 1982.

[8] C. Hsu, “Channel River Routing Algorithm,” Proceedings of Design Automation Conference, pp.241–246, 1983.

[9] T. Lengauer, ”Combinatorial algorithms for integrated circuit layout”, John Wiley & Sons, Inc., New York, NY, USA, 1990.

[10] M. Ozdal and M. Wong, “A Length-Matching Routing Algorithm for High Performance Printed Circuit Boards,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol.25, no.12, pp.2784–2794, 2006.

[11] M. Ozdal and M. Wong, “Algorithmic Study of Single-Layer Bus Routing for High-Speed Boards,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol.25, no.3, pp.490–503, 2006.

[12] T. Yan and M. Wong, “BSG-route: a length-constrained routing scheme for general planar topology,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol.28, no.11, pp.1679–1690, 2009.

[13] N. Ito and H. Katagiri, “Diagonal Routing in High Performance Microprocessor Design,” Proceedings of Asia and South Pacific Design Automation Conference (ASP-DAC), pp.623–629, 2006.

[14] C. Chiang, Q. Su, and C. Chiang, “Wirelength Reduction by Using Diagonal Wire,” Proceedings of Great Lakes Symposium on VLSI, pp.104–107, 2003.

[15] S.L. Teig, “The X architecture: not your father’s diagonal wiring,” SLIP ’02: Proceedings of the 2002 international workshop on System-level interconnect prediction, New York, NY, USA, pp.33–37, ACM, 2002.

[16] J.A. Bondy and U.S.R. Murty, ”Graph Theory with Applications”, Elsevier Science Ltd, 1976.