論文 / 著書情報
Article / Book Information

Title	Single-Layer Trunk Routing Using Minimal 45-Degree Lines
Authors	Kyosuke Shinoda, Yukihide Kohira, Atsushi Takahashi
出典 / Citation	IEICE Trans. Fundamentals, Vol. E94-A, No. 12, pp. 2510-2518
発行日 / Pub. date	2011, 12
URL	http://search.ieice.org/
権利情報 / Copyright	本著作物の著作権は電子情報通信学会に帰属します。 Copyright (c) 2011 Institute of Electronics, Information and Communication Engineers.

PAPER Special Section on VLSI Design and CAD Algorithms

Single-Layer Trunk Routing Using Minimal 45-Degree Lines*

Kyosuke SHINODA[†a)], *Nonmember*, **Yukihide KOHIRA**[††], *and* **Atsushi TAKAHASHI**[†††], *Members*

SUMMARY In recent Printed Circuit Boards (PCB), the design size and density have increased, and the improvement of routing tools for PCB is required. There are several routing tools which generate high quality routing patterns when connection requirement can be realized by horizontal and vertical segments only. However, in high density PCB, the connection requirements cannot be realized when only horizontal and vertical segments are used. Up to one third nets can not be realized if no non-orthogonal segments are used. In this paper, a routing method for a single-layer routing area that handles higher density designs in which 45-degree segments are used locally to relax the routing density is introduced. In the proposed method, critical zones in which non-orthogonal segments are required in order to realize the connection requirements are extracted, and 45-degree segments are used only in these zones. By extracting minimal critical zones, the other area that can be used to improve the quality of routing pattern without worry about connectivity issues is maximized. Our proposed method can utilize the routing methods which generate high quality routing pattern even if they only handle horizontal and vertical segments as subroutines. Experiments show that the proposed method analyzes a routing problem properly, and that the routing is realized by using 45-degree segments effectively.
key words: *printed circuit board, planar routing, trunk routing, routing density, 45-degree line*

1. Introduction

Printed circuit board routing realizes connections between elements, and is requested to achieve the specifications related to delay and noise, for example. Designers generate a routing pattern that achieves the specifications by using orthogonal segments as well as non-orthogonal segments effectively. While the quality of the routing pattern obtained by automatic PCB routing tools is inferior to the routing pattern obtained by designers, and the routing pattern of high-density PCBs are still obtained by hands. However, the number of nets on a PCB has increased due to the emergence of elements which have numerous I/O pins such as BGA packages. The design scale increases and approaches the limit of manual design, and the time required to complete PCB design by hands increases. Therefore, the quality enhancement of automatic PCB routing tools is needed in order to shorten the design period.

There are several routing tools that generate high-quality routing patterns when only horizontal and vertical segments are used. However, non-orthogonal segments are often essential in order to realize the connection requirements. Up to one third nets can not be realized if no non-orthogonal segments are used. 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 [2]–[4]. However, in the routing methods proposed in [2]–[4], non-orthogonal segments are primarily used to shorten the wirelength. These methods do not effectively use non-orthogonal segments to realize connection requirements when the routing resources are limited.

In a typical PCB routing design flow, first, in the global routing phase, the global structure of the routing pattern 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 routing pattern 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 [5]–[9]. 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.

In the river problem, a routing pattern should connect the terminals of each net and achieve various specifications [10]–[16]. 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 [16]. 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 routing tools. 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

Manuscript received March 26, 2011.
Manuscript revised June 20, 2011.
[†]The author is 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 Graduate School of Engineering, Osaka University, Suita-shi, 565-0871 Japan.
[*]The preliminary version was presented at [1].
a) E-mail: shinoda@lab.ss.titech.ac.jp
DOI: 10.1587/transfun.E94.A.2510

obtained routing pattern will contain a number of unnecessary bends and detours, 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. The quality improvement in the trunk routing problem is needed in order to improve the quality of automatic PCB routing tools.

In this work, in order to generate a satisfactory routing pattern in trunk routing problem in short time, we propose a routing method for a trunk routing problem in which 45-degree segments are used locally to relax the routing density. In the proposed method, critical zones in which non-orthogonal segments are required in order to realize the connection requirements are extracted, and 45-degree segments are used only in these zones. By extracting minimal critical zones, the other area that can be used to improve the quality of routing patterns without worry about connectivity issues is maximized. This will relax the design constraints and will increase the probability that the routing methods generate the high quality routing patterns even if the methods can not use non-orthogonal segments outside the critical zones.

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 non-critical zones, in which no non-orthogonal segments are required. The proposed method efficiently generates a feasible routing pattern. Although the detail of the quality improvement of a routing pattern is beyond the scope of this paper, the proposed method will enable us to obtain a satisfactory routing pattern in short time.

2. Preliminaries

In this paper, we consider the single-layer routing problems which satisfies the *trunk routing topology condition*. The trunk routing topology condition is defined in [16] 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.

In the following, we assume that the boundary of the routing area is divided into the left, the right, the top, the bottom boundaries and that the source terminals and the sink terminals are set on the left boundary and the right boundary, respectively. A single-layer routing problem that satisfies this topology condition, called a *trunk routing problem*, is defined as follows:

Trunk routing problem

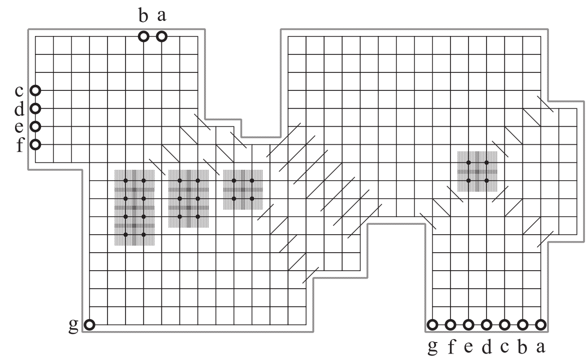


Fig. 1 Single-layer multiple-nets problem.

Input: Single-layer routing area A with obstacles B, and the set of nets N where $n = |N|$ that satisfies the trunk routing topology condition.

Output: Routing pattern W that connects all nets by wires without violating the design rule.

The routing area is modeled on the XY coordinate plane. In this paper, the distance is the Euclidean distance. Also, the x-distance and the y-distance are the x-component and the y-component of the Manhattan distance. In our problem formulation, the width of each wire, the minimum distance between adjacent wires, and the minimum distance between a wire and an obstacle are set to zero, one, and one unit length as the design rule, respectively. The number of nets that can be connected by wires without violating the design rule depends on the design style.

The design styles used in this paper are defined by using the concept of track. 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, and referred to as *HV tracks*. X tracks are defined locally with the same separation. Details are given in Sect. 3. When wires run only on HV tracks, it is called the *HV design style*. While, when wires run on X tracks as well as HV tracks, it is called the *HVX design style*.

The intersection of an H track and a V track is referred to as a grid-point. Let (x_p, y_p) be the coordinate of grid-point p. Let $\Delta x_{pq} = |x_p - x_q|$ and $\Delta y_{pq} = |y_p - y_q|$ be the x-distance and y-distance between grid-points p and q, respectively. 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 an *off-point* if it is contained in an obstacle and is otherwise called an *on-point*.

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

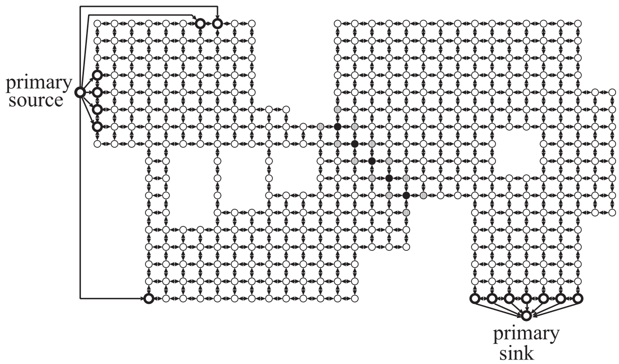


Fig. 2 Flow graph with vertex capacity.

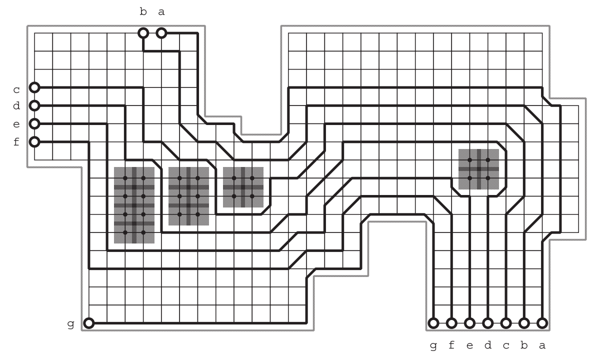


Fig. 3 Example of HVX routing pattern.

gray region that surrounds a black dot. A terminal of a net is represented by a thick white circle.

Flow graph is defined to help analyze a given trunk routing problem. The flow graph G with vertex capacity is defined in [16] 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, one vertex is included in the vertex set of G. The edge set of G consists of the following three types of directed edges: **type-1)** a directed edge from the vertex of an on-point to the vertex of an adjacent on-point; **type-2)** a directed edge from the primary source to the vertex of the on-point to which a source terminal is assigned; **type-3)** a directed edge from the vertex of the on-point to which the sink terminal is assigned to the primary sink. The capacity of a vertex is one if the vertex corresponds to an on-point, and is infinite if the vertex is the primary source or the primary sink. The capacity of an edge is infinite. A vertex set V_c is called a cut if the deletion of V_c from the flow graph disconnects the primary source and the primary sink, and if the deletion of any proper subset of V_c does not. The capacity $C(V_c)$ of a cut V_c is the sum of the capacities of vertices in V_c. A cut whose capacity is minimum among all cuts is called a minimum cut. The capacity of a minimum cut is equal to the maximum number of unit flows from the primary source to the primary sink in G [17]. Unit flows in G do not intersect because the capacity of each vertex is set to one. In the trunk routing problem, each unit flow in G in unit flows of the maximum number corresponds to a routing pattern using only HV tracks if the maximum number is equal to the number of nets. The maximum number of nets which can be connected using only HV tracks is determined by the capacity of the minimum cut of G.

For example, the flow graph with vertex capacity corresponding to the routing area shown in Fig. 1 is shown in Fig. 2. The cut which is represented by black vertices is a minimum cut of the flow graph. The capacity of the cut is 5. Although the maximum number of routes using only HV tracks is 5, the maximum number of routes that satisfies the design rule is 7. The routing pattern using HV and X tracks shown in Fig. 3 satisfies the design rule.

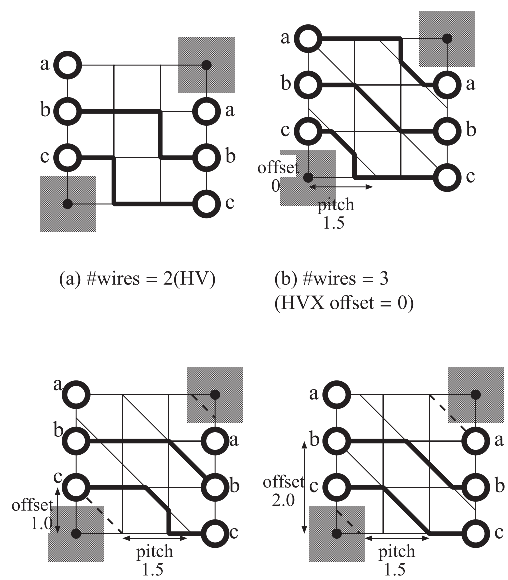
3. X Track

The X tracks defined in part of the routing area are specified by the pitch and the offset. The pitch of the X tracks is the Manhattan 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.

The distance between adjacent wires should be at least one by the design rule. If the distance between adjacent tracks is set to less than one, whether a wire can use a track depends on whether an adjacent track is used by other wire. In order to make the routing procedure simple, that is, in order to make the decision whether a wire can use a track possible independent of an adjacent track, the pitch of X tracks is determined so that the distance between adjacent tracks is at least one. Although the number of X tracks is 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.

Since the pitch is defined to make the distance of X tracks at least one, the offset of the X tracks also affects the connectivity. Let us consider the number of wires that run between the two obstacles at $(0, 0)$ and $(3, 3)$ shown in Fig. 4(a) where the pitch of the X tracks is 1.5. The routing results when the offset of the X tracks is set to 0.0, 1.0, and 2.0 are shown in Figs. 4(b), 4(c), and 4(d), respectively. The maximum number of wires that can run between the obstacles is 3 only if the offset is 0.0. In general, the number of wires that can run between obstacle at grid-point p and obstacle at grid-point q, where $x_p < x_q$, is maximized if the offset is set to $(x_p + y_p)$ mod 3. In other words, the number of wires is maximized when the offset of the X tracks is set so that the Manhattan distance between the obstacle and the X tracks is 1.5.

The maximum number of wires that can run between



(a) #wires = 2(HV)
(b) #wires = 3 (HVX offset = 0)
(c) #wires = 2 (HVX offset = 1.0)
(d) #wires = 2 (HVX offset = 2.0)

Fig. 4 The number of wires between obstacles.

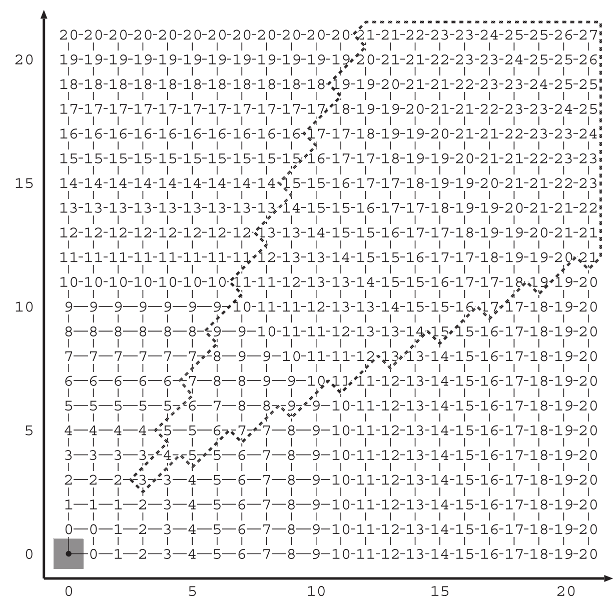


Fig. 5 The HVX capacity between an obstacle at (0, 0) and another obstacle at each grid-point.

obstacles depends on the design style. Let HV, X, and HVX capacities between two obstacles B_i and B_j be the maximum number of wires that can run between B_i and B_j by using HV tracks, by using X tracks, and by using HVX tracks, respectively, when the pitch of the X tracks is 1.5 and the offset of the X tracks is set appropriately. HV, X, and HVX capacities are denoted by $C_{HV}(B_i, B_j)$, $C_X(B_i, B_j)$, and $C_{HVX}(B_i, B_j)$, respectively. Note that

$$C_{HV}(B_i, B_j) = \min_{p \in B_i, q \in B_j} \left(\max\left(\Delta x_{pq}, \Delta y_{pq}\right)\right) - 1,$$

$$C_X(B_i, B_j) = \min_{p \in B_i, q \in B_j} \max\left(0, \left\lfloor \frac{2\left(\Delta x_{pq} + \Delta y_{pq}\right)}{3} \right\rfloor - 1\right),$$

and

$$C_{HVX}(B_i, B_j) = \max(C_{HV}(B_i, B_j), C_X(B_i, B_j)).$$

The wires that can run between B_i and B_j increase by changing design style from HV to HVX if grid-points p and q that determine $C_{HVX}(B_i, B_j)$ satisfy $2\Delta x_{pq} - 3 \geq \Delta y_{pq} \geq \frac{\Delta x_{pq}+3}{2}$. The HVX capacity between an obstacle at $(0, 0)$ and another obstacle at (x, y) is shown in Fig. 5. In Fig. 5, the number on the each grid-point means the HVX capacity between an obstacle at $(0, 0)$ and another obstacle at each grid-point. In the dotted frame, the HVX capacity between two obstacles is larger than the HV capacity.

The wires equal to the HVX capacity can run between B_i and B_j if the offset is set appropriately. An appropriate offset of X tracks depends on the pair of obstacles. In the proposed method, the *critical zones*, in which X tracks are essential in order to realize the connection requirements, are identified between two obstacles, and the offset of the X tracks is defined for each critical zone.

4. Critical Zone

In this section, we discuss the critical zone which is defined in a routing problem that satisfies the trunk routing topology condition.

Let HV and HVX capacities of a cut V_c be the maximum number of wires that can pass over the line in the routing area that corresponds to V_c by using HV tracks and by using HVX tracks, respectively, when the pitch of the X tracks is 1.5 and the offset is set appropriately. HV and HVX capacities of V_c are denoted by $C_{HV}(V_c)$ and $C_{HVX}(V_c)$, respectively. Note that $C_{HV}(V_c) = C(V_c)$. If $C_{HV}(V_c) \geq n$ for all cuts V_c, then all nets can be connected using only HV tracks. Whereas, if there exists a cut V_c such that $C_{HV}(V_c) < n$, 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 $C_{HVX}(V_c) \geq n$ for all cut V_c.

A cut V_c is said to be critical if $C_{HV}(V_c) < n$. Not all nets can pass over the line that corresponds to a cut V_c using only HV tracks if V_c is critical. In order to obtain a feasible routing pattern, X tracks should be introduced for each critical cut. However, the number of critical cuts would become exponential. Thus, it is impractical to introduce X tracks for all critical cuts individually.

The line that corresponds to a cut may pass over obstacles. An interval of a cut is a subset of the cut that corresponds to a part of the line that corresponds to the cut such that both ends are located on an obstacle or the boundary of the routing area, which does not pass over any obstacle. That is, a cut consists of several intervals.

Let V'_c be a cut which consists of intervals $I'_0, I'_1, \ldots, I'_m$, where I'_i is an interval between obstacles B_i and B_{i+1} ($0 \leq$

$i \leq m$). Note that B_0 and B_{m+1} are pseudo obstacles that correspond to the top boundary and the bottom boundary of the routing area, respectively. The sequence of obstacles $B_0, B_1, \ldots, B_{m+1}$ is derived from V'_c. Let $C_{HV}(I'_i)$ and $C_{HVX}(I'_i)$ be the capacity of a interval I'_i in the HV and HVX design styles, respectively. Note that $C_{HV}(V'_c) = \sum_{i=0}^{m} C_{HV}(I'_i)$ and that $C_{HV}(I'_i) \geq C_{HV}(B_i, B_{i+1})$ for any i ($1 \leq i \leq m$).

Let $\mathcal{B} = (B_0, B_1, \ldots, B_{m+1})$ be a sequence of obstacles where B_0 and B_{m+1} are pseudo obstacles that correspond to the top boundary and the bottom boundary of the routing area, respectively. Note that $C_{HV}(V_c) \geq \sum_{i=0}^{m} C_{HV}(B_i, B_{i+1})$ for any cut V_c from which $\mathcal{B}$ is derived. If $\sum_{i=0}^{m} C_{HV}(B_i, B_{i+1}) \geq n$, then, in the HV design style, all nets can pass $\mathcal{B}$. More precisely, any line that corresponds to a cut from which $\mathcal{B}$ is derived can be passed in the HV design style. The number of nets that can pass over between B_i and B_{i+1} in the HVX design style is at most $C_{HVX}(B_i, B_{i+1})$. In order for all nets to pass over $\mathcal{B}$, the number of nets that are requested to pass over between B_i and B_{i+1} when the number of nets that pass over $\mathcal{B}$ except between B_i and B_{i+1} in the HV design style is maximized is $n - \sum_{j \neq i} C_{HV}(B_j, B_{j+1})$. Let $K(i, \mathcal{B}) = \min(C_{HVX}(B_i, B_{i+1}), n - \sum_{j \neq i} C_{HV}(B_j, B_{j+1}))$. Note that $\sum_{i=0}^{m} K(i, \mathcal{B}) \geq n$ if $\sum_{i=0}^{m} C_{HVX}(B_i, B_{i+1}) \geq n$. In the proposed method, a critical zone between B_i and B_{i+1} in terms of $\mathcal{B}$ is extracted by assuming that the number of nets that pass over between B_i and B_{i+1} is $K(i, \mathcal{B})$.

An example is shown in Fig. 6. In Fig. 6, a cut V_c where the sequence of obstacles is (B_0, B_1, B_2) consists of two intervals I_0 and I_1. Note that $C_{HV}(B_0, B_1) = 7$, $C_{HV}(B_1, B_2) = 4$, $C_{HVX}(B_0, B_1) = 9$, and $C_{HVX}(B_1, B_2) = 5$. I'_0 and I''_0 are intervals between obstacles B_0 and B_1. Note that $C_{HV}(I_0) = 7$, $C_{HV}(I'_0) = 9$, $C_{HV}(I''_0) = 11$, $C_{HV}(I_1) = 4$, and $C_{HV}(V_c) = 11$. The capacity of I_0 is minimum among all intervals between obstacles B_0 and B_1. Also, the capacity of V_c is minimum among all cuts from which the sequence of obstacles (B_0, B_1, B_2) is derived.

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 interval between B_i and B_{i+1} that passes g in the HV design style. Note that $C_{HV}(B_i, g, B_{i+1}) = C_{HV}(g, B_i) + C_{HV}(g, B_{i+1}) + 1$. Let a critical zone between B_i and B_{i+1} in terms of $\mathcal{B}$ be a maximal set of grid-points g such that $C_{HV}(B_i, g, B_{i+1}) < K(i, \mathcal{B})$ and that corresponds to a connected part of the routing area. Note that if $C_{HV}(B_i, g, B_{i+1}) < K(i, \mathcal{B})$, then $K(i, \mathcal{B})$ nets can not pass any interval between B_i and B_{i+1} that passes g without using any X track.

In Fig. 6, the capacity $C_{HV}(B_0, g, B_1)$ is 7, and the capacity $C_{HV}(B_0, g', B_1)$ is 8. If $n = 12$, $K(0, \mathcal{B}) = 8$ and $K(1, \mathcal{B}) = 5$. In this case, the grid-point g is critical since $C_{HV}(B_0, g, B_1) < K(0, \mathcal{B})$. However, the grid-point g' is non-critical since $C_{HV}(B_0, g', B_1) \geq K(I'_0, \mathcal{B})$. If $n = 14$, $K(0, \mathcal{B}) = 9$ and $K(1, \mathcal{B}) = 5$. In this case, the grid-points g and g' are critical since $C_{HV}(B_0, g, B_1) < K(I_0, \mathcal{B})$ and $C_{HV}(B_0, g', B_1) < K(I'_0, \mathcal{B})$.

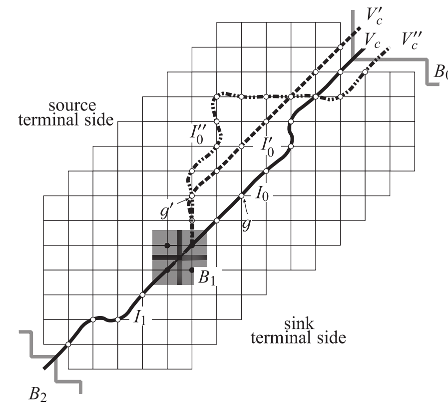


Fig. 6 Dividing line and dividing intervals.

Let us consider a critical zone between B_i and B_{i+1} in terms of $\mathcal{B}$ over which $K(i, \mathcal{B})$ nets pass. The boundary of the critical zone is the set of grid-points g such that $C_{HV}(B_i, g, B_{i+1}) = K(i, \mathcal{B})$ and that surround the grid-points of the critical zone. The boundary of a critical zone is divided into the *source terminal side boundary* and *the sink terminal side boundary* each of which has at least $K(i, \mathcal{B})$ grid-points. The i-th X track from the top is labeled x_i ($1 \leq i \leq K(i, \mathcal{B})$). Among the grid-points in the source terminal side boundary, $K(i, \mathcal{B})$ grid-points are selected so that they can be connected to X tracks by using HV tracks without intersecting each other, and labeled $s_1, s_2, \ldots, s_{K(i,\mathcal{B})}$ so that s_j is connected to x_j ($1 \leq j \leq K(i, \mathcal{B})$). H track and V track that passes over s_i are labeled h_i^s, v_i^s, respectively. Similarly, $K(i, \mathcal{B})$ grid-points in the sink terminal side boundary are selected and labeled $t_1, t_2, \ldots, t_{K(i,\mathcal{B})}$. H track and V track that passes over t_i are labeled h_i^t, v_i^t, respectively.

5. 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 outside the critical zone is determined.

5.1 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 in the HVX design style, 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 of flow graph is obtained by a flow method [17] to check whether there exists a critical cut in the flow graph. Let V_c be the found minimum cut. If $C_{HV}(V_c) \geq n$, then there is no critical cut and the proposed method proceeds to the third phase.

Otherwise V_c is a critical cut. If $n > C_{HVX}(V_c)$, then the proposed method stops since no feasible routing pattern is obtained. Otherwise, that is, if $C_{HVX}(V_c) \geq n > C_{HV}(V_c)$, then the proposed method extracts the critical zones each of which corresponds to an interval of V_c.

Even if the critical zones in terms of V_c are extracted, there may exist other critical zones in terms of other critical cuts. In order to find the other critical zones by eliminating V_c from the flow graph, the flow graph is modified as follows.

1. The vertices corresponding to the critical zones and the edges incident to the vertices are deleted.
2. Edges with infinite capacity connecting a vertex on the source terminal side boundary of the critical zones and a vertex on the sink side boundary of the critical zones are inserted.

Then, a minimum cut in the modified flow graph is obtained and the capacities of the found minimum cut are checked as above.

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 cut that cannot be passed is found or critical zones with overlap are found.

For example, the minimum cut obtained by the flow method that consists of black vertices is shown in Fig. 2. In this example, the black vertices and gray vertices are extracted as a critical zone. The modified flow graph obtained from the flow graph in Fig. 2 is shown in Fig. 7. The obtained minimum cut in Fig. 7 is represented by black vertices. Figure 8 shows the flow graph before extracting the last critical zone. The last obtained minimum cut is represented by black vertices. The minimum cut consists of two intervals I_0 and I_1. The capacity of the critical zone that corresponds to I_0 can not be increased more, since it had been already extracted and HVX design style is assumed in it. While the critical zone that corresponds to I_1 is extracted. The amount of flow becomes equal to the number of nets by modifying the flow graph in terms of the critical zone that corresponds to I_1.

The proposed method stops if critical zones with overlap are found. In cases that there exist intersections of critical zones defined between different pairs of obstacles, the proposed method can not find a feasible routing pattern. For example, critical zones with overlap are shown in Fig. 9. The white squares represents the critical zone between B_0 and B_1 and the circles indicated by striped lines represents the critical zone between B_1 and B_2 when the numbers of wires that pass over are 7 and 9, respectively. The grid-point at lower-right of the obstacle B_1 is contained in both of these critical zones. In Fig. 9, the routing pattern of each critical zone generated by the procedure described in the next subsection is shown. Although a feasible routing pattern can be obtained in each critical zone, the routing pattern obtained by merging is them is infeasible. While, there exists a feasible routing pattern as shown in Fig. 10. In this situation, there is the critical zone between B_0 and B_2. The white

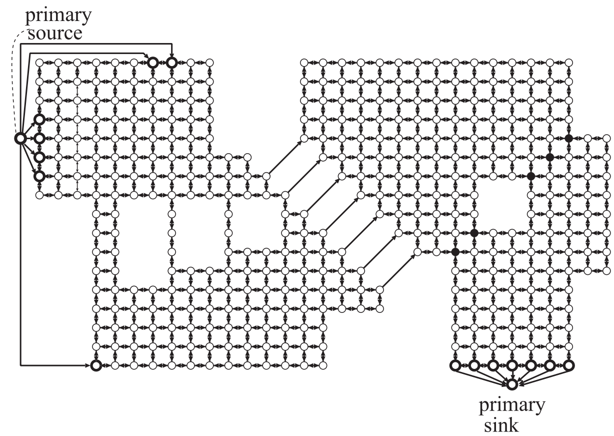


Fig. 7 Modified flow graph after extracting the first critical zone.

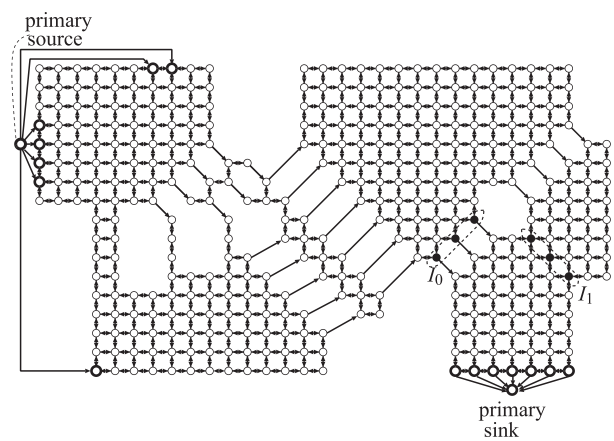


Fig. 8 Flow graph before extracting the last critical zone.

circles represents the boundary of three critical zones. Enhancement of the proposed method to handle such cases is in our future works.

5.2 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. In other words, the offset of the X tracks is set so that the Manhattan distance between the obstacle and the X tracks is 1.5.

The route of a net that passes s_i on the boundary of a critical zone 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. First, the net passes through h_i^s or v_i^s to the cross point of the track and x_i. Then, the net passes through the x_i to the cross point of h_i^t or v_i^t and x_i. Finally, the net goes to the t_i. The wire of each net uses a part of an X track in the zone and uses HV tracks if necessary. An example of routing patterns in a critical zone are shown in Fig. 11. The grid-points on the source terminal side boundary and the sink terminal side boundary are represented by circles.

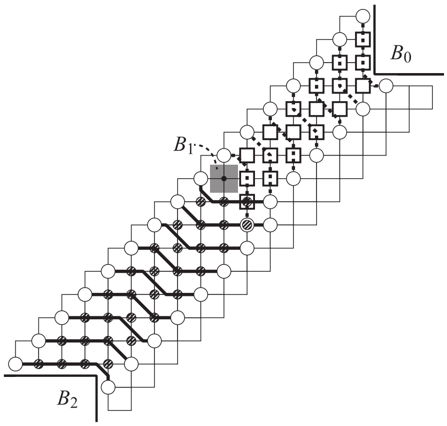


Fig. 9 Critical zones with overlap between boundaries B_0 and B_2.

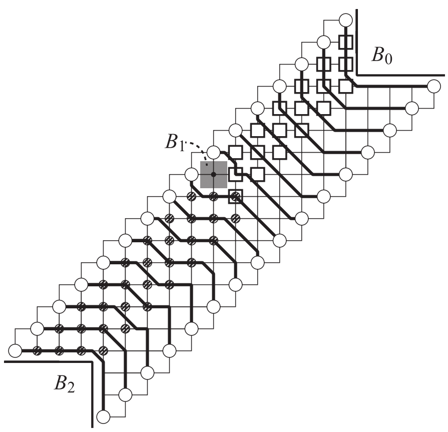


Fig. 10 A feasible routing pattern between boundaries B_0 and B_2.

In the first phase, each critical zone between B_i and B_{i+1} in terms of $\mathcal{B}$ is defined assuming that $K(i, \mathcal{B})$ nets pass over between B_i and B_{i+1}. However, the number of wires that pass over between B_i and B_{i+1}, which corresponds to the amount of flow that passes over vertices that correspond to the grid-points between B_i and B_{i+1} in the final flow graph, might be less than $K(i, \mathcal{B})$. If the number of wires that pass over between B_i and B_{i+1} is less than $K(i, \mathcal{B})$, then each critical zone is redefined using the number of wires that pass over between B_i and B_{i+1} instead of $K(i, \mathcal{B})$. The number of wires that pass over between obstacles does not necessarily be equal to the amount of flow. It may have flexibility, and the quality of routing patterns may depend on it. The assignment of the number of wires between obstacles to improve the quality of routing patterns is in our future works.

5.3 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

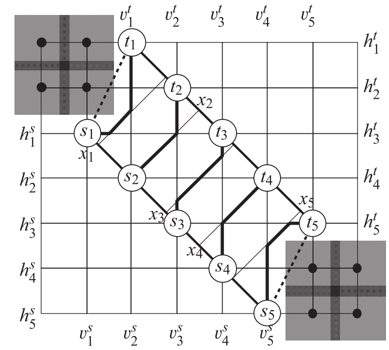


Fig. 11 HVX routing pattern in a critical zone.

realized using only HV tracks. Although the detail of the routing method used in this phase is out of scope of this paper, any routing method can be applied, even methods that cannot handle X tracks appropriately.

The routing pattern of the proposed method is obtained by merging the routing patterns of critical zones obtained in the second phase and the routing patterns of non-critical zones obtained in the third phase. Since the routing patterns of zones are merged without taking various indices into account, the routing pattern obtained might not be satisfactory. For example, it may contain a lot of unnecessary bends. The right angle of a bend might not be allowed in the design rule. However, the number of bends would be easily reduced by post processing. A right angle would be easily modified to an obtuse angle by post processing. The post processing that improves various indices is beyond the scope of this paper.

6. 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 routing patterns in non-critical zone are generated by using CAFE router [16]. In the experiments, the computation time of the proposed method to get a routing pattern is less than 1 second.

Three sets of instances that reflect the characteristics of various practical instances are generated to evaluate the proposed method. The instances in the first set contain two obstacles located diagonally. The instances in the second set contain many obstacles in the routing area. The instance in the third set contains many critical zones with complex structure generated artificially.

In the first experiment, the proposed method applies 400 instances each of which contains 50 times 40 grid-points and two obstacles B_0 and B_1. In each instance, obstacles B_0 and B_1 are put so that all nets pass over between B_0 and B_1. The x-distance and y-distance are ranged from 1 to 20, respectively. The number of nets are set to $C_{HVX}(B_0, B_1)$. Our proposed method obtains feasible HVX routing patterns in all instances. An example of the obtained routing pattern is shown in Fig. 12. In Fig. 12, both the x-distance and y-distance between B_0 and B_1 are 20, and the number of nets is 27.

In the second experiment, the proposed method is applied to four instances each of which contains 50 times 40 grid-points and many obstacles. In each instance, two obstacles B_0 and B_1 are 20. Also, obstacles whose sizes are 1 are put in the middle of the routing area of 20 times 20 grid-points. 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 are set to 27. In Table 1, the numbers of nets that are realized in HV and HVX design style are shown. Bold font is used if the number of nets that are realized in the HVX design style is larger than that in the HV design style. The obtained routing pattern of data 2 in which the obstacle ratio is 0.05 is shown in Fig. 13.

In the third experiment, the proposed method is applied to the instance generated artificially. The size of the routing area is 31 times 40, and the number of nets is 7. The critical zones in the instance are properly identified by the proposed method.

Although the detail of a routing pattern generation in non-critical zones is beyond of the scope of this paper, the routing patterns in non-critical zones are generated in order to confirm that the proposed method is useful to achieve specifications. The routing patterns in non-critical zone are generated by using CAFE router [16] by three routing modes, namely Shortest, Equal, and Longest. Routing patterns by three modes are generated so that the wirelength of each net is minimized, the difference of the wirelengths is minimized, and maximized, respectively. In CAFE router, the target length of each net in Shortest, Equal, and Longest is set to 1, the maximum wirelength of nets obtained by Shortest, and the size of routing area, respectively.

The routing pattern of three instances selected from three sets of instances are obtained by CAFE routing in three modes. In Table 2, the routing results of three instances are summarized. In this experiment, Equal mode achieves the minimum difference of the wirelengths under the assumption that HV routing is used in non-critical zones. Examples of routing pattern obtained are shown in Figs. 12, 13, and 14. In Figs. 12, 13, and 14, the routing patterns obtained by using Shortest, Equal, and Longest modes are shown, respectively. These results show that the proposed method is expected to be able to achieve various specifications efficiently.

7. Conclusion

In this paper, we introduce 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

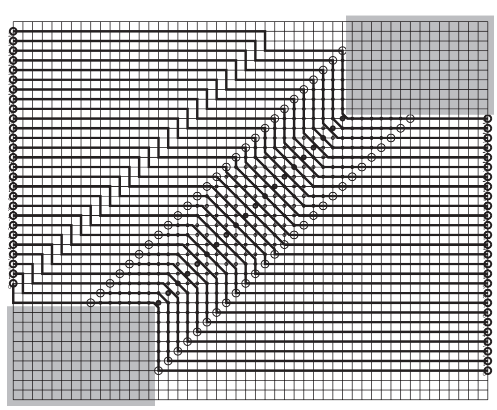


Fig. 12 HVX routing pattern of an instance in the first set by Shortest. (min = 53.9, max = 57.7, sum = 1509.2)

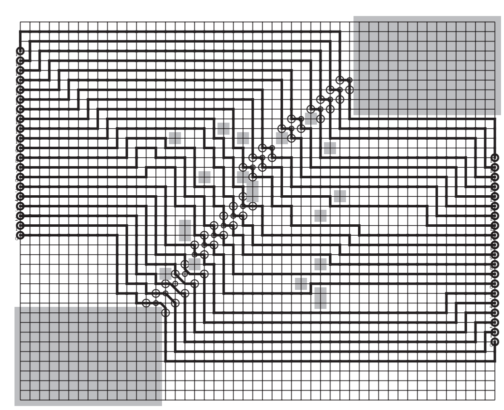


Fig. 13 HVX routing pattern of an instance in the second set by Equal. (target = 64.0, min = 63.4, max = 64.0, diff = 0.6)

Table 1 The number of nets in HV [16] and HVX design style.

Ratio	Data1		Data2		Data3		Data4		Data5	
	HV	HVX	HV	HVX	HV	HVX	HV	HVX	HV	HVX
0.05	18	18	19	**20**	18	**19**	18	**19**	19	**20**
0.10	15	15	14	**16**	13	13	13	**14**	15	**16**
0.15	12	**13**	14	14	13	**14**	13	**14**	14	14
0.20	10	10	11	11	10	10	10	10	11	11

Table 2 Experimental result.

	HV [16]	HVX	Longest				Shortest				Equal			
Instance	#net	#net	min	max	diff	sum	min	max	diff	sum	min	max	diff	sum
(a)	5	7	90.4	133.2	42.8	815.3	37.2	42.1	4.9	281.3	42.1	43.2	1.1	299.3
(b)	20	27	57.1	79.7	22.6	1663.2	53.9	57.7	3.8	1509.2	54.2	57.9	3.7	1519.2
(c)	19	20	68.0	117.7	49.7	1639.7	60.0	64.0	4.0	1243.7	63.4	64.0	0.6	1277.7

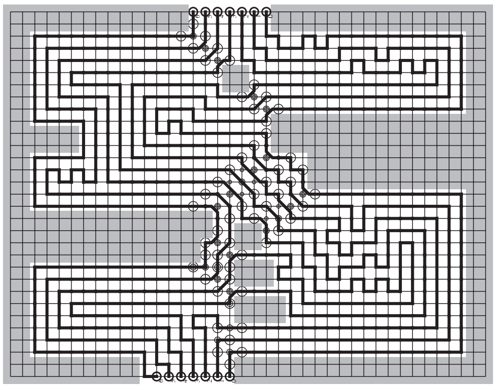


Fig. 14 HVX routing pattern of an instance in the third set by Longest. (min = 90.4, max = 133.2, sum = 815.3)

as horizontal and vertical segments. The method extracts every critical zone efficiently and then obtains a routing pattern in the critical zone using horizontal segments, vertical segments, and 45-degree segments. Although the achievement of various specifications is beyond the scope of this paper, the proposed method will be helpful for improving various indices that related to various specifications. The improvement of various indices will be achieved by applying routing tools that effectively take various indices into account in non-critical zones. The routing tools that handle horizontal and vertical segments only are applicable because non-orthogonal segments are not required. Of course, the routing tools that handle non-orthogonal segments can be used. These routing tools can be used to achieve various specifications without worrying about connectivity issues.

In the future, we intend to enhance the proposed method to enable various indices, 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] K. Shinoda, Y. Kohira, and A. Takahashi, "Single-layer trunk routing using 45-degree lines within critical areas for PCB routing," Proc. SASIMI, pp.278–283, 2010.

[2] N. Ito and H. Katagiri, "Diagonal routing in high performance microprocessor design," Proc. ASP-DAC, pp.623–629, 2006.

[3] C. Chiang, Q. Su, and C. Chiang, "Wirelength reduction by using diagonal wire," Proc. GLSVLSI, pp.104–107, 2003.

[4] S.L. Teig, "The X architecture: Not your father's diagonal wiring," Proc. SLIP'02, pp.33–37, 2002.

[5] Y. Kajitani, "The potential router," IEICE Technical Report, VLD2006–129, 2007. (In Japanese).

[6] M. Inagi, Y. Takashima, and Y. Kajitani, "Escape fitting between a pair of pin-sets," IEICE Technical Report, VLD2006–130, 2007. (In Japanese).

[7] T. Yan and M. Wong, "Untangling twisted nets for bus routing," Proc. ICCAD, pp.396–400, 2007.

[8] L. Luo and M. Wong, "Ordered escape routing based on Boolean satisfiability," Proc. ASP-DAC, pp.244–249, 2008.

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

[10] R. Pinter, "On routing two-point nets across a channel," Proc. DAC, pp.894–902, 1982.

[11] C. Hsu, "Channel river routing algorithm," Proc. DAC, pp.241–246, 1983.

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

[13] M. Ozdal and M. Wong, "A length-matching routing algorithm for high performance printed circuit boards," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.25, no.12, pp.2784–2794, 2006.

[14] M. Ozdal and M. Wong, "Algorithmic study of single-layer bus routing for high-speed boards," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.25, no.3, pp.490–503, 2006.

[15] T. Yan and M. Wong, "BSG-route: A length-constrained routing scheme for general planar topology," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.28, no.11, pp.1679–1690, 2009.

[16] Y. Kohira and A. Takahashi, "CAFE router: A fast connectivity aware multiple nets routing algorithm for routing grid with obstacles," IEICE Trans. Fundamentals, vol.E93-A, no.12, pp.2380–2388, Dec. 2010.

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

Kyosuke Shinoda received his B.E., and M.E. degrees from Tokyo Institute of Technology, Tokyo, Japan, in 2008, and 2010, respectively. He is currently with the Department of Communications and Integrated Systems in Tokyo Institute of Technology as a Ph.D. student since 2010. His research interests are in VLSI 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 ACM, 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.