論文 / 著書情報
Article / Book Information

Title	SDG Channel Routing to Minimize Wirelength for Generalized Channel
Authors	Zezhong Wang, Masayuki Shimoda, Atsushi Takahashi
Citation	IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol. E108-A, No. 3, pp. 500-508
Pub. date	2025, 3
Copyright	Copyright(C)2025 IEICE

IEICE TRANSACTIONS

on Fundamentals of Electronics, Communications and Computer Sciences

VOL. E108-A NO. 3
MARCH 2025

A PUBLICATION OF THE ENGINEERING SCIENCES SOCIETY
The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER Special Section on VLSI Design and CAD Algorithms

SDG Channel Routing to Minimize Wirelength for Generalized Channel*

Zezhong WANG[†a)], *Nonmember*, **Masayuki SHIMODA**[†b)], *Student Member*, *and* **Atsushi TAKAHASHI**[†c)], *Fellow*

SUMMARY In this paper, SDG channel routing algorithm for generalized channel is proposed. In generalized channel, it is assumed that horizontal routing capacity is tight and that all pins of a net are needed to be connected by a Single Trunk Steiner Tree. Also, it is requested to reduce the total length of nets to accomplish the routing, and our goal is to reduce the total vertical length as much as possible. Our proposed algorithm determines the track assignment of nets iteratively according to the net priority that is defined to reduce the vertical length. In experiments, it is confirmed that the vertical length is reduced by around 30% to 50% compared with the track assignment by Left-Edge, and that it is very close to a lower bound.
key words: *generalized channel, symmetric difference, length minimization*

1. Introduction

Routing is an important design step, and a better routing strategy is crucial to obtain a high-performance chip [2]. Modern advanced chips which contain a huge number of regularly placed electronic components such as sensors and memory require sophisticated routing strategies. For example, obstacles corresponding to bonding pads used for communication between stacked chips are defined in a routing layer of a complementary metal-oxide-semiconductor (CMOS) chip for flash memory [3]–[5]. In such chips, routing layers that contain a huge number of regularly placed obstacles are included (See Fig. 1). Even though conventional routing algorithms are successful when routing layers contain few or no obstacles, they cannot efficiently utilize these routing layers. A routing strategy that utilizes these routing layers efficiently and that can provide an earlier evaluation of chip design is required by such modern advanced chips.

A routing layer where horizontal capacity is tight is called a critical routing layer here, and the routing layer is modeled as generalized channel as in [1], [6]. In the generalized channel, horizontal tracks that span the area of the routing layer while avoiding obstacles and preserving horizontal capacity are defined in advance. Pins of nets are located inside the channel, and vertical wires are routed in other routing layers whose capacity is not so tight. Specifically, due to tight horizontal routing capacity, we assume that all pins of a net are needed to be connected by a Single Trunk Steiner Tree (STST).

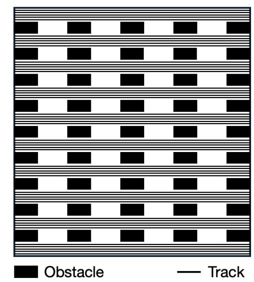


Fig. 1 Routing layer with regularly placed obstacles and horizontal tracks.

In this paper, we propose the Symmetric Difference General (SDG) channel routing algorithm for the Generalized Channel Routing Problem (GCRP). In order to complete the routing in routing layers for vertical wires as well as the routing for the critical routing layer in GCRP, a small vertical congestion is required. SDG connects nets by STST and achieves a small vertical congestion efficiently. SDG is based on the greedy routing framework, called Greedy Routing Framework with Guarantee (GRFG). GRFG guarantees to complete the routing when the required number of tracks are given and no vertical constraint is enforced. In order to achieve a small vertical congestion, SDG assigns nets to tracks according to the net priority which is defined based on the symmetric difference (SD) of pin locations of a net in terms of a track location. Although SDG can accomplish routing efficiently, it is a heuristic, and does not guarantee optimality in terms of the total vertical length. In order to get a local optimal solution, simple post-processing is also introduced.

The rest of the paper is organized as follows: The background and the definition of problem are described in Sect. 2. The proposed SDG and post-processing are given in Sect. 3. The experimental results of comparing SDG with others are shown in Sect. 4, and the conclusion is presented in Sect. 5.

2. Preliminaries

2.1 Problem Background

A routing layer whose horizontal capacity is tight due to a large number of regularly placed components that act as obstacles during the routing process exists in some types of modern chips, and the routing layer is modeled as a gen-

Manuscript received March 5, 2024.
Manuscript revised June 28, 2024.
Manuscript publicized January 9, 2025.
†Tokyo Institute of Technology, Tokyo, 152-8550 Japan.
*The preliminary version was presented at [1].
a) E-mail: zezhong@eda.ict.e.titech.ac.jp
b) E-mail: shimoda@eda.ict.e.titech.ac.jp
c) E-mail: atsushi@ict.e.titech.ac.jp
DOI: 10.1587/transfun.2024VLP0001

eralized channel with predefined horizontal tracks spanning the whole area, ensuring obstacle avoidance, and preserving the horizontal capacity for routing. As a result, no obstacles exist in the generalized channel.

The definition of the generalized channel is similar to the classical channel defined mainly for standard cell design with a few metal layers. Routing problems of the generalized channel and classical channel are called generalized channel routing problem (GCRP) and classical channel routing problem (CCRP), respectively [6].

In CCRP where pins are restricted on channel boundaries, several algorithms [7]–[18] have been developed. For example, routing algorithms for rectangle channel using HV-rule in which each layer uses only either horizontal or vertical segments have been discussed in [7]–[12], [14], while a routing algorithm for L-shaped channel using three-layer has been proposed in [13]. In [15]–[18], routing algorithms for bottleneck channel in which horizontal segments share the same track in adjacent layers to satisfy many connection requirements in small region have been proposed. However, these algorithms are not directly applicable to GCRP. GCRP allows pins to exist at any location within the channel. The objective of GCRP, in this paper, is to accomplish routing in the generalized channel and to achieve a small vertical length as much as possible. Even though various algorithms have been proposed for CCRP, no relevant algorithms for GCRP have been developed until recently due to the lack of needs for GCRP.

Steiner Minimum Tree (SMT) is often used to find a desirable route for a multi-pin net, and Rectilinear Steiner Minimum Tree (RSMT) [19]–[21] has been widely used in VLSI routing. However, RSMT often consists of several horizontal trunks. It is not well suit our assumption that horizontal routing capacity at the critical routing layer is tight. Dogleg routing [8] divides the horizontal trunk of a net into several segments and assigns them to different tracks. When vertical constraints exist on nets in CCRP, dogleg routing efficiently reduces the total number of tracks required. However, dogleg increases the number of vertical segments which leads to the increase of the number of vias to connect horizontal and vertical segments, and may interfere to complete routing in GCRP.

In contrast, Single-Trunk Steiner Tree (STST) contains only one trunk, minimizing the use of horizontal routing resources and requiring the minimum number of vertical segments as well as the minimum number of vias under the constraint. Therefore, we assume that all pins of a net are to be connected by STST.

In this paper, we assume that GCRP contains no vertical constraint. This assumption is reasonable in the situation where generalized channels are defined. Nets in the netlist in the inputs are typically subnets, and pins of nets are defined virtually. They might be defined on a module or on an adjacent layer, and pin locations given as inputs are not absolute. Vertical segments required as the result of horizontal trunk assignment are assumed to be realized in the other routing layers. In case that conflicts among vertical segments occur, they will be resolved by utilizing flexibility in module design and location in followed detailed placement and routing. Therefore, it is not strange even if it is assumed that pin locations are a little flexibility enough to eliminate the vertical constraints, but not enough to relax the horizontal connection demands.

In GCRP, Left-Edge [7] can effectively find a better track assignment of nets that uses the minimum number of tracks by the leftmost principle if no vertical constraints exist. However, Left-Edge is not good enough to reduce vertical congestion. The algorithm proposed in [14] substantially reduces the total (vertical) length compared to Left-Edge. However, it works well in CCRP when vertical constraints are imposed, but is not directly applicable in GCRP. BCA routing which dedicates for 2-pin nets in GCRP was proposed in [6], however, a routing algorithm for multi-pin nets in GCRP is desired. In [22], routing algorithms in GCRP with wire various width wires used are discussed, whereas this paper focuses on routing using unit-width wires. They find better combinations of horizontal segments to utilize critical routing layer well, but the minimization of the total vertical length is out of concern.

2.2 Generalized Channel Routing Problem

The objective of the Generalized Channel Routing Problem (GCRP) in this paper is to minimize the total vertical length while assigning all nets to the given tracks, ensuring horizontal constraints are satisfied. The problem seems to be NP-hard, but the time complexity of this problem is not known as far as we know.

The problem GCRP is formulated as an assignment task, where a set of nets $N_{\text{in}} = \{n_1, n_2, \ldots, n_m\}$ and a set of horizontal tracks $T_{\text{in}} = \{t_1, t_2, \ldots, t_k\}$ of a channel are given as input.

A net is a set of pins to be connected by wire segments. A pin p location is represented by $(x(p), y(p))$. The minimum and maximum x-coordinate of all its pins of net n are denoted by $x_{\min}(n) = \min_{p\in n} x(p)$ and $x_{\max}(n) = \max_{p\in n} x(p)$, respectively. Similarly, $y_{\min}(n)$ and $y_{\max}(n)$ are defined. The x-distance and the total y-distance of net n are defined as $d^x(n) = x_{\max}(n) - x_{\min}(n)$ and $d^y(n) = \sum_{p\in n} |y(p) - y(p_{\text{mid}})|$ where p_{mid} is a pin which has $\lceil |n|/2 \rceil$-th largest y-coordinate among pins of net n, respectively. Note that, in STST, the x-distance of a net is the horizontal length of the net, and that the total y-distance of a net gives a lower bound of the total vertical length of the net.

The interval of a net n is defined as $I(n) = [x_{\min}(n), x_{\max}(n)]$. For a set of nets N, the set of x-coordinates of intervals of nets in N is denoted by $I(N) = \{x \mid x \in I(n), n \in N\}$ and $x_{\max}(N) = \max_{p\in n\in N} x(p)$. If N is empty, then $x_{\max}(N)$ is defined as $-\infty$.

A horizontal constraint between two nets n_i and n_j is defined if $I(n_i) \cap I(n_j) \neq \emptyset$. In case that each net is connected by STST, the trunks of nets with horizontal constraints cannot be assigned to the same track. Also, a vertical constraint between two nets is usually defined if there exist $p_1 \in n_1$

and $p_2 \in n_2$ such that $x(p_1) = x(p_2)$. A vertical constraint restricts the assignment of horizontal segments to tracks to prevent the collision of vertical wire segments of nets. However, according to the assumption on GCRP in this paper, we regard no vertical constraint exists, even when the x-coordinate of pins $p_1 \in n_1$ and $p_2 \in n_2$ are the same, that is, $x(p_1) = x(p_2)$.

The length of net n when n is assigned to track t, is formulated as $w(n,t) = w^x(n) + w^y(n,t)$, where $w^x(n) = x_{\max}(n) - x_{\min}(n)$ and $w^y(n,t) = \sum_{p\in n} |y(p) - y(t)|$, where $y(t)$ is the y-coordinate of track t. Since the horizontal length w^x is independent of track assignment, the length is evaluated only on the vertical length w^y.

A track assignment a that assigns net n to a track t is represented by $a(n) = t$. The total vertical length of nets N by track assignment a is represented by $w^y(N,a) = \sum_{n\in N} w^y(n,a(n))$. It is used to evaluate the vertical congestion in this paper.

The objective of GCRP in this paper is the minimization of the total vertical length while assigning all nets N_{in} to the given tracks T_{in} to satisfy horizontal constraints, as defined by

$$\begin{aligned} \min \quad & w^y(N_{\text{in}}, a) = \sum_{n\in N_{\text{in}}} w^y(n, a(n)), \\ \text{s.t.} \quad & a(n) \in T_{\text{in}}, \forall n \in N_{\text{in}}, \\ & I(n_i) \cap I(n_j) = \emptyset \text{ if } a(n_i) = a(n_j), \forall n_i, n_j \in N_{\text{in}}. \end{aligned}$$

Let N $(\subseteq N_{\text{in}})$ and T $(\subseteq T_{\text{in}})$ be a set of unassigned nets and a set of tracks to which no net is assigned so far and to which N will be assigned, respectively. *Density* at x of N is the number of nets in N whose interval contains x and is denoted by $d(x,N) = |\{n \in N \mid x \in I(n)\}|$, and the maximum density of N is defined as $D(N) = \max_x d(x,N)$. *Capacity* of T is the number of tracks in T and is denoted by $|T|$. *Slack* at x of N and T is defined as the difference between capacity and density at x and is represented by $s(x,T,N) = |T| - d(x,N)$. Note that there is no feasible track assignment of N to T if $s(x,T,N) < 0$ for some x.

A set of nets $N(\subseteq N_{\text{in}})$ cannot be assigned to the same track simultaneously if $I(n_i) \cap I(n_j) \neq \emptyset, \exists n_i, n_j \in N$. It is said that N meets *horizontal constraint* (HC) if $I(n_i) \cap I(n_j) = \emptyset, \forall n_i, n_j \in N$. For any set of nets N that meets HC, if $I(N) \cap I(n) = \emptyset$, then $N \cup \{n\}$ meets HC. A feasible track assignment of N to T where no vertical constraint is imposed on N exists if and only if $D(N) \leq |T|$ is satisfied. We call $D(N) \leq |T|$ the *density constraint* (DC) of N for T. It is said that N meets DC for T if $D(N) \leq |T|$.

We assume that a set of given nets N_{in} and a set of tracks T_{in} meet DC, and that no vertical constraint exists among nets in N_{in}.

In order to accomplish the routing by determining track assignment track by track in greedy manner, the critical zone of a channel has to be taken care. It is defined in terms of slack which is the difference of capacity and density.

Critical zone (CZ) of channel for N and T is the set of x-coordinates of channel where there is no slack and is denoted as $Z(T,N) = \{x \mid s(x,T,N) \leq 0\}$. A set of nets $N'(\subseteq N)$ is said to cover the critical zone if $Z(T,N) \subseteq I(N')$. In any feasible track assignment a of N to T, for any $x \in Z(T,N)$ and for any track $t \in T$, there exists net $n \in N$ that contains x at t, that is, $a(n) = t$ and $x \in I(n)$.

Let $N'(\subseteq N)$ be a set of nets. In case that N meets DC for T, N' meets HC, and $Z(T,N) \subseteq I(N')$, if nets in N' are assigned to track $t(\in T)$, then $N \setminus N'$ meets DC for $T \setminus \{t\}$. A set of nets $N'(\subseteq N)$ is said to be *CZ-cover* for N and T if and only if N' meets HC and covers the critical zone of channel for N and T.

3. SDG Channel Routing Algorithm

Symmetric Difference General channel routing algorithm (SDG) is proposed to minimize the total vertical length in GCRP. SDG is a greedy heuristic, and does not guarantee the optimality in terms of the total vertical length. In order to get a local optimal solution, simple post-processing is also introduced.

3.1 Greedy Routing Framework with Guarantee

In GCRP, the primal objective is to accomplish routing under tight horizontal capacity condition. The Greedy Routing Framework with Guarantee (GRFG) is designed to complete routing in GCRP when N_{in} meets DC for T_{in} and no vertical constraint exists among nets in N_{in}. SDG follows GRFG, and is guaranteed to complete routing.

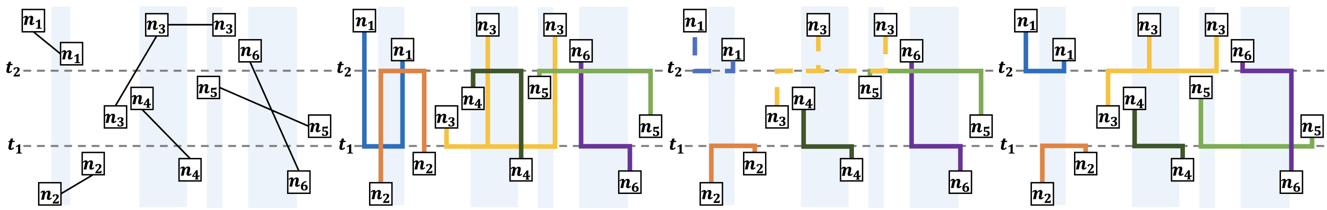
In order to accomplish routing when N_{in} meets DC for T_{in} and no vertical constraint exists among nets in N_{in}, a set of nets assigned to a track in T_{in} has to be a CZ-cover. In GRFG, CZ-aware net selection is introduced to find a CZ-cover for each track. In each track, GRFG assigns trunks one-by-one from a start position toward left and right directions so that CZ is covered by higher priority nets as much as possible. The pseudo code of GRFG is given in Algorithm 1. For simplicity, in the pseudo code, a start point of each track is set to the leftmost point of the track, that is, trunks are assigned from left to right in each track. The following discussion follows this pseudo code.

A set of nets N' is said to be *CZ-aware* in track assignment of N to T if $[-\infty, x_{\max}(N')] \cap Z(T,N) \subseteq I(N')$. For any CZ-aware N', CZ at the left of the interval of a net in

Algorithm 1 GRFG (left-to-right in each track)

Require: set of nets N_{in}, set of tracks $T_{\text{in}} = \{t_1, t_2, \ldots, t_k\}$
Ensure: track assignment $a(n)$ for all $n \in N_{\text{in}}$

$i \leftarrow 0$, $N \leftarrow N_{\text{in}}$, $T \leftarrow T_{\text{in}}$
while $N \neq \emptyset$ **do**
  $T \leftarrow T \setminus \{t_i\}$, $i \leftarrow i + 1$, $x \leftarrow -\infty$, $U \leftarrow N$
  while $U \neq \emptyset$ **do**
    $n \leftarrow \text{Top\_Priority}(U, t_i)$, $U \leftarrow U \setminus \{n\}$
    if $x \leq x_{\min}(n)$ **and** $(x, x_{\min}(n)) \cap Z(T,N) = \emptyset$ **then**
      $a(n) \leftarrow t_i$, $x \leftarrow x_{\max}(n)$, $N \leftarrow N \setminus \{n\}$, $U \leftarrow N$
    end if
  end while
end while



(a) Input and critical zone (b) Priority LEP $(n_1, n_2, n_3, n_4, n_5, n_6)$ (c) Priority SDP (no CZ-aware) $(n_2, n_4, n_6, n_5, n_1, n_3)$ (d) Priority SDP $(n_2, n_4, n_6, n_5, n_1, n_3)$

Fig. 2 Track assignments by GRFG.

N' is covered by $I(N')$, and it is guaranteed that, for any x in CZ not covered by $I(N')$, there is net $n \in N \setminus N'$ such that $N' \cup \{n\}$ meets HC and $x \in I(n)$. In case that CZ-aware N' covers CZ, it is also a CZ-cover.

In *CZ-aware net selection* for CZ-aware N', net n ($\notin N'$) is selected to add to N' only if $(x_{\max}(N'), x_{\min}(n)) \cap Z(T,N) = \emptyset$. That is, the slack is positive for all x-coordinates in open interval $(x_{\max}(N'), x_{\min}(n))$. CZ-aware net selection ensures that there is no uncovered CZ to the left of a net whenever the net is assigned, and that $N' \cup \{n\}$ is CZ-aware.

In step 5 of Algorithm 1, `Top_Priority` gives the top priority net for t_i among unassigned nets. It is not assigned to t_i if uncovered CZ to the left remains. This is checked by step 6. Note that, even if it is skipped to be assigned to t_i at this point, it may be assigned to t_i after some low priority nets are assigned to t_i to cover CZ.

Left-Edge [7] in which priority LEP defined in terms of the leftmost principle is used is a variation of GRFG if nets are assigned track-by-track. Note that the selection of highest priority net of LEP among unassigned nets is always a CZ-aware net selection.

Figure 2 shows an example of GRFG. In Fig. 2(a), the set of nets and tracks which are given as the input are shown. Also, CZ is shown by shadow region. In Fig. 2(b), track assignment by GRFG according to the priority LEP corresponding to Left-Edge principle is shown. In Figs. 2(c) and (d), priority SDP for track t_1 is used. The definition of priority SDP is defined in Sect. 3.3. In Fig. 2(c), track assignment without CZ-aware is shown. First, according to SDP, n_2, n_4 and n_6 are assigned to t_1. Then, according to SDP, n_5 is assigned to t_2. Due to GRFG that assigns trunks one-by-one from left to right in each track, n_1 and n_3 cannot be assigned to t_2 because n_5 is already assigned to t_2, and n_1 and n_3 are to the left of n_5, requiring one more track. In track t_1, uncovered CZ between n_4 and n_6 remains. To prevent a failure, it is necessary to avoid leaving any CZ uncovered. In Fig. 2(d), track assignment by GRFG is shown. Due to CZ-aware net selection, n_5 is assigned to t_1 instead of n_6, and completes routing. In track t_2, n_6 and n_3 cannot be selected at the beginning, and n_1 is selected first. Then, n_3 and n_6 follow.

The following theorem guarantees that Algorithm 1 finds a track assignment that satisfies HC when N_{in} meets DC for T_{in} and no vertical constraint is enforced.

Theorem. *The number of tracks used by Algorithm 1 for N_{in} to T_{in} is at most $|T_{\text{in}}|$ if $D(N_{\text{in}}) \leq |T_{\text{in}}|$ and there are no vertical constraints.*

Proof. Assume contrary that there is a net ($\in N_{\text{in}}$) not assigned to any track ($\in T_{\text{in}}$) by Algorithm 1. Let $k = |T_{\text{in}}|$, and let N_i be the set of nets assigned to track t_i ($1 \leq i \leq k$) by Algorithm 1. Let $n^* \in N_{\text{in}}$ be a net where $x_{\min}(n^*)$ ($= x_1$) is the minimum among nets not assigned to a track. That is, all trunks of nets are assigned within k tracks to the left of x_1. There is a track in T_{in} where no trunk of a net is assigned at x_1, otherwise, it contradicts the assumption that $D(N_{\text{in}}) \leq |T_{\text{in}}|$. Let $t_j \in T_{\text{in}}$ be the track with the largest index among them.

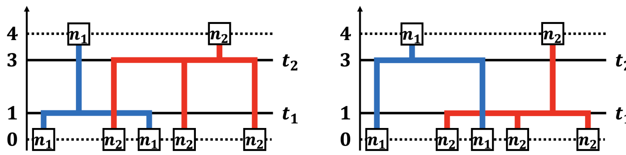
Let $N = N_j \cup N_{j+1} \cup \cdots \cup N_k$ and $T = \{t_j, t_{j+1}, \ldots, t_k\}$. Note that $d(x_1, N) \geq |T|$ since x_1 of track $t_{j'}$ ($j+1 \leq j' \leq k$) is covered by the net assigned by Algorithm 1 and net n^* contributes to $d(x_1, N)$ in addition. Therefore, $s(x_1, T, N) = |T| - d(x, N) \leq 0$. That is, x_1 is in CZ of N for T, even though x_1 may not be in CZ of N_{in} for T_{in}.

Note that N_j is CZ-aware due to CZ-aware net selection by Algorithm 1, and there are no trunks of nets in N_j to the right of x_1 on track t_j since x_1 is in CZ. Let $x_0 = x_{\max}(N_j)$. Then, $x_0 < x_1$. Also, $s(x, T, N) > 0$ for $x_0 < x < x_1$ since all trunks of nets are assigned within k tracks to the left of x_1. However, this contradicts the behavior of Algorithm 1 since net n^* satisfies the condition to be assigned to t_j as $(x_0, x_1) \cap Z(T, N) = \emptyset$, and n^* can be assigned to t_j since there are no vertical constraints. □

3.2 Symmetric Difference (SD) of Pins of a Net

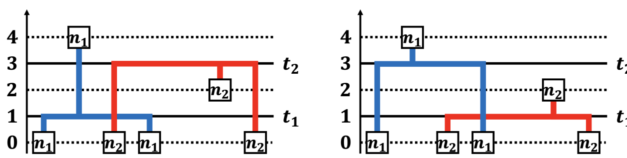
The vertical length of a net depends on the track to be assigned. In case that all pins of net n are connected by STST, it is $w^y(n,t) = \sum_{p \in n} |y(p) - y(t)|$ where t is the track to be assigned. The Symmetric Difference (SD) of net n in terms of track t is defined as follows: $\mathrm{SD}(n,t) = |\{p \in n \mid y(p) < y(t)\}| - |\{p \in n \mid y(p) > y(t)\}|$. That is, it is the number of pins of n below t minus the number of pins of n above t. Note that the smaller $|\mathrm{SD}(n,t)|$ is, the smaller the vertical length of n when n is assigned to t is, and that the vertical length of n is minimum if $\mathrm{SD}(n,t) = 0$. In order to achieve a smaller vertical length, it is preferable to assign a net to a track with smaller $|\mathrm{SD}(n,t)|$.

In Fig. 3, two track assignments of n_1 and n_2 to t_1 and t_2 are given where $w^y(n_1,t_1) = 5$, $w^y(n_1,t_2) = 7$, $w^y(n_2,t_1) =$



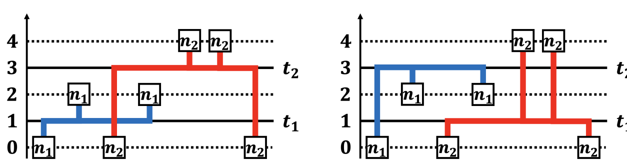
(a) $w^y = 15$ (b) $w^y = 13$

Fig. 3 $\mathtt{SD}(n_1, t_1) = 1 < \mathtt{SD}(n_2, t_1) = 2$, $\mathtt{SD}(n_1, t_2) = 1$, $\mathtt{SD}(n_2, t_2) = 2$.



(a) $w^y = 12$ (b) $w^y = 10$

Fig. 4 $\mathtt{SD}(n_1, t_1) = \mathtt{SD}(n_2, t_1) = 1$, $\mathtt{SD}(n_1, t_2) = 1 < \mathtt{SD}(n_2, t_2) = 3$.



(a) $w^y = 11$ (b) $w^y = 13$

Fig. 5 $\mathtt{SD}(n_1, t_1) = -1 < \mathtt{SD}(n_2, t_1) = 0$, $\mathtt{SD}(n_1, t_2) = 3 > \mathtt{SD}(n_2, t_2) = 0$.

6, and $w^y(n_2, t_2) = 10$. Even though the vertical lengths of n_1 and n_2 are smaller if these are assigned to t_1, both of them cannot be assigned to t_1 simultaneously due to HC. The total vertical length is smaller if n_2 is assigned to t_1 (Fig. 3(b)) rather than n_1 is assigned to t_1 (Fig. 3(a)). The increase of length of net n when it is assigned to an upper track t' instead of track t is proportional to $\mathtt{SD}(n, t)$ if $\mathtt{SD}(n, t) = \mathtt{SD}(n, t')$. That is, in order to achieve a small total length, it is typically preferable to assign nets with large SD to lower tracks and nets with small SD to upper tracks. This example suggests the strategy "assign nets with large SD to lower tracks" to achieve a small total vertical length.

In Fig. 4, two track assignments of n_1 and n_2 to t_1 and t_2 are given where $w^y(n_1, t_1) = 5$, $w^y(n_1, t_2) = 7$, $w^y(n_2, t_1) = 3$, and $w^y(n_2, t_2) = 7$. The total vertical length is smaller if n_2 which has larger SD in terms of t_2 is assigned to t_1 (Fig. 4(b)) rather than n_1 is assigned to t_1 (Fig. 4(a)), though SDs in terms of t_1 are the same. This example suggests the strategy "assign net n with large SD-sequence $(\mathtt{SD}(n, t_1), \mathtt{SD}(n, t_2), \ldots, \mathtt{SD}(n, t_m))$ in lexicographical order to lower track" when $y(t_1) < y(t_2) < \cdots < y(t_m)$.

In Fig. 5, two track assignments of n_1 and n_2 to t_1 and t_2 are given where $w^y(n_1, t_1) = 3$, $w^y(n_1, t_2) = 5$, and $w^y(n_2, t_1) = w^y(n_2, t_2) = 8$. The total vertical length is smaller if n_1 which has smaller SD in terms of t_1 is assigned to t_1 (Fig. 5(a)) rather than n_2 is assigned to t_1 (Fig. 5(b)). This example shows that the strategies suggested so far do not always work well.

3.3 Priority by Symmetric Difference

Here, a priority of nets in N to track t_i ($1 \leq i \leq k$) where $y(t_i) < y(t_{i+1}) < \cdots < y(t_k)$ is discussed.

The SD-sequence of net n for track t_i is defined according to SDs of n as

$$(\mathtt{SD}(n, t_i), \mathtt{SD}(n, t_{i+1}), \ldots, \mathtt{SD}(n, t_k)).$$

As we discussed, it is preferable to assign nets with large SD-sequence for track t_i in lexicographical order to t_i if $y(t_i) < y(t_{i+1}) < \cdots < y(t_k)$. Thus, the priority SDP for track t_i is defined as the descending lexicographical order of SD-sequence of net for t_i. Ties are broken arbitrary, but the net index is used here.

For example, in Fig. 2, SD-sequence of n_1, n_2, n_3, n_4, n_5, and n_6 for t_1 are $(-2, -2)$, $(2, 2)$, $(-3, -1)$, $(0, 2)$, $(-2, 2)$, and $(0, 0)$, respectively, and for t_2 are (-2), (2), (-1), (2), (2), and (0), respectively. SDPs for t_1 and t_2 are $(n_2, n_4, n_6, n_5, n_1, n_3)$ and $(n_2, n_4, n_5, n_6, n_3, n_1)$, respectively. In Fig. 4, SD-sequence of n_1 and n_2 for t_1 are $(\mathtt{SD}(n_1, t_1), \mathtt{SD}(n_1, t_2)) = (1, 1)$ and $(\mathtt{SD}(n_2, t_1), \mathtt{SD}(n_2, t_2)) = (1, 3)$, respectively, and SDP for t_1 is (n_2, n_1).

In [6], BCA algorithm was introduced for two-pin nets. It classifies two-pin nets in three categories Below, Cross, and Above in terms of track to be assigned which correspond to SD is 2, 0, and -2 of the net in terms of the track, respectively. In that sense, the priority BCA used in BCA algorithm is a special case of SDP.

3.4 SDG Channel Routing Algorithm

Symmetric Difference General channel routing algorithm (SDG) is given in Algorithm 2. This algorithm aims to complete routing when the given set of nets meets DC and no vertical constraints are enforced, and to minimize length by GRFG with the priority derived by the symmetric difference of pins of a net.

In Algorithm 2, the given set of tracks $T_{\text{in}} = \{t_1, t_2, \ldots, t_m\}$ where $y(t_1) < y(t_2) < \ldots < y(t_m)$ is assumed. In step 5, `Largest_SD` gives the top priority net for

Algorithm 2 SDG Channel Routing Algorithm

Require: set of nets N_{in}, set of tracks $T_{\text{in}} = \{t_1, t_2, \ldots, t_m\}$ where $y(t_1) < y(t_2) < \ldots < y(t_m)$
Ensure: track assignment $a(n)$ for all $n \in N_{\text{in}}$

1: $i \leftarrow 0$, $N \leftarrow N_{\text{in}}$, $T \leftarrow T_{\text{in}}$
2: **while** $N \neq \emptyset$ **do**
3: $\quad T \leftarrow T \setminus \{t_i\}$, $i \leftarrow i + 1$, $x \leftarrow -\infty$, $U \leftarrow N$
4: $\quad$ **while** $U \neq \emptyset$ **do**
5: $\quad\quad n \leftarrow \text{Largest\_SD}(U, t_i)$, $U \leftarrow U \setminus \{n\}$
6: $\quad\quad$ **if** $(x, \infty) \cap Z(T, N) = \emptyset$ **and** $\mathtt{SD}(n, t_i) < 0$ **then**
7: $\quad\quad\quad$ **break**
8: $\quad\quad$ **end if**
9: $\quad\quad$ **if** $x \leq x_{\min}(n)$ **and** $(x, x_{\min}(n)) \cap Z(T, N) = \emptyset$ **then**
10: $\quad\quad\quad a(n) \leftarrow t_i$, $x \leftarrow x_{\max}(n)$, $N \leftarrow N \setminus \{n\}$, $U \leftarrow N$
11: $\quad\quad$ **end if**
12: $\quad$ **end while**
13: **end while**

t_i among unassigned nets in terms of SD-sequence of net for t_i, that is, the top net in SDP for t_i.

In GRFG, nets are assigned to tracks as early as possible if HC is satisfied. Therefore, in case that nets are assigned to from lower tracks to upper tracks as in Algorithm 2, a net tends to be assigned to a lower track. A smaller vertical length of a net is achieved when it is assigned to track t with smaller $|\mathtt{SD}(n,t)|$. The higher the track, the larger SD of a net. In case that SD of a net for a track is negative, assigning it to an upper track may shorten the vertical length of the net. Therefore, in Algorithm 2, the top priority net may not be assigned to a track even though it can be assigned to the track without violating HC. The assignment of nets to a track is terminated at step 6 if CZ of the track is covered by nets assigned to the track so far and no net whose SD is non-negative can be assigned to the track without violating HC under GRFG.

3.5 Post-Processing for Length Reduction

SDG can complete routing efficiently, however, it is a heuristic, and does not guarantee optimally in terms of the total vertical length. Here, the post-processing outlined in Algorithm 3 that iteratively reduces the vertical length in greedy manner is introduced.

For each net n whose current SD value is not zero, the post-processing examines all tracks t where the absolute SD value of n is less than the current absolute SD value of n in terms of the current track. Should no intersections exist between the interval $I(n)$ and the intervals of nets assigned to track t, the post-processing proceeds to `shift` the net to track t. This `shift` is enacted if the new assignment a' results in a decrease in length, and upon success, the `flag` is reset to signal the possibility of additional reduction.

When a `shift` is not viable due to intersecting intervals, the algorithm explores the possibility of an `exchange` swapping the position of net n with another net n' currently assigned to track t. This action is executed only if it maintains all necessary constraints (making it `feasible`) and leads to a reduced length.

This iterative process terminates if no modification is made during a loop. The resulting set of track assignments is a local optimal solution.

For the example shown in Fig. 5, post-processing outputs the better solution shown in Fig. 5(a), when SDG outputs the solution shown in Fig. 5(b).

Algorithm 3 Post-Processing for SDG

Require: track assignment $a(n)$ for all $n \in N_{\text{in}}$
Ensure: track assignment $a(n)$ for all $n \in N_{\text{in}}$
$w \leftarrow w^y(N_{\text{in}}, a)$, $\mathit{flag} \leftarrow \mathrm{T}$
while $\mathit{flag} == \mathrm{T}$ **do**
  $\mathit{flag} \leftarrow \mathrm{F}$
  for all $n \in N$ **do**
    for all $t \in \{t \in T_{\text{in}} \mid |\mathtt{SD}(n,t)| < |\mathtt{SD}(n,a(n))|\}$ **do**
      if $I(n) \cap I(\{n' \mid a(n') = t\}) = \emptyset$ **then** ▷ try shift
        $a' \leftarrow \mathrm{shift}(a, n, t)$
        if $w^y(N_{\text{in}}, a') < w$ **then**
          $w \leftarrow w^y(N_{\text{in}}, a')$, $a \leftarrow a'$, $\mathit{flag} \leftarrow \mathrm{T}$
        end if
      else ▷ try exchange
        $a' \leftarrow \mathrm{exchange}(a, n, t)$
        if feasible(a') **and** $w^y(N_{\text{in}}, a') < w$ **then**
          $w \leftarrow w^y(N_{\text{in}}, a')$, $a \leftarrow a'$, $\mathit{flag} \leftarrow \mathrm{T}$
        end if
      end if
    end for
  end for
end while

4. Experimental Results

We demonstrated experiments of comparing SDG with others which were developed in Java 21.0.2 and executed on an Apple M1 Pro CPU. The largest benchmark assessments were completed within a maximum time of under 5 minutes.

In this paper, all benchmarks are generated randomly. Pin locations for each net are uniformly distributed, with $x(p), y(p) \sim \mathcal{U}(0,1)$ for all $p \in n$. Similarly, the y-coordinates for the $D(N_{\text{in}})$ tracks are randomly generated using the same uniform distribution. The number of pins per net in the multi-pin benchmarks ranges from 2 to 10 and is selected randomly.

We generated two distinct sets of random benchmarks for GCRP evaluation: two-pin benchmarks, denoted by the "gt-" prefix, and multi-pin benchmarks, denoted by the "gm-" prefix.

The results on two-pin benchmarks are given in Tables 1 and 2, and on multi-pin benchmarks are given in Tables 3, and 4. In these tables, "LE", "BCA", "SDG w/o step6", "SDG", and "PP" refer to Left-Edge [7], BCA [6], SDG without step 6, SDG, and post-processing, respectively. For multi-pin nets, BCA is enhanced by categorizing nets into Below, Cross, and Above when the SD is positive, zero, and negative, respectively. "Distance" represents the total x-distance $\sum_n d^x(n)$ and the total y-distance of nets $\sum_n d^y(n)$. "y-length" gives the total y-length of nets, and the total y-distance is used as a reference for it. "Time" represents the computation time, with the computation time by SDG with post-processing used as the reference. Note that the implementation of post-processing used in [6] is slightly different from Algorithm 3.

By experiments, it is confirmed that the total y-length of the solution obtained by using SDG is better than others in most cases. Better solutions are obtained in short time by SDG, and which are slightly improved by post-processing. The validity of the priority SDP which is used in SDG as well as the impact of postponement in step 6 in SDG are confirmed. The ratio of post-processing in computation time is dominant when the size of input is large. The density of multi-pin benchmarks is large compared to two-pin benchmarks, and is around 90% of the number of nets. The assignments of multi-pin benchmarks are restricted since it is hard to share a track by plural nets, and there may not exist assignments with vertical length closed to the total y-distance. The results of gm-5 are depicted in Fig. 6.

Table 1 Results on two-pin nets.

Benchmark (#net)	Density Vertical	Distance x	Distance y (%)	y-length (w^y) (%) LE	BCA	SDG w/o step 6	SDG
gt-1 (5)	3	1.5	1.4 (100)	2.9 (207)	1.8 (129)	1.8 (129)	1.8 (129)
gt-2 (10)	5	3.1	2.2 (100)	6.0 (273)	5.1 (232)	5.1 (232)	4.7 (214)
gt-3 (100)	53	32.4	29.6 (100)	79.5 (269)	39.9 (135)	39.9 (135)	35.0 (118)
gt-4 (500)	258	169.4	167.0 (100)	350.9 (210)	196.3 (118)	196.3 (118)	176.6 (108)
gt-5 (1000)	517	343.0	331.6 (100)	674.7 (203)	381.2 (115)	381.2 (115)	342.8 (103)
gt-6 (5000)	2474	1682.0	1676.1 (100)	3352.4 (200)	1963.7 (117)	1963.7 (117)	1727.9 (103)
gt-7 (10000)	4997	3365.6	3340.6 (100)	6844.5 (205)	3932.4 (118)	3932.4 (118)	3420.5 (102)
gt-8 (50000)	25102	16755.2	16734.0 (100)	33683.7 (201)	19751.9 (118)	19742.2 (118)	16981.0 (101)
gt-9 (100000)	49894	33302.8	33396.2 (100)	67716.8 (203)	39545.6 (118)	39444.1 (118)	33828.7 (101)

Table 2 Results on two-pin nets with post-processing.

Benchmark (#net)	LE+PP y-length (%)	LE+PP Time [s] LE	LE+PP Time [s] PP	BCA+PP y-length (%)	BCA+PP Time [s] BCA	BCA+PP Time [s] PP	SDG+PP y-length (%)	SDG+PP Time [s] SDG	SDG+PP Time [s] PP
gt-1 (5)	2.5 (180)	0.1	0.0	1.8 (129)	0.1	0.0	1.8 (129)	0.1	0.0
gt-2 (10)	4.7 (214)	0.1	0.0	3.7 (168)	0.1	0.0	3.7 (168)	0.1	0.0
gt-3 (100)	33.6 (114)	0.1	0.1	31.3 (106)	0.1	0.1	31.6 (107)	0.1	0.1
gt-4 (500)	176.1 (104)	0.1	0.6	170.4 (102)	0.2	0.2	170.3 (102)	0.2	0.3
gt-5 (1000)	347.0 (105)	0.2	1.6	335.0 (101)	0.2	0.8	335.2 (101)	0.2	1.0
gt-6 (5000)	1749.5 (104)	0.3	22.7	1688.5 (101)	1.5	11.3	1690.0 (101)	2.0	14.6
gt-7 (10000)	3483.6 (104)	0.5	87.3	3359.8 (101)	6.3	61.1	3359.0 (101)	8.7	41.2
gt-8 (50000)	17480.3 (104)	1.8	3374.3	16829.4 (101)	224.2	2174.7	16829.2 (101)	214.5	1447.7
gt-9 (100000)	34715.3 (104)	7.5	18990.9	33588.7 (101)	1283.1	12433.2	33588.3 (101)	1201.2	8256.9

Table 3 Results on multi-pin nets.

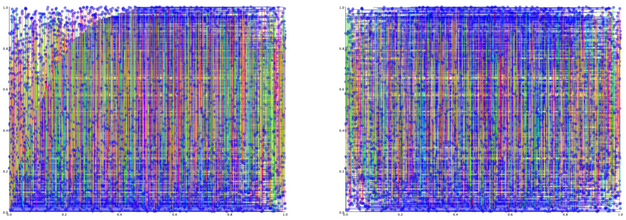
Benchmark (#net)	Density Vertical	Distance x	Distance y (%)	y-length (w^y) (%) LE	BCA	SDG w/o step 6	SDG
gm-1 (5)	5	3.1	4.4 (100)	7.2 (164)	6.8 (155)	6.6 (150)	6.6 (150)
gm-2 (10)	10	6.8	13.7 (100)	21.1 (154)	18.1 (132)	16.7 (122)	16.7 (122)
gm-3 (100)	89	65.7	128.9 (100)	203.7 (158)	179.4 (139)	151.8 (118)	149.9 (116)
gm-4 (500)	455	334.9	653.8 (100)	995.6 (152)	905.5 (138)	735.6 (113)	723.2 (111)
gm-5 (1000)	893	659.3	1260.8 (100)	1937.1 (154)	1709.3 (136)	1396.9 (111)	1377.4 (109)
gm-6 (5000)	4466	3315.8	6405.8 (100)	10077.0 (157)	8782.8 (137)	7194.4 (112)	7076.6 (110)
gm-7 (10000)	8901	6621.6	12652.7 (100)	19680.6 (156)	17292.2 (137)	14146.5 (112)	13891.3 (110)
gm-8 (50000)	44425	33094.3	63586.8 (100)	99454.0 (156)	86677.9 (136)	70838.7 (111)	69574.2 (109)
gm-9 (100000)	88832	66146.9	127098.3 (100)	198695.7 (156)	173344.5 (136)	141754.7 (111)	139113.5 (109)

Table 4 Results on multi-pin nets with post-processing.

Benchmark (#net)	LE+PP y-length (%)	LE+PP Time [s] (%) LE	LE+PP Time [s] (%) PP	SDG+PP y-length (%)	SDG+PP Time [s] (%) SDG	SDG+PP Time [s] (%) PP	SDG+PP Time [s] (%) SDG+PP
gm-1 (5)	4.8 (109)	0.1 (97)	0.0 (42)	4.8 (109)	0.1 (89)	0.0 (11)	0.1 (100)
gm-2 (10)	15.9 (116)	0.1 (89)	0.0 (54)	15.9 (116)	0.1 (86)	0.0 (14)	0.1 (100)
gm-3 (100)	142.2 (110)	0.1 (50)	0.1 (49)	143.2 (111)	0.1 (56)	0.1 (44)	0.2 (100)
gm-4 (500)	703.6 (108)	0.2 (20)	0.6 (78)	704.4 (108)	0.2 (23)	0.6 (77)	0.8 (100)
gm-5 (1000)	1343.6 (107)	0.2 (7)	1.8 (77)	1341.0 (106)	0.3 (12)	2.1 (88)	2.4 (100)
gm-6 (5000)	6944.6 (108)	0.3 (1)	41.6 (98)	6906.2 (108)	3.4 (8)	39.1 (92)	42.5 (100)
gm-7 (10000)	13616.6 (108)	0.5 (0)	298.8 (110)	13554.8 (107)	14.7 (5)	257.1 (95)	271.8 (100)
gm-8 (50000)	68991.7 (108)	3.3 (0)	12889.2 (129)	68228.6 (107)	438.3 (4)	9553.3 (96)	9991.6 (100)
gm-9 (100000)	137774.6 (108)	13.8 (0)	73687.9 (130)	133834.5 (107)	2211.5 (4)	54471.5 (96)	56683.0 (100)

5. Conclusion

This work introduced Symmetric Difference General channel routing algorithm (SDG), tailored for critical routing layers. SDG is applied to generalized channels which model critical routing layer, and derives routing solutions which consist of the connection of a net in single-trunk Steiner tree structure. SDG outperforms conventional algorithms in terms of the total length in general. In experiments, the vertical length by SDG is about 50% smaller than Left-Edge, about 13% smaller than BCA in 2-pin benchmarks, and about 30% smaller than Left-Edge, about 20% smaller than BCA in multi-pin benchmarks.



(a) Left-Edge ($w^y = 1937.1$) (b) SDG ($w^y = 1377.4$)

Fig. 6 Results on gm-5 (#net=1000, $D(N_{\text{in}})$ = 893).

GCRP formulation adopted in this paper assumes that the wire width is unique and that there are no layout constraints except HC. However this formulation is too simple to accomplish the routing in actual critical layers where various layout constraints such as local vertical congestion, wire width and shield specification are enforced. In order to accomplish the routing in actual critical routing layers, enhancements of SDG to take the various design constraints are required. SDG will be utilized as a basic tool for advanced chip designs, especially for critical routing layers.

Acknowledgments

This work was partially supported by MEXT Initiative to Establish Next-generation Novel Integrated Circuits Centers (X-NICS) Grant Number JPJ011438.

References

[1] Z. Wang, M. Shimoda, and A. Takahashi, “Single trunk routing problem for generalized channel,” IEICE Technical Report, VLD2023-104, 2024.
[2] G. Posser, E.F. Young, S. Held, Y.L. Li, and D.Z. Pan, “Challenges and approaches in VLSI routing,” Proc. International Symposium on Physical Design (ISPD), pp.185–192, 2022.
[3] M. Sako, T. Nakajima, F. Kono, T. Nakano, M. Fujiu, J. Musha, D. Nakamura, N. Kanagawa, Y. Shimizu, K. Yanagidaira, T. Utsumi, T. Kawano, Y. Hosomura, H. Yabe, M. Kano, H. Sugawara, A.H. Sravan, K. Hayashi, T. Kouchi, and Y. Watanabe, “A 1 Tb 3b/cell 3D-flash memory of more than 17 Gb/mm^2 bit density with 3.2 Gbps interface and 205 MB/s program throughput,” 2023 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits), pp.1–2, June 2023.
[4] S. Kobayashi, K. Tashiro, Y. Minemura, K. Nakagami, K. Arita, T. Oohashi, K. Funayama, H. Sakai, M. Mushiga, K. Okabe, Y. Kanno, S. Shimizu, E. Fujikura, A. Nakae, K. Yamaguchi, H. Yamawaki, K. Nakajima, and M. Sato, “High peformance 3D flash memory with 3.2 Gbps interface and 205 MB/s program throughput based on CBA (CMOS directly bonded to array) technology,” 2023 International Electron Devices Meeting (IEDM), pp.1–4, 2023.
[5] M. Tagami, “CMOS directly bonded to array (CBA) technology for future 3D flash memory,” 2023 International Electron Devices Meeting (IEDM), pp.1–4, 2023.
[6] Z. Wang, M. Shimoda, and A. Takahashi, “BCA channel routing to minimize wirelength for generalized channel problem,” 2024 IEEE International Symposium on Circuits and Systems (ISCAS), pp.1–5, 2024.
[7] A. Hashimoto and J. Stevens, “Wire routing by optimizing channel assignment within large apertures,” Proc. 8th Design Automation Workshop (DAC), pp.155–169, 1971.
[8] D.N. Deutsch, “A “dogleg” channel router,” Proc. 13th Design Automation Conference (DAC), pp.425–433, 1976.
[9] A.S. LaPaugh, “Algorithms for integrated cricuit layout: An analytic approach,” Ph.D. thesis, Massachusetts Institute of Technology, USA, 1980.
[10] T. Yoshimura and E. Kuh, “Efficient algorithms for channel routing,” IEEE Trans. Comput.-Aided Design of Integr. Circuits Syst., vol.1, no.1, pp.25–35, 1982.
[11] T. Szymanski, “Dogleg channel routing is NP-complete,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol.4, no.1, pp.31–41, 1985.
[12] T.T. Ho, S. Iyengar, and S.Q. Zheng, “A general greedy channel routing algorithm,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol.10, no.2, pp.204–211, 1991.
[13] A. Takahashi and H. Murata, “Three-layer L-shaped channel routing algorithm,” IPSJ Journal, vol.40, no.4, pp.1618–1625, 1999 (in Japanese).
[14] S.S. Sau, A. Pal, T.N. Mandal, A.K. Datta, R.K. Pal, and A. Chaudhuri, “A graph based algorithm to minimize total wire length in VLSI channel routing,” IEEE International Conference on Computer Science and Automation Engineering, pp.61–65, 2011.
[15] K. Taniguchi, S. Tayu, A. Takahashi, Y. Todoroki, and M. Minami, “Bottleneck channel routing to reduce the area of analog VLSI,” Proc. 24th Workshop on Synthesis And System Integration of Mixed Information technologies (SASIMI), pp.26–31, 2022.
[16] K. Taniguchi, S. Tayu, A. Takahashi, M. Molongo, M. Minami, and K. Nishioka, “A fast three-layer bottleneck channel track assignment with layout constraints using ILP,” Proc. 25th Workshop on Synthesis And System Integration of Mixed Information Technologies (SASIMI), pp.50–55, 2024.
[17] K. Taniguchi, S. Tayu, A. Takahashi, M. Molongo, M. Minami, and K. Nishioka, “Two-layer bottleneck channel track assignment for analog VLSI,” IPSJ Trans. System LSI Design Methodology, vol.17, pp.67–76, 2024.
[18] K. Taniguchi, S. Tayu, A. Takahashi, M. Molongo, M. Minami, and K. Nishioka, “A fast three-layer one-side bottleneck channel routing with layout constraints using ILP,” IEICE Trans. Fundamentals, vol.108, 2025 (to appear).
[19] M. Borah, R. Owens, and M. Irwin, “An edge-based heuristic for Steiner routing,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol.13, no.12, pp.1563–1568, 1994.
[20] H. Chen, C. Qiao, F. Zhou, and C.K. Cheng, “Refined single trunk tree: A rectilinear Steiner tree generator for interconnect prediction,” Proc. International Workshop on System-Level Interconnect Prediction (SLIP), pp.85–89, 2002.
[21] S.E.D. Lin and D.H. Kim, “Construction of all rectilinear Steiner minimum trees on the Hanan grid and its applications to VLSI design,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol.39, no.6, pp.1165–1176, 2020.
[22] M. Shimoda and A. Takahashi, “Gridless gap channel routing with variable-width wires,” IEICE Trans. Fundamentals, vol.108, 2025 (to appear).

Zezhong Wang received the B.E. degree in Electrical Engineering from North Carolina Agricultural and Technical State University in 2018 and the M.E. degree in Electrical and Computer engineering from Boston University, Boston, USA in 2020. He is currently a Doctoral student with the Department of Information and Communications Engineering of Tokyo Institute of Technology. His current research focus on VLSI physical design.

Masayuki Shimoda received the B.E. and M.E. degree in engineering from Tokyo Institute of Technology, Tokyo, Japan, in 2018 and 2020, respectively. He is currently a Doctoral student with the Department of Information and Communications Engineering of Tokyo Institute of Technology. His current research interests include machine learning, VLSI physical design, and computer architecture.

Atsushi Takahashi received the 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 was at the Tokyo Institute of Technology as a Research Associate from 1991 to 1997, and as an Associate Professor from 1997 to 2009 and from 2012 to 2015. He is currently a Professor with Department of Information and Communications Engineering, School of Engineering, Tokyo Institute of Technology. His current research interests include VLSI layout design and combinational algorithms. He is a senior member of IEEE and IPSJ, and a member of ACM.