

# **IEICE** **TRANSACTIONS**

## **on Fundamentals of Electronics, Communications and Computer Sciences**

DOI:10.1587/transfun.2024VLP0002

Publicized:2024/09/09

This advance publication article will be replaced by  
the finalized version after proofreading.



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

# A Fast Three-layer One-side Bottleneck Channel Routing with Layout Constraints using ILP\*

Kazuya TANIGUCHI<sup>†a)</sup>, *Nonmember*, Satoshi TAYU<sup>†b)</sup>, *Member*, Atsushi TAKAHASHI<sup>†c)</sup>, *Fellow*, Mathieu MOLONGO<sup>††</sup>, Makoto MINAMI<sup>††</sup>, and Katsuya NISHIOKA<sup>††</sup>, *Nonmembers*

**SUMMARY** An algorithm for three-layer bottleneck channel routing problem that uses ILP is proposed. The proposed algorithm determines the track and layer assignment of nets for problems with layout constraints in which pins of each net are placed on the upper boundary of the adjacent regions on both sides of the bottleneck channel. The proposed algorithm restricts the routing pattern of each net to one of three patterns by taking feasibility into account, and outputs a solution in a few seconds when the number of nets is 300.

**key words:** *VLSI Layout Design, Three-layer Bottleneck Channel Routing, Integer Linear Programming*

## 1. Introduction

In VLSI layout design, it is required not to deteriorate the circuit performance. On the other hand, it is important to realize a layout in a small area to reduce manufacturing costs. The objective of our research is to develop a routing framework that enables us to layout a circuit in small area while meeting performance specifications.

In VLSI with fewer routing layers, cell-based design where the routing area is defined between cells is often adopted, and may contain bottleneck routing regions which are bottleneck for area reduction. While a routing style for standard cell design in advanced node was discussed in [3], we focus on routing problem defined between cells. In this paper, a routing problem on the bottleneck routing region where performance specifications are not violated even if multiple wires go through a track is discussed. A medium-scale routing problem in which the number of nets is several hundred and layout constraints are imposed to consider circuit performance is assumed.

For routing problems which contain layout constraints, it is not easy to construct constructive algorithms that consider all constraints. If the routing problem can be formulated as an Integer Linear Programming (ILP), layout constraints can be considered more easily than constructive algorithms. However, ILP is generally NP-hard and often difficult to solve

in practical time, even for small-scale problems.

In cell-based design [4], the routing area is partitioned into small routing regions called channel or switchbox, and various design flows and algorithms have been proposed. For example, channel routing algorithms assuming HV routing in which the routing in each layer consists of either horizontal segments (H) or vertical segments (V), and pins of nets located on the boundary of routing region were proposed in [5]–[8]. Three layer L-shaped channel routing was discussed in [9]. Track assignment procedure considering cross-talk minimization was proposed in [10], and track assignment in cases that pins of nets are located inside routing region was discussed in [11], [12]. Sufficient conditions to complete a switchbox routing was discussed in [13]. Routing architectures which consist of extremal switch-block structures were discussed in [14]. A design flow without repeating design is discussed in [15]–[17].

For general routing problems, various types of routing algorithms such as maze [18] and A\* [19] have been introduced for a single net. For routing of multiple nets, rip-up and reroute technique [20] and length matching [21], [22] to improve the completion ratio of routing and to improve the quality of routing have been discussed. However, they may not fit to cell-based design in which rectangle routing regions without obstacles are defined. Challenges and approaches of VLSI routing was discussed in [23].

A routing design flow with HV routing without repeating routing design has been established. However, the obtained layout may contain a routing region which is a bottleneck for area reduction (Fig. 1(a)). In order to resolve bottleneck, *Bottleneck channel routing* (Fig. 1(b)) was proposed in [24], [25]. In [25], the algorithm U2TLA-2.0 for two-layer bottleneck channel problem was proposed. However it is a greedy algorithm, and is not easy to enhance to three or more layers with layout constraints.

In this paper, we focus on three-layer bottleneck channel routing with layout constraints in which pins of nets are located on the boundary of routing region. The number of tracks required in three-layer bottleneck region is at least one-third of the number of nets that go through the region horizontally in bottleneck channel routing. While, it is at least two-thirds of the number of nets if HVH routing where layers 1 and 3 consist of horizontal segments and layer 2 consists of vertical segments is adopted. In case that horizontal tracks are used as much as possible, the height of the bottleneck region required by three-layer bottleneck channel

<sup>†</sup>The authors are with Tokyo Institute of Technology, 2-12-1 Ookayama, Meguro-ku, Tokyo, 152-8550 Japan.

<sup>††</sup>The authors are with Jedat, 1-1-12 Minato, Chuo-ku, Tokyo, 104-0043 Japan.

\*The preliminary versions were presented at [1], [2].

a) E-mail: taniguchi@eda.ict.e.titech.ac.jp

b) E-mail: tayu@ict.e.titech.ac.jp

c) E-mail: atsushi@ict.e.titech.ac.jp

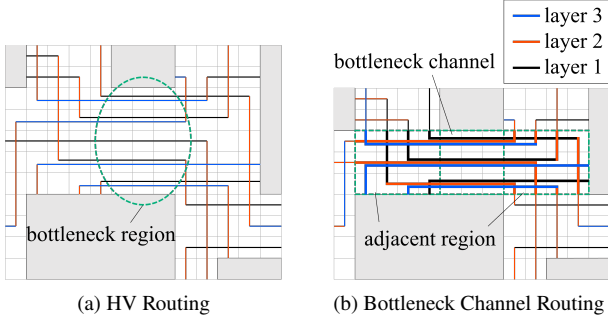


Fig. 1 Circuit layouts without and with bottleneck channel

routing is 50% smaller compared to that by HVH routing.

In this paper, we propose U3TLA-ILP3.0 that determines the routing of nets by using ILP for U-shaped three-layer bottleneck channel routing problem (U3BCRP) with layout constraints. A layout pattern obtained by using U3TLA-ILP3.0 will be utilized as an initial solution to explore a detailed routing in bottleneck region efficiently.

U3TLA-ILP3.0 restricts the routing pattern of each net to one of three patterns by taking feasibility into account. In U3TLA-ILP3.0, decision variables are used to determine the pattern used by each net, the track and layer assignment is determined according to the decision variables, and the routing pattern is determined. U3TLA-ILP3.0 outputs solutions in a few seconds for medium-scale problems about 300 nets. The routing pattern that satisfies layout constraints can be obtained in a short time by U3TLA-ILP3.0.

## 2. Bottleneck Channel Routing Problem

A routing problem is to find a better routing pattern that satisfies the connection requirement under the design rule. The connection requirement among pins is called a *net*. Among routing patterns which satisfy connection requirement, a routing pattern is *infeasible* if there is a design rule violation, *feasible* otherwise.

In grid based design, the wires of different nets have a *conflict* if they share the same coordinate in the same layer. In grid based design, a solution which satisfies connection requirement and have no conflicts is feasible. The solution of a routing problem must be feasible and must meet layout constraints.

Three-layer bottleneck channel routing problem is defined on routing area that consists of a *bottleneck channel* and *adjacent regions* on both sides as shown in Fig. 2. A wire which connects pins of a net goes through a track in the bottleneck channel. A set of two-pin nets  $N = \{n_1, n_2, \dots, n_k\}$  is given as an input. Pins of each net are placed on the boundary of left and right adjacent regions. The left sequence  $L = (l_1, l_2, \dots, l_k)$  where  $l_i \in N$  is defined as the sequence of nets whose pins are aligned counterclockwise order on the boundary of the left-adjacent region. Similarly, the right sequence  $R = (r_1, r_2, \dots, r_k)$  where  $r_i \in N$  is defined as the sequence of nets whose pins are aligned clockwise order on the boundary of the right-adjacent re-

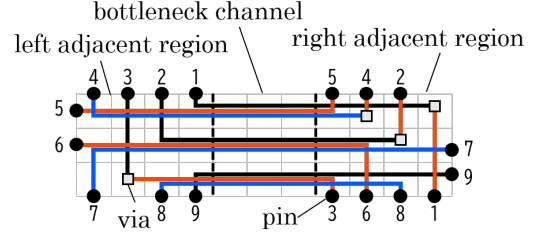


Fig. 2 Bottleneck channel routing

### U3BCRP

Input:

- net set  $N = \{n_1, n_2, \dots, n_k\}$  ( $k > 0$ )
- right sequence  $R = (r_1, r_2, \dots, r_k)$
- the number of tracks  $T \in \mathbb{N}^+$

Output:

- track assignment of nets  
 $A_T : N \rightarrow \{1, 2, \dots, T\}$
- layer assignment of segments of nets  
 $A_L : N \rightarrow \{1, 2, 3\}$  (left vertical)  
 $A_M : N \rightarrow \{1, 2, 3\}$  (horizontal)  
 $A_R : N \rightarrow \{1, 2, 3\}$  (right vertical)

Fig. 3 U-shaped three-layer bottleneck channel routing problem

gion.  $N^l(n)$  and  $N^r(n)$  are the sets of nets whose left and right pins are before the left pin and the right pin of net  $n$  in the left and right sequences, respectively. That is, they are defined as

$$N^l(n) = \{l_i \mid i < j, n = l_j\}, \quad (1)$$

$$N^r(n) = \{r_i \mid i < j, n = r_j\}. \quad (2)$$

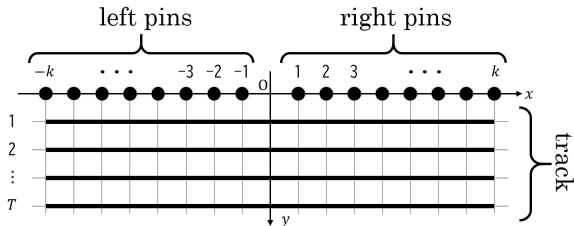
In this paper, U-shaped three-layer bottleneck channel routing problem (U3BCRP) shown in Fig. 3 which is the most basic three-layer bottleneck channel routing problem is formulated. In U3BCRP, routing area  $G_{k,T}$  for  $k$  two-pin nets and  $T$  tracks is modeled by the routing grid  $(-k \leq x \leq k, 0 \leq y \leq T)$  as shown in Fig. 4 where the  $y$ -axis corresponds to the degenerated bottleneck channel, the region  $x < 0$  corresponds to the left-adjacent region, and the region  $x > 0$  corresponds to the right-adjacent region. Left pin of net  $l_i$  and right pin of net  $r_j$  are placed at grid point  $(-i, 0)$  and  $(j, 0)$ , respectively. Track  $t$  ( $1 \leq t \leq T$ ) is the grid line connecting  $(-k, t)$  and  $(k, t)$ .

Wires of at most three nets can go through a single track. In the following, we assume that an input which satisfies

$$k \leq 3T \quad (3)$$

is given. Also, we assume, without loss of generality, that left sequence  $L$  is fixed as  $l_i = n_i$  for any net  $n_i \in N$ . Therefore, right sequence  $R$  is given as input. In addition, we assume that layout constraints may be given, such as wires of two specified nets must not be close to each other.

In U3BCRP, an output routing pattern has to satisfy the

Fig. 4 Routing area  $G_{k,T}$ , tracks, and pins

following four conditions.

- Pin of all nets are connected by wire.
- Wires of different nets have no conflicts.
- The wire of each net consists of one horizontal and two vertical segments.
- Each segment is assigned to either layer 1, 2, or 3.

The wire of a net consists of two vertical segments and one horizontal segment which is assigned to a track.

Output of the problem is the track assignment  $A_T$  and the layer assignment of three segments  $A_L$ ,  $A_M$ , and  $A_R$ . The routing pattern is uniquely determined under the four conditions above if the track and layer assignment is determined. If the wires of different nets share the same coordinate, they must be assigned to the different layer to avoid conflicts.

Once the track assignment  $A_T(n)$  is determined, the track used by the horizontal segment of net  $n$  and the length of the vertical segments of net  $n$  are determined, and the routing shape of net  $n$  is determined.

Fig. 5 shows an example of routing pattern. The black, red, and blue line segments represent the wires on layer 1, layer 2, and layer 3, respectively.

For each net, via must be inserted to a wire when the routing layer of the wire is changed. A via placed between layer  $i$  and layer  $j$  is called  $i$ - $j$  via. A conflict occurs when a 1-3 via and a wire on layer 2 share the same coordinate. Note that a 1-2 via can share the same coordinate with a wire on layer 3 without conflict.

### 3. Track Assignment to avoid Conflict

Before introducing ILP formulations for U3BCRP, necessary conditions in track assignment to avoid conflicts when the routing layer of each segment of nets is specified are discussed.

Let  $N_{i,i}$  be the set of nets whose left vertical segment and horizontal segment both use layer  $i$ . All horizontal segments of nets belonging to  $N_{i,i}$  use the layer  $i$ , and all nets belonging to  $N_{i,i}$  are assigned to different tracks if the routing pattern has no conflicts. Let  $a$  and  $b$  be nets belonging to  $N_{i,i}$  such that  $a \in N^l(b)$ . If net  $a$  is assigned to a track below the track to which net  $b$  is assigned, the left vertical segment of net  $a$  and the horizontal segment of net  $b$  intersect on the same layer, and conflict occurs as shown in Fig. 6(a). To avoid the conflict between net  $a$  and net  $b$ , net  $a$  must be assigned to a track above the track to which net  $b$  is assigned

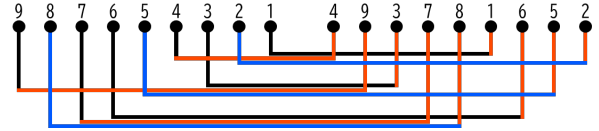
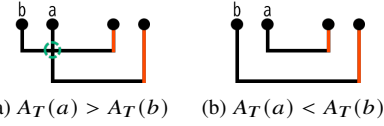
Fig. 5 A routing pattern for  $R = (n_4, n_9, n_3, n_7, n_8, n_1, n_6, n_5, n_2)$ 

Fig. 6 Track assignment with and without conflict

as shown in Fig. 6(b).

If a routing pattern contains no conflicts, all nets belonging to  $N_{i,i} \cap N^l(n)$  are assigned to a track above the track to which net  $n (\in N_{i,i})$  is assigned. In this situation, the following equation holds.

$$\forall n \in N_{i,i}, A_T(n) > |N_{i,i} \cap N^l(n)| \quad (4)$$

If a track assignment satisfies the condition of Eq. (4), the left vertical segment and the horizontal segment of nets belonging to  $N_{i,i}$  have no conflicts.

### 4. ILP Formulation which requires much calculation time

In this section, a straightforward ILP formulation for U3BCRP is introduced. This formulation is natural and intuitive but impractical.

Fig. 7 gives a part of ILP formulation U3TLA-TLLL for U3BCRP. Since the logical product ( $\wedge$ ) can be converted to an equivalent linear expression using auxiliary variables, a formulation which contains logical product is used here for clarity.

U3TLA-TLLL represents the routing of each net by using four variables defined in Eqs. (5), (6) which specify the track assignment and the layer assignment of three segments.  $x_{i,n}^T$  specifies track assignment  $A_T$ . The net  $n$  is assigned to track  $t$  if  $x_{i,n}^T = 1$ .  $x_{i,n}^L, x_{i,n}^M$ , and  $x_{i,n}^R$  specify layer assignment  $A_L, A_M$ , and  $A_R$ , respectively. For example, the left vertical segment of net  $n$  is assigned to layer  $i$  if  $x_{i,n}^L = 1$ .

In U3BCRP, each net is assigned to one track and each segment of a net uses one layer. They are forced by Constraint on Pattern in U3TLA-TLLL (Eqs. (7), (8)). The conflicts between the wires of different nets (Eqs.(9), (10)), and the conflicts between the 1-3 via and the wire on layer 2 (Eq. (11)) on left region are prohibited in the Constraint on Conflict in U3TLA-TLLL. The conflict on right region can be prohibited similarly, but the description is omitted here.

### 5. Proposed Method U3TLA-ILP3.0

In this section, the proposed algorithm for U3BCRP with ILP formulation U3TLA-ILP3.0 shown in Fig. 8 is introduced.

**U3TLA-TLLL**  
**Decision variables**

$$x_{t,n}^T \in \{0, 1\} \quad (\forall (t, n) \in \{1, 2, \dots, T\} \times N) \quad (5)$$

$$x_{i,n}^L, x_{i,n}^M, x_{i,n}^R \in \{0, 1\} \quad (\forall (i, n) \in \{1, 2, 3\} \times N) \quad (6)$$

**Constraints**

- Pattern

$$\sum_{t=1}^T x_{t,n}^T = 1 \quad (\forall n \in N) \quad (7)$$

$$\sum_{i=1}^3 x_{i,n}^L = \sum_{i=1}^3 x_{i,n}^M = \sum_{i=1}^3 x_{i,n}^R = 1 \quad (\forall n \in N) \quad (8)$$

- Conflict

$$\sum_{n \in N} x_{t,n}^T \wedge x_{i,n}^M \leq 1 \quad (\forall t \in [T], \forall i \in \{1, 2, 3\}) \quad (9)$$

$$\sum_{t=1}^T t \cdot x_{t,n}^T < \sum_{t=1}^T t \cdot x_{t,n'}^T + T(1 - \sum_{i=1}^3 x_{i,n}^L \wedge x_{i,n'}^M) \quad (\forall (n, n') \in N^2, n \in N^l(n')) \quad (10)$$

$$x_{t,n}^T \wedge x_{t,n'}^T \wedge x_{2,n}^M \wedge x_{1,n'}^L \wedge x_{3,n'}^M = 0$$

$$x_{t,n}^T \wedge x_{t,n'}^T \wedge x_{2,n}^M \wedge x_{3,n'}^L \wedge x_{1,n'}^M = 0 \quad (\forall (t, n, n') \in [T] \times N \times N, n' \in N^l(n)) \quad (11)$$

Fig. 7 ILP formulation U3TLA-TLLL

### 5.1 Routing pattern

U3TLA-ILP3.0 restricts the layer assignment of each net to three patterns shown in Fig. 9. U3TLA-ILP3.0 uses a variable that specifies the pattern used by each net, but does not use variables for track assignment.

The variables used in U3TLA-ILP3.0 are

$$p_n^i := \begin{cases} 1 & (\text{net } n \text{ uses } P_i) \\ 0 & (\text{otherwise}) \end{cases} \quad (12)$$

where  $i \in \{1, 2, 3\}$  and  $n \in N$ . The net  $n$  uses  $P_i$  if  $p_n^i = 1$ . The Constraint on Pattern

$$p_n^1 + p_n^2 + p_n^3 = 1 \quad (\forall n \in N) \quad (13)$$

restricts the pattern of each net to one of three patterns.

### 5.2 Track Assignment

Proposed algorithm determines track assignment according to the patterns of nets determined by U3TLA-ILP3.0.

When all variables are determined to satisfy Eq. (13), the track assignment of net  $n$  using  $P_1$  is defined as follows (see also Fig. 10).

$$A_T(n) := \sum_{m \in N^l(n)} p_m^1 + 1 \quad (14)$$

Note that  $\sum_{m \in N^l(n)} p_m^1$  is equal to the number of nets which appear before  $n$  in the left-sequence  $L$  and that use  $P_1$ . Both the left vertical segments and the horizontal segments of nets using  $P_1$  use layer 1, and there is no conflict among

### Algorithm for U3BCRP

Input:  $N$

Output:  $A_T(N), A_L(N), A_M(N), A_R(N)$

\*\*\*\*\* 1. Pattern selection using ILP \*\*\*\*\*

$P(p)$ : **ILP-formulation**

**Variables:** Eq. (12)

**Constraints**

- Pattern: Eq. (13)
- Tracks: Eq. (15)
- Conflicts: Eq. (16, 17, 18)
- Layout: Eq. (21)

solve( $P(p)$ )

**for**( $n \in N$ )

$$p(n) := \begin{cases} 1 & \text{if } p_n^1 = 1 \\ 2 & \text{if } p_n^2 = 1 \\ 3 & \text{if } p_n^3 = 1 \end{cases}$$

**endfor**

\*\*\*\*\* 2. Track and layer assignment \*\*\*\*\*

**for**( $n \in N$ )

**switch**( $p(n)$ )

**case**(1):

$$A_T(n) := |\{m \mid m \in N^l(n), p(m) = 1\}| + 1$$

$$(A_L(n), A_M(n), A_R(n)) := (1, 1, 2)$$

**case**(2):

$$A_T(n) := |\{m \mid m \in N^r(n), p(m) = 2\}| + 1$$

$$(A_L(n), A_M(n), A_R(n)) := (1, 2, 2)$$

**case**(3):

$$A_T(n) := |\{m \mid m \in N^l(n), p(m) = 3\}| + 1$$

$$(A_L(n), A_M(n), A_R(n)) := (3, 3, 2)$$

**endswitch**

**endfor**

Fig. 8 Proposed Algorithm for U3BCRP

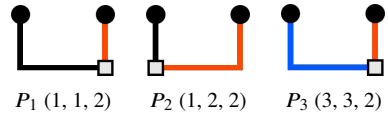
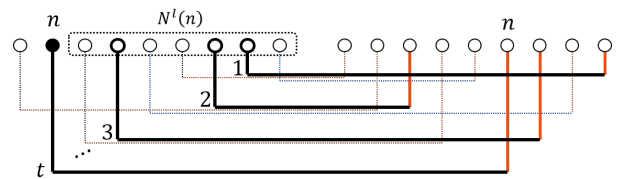


Fig. 9 Three patterns used by U3TLA-ILP3.0


 Fig. 10 Track assignment of  $P_1$ 

them since the track assignments defined by Eq. (14) satisfy the condition of Eq. (4). The right vertical segments and the horizontal segments of nets using  $P_1$  have no conflicts among them since the former use layer 2 and the latter use layer 1. Therefore, any nets using  $P_1$  have no conflicts among them.

For any nets using  $P_2$  (or  $P_3$ ), the track assignments of nets are defined similarly so that they have no conflicts among them. Therefore, any nets using the same pattern have no conflicts.

The number of nets to be used for each pattern is less

than or equal to the number of tracks  $T$ . The Constraint on Tracks

$$\sum_{n \in N} p_n^i \leq T \quad (\forall i \in \{1, 2, 3\}) \quad (15)$$

restricts the number of nets to be used for each pattern.

### 5.3 Constraints in ILP to avoid conflicts

From the restriction of routing pattern and the track assignment, any nets using the same pattern have no conflicts. A conflict occurs only if a vertical segment and a horizontal segment belong to different patterns. U3TLA-ILP3.0 imposes constraints which prevent conflicts caused by such segments belonging to different patterns.

For any pattern  $P_i$ , a conflict occurs when a vertical segment of  $P_i$  intersects the horizontal segment of other pattern  $P_j (\neq P_i)$  which use the same layer. Among 12 pairs of vertical and horizontal segments belonging to different patterns, the following three are pairs of segments which use the same layer.

- right vertical segment of  $P_1$ , horizontal segment of  $P_2$
- left vertical segment of  $P_2$ , horizontal segment of  $P_1$
- right vertical segment of  $P_3$ , horizontal segment of  $P_2$

The intersection of segments in these three pairs are prohibited by the Constraint on Conflict shown below. The variable  $M$  used in these constraints is a big-constant defined as  $M := T + 1$ .

$$\sum_{m \in N^l(n)} p_m^1 < \sum_{m \in N^r(n)} p_m^2 + M(1 - p_n^1) \quad (\forall n \in N) \quad (16)$$

$$\sum_{m \in N^r(n)} p_m^2 < \sum_{m \in N^l(n)} p_m^1 + M(1 - p_n^2) \quad (\forall n \in N) \quad (17)$$

$$\sum_{m \in N^l(n)} p_m^3 < \sum_{m \in N^r(n)} p_m^2 + M(1 - p_n^3) \quad (\forall n \in N) \quad (18)$$

In the following, we show that a conflict caused by segments belonging to different patterns are prohibited either by Eqs. (16), (17), or (18). The value of each term in each formula is nonnegative, and the value of the left side is at most  $T$  in each formula. Therefore, note that an inequality is satisfied when  $p_n^i (i = 1, 2, 3)$  is 0 since the value of the right side is at least  $M (= T + 1)$ . Also, Eqs. (16), (17), and (18) may not be satisfied when  $p_n^1, p_n^2,$  and  $p_n^3$  are 1, respectively.

Here, we show that the left vertical segment of nets using  $P_2$  and the horizontal segments of nets using  $P_1$  do not intersect if Eq. (17) is satisfied. Let's consider a case that net  $n$  uses pattern  $P_2$ , that is,  $p_n^2 = 1$ . We show that the left vertical segment of net  $n$  and the horizontal segments of nets using  $P_1$  do not intersect if Eq. (17) is satisfied.

Let  $t$  be the value on the left side of Eq. (17).

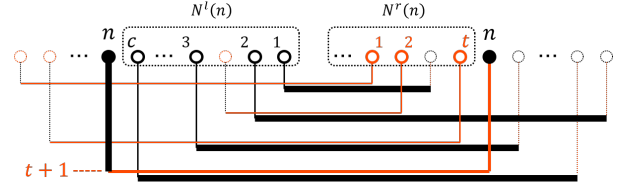


Fig. 11 No conflict at left vertical segment of  $P_2 (t < c)$

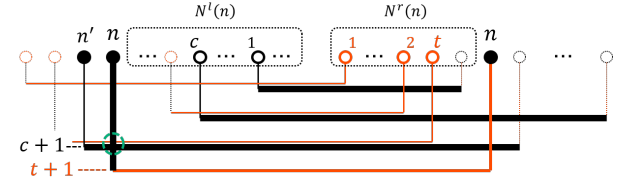


Fig. 12 Conflict at left vertical segment of  $P_2 (t \geq c)$

$$\sum_{m \in N^r(n)} p_m^2 = t \quad (19)$$

From Eq. (14),  $t + 1$  is equal to the track number of net  $n$ . The left vertical segment of net  $n$  on layer 1 spans from track 1 to track  $t + 1$  as shown in Figs. 11 and 12. Let  $c$  be the value of the first term on the right side of Eq. (17).

$$\sum_{m \in N^l(n)} p_m^1 = c \quad (20)$$

Note that  $c$  represents the number of nets using  $P_1$  which appear before net  $n$  in the left-sequence  $L$ . The horizontal segments of these nets terminate to the right of the left vertical segment of net  $n$  as shown in Figs. 11 and 12.

Note that  $t < c$  holds if Eq. (17) is satisfied. If  $t < c$ , the horizontal segments of all nets using  $P_1$  which are assigned to track 1 to track  $t + 1$  terminate to the right of the left vertical segment of net  $n$  as shown in Fig. 11. The left vertical segment of net  $n$  and the horizontal segments of nets using  $P_1$  do not intersect, and no conflict occurs.

If  $t \geq c$ , conflict may occur as shown in Fig. 12. Suppose there exists a net  $n'$  which appear first in left-sequence  $L$  among nets using  $P_1$  and whose left pin is to the left of that of net  $n$ . The net  $n'$  is assigned to track  $c + 1 (\leq t + 1)$ , and the horizontal segment of net  $n'$  on layer 1 terminates to the left of the left vertical segment of net  $n$ . The left vertical segment of net  $n$  and the horizontal segments of net  $n'$  intersect, and conflict occurs.

In summary, Eq. (17) restricts to  $t < c$ , and prohibit the intersection of the left vertical segment of nets using  $P_2$  and the horizontal segments of nets using  $P_1$ .

It can be shown similarly that the conflict between  $P_1$  and  $P_2$  and the conflict between  $P_3$  and  $P_2$  do not occur if Eqs. (16) and (18) are satisfied, respectively, but the description is omitted here.

### 5.4 Constraints in ILP to satisfy layout constraints

Among various types of layout constraints, ILP formulation

that prevents crosstalk noise is discussed here as an example.

Assume that a pair of a noise source *aggressor* and its *victim* are given, and that the wire of the aggressor and a victim neither share the same coordinate nor adjacent is requested even if they are assigned in different layer is given as constraint.

Let  $n_a$  be the aggressor, and let  $n_v$  be the victim. If the input satisfies both  $n_a \in N^l(n_v)$  and  $n_v \in N^r(n_a)$ , the wires of  $n_a$  and  $n_v$  must intersect and no feasible solution exists. So we focus on the other situations. Suppose, without loss of generality, that an input which satisfies  $n_a \in N^l(n_v) \cap N^r(n_v)$  is given. In order to satisfy the layout constraints, net  $n_a$  must be assigned to two or more tracks above the track to which  $n_v$  is assigned, regardless of patterns of them used.

The Constraints on Layout

$$\sum_{m \in N_i(n_a)} p_m^i \leq \sum_{m \in N_j(n_v)} p_m^j + (M+1)(2 - p_{n_a}^i - p_{n_v}^j) - 2$$

$$\left( N_i(n) := \begin{cases} N^l(n) & (i = 1, 3) \\ N^r(n) & (i = 2) \end{cases} \right) \quad (21)$$

where  $(i, j) \in \{1, 2, 3\}^2$  prohibit the intersection of the wires of  $n_a$  and  $n_v$  and secure the gap between the wires of them two tracks or more. The first term on the left and right sides represent the (track number)–1 of  $n_a$  and  $n_v$ , respectively. Note that Eq. (21) for  $(i, j)$  is satisfied if either  $p_{n_a}^i = 0$  or  $p_{n_v}^j = 0$  because of the second term on the right side.

In case that  $p_{n_a}^i = 1$  and  $p_{n_v}^j = 1$ , the second term on the right side of Eq. (21) is zero, and Eq. (21) forces that

$$(\text{track number of } n_a) \leq (\text{track number of } n_v) - 2.$$

By the Eq. (21), net  $n_a$  is assigned to two or more tracks above the track to which  $n_v$  is assigned.

The Constraint on Layout given here is for a single pair of aggressor and victim, but the number of pairs is often more than one in practice. If more than one pairs are given, the layout constraints can be realized by adding constraint given here for each pair. Even if any other layout constraint is given, a routing pattern which satisfies the constraint can be formulated if the constraint can be expressed in a linear equation.

## 5.5 Proposed Algorithm

Propose algorithm shown in Fig. 8 first formulates U3BCRP as U3TLA-ILP3.0. Proposed algorithm then solves the ILP using a solver to determine the pattern of each net.

Once the pattern of each net is determined, the track and layer assignment is determined uniquely. The track number of net  $n$  is set equal to the number of nets which appear before  $n$  in left or right sequence and which use the same pattern with  $n$ .

## 5.6 An example of ideas to increase the feasible ratio

There is a problem instance for which U3TLA-ILP3.0 does

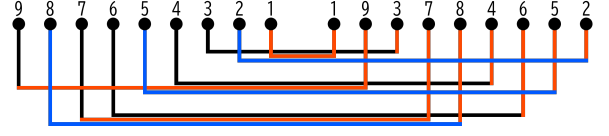


Fig. 13 A routing pattern obtained by the extended ILP3.1

not output a feasible solution even though a feasible solution exists. In order to increase the ratio of feasible solutions by U3TLA-ILP3.0, an extension of U3TLA-ILP3.0 in which a feasible solution is obtained in such cases is discussed.

Assume that the first net in the left pin sequence and the first net in the right pin sequence are both  $n_1$ . In this case, U3TLA-ILP3.0 can not find a feasible solution since the horizontal segment of  $n_1$  is assigned to track 1 regardless of the pattern used for  $n_1$ , and one of vertical segment interferes another one layer of track 1. Therefore, only two nets can be assigned to track 1 in a feasible solution.

In such cases, a feasible solution might be obtained if  $n_1$  uses a pattern without via. An ILP formulation which obtains such routing pattern is defined as U3TLA-ILP3.1. It is defined from U3TLA-ILP3.0 by imposing Conflict Constraint Eq. (16, 17, 18) to all nets in  $N$  except  $n_1$ . For example, Eq. (16) is replaced with

$$\sum_{m \in N^l(n)} p_m^1 < \sum_{m \in N^r(n)} p_m^2 + M(1 - p_n^1) \quad (\forall n \in N \setminus \{n_1\}) \quad (22)$$

From Eq. (14),  $n_1$  is assigned to track 1. All three segments of  $n_1$  are assigned to the same layer, and  $n_1$  have no conflicts with other nets. A routing pattern obtained by the extended ILP3.1 is shown in Fig. 13.

## 6. Experimental Results

U3TLA-ILP3.0, ILP3.1, and TLLL are implemented in Python 3.10.11, and executed on a PC with 3.60GHz Intel Core-i5 CPU, 32GB RAM. The solver used is IBM CPLEX 22.1.1.0 [26] where ‘‘Mixed Integer Linear Programming’’ without parallel optimization is selected as problem type.

The problem instances are randomly generated by selecting the right pin sequence among a permutation of nets. The number of nets  $k$  is set to be three times the number of tracks  $T$ . The symbol ‘\*’ is added when a layout constraint to prevent crosstalk in which a pair of one aggressor and 5 victims is defined is imposed to the problem instances. A routing pattern obtained by using U3TLA-ILP3.0 is shown in Fig. 14.

Table 1 shows the computation times, the number of variables, and the number of constraints of U3TLA-TLLL and U3TLA-ILP3.0 for one randomly generated instance where a feasible solution is obtained by U3TLA-ILP3.0 in each net size. In table, ‘‘time’’ represents the computation time taken to solve the formulated ILP for one problem instance. The number of variables used in ILP for

**Table 1** Computation time, the number of variables, and the number of constraints

The symbol ‘\*’ is added when a layout constraint to prevent crosstalk in which a pair of one aggressor and 5 victims is defined is imposed.

#net	#variable		#constraint		time [ms]	
	TLLL	3.0	TLLL	3.0	TLLL	3.0
12	1752	36	4440	51	596	8
18	5184	54	13896	75	141452	12
24	11472	72	31632	99	628764	13
30	21480	90	60240	123	>1h	13
60	157560	180	457080	243	>1h	21
150	2333400	450	6907200	603	>1h	89
*18	5184	54	13901	120	80327	11
*24	11472	72	31637	144	>1h	67

**Table 2** Computation time and Feasible ratio  
#LC represents the number of layout constraints.

#net	#LC	time [ms]	ratio [%]	
		ILP3.0	ILP3.0	ILP3.1
30	-	12	97	99
60	-	20	98	100
150	-	78	100	100
300	-	279	100	100
600	-	1153	100	100
*60	45	43	94	96
*300	45	1087	98	98

U3TLA-TLLL and U3TLA-ILP3.0 are  $2/3k^3 + 11/3k^2 + 6k$  and  $3k$ , respectively. The number of constraints in case that layout constraints are not imposed in ILP of U3TLA-TLLL and U3TLA-ILP3.0 are  $2k^3 + 7k^2 - 2k$  and  $4k + 3$ , respectively. The number of constraints imposed for the layout constraints of U3TLA-TLLL and U3TLA-ILP3.0 are 1 and 9 per a pair of one aggressor and one victim.

Table 2 shows the average computation times and the ratio of feasible solutions among 100 problem instances for each net size for U3TLA-ILP3.0 and for the extended U3TLA-ILP3.1. The ratio is reasonable since U3TLA-ILP3.0 cannot obtain a feasible solution if the first net in the right pin sequence is  $n_1$ . Higher ratio is realized by U3TLA-ILP3.1. A slightly smaller ratio is observed if a crosstalk constraint is imposed.

## 7. Conclusion

In this paper, U3TLA-ILP3.0 which can consider layout constraints for three-layer bottleneck channel routing was proposed. For U3BCRP, U3TLA-ILP3.0 obtains a routing pattern which satisfies the layout constraints in a short time by using ILP and by restricting the layout patterns. A layout pattern obtained will be utilized as an initial solution to explore a detailed routing in bottleneck region with layout constraints efficiently.

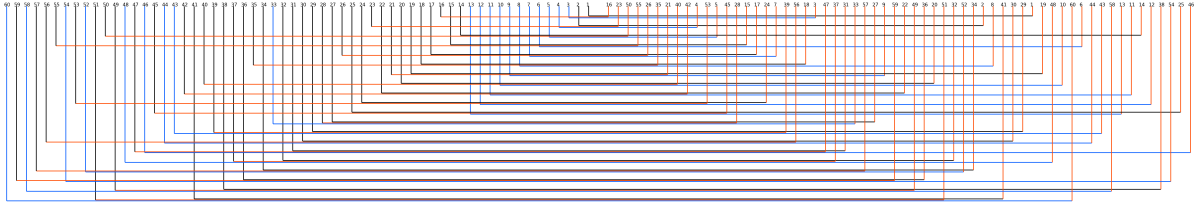
Our future works include the extension of U3TLA-ILP3.0 to adopt more routing patterns and adapt it to the more general situations. For example, a pattern in which all three segments use the same layer can improve feasibility and reduce the number of vias. The formulation of U3TLA-ILP3.0 can be easily extended when more routing layers are avail-

able. It is challenging to be able to handle problems where pins are placed not only on upper boundaries.

## References

- [1] K.Taniguchi, S.Tayu, A.Takahashi, M.Molongo, M.Minami, and K.Nishioka, “Three-layer bottleneck channel track assignment by ILP,” DA Symposium 2023, vol.2023, pp.199–206, 2023. (in Japanese).
- [2] 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. SASIMI 2024, Taipei, Taiwan, 2024.
- [3] B. Chehab, O. Zografos, E. Litta, Z. Ahmed, P. Schuddinck, D. Jang, G. Hellings, A. Spessot, P. Weckx, and J. Ryckaert, “Two-level MOL and VHV routing style to enable extreme height scaling beyond 2nm technology node,” Proc. 2021 IEEE International Interconnect Technology Conference, pp.1–3, IEEE, 2021.
- [4] A.Kahng, J.Lienig, I.Markov, and J.Hu, VLSI physical design: from graph partitioning to timing closure, Springer, 2011.
- [5] A.Hashimoto and J.Stevens, “Wire routing by optimizing channel assignment within large apertures,” Proc. 8th Design Automation Workshop, pp.155–169, 1971.
- [6] D.N. Deutsch, “A dogleg channel router,” Proc. 13th Design Automation Conference, pp.425–433, 1976.
- [7] T.Yoshimura and E.S.Kuh, “Efficient algorithms for channel routing,” IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol.CAD-1, no.1, pp.25–35, 1982.
- [8] T.T. Ho, S. Iyengar, and S.Q. Zheng, “A general greedy channel routing algorithm,” IEEE Trans. Computer-Aided Design of Integrated Circuits and systems, vol.10, no.2, pp.204–211, 1991.
- [9] A.Takahashi and H.Murata, “Three-layer L-shaped channel routing algorithm,” IPSJ Journal, vol.40, no.4, pp.1618–1625, 1999. (in Japanese).
- [10] Y.Takashima, A.Takahashi, and Y.Kajitani, “Assignment of intervals to parallel tracks with minimum total cross-talk,” IEICE Trans. Fundamentals, vol.E81–A, no.9, pp.1909–1915, 1998.
- [11] Z. Wang, M. Shimoda, and A. Takahashi, “SDG channel routing to minimize wirelength for generalized channel,” IEICE Trans. Fundamentals, vol.108, 2025. (submitted).
- [12] M. Shimoda and A. Takahashi, “Gridless gap channel routing with variable-width wires,” IEICE Trans. Fundamentals, vol.108, 2025. (submitted).
- [13] A.Takahashi and Y.Kajitani, “A switch-box router ‘BOX-PEELER’ and its tractable problems,” The Transactions of the IEICE, vol.E 72, no.12, pp.1367–1373, 1989.
- [14] Y.Takashima, A.Takahashi, and Y.Kajitani, “Routability of FPGAs with extremal switch-block structures,” IEICE Trans. Fundamentals, vol.E81–A, no.5, pp.850–856, 1998.
- [15] Y.Kajitani, “Order of channels for safe routing and optimal compaction of routing area,” IEEE Trans. Computer-Aided Design, vol.CAD-2, no.4, pp.293–300, 1983.
- [16] W.M.Dai, T.Asano, and E.S.Kuh, “Routing region definition and ordering scheme for building-block layout,” IEEE Trans. Computer-Aided Design, vol.CAD-4, no.3, pp.189–197, 1985.
- [17] D.F.Wong and C.L.Liu, “A new algorithm for floorplan designs,” Proc. 23rd ACM/IEEE Design Automation Conference, pp.101–107, 1986.
- [18] C.Y. Lee, “An algorithm for path connection and its application,” IRE Trans. Electronic Computer, vol.EC-10, no.3, pp.346–365, 1961.
- [19] P.E. Hart, N.J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimal cost paths,” IEEE Trans. Systems Science and Cybernetics, vol.4, no.2, pp.100–107, 1968.
- [20] W. Dees and P. Karger, “Automated rip-up and reroute techniques,” Proc. 19th Design Automation Conference, pp.433–439, 1982.
- [21] Y.Kohira and A.Takahashi, “CAFE router: A fast connectivity aware





**Fig. 14** A routing obtained by U3TLA-ILP3.0. #net= 60, (Agg, Vic)= ({37}, {15, 24, 41, 49, 58})

multiple nets routing algorithm for routing grid with obstacles,” IEICE Trans. Fundamentals, vol.E93-A, no.12, pp.2380–2388, 2010.

- [22] S.Sato, K.Akagi, and A.Takahashi, “A fast length matching routing pattern generation method for set-pair routing problem using selective pin-pair connections,” IEICE Trans. Fundamentals, vol.E103-A, no.9, pp.1037–1044, 2020.
- [23] G. Posser, E.F. Young, S. Held, Y.L. Li, and D.Z. Pan, “Challenges and approaches in VLSI routing,” Proc. 2022 International Symposium on Physical Design, pp.185–192, 2022.
- [24] K.Taniguchi, S.Tayu, A.Takahashi, Y.Todoroki, and M.Minami, “Bottleneck channel routing to reduce the area of analog VLSI,” Proc. SASIMI 2022, Hiroasaki, Japan, pp.26–31, 2022.
- [25] 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.
- [26] IBM, “User’s Manual for CPLEX,” <https://www.ibm.com/docs/en/icos/22.1.1?topic=optimizers-users-manual-cplex>, 2022 (Accessed on May 25, 2024).



**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 as an Associate Professor from 1997 to 2009 and from 2012 to 2015, and as a Professor from 2015. He had been with the Osaka University as an Associate Professor from 2009 to 2012. He visited University of California, Los Angeles, U.S.A., as a Visiting Scholar from 2002 to 2003. He is currently a Professor with Department of Information and Communications Engineering, School of Engineering, Tokyo Institute of Technology. His research interests include in VLSI layout design and combinational algorithms. He is a senior member of IEEE and IPSJ, and a member of ACM.



**Mathieu MOLONGO** received his B.E. degree in information and communications engineering from Telecom Paris Tech, Paris, France, in 2005. He is currently chief engineer at Jedat Inc. where he develops routing tools for LSI layout design.



**Kazuya TANIGUCHI** received his B.E. degree in information and communications engineering from Tokyo Institute of Technology, Tokyo, Japan, in 2022. He is currently a master course student of department of information and communications engineering in Tokyo Institute of Technology. His research interests include in VLSI layout design and combinational algorithms.



**Makoto MINAMI** received the B.E., and M.E. degrees in precision engineering from the University of Tokyo, Tokyo, Japan, in 1995, and 1997, respectively. He is currently chief engineer at Jedat Inc. where he develops routing tools for LSI layout design.



**Satoshi TAYU** received his B.E., M.E., and D.E. degrees in electrical and electronic engineering from Tokyo Institute of Technology, Tokyo, Japan, in 1992, 1994, and 1997, respectively. From 1997 to 2003, he was a research associate in the School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa, Japan. He is currently an assistant professor with Department of Information and Communications Engineering, School of Engineering, Tokyo Institute of Technology.

His research interests are in parallel computation. He is a member of IPSJ.



**Katsuya NISHIOKA** received the B.S degrees in computer science and information engineering from Ibaraki University, Mito, Japan, in 1992. He is currently a Senior Manager, R&D, Jedat Inc., Japan. From 2010 to 2017 he was with Renesas Electronics Corp. as an engineer in EDA & Design Methodology Division.