A Layout Adjustment Problem for Disjoint Rectangles Preserving Orthogonal Order

Kunihiko Hayashi, Michiko Inoue, Toshimitsu Masuzawa, and Hideo Fujiwara

Graduate School of Information Science,
Nara Institute of Science and Technology, Nara 630-0101, Japan.
{kunihi-h, kounoe, masuzawa, fujiwara}@is.aist-nara.ac.jp

Abstract. For a given set of n rectangles place on a plane, we consider a problem of finding the minimum area layout of the rectangles that avoids intersections of the rectangles and preserves the orthogonal order. Misue et al. proposed an $O(n^2)$-time heuristic algorithm for the problem. We first show that the corresponding decision problem for this problem is NP-complete. We also present an $O(n^2)$-time heuristic algorithm for the problem that finds a layout with smaller area than Misue's.

1 Introduction

Several algorithms for automatic graph drawing have been proposed[1][2]. Most of the algorithms are designed to create layouts (i.e., drawings) of graphs from scratch. However many systems (e.g., interactive systems) need to adjust layouts after some modifications are made in graphs, and it is desirable to adjust layouts with preserving some geometric properties of the layouts. Thus, it is important to design layout adjustment algorithms appropriate to the systems.

Geometric relations among vertices are very important geometric properties that should be preserved in adjustment of the layout. By preserving the geometric relations in the layout adjustment, we can easily recognize the correspondence between vertices in the previous layout and those in the new layout. Eades et al.[3] proposed the following geometric relations.

– orthogonal order: top-and-bottom and right-and-left relations between any two vertices;
– proximity relation: a geometric proximity relation (e.g., the nearest relation between vertices);
– topology: adjacent relations between regions of the layout.

In this paper, we consider the orthogonal order as a geometric relation that should be preserved in layout adjustment.

In some systems, vertices of a graph are sometimes represented by geometric figures such as rectangles or circles. Some modifications made on the graph, such as vertex insertion or vertex extension, may cause intersections of vertices. To avoid the intersections, layout adjustment is needed. Considering the display area

S.H. Whitesides (Ed.): GD'98, LNCS 1547, pp. 183–197, 1998.

of the systems, it is important to find the intersection-free layout with minimum area.

In this paper, we consider graphs where each vertex is represented by a rectangle and investigate the layout adjustment problem for minimizing the area under the following constraints.

- The vertices (i.e., rectangles) should not intersect;
- The orthogonal order of the vertices should be preserved.

Misue et al.[4] proposed a heuristic algorithm for the problem. The main contribution of this paper is as follows.

1. We prove that a corresponding decision problem of the layout adjustment problem is NP-complete.
2. We propose a new heuristic algorithm for the layout adjustment problem. Our algorithm is superior to Misue's; it finds a layout with smaller area than Misue's while its time complexity $O(n^2)$ is the same as Misue's where n is the number of vertices.

This paper is organized as follows. In Section 2 and 3, we introduce some preliminaries and define the layout adjustment problem. We show the NP-complete result in Section 4, and present our heuristic algorithm in Section 5. In Section 6, we conclude this paper.

2 Definition

Let R be a set of n rectangles $v_1, v_2, \cdots, v_n$. Each rectangle v_i has horizontal width w_i and vertical height h_i, where w_i and h_i are integers. We sometimes denote v_i by $\langle w_i, h_i \rangle$. A layout of R is a function from R to coordinates on the plane. We denote a layout of R by $\pi_R : R \to \mathbb{Z}^2$ for integral coordinate system, and $\pi_R : R \to \mathbb{R}^2$ for real coordinate system, where $\mathbb{Z}^2$ is an integral two dimensional space, and $\mathbb{R}^2$ is a real one.

Let x_i and y_i be x-coordinate and y-coordinate of a rectangle $v_i \in R$ in π_R, respectively. That is, $\pi_R(v_i) = (x_i, y_i)$. This indicates that the coordinates of the center of v_i is (x_i, y_i) in π_R. We assume that every rectangle is placed so that the boundary with length w_i is parallel to x-axis, and do not allow rotation of rectangles.

Let $\mathit{left}_\pi(v_i)$ and $\mathit{right}_\pi(v_i)$ be the x-coordinates of the left and right boundaries of $v_i \in R$, respectively. The y-coordinates $\mathit{top}_\pi(v_i)$ and $\mathit{bottom}_\pi(v_i)$ are defined similarly. Formally, we define them as follows.

$$\mathit{left}_\pi(v_i) = x_i - w_i/2, \quad \mathit{right}_\pi(v_i) = x_i + w_i/2,$$
$$\mathit{top}_\pi(v_i) = y_i - h_i/2, \quad \mathit{bottom}_\pi(v_i) = y_i + h_i/2$$

We also define similar notations for the layout π_R as follows.

$$\mathit{left}(\pi_R) = \min_{v_i \in R} \mathit{left}_\pi(v_i), \quad \mathit{right}(\pi_R) = \max_{v_i \in R} \mathit{right}_\pi(v_i),$$
$$\mathit{top}(\pi_R) = \min_{v_i \in R} \mathit{top}_\pi(v_i), \quad \mathit{bottom}(\pi_R) = \max_{v_i \in R} \mathit{bottom}_\pi(v_i)$$

Let $W_x(\pi_R)$ and $W_y(\pi_R)$ denote the horizontal width and the vertical width of π_R, respectively. That is,

$$W_x(\pi_R) = \mathit{right}(\pi_R) - \mathit{left}(\pi_R),\ W_y(\pi_R) = \mathit{bottom}(\pi_R) - \mathit{top}(\pi_R).$$

We also use a notation $\langle W_x(\pi_R), W_y(\pi_R)\rangle$ for π_R. We define an area $S(\pi_R)$ of π_R as $S(\pi_R) = W_x(\pi_R)W_y(\pi_R)$.

3 A Layout Adjustment Problem

We consider a layout adjustment problem for minimizing the area under the constraints that intersections of rectangles should be avoided and the orthogonal order of rectangles should be preserved. First, we define the problem as a decision problem, as follows.

INSTANCE: A rectangle set R, its layout π_R, and a positive integer K, where $\pi_R(v_i) \neq \pi_R(v_j)$ for any two rectangles $v_i, v_j \in R (i \neq j)$.
QUESTION: Is there a layout π'_R with $S(\pi'_R) \leq K$ satisfying the following constraints (1) and (2)?

Let (x_i, y_i) and (x'_i, y'_i) be $\pi_R(v_i)$ and $\pi'_R(v_i)$, respectively.
(1) π'_R preserves the orthogonal order of π_R. That is, for any two rectangles $v_i, v_j \in R$,

$$x_i < x_j \Leftrightarrow x'_i < x'_j, \text{ and } x_i = x_j \Leftrightarrow x'_i = x'_j, \text{ and}$$
$$y_i < y_j \Leftrightarrow y'_i < y'_j, \text{ and } y_i = y_j \Leftrightarrow y'_i = y'_j.$$

(2) Any two rectangles do not intersect with each other in π'_R. That is, for any two rectangles $v_i, v_j \in R (i \neq j)$,

$$|x'_i - x'_j| \geq \frac{w_i + w_j}{2} \text{ or } |y'_i - y'_j| \geq \frac{h_i + h_j}{2}.$$

We denote the above problem by *LADR* and especially by *ILADR* in the case of integral coordinate system.

4 The NP-Completeness of LADR

We show that ILADR is NP-complete. It is easy to see that ILADR is in NP. Therefore, it is sufficient to show NP-hardness of ILADR. We reduce a well-known NP-complete problem 3-SAT[6] into ILADR.

Let $X = \{x_1, x_2, \cdots, x_r\}$ be a set of boolean variables. We call x_i and $\overline{x_i}$ *literals*, and disjunction of literals *clause*. 3-SAT is defined as follows:

INSTANCE: A set X of boolean variables and a boolean expression $E = F_1 \wedge F_2 \wedge \cdots \wedge F_m$, where E is a conjunction of a finite number m of clauses, and each clause $F_i = y_{i,1} \vee y_{i,2} \vee y_{i,3}$ consists of three different literals over X.
QUESTION: Is there a truth assignment for X that satisfies E?

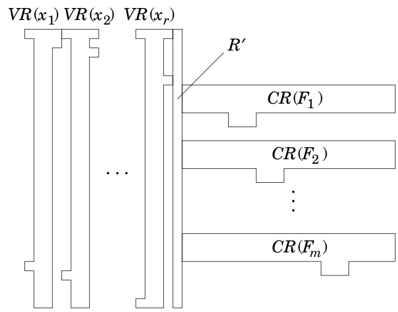


Fig. 1. Outline of the initial layout $\pi_{R(E)}$ of a rectangle set $R(E)$.

4.1 The Transformation of 3-SAT into ILADR

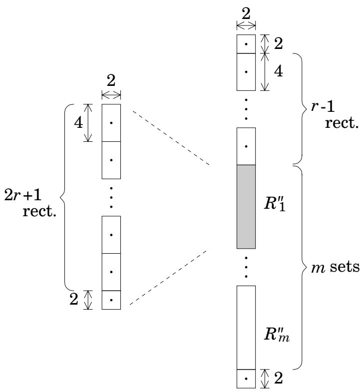
We transform 3-SAT with a boolean expression E into ILADR with a rectangle set $R^*(E)$ and its initial layout $\pi_{R^*(E)}$ from E. First, we construct a partial set $R(E)$ of $R^*(E)$ and its initial layout $\pi_{R(E)}$. Other part of $R^*(E)$ is shown only in the proof. We set the coordinate of the upper-left corner of $\pi_{R(E)}$ to $(0, 0)$. The rectangle set $R(E)$ includes a rectangle set $VR(x_k)$ for each variable x_k, a rectangle set $CR(F_i)$ for each clause F_i, and a rectangle set R'(see Fig. 1).

The rectangle set R' plays a role to restrict the positions of $VR(x_k)$ and $CR(F_i)$. R' includes R''_i for each $F_i (i = 1, \cdots, m)$. The initial layout of R''_i and R' is shown in Fig. 2.

Figure 3(a) illustrates an initial layout of $VR(x_k)$ for a variable x_k. $\mathit{VR}(x_k)$ includes rectangles $vc_{k,i}$ for $F_i(i{=}1, \cdots, m)$. $\mathit{VR}(x_k)$ is placed so that $\mathit{top}(\pi_{VR(x_k)})$ $= \mathit{top}(\pi_{R'})$ holds (see Fig. 1). It has only two layouts shown in Fig. 3, if the area of $VR(x_k)$ is restricted to $\langle 8, 2 + 4(k - 1) + 2 + 2(4r + 3)m + 4(r - k)\rangle$. We consider that Fig. 3(a) (resp. Fig. 3(b)) corresponds to assigning *true* (resp. *false*) to x_k. Note that the two layouts differ in y-coordinate of $vc_{k,i}$.

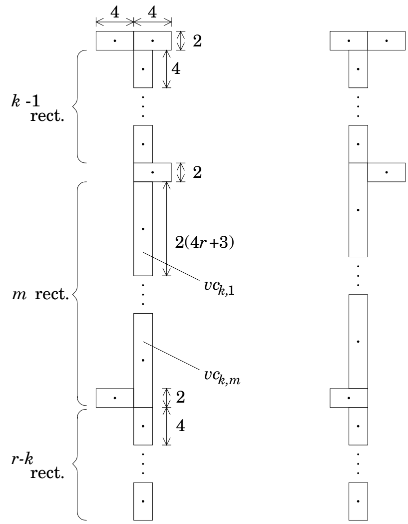
The rectangle set $CR(F_i)$ includes a rectangle set $LR(y_{i,j})$ for each literal $y_{i,j} (j = 1, 2, 3)$. There are two kinds of $LR(y_{i,j})$ and their initial layouts are shown in Figs. 4(a) and 5(a). Figure 4 shows $LR(y_{i,j})$ in the case where $y_{i,j} = x_k$ for some x_k, and Fig. 5 shows the case where $y_{i,j} = \overline{x_k}$ for some x_k. Every $LR(y_{i,j})$ includes a rectangle $\langle 2, 2\rangle$ denoted by $vs_{i,j}$. Let d_s be the difference of y-coordinates between the upper boundary of $LR(y_{i,j})$ and the center of $vs_{i,j}$. The layouts of $LR(y_{i,j})$ are restricted only to the layouts in Figs. 4 and 5 if the height is 8, $d_s = 5$ or $d_s = 3$, and the width is the minimum. We place $vs_{i,j}$ so to have the same y-coordinate as $vc_{k,i}$ in $VR(x_k)$ if $y_{i,j} = x_k$ or $\overline{x_k}$. The case of $d_s = 5$ corresponds to $x_k = \mathit{true}$ and the case of $d_s = 3$ corresponds to $x_k = \mathit{false}$. Consider the case of where the height of $LR(y_{i,j})$ is 8. If $y_{i,j} = \mathit{true}$, that is, $y_{i,j} = x_k$ and $x_k = \mathit{true}$, or $y_{i,j} = \overline{x_k}$ and $x_k = \mathit{false}$, the width can be 10. On the other hand, if $y_{i,j} = \mathit{false}$, the width must be 12 or more.

We now show the rectangle set $CR(F_i)$ for the clause F_i (see Fig. 6). $CR(F_i)$ includes $LR(y_{i,j})$ and $LR'(y_{i,j})$ $(j = 1, 2, 3)$, a rectangle $vl_i = \langle 36(m + i - 2), 4\rangle$,



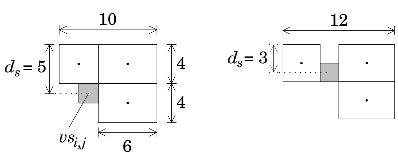
(a) R''_i (b) R'

Fig. 2. The layout of R''_i and R'.



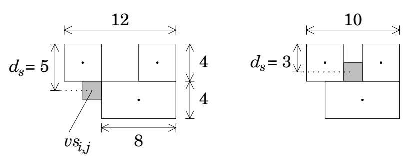
(a) $x_k = true$ (b) $x_k = false$

Fig. 3. Two layouts of $VR(x_k)$.



(a) $x_k = true$ (b) $x_k = false$

Fig. 4. Two layouts of $LR(y_{i,j})$, where $y_{i,j} = x_k$.



(a) $x_k = true$ (b) $x_k = false$

Fig. 5. Two layouts of $LR(y_{i,j})$, where $y_{i,j} = \overline{x_k}$.

and a rectangle $vr_i = \langle 36(2m - i - 1) + 12, 4\rangle$. $LR'(y_{i,j})$ consists of $ve_{i,j} = \langle 4, 4\rangle$ and $vf_{i,j} = \langle 6, 4\rangle$. We place each $LR(y_{i,j})$ so that $top(\pi_{LR(y_{i,j})}) = bottom(\pi_{LR'(y_{i,j})}) + 4(k - 1)$ holds if $y_{i,j} = x_k$ or $\overline{x_k}$. We place $LR'(y_{i,j})$ and $LR(y_{i,j})$ so that $left(\pi_{LR(y_{i,j})}) = left_\pi(ve_{i,j})$ and $right(\pi_{LR(y_{i,j})}) = right_\pi(vf_{i,j})$ hold.

The initial layout of $R(E)$ is shown in Fig. 1. We place $CR(F_i)$ and R' so that $top_\pi(vl_i) = top(\pi_{R''_i})$ holds (see Fig. 7). In the initial layout, for each rectangle in $CR(F_i)$ except for $vs_{i,j}$, there exists a rectangle in R''_i with the same y-coordinate. When $y_{i,j} = x_k$ or $\overline{x_k}$, the y-coordinates of $vs_{i,j}$ is the same as y-coordinates of $vc_{k,i}$ in $VR(x_k)$. Therefore, they have the same y-coordinate in any adjusted layout satisfying the constraint (1). $CR(F_i)$ and $CR(F_{i+1})$ are apart enough for the rectangles in them not to intersect (see Fig. 7).

Each of $VR(x_k)$ and $CR(F_i)$ includes the polynomial number and size of rectangles on r and m. $R(E)$ includes polynomial number of $VR(x_k)$ and $CR(F_i)$. Therefore the initial layout of $R(E)$ can be constructed in polynomial time.

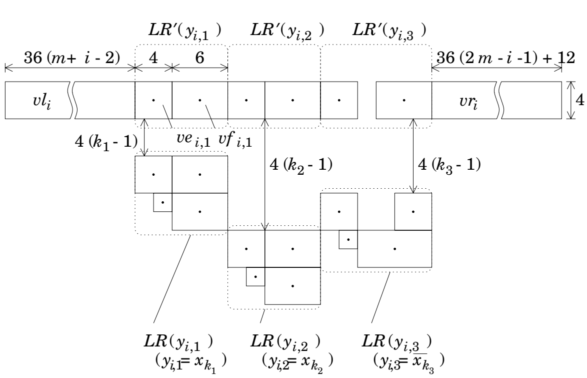


Fig. 6. The initial layout of $CR(F_i)$.

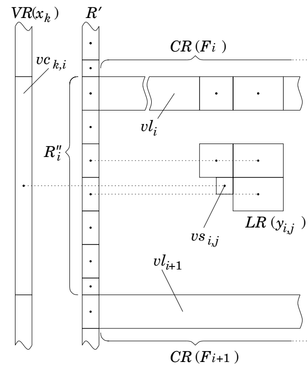


Fig. 7. The layout of $CR(F_i)$ and R'.

Example. Figure 8(a) shows the initial layout of $R(E)$ for an expression $E = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_2} \vee \overline{x_3} \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_4} \vee x_2)$. This corresponds to the truth assignment $x_1 = x_2 = x_3 = x_4 = true$, which does not satisfy E and requires $W_x(\pi_R) = 108m + 8r - 58$ and $W_y(\pi_R) = 8mr + 6m + 4r$. The expression E is satisfied by the truth assignment $x_1 = x_2 = true, x_3 = x_4 = false$. Figure 8(b) shows the corresponding layout. In this case, the width is reduced to $W_x(\pi_R) = 108m + 8r - 62$.

4.2 Proof

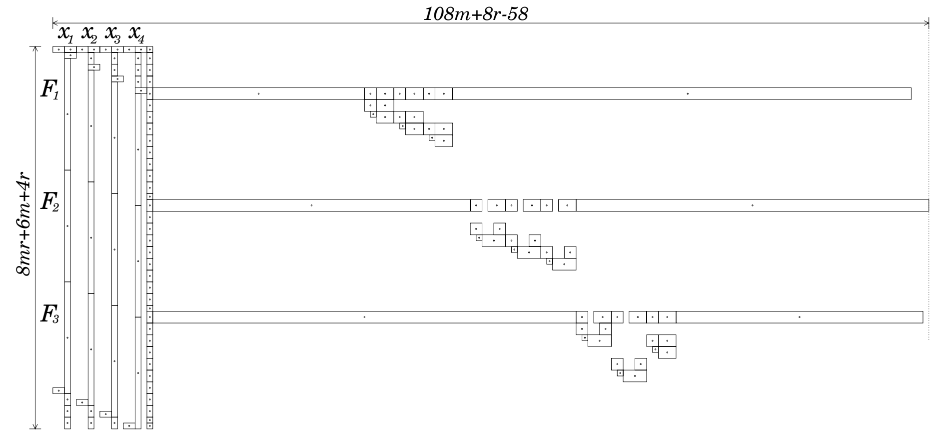
We show the reducibility of 3-SAT to ILADR.

Lemma 1. *E is satisfiable if and only if there exists a layout $\pi'_{R(E)}$ of $R(E)$, such that it satisfies the constraints (1) and (2), and $W_x(\pi'_{R(E)}) \leq 108m+8r-60$, $W_y(\pi'_{R(E)}) = 8mr + 6m + 4r$.*

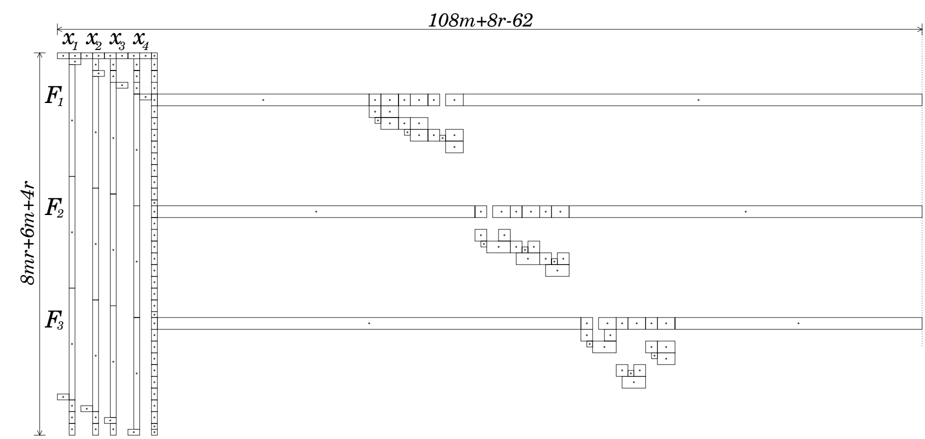
Proof. ($\Rightarrow$) We define π'_R as the minimum width layout satisfying the following. Assuming that E is satisfiable, there is a truth assignment that satisfies E. First, we place each $VR(x_k)$ as Fig. 3(a) if $x_k = true$, or as Fig. 3(b) if $x_k = false$. In either case, $W_x(\pi'_{VR(x_k)}) = 8$ and $W_y(\pi'_{VR(x_k)}) = 8mr + 6m + 4r$ hold. The layout $\pi'_{R'}$ is the same as its initial layout in Fig. 2, where there is no gap between rectangles in the y-direction.

Let $y_{i,j}$ be x_k or $\overline{x_k}$. Since each rectangle in $LR(y_{i,j})$ except for $vs_{i,j}$ has the same y-coordinate as some rectangle in R''_i, and $vs_{i,j}$ has the same y-coordinate as $vc_{k,i}$, d_s of $LR(y_{i,j})$ is 5 if $x_k = true$ and d_s is 3 if $x_k = false$. We place $LR(y_{i,j})$ as Fig. 4(a) or Fig. 5(a) if $d_s = 5$, and as Fig. 4(b) or Fig. 5(b) if $d_s = 3$. From Figs. 4 and 5, we find $W_x(\pi'_{LR(y_{i,j})}) = 10$ if $y_{i,j}$ is *true*, and $W_x(\pi'_{LR(y_{i,j})}) = 12$ if $y_{i,j}$ is *false*.

By the hypothesis,at least one literal in each F_i is *true*. Therefore,$W_x(\pi'_{LR(y_{i,1})}) + W_x(\pi'_{LR(y_{i,2})}) + W_x(\pi'_{LR(y_{i,3})}) \leq 34$, and then $W_x(\pi'_{CR(F_i)}) \leq 108m - 62$



(a) The initial layout ($x_1 = x_2 = x_3 = x_4 = \textit{true}$)



(b) The adjusted layout ($x_1 = x_2 = \textit{true}, x_3 = x_4 = \textit{false}$)

Fig. 8. An example of the layout of $R(E)$.

hold. That is, $W_x(\pi'_{R(E)}) \leq 8r + 2 + (108m - 62) = 108m + 8r - 60$, and $W_y(\pi'_{R(E)}) = W_y(\pi'_{VR(x_k)}) = 8mr + 6m + 4r$ hold.

($\Leftarrow$) Assume that there exists a layout $\pi'_{R(E)}$ of $R(E)$ with $W_x(\pi'_{R(E)}) \leq 108m + 8r - 60$ and $W_y(\pi'_{R(E)}) = 8mr + 6m + 4r$. We show that there exists a truth assignment that satisfies E.

If a clause F_i has both x_k and $\overline{x_k}$, F_i is *true* for any assignment. In the following, we consider a truth assignment for clauses that consists of three literals relevant to distinct variables. For a clause F_i, let each $y_{i,j} (j = 1, 2, 3)$ be x_{k_j} or $\overline{x_{k_j}}$. The sum of the widths of all $VR(x_{k_j})$ and all $LR'(y_{i,j})$ in $\pi'_{R(E)}$ is

$$W_x(\pi'_{R(E)}) - \{\sum_{k\neq k_1,k_2,k_3} W_x(\pi_{VR(x_k)}) + W_x(\pi_{R'}) + (W_x(\pi_{vl_{i,j}})) + (W_x(\pi_{vr_{i,j}}))\}$$
$$\leq (108m - 8r - 60) - \{8(r-3) + 2 + 36(m+i-2) + 36(2m-i-1) + 12\} = 58.$$

Therefore, for some j, the sum of the widths of $VR(x_{k_j})$ and $LR'(y_{i,j})$ is 19 or less. Because of $W_x(\pi'_{VR(x_{k_j})}) \geq 8$ and $W_x(\pi'_{LR'(y_{i,j})}) \geq 10$, $W_x(\pi'_{VR(x_{k_j})}) \leq 9$ and $W_x(\pi'_{LR'(y_{i,j})}) \leq 11$ hold. Since, the hight of the whole layout is $8mr+6m+4r$ and the initial layout restricts $W_y(\pi'_{VR(x_{k_j})}) \geq 8mr+6m+4r$, $W_y(\pi'_{VR(x_{k_j})}) = 8mr+6m+4r$ holds. From $W_x(\pi'_{VR(x_{k_j})}) \leq 9$, m rectangles $vc_{k_j,1}, \cdots, vc_{k_j,m}$ are placed in π'_R without any gap in the y-direction. In this case, all the y-coordinates of m rectangles are the same as either Fig. 3(a) or Fig. 3(b).

We also find that $W_y(\pi'_{R'}) = 8mr+6m+4r$. This implies that the rectangles in R' and in $LR(y_{i,j})$ except for $vs_{i,j}$ do not change their y-coordinates from the initial layout. Therefore, $W_y(\pi_{LR(y_{i,j})}) = 8$ holds.

Now, we consider a partial truth assignment that assigns *true* to x_k if all of $vc_{k,1}, \cdots, vc_{k,m}$ have the same y-coordinates as Fig. 3(a), and assigns *false* to x_k if they have the same y-coordinates as Fig. 3(b). We do not care any other variables. Since $vs_{i,j}$ has the same y-coordinates as $vc_{k,i}$, $d_s = 5$ holds for $y_{i,j}$ if we assign *true* to x_{k_j}, and $d_s = 3$ holds if we assign *false* to x_{k_j}. (Figs. 4 and 5). If $d_s = 5$ and $y_{i,j} = \overline{x_{k_j}}$, then $W_x(\pi'_{LR(y_{i,j})}) \geq 12$ and $W_x(\pi'_{LR'(y_{i,j})}) \geq 12$ hold. If $d_s = 3$ and $y_{i,j} = x_k$, then $W_x(\pi'_{LR'(y_{i,j})}) \geq 12$ hold. Therefore, because of $W_x(\pi_{LR'(y_{i,j})}) \leq 11$, $y_{i,j} = x_{k_j}$ holds in the case of $d_s = 5$, and $y_{i,j} = \overline{x_{k_j}}$ holds in the case of $d_s = 3$. In either case, $y_{i,j}$ is *true*. That is there is a truth assignment that satisfies at least one literal in each clause, that is, E is satisfiable. □

We construct $R^*(E)$ by adding a rectangle $\langle 32mr + 8r, 4\rangle$ at the left side of $R(E)$ so that the upper boundary of this rectangle and $R(E)$ are the same. We show that 3-SAT can be reduced into ILADR using $R^*(E)$.

Lemma 2. *E is satisfiable if and only if there exists a layout $\pi'_{R^*(E)}$, where $\pi'_{R^*(E)}$ satisfy the constraints (1) and (2), and $S(\pi'_{R^*(E)}) \leq (32mr + 108m + 16r - 60)(8mr + 6m + 4r)$.*

Proof. If E is satisfiable, from Lemma 1, $\pi'_{R(E)}$ can be constructed so that $S(\pi'_{R^*(E)}) \leq (32mr + 108m + 16r - 60)(8mr + 6m + 4r)$. Let S be $(32mr + 108m+16r-60)(8mr+6m+4r)$. We show that E is satisfiable if $S(\pi_{R^*(E)}) \leq S$. From the definition of $R^*(E)$, $W_x(\pi'_{R^*(E)}) \geq 32mr + 108m + 16r - 64$ and $W_y(\pi'_{R^*(E)}) \geq 8mr + 6m + 4r$. When $W_y(\pi'_{R^*(E)}) > 8mr + 6m + 4r$,

$$S(\pi'_{R^*(E)}) \geq (32mr + 108m + 16r - 64)(8mr + 6m + 4r + 1)$$
$$= S + 84m - 64.$$

From $m \geq 1$, $84m - 64 > 0$ holds.

Therefore, $W_y(\pi'_{R^*(E)}) = 8mr + 6m + 4r$ and $W_x(\pi'_{R^*(E)}) \leq 32mr + 108m + 16r-60$ if $S(\pi'_{R^*(E)}) \leq S$. In this case, $W_y(\pi'_{R(E)}) = 8mr+6m+4r$, $W_x(\pi'_{R(E)}) \leq 108m + 8r - 60$ hold, and from Lemma 1, E is satisfiable. □

Since ILADR is in NP, we obtain the following theorem.

Theorem 1. *ILADR is NP-complete.*

5 A Layout Adjustment Algorithm

Misue et al. proposed **PFS** (Push Force-Scan Algorithm) in [4], which is a heuristic algorithm to find the minimum area adjusted layout under the constraints that intersections of rectangles should be avoided and the orthogonal order of rectangles should be preserved, for a given rectangle set and its layout. In this section, we show a new heuristic algorithm **PFS′** based on **PFS**. This algorithm obtains an adjusted layout with smaller area than **PFS**.

5.1 Push Force-Scan Algorithm

An algorithm **PFS** uses a measure called a *force* to avoid intersections between rectangles. The force is a vector defined for each pair of rectangles. The force $f_{i,j}$ for rectangles v_i and v_j is used in the way that if two rectangles intersect then $f_{i,j}$ pushes v_j away from v_i. The direction is chosen by experience not only to make v_i and v_j disjoint but to keep the layout as compact as possible and to preserve the orthogonal order.

We define a force and other terminologies, and briefly introduce **PFS**. For a given rectangle set R and its layout π_R, let (x_i, y_i) denote a coordinate of the center of a rectangle $v_i(\in R)$, that is, $\pi_R(v_i) = (x_i, y_i)$. Differences $\Delta x_{i,j}$ and $\Delta y_{i,j}$ of coordinates between v_i and v_j are defined as follows.

$$\Delta x_{i,j} = x_j - x_i,\ \Delta y_{i,j} = y_j - y_i$$

Two different rectangles v_i and v_j intersect each other if the following condition holds.

$$|\Delta x_{i,j}| < \frac{w_i + w_j}{2} \quad \text{and} \quad |\Delta y_{i,j}| < \frac{h_i + h_j}{2}.$$

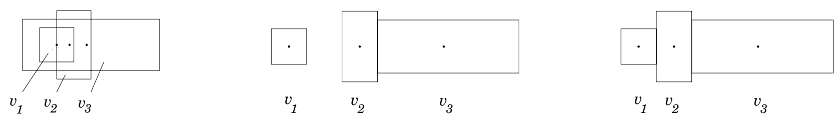
Let L be the line from the v_i's center to the v_j's center. Consider that we move v_j along L to the point where v_j touches v_i without intersections and preserving the orthogonal order. A force $f_{i,j} = (f^x_{i,j}, f^y_{i,j})$ is defined as the vector from (x_i, y_i) to that point. Let $g_{i,j}$ be the gradient of L, that is, $g_{i,j} = \Delta y_{i,j}/\Delta x_{i,j}$ ($g_{i,j} = \infty$ if $\Delta x_{i,j} = 0$). Let $G_{i,j}$ be $(h_i + h_j)/(w_i + w_j)$.

a) The case where v_i and v_j touch with y-direction boundaries, that is, the case of $G_{i,j} \geq g_{i,j} > 0$, $-G_{i,j} \leq g_{i,j} < 0$ or $g_{i,j} = 0$.

$$f^x_{i,j} = \frac{\Delta x_{i,j}}{|\Delta x_{i,j}|}\left(\frac{w_i + w_j}{2} - |\Delta x_{i,j}|\right), f^y_{i,j} = f^x_{i,j} \cdot g_{i,j}$$

```
[Algorithm Horizontal-PFS]
begin
  i := 1;
  while (i < n) do begin
    k := max{j | x_i = x_j};  /* x_i = x_{i+1} = ... = x_k */
    δ := max(0,   max          f^x_{m,j});
             i≤m≤k<j≤n
    for j := k + 1 to n do
      x_j := x_j + δ;
    i := k + 1;
  end;
end.
```

Fig. 9. Algorithm Horizontal-PFS.



(a) An initial layout (b) An adjusted layout by **PFS** (c) An adjusted layout by **PFS′**

Fig. 10. An example of **PFS** and **PFS′** (1).

b) The case where v_i and v_j touch with x-direction boundaries, that is, the case of $(G_{i,j} < g_{i,j}) \wedge (g_{i,j} > 0)$, or $(-G_{i,j} > g_{i,j}) \wedge (g_{i,j} < 0)$.

$$f^y_{i,j} = \frac{\Delta y_{i,j}}{|\Delta y_{i,j}|}\left(\frac{h_i + h_j}{2} - |\Delta y_{i,j}|\right), f^x_{i,j} = f^y_{i,j}/g_{i,j}$$

Now, we introduce **PFS**. **PFS** finds the adjusted layout satisfying the constraints in $O(n^2)$-time $(n = |R|)$. **PFS** applies forces in the x-direction first, then in the y-direction. First one is called Horizontal-PFS and the other is called Vertical-PFS. Vertical-PFS is the same as Horizontal-PFS except for the applied direction. Therefore, we present Horizontal-PFS only.

Horizontal-PFS is shown in Fig. 9. Assume that $x_1 \leq x_2 \leq \cdots \leq x_n$. Horizontal-PFS decides x-coordinates of rectangles in the order $v_1, \cdots, v_n$. The rectangles with the same initial x-coordinate are decided at the same time. When it decides the x-coordinates for $v_i, \cdots, v_k$, it also moves all the rectangles $v_m(i \leq m \leq k)$ and $v_j(k < j \leq n)$ by the same distance in the x-direction. This distance depends on $v_j(k < j \leq n)$ as well as $v_m(i \leq m \leq k)$.

PFS restricts the movement only to the positive direction. Misue et al. also proposed another algorithm, *Push-Pull Force-Scan* algorithm, which allows the movement in the negative direction. This algorithm does not always guarantee the disjointness. Therefore, we do not deal with it.

5.2 The Improvement of PFS

In some case, **PFS** is not efficient. We now consider the case in Fig. 10. Figure 10(a) shows an initial layout, and Fig. 10(b) shows its adjusted layout by

```
[Algorithm Horizontal-PFS′]
begin
  i := 1;
  σ := 0;
  lmin := 1;
  while (i ≤ n) do begin
    k := max{j|x_i = x_j};  /* x_i = x_{i+1} = ··· = x_k */
    γ := 0;
    if (x_i > x_1) then
      for m := i to k do begin
        γ'' := max_{1≤j<i} (γ_j + f^x_{j,m});
        γ' :=
          { σ    if l_bnd(v_m, x_m) + γ'' < l_bnd(v_lmin, x_lmin)
          { γ''  otherwise
        γ := max(γ, γ');
      end;
    for m := i to k do begin
      γ_m := γ;
      x_m := x_m + γ_m;
      if l_bnd(v_m, x_m) < l_bnd(v_lmin, x_lmin) then
        lmin := m;
    end;
    σ := σ + max(0, max_{i≤m≤k<j≤n} f^x_{m,j});
    i := k + 1;
  end;
end.
```

Fig. 11. Algorithm Horizontal-PFS′.

PFS. In this case, first, v_2 and v_3 are moved to the right by $f^x_{1,3}$, and then v_3 is moved by $f^x_{2,3}$ again. Therefore, a needless gap appears between v_1 and v_2.

Here, we propose an algorithm **PFS′**, which obtains an adjusted layout with smaller area than the layout obtained by **PFS**. **PFS′** has the same time complexity $O(n^2)$ as **PFS′**. Similarly to **PFS**, **PFS′** executes Horizontal-PFS′ and then Vertical-PFS′. We show only Horizontal-PFS′.

Again, we consider the example in Fig. 10. In this case, it is sufficient for v_2 to be moved by $f^x_{1,2}$ and for v_3 to be moved by $\max\{f^x_{1,2} + f^x_{2,3}, f^x_{1,3}\}$. **PFS′** generalizes this idea. Assume that $x_1 \leq x_2 \leq \cdots \leq x_n$. Horizontal-PFS′ is shown in Fig. 11. A function $l\_bnd(v_i, x_i)$ is the x-coordinate of the left boundary of v_i when the x-coordinate of v_i is x_i. Horizontal-PFS′ decides x-coordinates of rectangles in the order $v_1, \cdots, v_n$, where the rectangles with the same initial x-coordinates are decided at the same time. When it decides the x-coordinate for $v_i, \cdots, v_k$, the movement distance depends only $v_1, \cdots, v_k$ except for some special case. This is different from **PFS**. We explain how to decide x-coordinates of $v_i, \cdots, v_k$. Assume that x-coordinates of $v_1, \cdots, v_{i-1}$ have been decided. Let γ_j be the distance by which v_j is moved by **PFS′** in the x-direction. Except for the special case mentioned later, Horizontal-PFS′ decides $\gamma_m (i \leq m \leq k)$ as the maximum value of $\gamma_j + f^x_{j,m}$ for $1 \leq j < i$ and $i \leq m \leq k$.

The exception is as follows. Let σ_m be the distance by which $v_m (i \leq m \leq k)$ is moved to the right in **PFS**. The movement γ_m may place some v_m so that the left boundary of v_m is farther left than any other rectangles whose x-coordinates have been decided. In this case, the area may become larger than **PFS**. To avoid this, we decide the movement distance as σ_m instead of γ_m in this case.

5.3 The Validity of the Algorithm

We prove that the area of the layout by **PFS′** is not larger than one by **PFS**, and that the layout by **PFS′** satisfies the constraints (1) and (2).

Let π'_R and π''_R be the layout of R by **PFS** and **PFS′**, respectively. Let x_i, x'_i and x''_i be x-coordinates of v_i in the initial layout, π'_R and π''_R, respectively ($1 \le i \le n$). Let σ_i and γ_i be the distance by which **PFS** and **PFS′** moves v_i in the x-direction, respectively. **PFS** calculates σ_i as follows, where l is the minimum m satisfying $x_i = x_m$.

$$\sigma_i = \delta_0 + \delta_1 + \cdots + \delta_{i-1}$$
$$\delta_i = \begin{cases} 0 & \text{if } i = 0 \text{ or } x_i = x_{i+1} \\ \max(0, \max\limits_{l \le m \le i < j \le n} f^x_{m,j}) & \text{if } x_i < x_{i+1} \end{cases}$$

PFS′ uses the following γ''_i and γ'_i to calculate γ_i, where l is the minimum m satisfying $x_i = x_m$, and

$$\gamma''_i = \max_{1 \le j < l} (\gamma_j + f^x_{j,i})$$
$$\gamma'_i = \begin{cases} \sigma_i \ \text{ if } l\_bnd(v_i, x_i) + \gamma''_i < \min\limits_{j<l} l\_bnd(v_j, x_j + \gamma_j) \\ \gamma''_i \text{ otherwise} \end{cases}$$
$$\gamma_i = \max_{x_i = x_m} \gamma'_m$$

Lemma 3. *For all $i (1 \le i \le n)$, (a) $\sigma_i \ge \gamma''_i$, and (b) $x'_i \ge x''_i$ hold.*

Proof. For all i, we show $\sigma_i \ge \gamma''_i$ and $x'_i \ge x''_i$ by induction. For all i such that $x_1 = x_i$, $\sigma_i = \gamma''_i = \gamma_i = 0$ hold. Therefore, $x'_i = x_i + \sigma_i = x_i + \gamma_i = x''_i$ holds. Let $x_l, \cdots, x_k$ be the maximal sequence with the same x-coordinate. Assume that $x'_j \ge x''_j$, that is $\sigma_j \ge \gamma_j$, for $j < l$. For all i such that $l \le i \le k$, $\gamma_i = \gamma_l$ and $\sigma_i = \sigma_l$ hold. Therefore, it is sufficient to show $\sigma_l \ge \gamma_l$ for $\sigma_i \ge \gamma_i$ and then $x'_i \ge x''_i (l \le i \le k)$. For $l \le i \le k$, γ''_i is calculated as follows.

$$\gamma''_i = \max_{1 \le j < l} (\gamma_j + f^x_{j,i}) \ \le \max_{1 \le j < l} (\sigma_j + f^x_{j,i})$$

Let l_j and k_j be the minimum and the maximum indices such that $x_{l_j} = x_j$ and $x_{k_j} = x_j$ hold, respectively.

$$\begin{aligned} f^x_{j,m} &\le \max_{l_j \le j' \le k_j} f^x_{j',m} \\ &\le \max_{l_j \le j' \le k_j < m' \le n} f^x_{j',m'} \qquad (\because k_j < m) \\ &\le \max(0, \max_{l_j \le j' \le k_j < m' \le n} f^x_{j',m'}) = \delta_{k_j} \end{aligned}$$

Because of $\sigma_j \le \sigma_{k_j}$, then we show $\sigma_i \ge \gamma''_i$ for $l \le i \le k$.

$\gamma''_i \leq \max_{1\leq j<l}(\sigma_j + f^x_{j,i}) \ \leq \max_{1\leq j<l}(\sigma_j + \delta_{k_j}) \ \leq \max_{1\leq j<l}(\sigma_{k_j} + \delta_{k_j}) \ \leq \sigma_{k_j+1} \leq \sigma_l = \sigma_i$

From $\sigma_i \geq \gamma''_i$, $\sigma_i \geq \gamma'_i$ holds. Because of $\sigma_l = \cdots = \sigma_k$, $\gamma_i = \max_{l\leq m\leq k} \gamma'_m \leq \max_{l\leq m\leq k} \sigma_m = \sigma_i$. Therefore, $x'_i \geq x''_i$ holds. □

Lemma 4. *π'_R and π''_R satisfy the following conditions.*

$$W_x(\pi'_R) \leq W_x(\pi''_R) \quad \textit{and} \quad W_y(\pi'_R) \leq W_y(\pi''_R)$$

Proof. We only prove $W_x(\pi'_R) \leq W_x(\pi''_R)$. Let l' be the smallest index among the rectangles whose left boundaries are the left boundary of π'_R. Let r' be the smallest index among the rectangles whose right boundaries are the right boundary of π'_R. We define l'' and r'' for π''_R similarly to l' and r' for π'_R, respectively. It is sufficient to prove that

$$\mathit{left}_{\pi'}(v_{l'}) \leq \mathit{left}_{\pi''}(v_{l''}) \text{ and } \ \mathit{right}_{\pi'}(v_{r'}) \geq \mathit{right}_{\pi''}(v_{r''}).$$

If $x_{l''} = x_1$, then $\gamma_{l''} = \sigma_{l''} = 0$. In this case, $\mathit{left}_{\pi'}(v_{l'}) \leq \mathit{left}_{\pi'}(v_{l''}) = \mathit{left}_{\pi''}(v_{l''})$ hold. Consider the case where $x_1 \neq x_{l''}$. The rectangle $v_{l''}$ is the widest among the rectangles $v_{l_{l''}}, \cdots, v_{k_{l''}}$ with the same x-coordinate. Let $lmin''$ be the value of a variable $lmin$ after **PFS′** decided σ_i for $i = 1, \cdots, l_{l''} - 1$. Since $\mathit{left}_{\pi''}(v_{l''}) \leq \mathit{left}_{\pi''}(v_{lmin''})$, **PFS′** finds $l\_bnd(v_{l''}, x_{l''}) + \gamma''_{l''} < l\_bnd(v_{lmin''}, x_{lmin''} + \gamma''_{lmin''})$. and sets $\gamma_{l''} = \sigma_{l''}$. This implies $\mathit{left}_{\pi'}(v_{l'}) \leq \mathit{left}_{\pi'}(v_{l''}) = \mathit{left}_{\pi''}(v_{l''})$.
From Lemma 3, $\mathit{right}_{\pi'}(v_{r''}) \geq \mathit{right}_{\pi''}(v_{r''})$ holds. Therefore, $\mathit{right}_{\pi'}(v_{r'}) \geq \mathit{right}_{\pi'}(v_{r''}) \geq \mathit{right}_{\pi''}(v_{r''})$ holds. □

Next, we show that the adjusted layout by **PFS′** satisfies the constraints.

Lemma 5. *For any two rectangles $v_i, v_j \in R (i \leq j)$, $\gamma_j - \gamma_i \geq f^x_{i,j}$ holds.*

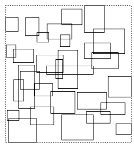
Proof. In the case of $x_i = x_j$, $\gamma_i = \gamma_j$ and $f^x_{i,j} = 0$, $\gamma_j - \gamma_i \geq f^x_{i,j}$ holds. Consider the case of $x_i < x_j$. Let l and k be the minimum and maximum indices such that $x_l = x_j$ and $x_k = x_j$, respectively. For all m such that $l \leq m \leq k$,

$$\gamma''_m = \max_{1\leq i'<l}(\gamma_{i'} + f^x_{i',m}) \geq \gamma_i + f^x_{i,j}.$$

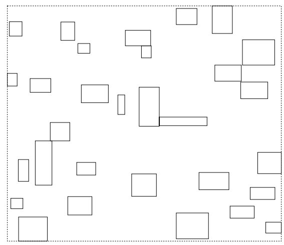
From Lemma 3, $\gamma''_m \leq \sigma_m$ holds, and moreover, $\gamma_j = \max_{l\leq m\leq k} \gamma'_m$ and $\gamma'_m = \sigma_m$ or γ''_m holds. We find $\gamma''_m \leq \gamma_j$ for $l \leq m \leq k$. Therefore, $\gamma_j \geq \gamma_i + f^x_{i,j}$ holds. □

Lemma 6. *The algorithm* **PFS′** *preserves the orthogonal order of the initial layout (the constraint (1)).*

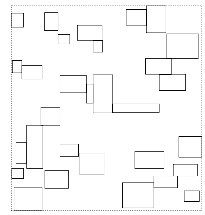
Proof. If $x_i = x_j$, then $x''_i = x''_j$ holds. Consider the case $x_i \neq x_j$. Assume $x_i < x_j$ w.l.o.g. By Lemma 5 and the definition of $f^x_{i,j}$, $\gamma_j - \gamma_i \geq f^x_{i,j}$ and $x_i \leq x_j + f^x_{i,j}$ hold. Therefore, $x''_i = x_i + \gamma_i \leq x_i + \gamma_j - f^x_{i,j} \leq x_j + \gamma_j = x''_j$ holds. □



(a) π_R



(b) (the layout by **PFS**)



(c) (the layout by **PFS**$'$)

Fig. 12. An example of **PFS** and **PFS**$'$ (2).

We can see that the layout by **PFS**$'$ satisfies the constraint (2) from Lemma 5 and the definition of $f^x_{i,j}$. Then we have the following lemma and theorem.

Lemma 7. *Algorithm* **PFS**$'$ *guarantees the disjointness of rectangles (the constraint (2)).*

Theorem 2. **PFS**$'$ *adjusts the layout in* $O(n^2)$ *time, and the result satisfies the constraints (1) and (2) and the area is smaller than the result of* **PFS**.

Example. Fig. 12 illustrates an example of applying **PFS** and **PFS**$'$ for a given set R and its layout π_R. In this case, **PFS**$'$(Fig. 12(c)) obtains much smaller area than **PFS**(Fig. 12(b)).

6 Conclusion

We considered the layout adjustment problem for minimizing the area under the constraints that intersections of rectangles should be avoided and the orthogonal order of rectangles should be preserved, and showed that the corresponding decision problem on the integral coordinate system is NP-complete. We also proposed a heuristic algorithm for this problem applicable to both (on the integral and real coordinate system). Our algorithm obtained smaller area than the algorithm proposed by Misue et al., while both algorithms have the same time complexity.

It would be interesting to find NP-completeness of layout adjustment problems that guarantee different constraints, and to provide much better heuristic algorithms.

References

1. G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis, "Algorithms for drawing graphs: an annotated bibliography," Computational Geometry Theory and Applications, vol. 4, pp. 235–282, 1994.

2. R. Tamassia, G. Di Battista, and C. Batini, "Automatic graph drawing and readability of diagrams," IEEE Trans. on System, Man and Cybernetics, vol. 18, no. 1, pp. 61–79, 1988.
3. P. Eades, W. Lai, K. Misue, and K. Sugiyama, "Preserving the mental map of a diagram," Proc. of COMPUGRAPHICS '91, pp. 34–43, 1991.
4. K. Misue, P. Eades, W. Lai, and K. Sugiyama, "Layout adjustment and the mental map," Journal of Visual Languages and Computing, vol. 6, pp. 183–210, 1995.
5. M-A. D. Storey and H. A. Müller, "Graph layout adjustment strategies," Proc. Graph Drawing '95, LNCS 1027, pp. 487–499. Springer-Verlag, 1995.
6. M. R. Garey and D. S. Johnson, "Computers and Intractability — A Guide to the Theory of NP-Completeness," W. H. Freeman and Company, New York, 1979.