アルゴリズム 55－6
（1997. 1. 23）

与えられた制約を満たす矩形双対グラフ描画手法

松本匡史　水野健二　渡邉敏正

広島大学工学部 第二類 回路システム工学講座
〒739 東広島市鏡山一丁目 4-1
(電話) 0824-24-7662 (渡邉)
(ファクシミリ) 0824-22-7195
(電子メール) watanabe@huis.hiroshima-u.ac.jp

概要： 矩形双対グラフとは，一つの矩形のいくつかの部分矩形への分割であり，PTP グラフと呼ばれる平面グラフの一つの幾何学的双対グラフ（つまり PTP グラフの頂点を矩形，辺を矩形間の隣接関係として表したグラフ）である．本稿では，矩形双対グラフの各部分矩形の縦横サイズが予め与えられた下限値を満たし、且つ全体矩形の面積が最小あるいは極小になるような矩形双対グラフの（発見的）描画手法を提案し，実験によりその有効性を示す．

キーワード： グラフ描画，矩形双対グラフ，PTP グラフ，矩形サイズ，面積最小化

Drawing a Rectangular Dual to Meet Prescribed Constraints

Tadafumi Matsumoto, Kenji Mizuno and Toshimasa Watanabe

Department of Circuits and Systems, Faculty of Engineering, Hiroshima University
4-1, Kagamiyama 1-chome, Higashi-Hiroshima, 739 Japan
Phone : +81-824-24-7662 (Watanabe)
Facsimile : +81-824-22-7195
E-mail : watanabe@huis.hiroshima-u.ac.jp

Abstract : A rectangular dual is a dissection of a rectangle into several subrectangles, representing a geometrical dual of a plane graph, called a PTP graph, where each subrectangle corresponds to a vertex of this PTP graph, and two subrectangles share a boundary if and only if the corresponding two vertices are adjacent in the graph. The subject of the paper is to propose a heuristic method for drawing a rectangular dual so that the length and the width of the whole rectangle may be (optimally or nearly) minimized, under the condition that those of each subrectangle are no less than given lower bounds. Experimental results show that the proposed method produces sharp approximate solutions very quickly.

Key words : graph drawing, rectangular duals, PTP graphs, rectangle sizes, area minimization

1 Introduction and Motivation

A *rectangular dual* [6] is a dissection of a rectangle into several subrectangles, representing a geometrical dual of a PTP graph (whose definition will be given later), where each subrectangle corresponds to a vertex of this PTP graph, and two subrectangles share a boundary if and only if the corresponding two vertices are adjacent in the graph (Fig.1). It has been widely used in placement algorithms for VLSI design [9].

Since the two subrectangles, which correspond to the two vertices of an edge, share a part of boundary, if this edge represents connection requirement then a layout, which assures feasible routing required by a given PTP graph, can be obtained. In [5, 12] for example, it is pointed out that rectangular duals are useful in designing printed wiring boards. Each of elements such as ICs, resistors and condensers is actually placed within the corresponding subrectangle, imposing lower bounds on the length and width of each subrectangle. Hence this application requires capability of drawing a rectangular dual such that the length and width of the whole rectangle are minimized, under the condition that those of each subrectangle are no less than given lower bounds. This is a quadratic programming problem, for which several optimization techniques are existing.

The subject of the paper is to propose a heuristic algorithm for this problem. Experimental results show that the proposed one produces sharp approximate solutions very quickly. Existing algorithms for optimum solutions become very slow as the number of vertices or edges increases. On the other hand, there are some choices on which the size of the resulting whole rectangle heavily depends, such as selecting four corners, deciding which side has the lower bound on length (or width), and so on. Furthermore, incorporating routing area may cause increase in the size of the whole rectangle. Consequently it is a pair of fast and sharply estimating the whole rectangle, with subrectangles of feasible size, and efficiently improving drawing of rectangular duals that meet our requirement in printed wiring board design utilizing rectangular duals. This motivated and led us to do research on the subject of this paper. The early version appeared in [7].

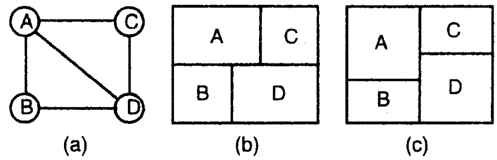

Figure 1: (a)An example of a PTP graph. (b),(c) Two different rectangular duals of this PTP graph.

We briefly summarize known results on the subject. A linear time algorithm to construct a rectangular dual of a given PTP graph [1, 3]; an algorithm to draw a given plane graph on the grid [2] and its extension to handle rectangular duals without constraints on subrectangles [8]; an algorithm to enumerate all rectangular duals of a given PTP graph [9]; an algorithm to find one of the most area-efficient drawings of a given rectangular dual with all constraints on subrectangles satisfied, as well as an algorithm to estimate the size of the whole rectangle [10]. These algorithms of [9, 10] spend a huge amount of computation time as the number of subrectangles increases, as mentioned in [7]. The critical point is that, even if any area-efficient drawing is obtained, we cannot always place any element inside the corresponding subrectangle, since the constraints on each subrectangle are given by lower bounds on the area size and its aspect ratio.

2 PTP graphs and Constraints

2.1 PTP graphs and rectangular duals

A *properly triangulated planar* (*PTP*) *graph* [1] is a connected planar graph satisfying P1-P3:

P1: Every face (except the exterior) is a triangle.

P2: All internal vertices have degree ≥ 4.

P3: All cycles that are not contours of faces have length ≥ 4.

In this paper, we assume that any given PTP graph is biconnected. (If a given PTP graph is not biconnected then, by adding some edges, we can make it biconnected.) The subrectangle representing a vertex v of a PTP graph is denoted by Rv. For notational simplicity in the following, we often denote Rv as just v unless any confusion arises.

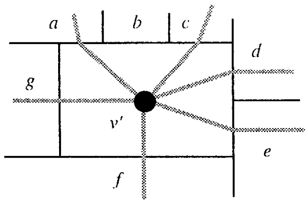

Figure 2: Schematic explanation of a virtual vertex representing a multi-terminal net.

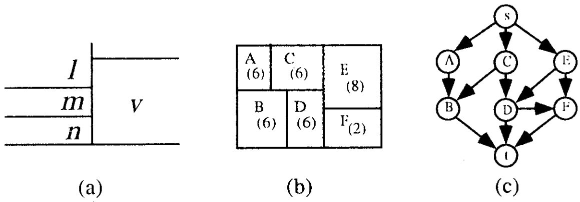

Figure 3: (a) The constraint left(v). (b) A rectangular dual with size_y(v) shown within parentheses. (c) The vertical constraint graph.

2.2 Physical conditions

In printed wiring board design based on rectangular duals, each element is placed inside its corresponding subrectangle and other subrectangles are used for routing or just for placing terminals. Therefore, lower bounds of their sizes are given as a physical condition. More specifically they are as follows:

- For each edge (u, v) of a PTP graph, Ru and Rv share one boundary by length no less than width(u, v), the width of the wire between u and v.
- Each subrectangle v representing an element has lower bounds height(v) and width(v) on its height and width, respectively.
- Each subrectangle v', which represents a virtual vertex inserted for decomposition of a multi-terminal net into a set of two-terminal nets, has lower bounds height(v') and width(v') on its height and width, respectively. In Fig.2, height(v') and width(v') are given as follows:
  height$(v') \leftarrow max\{$width(v', d) + width(v', e), width$(v', g)\}$
  width$(v') \leftarrow max\{$width(v', a) + width(v', b) + width(v', c), width$(v', f)\}$
  (In this example, width(v', b)=0.)

3 Estimating the size of a whole rectangle

Given a drawing of a rectangular dual with each subrectangle satisfying the constraints, we estimate the sizes of the whole rectangle. As stated in Section 1, the method proposed in [10] for such estimation does not meet our purpose. In addition, this method is weak in identifying one or more subrectangles that may be bottlenecks in reducing the present size of the whole rectangle.

The propose algorithm can find such subrectangles easily by searching all longest paths and, therefore, it is very useful to modify drawings of rectangular duals. Since each of the vertical and horizontal operations can be executed independently, the vertical operation is exclusively explained in the following.

We estimate the height size_y(v) of each Rv as follows. First, we initialize size_y(v) = height(v) for all vertices v. Secondly, the whole rectangle is divided into some columns and we process each column one by one from left to right. For each Rv in the current column, the constraint from the left, denoted by left(v), to be imposed on size_y(v) is computed by checking subrectangles sharing a part of the left boundary of Rv. For example, if v adjoins l, m and n as shown in Fig.3 (a) then we set

$$\text{left}(v) = \text{width}(l, v) + \text{size\_y}(m) + \text{size\_y}(n)$$

If size_y(v) < left(v) then size_y(v) $\leftarrow$ left(v). Execute similar computation by conversely processing each column from right to left, and obtain the constraint from the right, right(v), for each Rv. If size_y(v) < right(v) then size_y(v) $\leftarrow$ right(v). Then set

$$\text{size\_y}(v) = max\{\text{height}(v), \text{left}(v), \text{right}(v)\}$$

Thirdly, find all longest paths from the top boundary to the bottom one of the given whole rectangle. The boundaries of the whole rectangle are called the external boundaries. We make a directed graph such that each vertex represents a subrectangle and such that each directed edge (v, w) from v to w satisfies (i) or (ii):

(i) Rv adjoins Rw so that they may share a part of the top boundary of Rw;

(ii) Rv adjoins Rw so that (1) through (3) may be satisfied:

(1) they share a part of the left or the right boundary of Rw,

(2) the top boundary of Rv is located strictly upper than that of Rw, and

(3) the bottom boundary of Rv is located no lower than that of Rw.

Finally we add new vertices s and t which represent the top and the bottom boundaries of the whole rectangle, respectively. A directed edge is added from s to each subrectangle sharing the top external boundary, and from each subrectangle sharing the bottom external boundary to t.

We call this graph the *vertical constraint graph* of the rectangular dual (see Fig.3 (b), (c)). We find all longest paths of the vertical constraint graph in which the following cost(v) is assigned to each vertex v, where the term "longest" means that the total sum of costs of vertices is maximum.

Let cost(v) be the length from the top external boundary to the bottom boundary of Rv: cost(v) will be computed by processing subrectangles in each column from the top to the bottom. If Rv is the first to be processed then $\text{cost}(v) \leftarrow \text{size\_y}(v)$. Any other subrectangle Rw will be processed as follows:

$$\text{cost}(w) \leftarrow \begin{cases} max\{\text{cost}(v) + \text{size\_y}(w), \\ \qquad \text{cost}(w)\} \text{ in (i)} \\ max\{\text{cost}(v) + \text{left}(w) - \text{width}(v, w), \\ \qquad \text{cost}(w)\} \text{ in (ii)} \end{cases}$$

With these costs on vertices except s and t of the (vertical) constraint graph, we find all longest paths from s to t, and, for every edge (u,v) on each path, we maintain parent $[v] = u$. In order to modify drawings of a rectangular dual efficiently, we use the *(vertical) longest path graph.* The longest path graph is a directed graph consisting of vertices and edges contained in any longest path. In Fig.3 (c), if the edge (E,F) is deleted then the longest path graph is obtained. Reducing the cost(v) of any vertex v in a longest path may result in decrease in the (vertical) size of the whole rectangle.

Computing left(v) for all subrectangles v and constructing the constraint graph can be done in $O(|V| + |E|)$ time, where $|V|$ and $|E|$ are the number of vertices and of edges in a given PTP graph. We can find all longest paths and construct the longest path graph in $O(|V_c| + |E_c|)$, where $|V_c|$ and $|E_c|$ are the number of vertices and edges in the constraint graph, and $|V_c| \leq |V|$. Since both the PTP graph and the constraint graph are planar graphs, $|E|$ or $|E_c|$ is $O(|V|)$ or $O(|V_c|)$, respectively. Hence the total time of this estimating operation is $O(|V|)$.

4 Modifying drawings of rectangular duals

By utilizing the estimate and the longest path graph given in Section 3, we explain how to reduce sizes of drawings of rectangular duals. We compare prescribed sizes with those estimated ones, and the direction (that is, vertical or horizontal) with larger difference is selected as the target of reduction. Next, for each edge of the longest path graph, we try to apply any one of the three local modifications to be given in the following. For each edge, we estimate how the local modification under consideration can reduce the whole rectangle size in the target direction. The local modification will be applied to the edge or the vertex having the largest estimated value in the longest path graph. These operations are repeated until estimated size falls within the prescribed size or reduction cannot be obtained any more.

Each local modification can be done in $O(1)$ time. Both checking whether or not any local modification can reduce the whole rectangle size and estimating the reduction size can be done in $O(1)$ time. These checking and estimating operations are executed on the longest path graph. Hence the time complexity of local modification is $O(|V_l| + |E_l|)$, where $|V_l|$ and $|E_l|$ are the number of vertices and of edges in the longest path graph, and $|V_l| \leq |V|$. Since the longest path graph is a planar graph, $|E_l|$ is $O(|V_l|)$. Local modification may be repeated if necessary. Hence the total time of mod-

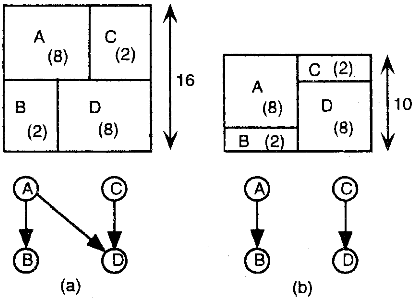

Figure 4: Modifying PDGs.

ifying a drawing is $O(k|V|)$, where k is the number of repetition.

We explain each of the four local modifications in the vertical direction only.

4.1 Modifying PDGs

Suppose that we construct a rectangular dual by using the method proposed in [1]. This method constructs a directed graph called the *path digraph* (PDG) from a given PTP graph. Then, a drawing of its rectangular dual is obtained based on this PDG. The directed edges of the PDG reflect the "on top of" relation defined by the rectangular dual. (See [1] for the details.)

Lemma 4.1 [1] If (i,j) is an edge in a PTP graph, then:

1. i is not a distant ancestor of j in any PDG.
2. j is not a distant ancestor of i in any PDG.

In this section, we execute modifications without violating Lemma 4.1.

In general, a PTP graph has one or more PDGs, each representing a rectangular dual. We consider reducing the height and/or width of the whole rectangle by modifying adjacency structure of a given PDG.

Fig.4 (a) shows a rectangular dual and its PDG. In this figure, by transforming the rectangular dual (a) to (b), the height of the whole rectangle is reduced from 16 to 10. This is a basic operation for modifying a PDG. This idea of transformation has already appeared in enumerating all possible rectangular drawings in [9].

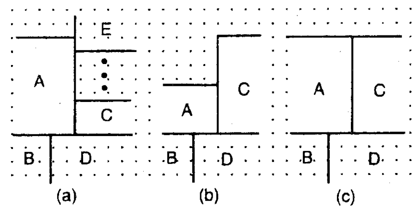

Figure 5: Adjacency relations for subrectangles A and C in the PDG of Fig.4.

First, we find four vertices A, B, C, D which have adjacent relations as shown in Fig.4 (a) in the vertical longest path graph, where the edge (A,D) is contained in the longest path graph. Secondly, these four vertices are divided into two pairs, the upper pair A, C and the lower one B, D. We estimate how this height can be reduced. We explain this estimation only for the upper pair A, C, and similarly for the other pair. There are three situations as shown in Fig.5 (a)-(c). For each of these cases, the feasibility condition for reduction and its estimated value of reduction are given as follows.

Case (a)

feasibility condition : size_y(A) > right(A)

estimated value : size_y(A) − right(A)

Cases (b) and (c)

feasibility condition : size_y(C) = left(C)

estimated value : *min*{size_y(A)−width(A, C), size_y(C)−*max*{height(C), right(C)}}

Finally, we compare the estimated value of the upper pair A,C with that of the lower pair B,D, and the smaller one is adopted as the estimated value of these four vertices.

There is possibility that the transformation for reducing the height may increase the width. Our method tries to keep such a secondary effect as small as possible. By using the costs computed in estimating the height of the whole rectangle, we can estimate the increase of the width, and we choose the one with the smallest increase in the width. Let cost_l(v) (cost_r(v), respectively) be the distance from the left (right) external boundary

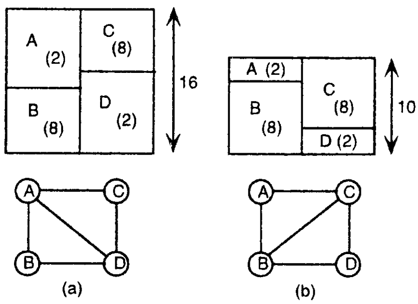

Figure 6: Modifying PTP graphs.

to the right (left) boundary of Rv. The only possible secondary effect of this transformation concerning A, B, C, D is that an edge (A,D) may be added to the horizontal constraint graph, possibly resulting in inserting the edge to the horizontal longest path graph. The length of this path in the horizontal longest path graph can be represented as cost_l(A) + cost_r(D).

Similarly we estimate another secondary effect of increasing the height in the transformation to reduce the width of the whole rectangle, and we execute the one with the least secondary effect.

4.2 Modifying PTP graphs

Usually some vertices or edges, called virtual vertices or virtual edges, are added to any given planar graph in order to construct a PTP graph. Virtual edges can be added arbitrarily as long as a PTP graph is obtained. By making use of this arbitrariness, we change incidence relation of virtual edges so that the height or width of the whole rectangle may be reduced.

In the PTP graph in the lower part of Fig.6 (a), let (A,D) be a virtual edge. The corresponding rectangular dual is shown in the upper part. By transforming the PTP graph from (a) to (b), the height of the whole rectangle can be reduced from 16 to 10. This is a basic operation to be used in modifying a PTP graph.

One important point to be taken into consideration is that changing the incidence relation of a given PTP graph may result in either

(i) a graph that is not a PTP graph, or

(ii) a PTP graph that has no rectangular dual.

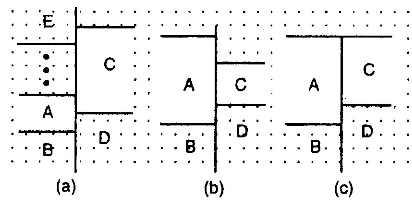

Figure 7: Adjacency relations for subrectangles A and C in the PTP graph of Fig.6.

A necessary and sufficient condition for any PTP graph to have a rectangular dual is given in [6], and we have to check this condition. The details are omitted due to shortage of space: see [7].

First, find four vertices A, B, C, D as shown in Fig.6 (a), where the edge (A,D) is a virtual edge in the vertical longest path graph. Secondly, we check whether or not changing incidence relation breaks the necessary and sufficient condition of [6]. If it holds then divide four vertices into two pairs, the upper pair A, C and the lower one B, D. For A and C in Fig.6, there are three situations as shown in Fig.7 (a)-(c). Then the feasibility condition for reduction and its estimated value are given as follows.

Case (a)

feasibility condition : size_y(C) > left(C)

estimated value : size_y(C) − left(C)

Cases (b) and (c)

feasibility condition : size_y(A) = right(A)

estimated value : min{size_y(C)−width(A, C), size_y(A)−max{height(A), left(A)}}

We compare the estimated value of the upper pair A,C with that of the lower pair B,D, and the smaller one is adopted as the estimated value of these four vertices.

There is no secondary effect, differently from the PDG transformation.

4.3 Combination of the two local modifications

Since the modification to be presented in this section is a combination of the two previous modifications, their conditions and estimated values can

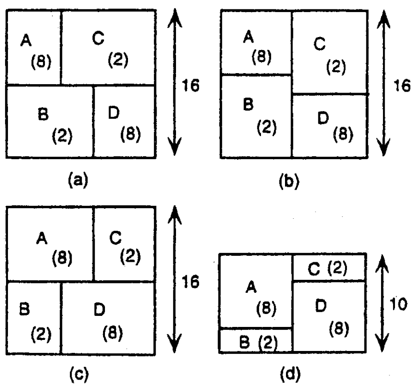

Figure 8: The two-step modification.

be used. In Fig.8, transforming (a) to (b) does not reduce the height of the whole rectangle. Now transform (a) to (c) instead. The height of the whole rectangle is still unchange. Transforming (c) to (d), however, can reduce the height from 16 to 10. We can modify drawings by using this two-step transformation as a basic operation. In Fig.8, only transformation from (c) to (d) can reduce the height, and this is the same as the PDG modification shown in Section 4.1. The feasibility condition and the estimated value can be considered independently for each pair of {A,C} and {B,D}. Hence, even for the adjacency relation shown in Fig.8 (a), we transform it to (c) and then we try to excute the size reduction in the same way as the PDG modification. Similarly for estimating the secondary effects.

4.4 Rotation of elment-subrectangles

An element-subrectangle is a subrectangle in which an element is actually placed. Unless the direction of placing the element is specified, we can interchange the lower bounds on the height and the width so that the size of the whole rectangle may be reduced. Fig.9 shows that rotating the lower left subrectangle with (6,3) in (a) to the one with (3,6) in (b) can reduce the size from 12×7 to 9×9.

We explain only reduction of height(v). Let size_x(v) be the current width of Rv. And let up(v) or down(v) denote the constraint, to be imposed on size_x(v), from the top or the bottom, defined similarly to left(v) or right(v) by interchanging columns, the left and the right with rows, the

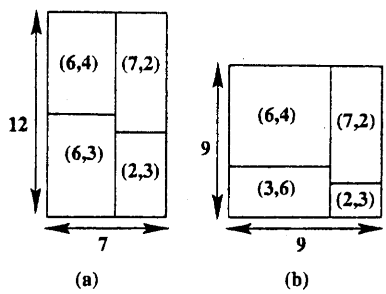

Figure 9: Reducing the size by rotating an element-subrectangle, where the pair (height,width) denotes the corresponding lower bounds.

top and the bottom, respectively. The feasibility condition for reduction, its estimated value and the secondary effect are given as follows.

feasibility condition :

height(A) $> max\{$width(A), left(A), right(A)$\}$

estimated value :

height(A) $- max\{$width(A), left(A), right(A)$\}$

secondary effect :

$max\{$height(A),up(A),down(A) $\}$ $-$ size_x(A)

5 Experimental results

The proposed method have been implemented on a personal computer GATEWAY2000 (CPU: Pentium/120MHz, OS: FreeBSD 2.1) with the C programming code. Randomly generated circuits are used as input data.

Let Sa denote the whole rectangle area estimated by our method, while let Sb denote the one obtained by applying Lemke's method [4]. For each randomly generated circuit, Table 1 shows the average, the minimum and the maximum of the ratio Sa/Sb. The column "CT(s)" shows the average computation time (in second) of our method and the Lemke's method over 50 input data, where the graph models have the number of vertices in the intervals given in the column "#node".

We have actually constructed layouts of printed wiring boards from rectangular duals by using MULTI-PRIDE [11, 12]. Let La denote layout area obtained from the drawings of rectangular duals given by our method. Let Lb denote layout area obtained by applying a linear programming (as in

Table 1: Experimental results on Sa/Sb.

#node	Ave	Min	Max	CT(s)	
				MW	Lemke
1-100	0.99	0.94	1.03	1.88	5.27
101-200	0.98	0.95	1.01	5.62	68.40
201-300	0.96	0.89	1.00	14.53	265.37
301-400	0.96	0.96	1.01	24.46	946.52

Table 2: Experimental results on La/Lb.

#node	Average	Minimum	Maximum
1-100	0.89	0.63	1.00
101-200	0.92	0.78	0.98
201-300	0.93	0.89	0.99
301-400	0.90	0.88	0.99

[11, 12]) to rectangular duals given by the method of [1]. Table 2 shows the average, the minimum and the maximum of the ratio La/Lb over 50 input data, where the graph models have the number of vertices in the intervals given in the column "#node".

It is observed that our method produces sharp approximate solutions to the quadratic programming problem in shorter time.

6 Concluding remarks

This paper has proposed a heuristic method for drawing a rectangular dual so that the length and the width of the whole rectangle may be (optimally or nearly) minimized, under the condition that those of each subrectangle are no less than given lower bounds. Experimental results show that the proposed method produces sharp approximate solutions very quickly.

Some problems left for future research are as follows:

- improving of the proposed method so as to make the whole rectangle closer to the prescribed or mimimum size;
- providing a method to estimate the board size with area of routing incorporated.

Acknowledgments

The research of T.Watanabe is partly supported by the Grant in Aid for Scientific Research of the Ministry of Education, Science and Culture of Japan: under (C) 08680371 and (A) 07308028.

References

[1] Bhasker, J. and Sahni, S., "A linear algorithm to find a rectangular dual of a planar triangulated graph", Technical Report TR 85-26, Computer Science Dept., University of Minnesota, Minneapolis, 1985. Also appeared in Proc.23th Design Automation Conference, pp. 108-114, 1986.

[2] Chrobak, M. and Nakano, S., "Minimum-width grid drawings of plane graphs", Technical Report, UCR-CS-94-5, Department of Computer Science, University of California at Riverside, 1994.

[3] He, X., "On finding the rectangular duals of planar triangulated graphs", SIAM J. Comput., 22(6), pp. 1218–1226, 1993.

[4] Ibaraki, T. and Fukushima, M., "Saitekika Puroguramingu (Optimization Programming)", Iwanami shoten, Tokyo, 1991. (in Japanese)

[5] Iwamoto, K., Watanabe, T., Araki, T. and Onaga, K., "Finding Jumpers in Printed Wiring Board Design for Analog Circuits", Proc.1991 IEEE Iut. symposium on Circuits and Systems, pp. 2854-2857, 1991.

[6] Kozminski, K. and Kinnen, E., "Rectangular duals of planar graphs", Networks, Vol.15, pp. 145-157, 1985.

[7] Matsumoto, T. and Watanabe, T., "Drawing Rectangular duals within a Rectangle of Prescribed Sizes", Tech. Rep. of IEICE of Japan, CAS95-101, pp. 9–16, 1996. (in Japanese)

[8] Rahman, M. S., Nakano, S. and Nishizeki, T., "Rectangular Grid Drawings of Plane Graphs", IPSJ, SIG Notes, IPS of Japan, 95-AL-48, pp. 39–46, 1995.

[9] Tani, K., Tsukiyama, S. and Shirakawa, I., "An Algorithm to Enumerate All Rectangular Dual Graphs", Tech. Rep. of IEICE of Japan, CAS86-178, pp. 31–38, 1986. (in Japanese)

[10] Tani, K., Tsukiyama, S. and Shirakawa, I., "Area-Efficient Drawings of Rectangular Dual Graphs for VLSI Floor-Plan", In Proc.8th Mathematical Programming Symposium, Japan, pp. 59–68, 1987. (in Japanese)

[11] Yasui, T., "A Support System MULTI-PRIDE for Designing Multi-Layered Printed Wiring Boards", Master's thesis of Information Engineering Major, Graduate School of Engineering, Hiroshima Univ., 1993.

[12] Yasui, T., Tōyama, N., Une, Y., Watanabe, T. and Onaga, K., "MULTI-PRIDE: A Support System for Designing Multi-Layered Printed Wiring Board Layouts of Analog Circuits", Proc. DA Symposium'92, IPS of Japan, pp. 137–140, 1992. (in Japanese)