# DOUBLE-ROW PLANAR ROUTING AND PERMUTATION LAYOUT

by

S. Tsukiyama and E. S. Kuh

Memorandum No. UCB/ERL M80/46

1 July 1980

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

# Double-Row Planar Routing and Permutation Layout*

By

Shuji TSUKIYAMA† and Ernest S. Kuh††

† Department of Electronic Engineering, Osaka University, Suita Osaka, 565 Japan.

†† Department of Electrical Engineering and Computer Sciences, and Electronics Research Laboratory, University of California, Berkeley California 94720.

* Research supported by the National Science Foundation Grant ENG-78-24425.

Abstract

Problems on layout for IC's (integrated circuits) and PCB's (printed circuit boards) are usually solved by heuristic approaches because they are complex. This paper considers a special problem of double-row planar routing. The problem represents a generalization of the permutation layout problem to which estimation of bounds and some algorithms have been proposed recently.

Our approach is based on the interval graphical representation introduced in the single-row single-layer PCB problem. The objective function for minimization is the breadth of the realization, i.e., the total number of vertical tracks required to realize a given net list specified in terms of terminals on two parallel rows.

The problem is shown to be intractable in the sense of NP-completeness; however, a polynomial-time heuristic algorithm is proposed. An upper bound for the breadth for an initial solution is given. Iterative improvement is next used. The algorithm has been programmed in FORTRAN and ran on the VAX 11/780 computer.

## 1. Introduction

This paper deals with the problem of double-row planar routing: Given a net list defined on the nodes placed on two parallel horizontal lines, connect all nodes in every net by conductor lines which are laid on a single layer. This problem is not only of theoretical interest as a generalization of the single-row single-layer routing problem[4,6,7] but also has possibities for practical applications to the wiring of PCB (Printed Circuit Board), to the layout for hybrid IC's, and to the routing of gate arrays (the master slice).

The permutation layout problem introduced by Cutler-Schiloach[1] can be considered as a special case of the double-row planar routing problem. In this paper, the breadth, i.e., the total number of the intersections between a row and the conductor lines is used as a criterion for optimization in realizing a given net list. In Reference [1] three types of permutation layout are given, namely: packed-packed layout, packed-spaced layout, and spaced-spaced layout. Among them, only in the packed-packed layout, a good algorithm for realization has been proposed. The packed-spaced layout algorithm is similar to the routing method in [6,7]. However, no algorithm was proposed for the spaced-spaced layout, which is the most general case.

As pointed out by Shirakawa[5], the permutation layout problem can be transformed into a single-row single-layer routing problem. Thus the technique introduced in the single-row case can be used to solve the double-row problem. In this paper, we introduce a more sophisticated transformation and solve the double-row problem which is a generalized spaced-spaced permutation layout. The crucial concept is the interval graphical representation introduced in Ref. [4].

In Section 2, necessary terminology and concepts are given, and the double-row planar routing problem is formulated as two subproblems with the use of the interval graphical representation. In Section 3, we analyze the computational complexities of these subproblems and show that these are intractable problems in the sense of NP-completeness[6]. We then devise a polynomial time algorithm for one of two subproblems, which is described in Section 4. By using this algorithm as a subroutine, we propose a heuristic algorithm to the double-row planar routing problem in Section 5. In order to evaluate the performance of the proposed algorithm, the upper bound for breadth in the initial solution is discussed, which is shown to be slightly better than that in Reference [1]. In Appendix I, the proof of Theorem 1 concerning NP-completeness is given, and in Appendices 2 and 3, the details of the algorithms in Sections 4 and 5 are discussed, respectively.

## 2. Preliminary Definitions

### 2.1 Double-Row Routing Problem

Let us consider two parallel horizontal lines called upper and lower rows, respectively, and consider nodes placed on these rows, as shown in Fig. 1(a), where r is the number of nodes on the upper (and the lower) row. A net is a set of nodes to be connected by conductor lines which are composed of horizontal and vertical line segments. A net list is a set of disjoint nets, and the total number of nets in the net list is denoted by n. In the following, we represent a given net list $\mathcal{L}$ by a combination of two sequences U and W of nets, (i.e., $\mathcal{L} = [U,W]$), where $U = (N_1^u, N_2^u, \cdots, N_r^u)$ and $W = (N_1^w, N_2^w, \cdots, N_r^w)$. Thus $N_i^u$ represents the net which contains the node $u_i$ in the upper row, and similarly, $N_i^w$ represents the net which contains the node $w_i$ in the lower row.

A realization of net list is a set of conductor lines, each of which connects all nodes in a net and does not intersect any other conductor lines. Fig. 1(b) shows a realization of net list $\mathcal{L} = [(1,2,3,4,2,3,5,6),(1,3,4,5,1,6,4,6)]$, where a net is represented by an integer. In this case, there are six nets altogether, where the first net is $N_1^u = N_1^w = N_5^w$ and the sixth net is $N_8^u = N_6^w = N_8^w$. In the realization of a net list $\mathcal{L}$, a crossing number $X_u$ on the upper row is the number of intersections between the upper row and the conductor lines, not counting the intersections at nodes. A crossing number $X_w$ on the lower row is defined similarly. For example, in a realization shown in Fig. 1(b), $X_u = 2$ and $X_w = 3$. A crossing number X is the maximum of $X_u$ and $X_w$, and a breadth is the sum of X and (r-1), where r-1 is the breadth if the realization does not have any intersections.

The double-row planar routing problem (abbreviated DRP) that we shall consider in this paper is stated as follows:

DRP Problem: Given a net list $\mathcal{L} = [U,W]$, find a realization with the minimum breadth.

Since we are considering the case where the number of nodes on the upper and the lower rows are equal, we can state the problem by using the crossing number X instead of the breadth. It is easy to show that our treatment can be generalized to the problem where the number of nodes on the upper and the lower rows are different. We also want to point out that a spaced-spaced layout defined in [1] is a special case where both sequences U and W consist of distinct nets.

Now, we impose a restriction on the pattern of the conductor lines.

Restriction: We do not allow the conductor line for a net to run from the upper row to the lower row more than once.

From the practical point-of-view, this is a reasonable assumption because we always prefer to minimize the total wire length in any layout design. The restriction is illustrated in Fig. 1(c), in which (I) is allowed, but not (II) or (III). With this assumption, we can see that there always exists a horizontal line between the upper and lower rows, which intersects exactly once with any conductor line of net that connect nodes in the upper and lower rows as shown in Fig. 1(b). Let us call such a horizontal line the middle row. Let $M = (N_1^m, N_2^m, \cdots, N_c^m)$ be a sequence of nets on the middle row such that for $1 \le i \le c$, $N_i^m$ indicates the net of the ith intersection.

We take the following approach to attack the problem DRP:

Step I: Construct a sequence $M = (N_1^m, N_2^m, \cdots, N_c^m)$, consisting of the nets which have nodes on both the upper and the lower rows. Then, create two DRP problems with net list $\mathcal{L}_u = [U, W_u{=}M]$ and $\mathcal{L}_w = [U_w{=}M, W]$.

Step II: Find a realization of $\mathcal{L}_u = [U, W_u{=}M]$ such that the breadth (i.e., $X_u$) is minimal and in addition, there exists no conductor line below the middle row as shown in Fig. 2(a). Also, find a realization of $\mathcal{L}_w = [U_w{=}M, W]$ such that the breadth (i.e., $X_w$) is minimal and there exists no conductor line above the middle row as shown in Fig. 2(b). For example, for the net list $\mathcal{L}$ given in Fig. 1(b), if sequence (3,1,4,5,6) is generated as M, and if realization of $\mathcal{L}_u$ and $\mathcal{L}_w$ are given as shown in Fig. 2(a) and (b), respectively, then the realization of $\mathcal{L}$ is obtained by combining these two realizations, which is the same as in Fig. 1(b).

The problem of finding a realization of $\mathcal{L}_w$ is similar to that of $\mathcal{L}_u$, since the problems are the same if we turn two rows of $\mathcal{L}_w$ upside down. Thus, our problem of DRP is reduced to the following two problems:

Half-DRP Problem: Given a net list $\mathcal{L}_u = [U,M]$, find a realization with a minimum crossing number X ( $= X_u$) such that no conductor line passes below the lower (i.e., middle) row.

Middle Sequence Problem (MSP): Given a net list $\mathcal{L} = [U,W]$, find a sequence M which minimizes max[ $X_u$, $X_w$ ], where $X_u$ and $X_w$ are the minimum crossing numbers of realizations of $\mathcal{L}_u = [U,M]$ and $\mathcal{L}_w = [M,W]$, respectively.

## 2.2 Interval Graphical Representation

In this section, we introduce the interval graphical representation[4] for formulating and solving the Half-DRP problem.

Given a net list $\mathcal{L}_u = [U,M]$, let us consider two subsequences $M_L$ and $M_R$ of M with $M_L \_ M_R = M$, where A _ B represents a concatination of two sequences. Construct a sequence $M_L \_ U \_ M_R$ by concatinating $M_L$, U, and $M_R$ in this order, and consider $(r+c)$ nodes on the single row as single-row problem. For example, given a net list $\mathcal{L}_u = [U,M]$ and subsequences $M_L$ and $M_R$ shown in Fig. 3(a), those nodes on the single row are shown in Fig. 3(b). Let us denote the net list $\mathcal{L}_s$ defined on these nodes similar to the double row case by the sequence $S = M_L \_ U \_ M_R$, i.e., $\mathcal{L}_s = [S]$.

The interval graphical representation of the net list $\mathcal{L}_s = [M_L \_ U \_ M_R]$ on the single row is defined in the same way as [4]. For example, given a net list shown in Fig. 3(b), consider an ordering $f : \mathcal{L}_s \to \{1,2,\cdots,n\}$ such that $f(3)=1$, $f(6)=2$, $f(1)=3$, $f(4)=4$, $f(2)=5$, and $f(5)=6$, then the interval graphical representation associated with f is depicted as in Fig. 3(c), where each horizontal line corresponding to a net is arranged according to the ordering f from top down. Nodes which pertain to a net are marked as shown by $u_i$ or $m_i$. Let us define the reference line[4] in an interval graphical representation: Introduce fictitious nodes (0) and (∞) on the top-left and the top-right of the representation, respectively. Connect node (0) to the node $u_1$ which belongs to the first net in sequence U with a line seqment. Then connect the nodes $u_2$, $u_3,\cdots,u_r$, and (∞) in succession from left to right serially with line segments as shown in Fig. 3(c). This continuous line from (0) to (∞) is called the reference line.

Now, let us stretch out the reference line and map it into the upper row. In the mean time, place the nodes $m_i$ $(1 \le i \le c)$ on the lower row as shown in Fig. 3(d). In such a topological transformation, each net represented by an interval line is transformed into a path composed of horizontal and vertical line segments. This gives a realization of the problem Half-DRP.

In order to ensure that, in the realization of Half-DRP Problem, conductor lines do not go beyond the lower row, we require that the following two conditions be satisfied:

$C_L$: For nets $N_i^m$ and $N_j^m$ in $M_L$ with $i<j$, there holds $f(N_i^m) < f(N_j^m)$.

$C_R$: For nets $N_i^m$ and $N_j^m$ in $M_R$ with $i<j$, there holds $f(N_i^m) > f(N_j^m)$.

It is clear that for each interval graphical representation associated with an ordering satisfying conditions $C_L$ and $C_R$, there corresponds a unique realization of Half-DRP Problem. Furthermore, the crossing number of such a realization is simply the number of intersections between the reference line and the interval lines, not counting the intersections at nodes. Let $X^f(M_L, U, M_R)$ be the number of such intersections between the reference line and the interval lines in the interval graphical representation of the net list $\mathcal{L}_s = [M_L \_ U \_ M_R]$ associated with ordering f. Then the crossing number X in the realization obtained by the above topological transformation is equal to $X^f(M_L,U,M_R)$.

Therefore, problem Half-DRP is formulated as follows:

Half-DRP Problem: Given a net list $\mathcal{L}_u = [U,M]$, find subsequences $M_L$ and $M_R$ and an ordering $f : \mathcal{L}_u \to \{1,2,\cdots,n\}$ such that

i) $M_L - M_R = M$,

ii) $(=C_L)$ for nets $N_i^m$ and $N_j^m$ $(i<j)$ in $M_L$, there holds $f(N_i^m) < f(N_j^m)$,

iii) $(=C_R)$ for nets $N_i^m$ and $N_j^m$ $(i<j)$ in $M_R$, there holds $f(N_i^m) > f(N_j^m)$, and

iv) $X^f(M_L,U,M_R)$ is minimum.

Without loss of generality, we may assume that a net list $\mathcal{L}_u = [U,M]$ of the problem Half-DRP does not have any net consisting of an isolated node or any net containing two consecutive nodes on a row.

Also, inherent in the approach of using the interval graphical representation, several patterns of conductor lines are excluded from considerations. These are shown in Fig. 4, where (a) indicates a tree-shaped connection and (b) a forward-backward zigzagging around a row, and (c) depicts a combination of conductor lines which cannot be generated by the method of the interval graphical representation.

## 3. Computational Complexity Analysis

In this section, let us analyze the computational complexities of problems Half-DRP and MSP. For this purpose, we introduce the following single-row routing problem stated as a decision problem.

Single-Row Problem (SR): Given a net list $\mathcal{L}_s = [S]$ and a positive integer K', does there exist an ordering $f : \mathcal{L}_s \to \{1,2,\cdots,n\}$ with $X^f(\lambda,S,\lambda) \le K'$, where $\lambda$ represents a null sequence?

For this problem, we have the following theorem.

Theorem 1: Problem SR is NP-complete[3].

Proof. See Appendix 1.

Now, given a net list $\mathcal{L}_s = [S]$ and a positive integer K as input for problem SR, consider net list $\mathcal{L}_u = [S,M=\lambda]$ and positive integer $K' = K$ as input for problem Half-DRP. (In this section, we consider decision-problem-version of Half-DRP.) We can easily see that problem SR is a restricted case of Half-DRP, where $M = \lambda$. Thus, we have the following corollary.

Corollary 1: Problem Half-DRP is NP-complete.

Let us next consider the Modified Half-DRP problem stated as a decision problem, in order to see the intractability of problem MSP.

Modified Half-DRP Problem (MHD): Given a net list $\mathcal{L}_u = [U,M]$ with $|\mathcal{L}_u| = |M|$ and a positive integer K, is there an ordering $f : \mathcal{L}_u \to \{1,2,\cdots,n\}$ such that $X^f(\lambda,U,M) \le K$?

This problem is different from Half-DRP in the following sense:

i) sequence M must contain all nets,

ii) there is no choice about the partition of M into $M_L$ and $M_R$, i.e., $M_L = \lambda$ and $M_R = M$, and

iii) the restriction $C_R$ imposed on the ordering f in Half-DRP does not exist.

Fig. 5 show such an interval graphical representation for a given net list

$\mathcal{L}_u = [U,M]$, and in the figure, $X^f(\lambda,U,M) = 10$.

By using the same approach to the proof of Theorem 1, we can verify the following theorem.

Theorem 2: Problem MHD is NP-complete.

Now, let us consider another problem called Restricted MSP Problem.

Restricted MSP Problem: When Half-DRP is solved by means of the interval graphical representation, we assume that subsequences $M_L$ and $M_R$ of M must be $\lambda$ and M, respectively. In this case, given a net list $\mathcal{L} = [U,W]$ and a positive integer K', does there exist a sequence M on the middle row such that $\max[X_u, X_w] \le K'$?

We can easily see that this problem is a restricted problem of decision-problem-version of MSP, in the sense that Half-DRP Problem is solved by means of the interval graphical representation and the conductor lines are not allowed to go to the left beyond the leftmost nodes $u_1$ and $w_1$.

Given a net list $\mathcal{L}_u = [U,M]$ and a positive integer K as input for problem MHD, consider net list $\mathcal{L} = [U,U]$ and positive integer $K' = K$ as input for problem Restricted MSP. Then, as readily seen from Fig. 6, the problem of finding an ordering f satisfying the condition in MHD is equivalent to that of finding a sequence M of Restricted MSP. Therefore, we have the following corollary.

Corollary 2: Problem Restricted MSP is NP-complete.

From the above observation, we can conclude that we better try to find a heuristic algorithm for the double-row planar routing problem. In a certain special case, however, we can devise an algorithm for finding an optimal solution to problem Half-DRP. We shall describe it in the next section.

## 4. Merging Algorithm

Let us consider the special case of problem Half-DRP where $|\mathcal{L}_u| = |M|$, i.e., sequence M contains all nets. We adopt the following approach to tackle the problem Half-DRP.

<Algorithm for Half-DRP>

Input : A net list $\mathcal{L}_u = [U,M]$ with $|\mathcal{L}_u| = |M|$.

Output: Subsequences $M_L$ and $M_R$ of M and an ordering $f : \mathcal{L}_u \to \{1,2,\cdots,n\}$ such that f satisfies conditions $C_L$ and $C_R$ and $X^f(M_L,U,M_R)$ is minimum.

Step I: Let $M \triangleq (N_1^m, N_2^m, \cdots, N_n^m)$, then set $M_L \leftarrow (N_1^m)$ and $M_R \leftarrow (N_2^m, N_3^m, \cdots, N_n^m)$. Put $i \leftarrow 1$ and $X \leftarrow \infty$.

Step II: Solve the following problem called Simple Half-DRP Problem.

Simple Half-DRP Problem: Given a net list $\mathcal{L}_s = [M_L \_ U \_ M_R]$, find an ordering $f : \mathcal{L}_s \to \{1,2,\cdots,n\}$ such that f satisfies $C_L$ and $C_R$, and $X^f(M_L,U,M_R)$ is minimum.

If for the solution f to Simple Half-DRP, there holds $X^f(M_L,U,M_R) < X$, then store $M_L$, $M_R$, and f as the current solution to problem Half-DRP, and set $X \leftarrow X^f(M_L,U,M_R)$.

Step III: Set $i \leftarrow i+1$. If $i \le n-1$, then return to Step II by setting $M_L \leftarrow M_L \_ (N_i^m)$ and $M_R \leftarrow (N_{i+1}^m, N_{i+2}^m, \cdots, N_n^m)$; else terminate.

If Simple Half-DRP is solved in polynomial time, then Half-DRP is also solved. So, let us consider problem Simple Half-DRP, in the following.

Since $|\mathcal{L}_s| = |M|$, we can see from conditions $C_L$ and $C_R$ that all possible orderings satisfying $C_L$ and $C_R$ correspond one-to-one to the sequences obtained by merging $M_L$ with $\overline{M}_R$, where $\overline{M}_R \triangleq (N_n^m, N_{n-1}^m, \cdots, N_{\ell+1}^m)$ for $M_R = (N_{\ell+1}^m, N_{\ell+2}^m, \cdots, N_n^m)$. Namely, for a merged sequence $Q \triangleq (q_1, q_2, \cdots, q_n)$ of $M_L$ and $\overline{M}_R$, consider ordering f such that $f(q_i) = i$ $(i = 1,2,\cdots,m)$, then this f automatically satisfies $C_L$ and $C_R$. Conversely, for an ordering f satisfies $C_L$ and $C_R$, consider the sequence $Q \triangleq (q_1,q_2,\cdots,q_n)$ such that $q_i = f^{-1}(i)$, then sequence Q is a sequence obtained by merging $M_L$ with $\overline{M}_R$.

From this observation, we can estimate the number of all possible orderings as follows:

$$\# \text{ of possible orderings} = \frac{|M_L + M_R|!}{|M_L|!\ |M_R|!} \sim \frac{n!}{(n/2)!\ (n/2)!}$$

$$\sim \frac{\sqrt{2\pi n}\ n^n\ e^{-n}}{[\ \sqrt{2\pi n/2}(n/2)^{n/2} e^{-n/2}\ ]^2} = \frac{\sqrt{2}\ 2^n}{\sqrt{\pi n}}\ .$$

Thus, an exhaustive search algorithm cannot lead to a polynomial time algorithm for Simple Half-DRP problem.

Now, let us define a labeled <u>grid digraph</u> $G = [V,E]$, in which we will see that all merged sequences of $M_L$ and $\overline{M}_R$ correspond one-to-one to the <u>directed paths</u> from <u>source</u> to <u>sink</u>.

Let $M_L \triangleq (N_1,N_2,\cdots,N_\ell)$ and $\overline{M}_R \triangleq (N_n,N_{n-1},\cdots,N_{\ell+1})$*. Then, each vertex corresponds to a pair of integers, and vertex set V is defined as

$$V \triangleq \{<i,j> \mid 1 \le i \le \ell+1,\ 1 \le j \le n-\ell+1\ \}.$$

In particular, vertices $<1,1>$ and $<\ell+1,n-\ell+1>$ are designated as source and sink, respectively. Edge set E consists of two disjoint sets $E_L$ and $E_R$ defined as

$$E_L \triangleq \{\ (<i,j>,<i+1,j>) \mid 1 \le i \le \ell,\ 1 \le j \le n-\ell+1\ \}, \text{ and}$$

$$E_R \triangleq \{\ (<i,j>,<i,j+1>) \mid 1 \le i \le \ell+1,\ 1 \le j \le n-\ell\ \}.$$

Each edge $(<i,j>,<i+1,j>)$ in $E_L$ has label net $N_i$, and each edge $(<i,j>,<i,j+1>)$ in $E_R$ has label net $N_{n-j+1}$. Fig. 7 shows the grid digraph G for $M_L = (N_1,N_2,N_3)$ and $\overline{M}_R = (N_8,N_7,N_6,N_5,N_4)$.

For each directed path from source to sink, we can create a sequence of labels according to the edges passed by the directed path, which is a sequence

* We have dropped the superscript m, for convenience, since there is no confusion.

obtained by merging $M_L$ and $\overline{M}_R$. And, we can easily verify that each directed path corresponds one-to-one to a merged sequence of $M_L$ and $\overline{M}_R$. Therefore, the ordering f which satisfies both $C_L$ and $C_R$ corresponds one-to-one to the directed path from source to sink in the grid digraph G.

For example, the directed path shown by the bold-line in Fig. 7 corresponds to merged sequence $(N_1,N_8,N_7,N_6,N_2,N_5,N_3,N_4)$, and hence corresponds to ordering $f: f(N_1)=1$, $f(N_2)=5$, $f(N_3)=7$, $f(N_4)=8$, $f(N_5)=6$, $f(N_6)=4$, $f(N_7)=3$, and $f(N_8)=2$.

Therefore, if we can assign an appropriate weight to each edge so that the total sum of the weights of all the edges on each directed path is exactly equal to the crossing number $X^f(M_L,U,M_R)$ in the interval graphical representation associated with the ordering f corresponding to the directed path, then we can solve problem Simple Half-DRP by using a shortest path algorithm on the grid digraph. Note here that if weights assigned to edges satisfy the following two conditions, then the weights are appropriate ones.

(i) The weight of edge $(\langle i,j\rangle,\langle i+1,j\rangle)$ with label $N_i$ $(1 \le i \le \ell)$ is equal to the number of intersections between the interval line of $N_i$ and the reference line, which are caused if $N_i$ is ordered between $N_{n-j}$ and $N_{n-j+1}$. That is, $N_i$, $N_{n-j}$, and $N_{n-j+1}$ are arranged in the order as $(\cdots,N_{n-j},\cdots,N_i,\cdots,N_{n-j+1},\cdots)$ in the merged sequence.

(ii) The weight of edge $(\langle i,j\rangle,\langle i,j+1\rangle)$ with label $N_{n-j+1}$ $(1 \le j \le n-\ell)$ is equal to the number of intersections between the reference line and the interval line of $N_{n-j+1}$, which are caused if $N_{n-j+1}$ is ordered between $N_{i-1}$ and $N_i$. Thus, let us consider how to assign such weights satisfying these conditions.

An interval between two consecutive nodes is called a <u>unit interval</u>, and the two nodes are designated as <u>endnodes</u> of the unit interval. A net containing an endnode of the unit interval is an <u>end-net</u> of the unit interval. If the net $N_i^u$ containing the ith node $u_i$ belongs to sequence $M_L$ or $M_R$, then

$u_i$ is called an L-node or an R-node, respectively. A unit interval is called an L-L interval, an L-R interval, or an R-R interval, if both its endnodes are L-nodes, one is an L-node and another an R-node, or both are R-nodes, respectively, where the fictitious nodes 0 and $\infty$ are both regarded as L-nodes. The portion of the reference line for a unit interval H is denoted by RL(H).

Consider a unit interval H with end-nets $N_a$ and $N_b$. As can be readily seen, only the interval lines of nets which cover interval H may or may not intersect RL(H), depending on the relative order with respect to $N_a$ and $N_b$. Let us consider it in the following.

I. Let $N_i$ be a net in $M_L$ which covers H. (See Fig. 8.)

i) H is an L-L interval. We can assume $a < b$ without loss of generality. In this case, RL(H) and $N_i$ intersect each other when $a < i < b$. Thus, edges with label $N_i$ $(a < i < b)$ must have weights corresponds to this intersection.

ii) L-R interval. Let $N_b \triangleq N_{n-j+1}$.

Case 1 $(a < i)$. In this case, RL(H) and $N_i$ intersects, if and only if the ordering f satisfies $f(N_i) < f(N_b)$. Therefore, only edges $(\langle i,h\rangle,\langle i+1,h\rangle)$ with $h \leq j$ must have weights corresponding to this intersection.

Case 2 $(a > i)$. In this case, RL(H) and $N_i$ intersect, if and only if the ordering f satisfies $f(N_i) > f(N_b)$. Therefore, only edges $(\langle i,h\rangle,\langle i+1,h\rangle)$ with $h > j$ must have weights corresponding to this intersection.

iii) R-R interval. Let $N_a \triangleq N_{n-k+1}$ and $N_b \triangleq N_{n-j+1}$, and assume $k < j$ without loss of generality. In this case, RL(H) and $N_i$ intersect, if and only if the ordering f satisfies $f(N_a) < f(N_i) < f(N_b)$. Therefore, only edges $(\langle i,h\rangle,\langle i+1,h\rangle)$ with $k < h \leq j$ must have weights corresponding to this intersection.

II. A similar analysis can be given for a net in $\overline{M}_R$.

Based on the above discussion, we can devise an algorithm of finding the desired weights for edges, by processing unit intervals successively from the left to the right. This is given in Appendix 2.

If we apply the Weight Assignment Algorithm in Appendix 2 to a net list $\mathcal{L}_s = [M_L \_ U \_ M_R]$ shown in Fig. 9(a), then we have the weights for all edges shown in Fig. 9(b). In the figure, the number in a bracket and the sequence of alphabets beside an edge show the weight of the edge and the unit intervals at which the weight of the edge is increased by one, respectively. The interval graphical representation shown in Fig. 9(a) is associated with the ordering corresponding to the directed path from source to sink through the lower left corner of the grid digraph. Hence, as shown by the weight and sequence (d,k) beside edge label $N_6$ in Fig. 9(b), net $N_6$ has 2 intersections at unit intervals d and k in the interval graphical representation of Fig. 9(a).

Let the <u>length</u> of a directed path in the grid digraph be the sum of the weights of all edges on the directed path, then the following lemma can be readily verified from the above discussion.

<u>Lemma 1</u>: The length of a directed path in the grid digraph for a net list $\mathcal{L}_s = [M_L \_ U \_ M_R]$ is equal to the crossing number $X^f(M_L, U, M_R)$ in the interval graphical representation associated with the ordering f corresponding to the directed path.

Thus, an optimum ordering f for Simple Half-DRP is obtained by a shortest path algorithm on the grid digraph. In the following, we describe an algorithm for Simple Half-DRP Problem.

<<u>Merging Algorithm</u>>

<u>Input</u> : A net list $\mathcal{L}_s = [M_L \_ U \_ M_R]$.

<u>Output</u>: An ordering $f : \mathcal{L}_s \rightarrow \{1,2,\cdots,n\}$ such that f satisfies $C_L$ and

$C_R$ and $X^f(M_L,U,M_R)$ is minimum.

Step 1. Create grid digraph $G=[V,E]$ for $M_L \triangleq (N_1^m,N_2^m,\cdots,N_\ell^m)$ and $\overline{M}_R \triangleq (N_n^m,N_{n-1}^m,\cdots,N_{\ell+1}^m)$.

Step 2. <Weight Assignment Algorithm>.

Step 3. Compute shortest distance from source to each vertex <i,j>. Noting that in the grid digraph any directed path from source to vertex <i,j> passes though vertex either <i-1,j> or <i,j-1>, we can implement this process in the processing time propotional to the number of vertices of the grid digraph.

Step 4. Find a shortest path from source <1,1> to sink <ℓ+1,n-ℓ+1> by tracing back from sink to source.

By substituting Merging Algorithm for Step II in <Algorithm for Half-DRP>, we can complete the algorithm, for which we have the following theorem.

Theorem 3: <Algorithm for Half-DRP> can find an optimum solution to problem Half-DRP in the processing time of order $0(n^3r)$ and in the memory space of order $0(n^2+r)$, where n and r are the numbers of nets and nodes in the upper row, respectively.

Proof. We can easily see from Lemma 1 that the algorithm can find an optimum solution. Noting that $0(n^2)$ space is required for the grid digraph and that $0(n+r)$ space is sufficient for a given nêt list and for other sets and sequences, we can also verify that the algorithm is implemented in $0(n^2+r)$ space.

Let us consider the processing time. In Merging Algorithm, Steps 1 and 3 can be executed in $0(n^2)$ time and Step 4 in $0(n)$ time. As is shown in Appendix 2, the total time required for Weight Assignment Algorithm is $0(n^2r)$, and therefore the total time of Merging Algorithm is $0(n^2r)$. Hence, the theorem has been proven, since the loop of Steps II-IV in Algorithm for Half-DRP requires n-1 times iterations. □

## 5. Heuristic Algorithm

In this section, we propose a heuristic algorithm for Problem DRP, in which Half-DRP is solved with the extended use of Merging Algorithm and MSP is solved by an iterative improvement. Iterative improvement is carried out by a displacement of a single net at a time. The proposed heuristic algorithm is shown briefly in the following.

In a given sequence $M \triangleq (N_1, N_2, \cdots, N_i, \cdots, N_c)$ on the middle row, if position $P_j$ is allocated for the ith net $N_i$ in M, then $P_j$ is

i) immediately in front of $N_i$, for $1 \le j < i$,

ii) at the same place as $N_i$, for $j = i$, or

iii) immediately back of $N_i$, for $i < j \le c$.

A sequence M' is said to be obtained by displacing $N_i$ in M to position $P_j$, if $M' = (N_1, \cdots, N_{j-1}, N_i, N_j, \cdots, N_{i-1}, N_{i+1}, \cdots, N_c)$ for $1 \le j < i$, $M' = M$ for $j = i$, or $M' = (N_1, \cdots, N_{i-1}, N_{i+1}, \cdots, N_j, N_i, N_{j+1}, \cdots, N_c)$ for $i < j \le c$.

<A Heuristic Algorithm for DRP>

Input : A net list $\mathcal{L} = [U,W]$.

Output: A realization of $\mathcal{L}$ with a minimal crossing number $X \triangleq \max[X_u, X_w]$.

Phase I (Initial Step):

Step I: Generate an initial sequence $M \triangleq (N_1, N_2, \cdots, N_c)$ of the nets containing nodes in both rows.

Step II: Solve problems Half-DRP with net list $\mathcal{L}_u = [U,M]$ and Half-DRP with net list $\mathcal{L}_w = [W,M]$.

Phase II (Iterative Improvement Step):

Step 0: Put all nets in M into queue arbitrarily.

Step I: Repeat the following process 1° - 3°, until no more improvement can be acheived.

1°; Delete a net $N_i$ from the front of queue and put it to the back of queue.

2°: Compute the expected difference of the crossing number*
$EDC_u(p_j)$ in the upper row and $EDC_w(p_j)$ in the lower row, for each position $p_j$ for $N_i$ in M, exclusive of $p_i$. Let $X_u$ and $X_w$ be the current crossing numbers for $\mathcal{L}_u = [U,M]$ and $\mathcal{L}_w = [W,M]$, respectively. And let $X \leftarrow \max[\ X_u,\ X_w\ ]$, $\underline{X} \leftarrow \min[\ X_u,\ X_w\ ]$, $Y(p_j) \leftarrow \max[\ X_u + EDC_u(p_j)\ ,\ X_w + EDC_w(p_j)\ ]$, and $\underline{Y}(p_j) \leftarrow \min[\ X_u + EDC_u(p_j),\ X_w + EDC_w(p_j)\ ]$. Then, set

$$P \leftarrow \{\ p_j \mid Y(p_j) < X,\ \text{or}\ Y(p_j) = X\ \text{and}\ \underline{Y}(p_j) < \underline{X}\ \}.$$

From the definitions of $EDC_u$, $EDC_w$, and P, we can see that, for every sequence M' obtained by displacing $N_i$ in M to $p_j \in P$, we can get a better solution than the current one.

3°: By conducting the following process (i) - (ii) for each $p_j \in P$, find the best position $p_b$ for $N_i$, and displace $N_i$ in M to $p_b$.

(i) $M' \leftarrow M$ and displace $N_i$ in M' to $p_j$.

(ii) Solve problems Half-DRP with net list $\mathcal{L}_u = [U,M']$ and Half-DRP with net list $\mathcal{L}_w = [W,M']$.

Phase III (Postprocessing Step):

Step I: Realize $\mathcal{L}_u = [U,M]$ and $\mathcal{L}_w = [W,M]$ by the topological transformation from the interval graphical representations specified by the solutions obtained in Phase II.

Step II: Turn the realization of net list $\mathcal{L}_w = [W,M]$ upside down, then combine two realizations of $\mathcal{L}_u$ and $\mathcal{L}_w$ into one.

In Step I of Phase I or Step I-3°-(ii) of Phase II, if the sequence M does not contain all the nets in $\mathcal{L}_u$, then we cannot use <Algorithm for

* $EDC_u(p_j)$ shows the expected increment from the current crossing number $X_u$ caused by displacing $N_i$ in M into position $p_j$. So, if $EDC_u(p_j) < 0$, then the crossing number decreases at least by $|EDC_u(p_j)|$. The details will be discribed in Appendix 3.

Half-DRP> described in the previous section directly. In such case, we adopt the following procedure.

1°: Find an ordering $f^* : \mathcal{L}_u \to \{1,2,\cdots,n\}$ such that $X^{f^*}(M,U,\lambda)$ is minimal. Then, let $M_u^* \triangleq (N_1^*, N_2^*, \cdots, N_n^*)$ such that $N_i^* = f^{*-1}(i)$ for $1 \le i \le n$.

2°: Solve problem Half-DRP with $\mathcal{L}_u = [U,M]$ by using <Merging Algorithm>, where conditions $C_L$ and $C_R$ are specified on $M_u^*$. Then, let $X^f(M_L,U,M_R)$ be the obtained solution.

3°: Improve the ordering f obtained in 2° by displacing the nets not contained in M so that $X^{f'}(M_L,U,M_R)$ for the improved ordering f' is minimal.

In this procedure, 1° and 3° is conducted by a displacement of a net not contained in M at a time, with the use of similar information to the expected difference $EDC_u$. The details are ommitted in this paper.

In the following, we shall consider the problem of finding an initial sequence M.

Let $u_j$ $(1 \le j \le r)$ and $w_j$ $(1 \le j \le r)$ be nodes on the upper and the lower rows, respectively. For a net $N_i$, the smallest and the largest subscript-numbers of nodes in $N_i$ on the upper row are denoted by $uln(N_i)$ and $urn(N_i)$, and those on the lower row by $wln(N_i)$ and $wrn(N_i)$, respectively. In the upper row, if the portion of the interval between $u_{uln(N_i)}$ and $u_{urn(N_i)}$ contained in the left-half of interval $[u_1,u_r]$ is longer than that contained in the right-half of $[u_1,u_r]$, then net $N_i$ is said to be an <u>upper-left net</u>. Namely, $N_i$ is a upper-left net, if $\lfloor r/2 \rfloor + 1 - uln(N_i) \ge urn(N_i) - \lfloor r/2 \rfloor$, where $\lfloor x \rfloor$ denotes the largest integer not greater than x. Otherwise, $N_i$ is called an <u>upper-right net</u>. Similarly, we define a <u>lower-left net</u> and a <u>lower-right net</u> on the lower row.

If a net has nodes on the both rows, then the net is said to be a <u>common net</u>. A set of common nets can be partitioned into four subsets $\mathcal{S}_1$, $\mathcal{S}_2$,

$\mathcal{S}_3$, and $\mathcal{S}_4$ as follows:

$\mathcal{S}_1 \triangleq \{N_i \mid N_i$ is a lower-left, upper-right common net$\}$.

$\mathcal{S}_2 \triangleq \{N_i \mid N_i$ is a lower-left, upper-left common net$\}$.

$\mathcal{S}_3 \triangleq \{N_i \mid N_i$ is a lower-right, upper-left common net$\}$.

$\mathcal{S}_4 \triangleq \{N_i \mid N_i$ is a lower-right, upper-right common net$\}$.

In an initial sequence M, the common nets are arranged in the order of $\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{S}_3$, and $\mathcal{S}_4$ from the left to the right, as shown in the following algorithm.

<<u>An Initial Sequence Algorithm</u>>

<u>Input</u> : A net list $\mathcal{L} = [U,W]$.

<u>Output</u>: An initial sequence M of common nets.

<u>Step 1</u>: Compute subsets $\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{S}_3$, and $\mathcal{S}_4$ of common nets.

<u>Step 2</u>: Let U1 be a sequence obtained from the left half subsequence $U_L \triangleq (N^u_1, N^u_2, \cdots, N^u_i, \cdots, N^u_{\lfloor r/2 \rfloor})$ of U by deleting $N^u_i$ such that $i \neq \mathrm{uln}(N^u_i)$, without changing the order of nets in $U_L$. And let U2 be a sequence obtained from the right half subsequence $U_R \triangleq (N^u_{\lfloor r/2 \rfloor+1}, N^u_{\lfloor r/2 \rfloor+2}, \cdots, N^u_i, \cdots, N^u_r)$ of U by deleting $N^u_i$ such that $i \neq \mathrm{urn}(N^u_i)$. Namely, U1 (U2) shows an order of nets according to which we meet each net for the first time, if we explore sequence U form the left (the right) to the right (the left). Let us define W1 and W2 similar to U1 and U2, respectively. Then set $S1 \leftarrow \overline{W1} \cap \mathcal{S}_1$, $S2 \leftarrow \overline{U1} \cap \mathcal{S}_2$, $S3 \leftarrow U1 \cap \mathcal{S}_3$, and $S4 \leftarrow \overline{W2} \cap \mathcal{S}_4$, where for a set $\mathcal{S}$ and a sequence A, $A \cap \mathcal{S}$ represents a subsequence of A by deleting all the elements not contained in $\mathcal{S}$ from A without changing the order of nets in A, and $\overline{A}$ denotes a sequence with the reverce order of A.

<u>Step 3</u>: Set $M \leftarrow S1\_S2\_S3\_S4$.

For example, if this algorithm is applied to net list $\mathcal{L} = [U,W]$ such that

$$U = U_L \_ U_R = (1,2,1,3,2,8,4) \_ (5,3,6,2,6,7,8) \text{ and}$$

$$W = W_L \_ W_R = (6,2,6,1,8,3,1) \_ (8,4,5,3,7,4,5),$$

then we have

{ upper-left common nets } = { 1,2,3,4 },

{ upper-right common nets } = { 5,6,7,8 },

{ lower-left common nets } = { 1,2,6,8 }, and

{ lower-right common nets } = { 3,4,5,7 },

and hence $\mathcal{S}_1 = \{ 6,8 \}$, $\mathcal{S}_2 = \{ 1,2 \}$, $\mathcal{S}_3 = \{ 3,4 \}$, and $\mathcal{S}_4 = \{ 5,7 \}$.

Moreover, we have

$$U1 = (1,2,3,8,4), \qquad \overline{U1} = (4,8,3,2,1),$$

$$W1 = (6,2,1,8,3), \qquad \overline{W1} = (3,8,1,2,6),$$

$$W2 = (5,4,7,3,8), \qquad \overline{W2} = (8,3,7,4,5).$$

Thus, sequence M is given as $(8,6) \_ (2,1) \_ (3,4) \_ (7,5)$.

Another example is shown in Fig. 10, which illustrates the frame of the initial sequence for the case where U and W are sequences of distinct nets.

As discussed in [1], let us consider the upper bound for the breadth of a realization obtained by Phase I in the proposed algorithm. We assume that a given net list $\mathcal{L} = [U,W]$ satisfies

i) U and W consist of distinct nets, and

ii) $|\mathcal{L}| = |M|$, i.e., $n = r$.

Then, a realization of $\mathcal{L}$ is a spaced-spaced layout defined in [1], and the breadth of a realization corresponds to the <u>length</u>[1] of a spaced-spaced layout. Let us show that Phase I gives a better upper bound for the breadth than [1] in what follows.

Given an interval graphical representation associated with ordering f, let us define an <u>unconnected node</u> and <u>block</u> similar to an <u>unconnected point</u> and <u>block</u> defined in [1]. For a net $N_a$ in the representation, a node contained in a net $N_b$ with $f(N_a) < f(N_b)$ is said to be an unconnected node for

$N_a$, and a maximal set of consecutive unconnected nodes is called a block for $N_a$. Moreover, let $\alpha_i$ be the number of blocks for net $f^{-1}(i)$ and let $\beta_i$ be the number of intersections between the reference line and the interval line of net $f^{-1}(i)$. Then, as seen from Fig. 11(a), we have the following lemma.

<u>Lemma 2</u>[1]: For a net $f^{-1}(i)$, there hold

$$\beta_i \le 2\alpha_i - 1 \text{ and}$$

$$\alpha_i \le \min[\ r - i + 1,\ i\ ],$$

where r is the number of nodes on the reference line.

Consider the interval graphical representations for $\mathcal{L}_u = [U,M]$ and $\mathcal{L}_w = [W,M]$ shown in Fig. 11(b). Let $X_u^*$ and $X_w^*$ be the crossing numbers of these representations, then we can easily verify from Merging Algorithm that $X_u \le X_u^*$ and $X_w \le X_w^*$, where $X_u$ and $X_w$ are the crossing numbers on the upper and the lower rows obtained by Step II of Phase I, respectively. By using $X_u^*$ and $X_w^*$ and Lemma 2, we can show the following theorem.

<u>Theorem 4</u>: Let $\mathcal{L} = [U,W]$ be a given net list with $n = r$ and let U and W be sequences of distinct nets. Then, for the crossing number $X_1 \triangleq \max[\ X_u,\ X_w\ ]$ obtained by Phase I (Initial Step), we have

$$X_1 \le n^2/8 + O(n),$$

and hence

$$\text{breadth} \triangleq X_1 + (n-1) \le n^2/8 + O(n).$$

<u>Proof</u>: Let us count $X_u^*$ and $X_w^*$.

$$\begin{aligned}
X_u^* = \sum_{i=1}^{n} \beta_i &\le 2\Big( \sum_{i=1}^{\lceil n/2 \rceil} \alpha_i + \sum_{i=\lceil n/2 \rceil + 1}^{n} \alpha_i \Big) - n \\
&\le 2\Big( \sum_{i=1}^{\lceil n/2 \rceil} \min[\ \lceil n/2 \rceil - i + 1,\ i\ ] + |\mathcal{S}_2| \Big) - n \\
&\le 4\Big( 1 + 2 + \cdots + \frac{\lceil n/2 \rceil + 1}{2} \Big) + 2|\mathcal{S}_2| - n \\
&\le \frac{(n+5)(n+9)}{8} + 2|\mathcal{S}_2| - n \\
&< \frac{(n+5)(n+9)}{8},
\end{aligned}$$

since $|\mathcal{S}_2| < \lfloor n/2 \rfloor$, where $\lceil x \rceil$ denotes the smallest integer not less than x.

$$X^*_w = \sum_{i=1}^{n} \beta_i \le 2\left( \sum_{i=1}^{\lceil n/2 \rceil} \alpha_i + \sum_{i=\lceil n/2 \rceil+1}^{n} \alpha_i \right) - n$$

$$\le 2\Big( |\mathcal{S}_4| + \sum_{i=|\mathcal{S}_4|+1}^{\lceil n/2 \rceil} \min[\, \lceil n/2 \rceil - i + 1,\ i\,]$$

$$+ |\mathcal{S}_1| + \sum_{i=|\mathcal{S}_1|+1}^{\lfloor n/2 \rfloor} \min[\, \lfloor n/2 \rfloor - i + 1,\ i\,] \Big) - n$$

$$\le 2\left( \sum_{i=|\mathcal{S}_4|+1}^{\lceil n/2 \rceil} \min[\, \lceil \tfrac{n}{2} \rceil - i + 1,\ i\,] + \sum_{i=|\mathcal{S}_1|+1}^{\lceil n/2 \rceil} \min[\, \lceil \tfrac{n}{2} \rceil - i + 1,\ i\,] \right)$$

$$+ 2\left( |\mathcal{S}_1| + |\mathcal{S}_4| \right) - n.$$

Noting that $|\mathcal{S}_1| + |\mathcal{S}_4| = \lceil n/2 \rceil$, there holds

$$X^*_w \le 2\left( \sum_{i=1}^{\lceil n/2 \rceil} \min[\, \lceil n/2 \rceil - i + 1,\ i\,] \right) + 2\lceil n/2 \rceil - n$$

$$\le \frac{(n+5)(n+9)}{8} + 1.$$

Thus, we have

$$X_1 \triangleq \max[\, X_u,\ X_w] \le \max[\, X^*_u,\ X^*_w\,] \le \frac{(n+5)(n+9)}{8} + 1$$

$$= n^2/8 + O(n). \qquad \square$$

In [1], no algorithm is proposed for a spaced-spaced layout. Therefore, their upper bound for a spaced-spaced layout is $n^2/4 + O(n)$, which is the same for a packed-spaced layout. We can see from the above discussion that the upper bound for a spaced-spaced layout is almost half of that for a packed-spaced layout, which can be expected from the definitions of both layouts.

Finally, it is easy to see that this Initial Sequence Algorithm is implemented in $O(n+r)$ time and $O(n+r)$ space.

## 6. Conclusion

In this paper, we have formulated the double-row planar routing problem with the total breadth as a criterion for minimization. The problem represents a generalization of the spaced-spaced layout[1] and of the single-row single-layer routing problem[4]. We have analyzed the computational complexity of the problem, and found that the problem is intractable in the sense of NP-completeness. For a certain special case, however, we have devised a polynomial-time algorithm called Merging Algorithm, which is to be used as a subroutine in the final heuristic algorithm. The heuristic algorithm involves iterative improvement from an initial solution. An analysis on the upper bound for the breadth of an initial solution is given. This leads to a slightly better results than that given in Ref. [1].

The crucial concept of our approach is to break up the complex problem into manageable subproblems. The main idea here is to introduce a middle row of nodes and consider two half layout problems each with specific constraints. Principally, for the middle row, nets are not allowed to go beyond the row. Also, in some of the derivations, we assume that the middle row contains all the nets in the upper and lower rows. This last assumption, however, can be removed with some modifications of the programs.

The proposed algorithm has been programmed in FORTRAN and run on VAX 11/780. Table 1 shows some experimental results. CASE I shows the results for the general case of double-row planar routing, while CASE II represents the case of Permutation Layout, i.e., for the net list $\mathcal{L} = [U,W]$, U and W are composed of distinct nets and $|\mathcal{L}| = |U| = |W|$. In each case, two examples are given, where the number n of nets and the number r of nodes are different, as shown in the table. For each example, 10 net list are randomly generated, and the average values taken over these 10 lists are shown in the table, where $\bar{X} \triangleq \max[\, X_u, X_w \,]$ and $\underline{X} \triangleq \min[\, X_u, X_w \,]$.

In Fig. 12, two output of the program are shown, namely: an example of CASE I in (a) and an example of CASE II in (b).

Table 1. Experimental Results.

| | | Phase I | | | Phase II | | |
|---|---|---|---|---|---|---|---|
| | | X | $\underline{X}$ | cpu time (sec) | X | $\underline{X}$ | cpu time (sec) |
| CASE I | n = 20, r = 30 | 40.8 | 31.5 | 0.3 | 26.8 | 25.6 | 24.4 |
| CASE I | n = 24, r = 36 | 56.7 | 43.1 | 0.5 | 39.1 | 37.3 | 46.8 |
| CASE II | n = 15, r = 15 | 13.4 | 10.8 | 0.1 | 8.0 | 7.4 | 10.0 |
| CASE II | n = 20, r = 20 | 21.6 | 16.5 | 0.3 | 13.1 | 12.6 | 38.1 |

REFERENCES

[1] M. Cutler and Y. Shiloach, "Permutation layout," Networks, vol. 8, pp. 253-278, 1978.

[2] M. R. Garey, D. S. Johnson, and L. Stockmeyer, "Some simplified NP-complete problems," Proc. 6th ACM Symp. Theory on Computing, pp. 47-63, 1974.

[3] R. M. Karp, "Reducibility among combinatorial problems," Complexity of Computer Computations, R. E. Miller and J. W. Thatcher (eds.), Plenum Press, N. Y., pp. 85-103, 1972.

[4] E. S. Kuh, T. Kashiwabara, and T. Fujisawa, "On optimum single-row routing," IEEE Trans. on Circuits and Systems, vol. CAS-26, pp. 361-368, 1979.

[5] I. Shirakawa, "Some comments on permutation layout," Networks, vol. 10, pp. 179-182, 1980.

[6] H. C. So, "Some theoretical results on the routing of multilayer printed wiring boards," Proc. IEEE Int. Symp. on Circuits and Systems, pp. 296-303, 1974.

[7] B. S. Ting, E. S. Kuh, and I. Shirakawa, "The multilayer routing problem: Algorithms and necessary and sufficient conditions for the single-row single-layer case," IEEE Trans. on Circuits and Systems, vol. CAS-23, pp. 768-778, 1976.

Appendix 1: (Proof of Theorem 1)

We shall show that the following Simple Linear Placement Problem, which is known as a NP-complete problem[2], is plynomially transformable[3] to problem SR.

Simple Linear Placement Problem (SLP): Given an undirected graph $G' = [V',E']$ and a positive integer $K'$, does there exist an ordering $f' : V' \to \{ 1,2,\cdots,|V'| \}$ such that

$$\sum_{(v_i,v_j)\in E'} |f'(v_i) - f'(v_j)| \leq K' \ ?$$

For a given $G' = [V',E']$ as input for problem SLP, let us consider the Euler graph $G'' = [V'',E'']$ generated as follows:

$$V'' = V' \triangleq \{ v_1,v_2,\cdots,v_n \}, \text{ and}$$

$$E'' = \{ (v_i,v_j)_1,(v_i,v_j)_2,\cdots,(v_i,v_j)_{2n^2} \mid (v_i,v_j) \in E' \}.$$

Without loss of generality, we assume that the graph $G'$ in SLP is connected. Hence, $G''$ is also connected. Thus, we have a closed walk in $G''$ passing through all edges. Let $A'$ be the vertex-sequence of such a closed walk starting and terminating at vertex $v_n$.

For $G' = [V',E']$ and $K'$, the corresponding input for problem SR is defined from $G''$ as follows: Each net $N_i$ corresponds to a vertex $v_i$, and a sequence $S$ of net list $\mathcal{L}_s = [S]$ is defined as $(N_1,N_2,\cdots,N_{n-1}) \_A\_ (N_{n-1},N_{n-2},\cdots,N_1)$, where $A$ is a sequence of nets generated from $A'$ by replacing a vertex by the corresponding net. Integer $K$ in SR is defined by

$$K = 2n^2(K' - e') + (n-1)(n-2),^*$$

where $e' \triangleq |E'|$. Note that these processes can be done in $O(n^2e')$ time.

Let $R_L$ and $R_R$ be the portions of the reference line which connect first

---

* We can see that by setting $K = 2n^2(K' - e') + (n-1)\,n$, this same approach can be applied to the proof of Theorem 2.

n nodes and last n nodes, respectively, and let $R_A$ be the remainder of the reference line, i.e., the reference line between first nth node and last nth node. By $x^f(R_A)$, we denote the number of intersections between $R_A$ and the interval lines of nets in the interval graphical representation associated with f. $x^f(R_L)$ and $x^f(R_R)$ are defined similarly. Then, we can easily see that for any ordering f of nets, there holds

$$x^f(R_L) \le \sum_{i=2}^{n-1}(i-1) = \sum_{i=1}^{n-2} i = \frac{(n-1)(n-2)}{2}, \text{ and}$$

$$x^f(R_R) \le \frac{(n-1)(n-2)}{2}.$$

Moreover, noting that the number of intersections between interval lines of nets and a unit interval on $R_A$ with end-nets $N_i$ and $N_j$ is equal to $|f(N_i) - f(N_j)| - 1$, then we see that there holds

$$x^f(R_A) = \sum_{(N_i,N_j)\in A} ( |f(N_i) - f(N_j)| - 1 ),$$

where the summation is taken over all the unit intervals on $R_A$. Since the number of unit intervals on $R_A$ is equal to the number of edges in the Euler graph G", we have

$$x^f(R_A) = \sum_{(N_i,N_j)\in A} |f(N_i) - f(N_j)| - 2n^2e'.$$

Suppose that there exists an ordering $f' : V' \to \{1,2,\cdots,n\}$ such that

$$\sum_{(v_i,v_j)\in E'} |f'(v_i) - f'(v_j)| \le K'.$$

Let us consider ordering $f : \mathcal{L}_s \to \{1,2,\cdots,n\}$ such that for a net $N_i$ corresponding to $v_i$, $f(N_i) = f'(v_i)$. Then, we have

$$\begin{aligned}
X^f(\lambda,S,\lambda) &= x^f(R_A) + x^f(R_L) + x^f(R_R) \\
&= \sum_{(N_i,N_j)\in A} |f(N_i) - f(N_j)| - 2n^2e' + x^f(R_L) + x^f(R_R) \\
&= 2n^2( \sum_{(v_i,v_j)\in E'} |f'(v_i) - f'(v_j)| ) - 2n^2e' + x^f(R_L) + x^f(R_R) \\
&\le 2n^2(K' - e') + x^f(R_L) + x^f(R_R) \\
&\le 2n^2(K' - e') + (n-1)(n-2) = K.
\end{aligned}$$

Therefore, this ordering f satisfies the condition in SR.

Conversely, suppose that there exists an ordering $f : \mathcal{L}_s \to \{1,2,\cdots,n\}$ such that $X^f(\lambda,S,\lambda) \le K$. Then, we have

$$x^f(R_A) + x^f(R_L) + x^f(R_R) \le 2n^2(K'-e') + (n-1)(n-2).$$

Consider ordering $f' : V' \to \{1,2,\cdots,n\}$ such that for a vertex $v_i$ corresponding to $N_i$, $f'(v_i) = f(N_i)$, then there holds

$$\begin{aligned} x^f(R_A) &= \sum_{(N_i,N_j)\in A} |f(N_i) - f(N_j)| - 2n^2e' \\ &= 2n^2\Big( \sum_{(v_i,v_j)\in E'} |f'(v_i) - f'(v_j)| \Big) - 2n^2e'. \end{aligned}$$

Thus, we have

$$\begin{aligned} &2n^2\Big( \sum_{(v_i,v_j)\in E'} |f'(v_i) - f'(v_j)| \Big) - 2n^2e' + x^f(R_L) + x^f(R_R) \\ &\le 2n^2(K'-e') + (n-1)(n-2), \end{aligned}$$

and

$$\begin{aligned} \sum_{(v_i,v_j)\in E'} |f'(v_i) - f'(v_j)| &\le K' + \frac{(n-1)(n-2) - \{ x^f(R_L) + x^f(R_R) \}}{2n^2} \\ &\le K' + \frac{(n-1)(n-2)}{2n^2} \\ &< K' + 1. \end{aligned}$$

Since the left side of the above inequality is an integer, there holds

$$\sum_{(v_i,v_j)\in E'} |f'(v_i) - f'(v_j)| \le K'. \qquad \square$$

Appendix 2: (Weight Assignment Algorithm)

<Weight Assignment Algorithm>

Input : A net list $\mathcal{L}_s = [M_L \_ U \_ M_R]$ and a grid digraph $G = [V,E]$ for $M_L \triangleq (N_1, N_2, \cdots, N_\ell)$ and $\overline{M}_R \triangleq (N_n, N_{n-1}, \cdots, N_{\ell+1})$.

Output: Weight WT(e) for each edge $e \in E$.

Step 1: Set WT(e) $\leftarrow$ 0 for every edge $e \in E$.

Step 2: Pick up the leftmost unit interval H, on which the following steps have not been conducted. If there is no such H, then terminate; else execute the followings on H.

Step 3: If H is an L-L interval, then go to Step 4. If H is an R-R interval, then go to Step 5. Otherwise, go to Step 6.

Step 4: Let $N_a$ and $N_b$ be end-nets of H with $a < b$.

1°. For each net $N_i$ in $M_L$ which passes through H and satsfies $a < i < b$, conduct (i).

(i). For every $e = (<i,h>,<i+1,h>)$ with $1 \le h \le n - \ell + 1$,

$$WT(e) \leftarrow WT(e) + 1.$$

2°. For each net $N_{n-j+1}$ in $\overline{M}_R$ which passes through H, conduct (ii).

(ii). For every $e = (<h,j>,<h,j+1>)$ with $a < h \le b$,

$$WT(e) \leftarrow WT(e) + 1.$$

3°. Then, return to Step 2.

Step 5: Let $N_{n-a+1}$ and $N_{n-b+1}$ be end-nets of H with $a < b$.

1°. For each net $N_i$ in $M_L$ which passes through H, conduct (iii).

(iii). For every $e = (<i,h>,<i+1,h>)$ with $a < h \le b$,

$$WT(e) \leftarrow WT(e) + 1.$$

2°. For each net $N_{n-j+1}$ in $\overline{M}_R$ which passes trough H and satisfies $a < j < b$, conduct (iv).

(iv). For every $e = (<h,j>,<h,j+1>)$ with $1 \le h \le \ell + 1$,

$$WT(e) \leftarrow WT(e) + 1.$$

<u>3°</u>. Then, return to Step 2.

<u>Step 6</u>: Let $N_a \in M_L$ and $N_{n-b+1} \in \overline{M}_R$ be end-nets of H.

<u>1°</u>. For each net $N_i$ in $M_L$ which passes through H and satisfies $a < i$, conduct (v).

(v). For every $e = (\langle i,h \rangle, \langle i+1,h \rangle)$ with $h \leq b$,

$$WT(e) \leftarrow WT(e) + 1.$$

<u>2°</u>. For each net $N_i$ in $M_L$ which passes through H and satisfies $a > i$, conduct (vi).

(vi). For every $e = (\langle i,h \rangle, \langle i+1,h \rangle)$ with $h > b$,

$$WT(e) \leftarrow WT(e) + 1.$$

<u>3°</u>. For each net $N_{n-j+1}$ in $\overline{M}_R$ which passes through H and satisfies $b < j$, conduct (vii).

(vii). For every $e = (\langle h,j \rangle, \langle h,j+1 \rangle)$ with $h \leq a$,

$$WT(e) \leftarrow WT(e) + 1.$$

<u>4°</u>. For each net $N_{n-j+1}$ in $\overline{M}_R$ which passes through H and satsifies $b > j$, conduct (viii).

(viii). For every $e = (\langle h,j \rangle, \langle h,j+1 \rangle)$ with $h > a$,

$$WT(e) \leftarrow WT(e) + 1.$$

<u>5°</u>. Then, return to Step 2.

Let us consider the processing time required in this algorithm. We can easily see that Steps 4, 5 and 6 are implemented at most in $O(n^2)$. Therefore, the total time required by the loop through Steps 2 to 6 is $O(n^2 r)$, and hence the total time required for Weight Assignment Algorithm is $O(n^2 r)$, since Step 1 is implemented in $O(n^2)$ time.

Appendix 3: (The expected Difference of the Crossing Number)

We shall describe briefly the way to compute the expected difference of the crossing number $EDC_u(p_j)$ in the upper row, which is introduced at Step I-2° of Phase II in the proposed heuristic algorithm. Let $M \triangleq (N_1, N_2, \cdots, N_c)$ be the current sequence on the middle row, and let $M_L$, $M_R$, f, and X be the current solution to Half-DRP problem with the upper case net list $\mathcal{L}_u = [U,M]$.

A strip $q_k$ in an interval graphical representation associated with an ordering f is defined as

i). the horizontal space above the interval line of net $f^{-1}(1)$, for $k=1$,

ii). the horizontal space between the interval lines of nets $f^{-1}(k-1)$ and $f^{-1}(k)$, for $2 \le k \le n$,

iii). the horizontal space below the interval line of net $f^{-1}(n)$, for $k=n+1$,

where n is the number of nets in the representation. In Fig. 13(a), an example of strips $q_k$ $(k=1,2,\cdots,8)$ is shown.

Let $N_i$ be a net selected at Step I-1° of Phase II for a displacement. As mentioned in Section 5, $EDC_u(p_j)$ indicates the number by which the crossing number may increase from X, if $N_i$ is displaced in M into position $p_j$. In other words, $EDC_u(p_j)$ is required to show that the crossing number increases at most by $EDC_u(p_j)$ by displacing $N_i$ in M into $p_j$, if $EDC_u(p_j) > 0$, and more essentially is required to show that the crossing number decreases at least by $|EDC_u(p_j)|$ by displacing $N_i$ in M into $p_j$, if $EDC_u(p_j) < 0$.

To compute such $EDC_u(p_j)$ for each $p_j$, we introduce the number $LDF(q_k)$ of intersections between the reference line and the interval lines of nets that will be increased from the current crossing number X, if $N_i$ is displaced into $q_k$ and into $M_L$, and the number $RDF(q_k)$ of intersections that will be increased from X, if $N_i$ is displaced into $q_k$ and $M_R$.

For example, if we displace net $N_2$ in the interval graphical representation shown in Fig. 13(a) into strip $q_2$ and into $\overline{M}_R$, then we have the interval graphical representation shown in Fig. 13(b). Therefore, the current crossing number $X = 17$ is reduced to 16, and hence $RDF(q_2) = -1$ as shown in the figure.

Note here that for $h = f(N_i)$, $LDF(q_h) = LDF(q_{h+1})$ and $RDF(q_h) = RDF(q_{h+1})$. Therefore, we need not provide two strips $q_h$ and $q_{h+1}$, and can combine them into one. For simplicity, however, we use both $q_h$ and $q_{h+1}$.

Once we obtain $LDF(q_k)$ and $RDF(q_k)$ for each strip $q_k$, it is easy to compute the expected difference $EDC_u(p_j)$ for each position $p_j$ for $N_i$. In Fig. 13(c), the relation among LDF, RDF, and $EDC_u$ is shown, where $EDC_u(p_1)$, for example, is given as $\min[\, LDF(q_1), LDF(q_2)\, ]$. From the relation shown in the figure, we can easily see the way to calculate $EDC_u$ values from LDF and RDF. The datailed description is omitted here.

We shall consider how to compute LDF and RDF in the following.

Consider the interval graphical representation of net list $\mathcal{L}_s = [M_L \_ U \_ M_R]$ associated with ordering f, and eliminate the interval line of $N_i$ together with the segments RL(H) of the reference line on the unit intervals H with a node of $N_i$ as endnode. With this elimination, we call the continuous line segments of the reference line from node (0) to the right the <u>left-RL</u>, and such line segments from node (∞) to the left the <u>right-RL</u>. The rest of the remaining reference line is called the <u>internal-RL</u>. We define $LOV(q_k)$, $IOV(q_k)$, and $ROV(q_k)$ for each strip $q_k$ as the number of the segments of the left-RL, the internal-RL, and the right-RL passing through strip $q_k$, respectively.

For net $N_i = N_2$ with $f(N_2) = 3$ in the interval graphical representation shown in Fig. 13(a), the left-RL, the internal-RL, and the right-RL are shown in Fig. 14(a), $LOV(q_k)$ and $ROV(q_k)$ are shown beside the figure, and $IOV(q_k)$ in Fig. 14(b).

The remainder of the reference line other than the left-RL, the internal-RL, and the right-RL, is on the unit intervals with $N_i$ as an end net. Let

$EX(q_k)$ be the number of intersections caused by such remainder of the reference line and the interval lines of other nets than $N_i$, when $N_i$ is placed on strip $q_k$ in the interval graphical representation. This $EX(q_k)$ can be counted by exploring every unit interval with $N_i$ as an end-net.

For example, consider the case shown in Fig. 14(b), in which there exist six unit intervals from a through f with $N_i = N_2$ as an end-net. It is not difficult to compute the number of intersections between RL(a) and the interval lines of nets, when $N_i = N_2$ is placed on strip $q_k$ $(1 \leq k \leq 8)$, which is shown on row $q_k$ of column a in the table beside the figure. Similarly, such nembers of intersections for other unit intervals b - f can be computed. Then, $EX(q_k)$ is obtained as the sum of all numbers on the kth row in the table.

Now, from the definitions of LOV, ROV, IOV, and EX, we can see that the current crossing number X is given as

$$X = \begin{cases} LOV(q_h) + EX(q_h) + IOV(q_h), & \text{if } N_i \in M_L, \\ ROV(q_h) + EX(q_h) + IOV(q_h), & \text{if } N_i \in M_R, \end{cases}$$

where $h = f^{-1}(N_i)$. We can also see that $LDF(q_k)$ and $RDF(q_k)$ for strip $q_k$ $(1 \leq k \leq n+1)$ is calculated by

$$LDF(q_k) = LOV(q_k) + EX(q_k) + IOV(q_k) - X, \text{ and}$$

$$RDF(q_k) = ROV(q_k) + EX(q_k) + IOV(q_k) - X.$$

Moreover, we note that for a specified $N_i$, LOV, IOV, EX, and ROV for all strips $q_k$'s are computed in $O(n \cdot r)$ time, and hence LDF and RDF for all $q_k$'s in $O(n \cdot r)$ time.

Thus, the expected difference $EDC_u(p_j)$ for all positions can be computed in $O(n \cdot r)$ time. Since the expected difference $EDC_w(p_j)$ of the crossing number on the lower row for each $p_j$ is calculated similarly, Step I - 2° in Phase II is totally implemented in $O(n \cdot r)$ time and in $O(n + r)$ space.

FIGURE CAPTIONS

Fig. 1 (a). Nodes on the upper and the lower row.

(b). A realization of net list $\mathcal{L}$ and the middle row.

(c). Allowed pattern (I) and prohibited patterns (II) and (III).

Fig. 2 (a). A realization of $\mathcal{L}_u$.

(b). A realization of $\mathcal{L}_w$.

Fig. 3 (a). Net list $\mathcal{L}_u = [U,M]$ and subsequences $M_L$ and $M_R$ of M.

(b). Net list $\mathcal{L}_s = [M_L\_U\_M_R]$.

(c). An interval graphical representation of $\mathcal{L}_s = [M_L\_U\_M_R]$ and the reference line.

(d). A realization transformed from the interval graphical representation of (c).

Fig. 4. Exceptions of routing patterns due to the interval graphical representation.

Fig. 5. Modified Half-DRP Problem.

Fig. 6. Restricted MSP Problem and Modified Half-DRP Problem.

Fig. 7. Grid digraph G for $M_L$ and $\overline{M}_R$, and a directed path corresponding to merged sequence $(N_1,N_8,N_7,N_6,N_2,N_5,N_3,N_4)$.

Fig. 8. Unit interval H and net $N_i$ which may intersect each other.

Fig. 9 (a). Net list $\mathcal{L}_s = [M_L\_U\_M_R]$ and an interval graphical representation.

(b). Grid digraph for net list in (a) and weights of edges.

Fig. 10. A frame of initial sequence M.

Fig. 11 (a). Unconnected nodes and blocks for net $f^{-1}(i)$.

(b). Interval graphical representations of $\mathcal{L}_u = [U,M]$ and $\mathcal{L}_w = [W,M]$.

Fig. 12 (a). A realization of a randomly generated net list. (CASE I).

(b). A realization of a randomly generated net list. (CASE II).

Fig. 13 (a). Strips in an interval graphical representations.

(b). LDF and RDF for each strip.

(c). The relation among LDF, RDF, and $EDC_u$.

Fig. 14 (a). The left-, internal-, and right-RL's, and LOV and ROV.

(b). EX and IOV for each strip.

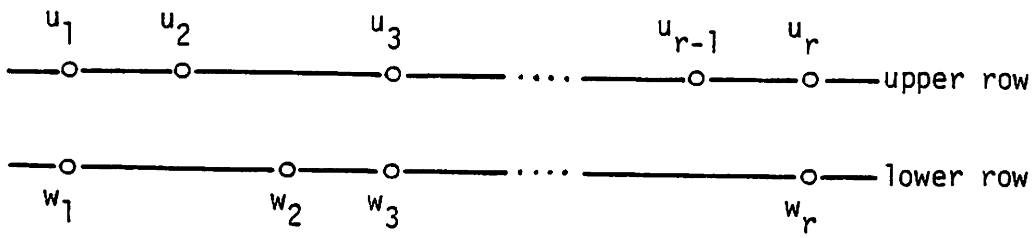


Fig. 1 (a). Nodes on the upper and the lower row.

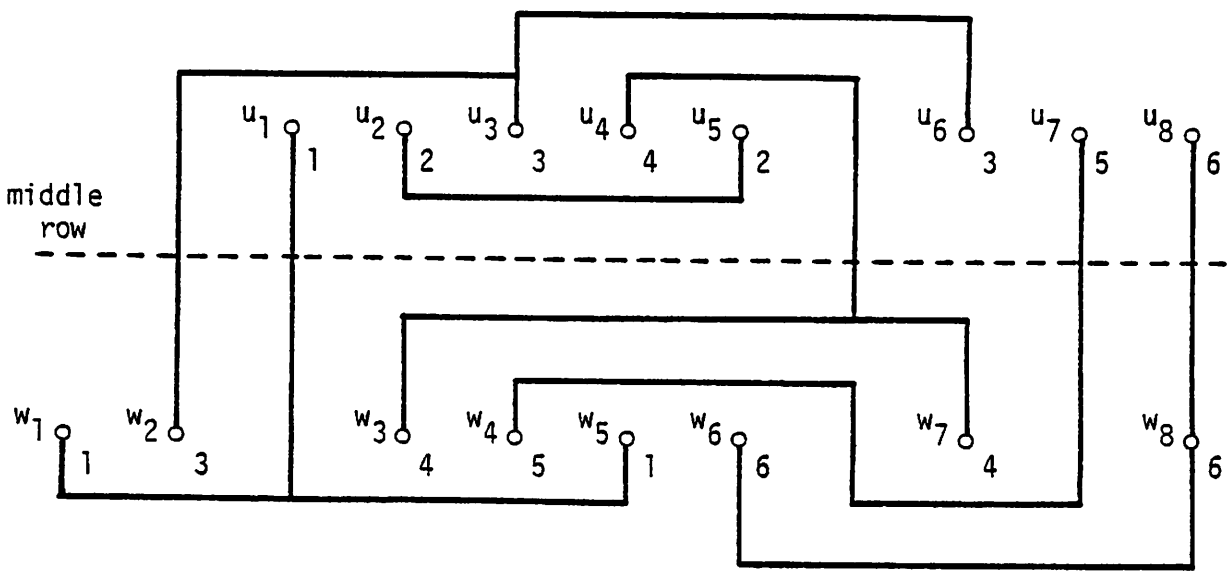


Fig. 1 (b). A realization of net list $\mathcal{L}$ and the middle row.

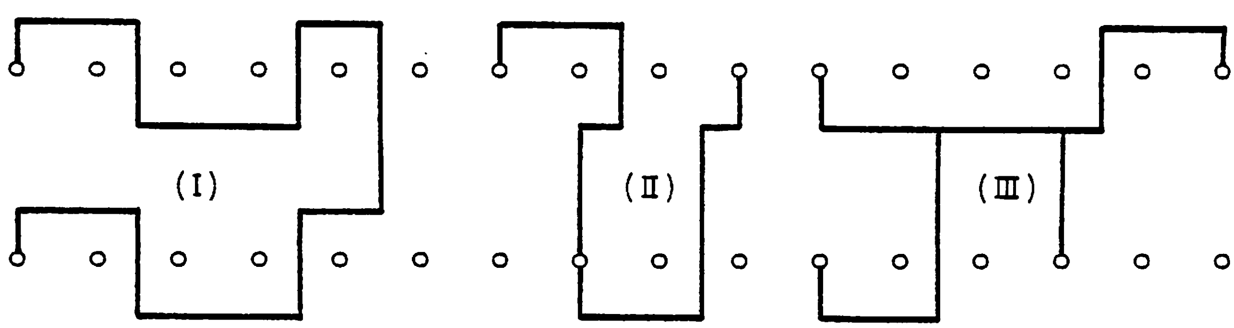


Fig. 1 (c). Allowed pattern (I) and prohibited patterns (II) and (III).

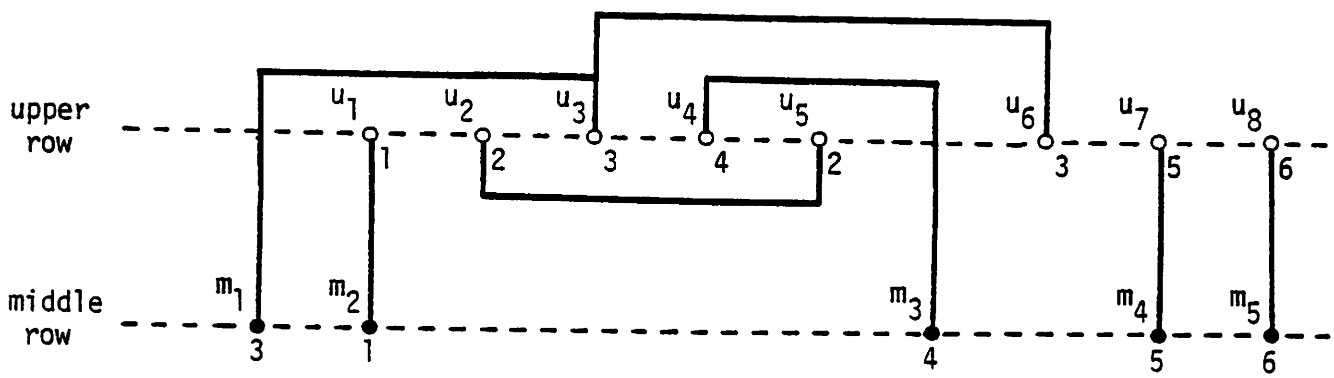


Fig. 2 (a). A realization of $\mathcal{L}_u$.

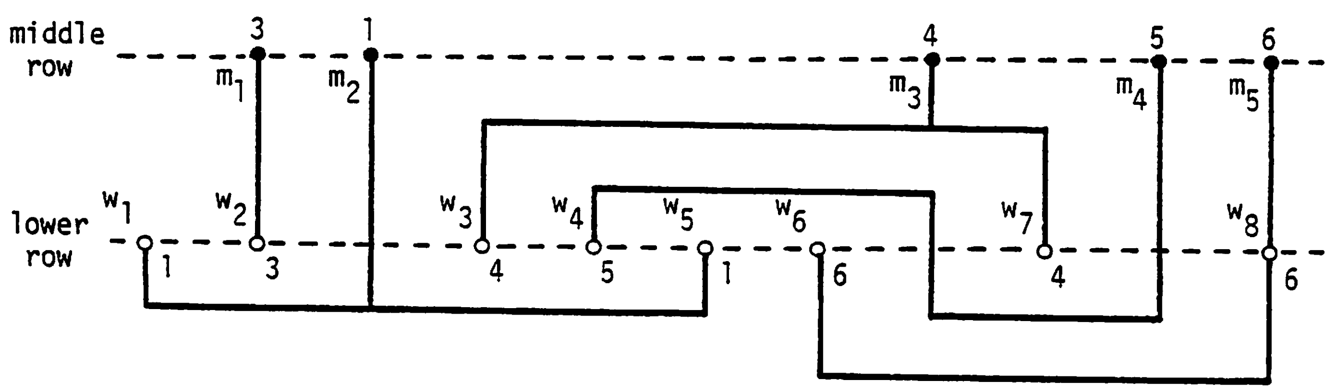


Fig. 2 (b). A realization of $\mathcal{L}_w$.

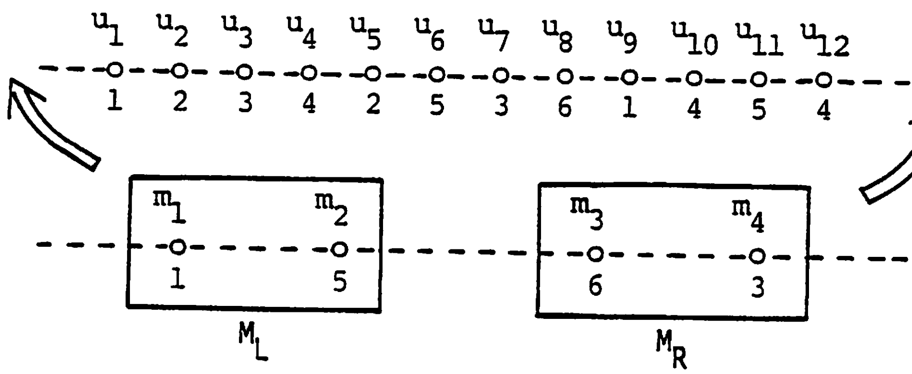


Fig. 3 (a). Net list $\mathcal{L}_u = [U,M]$ and subsequences $M_L$ and $M_R$ of M.

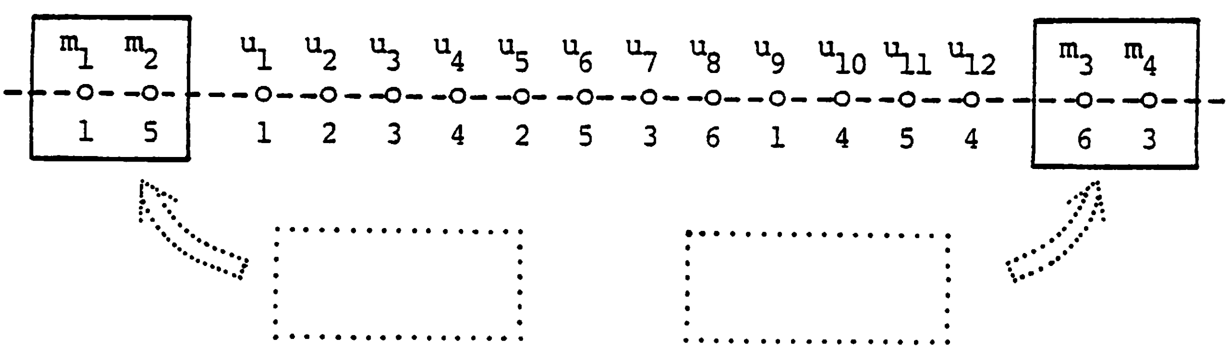


Fig. 3 (b). Net list $\mathcal{L}_s = [M_L\_U\_M_R]$.

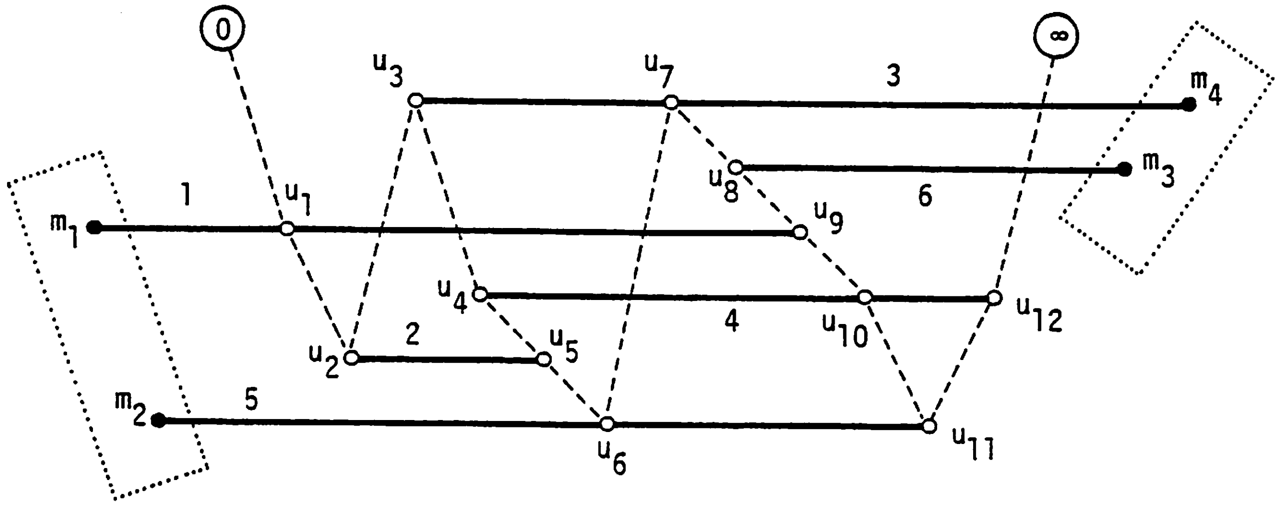


Fig. 3 (c). An interval graphical representation of $\mathcal{L}_s = [M_L - U - M_R]$ and the reference line.

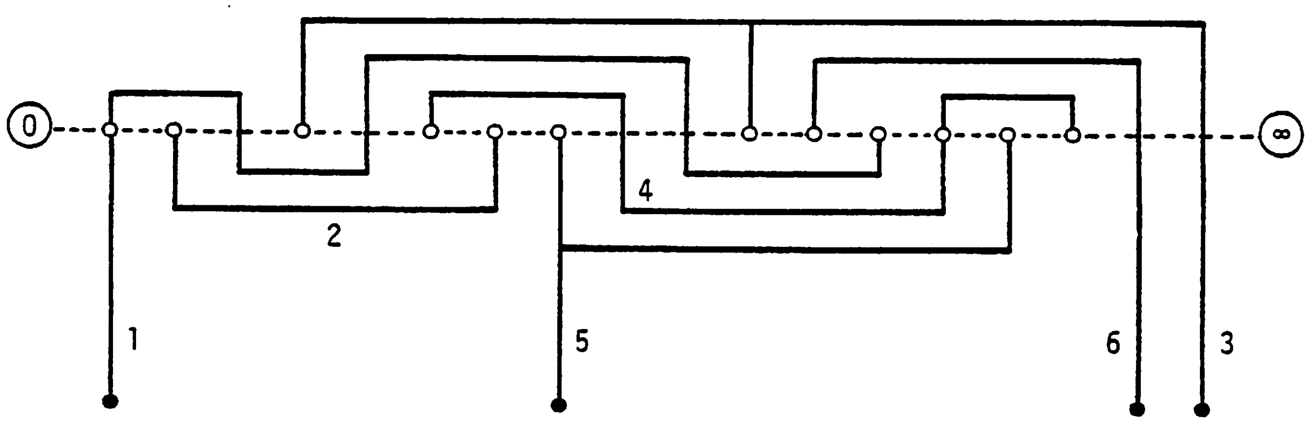


Fig. 3 (d). A realization transformed from the interval graphical representation of (c).

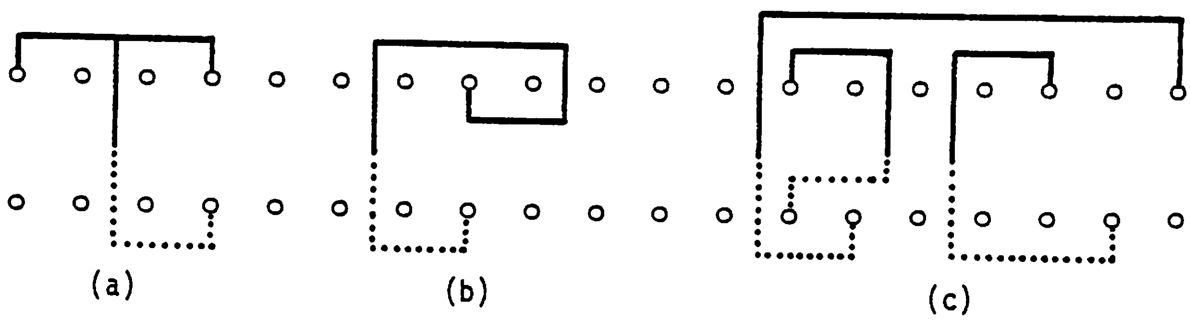


Fig. 4. Exceptions of routing patterns due to the interval graphical representation.

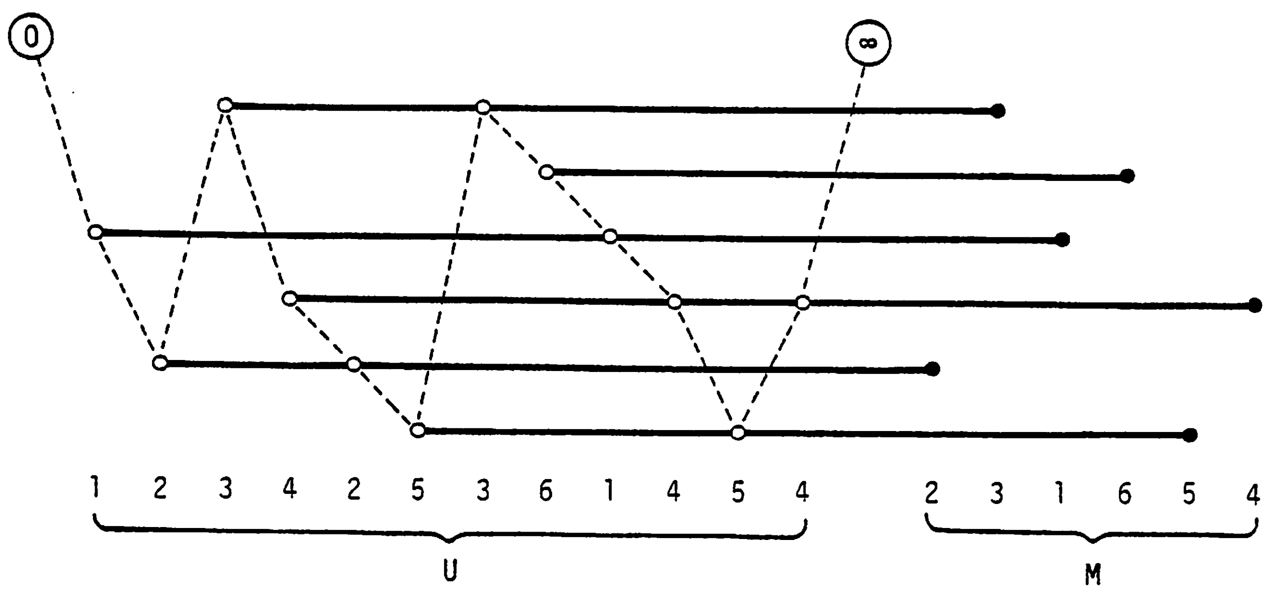


Fig. 5. Modified Half-DRP Problem.

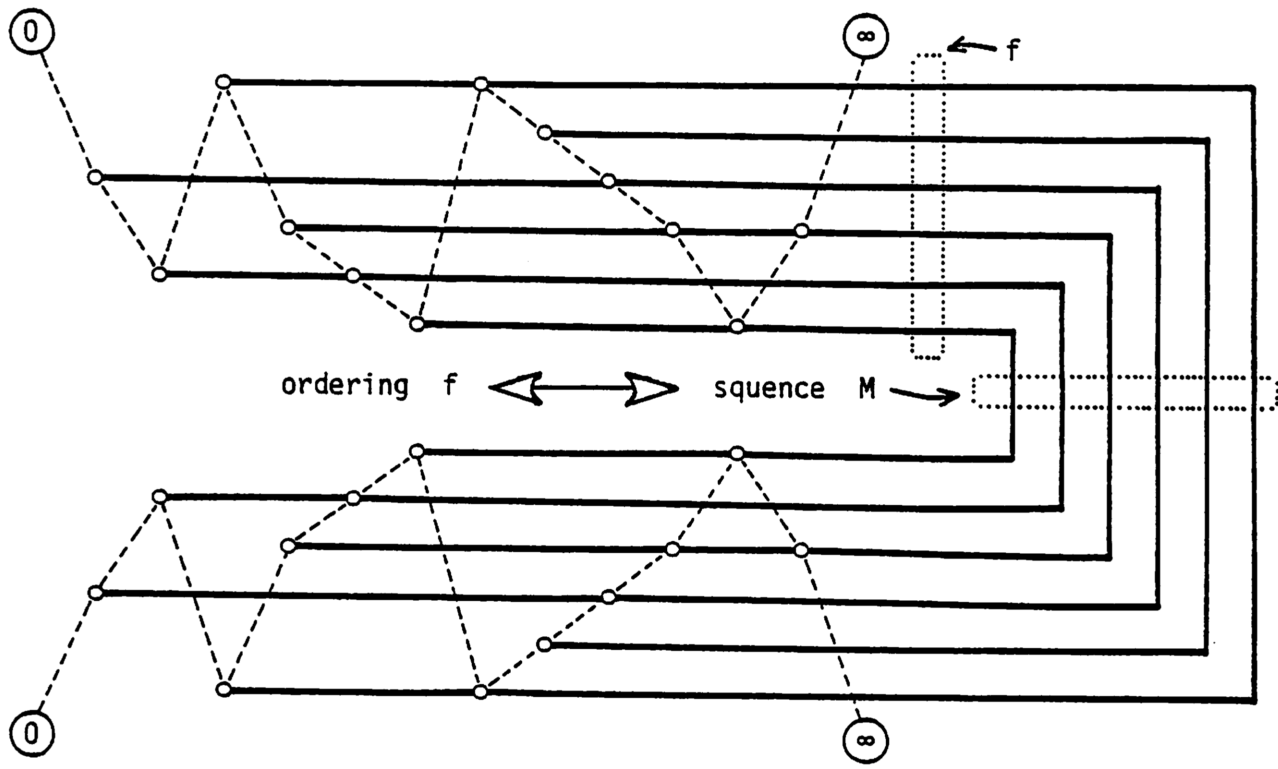


Fig. 6. Restricted MSP Problem and Modified Half-DRP Problem.

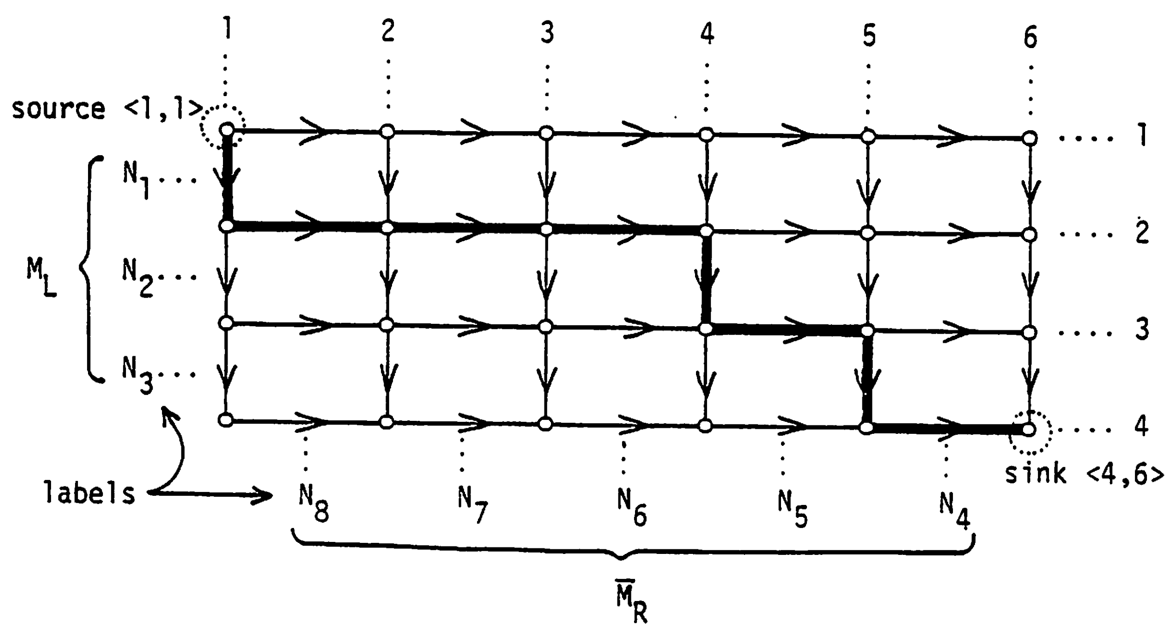


Fig. 7. Grid digraph G for $M_L$ and $\overline{M}_R$, and a directed path corresponding to merged sequence $(N_1,N_8,N_7,N_6,N_2,N_5,N_3,N_4)$.

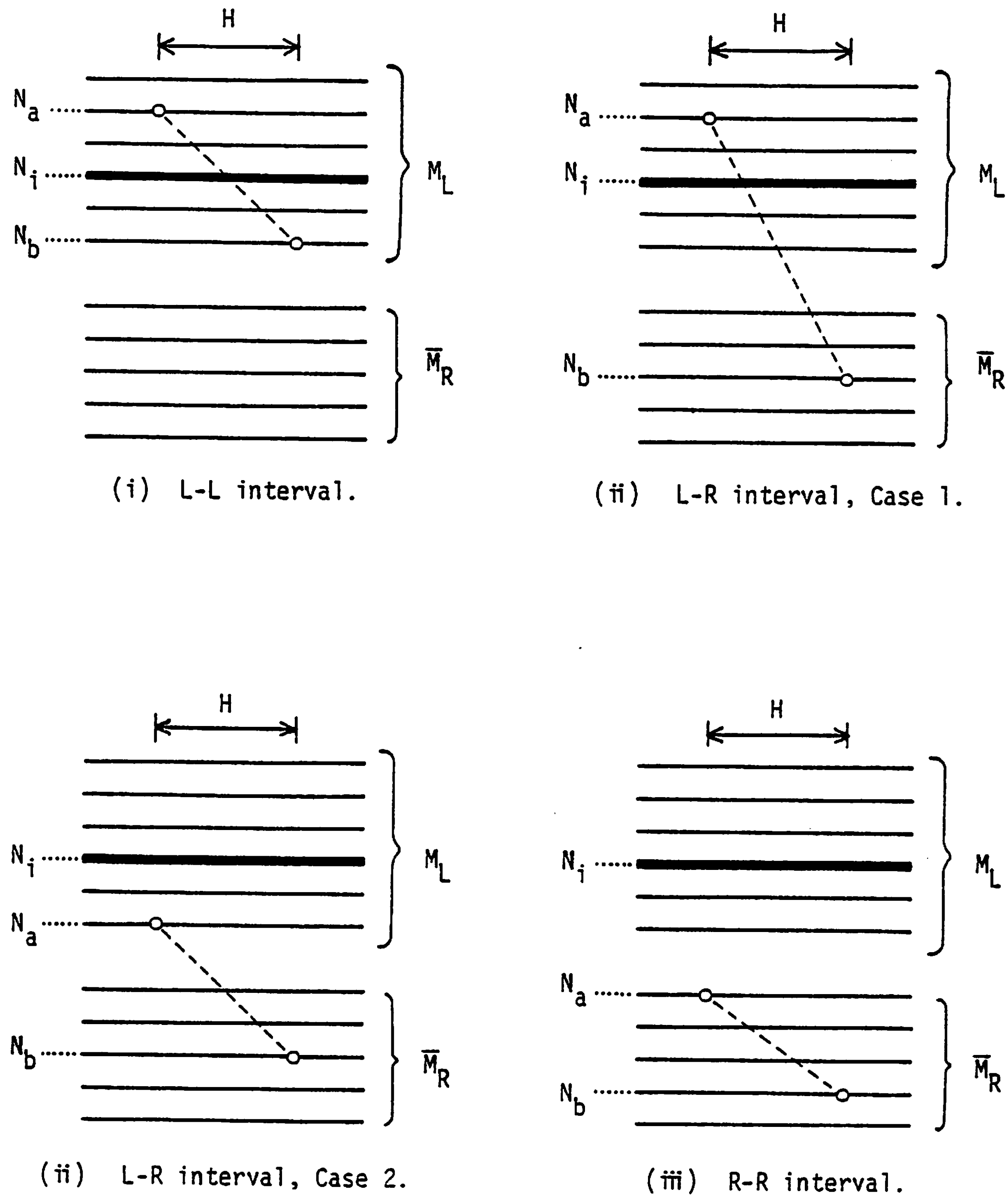


Fig. 8. Unit interval H and net $N_i$ which may intersect each other.

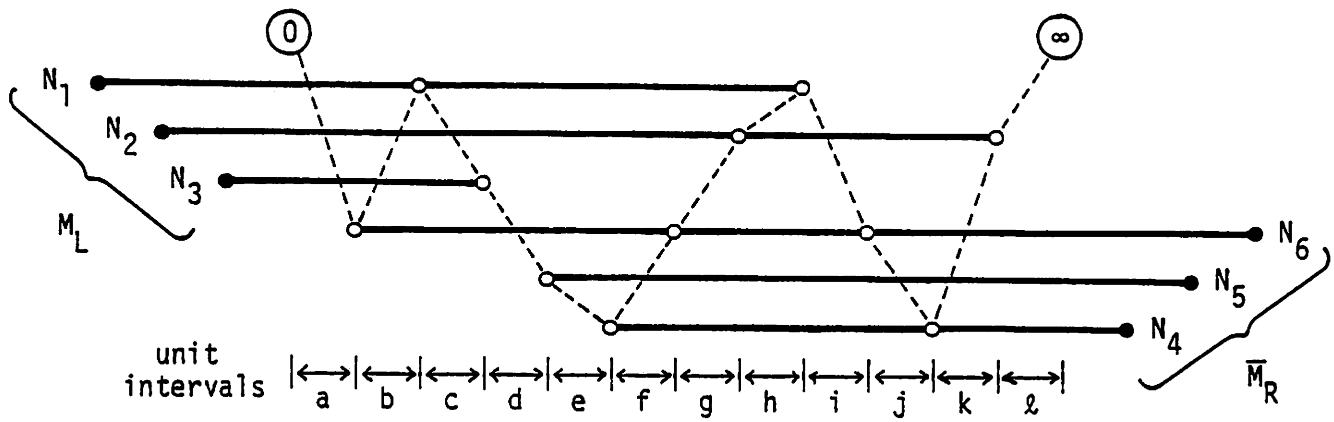


Fig. 9 (a). Net list $\mathcal{L}_s = [M_L \_ U \_ M_R]$ and an interval graphical representation.

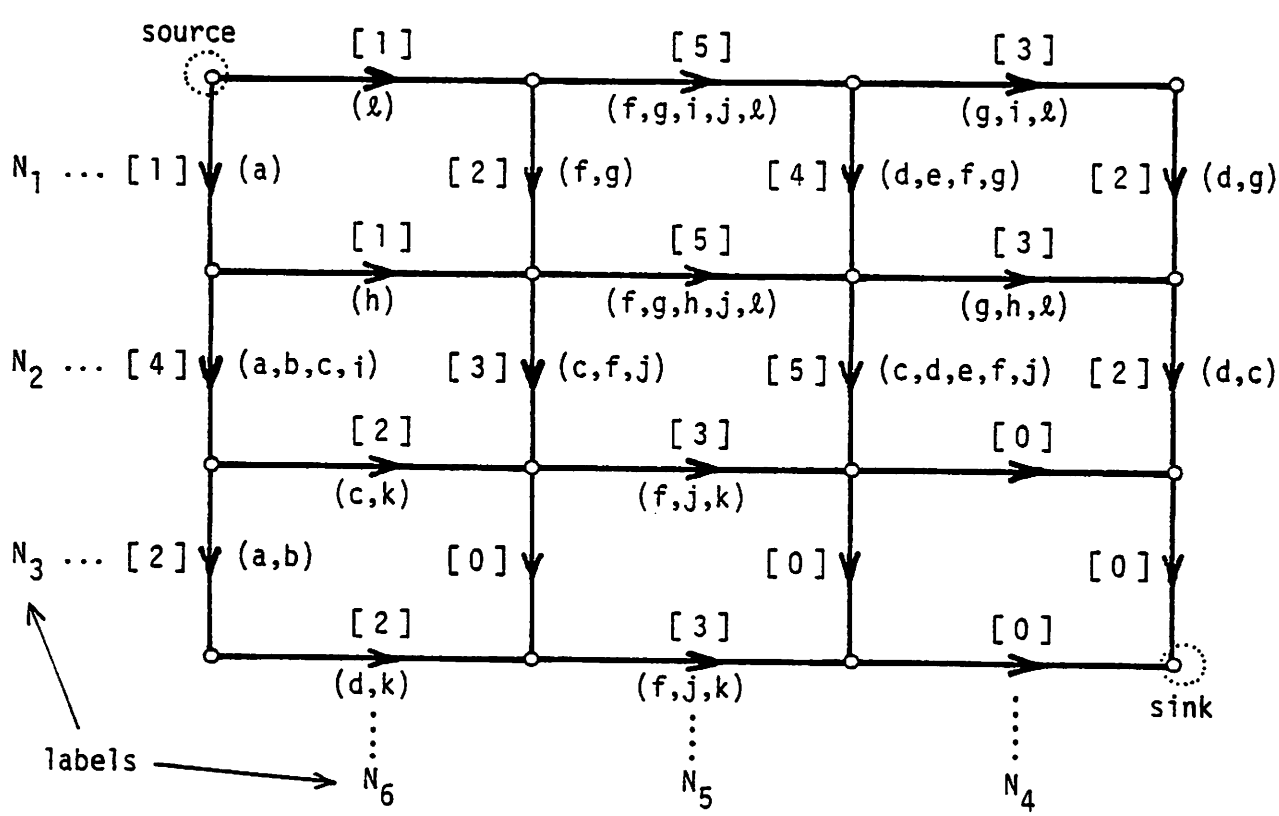


Fig. 9 (b). Grid digraph for net list in (a) and weights of edges.

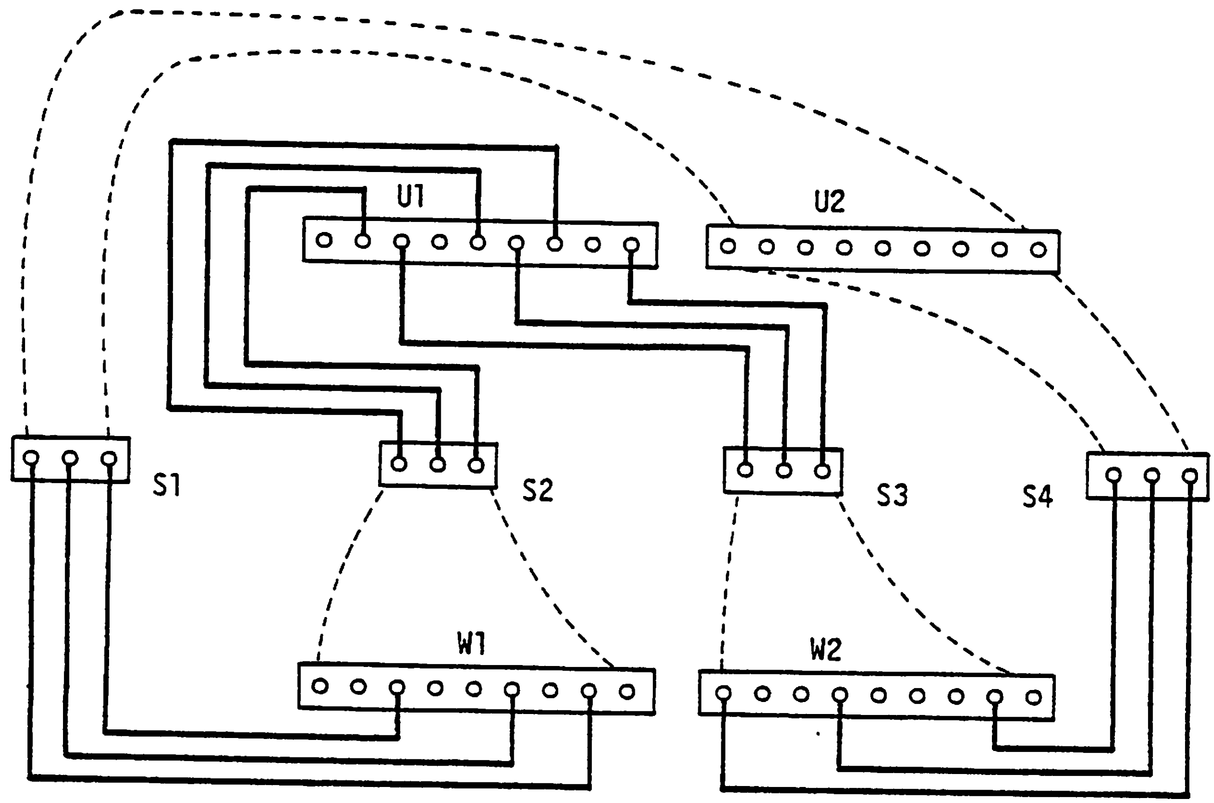


Fig. 10. A frame of initial sequence M.

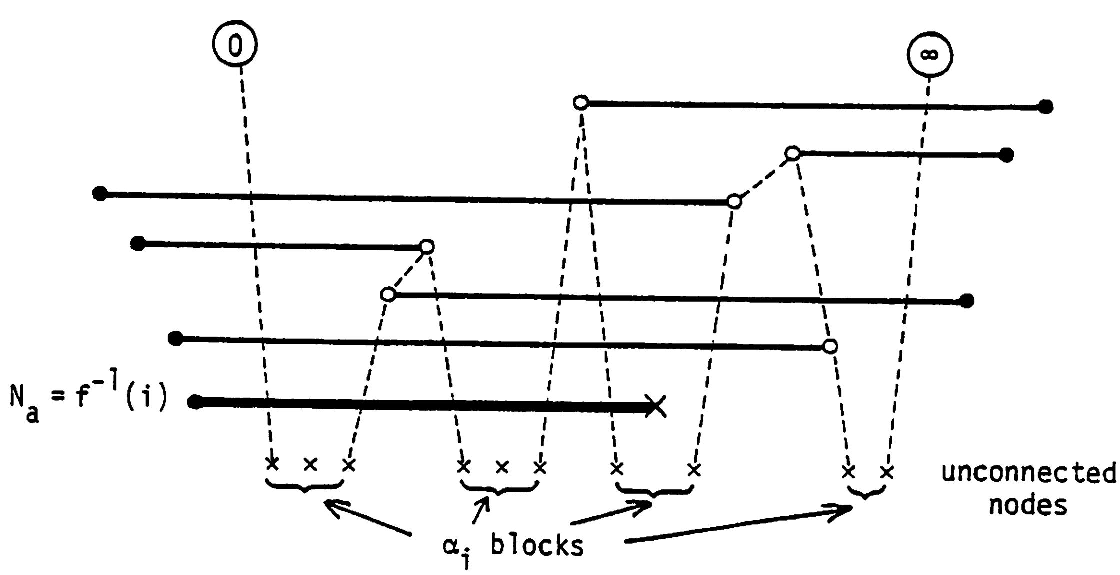


Fig. 11 (a). Unconnected nodes and blocks for net $f^{-1}(i)$.

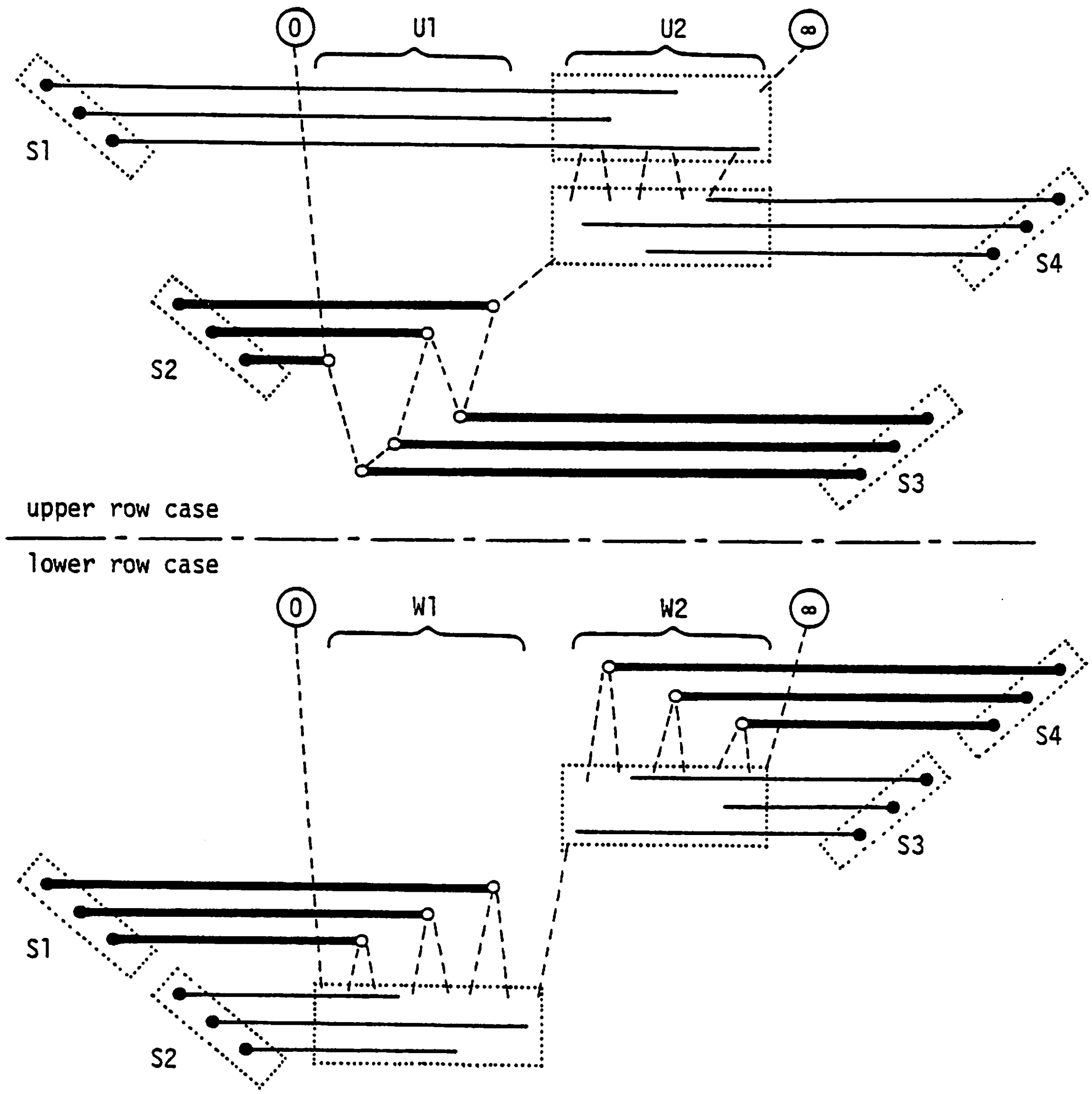


Fig. 11 (b). Interval graphical representations of $\mathcal{L}_u$ = [U,M] and $\mathcal{L}_w$ = [W,M].

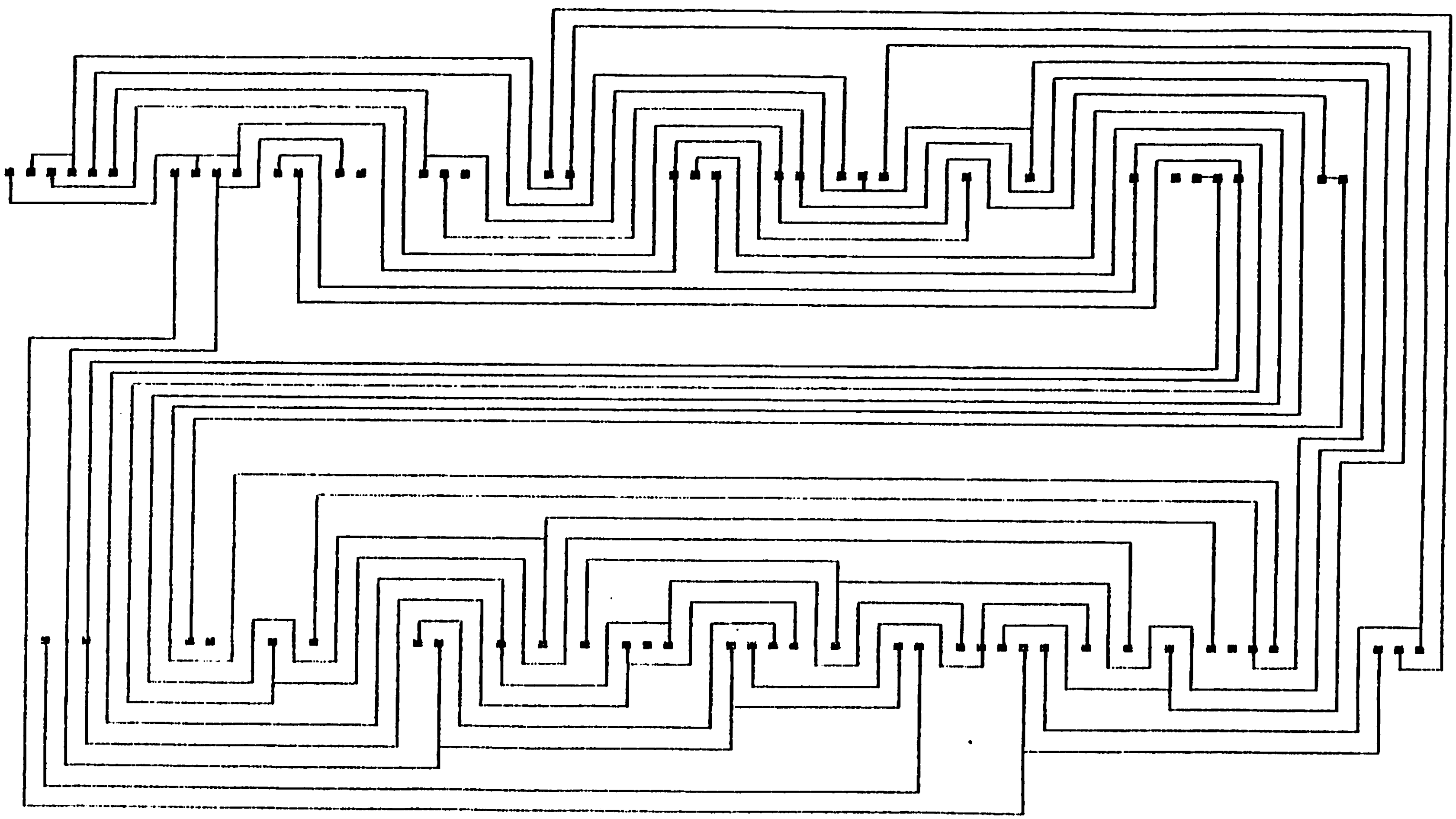

Fig. 12 (a). A realization of a randomly generated net list. (CASE I).

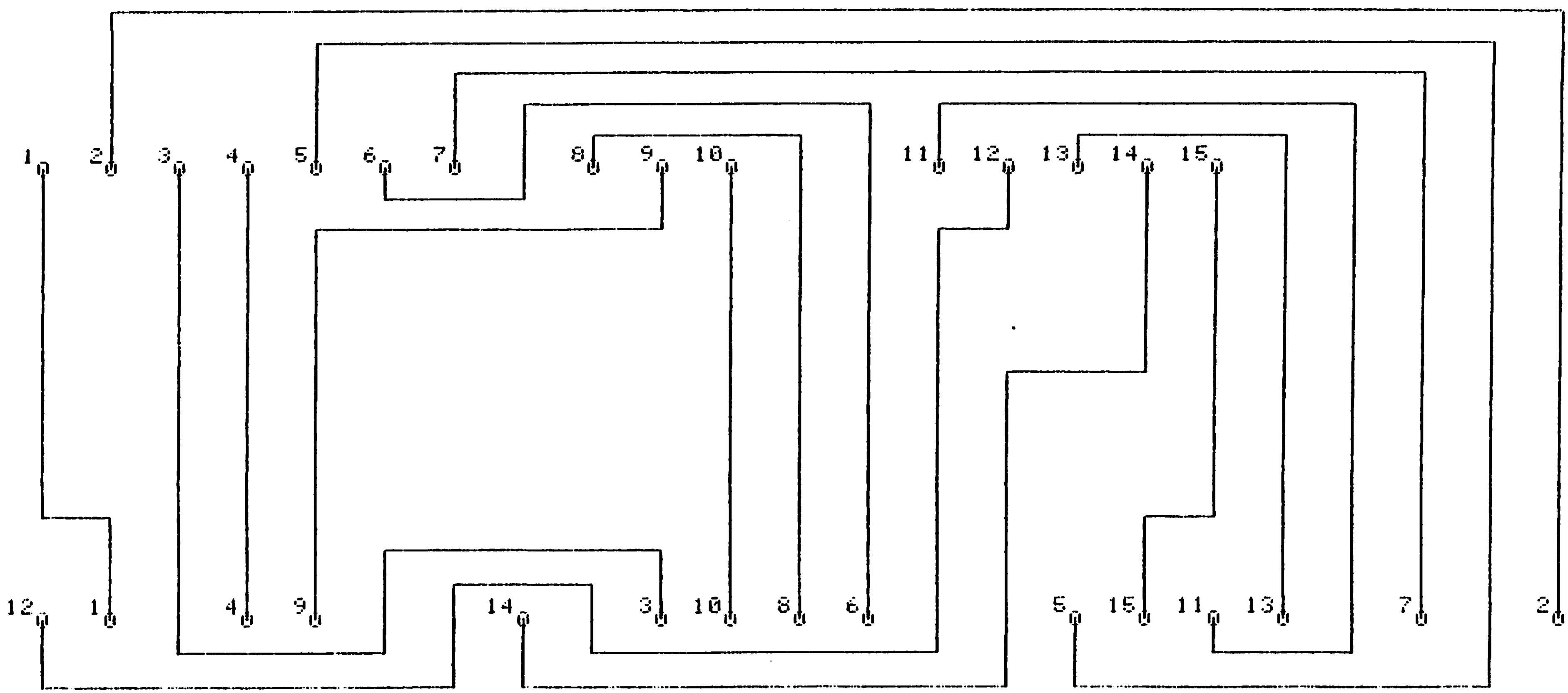


Fig. 12 (b). A realization of a randomly generated net list. (CASE II).

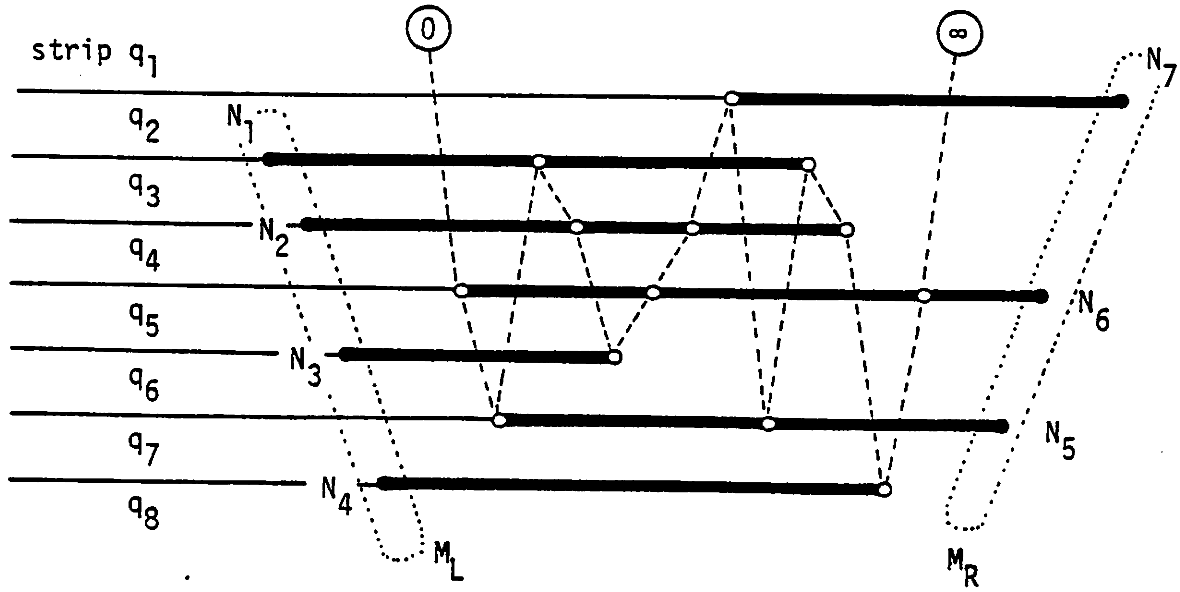


Fig. 13 (a). Strips in an interval graphical representation.

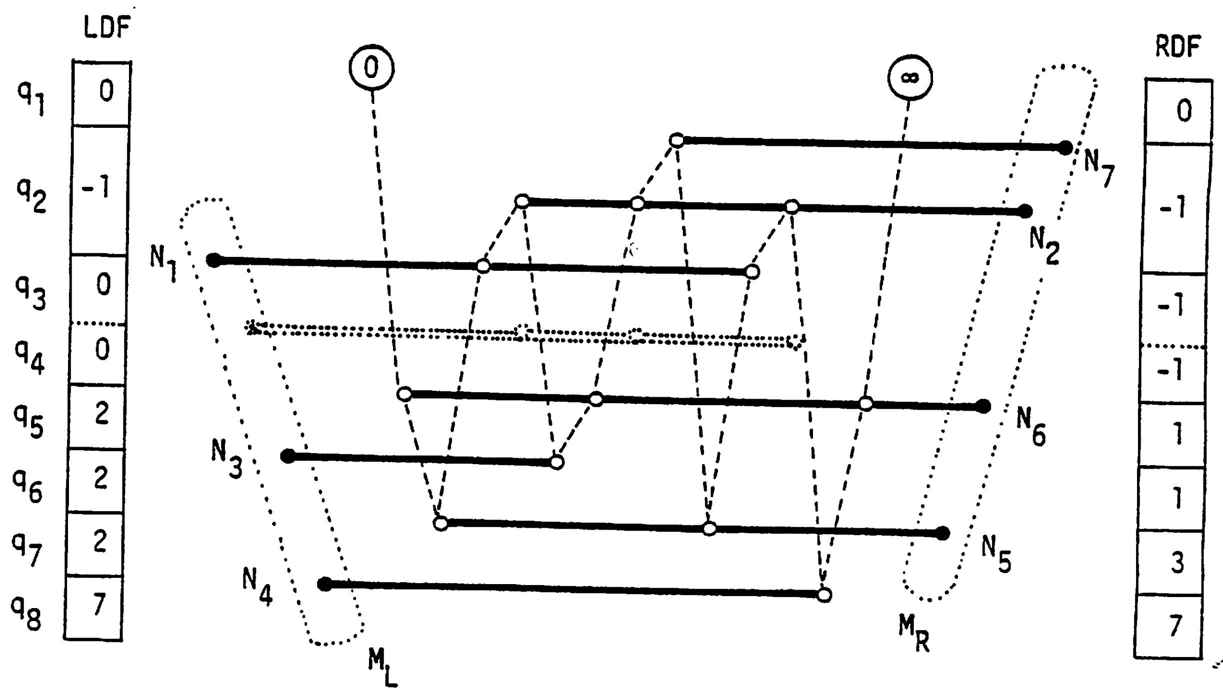


Fig. 13 (b). LDF and RDF for each strip.

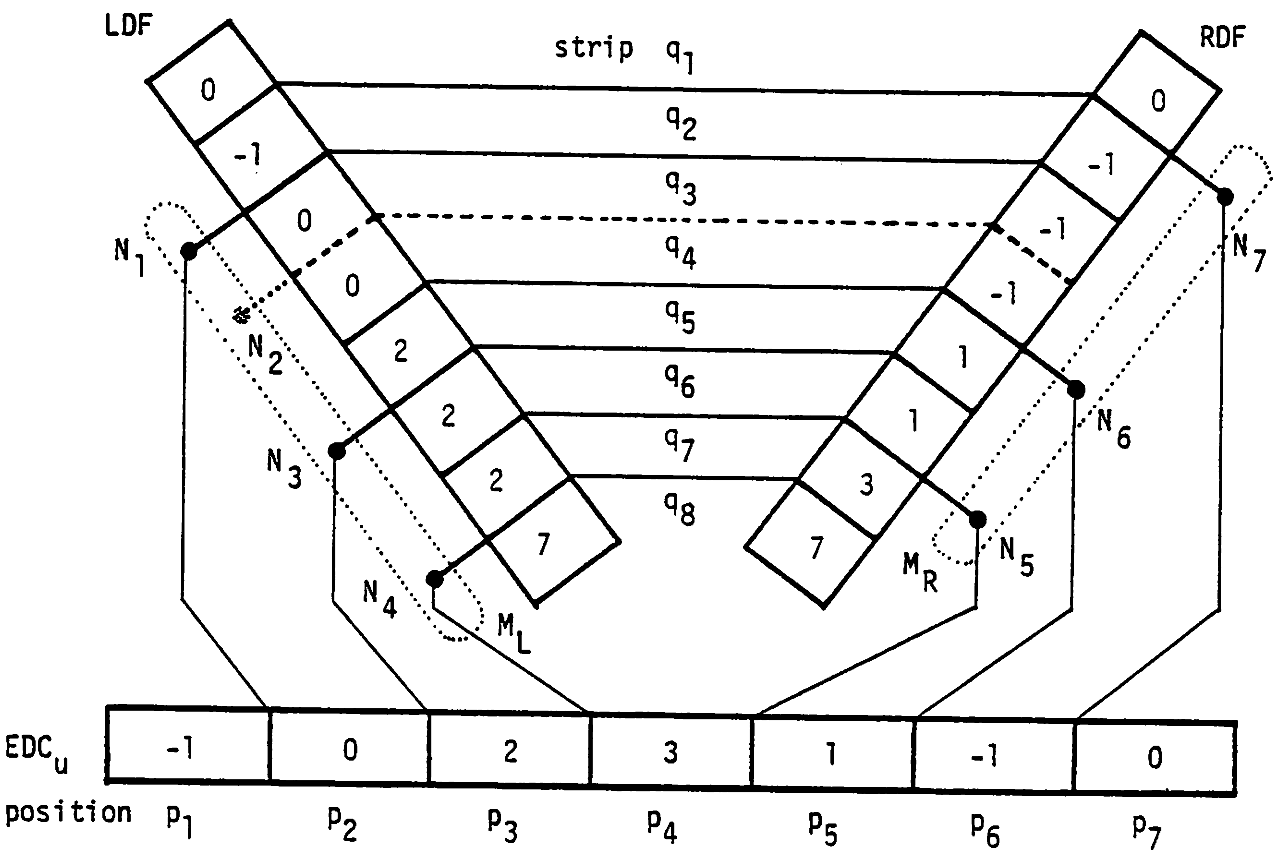


Fig. 13 (c). The relation among LDF, RDF, and $EDC_u$.

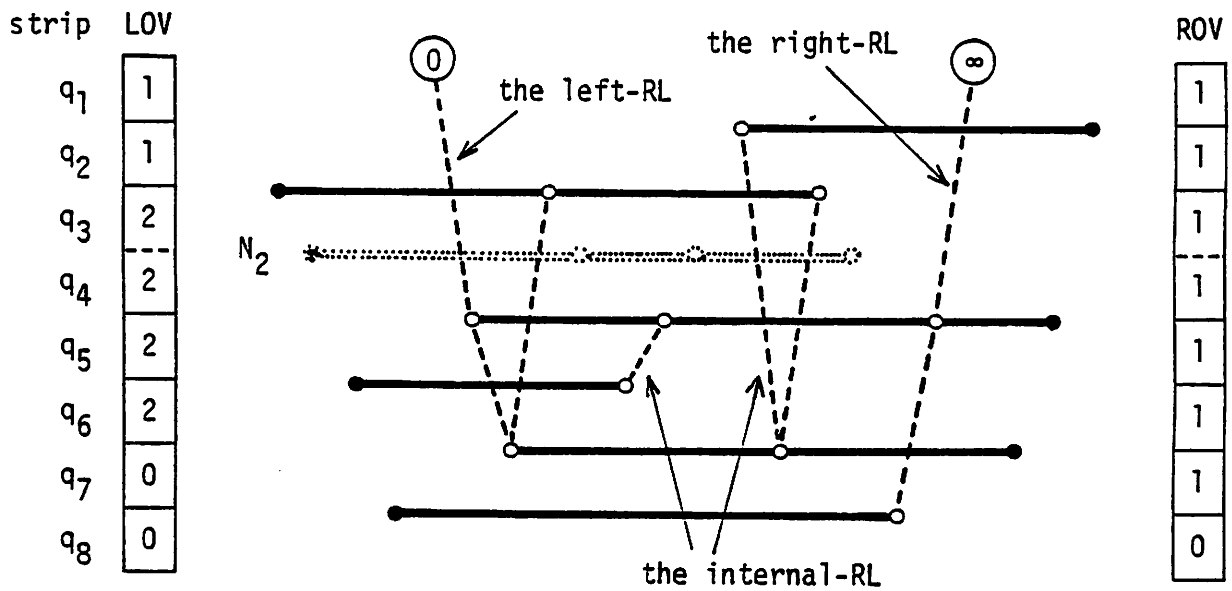


Fig. 14 (a). The left-, internal-, and right-RL's, and LOV and ROV.

| | a | b | c | d | e | f | EX | IOV |
|---|---|---|---|---|---|---|---|---|
| $q_1$ | 0 | 2 | 1 | 0 | 1 | 3 | 7 | 0 |
| $q_2$ | 0 | 2 | 1 | 0 | 0 | 2 | 5 | 1 |
| $q_3$ | 0 | 1 | 0 | 1 | 0 | 2 | 4 | 2 |
| $q_4$ | 0 | 1 | 0 | 1 | 0 | 2 | 4 | 2 |
| $q_5$ | 1 | 0 | 0 | 2 | 1 | 1 | 5 | 3 |
| $q_6$ | 2 | 0 | 0 | 2 | 1 | 1 | 6 | 2 |
| $q_7$ | 3 | 1 | 1 | 3 | 2 | 0 | 10 | 0 |
| $q_8$ | 4 | 2 | 2 | 4 | 3 | 0 | 15 | 0 |

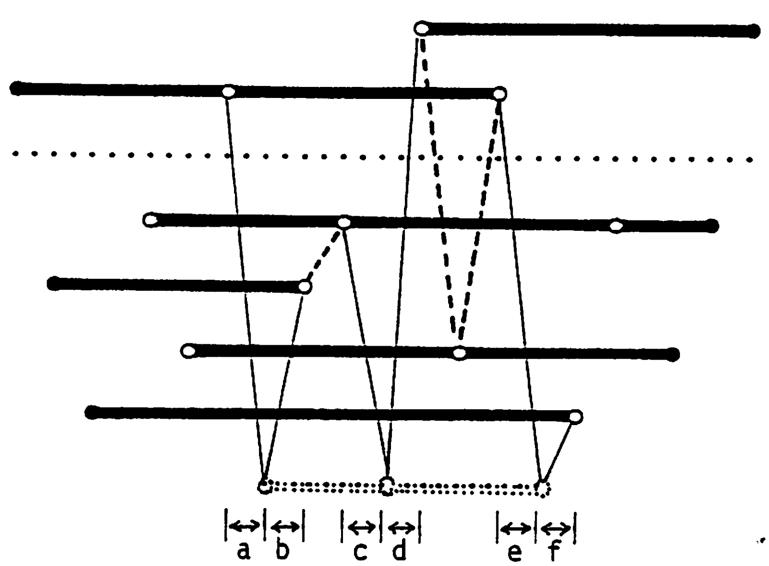


Fig. 14 (b). EX and IOV for each strip.