would include risk-aversion as well as the trade-off between efficiency *after* disagreement and efficiency ex ante (before the process begins).

ACKNOWLEDGMENT

I wish to acknowledge the invaluable help of Howard Raiffa, who kindled my interest in this subject, and James Healy, who shared some of his experiences as an arbiter with me as well as introducing me to part of the industrial relations literature. I also wish to thank three anonymous referees for comments.

REFERENCES

[1] K. Chatterjee, "Interactive decision problems with differential information," unpublished DBA dissertation, Harvard Univ., Cambridge, MA, 1979.
[2] K. Chatterjee and W. Samuelson, "The simple economics of bargaining." mimeo, Pennsylvania State University and Boston University, 1979.
[3] K. Chatterjee and J. W. Ulvila, "Bargaining with shared information." mimeo, Pennsylvania State University and Decision Science Consortium, 1980.
[4] V. P. Crawford, "On compulsory arbitration schemes," *J. Political Econ.*, vol. 87, pp. 131–159, Feb. 1979.
[5] ——, "Arbitration and conflict resolution in labor management bargaining," in *Papers and Proc.*, American Economic Association Meeting, Sept. 1980, to appear.
[6] Henry S. Farber and H. C. Katz, "Interest arbitration, outcomes and the incentives to bargain," *Industrial, Labor Relations Rev.*, vol. 33, no. 1, Oct. 1979.
[7] P. Feuille, "Final-offer arbitration; concepts, developments, techniques," International Personnel Management Association, Chicago, IL, 1975.
[8] T. A. Kochan, R. G. Ehrenberg, J. Baderschneider, T. Jick, and M. Mironi, "An evaluation of impasse procedures for police and firefighters in New York State," Cornell Univ., New York State School of Industrial and Labor Relations, 1976.
[9] D. B. Lipsky, T. A. Barocci with W. Suoyanen, "The impact of final-offer arbitration in Massachusetts," mimeo, Massachusetts Institute of Technology, Cambridge, MA., 1977.
[10] R. Duncan Luce and Howard Raiffa, *Games and Decisions*. New York: Wiley, 1957.
[11] R. B. Myerson, "Incentive compatability and the bargaining problem," *Econometrica*, vol. 47, pp. 61–73, Jan. 1979.
[12] R. W. Rosenthal, "Arbitration under uncertainty," in *Rev. Economic Studies*, vol. 15, pp. 595–604, 1979.
[13] P. C. Somers, "An evaluation of final-offer arbitration in Massachusetts," Massachusetts League of Cities and Towns, Boston, MA., 1976.
[14] J. L. Stern, C. M. Rehmus, J. J. Loewenberg, H. Kasper, and B. D. Dennis, *Final-Offer Arbitration*. Cambridge, MA: Lexington Books, 1975.

Methods for Visual Understanding of Hierarchical System Structures

KOZO SUGIYAMA, MEMBER, IEEE, SHOJIRO TAGAWA, AND MITSUHIKO TODA, MEMBER, IEEE

***Abstract*—Two kinds of new methods are developed to obtain effective representations of hierarchies automatically: theoretical and heuristic methods. The methods determine the positions of vertices in two steps. First the order of the vertices in each level is determined to reduce the number of crossings of edges. Then horizontal positions of the vertices are determined to improve further the readability of drawings. The theoretical methods are useful in recognizing the nature of the problem, and the heuristic methods make it possible to enlarge the size of hierarchies with which we can deal. Performance tests of the heuristic methods and several applications are presented.**

Manuscript received February 18, 1980; revised October 14, 1980. The main part of this paper was presented at the International Conference on Cybernetics and Society held in Denver, Oct. 1979.

The authors are with the International Institute for Advanced Study of Social Information Science, Fujitsu Limited, 140, Miyamoto, Numazu-shi, Shizuoka, 410-03 Japan.

DIGRAPHS are widely utilized in modeling structures of complex systems in various fields where vertices correspond to elements of the systems and edges correspond to relations among the elements. It is empirically recognized that drawings of the digraphs are useful as a visual aid to understand overall images of the structures of the complex systems. For example, block diagrams and flowcharts are commonly used by engineers in performing tasks such as structural modeling, project scheduling, computer programming, etc.

Multilevel digraphs, called hierarchies, constitute an important subclass of digraphs. Interpretive Structural Modeling (ISM) [1] and Program Evaluation and Review Technique (PERT) [2] are the well-known techniques in which hierarchies are utilized for modeling structures of systems.

0018-9472/81/0200-0109$00.75

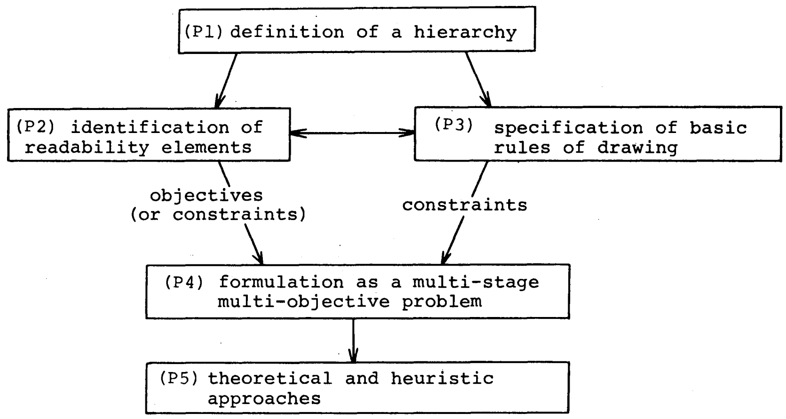


Fig. 1. Schematic diagram of procedures to develop methods for generating readable maps.

In ISM, structural models of systems are developed with the aid of computers through interactive learning processes, where participants to ISM exercises improve the models, observing the drawings of the hierarchies displayed on graphic terminals. Therefore it is important for the participants that the hierarchies are drawn in a form readily grasped by human eyes. Moreover an automatic method for the drawing of hierarchies by a computer is indispensable because many hierarchies produced through the learning process should be drawn one after another without much lapse of time. In PERT, precedence relations among many activities constituting a project are illustrated by hierarchies. The number of vertices and edges (or activities and relations) included in PERT hierarchies is usually quite large, and engineers engaged in the project need to see, in their daily work, the drawings of the hierarchies which are visually understandable in terms of both local and holistic structures. It is, however, extremely difficult to manually draw the visually understandable hierarchies when the sizes of the hierarchies are large.

This paper is intended to present methods to generate a visually understandable drawing of a hierarchy (which we call "a readable map" for simplicity) automatically by a computer. In the whole algorithm developed, a set of directed pairwise relations among elements of a system is given and a readable map is obtained. Such an automatic drawing of a hierarchy is of great assistance to an efficient analysis of complex systems and interactive use of a computer.

We analyze, formulate, and solve the problem to generate a readable map by the procedures illustrated in Fig. 1. Procedures P1 to P5 are explained in the following.

(P1) Definition of a Hierarchy

A hierarchy is defined by terms of graph theory. This definition clarifies boundaries of discussions on readability.

(P2) Identification of Readability Elements

"Readability" of a map is analyzed. The readability may intrinsically depend upon problems studied and on the audiences of the map. The purpose of this analysis, however, is to consider common aspects of the readability. Generally speaking, it is difficult to grasp the structure of a digraph readily unless vertices are laid out in some regular form (e.g., clustered layout) and/or unless edges are drawn in such a form that paths can be readily traced by human eyes. In the case of hierarchies, the former, regular layout of vertices, is identified as the following readability element.

Element A: "Hierarchical" layout of vertices.

The latter, traceability of paths, is broken down into the following four readability elements.

Element B: "Less-crossings" of lines (edges). The greatest difficulty in tracing paths is line crossings.

Element C: "Straightness" of lines. It is easy to trace straight lines. This element is further broken down into "straightness" of one-span edges (Element C_1) and "straightness" of long span edges (Element C_2).

Element D: "Close" layout of vertices connected to each other. It is desirable that paths are short.

Element E: "Balanced" layout of lines coming into or going from a vertex. This means that the structural information on branching and joining of paths is drawn clearly.

(P3) Specification of Basic Rules of Drawing

Basic rules to draw a hierarchy are specified. In order to draw a hierarchy, we should determine a layout of vertices and how to draw edges. We specify the basic rules regarding these two aspects as follows.

Rule a) Vertices are placed on horizontal lines in each level of the hierarchy without overlapping.

Rule b) Each edge is drawn with a straight line.

The problem is simplified by the specification of the basic rules for drawing. We have only to determine horizontal positions of vertices which attain readability. It should be noted that Rule a) and Rule b) correspond to Element A and Element C_1, respectively.

(P4) Formulation as a Multstage Multiobjective Problem

The problem to generate readable maps is formulated as a multistage multiobjective programming problem. The whole algorithm developed includes four steps, Step I to Step IV. Step I is the preparatory step for a transformation. Step II and Step III are main steps where the layout of vertices is determined. Rule a) (Element A) and Rule b) (Element C_1) are constraints for the main steps. In the first main step, Element B, which is the most common among the elements except A and C_1, is attained and in the second main step, Elements C_2, D, and E are attained. Step IV is the step for drawing. An outline of the whole procedure is described in the following.

Step I: A "proper" hierarchy [3] is formed from a given set of directed pairwise relations among elements of a system. If the digraph generated from the relations has cycles, it is condensed and we obtain a multilevel digraph or a hierarchy. Then, if the hierarchy has long span edges, it is converted into a proper hierarchy by adding dummy vertices and edges.

Step II: The number of crossings of edges in the proper hierarchy is reduced by permuting orders of vertices in each level.

Step III: Horizontal positions of vertices are determined by considering three elements C_2, D, and E. The order of the vertices determined in Step II is given as constraints to preserve the reduced number of crossings.

Step IV: A (two-dimensional) picture of the hierarchy is automatically drawn where the dummy vertices and edges are deleted and the corresponding long span edges are regenerated.

An example of improvements of a drawing in Steps II and III is shown in Fig. 2.

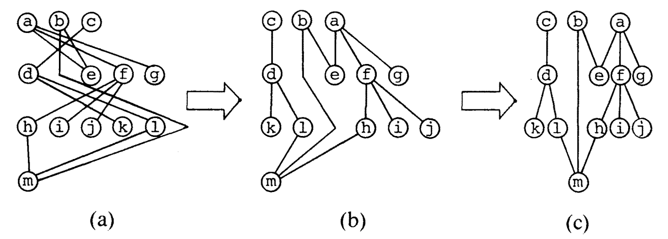


Fig. 2. An example of the improvements of a drawing. Step II: (a)→(b). Step III: (b)→(c).

(P5) Theoretical and Heuristic Approaches

Both theoretical and heuristic approaches in developing algorithms are carried out, since the theoretical methods are useful in recognizing the nature of the problem, and the heuristic methods make it possible to enlarge the size of hierarchies with which we can deal. Crossing theory of multilevel digraphs for the algorithms of Step II, has been discussed by Warfield [3]. We extend his algorithm that finds the minimum solutions of a two-level hierarchy to the cases where the number of vertices in each level is greater than five. This problem is formulated as a "minimum feedback arc set" problem [4] in graph theory, and this algorithm is called the penalty minimization (PM) method. Since the crossing problem is combinatorial in nature, the minimum solutions require extensive use of computing time. Therefore, we have developed a heuristic algorithm called the barycentric (BC) method. Its usefulness is justified by testing the performance of the algorithm.

Two different kinds of optimization and heuristic methods have been developed as the algorithms for Step III. The former is called the quadratic programming (QP) layout method since the problem is formulated in the form of quadratic programming. The latter is called the priority (PR) layout method where the computing cost is significantly less than the QP method.

The analysis and discussions in this section are summarized in Table I. Step I and Step IV are self-explanatory. Therefore in the subsequent sections we will discuss Step II and Step III.

II. Basic Definitions

A. n-Level Hierarchy and Map

An n-level hierarchy ($n \geqslant 2$) is defined as a directed graph (V,E), where V is called a set of vertices and E a set of edges, which satisfies the following conditions.

1) V is partitioned into n subsets, that is

$$V = V_1 \cup V_2 \cup \cdots \cup V_n \qquad (V_i \cap V_j = \varnothing, i \neq j)$$

where V_i is called the ith level and n the length of the hierarchy.

2) Every edge $e = (v_i, v_j) \in E$, where $v_i \in V_i$ and $v_j \in V_j$, satisfies $i < j$, and each edge in E is unique.

The n-level hierarchy is denoted by $G = (V,E,n)$.

An n-level hierarchy is called "proper" when it satisfies further the following conditions.

3) E is partitioned into $n-1$ subsets, that is

$$E = E_1 \cup E_2 \cup \cdots \cup E_{n-1} \qquad (E_i \cap E_j = \varnothing, i \neq j),$$

where $E_i \subset V_i \times V_{i+1}$, $i = 1, \cdots, n-1$.

4) An order σ_i of V_i is given for each i, where the term "order" means a sequence of all vertices of V_i; $\sigma_i = v_1 v_2 \cdots v_{|V_i|}$ ($|V_i|$ denotes the number of vertices of V_i). The n-level hierarchy is denoted by $G = (V,E,n,\sigma)$, where $\sigma = (\sigma_1, \cdots, \sigma_n)$.

This definition of the n-level proper hierarchy is slightly different from Warfield's definition [3] in the following two points.

a) In the former definition, edges are directed with ascending orders of levels, while descending in the latter.

b) In the former definition, orders of vertices are explicitly specified by σ, while not specified in the latter.

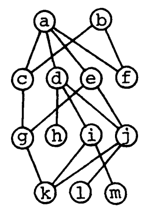
The drawing of a hierarchy is called a map. In a map all the vertices belonging to the ith level V_i are arranged on the ith line of the n horizontal real lines which are numbered from the top to the bottom (Rule a)). The coordinate

TABLE I
SUMMARY TABLE OF ANALYSIS AND DISCUSSIONS IN INTRODUCTION

	functions	readability elements	methods	
			theoretical	heuristic
basic rules of drawing	constraints for algorithms	Elements (A), (C_1)		
STEP I	transformation into proper hierarchy			
STEP II	reduction of the number of crossings	Element (B)	Penalty Minimization (PM) method	Barycentric (BC) method
STEP III	improvement of horizontal positions of vertices	Elements (C_2), (D), (E)	Quadratic Programming (QP) method	Priority (PR) method
STEP IV	display on graphic terminals			

$x(v_i)$ of vertex v_i on a real line is called a horizontal position. Every edge is drawn with a straight line (Rule b)).

By specifying the rules of drawing, the problem is significantly simplified since the number of crossings of a proper hierarchy is determined by orders σ of vertices in each level and vertical coordinates of vertices are fixed according to the level where each vertex is included. In the subsequent sections, discussions are restricted to proper hierarchies. Accordingly, a proper hierarchy is called a hierarchy for simplicity.



(a)

$$M^{(1)} = \begin{array}{c|cccc} & c & d & e & f \\ \hline a & 1 & 1 & 1 & 1 \\ b & 1 & 0 & 0 & 1 \end{array} \qquad M^{(2)} = \begin{array}{c|cccc} & g & h & i & j \\ \hline c & 1 & 0 & 0 & 0 \\ d & 0 & 1 & 1 & 1 \\ e & 1 & 0 & 0 & 1 \\ f & 0 & 0 & 0 & 0 \end{array} \qquad M^{(3)} = \begin{array}{c|ccc} & k & l & m \\ \hline g & 1 & 0 & 0 \\ h & 0 & 0 & 0 \\ i & 1 & 0 & 1 \\ j & 1 & 1 & 0 \end{array}$$

(b)

Fig. 3. (a) Four-level hierarchy. (b) Its matrix realization.

B. Matrix Realization of n-Level Hierarchies

For an n-level hierarchy $G=(V,E,n,\sigma)$, the matrix realization of G is defined as follows.

1) A matrix $M^{(i)}=M(\sigma_i,\sigma_{i+1})$ is a $|V_i|\times|V_{i+1}|$ matrix whose rows and columns are ordered according to σ_i and σ_{i+1}, respectively.
2) Let $\sigma_i=v_1\cdots v_k\cdots v_{|V_i|}$ and $\sigma_{i+1}=w_1\cdots w_l\cdots w_{|V_{i+1}|}$. Then the (v_k,w_l) element of $M^{(i)}$, denoted by $m_{kl}^{(i)}$, is given by
$$m_{kl}^{(i)}=\begin{cases}1 & \text{if } (v_k,w_l)\in E_i\\ 0 & \text{otherwise,}\end{cases} \tag{1}$$
where $M^{(i)}$ is called an interconnection matrix.
3) A matrix realization $\boldsymbol{g}$ of G is given by the formula
$$\boldsymbol{g}(\sigma_1,\cdots,\sigma_n)=M^{(1)}\cdots M^{(n-1)} \qquad (=\boldsymbol{g}(V,E,n,\sigma)) \tag{2}$$

In the following discussions $\boldsymbol{g}$ will be limited to a standard form [3, p. 514] where there do not exist identical rows or columns. An example of a hierarchy and its matrix realization is shown in Fig. 3.

C. The Number of Crossings of n-Level Hierarchies

Formulas to calculate the number of crossings on the map of n-level hierarchies have been given by Warfield [3]. In the ith interconnection matrix $M^{(i)}=M(\sigma_i,\sigma_{i+1})$ of $\boldsymbol{g}$, let $\sigma_i=v_1\cdots v_j\cdots v_k\cdots v_{|V_i|}$. Further, let the row vector of $M^{(i)}$ corresponding to a vertex $v\in V_i$ be denoted by $r(v)$, then the number of crossings $k(r(v_j),r(v_k))$ produced by the ordered pair of row vectors $(r(v_j),r(v_k))$ is given by the formula
$$k(r(v_j),r(v_k))=\sum_{\alpha=1}^{q-1}\sum_{\beta=\alpha+1}^{q} m_{j\beta}^{(i)}m_{k\alpha}^{(i)} \tag{3}$$
where $q=|V_{i+1}|$. Consequently the formula
$$K(M^{(i)})=\sum_{j=1}^{p-1}\sum_{k=j+1}^{p} k(r(v_j),r(v_k)) = \sum_{j=1}^{p-1}\sum_{k=j+1}^{p}\left(\sum_{\alpha=1}^{q-1}\sum_{\beta=\alpha+1}^{q} m_{j\beta}^{(i)}m_{k\alpha}^{(i)}\right) \tag{4}$$
gives the number of crossings of $M^{(i)}$ where $p=|V_i|$. Similar expressions can be obtained starting from ordered pairs of column vectors. From (4), the total number $K(\boldsymbol{g})$ of crossings of $\boldsymbol{g}$ is given by
$$K(\boldsymbol{g})=K(M^{(1)})+\cdots+K(M^{(n-1)}). \tag{5}$$

D. Connectivity

In an n-level hierarchy $G=(V,E,n,\sigma)$ if $\sigma_i=v_1^i\cdots v_k^i\cdots v_{|V_i|}^i$, $i=1,\cdots,n$, then the upper connectivity C_{ik}^U of vertex v_k^i and the lower connectivity C_{ik}^L of vertex v_k^i are defined by the formulas
$$C_{ik}^U=\sum_{j=1}^{|V_{i-1}|} m_{jk}^{(i-1)}, \qquad k=1,\cdots,|V_i|,\quad i=2,\cdots,n \tag{6}$$
$$C_{ik}^L=\sum_{l=1}^{|V_{i+1}|} m_{kl}^{(i)}, \qquad k=1,\cdots,|V_i|,\quad i=1,\cdots,n-1. \tag{7}$$

E. Barycenters

Here three expressions of barycenters are defined. First of all, barycenter D_y of a binary vector $y=(y_1,\cdots,y_m)$ to be used in the definition of a specific generating matrix in Section III is given by

$$D_y = \sum_{j=1}^{m} j \cdot y_j \Big/ \sum_{j=1}^{m} y_j. \tag{8}$$

Next the formulas

$$B_{ik}^{R} = \sum_{l=1}^{q} l \cdot m_{kl}^{(i)} \Big/ \sum_{l=1}^{q} m_{kl}^{(i)}, \qquad k=1,\cdots,p \ (=|V_i|) \tag{9}$$

$$B_{il}^{C} = \sum_{k=1}^{p} k \cdot m_{kl}^{(i)} \Big/ \sum_{k=1}^{p} m_{kl}^{(i)}, \qquad l=1,\cdots,q \ (=|V_{i+1}|) \tag{10}$$

give row and column barycenters of a binary interconnection matrix $M^{(i)}=(m_{kl}^{(i)})$ respectively, which will be used in BC method.

Finally, upper and lower barycenters B_{ik}^{U}, B_{ik}^{L} of upper and lower vertices connected to the kth vertex v_k^i in the ith level are defined by

$$B_{ik}^{U} = \sum_{j=1}^{p} x\left(v_j^{i-1}\right) m_{jk}^{(i-1)} \Big/ C_{ik}^{U}, \qquad k=1,\cdots,|V_i| \tag{11}$$

$$B_{ik}^{L} = \sum_{l=1}^{q} x\left(v_l^{i+1}\right) m_{kl}^{(i)} \Big/ C_{ik}^{L}, \qquad k=1,\cdots,|V_i| \tag{12}$$

where $p=|V_{i-1}|$ and $q=|V_{i+1}|$, and $x(v)$ is the horizontal position of a vertex v. These barycenters will be used in the PR method.

III. Reduction of the Number of Crossings (Step II)

Let S_i be a set of all possible orders σ_i in an n-level hierarchy $G=(V,E,n,\sigma)$ and $S=S_1\times\cdots\times S_n$, then the problem minimizing the number of crossings of the n-level hierarchy is stated as follows:

$$\text{minimize}\{K(\mathbf{g}(\sigma))|\sigma\in S\} \tag{13}$$

according to formulas (2)–(5). This problem, however, is combinatorial in nature, therefore, it is difficult to obtain the optimum solution when the size of the problem is not small.

In this section algorithms are considered for two-level hierarchies, and then these algorithms are extended to cases for n-level hierarchies.

A. Algorithms for Two-Level Hierarchies

1) Penalty minimization method (PM method): Consider a two-level hierarchy whose matrix realization $\mathbf{g}(\sigma_1,\sigma_2)=M$ is given. We describe an algorithm to determine the set of row (or column) orders that minimize the number of crossings when the column (or row) order is fixed. In Warfield [3, p. 512] this set is called P_j (or Q_i)-set and is obtained by referring to generating matrices for the cases where there are not more than five vertices in each level. The PM method can be used to obtain P_j (or Q_i)-set for general two-level hierarchies where there are more than five vertices. It should also be noted that by applying this algorithm to the permutation of row and column orders in turn repeatedly, the number of crossings can be monotonically reduced.

Recall that $k(r(u),r(v))$ denotes the number of crossings produced by the ordered row vectors corresponding to the vertices u, $v\in V_1$ when u precedes v in σ_1, while $k(r(v),r(u))$ denotes the corresponding number when v precedes u in σ_1. Formula (4) shows that the number of crossings of the two-level hierarchy is a sum of $p(p-1)/2$ terms which are the numbers of crossings produced by the ordered pairs of row vectors corresponding to all the two-combinations of p vertices in V_1. According to the order of u and v in σ_1, one alternative of $k(r(u),r(v))$ and $k(r(v),r(u))$ is taken for each two-combination $\{u,v\}$ in the summation of (4). Therefore, when we permute σ_1, some of $p(p-1)/2$ terms are replaced from one alternative to another according to the permuted order. We consider the problem to determine the optimal order of vertices as the problem to determine the set of orders in all the two-combinations with some restrictions as stated subsequently.

The sum of the numbers of crossings which correspond to all the two-combinations of p vertices is minimized if every term takes its minimum, i.e., $\min\{k(r(u),r(v)),k(r(v),r(u))\}$. However there might not be any order of vertices which corresponds to the sum so obtained. For this reason, some terms might have to take nonminima. Let us set $p(u,v)=k(r(v),r(u))-k(r(u),r(v))$ if $k(r(v),r(u))>k(r(u),r(v))$, where the number $p(u,v)$ is called a penalty. Now our problem is to determine appropriate orders in all the two-combinations of vertices such that the sum of penalties induced by nonminimal terms is minimized under the condition that there exists a corresponding order of vertices.

In order to describe the algorithm a penalty digraph is introduced.

Penalty digraph: Let $\mathbf{g}(\sigma_1,\sigma_2)=M$ be given. Then we consider the case where the row order σ_1 is permuted under the fixed column order σ_2. The penalty digraph H is defined by

$$H=(W,F,p)$$

where

$W=V_1$

$F=\{(u,v)\in W\times W|k(r(u),r(v))<k(r(v),r(u))\}$

$N=\{1,2,\cdots\}$

$p:F\to N$ is defined by $p(u,v)=k(r(v),r(u))-k(r(u),r(v))$ for each $(u,v)\in F$.

This penalty digraph H is used for the reordering of σ_1 as described subsequently.

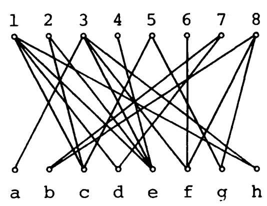
Algorithm: If H is cycle-free, the row order to minimize the number of crossings can be determined easily. However, if $|V_2|>5$, H is not always cycle-free. This fact directly follows from Warfield's discussions of generating matrices [3]. In order to determine an order of all the vertices, we must eliminate all the cycles in H by reversing directions of appropriate edges in F. The order that minimizes the number of crossings can be obtained by minimizing the sum of penalties corresponding to the edges of which directions are reversed. The algorithm is given as follows.

Step 1: The penalty digraph H is obtained.
Step 2: All the strongly connected components are found in H (see [4]).
Step 3: In every component which contains more than two vertices, cycles are eliminated by reversing directions of edges so that the sum of penalties is minimized. It should be noted that this step is identical with the "minimum feedback arc set" problem in graph theory (see [5]).
Step 4: The order of V_1 is determined according to the revised digraph of H which is obtained in Step 3.

Example: Consider a two-level hierarchy $G=(V, E, 2, \sigma)$, where

$$V=V_1 \cup V_2,$$
$$V_1=\{1,2,3,4,5,6,7,8\},$$
$$V_2=\{a,b,c,d,e,f,g,h\},$$
$$E=\{(1,c),(1,d),(1,e),(1,h),(2,c),(2,e),(3,a),(3,e),(3,f),(3,h),(4,e),(5,c),(5,g),(6,f),(7,b),(7,d),(8,b),(8,f),(8,g)\},$$
$$\sigma=(\sigma_1,\sigma_2),$$
$$\sigma_1=12345678,$$
$$\sigma_2=abcdefgh.$$

A map of G and a matrix realization $\boldsymbol{g}$ of G are as follows:

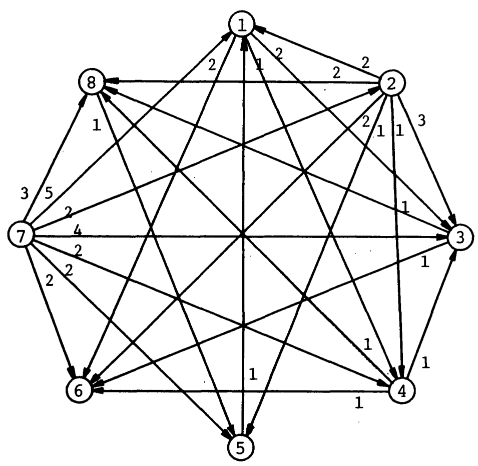


$g =$

	a	b	c	d	e	f	g	h
1	0	0	1	1	1	0	0	1
2	0	0	1	0	1	0	0	0
3	1	0	0	0	1	1	0	1
4	0	0	0	0	1	0	0	0
5	0	0	1	0	0	0	1	0
6	0	0	0	0	0	1	0	0
7	0	1	0	1	0	0	0	0
8	0	1	0	0	0	1	1	0

The number of crossings $K(\boldsymbol{g})$ is 69.

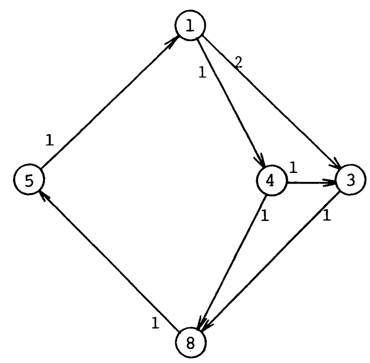
Step 1: We calculate the penalty digraph H for V_1. A drawing of the penalty digraph H is as follows where $u \xrightarrow{q} v$ denotes the penalty relation $p(u,v)=q$.



Step 2: We obtain all the strongly connected components in H

$$\{1,3,4,5,8\} \quad \{2\} \quad \{6\} \quad \{7\}. \qquad \text{(see [4])}$$

The first strongly connected component is illustrated below.



Step 3: We eliminate all the cycles in the strongly connected component $\{1,3,4,5,8\}$ (see [5]). The arc adjacency matrix A and $\bar{A}=A+I$ (I is the identity matrix) are

$$A=\begin{array}{c} \\ 1\\3\\4\\5\\8 \end{array}\begin{array}{c} \begin{array}{ccccc} 1 & 3 & 4 & 5 & 8 \end{array} \\ \begin{pmatrix} 0 & u_1^2 & u_2 & 0 & 0\\ 0 & 0 & 0 & 0 & u_3\\ 0 & u_4 & 0 & 0 & u_5\\ u_6 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & u_7 & 0 \end{pmatrix}\end{array}, \quad \bar{A}=\begin{pmatrix} 1 & u_1^2 & u_2 & 0 & 0\\ 0 & 1 & 0 & 0 & u_3\\ 0 & u_4 & 1 & 0 & u_5\\ u_6 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & u_7 & 1 \end{pmatrix},$$

where u_1^2 has penalty 2 and the other variables have penalty 1. We next calculate the permanent expansion $\overset{+}{|}\,\bar{A}\,\overset{+}{|}$ of $\bar{A}$:

$$\overset{+}{|}\,\bar{A}\,\overset{+}{|} = u_1^2u_3u_6u_7 + u_2u_5u_6u_7 + u_2u_3u_4u_6u_7.$$

Replacing u_i with Boolean variables e_i and interchanging product and sum, we obtain the following a-set function:

$$f=\left(e_1^2+e_3+e_6+e_7\right)\left(e_2+e_5+e_6+e_7\right)\cdot\left(e_2+e_3+e_4+e_6+e_7\right).$$

By applying the absorption law we have

$$f=e_1^2e_4e_5+e_1^2e_2+e_3e_5+e_2e_3+e_6+e_7.$$

Therefore, we get minimum feedback arc set solutions $\{e_6\}$ or $\{e_7\}$ with penalty 1.

Step 4: We get the optimal orders of the vertices in V_1 by reversing the edge $5 \to 1$ (or $8 \to 5$) in H corresponding to e_6 (or e_7). For e_6 we have the following three orderings:

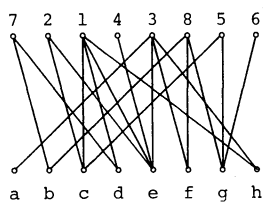
$$\begin{aligned}\sigma_1 = 7\quad 2\quad 1\quad 4\quad 3\quad 8\quad 5\quad 6\\ 7\quad 2\quad 1\quad 4\quad 3\quad 8\quad 6\quad 5\\ 7\quad 2\quad 1\quad 4\quad 3\quad 6\quad 8\quad 5,\end{aligned}$$

since there are no penalty relations for (6, 8) and (6, 5). For e_7 we obtain

$$\begin{aligned}\sigma_1 = 7\quad 2\quad 5\quad 1\quad 4\quad 3\quad 8\quad 6\\ 7\quad 2\quad 5\quad 1\quad 4\quad 3\quad 6\quad 8.\end{aligned}$$

Consequently, we have the following map and matrix realization for a solution

$$\sigma_1 = 7\quad 2\quad 1\quad 4\quad 3\quad 8\quad 5\quad 6.$$



	a	b	c	d	e	f	g	h
7	0	1	0	1	0	0	0	0
2	0	0	1	0	1	0	0	0
1	0	0	1	1	1	0	0	1
4	0	0	0	0	1	0	0	0
3	1	0	0	0	1	1	0	1
8	0	1	0	0	0	1	1	0
5	0	0	1	0	0	0	1	0
6	0	0	0	0	0	1	0	0

The number of crossings is now reduced to 48.

2) Barycentric method (BC method): We consider a heuristic method for the reordering of the row order $\sigma_1 = v_1v_2\cdots v_{|V_1|}$ to reduce the number of crossings under the fixed column order σ_2 in $M(\sigma_1,\sigma_2)$. The essential idea of this method is to reorder σ_1 according to the row barycenters $B_{1k}^R, k=1,\cdots,|V_1|$ which are calculated by (9). These are ordered from the smallest to the largest, that is $B_{1s_1}^R \leqslant B_{1s_2}^R \leqslant \cdots \leqslant B_{1s_{|V_1|}}^R$, where if there are sets of rows of which barycenters are equal, the original order is preserved. Then if the reordering of σ_1 is denoted by σ_1', we have $\sigma_1' = v_{s_1}v_{s_2}\cdots v_{s_{|V_1|}}$. This operation which transforms $M(\sigma_1,\sigma_2)$ to the reordered matrix $M(\sigma_1',\sigma_2)$, is called "barycentric ordering" of rows and is denoted by β_R, i.e., $M(\sigma_1',\sigma_2)=\beta_R(M(\sigma_1,\sigma_2))$. The barycentric ordering of columns is similarly defined and is denoted by β_C. We can reduce the number of crossings by repeating the barycentric ordering of rows and columns in turn.

Effectiveness of the barycentric ordering: A matrix called a generating matrix, which is very useful in considering the ordering problem, has been defined by Warfield [3]. To evaluate the effectiveness of the barycentric ordering we introduce a specific generating matrix called a barycenter-ordered generating matrix (BGM). This is a square matrix T_m of dimensions 2^m-1 by 2^m-1 with the following properties.

a) Rows and columns of T_m have the same index set which consists of all possible binary vectors, except zero vector, of dimension m.
b) The index set consisting of 2^m-1 binary vectors is ordered from the smallest to the largest according to their barycenters calculated by (8). This ordered index set is denoted by $r_1, r_2, \cdots, r_{2^m-1}$.
c) Denoting the (r_i,r_j) element of T_m by t_{ij}, $t_{ij}=$ "+" and $t_{ji}=$ "−" if $k(r_i,r_j)<k(r_j,r_i)$, and $t_{ij}=t_{ji}=$ "0" if $k(r_i,r_j)=k(r_j,r_i)$ (see Fig. 4).

It should be recognized that when the (i, j) element in the upper triangular part of BGM is "−", the barycentric ordering is inconsistent with the order of a pair of vectors (r_i, r_j) that minimizes the number of crossings produced by the pair. The numbers of "−", "+", and "0" elements in the upper triangular part, except diagonal elements, of BGM, N_m^-, N_m^+, and N_m^0, for $m=2,\cdots,11$ have been counted, which are shown in Table II.

From these numbers, the inconsistency rate δ_m for $m=2,\cdots,11$ is calculated by

$$\delta_m = \frac{N_m^-}{N_m^- + N_m^+ + N_m^0}\times 100 = \frac{N_m^-}{(2^m-1)(2^{m-1}-1)}\times 100 \text{ percent} \tag{14}$$

and those for $m=20$ and 30 are obtained with extrapolation. It is remarkable that the inconsistency rates are significantly low, which tells us that the barycentric ordering is expected to be sufficiently effective for the reduction of the number of crossings.

Algorithm: The algorithm consists of two phases, Phase 1 and Phase 2. Phase 2 uses Phase 1 as a subalgorithm. In Phase 1 the barycentric ordering of rows or columns is repeated in turn. In this operation for Phase 1, the orders of the rows (or columns) which have equal barycenters are preserved. However, we may be able to reduce the number of crossings by changing these orders. Therefore, Phase 2 is introduced to carry out the operation to reorder rows (or columns) with equal barycenters just after the execution of Phase 1. In Phase 2 the orders of these rows (or columns) are reversed in each set (this operation is called "reversion" of rows (or columns) and is denoted by $\boldsymbol{R}_R(M)$ (or $\boldsymbol{R}_C(M)$), and then Phase 1 is reexecuted starting with columns (or rows). This operation has been empirically found to be effective.

The algorithm is given as follows: Let M_0 be a matrix realization of a two-level hierarchy and let M^* be a solution matrix, and K^* the number of crossings of M^*.

	2	3	1	D_{r_i}	r_i
2	0	+	+	1.0	(10)
3	-	0	+	1.5	(11)
1	-	-	0	2.0	(01)

(a)

	4	6	2	5	7	3	1	D_{r_i}	r_i
4	0	+	+	+	+	+	+	1.0	(100)
6	-	0	+	+	+	+	+	1.5	(110)
2	-	-	0	0	0	+	+	2.0	(010)
5	-	-	0	0	0	+	+	2.0	(101)
7	-	-	0	0	0	+	+	2.0	(111)
3	-	-	-	-	-	0	+	2.5	(011)
1	-	-	-	-	-	-	0	3.0	(001)

(b)

	8	12	4	10	14	13	9	6	15	11	2	5	7	3	1	D_{r_i}	r_i
8	0	+	+	+	+	+	+	+	+	+	+	+	+	+	+	1.0	(1000)
12	-	0	+	+	+	+	+	+	+	+	+	+	+	+	+	1.5	(1100)
4	-	-	0	0	0	0	0	+	+	+	+	+	+	+	+	2.0	(0100)
10	-	-	0	0	0	+	+	+	+	+	+	+	+	+	+	2.0	(1010)
14	-	-	0	0	0	+	+	+	+	+	+	+	+	+	+	2.0	(1110)
13	-	-	0	-	-	0	+	+	+	+	+	+	+	+	+	2.3	(1101)
9	-	-	0	-	-	-	0	0	0	0	0	+	+	+	+	2.5	(1001)
6	-	-	-	-	-	-	0	0	0	+	+	+	+	+	+	2.5	(0110)
15	-	-	-	-	-	-	0	0	0	0	+	+	+	+	+	2.5	(1111)
11	-	-	-	-	-	-	0	-	0	0	0	+	+	+	+	2.7	(1011)
2	-	-	-	-	-	-	0	-	-	0	0	0	0	+	+	3.0	(0010)
5	-	-	-	-	-	-	-	-	-	-	0	0	0	+	+	3.0	(0101)
7	-	-	-	-	-	-	-	-	-	-	0	0	0	+	+	3.0	(0111)
3	-	-	-	-	-	-	-	-	-	-	-	-	-	0	+	3.5	(0011)
1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	4.0	(0001)

(c)

	16	24	8	20	28	26	18	12	30	25	22	29	4	17	10	21	14	27	31	23	19	13	9	6	15	11	2	5	7	3	1	D_{r_i}	r_i
16	0	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	1.0	(10000)
24	-	0	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	1.5	(11000)
8	-	-	0	0	0	0	0	+	+	+	+	+	+	0	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	2.0	(01000)
20	-	-	0	0	0	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	2.0	(10100)
28	-	-	0	0	0	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	2.0	(11100)
26	-	-	0	-	-	0	0	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	2.33	(11010)
18	-	-	0	-	-	0	0	0	0	+	0	+	0	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	2.5	(10010)
12	-	-	-	-	-	-	0	0	0	-	+	+	+	0	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	2.5	(01100)
30	-	-	-	-	-	-	0	0	0	0	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	2.5	(11110)
25	-	-	-	-	-	-	-	+	0	0	0	+	+	0	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	2.67	(11001)
22	-	-	-	-	-	-	0	-	-	0	0	0	0	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	2.67	(10110)
29	-	-	-	-	-	-	-	0	-	-	0	0	+	0	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	2.75	(11101)
4	-	-	-	-	-	-	0	-	-	-	0	-	0	0	0	0	0	0	0	+	+	0	0	+	+	+	+	+	+	+	+	3.0	(00100)
17	-	-	0	-	-	-	-	0	-	0	-	0	0	0	0	0	0	0	0	0	0	+	+	0	+	+	0	+	+	+	+	3.0	(10001)
10	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	+	+	+	+	+	+	+	+	+	+	+	+	3.0	(01010)
21	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	+	+	+	+	+	+	+	+	+	+	+	+	3.0	(10101)
14	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	+	+	+	+	+	+	+	+	+	+	+	+	3.0	(01110)
27	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	+	+	+	+	+	+	+	+	+	+	+	+	3.0	(11011)
31	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	+	+	+	+	+	+	+	+	+	+	+	+	3.0	(11111)
23	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	0	+	0	+	0	+	+	+	+	+	+	+	3.25	(10111)
19	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	0	0	+	-	0	+	0	+	+	+	+	3.33	(10011)
13	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	0	0	0	0	+	+	+	+	+	+	+	+	3.33	(01101)
9	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	-	-	0	0	0	0	0	0	+	+	+	+	3.5	(01001)
6	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	0	+	-	0	0	0	+	+	+	+	+	+	3.5	(00110)
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	0	0	0	+	+	+	+	+	+	3.5	(01111)
11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	0	0	+	+	+	+	3.67	(01011)
2	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	-	-	0	-	0	-	-	0	0	0	0	+	+	4.0	(00010)
5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	+	+	4.0	(00101)
7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	+	+	4.0	(00111)
3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	+	4.5	(00011)
1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	5.0	(00001)

(d)

Fig. 4. Barycenter-ordered generating matrices (BGM's). (a) BGM T_2. (b) BGM T_3. (c) BGM T_4. (d) BGM T_5.

TABLE II
NUMBERS OF "−," "+," AND "0" ELEMENTS IN UPPER TRIANGULAR PART OF BGM AND INCONSISTENCY RATES

m	N^-_m	N^+_m	N^0_m	total	δ_m(%)
2	0	3	0	3	0
3	0	18	3	21	0
4	0	90	15	105	0
5	2	400	63	465	0.43
6	17	1707	229	1953	0.87
7	114	7082	805	8001	1.42
8	591	29030	2764	32385	1.82
9	2825	117991	9489	130305	2.17
10	12607	477467	32679	522753	2.41
11	54434	1926100	113547	2094081	2.60
20					4.39
30					5.55

Phase 1:

Step 1: $M^* := M_0$, $K^* := K(M_0)$
Step 2: $M_1 := \beta_R(M_0)$
Step 3: If $K(M_1) < K^*$ then $M^* := M_1$ and $K^* := K(M_1)$.
Step 4: $M_2 := \beta_C(M_1)$
Step 5: If $K(M_2) < K^*$ then $M^* := M_2$ and $K^* := K(M_2)$.
Step 6: If M_0 and M_2 are equal or if the number of iterations in Phase 1 attains an initially given number, Phase 1 is terminated and go to Step 7. Otherwise, go to Step 2.

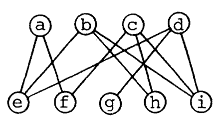
Phase 2:

Step 7: $M_3 := \boldsymbol{R}_R(M_2)$
Step 8: When the barycenters of columns of M_3 are not arranged in an increasing order, go to Step 11 with $M_0 := M_3$; otherwise, go to Step 9.
Step 9: $M_4 := \boldsymbol{R}_C(M_3)$
Step 10: When the barycenters of rows of M_4 are not arranged in an increasing order, go to Step 11 with $M_0 := M_4$; otherwise, terminate calculations.
Step 11: If the number of iterations in Phase 2 attains an initially given number, then terminate calculations; otherwise, go to Step 2.

In the algorithms above, if reordering of columns β_C is selected as the initial operation for Phase 1, β_R in Step 2 is replaced by β_C and β_C in Step 4 is replaced by β_R. Similarly for Phase 2, $\boldsymbol{R}_C$ is used instead of $\boldsymbol{R}_R$ in Step 7 and $\boldsymbol{R}_R$ is used instead of $\boldsymbol{R}_C$ in Step 9.

Example: Consider a two-level hierarchy G_0 of which map is as follows:

G_0



1) M_0 is the initial interconnection matrix of G_0:

$$M_0 = \begin{array}{c|ccccc|c} & e & f & g & h & i & B^R_{ik} \\ \hline a & 1 & 1 & 0 & 0 & 0 & 1.5 \\ b & 1 & 0 & 0 & 1 & 1 & 3.3 \\ c & 0 & 1 & 0 & 1 & 1 & 3.7 \\ d & 1 & 0 & 1 & 0 & 1 & 3.0 \\ \hline B^C_{il} & 2.3 & 2.0 & 4.0 & 2.5 & 3.0 & \end{array} \quad K(M_0) = 14.$$

2) By reordering rows b, c, d of M_0, M_1 is obtained:

$$M_1 = \begin{array}{c|ccccc|c} & e & f & g & h & i & \\ \hline a & 1 & 1 & 0 & 0 & 0 & 1.5 \\ d & 1 & 0 & 1 & 0 & 1 & 3.0 \\ b & 1 & 0 & 0 & 1 & 1 & 3.3 \\ c & 0 & 1 & 0 & 1 & 1 & 3.7 \\ \hline & 2.0 & 2.5 & 2.0 & 3.5 & 3.0 & \end{array} \quad K(M_1) = 11.$$

3) By reordering columns f, g, h, i of M_1, M_2 is obtained (end of Phase 1):

$$M_2 = \begin{array}{c|ccccc|c} & e & g & f & i & h & \\ \hline a & 1 & 0 & 1 & 0 & 0 & 2.0 \\ d & 1 & 1 & 0 & 1 & 0 & 2.3 \\ b & 1 & 0 & 0 & 1 & 1 & 3.3 \\ c & 0 & 0 & 1 & 1 & 1 & 4.0 \\ \hline & 2.0 & 2.0 & 2.5 & 3.0 & 3.5 & \end{array} \quad K(M_2) = 9.$$

4) By reversing the order of columns e, g of M_2, M_3 is obtained (start of Phase 2):

$$M_3 = \begin{array}{c|ccccc|c} & g & e & f & i & h & \\ \hline a & 0 & 1 & 1 & 0 & 0 & 2.5 \\ d & 1 & 1 & 0 & 1 & 0 & 2.3 \\ b & 0 & 1 & 0 & 1 & 1 & 3.7 \\ c & 0 & 0 & 1 & 1 & 1 & 4.0 \\ \hline & 2.0 & 2.0 & 2.5 & 3.0 & 3.5 & \end{array} \quad K(M_3) = 9.$$

5) By reordering rows a, d of M_3, M_4 is obtained:

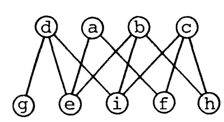
$$M_4 = \begin{array}{c|ccccc|c} & g & e & f & i & h & \\ \hline d & 1 & 1 & 0 & 1 & 0 & 2.3 \\ a & 0 & 1 & 1 & 0 & 0 & 2.5 \\ b & 0 & 1 & 0 & 1 & 1 & 3.7 \\ c & 0 & 0 & 1 & 1 & 1 & 4.0 \\ \hline & 1.0 & 2.0 & 3.0 & 2.7 & 3.5 & \end{array} \quad K(M_4) = 8.$$

6) By reordering columns f, i of M_4, M_5 is obtained (end of Phase 2):

$$M_5 = \begin{array}{c|ccccc|c} & g & e & i & f & h & \\ \hline d & 1 & 1 & 1 & 0 & 0 & 2.0 \\ a & 0 & 1 & 0 & 1 & 0 & 3.0 \\ b & 0 & 1 & 1 & 0 & 1 & 3.3 \\ c & 0 & 0 & 1 & 1 & 1 & 4.0 \\ \hline & 1.0 & 2.0 & 2.7 & 3.0 & 3.5 & \end{array} \quad K(M_5) = 7 = \min$$

A map of the hierarchy G_5 which corresponds to M_5 is as follows:

G_5



B. Algorithm for n-Level Hierarchies

We consider an n-level hierarchy $\mathbf{g}(\sigma_1, \cdots, \sigma_n) = M(\sigma_1, \sigma_2) M(\sigma_2, \sigma_3) \cdots M(\sigma_{n-1}, \sigma_n)$ where $n \geqslant 3$. The problem (13) is rewritten as follows:

$$\text{minimize} \quad f(z)=z_1+z_2+\cdots+z_{n-1}$$

subject to

$$z_i=K(M(\sigma_i,\sigma_{i+1})), \qquad i=1,\cdots,n-1$$

$$\sigma_i \in S_i, \qquad i=1,\cdots,n.$$

Note that the function $f(z)$ is so-called separable. This fact might justify a decomposition into two-level subproblems (P_i): $\min\{K(M(\sigma_i,\sigma_{i+1}))|\sigma_1 \in S_i,\ \sigma_{i+1} \in S_{i+1}\}$, $i=1,\cdots,n-1$. But we cannot solve subproblem (P_i) separately since problem (P_i) is dependent on problems (P_{i-1}) and (P_{i+1}). Therefore, we introduce other subproblems $(P_i^D(\sigma_i^*))$: $\min\{K(M(\sigma_i^*,\sigma_{i+1}))|\sigma_{i+1} \in S_{i+1}\}$ or $(P_i^U(\sigma_{i+1}^*))$: $\min\{K(M(\sigma_i,\sigma_{i+1}^*))|\sigma_i \in S_i\}$, where σ_i^* or σ_{i+1}^* are given orders, $i=1,\cdots,n-1$. Subproblems $(P_i^D(\sigma_i^*))$ or $(P_i^U(\sigma_{i+1}^*))$ are weaker than subproblems (P_i), but by taking a solution σ_{i+1}^* of $(P_i^D(\sigma_i^*))$ as an order σ_{i+1}^* in $(P_{i+1}^D(\sigma_{i+1}^*))$ sequentially, the number of crossings is reduced. Consequently, our algorithm is stated as follows.

Step 1: An initial order $\sigma_1^* \in S_1$ is given. Let $i:=1$.
Step 2: Solve the subproblem $(P_i^D(\sigma_i^*))$ and let a solution of $(P_i^D(\sigma_i^*))$ be denoted by σ_{i+1}^*.
Step 3: If $i<n-1$, then $i:=i+1$ and go to Step 2. If $i=n-1$, then go to Step 4.
Step 4: Solve the subproblem $(P_i^U(\sigma_{i+1}^*))$ and let a solution of $(P_i^U(\sigma_{i+1}^*))$ be denoted by σ_i^*.
Step 5: If $i>1$, then $i:=i-1$, and go to Step 4. If $i=1$, then STOP.

The procedure to solve $(P_1^D(\sigma_1^*)),\cdots,(P_{n-1}^D(\sigma_{n-1}^*))$ (or $(P_{n-1}^U(\sigma_n^*)),\cdots,(P_1^U(\sigma_2^*))$) in turn is called DOWN (or UP) procedure. The whole procedure above is called DOWN-UP procedure. Then, we iterate DOWN-UP procedures until at least one of the following conditions is attained:

a) the same matrix realization appears periodically,
b) the number of times of iterations reaches an initially given number.

On the other hand, we can start with an initial order σ_n^* instead of σ_1^* and this procedure is called UP-DOWN procedure. The iteration is performed similarly as for DOWN-UP procedure.

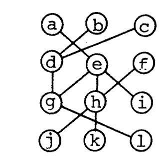
To solve a subproblem $(P_i^D(\sigma_i^*))$ or $(P_i^U(\sigma_{i+1}^*))$, PM method or BC method can be applied. If BC method is applied, the above procedure of iteration is called Phase 1. It has been observed that in most cases the period of the termination condition a) is one. Particularly, this condition is accomplished when all rows and columns are arranged in the increasing order of barycenters when BC method is applied. When there exist sets of rows (or columns) which have equal barycenters just after the execution of Phase 1, Phase 2 is executed, where a procedure similar to Phase 2 of the two-level algorithm is applied.

Phase 2 consists of two procedures, DOWN and UP. In DOWN (or UP) procedure, the order of columns (or rows) in level i with equal barycenters is reversed and Phase 1 starts with DOWN (or UP) procedure, where i runs 2 through n (or $n-1$ through 1). When Phase 1 has been terminated with DOWN (or UP) procedure, Phase 2 starts with DOWN (or UP) procedure. The whole procedure is called the n-level BC method.

We considered another procedure for n-level hierarchies. In this procedure the barycenter of a vertex in level i is calculated by regarding connected vertices in both level $i-1$ and level $i+1$, and the same DOWN-UP (or UP-DOWN) iterations were adopted (in the n-level BC method either level $i-1$ or level $i+1$ is considered). However the results of the experiments for several examples were all worse than the results executed by the n-level BC method. The reason of the difference of performance might be that, in the above procedure, vertices whose orders were not yet improved are considered in calculating barycenters, whereas in the n-level BC method only vertices whose orders were already improved are considered.

Example: Consider a four-level hierarchy G_0 of which a map follows:

G_0



1) $\mathbf{g}_0$ is the initial matrix realization of G_0:

$$\mathbf{g}_0 = \begin{array}{c|ccc|c|ccc|c|ccc|}
\multicolumn{1}{c}{} & \multicolumn{3}{c}{M^{(1)}} & \multicolumn{1}{c}{} & \multicolumn{3}{c}{M^{(2)}} & \multicolumn{1}{c}{} & \multicolumn{3}{c}{M^{(3)}} \\
\multicolumn{1}{c}{} & d & e & \multicolumn{1}{c}{f} & \multicolumn{1}{c}{} & g & h & \multicolumn{1}{c}{i} & \multicolumn{1}{c}{} & j & k & \multicolumn{1}{c}{l} \\
\cline{2-4}\cline{6-8}\cline{10-12}
a & 0 & 1 & 0 & d & 1 & 0 & 0 & g & 0 & 0 & 1 \\
b & 1 & 0 & 0 & e & 1 & 1 & 1 & h & 1 & 1 & 0 \\
c & 1 & 0 & 0 & f & 0 & 1 & 0 & i & 0 & 0 & 0 \\
\cline{2-4}\cline{6-8}\cline{10-12}
\multicolumn{1}{c}{} & 2.5 & 1.0 & \multicolumn{1}{c}{} & \multicolumn{8}{c}{}
\end{array} \qquad K(\mathbf{g}_0)=5.$$

2) By reordering vertices d, e (columns d, e of $M^{(1)}$ and rows d, e of $M^{(2)}$) in $\mathbf{g}_0$, $\mathbf{g}_1$ is obtained (start of Phase 1-DOWN);

$$\mathbf{g}_1 = \begin{array}{c|ccc|c|ccc|c|ccc|} & e & d & f & & g & h & i & & j & k & l \\ \cline{2-4}\cline{6-8}\cline{10-12} a & 1 & 0 & 0 & e & 1 & 1 & 1 & g & 0 & 0 & 1 \\ b & 0 & 1 & 0 & d & 1 & 0 & 0 & h & 1 & 1 & 0 \\ c & 0 & 1 & 0 & f & 0 & 1 & 0 & i & 0 & 0 & 0 \\ \cline{2-4}\cline{6-8}\cline{10-12} & 1.0 & 2.5 & & & 1.5 & 2.0 & 1.0 & & & & \end{array} \qquad K(\mathbf{g}_1)=5.$$

3) By reordering vertices g, h, i in $\mathbf{g}_1$, $\mathbf{g}_2$ is obtained:

$$\mathbf{g}_2 = \begin{array}{c|ccc|c|ccc|c|ccc|} & e & d & f & & i & g & h & & j & k & l \\ \cline{2-4}\cline{6-8}\cline{10-12} a & 1 & 0 & 0 & e & 1 & 1 & 1 & i & 0 & 0 & 0 \\ b & 0 & 1 & 0 & d & 0 & 1 & 0 & g & 0 & 0 & 1 \\ c & 0 & 1 & 0 & f & 0 & 0 & 1 & h & 1 & 1 & 0 \\ \cline{2-4}\cline{6-8}\cline{10-12} & & & & & 1.0 & 1.5 & 2.0 & & 3.0 & 3.0 & 2.0 \end{array} \qquad K(\mathbf{g}_2)=3.$$

4) By reordering vertices j, k, l in $\mathbf{g}_2$, $\mathbf{g}_3$ is obtained (end of Phase 1):

$$\mathbf{g}_3 = \begin{array}{c|ccc|cc|ccc|cc|ccc|c} & e & d & f & & & i & g & h & & & l & j & k & \\ \cline{2-4}\cline{7-9}\cline{12-14} a & 1 & 0 & 0 & 1.0 & e & 1 & 1 & 1 & 2.0 & i & 0 & 0 & 0 & \\ b & 0 & 1 & 0 & 2.0 & d & 0 & 1 & 0 & 2.0 & g & 1 & 0 & 0 & 1.0 \\ c & 0 & 1 & 0 & 2.0 & f & 0 & 0 & 1 & 3.0 & h & 0 & 1 & 1 & 2.5 \\ \cline{2-4}\cline{7-9}\cline{12-14} & 1.0 & 2.5 & & & & 1.0 & 1.5 & 2.0 & & & 2.0 & 3.0 & 3.0 & \end{array} \qquad K(\mathbf{g}_3)=1.$$

5) Although vertices j, k of which barycenters are equal are reversed in $\mathbf{g}_3$, same matrices are obtained ($\mathbf{g}_3'$) (Phase 2–DOWN).

6) By reversing vertices e, d in $\mathbf{g}_3'$, $\mathbf{g}_4$ is obtained (start of Phase 2–UP).

$$\mathbf{g}_4 = \begin{array}{c|ccc|cc|ccc|cc|ccc|c} & d & e & f & & & i & g & h & & & l & k & j & \\ \cline{2-4}\cline{7-9}\cline{12-14} a & 0 & 1 & 0 & 2.0 & d & 0 & 1 & 0 & 2.0 & i & 0 & 0 & 0 & \\ b & 1 & 0 & 0 & 1.0 & e & 1 & 1 & 1 & 2.0 & g & 1 & 0 & 0 & 1.0 \\ c & 1 & 0 & 0 & 1.0 & f & 0 & 0 & 1 & 3.0 & h & 0 & 1 & 1 & 2.5 \\ \cline{2-4}\cline{7-9}\cline{12-14} \end{array} \qquad K(\mathbf{g}_4)=3.$$

7) By reordering rows a, b, c in $\mathbf{g}_4$ (Phase 1–UP), $\mathbf{g}_5$ is obtained:

$$\mathbf{g}_5 = \begin{array}{c|ccc|c|ccc|c|ccc|} & d & e & f & & i & g & h & & l & k & j \\ \cline{2-4}\cline{6-8}\cline{10-12} b & 1 & 0 & 0 & d & 0 & 1 & 0 & i & 0 & 0 & 0 \\ c & 1 & 0 & 0 & e & 1 & 1 & 1 & g & 1 & 0 & 0 \\ a & 0 & 1 & 0 & f & 0 & 0 & 1 & h & 0 & 1 & 1 \\ \cline{2-4}\cline{6-8}\cline{10-12} & 1.5 & 3.0 & & & 2.0 & 1.5 & 2.5 & & & & \end{array} \qquad K(\mathbf{g}_5)=1.$$

8) By reordering columns i, g in $\mathbf{g}_5$ (Phase 1–DOWN), $\mathbf{g}_6$ is obtained (end of Phase 2):

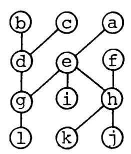
$$\mathbf{g}_6 = \begin{array}{c|ccc|cc|ccc|cc|ccc|c} & d & e & f & & & g & i & h & & & l & k & j & \\ \cline{2-4}\cline{7-9}\cline{12-14} b & 1 & 0 & 0 & 1.0 & d & 1 & 0 & 0 & 1.0 & g & 1 & 0 & 0 & 1.0 \\ c & 1 & 0 & 0 & 1.0 & e & 1 & 1 & 1 & 2.0 & i & 0 & 0 & 0 & \\ a & 0 & 1 & 0 & 2.0 & f & 0 & 0 & 1 & 3.0 & h & 0 & 1 & 1 & 2.5 \\ \cline{2-4}\cline{7-9}\cline{12-14} & 1.5 & 3.0 & & & & 1.5 & 2.0 & 2.5 & & & 1 & 3.0 & 3.0 & \end{array} \qquad K(\mathbf{g}_6)=0.$$

TABLE III
RESULTS OF PERFORMANCE TESTS OF n-LEVEL BC METHOD ($N_i = |V_i|$)

No.	n	$(N_1,\ldots,N_n)$	π	$\overline{K_{ini}}$	$\overline{K_{ini}-K_{min}}$	$\overline{K_{ini}-K_1}$	$\overline{K_{ini}-K_2}$	$\overline{I_1}$	$\overline{I_2}$
1	2	(3,4)	0.3	3.90	3.68	3.51	3.63	0.944	0.983
2	2	(3,4)	0.5	5.70	4.33	4.12	4.30	0.924	0.991
3	2	(5,6)	0.3	19.8	16.9	15.9	16.7	0.934	0.990
4	2	(5,6)	0.5	39.4	23.8	22.7	23.5	0.940	0.988
5	2	(5,6)	0.7	74.8	29.9	28.7	29.4	0.945	0.979
6	4	(4,4,4,4)	0.3	13.1	11.1	10.6	10.9	0.942	0.983
7	4	(4,4,4,4)	0.5	28.4	17.6	16.4	17.3	0.914	0.969
8	4	(4,4,4,4)	0.7	52.4	20.8	20.0	20.4	0.935	0.980

A map of hierarchy G_6 which corresponds to g_6 is as follows.

G_6



C. Performance Tests of BC Method

In order to evaluate the performance of the BC method, we chose eight different types of hierarchies, in the length n of levels, the numbers of vertices $|V_i|$, and a parameter π specifying density of edges. For each type of hierarchy a hundred matrix realizations were generated by using random numbers generated by a computer. The number of edges generated was controlled by π, which is the expected value of $|E|/|E^c|$, where $|E|$ is the number of edges of a generated hierarchy and $|E^c|$ is that of the complete multilevel digraph having the same configuration of nodes $(N_1,\cdots,N_n)$. For each one of the hundred matrices the initial number of crossings K_{ini} and the minimum number of crossings K_{min} were calculated. K_{min} was obtained by a combinatorial (exhaustive) search. Further, by applying the BC method to each one of the hundred hierarchies, two types of performance indices were calculated by

$$I_1 = (K_{ini} - K_1)/(K_{ini} - K_{min}) \tag{15}$$

$$I_2 = (K_{ini} - K_2)/(K_{ini} - K_{min}) \tag{16}$$

where K_1 and K_2 mean the numbers of crossings just after the execution of Phase 1 and Phase 2, respectively. Results of the tests are presented in Table III, which shows that the BC method is significantly effective as expected from the inconsistency rate in Table II. In Table III the overbar (e.g., $\bar{I}_2$) denotes the mean value of the corresponding variables (e.g., I_2) of the hundred randomly generated hierarchies.

IV. DETERMINATION OF HORIZONTAL POSITIONS OF VERTICES (STEP III)

A. Quadratic Programming Layout Method (QP Method)

An n-level hierarchy $G=(V,E,n,\sigma^*)$ is given, where σ^* is the orders of vertices which are determined by the foregoing step to reduce the number of crossings. Let $\sigma_i^* = v_1^i \cdots v_k^i \cdots v_{|V_i|}^i$, $i=1,\cdots,n$, then we denote the horizontal position of the kth vertex on the ith level by $x(v_k^i)$ or simply x_k^i. The problem to determine horizontal positions of vertices to improve the readability elements (C_2), (D), and (E) is formulated as the following quadratic programming.

Objective function:

$$\text{minimize } f = cf_1 + (1-c)f_2, \qquad 0 < c \leqslant 1 \tag{17}$$

a) "close" layout of vertices connected to each other (readability Element D):

$$f_1 = \sum_{i=1}^{n-1}\left\{\sum_{k=1}^{|V_i|}\sum_{l=1}^{|V_{i+1}|}\left(x_k^i - x_l^{i+1}\right)^2 m_{kl}^{(i)}\right\} \tag{18}$$

b) "balanced" layout of edges coming into or outgoing from a vertex (readability Element E):

$$f_2 = \sum_{i=1}^{n-1}\left\{\sum_{k=1}^{|V_i|}\left(x_k^i - B_{ik}^L\right)^2 u\left(C_{ik}^L - 1\right)\right\} + \sum_{i=2}^{n}\left\{\sum_{k=1}^{|V_i|}\left(x_k^i - B_{ik}^U\right)^2 u\left(C_{ik}^U - 1\right)\right\} \tag{19}$$

where

$$u(z) = \begin{cases} 1 & \text{if } z > 0 \\ 0 & \text{if } z \leqslant 0. \end{cases}$$

Constraints:

a) Fixed orders (readability Element B):

$$x_1^1 = x_0 \ (= \text{given constant}) \tag{20}$$

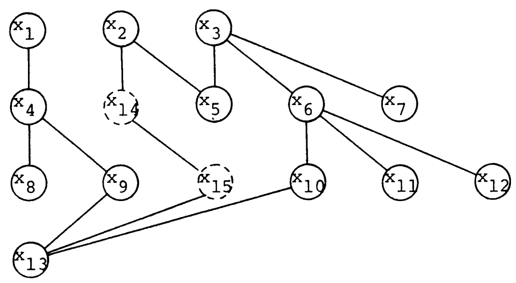
$$x_{k+1}^i - x_k^i \geqslant a \ (= \text{given positive number}), \quad k = 1,\cdots,|V_i|-1,\ i=1,\cdots,n; \tag{21}$$

b) "straightness" of long span edges (readability Element C_2)

$$x_k^i = x_l^{i+1} \qquad \text{if } m_{kl}^{(i)} = 1$$

and both v_k^i and v_l^{i+1} are dummy vertices, $k=1,\cdots,|V_i|$, $l=1,\cdots,|V_{i+1}|$, $i=2,\cdots,n-2$. (22)

Example: Consider the following four-level hierarchy where x_{14} and x_{15} denote dummy vertices.



Objective function:

$$\text{minimize } f = cf_1 + (1-c)f_2$$

a) $f_1 = (x_1-x_4)^2 + (x_2-x_{14})^2 + (x_2-x_5)^2 + (x_3-x_5)^2 + (x_3-x_6)^2 + (x_3-x_4)^2 + (x_4-x_8)^2 + (x_4-x_9)^2 + (x_{14}-x_{15})^2 + (x_6-x_{10})^2 + (x_6-x_{11})^2 + (x_6-x_{12})^2 + (x_9-x_{13})^2 + (x_{15}-x_{13})^2 + (x_{10}-x_{13})^2$

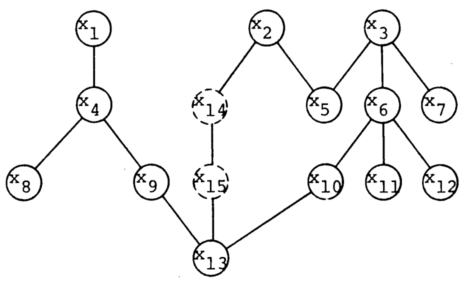
b) $f_2 = (x_2-(x_{14}+x_5)/2)^2 + (x_3-(x_5+x_6+x_7)/3)^2 + (x_4-(x_8+x_9)/2)^2 + (x_6-(x_{10}+x_{11}+x_{12})/3)^2 + (x_5-(x_2+x_3)/2)^2 + (x_{13}-(x_9+x_{15}+x_{10})/3)^2$

Constraints:

a) $x_1 = 100\ (=x_0)$ $x_2 - x_1 \geqslant 1\ (=a)$ $x_3 - x_2 \geqslant 1$
$x_{14} - x_4 \geqslant 1$ $x_5 - x_{14} \geqslant 1$ $x_6 - x_5 \geqslant 1$
$x_7 - x_6 \geqslant 1$ $x_9 - x_8 \geqslant 1$ $x_{15} - x_9 \geqslant 1$
$x_{10} - x_{15} \geqslant 1$ $x_{11} - x_{10} \geqslant 1$ $x_{12} - x_{11} \geqslant 1$

b) $x_{14} = x_{15}$

Solution (c=0.5):



B. Priority Layout Method (PR Method)

This method is a heuristic method which is developed to reduce the computing cost needed to obtain horizontal positions of vertices that realize a readable layout of a given n-level hierarchy. The fundamental idea for this method is similar to that for the multilevel BC method, i.e., a decomposition of the problem and "sequential" application of level operations (called improvements of horizontal positions). In case of the multilevel BC method, the reordering of vertices is performed according to barycenters of the vertices, while in this method the improvement is carried out according to "priority numbers" given to vertices. When we let $x^i = (x_1^i, \cdots, x_{|V_i|}^i)$, the algorithm is outlined in the following.

1) Initial values of horizontal positions of vertices in each level are given by

$$x_k^i = x_0 + k, \qquad k = 1, \cdots, |V_i|, \quad i = 1, \cdots, n \qquad (23)$$

where x_0 is a given integer.

2) Positions of vertices in each level are improved in the order of levels $2, \cdots, n, n-1, \cdots, 1, t, \cdots, n$ where t is a given integer ($2 \leqslant t \leqslant n-1$). The improvements of the positions of vertices in levels $2, \cdots, n$ and $t, \cdots, n$ are called DOWN procedures, while those for levels $n-1, \cdots, 1$ are called UP procedures.

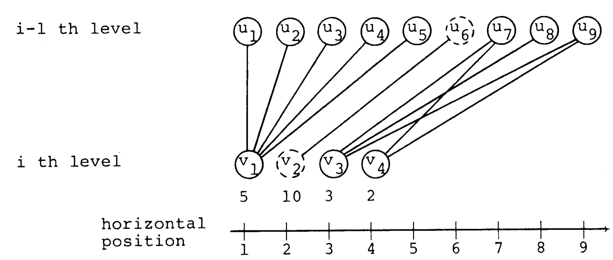
3) Positions of vertices in each level are determined one by one according to their priority numbers. The highest priority number, an integer more than the maximum of connectivities of all vertices, is given to dummy vertices to improve the readability Element C_2. Priority numbers given to the other vertices are the connectivities of the vertices calculated by (6) or (7) to improve the readability Element E through the operations described in 4).

4) The principle to improve the position of a vertex is to minimize the difference between the present position of the vertex and the upper (or lower) barycenter given by (11) (or (12)), of the vertex in DOWN (or UP) procedure under the following conditions.

a) The position of the vertex should be integer and can not be equal to the positions of other vertices in the same level.
b) The order of vertices of each level should be preserved (readability Element B).
c) Positions of only vertices of which priorities are less than the priority of the vertex can be changed, where the distance displaced should be as small as possible (readability Element D).

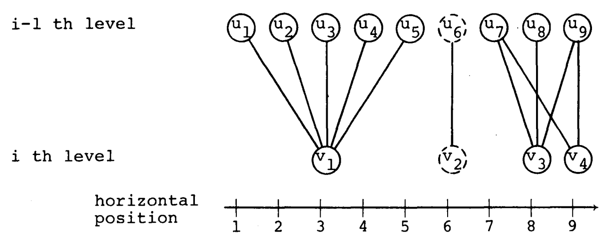
The details of the algorithm are described in Sugiyama *et al.* [6].

Example: In order to illustrate the process of improving the positions of vertices by the PR method, consider the following two-level subhierarchy which contains the $i-1$th and the ith levels of an n-level hierarchy where u_6 and v_2 are dummy vertices.



Now we improve the positions of vertices of the ith level in the DOWN procedure. The priority numbers of vertices v_1, v_2, v_3, and v_4 are 5, 10, 3, and 2, respectively, where 10 means the highest priority number which is given to the dummy vertices. First, the position of v_2 is improved, since it has the highest priority number. By this improvement, the positions $\{x(v_1), \cdots, x(v_4)\}$ of vertices v_1, v_2, v_3, and

v_4 are displaced from $\{1,2,3,4\}$ to $\{1,6,7,8\}$ so that v_2 is connected to u_6 by a vertical line. Next, the position of v_1 that has the second highest priority number is improved and we have $\{3,6,7,8\}$. Then, after the improvement of the position of v_3, we have $\{3,6,8,9\}$. Finally, although the desirable position of v_4, B_{i4}^U, is 8, the position of v_4 can not be displaced because the priority number of v_3 is higher than the priority number of v_4. As a result, we have the following two-level subhierarchy.



V. Applications

The multilevel BC method is applied to the following five practical hierarchies that have already been obtained by the following quoted references.

1) the six-level hierarchy which appeared in Warfield [3],
2) a 14-level hierarchy which represents reference relations among a set of papers concerning the theory of successive process of statistical inferences [7],
3) a two-level hierarchy which also represents reference relations among a set of papers in CAD/CAM [8],
4) an 11-level hierarchy which is used to analyze the structure of interactions among subjects in a curriculum of a university [9], and
5) a 19-level hierarchy illustrating a structure of the propagation of events when an earthquake occurs [10].

Results of the applications are summarized in Table IV. Table IV shows the number of crossings remaining and the computation time on FACOM M190 needed just after the execution of Phase 1 and Phase 2. The reduction of crossing numbers is significant and the central processing unit (CPU) time (s) is small for these practical examples.

Presented in Fig. 5 are drawings of the nine-level hierarchy which appeared in Malone [11] and represent the interdependence of obstacles to investment in the Columbus, Ohio, central business district. The BC method has been applied to the initial map (a), then the QP method has been applied to the resulting hierarchy to obtain the readable maps (b), (c), and (d). The parameter c in the objective function was varied to obtain the three maps; $c = 1$ for (b), $c = 0.5$ for (c), and $c = 0.05$ for (d). Fig. 5(e) was obtained by applying the BC method and the PR method. The CPU time needed for these computations on FACOM M190 was 1.1 s for the BC method, and for Fig. 5

(b)	32 s	QP method
(c)	30 s	QP method
(d)	1108 s	QP method
(e)	0.1 s	PR method

TABLE IV
Results of Practical Applications

No.	n	$\Sigma\|V_i\|$	number of edges	K_{ini}	K_{min}	Phase 1 K_1	Phase 1 CPU	Phase 2 K_2	Phase 2 CPU
1)	6	21	41	46	21	21	0.07	21	0.17
2)	14	152	159	151	?	31	2.3	16	7.7
3)	2	76	99	2067	?	446	1.22	446	1.34
4)	11	38	42	24	3	3	0.1	3	0.37
5)	19	189	227	404	?	82	1.03	75	15.2

where the CPU time for PR method is considerably less than that for QP method. Only "close" layout of the vertices is considered in Fig. 5(b) ($c=1$ in the objective function), hence the map is compact. More weight is given to "balanced" layout of edges as c decreases, hence the map expands to Fig. 5(d). From a practical viewpoint, it may be said that the PR method is preferable to the QP method.

Drawings of the 14-level hierarchy (Example 2) are presented in Fig. 6, where (a) is the initial map and (b) is the readable map resulting after the BC method and the PR method were applied. For each of the readability elements satisfactory results were achieved.

We considered condensation of a digraph in Step I to obtain a hierarchy, but other schemes can be applied to generate a hierarchy for a digraph with cycles. We present an example of another scheme to generate a hierarchy. In the example the effective representation algorithms were applied to the diagram that illustrates the dependence of the variables of system dynamics. The diagram in Fig. 7(a) shows the schematic relations of the variables that have been used to simulate the future dynamics of the world [12, pp. 20, 21]. A two-dimensional diagram for system dynamics contains cycles, but a cycle contains at least one level variable (e.g., variables denoted by rectangular vertices in Fig. 7(a)) due to a constraint imposed by a Dynamo compiler. Therefore, the cycles can be eliminated by drawing the level variables twice as described below.

The digraph can then be represented by a hierarchical map. An application of the BC method and the PR method to the map resulted in the drawing in Fig. 7(b). The bottom vertices (solid line square vertices) in Fig. 7(b) denote the level variables at some time, and the top vertices (broken line square vertices) are the same variables after the unit-time of the simulation. Since the vertices at the top and bottom levels denote the same variables, it is desirable to locate the vertices in the same order. The definition of recurrent hierarchies in the appendix enabled the application of the BC method and the PR method to the hierarchy.

Note that Fig. 7(b) exhibits the order by which auxiliary variables (circular vertices) are determined at each time step of the simulation. Each variable can be determined only after all the lower level variables have been determined that are connected to the variable.

It may be difficult to determine which figure, 7(a) or 7(b), is more effective in representing the simulation model. Considering that (a) was manually drawn to communicate the simulation model to human readers, and that (b) was automatically drawn by a computer, we might justify that (b) is a reasonably good representation of the model.

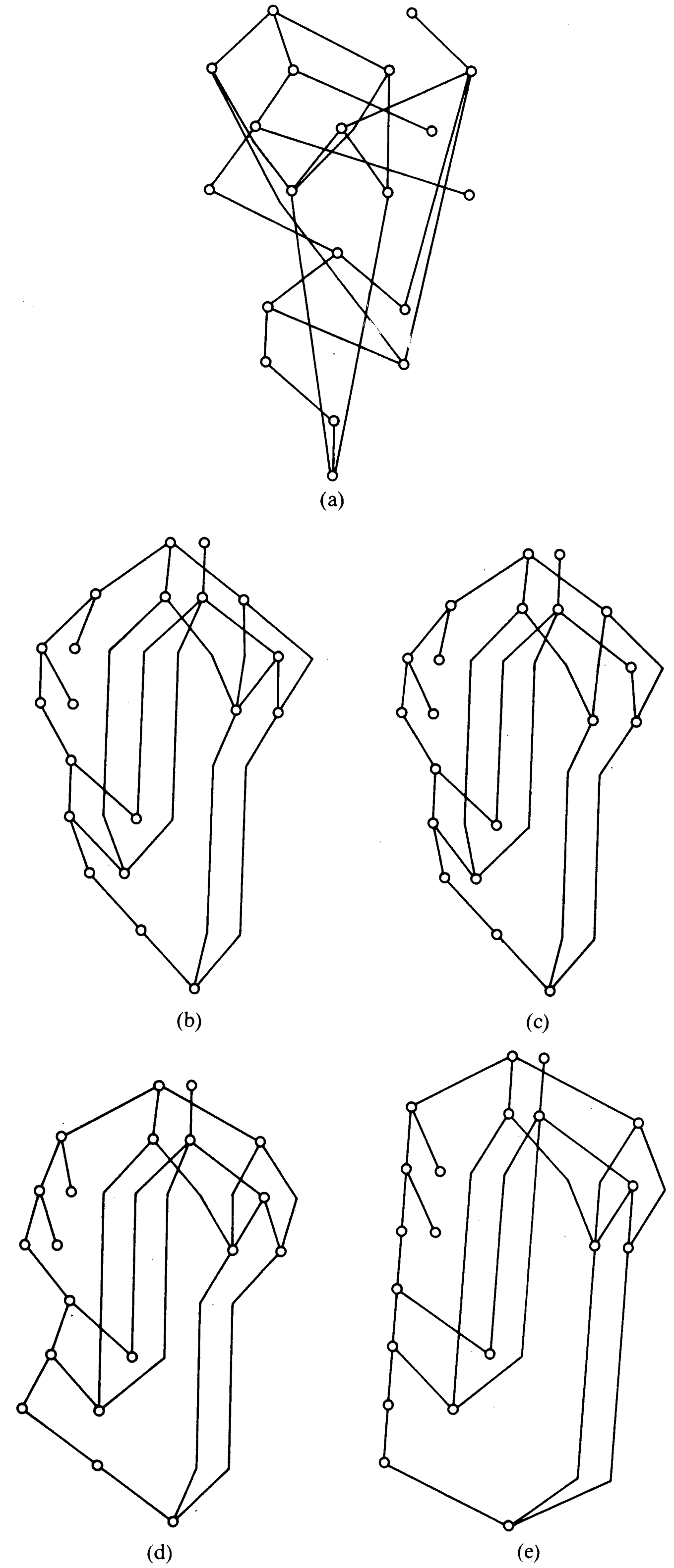


Fig. 5. Maps of the nine-level hierarchy [11]. (a) Initial. (b) BC + QP methods ($c = 1.0$). (c) BC + QP methods ($c = 0.5$). (d) BC + QP methods ($c = 0.05$). (e) BC + PR methods.

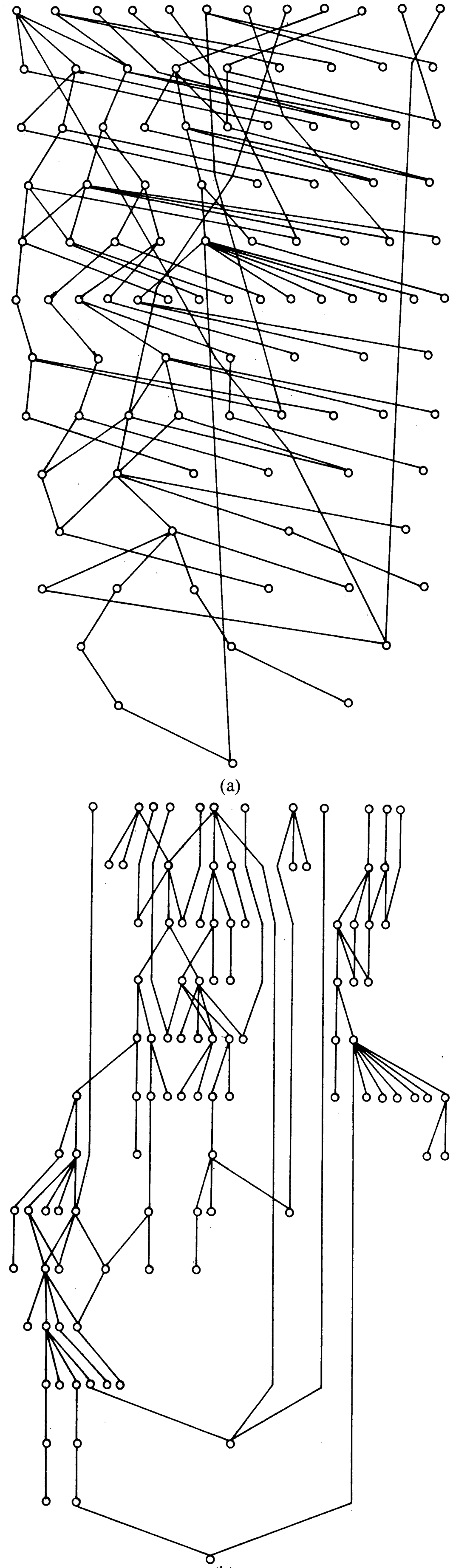
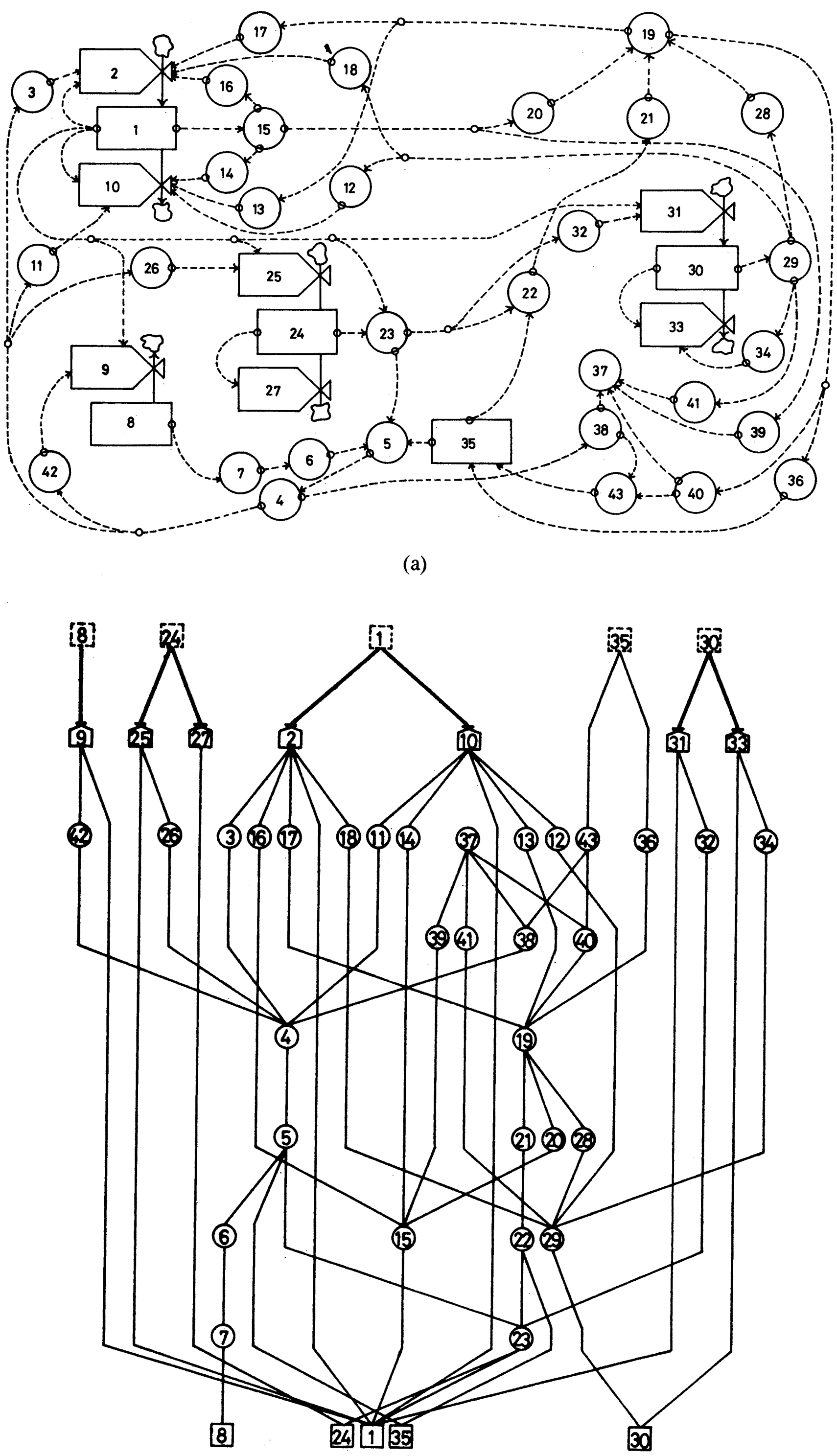


Fig. 6. Maps of a 14-level hierarchy (Example 2). (a) Initial. (b) Result (BC + PR method).



(a)

(b)

Fig. 7. (a) Diagram of the world model (Forrester [12]). (b) Hierarchical representations of the world model.

VI. Concluding Remarks

We have discussed common aspects of the readability of maps and have developed theoretical algorithms (PM, QP methods) and heuristic algorithms (BC, PR methods). We have identified five readability elements. The procedure to generate maps attaining these elements consists of two steps. In the first main step the number of crossings of a hierarchy is reduced by the PM or BC method, and in the second main step, other readability elements such as "straightness" of long span edges, "close" layout of vertices, and "balanced" layout of edges are realized by the QP or PR method. The whole procedure from Step I to Step IV has been coded by Fortran on FACOM M190, where the heuristic algorithms, BC and PR methods, have been employed for Step II and Step III, respectively. The theoretical algorithms, PM and QP methods, have also been coded and have been applied to solve examples of small size due to computing cost. By developing the theoretical algorithms, we can recognize the nature of the problem to generate the readable representations of hierarchies. On the other hand, by developing the heuristic algorithms, we can enlarge the size of problems we can deal with. We have already applied BC and PR methods to a problem which has more than 500 vertices.

Furthermore, the BC method has been extended for the case when vertices in each level of hierarchies are partitioned into several subsets and all vertices in a subset must be arranged on a line adjacently. QP and PR methods have also been extended for the case when each vertex is drawn with a square whose size is determined by the importance of the vertex or by the number of characters labeling the vertex. As a further extension of the algorithms, important cases are when the area to draw a hierarchy is limited; for example, the area on an XY plotter is limited horizontally or the area on a graphic display is limited both horizontally and vertically. Finally, the following studies are envisaged for future research:

(1) studies of relationships between readability and level assignments [13] in acyclic digraphs,
(2) developments of methods for readable drawing of digraphs with cycles, and
(3) developments of methods for readable drawing of undirected graphs.

APPENDIX
RECURRENT HIERARCHIES

The definition of an n-level recurrent hierarchy is slightly different from that of an n-level hierarchy: a directed graph (V, E) is called an n-level recurrent hierarchy if

1) V is partitioned into n subsets, that is
$$V=V_1\cup V_2\cup\cdots\cup V_n\qquad (V_i\cap V_j=\varnothing,\ i\neq j);$$
2) E is partitioned into n subsets, that is
$$E=E_1\cup E_2\cup\cdots\cup E_n\qquad (E_i\cap E_j=\varnothing,\ i\neq j).$$
where $E_i\subset V_i\times V_{i+1}$, $i=1,\cdots,n-1$, and $E_n\subset V_n\times V_1$;
3) an order σ_i of all the vertices of V_i is given for each i, $i=1,\cdots,n$.

ACKNOWLEDGMENT

The authors would like to acknowledge the continuing guidance and encouragement of Dr. Tosio Kitagawa, the director of their institute.

REFERENCES

[1] J. N. Warfield, *Societal Systems*. New York: Wiley, 1976.
[2] S. E. Elmaghraby, *Activity Networks: Project Planning and Control by Network Models*. New York: Wiley, 1977.
[3] J. N. Warfield, "Crossing theory and hierarchy mapping," *IEEE Trans. Syst. Man, Cybern.*, vol. SMC-7, no. 7, pp. 505–523, 1977.
[4] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*. Reading, MA: Addison-Wesley, 1974.
[5] A. Lempel and I. Cederbaum, "Minimum feedback arc and vertex sets of a directed graph," *IEEE Trans. Circuit Theory*, vol. CT-13, no. 4, pp. 399–403, 1966.
[6] K. Sugiyama, S. Tagawa, and M. Toda, "Effective representations of hierarchical structures," International Institute for Advanced Study of Social Information Science, Fujitsu Limited, Research Rep. No. 8, pp. 1–29, 1979.
[7] T. Kitagawa and K. Sugiyama, "A method for informative approaches to analysing scientific research activities," *Proc. 19th Annu. Convention Inform. Processing Soc., Japan*, 3I-10, pp. 863–864, 1978, (in Japanese).
[8] N. Okino and S. Teshima, "Automatic clustering of publications; Organization of CAD/CAM literatures as an example," Rep., Proj. No. 220902 for Ministry of Education, Japan, 1978.
[9] Special Research Project, "Formation Process of Information System and Organization of Scientific Information," research report on the system science oriented approaches to systems utilizing scientific informations sponsored by Ministry of Education, Japan, pp. 5–19, 1978 (in Japanese).
[10] K. Sugiyama, T. Hiramatsu, and Y. Shimazu, "Toward development of earthquake disaster relational model," *Proc. Intern. Conf. Cybern., Soc.*, Nov. 1978, Tokyo and Kyoto, pp. 788–793.
[11] D. W. Malone, "An introduction to the application of interpretive structural modeling," *Proc. IEEE*, vol. 63, no. 3, pp. 397–404, 1975.
[12] J. W. Forrester, *World Dynamics*. Cambridge, MA: Wright-Allen, 1971.
[13] F. Harary, R. Z. Norman, and D. Cartwright, *Structural Models: An Introduction to the Theory of Directed Graphs*. New York: Wiley, 1965.