コンピュータ・グラフィクス　1－1
（1981. 6. 23）

INTERACTIVE FACILITIES FOR GRAPHICAL EDITING AND ANALYSIS SYSTEM FOR NETWORK SYSTEM

Kiyoshi ITOH, Katsumi MURAMATSU, Mabito MATSUI and Shigemichi SUZUKI

Laboratory of Systems Engineering, Department of Mechanical Engineering, Faculty of Science and Technology, Sophia University, Tokyo, 102, Japan

1. Introduction

Network analysis is powerful in building formal network models of various computational systems or social systems and obtaining their functions and performance exactly because of simplicity of binary relations in networks where entities and their relationships are enumerated and then nodes and arcs are assigned to them, respectively. For example, in modeling a concurrent processing system as a network model such as a Petri Net, an entity (i.e., node) represents a component or a condition which occurs an event and a relationship (i.e., arc) represents a connection link between components or conditions; in a transportation analysis problem, a node and an arc represent a flow branching point and a flow link, respectively.

Traditionally, it takes lots of efforts in network analysis because tables or maps of nodes and arcs were manually punched in the numerical form on a batch of many cards and analyzed by computer programs; because network analysis results were shown in the numerical form with ambiguous readability; because a batch of cards representing a numerical network were changed or replaced with tremendous effort even when a network must be slightly modified. Effective network analysis activity, satisfying the system requirements based on consideration for the above traditional manner with high cost, requires semiautomatic or computer aided man-machine facilities with interactive editing, analysis, and result viewing functions.

We have developed Graphical Editing and Analysis system for Network System (Geans) on DEC PDP-11 / 34 (RT-11 Fortran IV with graphic utility extension) minicomputer with a graphic display which enables us to edit or modify various network representations and analyze their qualitative and quantitative property, smoothly and interactively with the use of graphics and data abstraction [Itoh 1980]. Fig.1 shows the system structure of Geans. Geans is composed of a network editing system with effective use of a graphic display, a network data management system with systematic use of network data abstraction, a network analysis system with construction of shared library routines and specific object problem oriented routines, and a node / arc shape definition system.

Positions of nodes and arcs are specified by light pen positioning on a graphic display. Their parameters are specified by light pen hitting and TTY typing. Deletion of nodes or arcs, i.e., derivation of partial networks or subnetworks from original ones, is also possible by light pen specifying of nodes or arcs. Geans adopts windowing function by which large-scale networks (in maximum limit with about 65,000X65,000 coordinates) can be edited or modified in a 10X10 coordinate movable window, and viewing function by which any large-scale network can be viewed on a reduced scale of the window.

Shape definition system is provided to define shapes of nodes and arcs arbitrarily in user's choice. In the current version, we can define them at source statement level in corresponding to object systems or problems to be analyzed such as transportation systems, Petri Nets, PERT / CPM charts, and build object system and problem oriented networks with the use of them.

Network information is treated as so-called abstract data type, while it is really organized as a unique data structure of table form of node and arcs. Implementers of network editors

PDP 11/34 with Graphic Display
- Shape Def. System for Node and Arc
  - object problem/system oriented
- Network Data Management System
  - abstract data type oriented
- Network Editing System
  - powerful "windowing function" and "viewing function"
- Network Property Analysis
  - connectivity, etc.
- Network Analysis
  - Petri Net Analysis
  - PERT/CPM Planning
  - Transportation Problem
  - etc.

Fi.1 Geans System Structure

or analyzers can construct their programs with the use of well-defined cluster operations for abstract data type of networks without complete knowledge about internal organization of network data, i.e., well-defined standard subroutine calls which point out operation types such as creation or deletion of nodes and arcs, and their parameter referencing or updating.

Qualitative analysis programs are provided for network connectivity checking, tie set / cut set detection, etc.. Quantitative analysis programs are provided for each of object problems or systems. Programs detecting shortest paths, maximum flow and minimum cost flow are provided for transportation systems. Marking simulator controlling movement of tokens in Petri Nets is provided. These analysis programs were easily implemented with the use of abstract data types of networks. Their results are shown on a graphic display with the use of editing functions of Geans.

2. Network Editing Facility

Geans editing facility is available for constructing various network models except for the fact that different shapes of nodes and arcs are used in each network model.

The display screen whose size is a 1024X1024 addressable dot square is partitioned into a network display area and a menu display area.

The network display area is called a "window" whose physical size is fixed in a 540X540 dot square and which is located at a fixed position on the screen. A node is located in two dimensional area where its both X coordinate and Y coordinate are ranged from -32768 to 32767. This two dimensional area is called a "virtual display area". The window can move to arbitrary positions in a large-scale virtual display area. Geans adopts "windowing function" by which a large-scale network can be edited or modified incrementally as altering this window and inspecting a partial network in the window. The position of the movable window is specified by defining the lower left coordinates of the window. Initially or in default, they are (1,1).

When a node is created, its coordinates are automatically calculated in terms of the current lower left coordinates of the window and recorded. A node or an arc is said to be visual if it exits inside the current window. When the lower left coordinates of the window are altered, the current version of Geans determines whether each of nodes is visual or not and displays visual nodes and arcs whose both start node and end node are visual, i.e., Geans displays a partial network which is composed of visual nodes and arcs connected between the visual nodes. In the next version of Geans, all visual nodes and all visual arcs including ones whose either start node or end node is visual can be displayed on the screen.

The logical size of the window can be altered on an arbitrary scale from 10X10 coordinates to 1,000X1,000 coordinates. This function is called "viewing function" by which large-scale networks in a varied scale jointly with the windowing function. The physical size of a node or an arc is automatically altered in proportion to the inverse of logical size of the window. The default value of the logical size of the window is 10X10 coordinates.

Fig.2 shows the relationships of a virtual display area and a window and their size and coordinates. Fig.3 - 5 shows the viewing function, where Fig.3 shows the original network, Fig.4 shows a reduced network in a quarter size of the original window and Fig.5 shows subsequent network editing.

Menu is displayed on the menu display area in order to select operation modes necessary for network editing and analysis functions depicted in Fig.6 where "main mode menu" represents two editing modes with the use of a graphic display and a TTY, hard-copy mode with the use of a plotter, file I / O mode, and analysis mode, and "graphics mode menu" represents editing mode with the use of a graphic display in performing editing subtasks to be mentioned below.

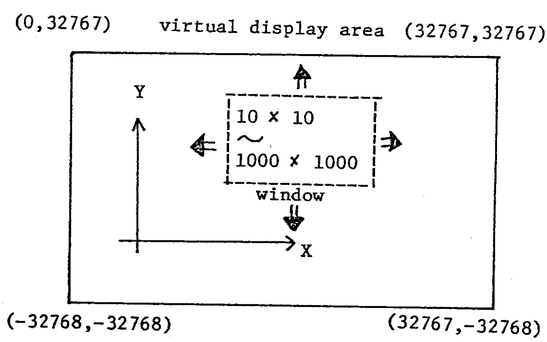


Fig.2 Virtual Display Area and Window

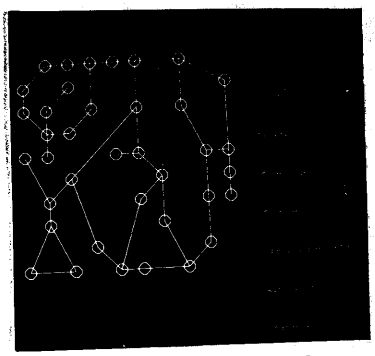


Fig.3 Original Network

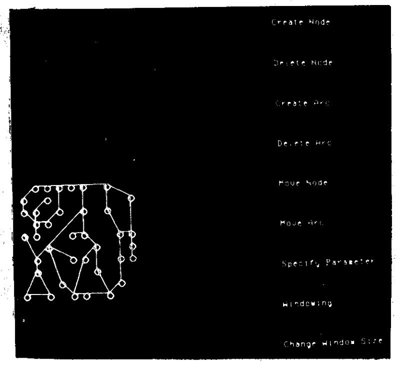


Fig.4 Reduced Network after altering window size

Main Mode

- Windowing
- Viewing
- Graphics Mode
- TTY Mode
- Hard-Copy Mode
- Saving/Restoring Mode
- Analyzing Mode

Graphics Mode

- Node Creation Mode
- Node Deletion Mode
- Arc Creation Mode
- Arc Deletion Mode
- Node Moving Mode
- Arc Moving Mode
- Parameter Specification Mode

Fig.6 Geans Main Mode Menu and Graphics Mode Menu

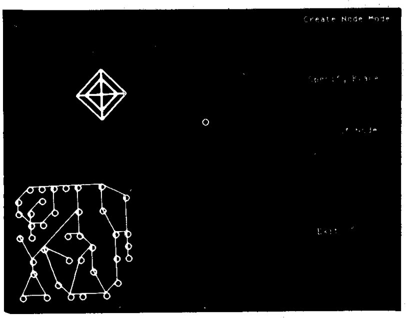


Fig.5 Subsequent Network editing

Network editing task is partitioned into several subtasks, i.e., node creation, node deletion, arc creation, arc deletion, node moving, arc moving and parameter specification subtasks.

Positioning of a node, i.e., node creation subtask, is performed by positioning a tracking object which can move inside the window area with the use of a light pen. Arc creation subtask is performed by specifying two nodes to be linked, i.e., start and end nodes of an arc, with the use of a light pen. Node deletion and arc deletion subtasks are performed by specifying a node and an arc, respectively, to be deleted with the use of a light pen. Node deletion subtask occurs a secondary effect of deletion of arcs connected with the deleted node. These two subtasks are corresponding to derivation of so-called partial networks and subnetworks from original networks, respectively. Node moving subtask means the task in which the position of an existing node is altered with the use of a light pen. Arc moving subtask means the task in which the start or end node of an arc is altered with the use of a light pen. Parameter specification subtask means the task in which the parameters of a node or arcs are inputted or modified with the use of a light pen and a TTY.

In order to perform positioning nodes with accurate coordinates, Geans provides an optional function by which a matrix of 10X10 visual dots to be hitted with the use of a light pen for positioning can be displayed on the window.

Fig.7 - 14 shows network editing subtasks. Fig.7 shows node creation subtask. Fig.8 shows parameter specification subtask about node parameter. Fig.9 shows arc creation subtask. Fig.10 shows parameter specification subtask about arc parameters. Fig.11 shows node moving subtask in which a node to be moved is hit by a light pen, and then a tracking object appears on the node. Fig.12 shows arc moving subtask in which either start node or end node of an arc is altered by the use of a tracking object and a light pen. Fig.13 shows a main mode menu which can be selected for network editing, analysis, property detection, or the like. Fig.14 shows a graphic mode menu with the use of a graphic display.

The constructed network can be saved on a disk file for the preparation for future use. The saved network can be restored from it.

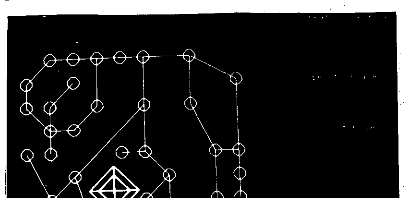
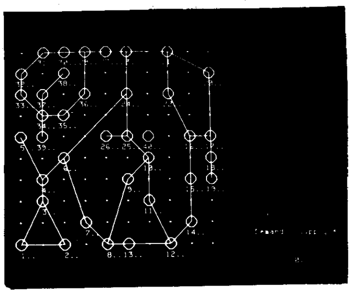
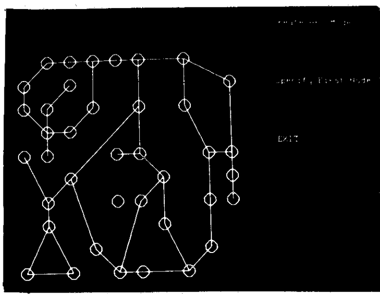


Fig.7 Node Creation





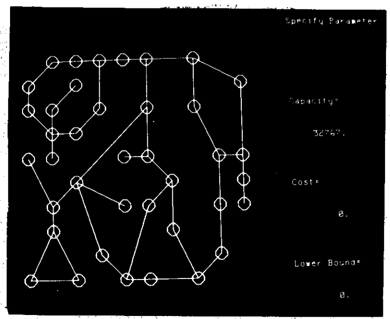


Fig.8 Node Parameter Specification Fig.9 Arc Creation Fig.10 Arc Parameter Specification

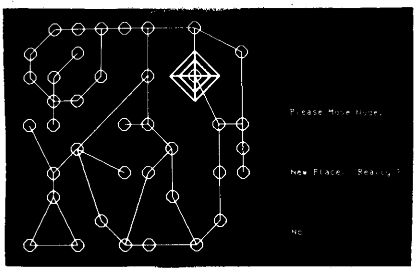


Fig.11 Node Moving

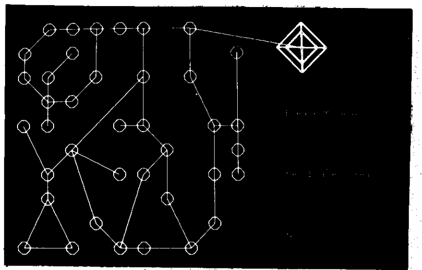


Fig.12 Arc Moving

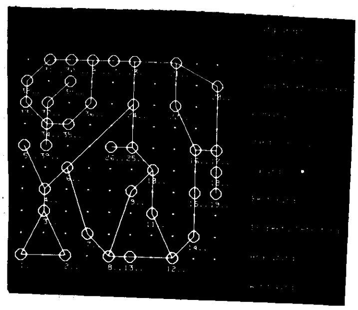


Fig.13 Main Mode Menu

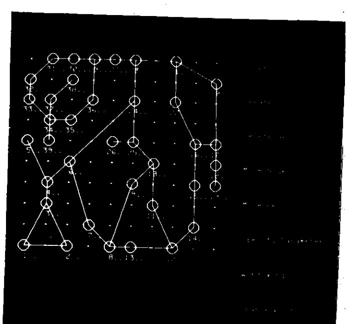


Fig.14 Graphics Mode Menu

We can build and modify large-scale networks easily and smoothly with the use of the above interactive editing, windowing and viewing functions.

3. Shape Definition Facility

In usual networks, a node is designated by a circle and an arc is designated by an arrow. It is desirable to define or modify the shape of a node or an arc in conforming to object systems or problems to be analyzed. This shape definition ability is considered to improve the understandability of object system or problem oriented networks to be analyzed. In the first version of the shape definition facility of Geans, the shape is modified at the Geans source statement (Fortran statement) level before Geans editing task. This definition task is easy because Geans has good modularity.

In the next version of the shape definition facility, the shape is defined or modified interactively at the display file library level at which a sequence of direction and length of beam vectors is specified step by step by a menu-hitting with the use of a light pen and TTY typing. The sequence is transmitted to the Geans load module in the form of a display file.

4. Abstract Data Type of Network Data

Network data aggregation is composed of node data items and arc data items. A node data item is composed of an identifier of each node, its coordinates in the virtual display area, and its several attributes. An arc data item is composed of an identifier of each arc, its positioning data, i.e., its start node and end node, and its several attributes. For transportation analysis problem, node attributes are the number of units supplied or consumed in each node, and arc attributes are upper or lower capacity, i.e., the maximum or minimum number of units which move through each arc, and costs or time length which it takes for one unit to move through it. For Petri Nets, node attributes are the number of tokens in each node representing a "place" of Petri Nets, and transition conditions of each node representing a "transition". For PERT / CPM Charts, arc attributes are costs and time length which it takes for each arc representing an "activity" of PERT / CPM Charts to be performed.

These data items are referenced and updated frequently when network editing and analyzing. On the aspects of improving productivity, it is desirable that implementers of network editors or analyzers possess detailed knowledge not about how network data aggregation is composed or organized but about how they use network data. The technique of "abstract data type" is appropriate for such requirements.

Modularization by abstract data [Liskov 1974], which is considered to be an elegant system development technique in beginning with developing data operations oriented from traditional bottom-up approach, constructs software systems by defining an abstract type of data which is a class of objects characterized by the cluster operations which may be performed on a class of them. Software implementers need only to be aware of the behavior of the operation clusters. Irrelevant details about how the data item is represented in storage and how the operations are implemented are hidden from them. A software system is constructed hierarchically by this modularization technique.

A group of cluster operations for the abstract data type of network includes PUT NODE / ARC with identifier, GET NODE / ARC with identifier, APPEND NODE / ARC, DELETE NODE / ARC, KNOW PAIR, SAVE and RESTORE operations. A PUT NODE / ARC operation specifies that node or arc data item is entered with its identifier and one of its attributes. A GET NODE / ARC operation specifies that an attribute of node or arc data item is inspected with its identifier. An APPEND NODE / ARC operation specifies that a node or an arc is entered with its identifier and coordinates or position in a virtual display area. A DELETE NODE / ARC operation specifies that a node or arc is erased in network. A DELETE NODE operation occurs a secondary effect in performing a DELETE ARC operation for arcs connected with the deleted node. A KNOW PAIR operation specifies that start and end nodes of an arc are recognized. A SAVE or RESTORE operation specifies that network data aggregation is saved on a disk file or restored from it in the user's unknown form for future use.

These operations are fully sufficient for constructing network editors and analyzers without complete knowledge about the organization of operation clusters and network data. Each operation is constructed in the subroutine or function subprogram form of Fortran IV, so we can implement network editors and analyzers by combining subroutine or function calls invoking the operations. By the way, network data aggregation is internally organized in a unique data structure of the table form in the current version of Geans, but implemeters of network editors and analyzers may not recognize this fact. Network editors and analyzers can be implemented with the use of a group of standard access operation routines for network data aggregation.

5. Network Analysis Facility

Network analyzers are divided into two groups: qualitative analysis programs which detect structural properties of networks and quantitative analysis programs which detect their dynamic properties.

Analyzers of qualitative properties of networks ,i.e., programs for connectivity checking, tie set / cut set detection, path detection etc., are common to various object systems or problems to be analyzed. They have been developed as utility library programs of Geans on the foundation of so-called "Graph Theory" and "Network Theory".

Analyzers of quantitative properties have been developed for each object system or problem to be analyzed. For transportation systems, there exist programs to analyze three typical properties i.e., shortest paths, maximal flow and minimal cost flow in networks. All of them are based on the modified "Out-of-Kilter Method" [Muramatsu 1980] which was originally devised to analyze only minimal cost flow [Ford 1962]. In addition, "Skeleton Network Approach" has been developed for reducing a large-scale network into a smaller equivalent network in order to perfom effective minimal cost flow analysis [Matsui 1981]. For Petri Nets [Peterson 1977], Marking Simulator and analyzers with the use of state reachability trees have been developed [Ichihara 1981]. For PERT / CPM Charts, there exist programs to analyze costs and time length which it takes in performing activities and projects in PERT / CPM Charts [Inoue 1981]. As an application of the analyzers of transportation systems, the guidance system of urban railway networks named Editing and Guidance System with Graphics (Eags-G) has been developed [Hayashi 1980]. This system uses Geans editing facility and abstract type of network data as well as the analyzers of transportation systems.

Both qualitative and quantitative programs were effectively implemented not in network data dependency but in algorithm dependency with the use of abstract type of network data and its associated operations.

6. Application Examples of Geans

In this section, current application examples of Geans are shown with the use of photographs, and some application areas to be prepared are shown.

Fig.15 - 16 shows Petri Net editing process and its marking simulation. Fig.17 - 18 shows a PERT / CPM chart and its analysis result. Fig.19 - 20 shows editing process of an urban railway network and output of guide information about shortest path on time by Eags-G.

Several applications of Geans are now considered as follows.
(1)Constructing a more useful Eags-G incorporated with a guidance facility of minimal cost paths.
(2)Utilizing a Geans editing facility as a symbolic design editor of Two-Stage Designing Supporting System [Itoh 1981].

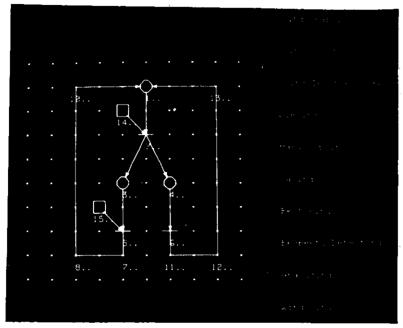


Fig.15 Petri Net Editing

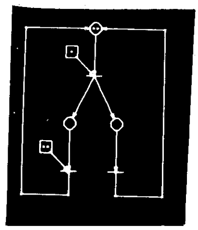


Fig.16 Petri Net Marking Simulation

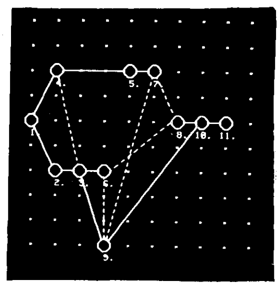


Fig.17 PERT/CPM Chart

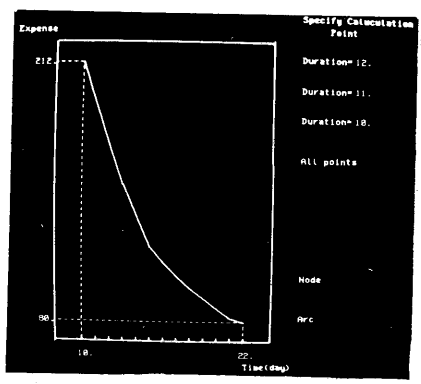


Fig.18 PERT/CPM Analysis Result

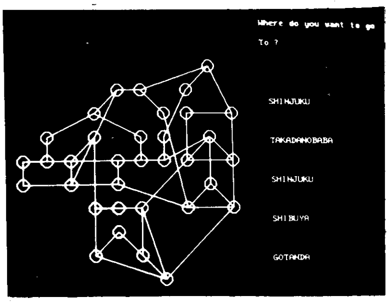


Fig.19 Editing of Railway Network

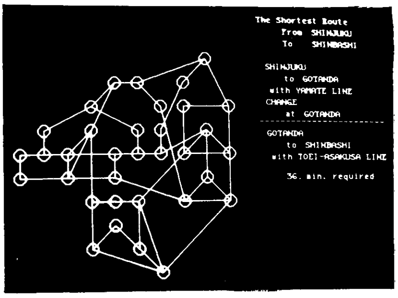


Fig.20 Guide Information on Shortest Path

7. Concluding Remarks

In this paper, we explained features of Graphical Editing and Analysis system for Network System (Geans) on DEC PDP-11 / 34 (RT-11 Fortran IV) minicomputer with a graphic display and a 32K word main memory which enables us to edit or modify various network representations and analyze their qualitative and quantitative properties, smoothly and interactively with the use of graphics and data abstraction. Interactive graphics techniques are useful for network editing and analysis task. Data abstraction technique is appropriate for developing network editors and analyzers effectively.

Currently, Geans is partitioned into three types of load modules: editing modules, analysis modules, and result viewing modules. Editing modules differs in each other in only shape definition form of object problem or system oriented nodes and arcs. Both analysis modules and result viewing modules are specialized at each object problem or systems with the use of data abstraction oriented routines, but only network property detection module is developed in the library form shared by these special purpose modules. Each module size is 25K words at maximum.

As future work, we will investigate three problems as follows.

(1)Introducing a large-capacity input device in order to increase editing ability of Geans, and incorporating it with a graphic display.
(2)Increasing level of data abstraction in order to facilate developing other network editors and analyzers.
(3)Constructing a shape definition system for problem or system oriented node and arc on the level of a display file.

Acknowledgement:
The authors wish to thank to Mr. H. Kurogane, Mr. I. Hayashi, Mr. T. Ichihara and Mr. M. Inoue for their discussion and cooperation for Geans.

References

Ford, L. R. et al. [1962]: Flows in Networks, Princeton Univ. Press, New Jersey.
Hayashi, I. et al. [1981]: Guidance System for Railway Network by Graphics, in Japanese, Proc. of 22nd Annual Convention of IPSJ, Vol.22, 1063-1064, March.
Ichihara, T. et al. [1981]: A Pilot Study on Computer Aided Design System for Concurrent Processing System by Petri Net, in Japanese, Proc. of 22nd Annual Convention of IPSJ, Vol.22, 319-320, March.
Inoue, M. [1981]: Development of Man-Machine Project Control System with Effective Use of Graphics, in Japanese, 4th grade Research Report of Sophia Univ., February.
Itoh, K. et al. [1980]: Development of General Purpose Network Editing and Analysis System - Geans - with the use of Graphics, in Japanese, Proc. of 21st Annual Convention of IPSJ, Vol.21, 1063-1064, May.
Itoh, K. [1981]: Two-Stage Designing Integrated by Functional Verification and Performance Evaluation for Online Software Systems, Proc. of 22nd Annual Convention of IPSJ, Vol.22, 317-318, March.
Liskov, B.H. and Zilles, S.N. [1974]: Programming with Abstract Data Types, SIGPLAN Notices, Vol.9, No.4, 50-59, April.
Matsui, M. et al. [1981]: Skeleton Network Approach for Minimal Cost Flow Problems, to be prepared for submitting.
Muramatsu, K. [1980]: Development of Analysis Programs for Transportation Metworks, in Japanese, 4th grade Research Report of Sophia Univ., February.
Peterson, J. L. [1977]: Petri Nets, ACM Computing Surveys, Vol.9, No.3, 223-252.