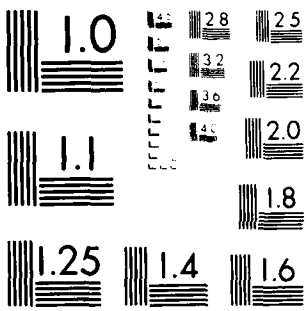
AD-A136 183 AN ITERATION ALGORITHM FOR OPTIMAL NETWORK FLOWS(U) 1/1
NAVAL POSTGRADUATE SCHOOL MONTEREY CA C J WOONG SEP 83

UNCLASSIFIED F/G 17/2 NL

END
DATE
FILMED
1-84
DTIC



MICROCOPY RESOLUTION TEST CHART

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

AN ITERATION ALGORITHM
FOR OPTIMAL NETWORK FLOWS

by

Chang Joon Woong

September 1983

Thesis Advisor: J. M. Wozencraft

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) An Iteration Algorithm for Optimal Network Flows		5. TYPE OF REPORT & PERIOD COVERED Masters Thesis; September 1983
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Chang Joon Woong		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		12. REPORT DATE September 1983
		13. NUMBER OF PAGES 84
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Packet Switching, Static Routing, Optimal Flow, Conservation Constraints, Capacity Constraints, Successive Saturation, Max-slack, Delay Analysis.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A packet switching network has the desirable feature of rapidly handling short (bursty) messages of the type often found in computer communication systems. In evaluating packet switching networks, the average time delay per packet is one of the most important measures of performance.

The problem of message routing to minimize time delay is analyzed here using two approaches, called "successive

saturation" and "max-slack", for various traffic requirement matricies and networks with fixed topology and link capacities.

Approved for public release, distribution unlimited.

AN ITERATION ALGORITHM FOR OPTIMAL NETWORK FLOWS

by

Chang Joon Woong
Lieutenant Colonel, Korea Marine Corps
B.S., Korea Naval Academy, 1965

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1983

Author: ______________________

Approved by: ______________________ Thesis Advisor

______________________ Second Reader

____
Chairman, Department of Electrical Engineering

____
Dean of Science and Engineering

ABSTRACT

A packet switching network has the desirable feature of rapidly handling short (bursty) messages of the type often found in computer communication systems. In evaluating packet switching networks, the average time delay per packet is one of the most important measures of performance.

The problem of message routing to minimize time delay is analyzed here using two approaches, called "successive saturation" and "max-slack", for various traffic requirement matricies and networks with fixed topology and link capacities.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENT

The author wishes to express his sincere appreciation to Professor J.W. Wozencraft for his assistance and guidance, in completing this work. Also, I want to give special thanks to my wife and family, (Chong and Hee-Won) for their combined support and understanding.

I. INTRODUCTION

A. THE PACKET SWITCHED NETWORK CONCEPT

A new technique for data communications that has evolved over 10 years is called PACKET SWITCHING. In general, communication networks may be conveniently divided into three types: CIRCUIT SWITCHING, MESSAGE SWITCHING and PACKET SWITCHING. Both message and packet switching uses a technique known as store and forward transmission.

A circuit switching network provides service by setting up a total path of connected lines from the source to the destination of the call. This complete circuit is set up by a special signaling message that threads its way through the network, seizing channels in the path as it proceeds. After the path is established, a return signal informs the source that data transmission may proceed, and all channels in the path are then used simultaneously.

The entire path remains allocated to the transmission, and only when the source release the circuit will all these channels be returned to the available pool for use in other paths. Circuit switching is the common method for telephone systems.

In message switching, only one channel is used at a time for a given transmission. The message first travels from its source node to the next node in its path, and when the entire message is received at this node, then the next step in its journey is selected. If this selected channel (link) is busy, the message waits in a queue, and finally when the channel becomes free, transmission begins. Thus, the message "hops" from node to node through the network using only one channel at a time, possibly queueing at busy channels, as it is successively stored and forwarded through the network.

Packet switching is basically the same as message switching except that the messages are decomposed into small equal pieces called packets, each of which has a minimum length. These packets are numbered and addressed and make their way through the net independently of each other. Thus, many packets of the same message may be in tranmission simultaneously, giving one of the main advantages of packet switching.

With packet switching systems, information is exchanged in the form of short packets. A packet-switched network can handle several different types of traffic concurrently. These include HIGH-THROUGHPUT traffic, for example, the transmission of large data files between computers, for which accuracy and high average data speed are the most important performance requirements; LOW-DELAY traffic, for example, interactive communication between a person at a terminal and a remote computer, for which accuracy and low average message delay are important; and REAL-TIME traffic, for example, packetized speech for which the performance of circuit-switched connection must be approached by maintaining a relatively constant data speed, but for which extreme accuracy is not important owing to the redundancy of the information.

The packet switched network is designed primarily for computer to computer communication. It has a much more rapid response which matches the internal behavior of computers and handles information in much the same way as does a computer. At the same time it can readily match the speed of attached computers to that of the terminal users, by virtue of its internal storage.

The prime purpose of store and forward packet switching is to enable communications resources to be used effectively and in such a way that they may be shared by many users operating in an intermittent fashion, giving each user a rapid response from the communication network just at the instant when this is required.

If there is need for transmitting a long continuous stream of data, then a circuit switched connection makes good sense. On the other hand, if the data flow is bursty, then some form of resource sharing can be used to great advantage; packet switching is an effective choice here.

Since packets are stored as they pass through switching nodes, it is possible to conduct speed, format and code conversion during the switching process. Another feature of packet switching is its ability to adaptively select good paths for packet journeys as a function of the network congestion.

Besides providing small network delays, packet switching has the desirable feature of rapidly handling small messages in spite of the presence of long messages that may be in transport at the same time, this is because of the decomposition of long messages into packets. Another useful property of this decomposition is that the nodal storage requirement is reduced.

In evaluating packet switching networks, the delay, throughput, cost and reliability are important measures.

Theoretical studies have been directed to the queueing and network flow problems in general and more specifically to such problems as delay analysis, route assignment, topological design and flow control, etc.

The topic chosen in this study is centered around optimal flow and minimum capacity assignment and delay analysis for packet-switched networks. The networks under consideration have a relatively complicated structure with a large number of source-destination node pairs.

B. THE ROUTING PROBLEM

A message routing procedure is merely a decision rule that determines the node a message will next visit in its path through the network. The objective of the routing procedures is to transport packets on a minimum delay path from source to destination.

In discussing routing policies for networks, an important distinction must be made between static and adaptive policies.

This distinction depends on the environment in which a policy is designed to operate. If network topology is not subject to changes (due to failure, modifications, growth) and traffic inputs are stationary, then the optimal routing solution is a static solution consisting of a set of fixed paths between all node pairs. The traffic between each source and destination pair may be distributed on several paths simultaneously in well defined proportions, where the proportions are fixed in time.

In a real network environment, however, the topology changes with time and user traffic requirements tend to fluctuate more or less rapidly. To minimize delay it is then necessary to implement an adaptive routing policy that can react and adjust to various changes. The adaptivity of a policy can be measured in terms of its response time. A reasonable procedure is to use a periodically refreshed static routing solution. Indeed, there is a continuum of solutions between the two extremes, static and adaptive, characterized by different response times and used for different applications.

Beside their use in operational networks, static routing policies find an important application in the network design process. During this process, for a given traffic pattern a prediction of the throughput and delay performance of a given topolocy is needed. The routing policy clearly has a major impact on such performance. Most routing implementations are adaptive, and unfortunately the analysis of adaptive routing policies is an extremely difficult task. To simplify the problem, the traffic pattern is usually approximated with a stationary pattern, and the routing policy with a static routing policy.

1. Static Routing

a. Routing Table Representation

A static routing policy is represented by a set of routing tables, one for each node, indicating how the traffic arriving at that node must be routed on the outgoing links, depending on its final destination. The routing table for node i is an Nx L matrix Pi(n,k), where N is the number of nodes in the network and Pi(n,k) is the fraction of traffic directed to node k which, upon arrival at node i, is routed through neighbor node n (Fig. 1.1). By the definition,

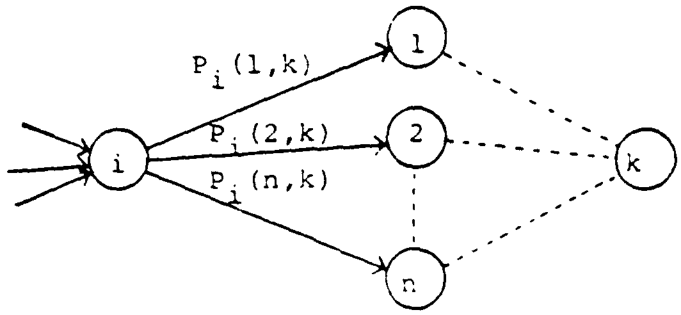
Pi (j,k) = 1

Where n is the number of neighbors of node i.

The actual distribution of the incoming traffic to outgoing links may be done randomly using as probability weights the values Pi(j,k). If for any node i there is only one permissible outgoing link from i to any destination k, then

Pi(j,k) = 1 for neighbor node j.
0 otherwise

Such a routing policy is referred to as a single path routing policy, since only one path is used from any node i to any destination node k. In general, the optimal static routing solution is a multipath solution, allowing for the simultaneous use of several routes in order to minimize delays. The routing table representation may also be used for adaptive policies.



n \ k	1	2		n
1				
k	$P_i(1,k)$	$P_i(2,k)$		$p(n,k)$

Figure 1.1. Routing Table.

b. Problem Formulation

In packet switched network, messages are segmented into small packets, and each packet travels from its source to its destination via a set of intermediate nodes. While awaiting transmission to the next node in the route, the packet must be stored until the link to that node is free. Thus, at each node there are several queues, one for each output channel.

Packet flow requirements between nodes arise at random times; therefore, link flows, queue length, and packet delay are random variables. A static routing problem can be defined as the problem of finding the static routing policy which optimizes the flow and minimizes the average time delay.

Considering the relationship between routing policy and link flows, the problem can be formulated for solution of optimal flow and minimum capacity with respect to the given requirement as follows.

given: (a) topology

(b) requirement matrix R

minimize: alpha

subject to: (a) conservation of flow

(b) flow $\leq$ capacity x alpha

Where alpha is a nonnegative scale factor.

The mathematical model and solution techniques are discussed in detail later.

2. Adaptive Routing

The adaptive routing problem consists of defining a procedure that dynamically updates routing tables according to changes in the network. This procedure will include the acquisition of the status of all neighboring nodes, which a packet associated with the source and destination node pair could visit next after being processed at the current node. The neighboring nodes are those with direct channels linking to a given node.

The structure of the routing table and the ways the information contained therein are to be used vary according to the individual routing procedure. For example, if single path routing is used, only one path links each source and destination node pair; therefore, only a single neighboring node associated with each node pair is listed in the routing table.

The design of routing is a process of determining the structure of routing tables and specifying the procedures of using them. The choice of routing depends on many factors such as network topology, available facilities, throughput and delay requirements, and so on. Adaptive routing techniques are ruled out in this study.

II. OPTIMAL FLOW AND MINIMUM CAPACITY ASSIGNMENT

The optimal flow is the best selection of paths between source and destination, given the traffic requirements and the network configuration. The definition of best paths may vary depending on the nature of the traffic. In this study, the best paths will be regarded as the paths of minimum "distance", where distance can be interpreted in various ways, for instance as time delay along the links.

To investigate average time delay of the networks, two different approaches are discussed using classical optimization procedures.

A. NETWORK MODELING

In a distributed packet-switched network, packets are transmitted over a collection of switching nodes in tandem. In modeling a packet switched network, the objective is usually to optimize the total flow of the networks, to minimize the capacity of each link, or to maximize the utilization of the network. The problem may be formulated as follows.

given	topology
	requirement matrix R
	capacity of each link (arbitrary)
minimize:	alpha
subject to:	(1) conservation constraints.

$$\sum F\ell(k) \quad - \quad \sum F\ell(k) \quad = \quad Ri(k)$$

outgoing links from node i	incoming links to node i	input at node i for node k

(2) capacity constraints

$F\ell(k) \leq \text{alpha} \times C\ell$

(3) positive constraints

$\sum F\ell(k) \geq 0$

Where $F\ell(k)$ is the flow on link destined for node k. We next look at each constraint further in detail.

1. Conservation Constraints

At each node i, the total outgoing packet flows destined for node k equal the incoming flows plus the input to be transmitted to node k.

$$\sum F\ell(k) \quad - \quad \sum F\ell(k) \quad = \quad Ri(k)$$

outgoing links from node i	incoming links to node i	input at node i for node k

where

k = destination node (k = 1,2,3...N).

N = number of nodes.

L = number of links.

i = source node (i = k, i = 1,2,3...N).

ℓ = link between two nodes (ℓ = 1,2,3...L).

$F\ell(k)$ = flow on link destined for node k.

$Ri(k)$ = input destined for node k at node i.

2. Capacity Constraints

The total flow destined for node k on a certain link should be less than or equal to the capacity of the link ℓ. Since linear programming algorithms have difficulties dealing with inequalities, slack variables are added to the constrains to absorb unused resources and thus to force equalities to appear.

When slack variables are introduced, the problem in standard form becomes.

$$\sum_{k=1}^{N} F\ell(k) - \text{alpha} \times C\ell + S\ell = 0$$

where

$C\ell$ = capacity of the link.

alpha = constant to optimally minimize the link capacity and total flow on link.

$S\ell$ = slack variables.

3. Positive Constraints

If we consider the network to have unidirectional links, the bidirectional flows between two nodes are divided into two one-way links in opposite direction, and the flow on each link is positive.

$F\ell(k) \geq 0$

Using matrix notation, the model simply may be expressed as a linear programming problem.

minimize: alpha

subject to: $AX = R$

$x \geq 0$

B. MODEL PROGRAMMING

In order to use a readily available Mathematical Programming System such as MPSIII, which can solve linear programming problems with up to 4000 rows and theoretically with an unlimited number of variables, the major programming problem with large scale networks is how to generate the input data for the MPSIII program. In other words, how do we generate the matrix form of the model.

1. Node to Link Incidence Matrix

For any network processing to take place in the computer, the network structure must be represented in some machine understandable form. A wide spectrum of different representation methods are available. Most of these are based on an incidence matrix scheme. A matrix form is convenient for the analysis of networks since the resulting matrices are in a format suitable for mathematical analysis.

The network of Fig. 2.1 will be used to illustrate the method. The node to link incidence matrix $E(i,\ell)$ describes the links connected to the node such that

$$E(i,\ell) = \begin{cases} 1 & \text{if link } \ell \text{ is outgoing from node i.} \\ -1 & \text{if link } \ell \text{ is incoming to node i.} \\ 0 & \text{if link } \ell \text{ is not connected to node i.} \end{cases}$$

The incidence matrix for input at node i destined for node k is a modification of the node to link incidence matrix $E(i,\ell)$ according to the following rules:

$$Ek(i,\ell) = \begin{cases} 0 & \text{for columns corresponding to links outgoing from node } k. \\ \text{delete the row if } i = k & \\ E(i,\ell) \text{ otherwise} & \end{cases}$$

For the illustration network (Fig. 2.1).

E(i,l) =

i \ ℓ	1	2	3	4	5
1	1	-1	1	-1	0
2	-1	1	0	0	-1
3	0	0	-1	1	1

for k = 1

E1(i,l) =

i \ ℓ	1	2	3	4	5
2	0	1	0	0	-1
3	0	0	0	1	1

E2(i,ℓ) and E3(i,ℓ) may be represented in the same manner.

for k = 2

E2(i,l) =

i \ ℓ	1	2	3	4	5
1	1	-1	1	-1	0
3	0	0	-1	1	1

for k = 3

E3(i,ℓ) =

i \ ℓ	1	2	3	4	5
1	1	-1	1	-1	0
2	-1	1	0	0	-1

$E(i,\ell)$ is a 3x5 matrix and $Ek(i,\ell)$ is a 2x5 matrix in size.

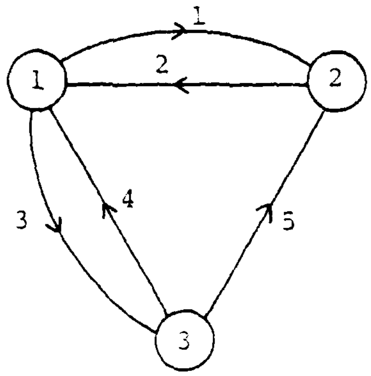


Figure 2.1. Illustration Network.

2. <u>Matrix Representation</u>

Matrix representation is a convenient tool for programming. The complete matrix representation for the illustration network is shown in Appendix A.

The general model matrix AX = R may be represented as follows:

$$
\underbrace{\begin{vmatrix}
E1 & & & & & \\
 & E2 & & & & \\
 & & E3 & & & \\
 & & & \ddots & & \\
 & & & & Ek & \\
I1 & I2 & I3 & \cdots & Ik & -C\ell
\end{vmatrix}}_{A}
\underbrace{\begin{vmatrix}
F\ell(1) \\ F\ell(2) \\ F\ell(3) \\ \vdots \\ F\ell(k) \\ F\ell(k) \\ S\ell \\ Alpha
\end{vmatrix}}_{X}
=
\underbrace{\begin{vmatrix}
Ri(1) \\ Ri(2) \\ Ri(3) \\ \vdots \\ Ri(k) \\ \left.\begin{matrix} 0 \\ \cdot \\ 0 \end{matrix}\right\} L
\end{vmatrix}}_{R}
$$

where

Ek = (N-1) x L incident matrix.

Ik = L x L identity matrix with zero elements corresponding to zero columns of the Ek matrix.

I = L x L identity matrix.

Ri(k) = requirement matrix which is the inputs to be sent from source node i to destination node k.

The size of the model in matrix representation is;

A = ((N-1) x N + L) x ((N+1) x L + 1)

X = ((N+1) x L + 1) x 1

R = ((N-1) x N + L) x 1

For example, the network consisting of 20 nodes and 40 links has the matrix which is A = 420 x 841, X = 841 x 1 and R = 420 x 1 in size.

3. Programs

To generate the MPSIII input data and to solve the model, requires two programs which are listed at the end of this study (Appendix C and D).

The first one is the data (model) generation program (Appendix C) written in the Fortran language. The output of this program uses special notation to designate the flow of the link connecting node pairs, destination node of the flow, flow variables, slack variables and input at the node. These notations are composed of a letter which is one of C, L, X or S and 7 numerical digits.

The notation Ci, Li, Xi, and Si imply conservation constraints, capacity constraints, flow variables, and slack variables respectively. The rest of the six numbers following Ci, Li, Xi and Si indicates link connecting node pair with first 4 digits and destination node or slack variable number with last 2 digits. For example, the notation C1000201 indicates the conservation constraint row which is the input at node 2 destined for node 1, L1020100 the link capacity row for the link connecting between node 2 and node 1, X1020101 the flow variable on the link connecting between node 2 and node 1 destined for node 1 and S1000001 the first slack variable.

The second program is the MPSIII program developed by Management Science Systems, Inc. It provides elaborate control language for the formulation of solution strategies for mathematical programming problems.

Individual instructions in this language bring in quite elaborate sections of code designed to execute the step as efficiently as possible.

Using the special notation discussed above, there is a limitation on the size of the problem which can be solved, because data names must consist of 8 characters maximum. This means that the maximum network size must be less than 100 nodes, if the computer capacity allows.

C. SOLUTION TECHNIQUES

As mentioned before, the classicial optimization procedures to solve the investigation model provide solutions which are the optimal paths and minimum link capacities for the network. Different optimal solutions may be expected depending on different methods of optimization. The two different approaches, each involving iteration of the linear programming solution procedure are discussed below, and illustrated in terms of one example network problem. These approaches are based on the characteristics of linear programming problems.

1. Successive Saturation Approach

The standard procedure of the MPSIII program listed at the end of this study (Appendix D and E) produces an "optimal" solution for alpha from the output data of the data generation program in the first run. But the corresponding flow solution is not a good solution in terms of the average time delay in the packet-switched network, when it is analyzed by the Kleinrock's delay analysis model (Ref. 4). The problem is that only the flow thru the saturated links is optimum.

In order to reduce the time delay, Iteration runs are attempted until all links are saturated. The detailed iteration method for the example network (Fig. 3.1) which has 5 nodes and 6 links, each with original link capacities 10 (to simplify the problem) is discussed below.

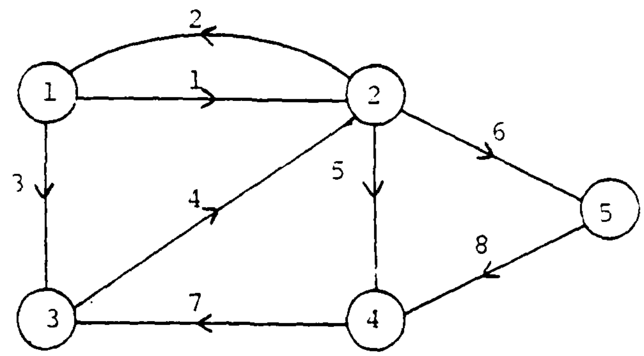


Figure 3.1. Example Network.

The ouput of MPSIII program is divided into row section and column section (Ref. 9 and Appendix D and E). The objective function value (which is the same as the alpha value) and the links saturated with the alpha value in the capacity constraint rows indicated with the first letter notation L can be read, and the flows on each link and the alpha value in column section. In the example network, for the first iteration the saturation level (alpha value) is 0.25, and the saturated links are L1010300 (link 3) and L1040300 (link 7). Their activity and slack activity levels are all zero, but their dual activity level is not (pp 48).

For the next iteration, the changed input datas are that the original capacity multiplied by the saturation level of first run are moved from the alpha column to the right hand side (RHS) column. In the example, the new link capacities 2.5 (0.25 x 10) are moved to RHS (pp. 50), and the problem is solved again. Repeating this iteration procedures until all links are saturated and moved to RHS, the objective function value becomes zero. When the objective function value is zero, the final flow solution is the desired solution of this successive saturation approach. In the example case, the six iterations are required to solve the problem. These whole procedures except programs and input datas from 2nd run for the example network are listed at the end of this study (Appendix D).

2. Max-Slack Approach

To simplify the solution of the above successive saturation approach, another approach called the max-slack approach is considered. It requires only two iteration of the program, but is otherwise quite similar. Like in the successive saturation approach, after the first run, all saturated link capacities are multiplied by the alpha value of the first run are moved to the RHS, and in addition, all slack variable columns are summed as the objective function in the data card for the second run.

In the example network, eight slack column variables are added on the objective rows, and all new capacities 2.5 (0.25 x 10) are moved to RHS. This is clearly shown on the output of control language "picture", and the program, input data and the output are also listed at the end (Appendix E).

D. DELAY ANALYSIS

The average time delay of a network is the average time a packet spends in the network traveling from its source to its destination nodes. To obtain a tractable expression for average time delay T in terms of the capacity of link ℓ, it is necessary to make simplifying assumptions. Packet lengths are assumed to be exponentially distributed, and are re-chosen with statistical independence at each node to form a Poisson process.

These assumptions known as the 'Independence assumption', introduced by D. Kleinrock (Ref. 4), make each queuing problem independent. In the analysis, each queue in a packet network is assumed to be an M/M/1 system with arrival rate of G_ℓ and to be independent of each other.

The average delay for link ℓ is given by the waiting time in the queue as

$$T_\ell = \frac{1}{uC_\ell - G_\ell}$$

Where G_ℓ is the packet traffic in the link and 1/u the average packet length in bits. To form a suitable average over all the queuing processes, the T are weighed by G_ℓ/r where r is the total input packet rate to the network. In this way, the total delay suffered by all packets per second of network operation $\sum_\ell G_\ell \cdot T$ is divided by the total number of packets carried by the network per second.

The total average time delay per packet through the network is then given by

$$T = \frac{1}{r} \sum_{\ell=1}^{L} \frac{G_\ell}{uC_\ell - G_\ell} = \frac{1}{r} \sum_{\ell=1}^{L} \frac{F_\ell}{C_\ell - F_\ell}$$

Where $G\ell/u = F\ell$, and L is the total number of links in the system and 1/u is equal to the average flow on link ℓ. It should be noted that the value of $C\ell$ (the minimum required capacity) must be greater than the total of all the flows on link ℓ. In other words, the value of $C\ell$ must be greater than the value which is the first run saturation level multiplied by the arbitrary capacity C for the first run.

The value of T includes both of the waiting time and the service time. The waiting time is subject to the interference of all other traffic within the network that consists of the data traffic as well as the control traffic. This average time delay can be used to analyze the relative goodness of the routing strategies considered in this study.

In deriving this basic formula, a number of factors (such as processing time and propagation delay) are neglected. In any realistic network, these variables as well as others must be considered in the analysis of the networks. According to the above formula, the average time delay depends only on the aggregate flow of the links. The reduction of the flow $F\ell$ of each link results in the reduction of the average time delay for the network which has constant input packet rate and link capacities.

For the example network (Fig. 3.1), the arbitrary capacity is 10 and the value of $C\ell$ must be greater than 2.5 (0.25 x 10). The average time delay for the capacity $F\ell$=10, and the input (requirement matrix) of 2 units from node 1 to node 4 and 5 units from node 2 to node 3 are Tss = 2.12/r for the successive saturation case and Tms = 2.083/r for the max-slack case. To make these calculations easy, the data summarizing the final output run are listed in Table I. We see that there is not much advantage of one procedure over the other in terms of performance, whereas the max-slack solution requires only two iterations, rather than 6 for the successive saturation algorithm.

TABLE I

LINK FLOW FOR THE EXAMPLE NETWORK

( ); Max-Slack

LINK (Nodes)	DESTINATION NODE 1	2	3	4	5	TOTAL FLOW
1-2				(2) 2		(2) 2
2-1		(2.5) 2.5				(2.5) 2.5
1-3		(2.5) 2.5				(2.5) 2.5
3-4						
2-4		(2.5) 2.25				(2.5) 2.25
2-5		0.25		(2) 2		(2) 2.25
4-3		(2.5) 2.5				(2.5) 2.5
5-4		0.25		(2) 2		(2) 2.25
Req Mat	C=10, $R_1(4)=2$, $R_2(3)=5$					

III. EXPERIMENTAL RESULTS AND CONCLUSIONS

A. EXPERIMENTS AND COMPARISON OF THE RESULTS

The experiment was conducted for two experimental networks to analyze both approaches, in addition to the example network. These networks are specified in Appendix B. The networks were used in the experiment with requirement matrices of 15 units from node 1 destined for node 5, 9 units from node 7 destined for node 2 for the 9 nodes/36 links network (Fig. B.1): and 15 units from node 2 destined for node 12, 4 units from node 5 destined for node 6, and 10 units from node 10 destined for node 1 for the 13 nodes/60 links network (Fig. B.2). Both the successive saturation and max-slack approaches were used for the successive saturation approach, the 21 iterations are required in this experiment.

The results and their comparison is listed in Tables II, III, and IV. These are obtained by writing each flow variable value from the final output run on each link of the networks, and summing these flow variable values to obtain the total flow on the link (Appendix F and G).

As shown in Tables II and III, the number of iterations required and the total link usage (utilization) are much greater for the successive saturation approach, but fewer large link capacities are required than in the max-slack approach. The average time delay depends on the capacity

TABLE II

RESULTS FOR THE 9/36 NETWORK

Fℓ MIN-CAP		4	3.75	3	2.25	1.88	1.75	1.63	1	0.5	TOTAL LINK USED	ITER REQ
NUM OF LINKS	SUCC - SAT	8	4		4	4	2	2		2	26	10
	MAX - SLACK	14		2					2		18	2

TABLE III

RESULTS FOR THE 13/60 NETWORK

FC MIN-CAP		3.75	3.33	2.92	2.75	2.6	2.5	2.36	2.08	2	1.97	1.88
NUM	SUCC - SAT	8	3	6		4		3	1	2	5	3
OF LINKS	MAX - SLACK	23			1		3					
FC MIN-CAP		1.56	1.39	1.3	1.17	0.63	0.47	0.7	0.25	TOT LINK USED	ITER REQ	
NUM	SUCC - SAT	2	1	2	3	2	1	2		48	21	
OF LINKS	MAX - SLACK								5	32	2	

value $C\ell$ (Table IV). When the capacity value $C\ell$ is close to the minimum required capacity, the time delay is less for the successive saturation approach than for the max-slack approach in both networks, and vice versa. It is significant for the large scale network, and for large enough link capacities $C\ell$, that the time delay difference becomes negligible.

Also, in the process of this study, it was noted that both approaches yield the same aggregate flows in the case of very simple networks (nodes less than or equal to 4), and for completely symmetric networks and requirement matrices (input and capacity matrices). For example, the time delays are the same for both approaches with 2 units input from the outmost nodes to the directly opposite nodes respectively (1 to 13, 2 to 12, 5 to 9, 12 to 2 and 13 to 1) for the 13 node network, because all links total flow are 3/4. The detail experimental results are listed in Appendix H.

B. CONCLUSIONS

This is an elementary study of the routing design problem for a packet network: given a traffic requirement matrix, minimize the average time delay per packet, subject to finding a feasible flow for a network with fixed topology and link capacities.

The general performance characteristics and advantages of the two approaches are investigated for small and simple networks. This illustrates one way to solve the design problem under limited conditions.

TABLE IV

TIME DELAY FOR THE CAPACITIES

CAPACITY		4	4.2	4.5	5	10	15
9/36 Net	SUCC - SAT		244.1/r	109.5/r	59.9/r	10.74/r	5.82/r
	MAX - SLACK		28.56/r	116.6/r	59.5/r	10.41/r	5.73/r
13/60 Net	SUCC - SAT	169.9/r	115.6/r	75.7/r	60.1/r	15.84/r	3.93/r
	MAX - SLACK	335.7/r	188.3/r	114.2/r	69.3/r	14.29/r	7.99/r

APPENDIX A

MATRIX REPRESENTATION FOR ILLUSTRATION NETWORK

$$
\underbrace{\begin{bmatrix}
0&1&0&0&1&&&&&&&&&&&&&&&& \\
0&0&0&1&1&&&&&&&&&&&&&&&& \\
&&&&&1&0&1&-1&0&&&&&&&&&&& \\
&&&&&0&0&-1&1&1&&&&&&&&&&& \\
&&&&&&&&&&1&-1&1&0&0&&&&&& \\
&&&&&&&&&&-1&1&0&0&0&&&&&& \\
0&&&&&1&&&&&1&&&&&1&&&&&-C1 \\
&1&&&&&0&&&&&1&&&&&1&&&&-C2 \\
&&0&&&&&1&&&&&1&&&&&1&&&-C3 \\
&&&1&&&&&1&&&&&0&&&&&1&&-C4 \\
&&&&1&&&&&1&&&&&0&&&&&1&-C5
\end{bmatrix}}_{A}
\underbrace{\begin{bmatrix}
F_1(1) \\ F_2(1) \\ F_3(1) \\ F_4(1) \\ F_5(1) \\ F_1(2) \\ F_2(2) \\ F_3(2) \\ F_4(2) \\ F_5(2) \\ F_1(3) \\ F_2(3) \\ F_3(3) \\ F_4(3) \\ F_5(3) \\ S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ ALPHA
\end{bmatrix}}_{X}
=
\underbrace{\begin{bmatrix}
R_2(1) \\ R_3(1) \\ R_1(2) \\ R_3(2) \\ R_1(3) \\ R_2(3) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0
\end{bmatrix}}_{R}
$$

Figure A.1. Matrix Representation for Illustration Network.

APPENDIX B

EXPERIMENTAL NETWORKS

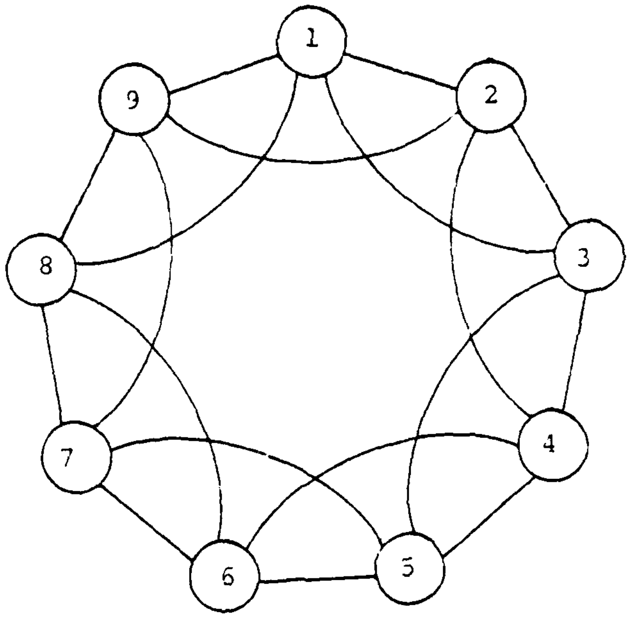


Figure B.1. The 9 Node/36 Link Network.

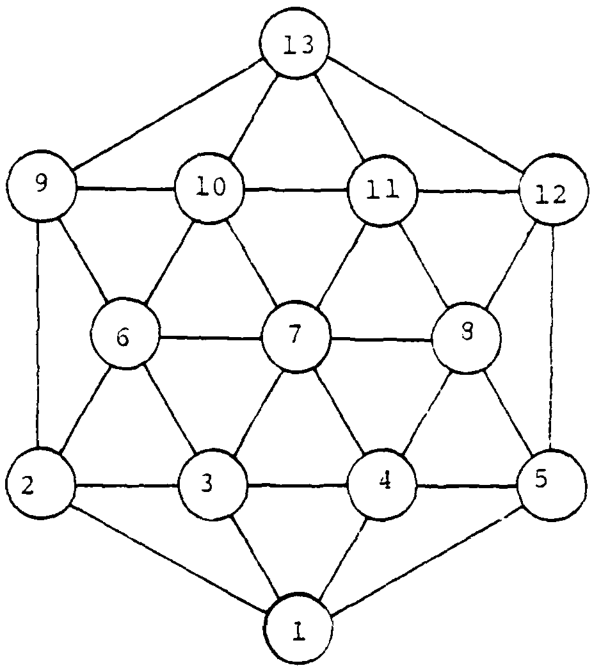


Figure B.2. The 13 Node/60 Link Network.

APPENDIX C

MODEL GENERATION PROGRAM FOR EXAMPLE PROBLEM

```
$JOB              1538P
C$OPTIONS FREE
C THIS IS THE DATA GENERATION PROGRAM FOR MPS...
C THE FORMAT SHOULD BE CHANGED ACCORDINGLY.
C N IS NUMBER OF NODES, L IS NUMBER OF LINKS.
C IE(I,L) IS DEFINED AS FOLLOWS:
C    1.LINK INCOMING TO NODEI IS -1.
C    2.LINK OUTGOING FROM NODEI IS 1.
C    3.LINK UNCONNECTED WITH NODEI IS 0.
C IZCOL(K,L) IS DEFINED AS FOLLOWS:
C    1.LINK OUTGOING FROM NODE K IS 0.
C    2.OTHERWISE, 1.
C IR(I,K) IS THE INPUT MATRIX FROM NODE I TO NODE K.
C CAP(L,I) IS THE LINK CAPACITY MATRIX.
C COLN1(1,J1) IS THE LINK MATRICX CONNECTING NODES.
C
      DIMENSION INC(13,60),IE(13,60),IZCOL(13,60),
     *  IRN(13,60),IR(13,13),CAP(60,1),ICOLN1(1,60),
     *  ICOLN(13,60),ICOLN2(13,60)
      INTEGER I,J,K,L,N,NULL,IDEN,KA,J1,INC2,IA,
     *  ID,IQ,IROWNO,IL,KC,KD,IRN1
C
C *** DATA INITIALIZATION *****
      READ(5,200) N,L
      READ(5,225)(ICOLN1(1,J1),J1=1,L)
      READ(5,220)((IE(I,J),I=1,N),J=1,L)
      READ(5,230)((IR(I,K),I=1,N),K=1,N)
      READ(5,240)(CAP(IQ,1),IQ=1,L)
  200 FORMAT(2I5)
  220 FORMAT(5I3)
  225 FORMAT(4I10)
  230 FORMAT(5I3)
  240 FORMAT(4E12.5)
C
C *** ZERO COLUMN GENERATION ****
      DO 19 J = 1,L
      DO 20 I = 1,N
      INCE = IE(I,J)
      IF(INCE .EQ. 1) GO TO 21
      INCE = 1
      GO TO 23
   21 INCE = 0
   23 IZCOL(I,J)=INCE
   20 CONTINUE
   19 CONTINUE
C *** COLUMN NUMBER GENERATION ****
      IAD = 1000000
      DO 1 K = 1,N
      IROWNO = 0
      DO 12 J1 = 1,L
      ICOLN(K,J1) = ICOLN1(1,J1) + K + IAD
      ICOLN2(K,J1) = ICOLN(K,J1) - K
   12 CONTINUE
C *** INCIDENT MATRICX GENERATION ***
      DO 2 I = 1,N
      IF(I .EQ. K) GO TO 2
      DO 3 J = 1,L
      NULL = IZCOL(K,J)
      IF(I .EQ. K) NULL = 0
      IDEN = IE(I,J)
      INC(I,J) = NULL*IDEN
    3 CONTINUE
```

```
      IROWNO = I*100
      IRN(I,K) = IROWNO + K + IAD
    2 CONTINUE
C *** ROW SECTION GENERATION ***
      ID = 1
      DO 61 I = 1,N
      IF(I .EQ. K) GO TO 61
      WRITE(6,140)IRN(I,K)
   61 CONTINUE
    1 CONTINUE
C *** CAPACITY ROW GENERATION ***
      DO 7 J = 1,L
      WRITE(6,145) ICOLN2(1,J)
    7 CONTINUE
C *** COLUMN SECTION GENERATION ***
      WRITE(6,500)
      DO 65 K = 1,N
      DO 62 J = 1,L
      NA = 0
      DO 63 I = 1,N
      IF(I .EQ. K) GO TO 63
      NULL = IZCOL(K,J)
      IDEN = IE(I,J)
      INC(I,J) = NULL*IDEN
      INC2 = INC(I,J)
      IF(INC2 .EQ. 0) GO TO 63
      NA = NA + 1
      WRITE(6,510)ICOLN(K,J),IRN(I,K),INC2
   63 CONTINUE
      IF(NA .EQ. 0) GO TO 62
      WRITE(6,520)ICOLN(K,J),ICOLN2(K,J),ID
   62 CONTINUE
   65 CONTINUE
C *** CAPACITY GENERATION ***
      DO 66 J = 1,L
      IA = J + 1000000
      WRITE(6,540) IA,ICOLN2(1,J),ID
   66 CONTINUE
C *** ALPHA COLUMN GENERATION ***
      WRITE(6,565) ID
      DO 67 J =1,L
      WRITE(6,570) ICOLN2(1,J),CAP(J,1)
   67 CONTINUE
C *** RHS GENERATION ***
      WRITE(6,550)
      DO 69 K = 1,N
      DO 68 I = 1,N
      IF(I .EQ. K) GO TO 68
      WRITE(6,560) IRN(I,K),IR(I,K)
   68 CONTINUE
   69 CONTINUE
      DO 5 IQ=1,L
    5 CONTINUE
      WRITE(6,580)
  140 FORMAT(1X,'E  C',I7)
  145 FORMAT(1X,'E  L',I7)
  500 FORMAT('COLUMNS')
  510 FORMAT(4X,'X',I7,2X,'C',I7,2X,I5)
  520 FORMAT(4X,'X',I7,2X,'L',I7,2X,I5)
  540 FORMAT(4X,'S',I7,2X,'L',I7,2X,I5)
  550 FORMAT('RHS')
  560 FORMAT(4X,'INPUT',5X,'C',I7,2X,I5)
  565 FORMAT(4X,'ALPHA',5X,'OBJ',7X,I5)
  570 FORMAT(4X,'ALPHA',5X,'L',I7,2X,E12.5)
  580 FORMAT('ENDATA')
      STOP
      END
```

```
$ENTRY
      5     8
      010200        020100      010300      030200
      020400        020500      040300      050400
  1  -1   0   0   0
 -1   1   0   0   0
  1   0  -1   0   0
  0  -1   1   0   0
  0   1   0  -1   0
  0   1   0   0  -1
  0   0  -1   1   0
  0   0   0  -1   1
  0   0   0   0   0
  0   0   0   0   0
  0   4   0   0   0
  4   0   0   0   0
  0   0   0   0   0
          -10.          -10.          -10.          -10.
          -10.          -10.          -10.          -10.
$ENTRY
```

APPENDIX D

OPTIMIZATION PROGRAM, INPUT DATA
MATRIX PICTURE AND OUTPUT FOR EXAMPLE PROBLEM

```
//CHANG JOB  (1538,1808),'CHANG JOON WOONG',CLASS=4
//*        MPSIII EXAMPLE PROBLEM
//*
//   EXEC  MSSMPS
//CPC.SYSIN  DD  *
             PROGRAM ('ND')
             INITIALZ
             TITLE ('EXAMPLE PROBLEM, 1ST RUN FOR SUCC-SAT')
             MOVE   (XOBJ,'OBJ')
             MOVE   (XRHS,'INPUT')
             MOVE   (XDATA,'LINEQS')
             MOVE   (XPBNAME,'PROJECT')
             CONVERT ('SUMMARY')
             SETUP ('MIN')
             BCDOUT
             PICTURE
             WHIZARD
             PRIMAL
             SOLUTION
             EXIT
             PEND
/*
//EXEC.SYSIN      DD   *
NAME              LINEQS
ROWS
 N   OBJ
 E   C1000201
 E   C1000301
 E   C1000401
 E   C1000501
 E   C1000102
 E   C1000302
 E   C1000402
 E   C1000502
 E   C1000103
 E   C1000203
 E   C1000403
 E   C1000503
 E   C1000104
 E   C1000204
 E   C1000304
 E   C1000504
 E   C1000105
 E   C1000205
 E   C1000305
 E   C1000405
 E   L1010200
 E   L1020100
 E   L1010300
 E   L1030200
 E   L1020400
 E   L1020500
 E   L1040300
 E   L1050400
COLUMNS
    X1020101   C1000201        1
    X1020101   L1020100        1
    X1030201   C1000201       -1
    X1030201   C1000301        1
    X1030201   L1030200        1
    X1020401   C1000201        1
    X1020401   C1000401       -1
    X1020401   L1020400        1
```

X1020501	C1000201	1
X1020501	C1000501	-1
X1020501	L1020500	1
X1040301	C1000301	-1
X1040301	C1000401	1
X1040301	L1040300	1
X1050401	C1000401	-1
X1050401	C1000501	1
X1050401	L1050400	1
X1010202	C1000102	1
X1010202	L1010200	1
X1010302	C1000102	1
X1010302	C1000302	-1
X1010302	L1010300	1
X1030202	C1000302	1
X1030202	L1030200	1
X1040302	C1000302	-1
X1040302	C1000402	1
X1040302	L1040300	1
X1050402	C1000402	-1
X1050402	C1000502	1
X1050402	L1050400	1
X1010203	C1000103	1
X1010203	C1000203	-1
X1010203	L1010200	1
X1020103	C1000103	-1
X1020103	C1000203	1
X1020103	L1020100	1
X1010303	C1000103	1
X1010303	L1010300	1
X1020403	C1000203	1
X1020403	C1000403	-1
X1020403	L1020400	1
X1020503	C1000203	1
X1020503	C1000503	-1
X1020503	L1020500	1
X1040303	C1000403	1
X1040303	L1040300	1
X1050403	C1000403	-1
X1050403	C1000503	1
X1050403	L1050400	1
X1010204	C1000104	1
X1010204	C1000204	-1
X1010204	L1010200	1
X1020104	C1000104	-1
X1020104	C1000204	1
X1020104	L1020100	1
X1010304	C1000104	1
X1010304	C1000304	-1
X1010304	L1010300	1
X1030204	C1000204	-1
X1030204	C1000304	1
X1030204	L1030200	1
X1020404	C1000204	1
X1020404	L1020400	1
X1020504	C1000204	1
X1020504	C1000504	-1
X1020504	L1020500	1
X1050404	C1000504	1
X1050404	L1050400	1
X1010205	C1000105	1
X1010205	C1000205	-1
X1010205	L1010200	1
X1020105	C1000105	-1
X1020105	C1000205	1
X1020105	L1020100	1
X1010305	C1000105	1

```
          X1010305    C1000305        -1
          X1010305    L1010300         1
          X1030205    C1000205        -1
          X1030205    C1000305         1
          X1030205    L1030200         1
          X1020405    C1000205         1
          X1020405    C1000405        -1
          X1020405    L1020400         1
          X1020505    C1000205         1
          X1020505    L1020500         1
          X1040305    C1000305        -1
          X1040305    C1000405         1
          X1040305    L1040300         1
          S1000001    L1010200         1
          S1000002    L1020100         1
          S1000003    L1010300         1
          S1000004    L1030200         1
          S1000005    L1020400         1
          S1000006    L1020500         1
          S1000007    L1040300         1
          S1000008    L1050400         1
          ALPHA       OBJ              1
          ALPHA       L1010200    -0.10000E  02
          ALPHA       L1020100    -0.10000E  02
          ALPHA       L1010300    -0.10000E  02
          ALPHA       L1030200    -0.10000E  02
          ALPHA       L1020400    -0.10000E  02
          ALPHA       L1020500    -0.10000E  02
          ALPHA       L1040300    -0.10000E  02
          ALPHA       L1050400    -0.10000E  02
RHS
          INPUT       C1000201         0
          INPUT       C1000301         0
          INPUT       C1000401         0
          INPUT       C1000501         0
          INPUT       C1000102         0
          INPUT       C1000302         0
          INPUT       C1000402         0
          INPUT       C1000502         0
          INPUT       C1000103         0
          INPUT       C1000203         5
          INPUT       C1000403         0
          INPUT       C1000503         0
          INPUT       C1000104         2
          INPUT       C1000204         0
          INPUT       C1000304         0
          INPUT       C1000504         0
          INPUT       C1000105         0
          INPUT       C1000205         0
          INPUT       C1000305         0
          INPUT       C1000405         0
ENDATA
/*
//
```

ROW	TYPE	X1020101	X1030201	X1020401	X1020501	X1040301	X1050401	X1010202	X1010302	X1030202	X1040302	X1050402	X1010203	X1020103	X1010303
OBJ	N														
C1C00201	E	1	-1	1	1										
C1C00301	E		1			-1									
C1C00401	E			-1		1	-1								
C1C00501	E				-1		1								
C1C00102	E							1	1						
C1C00302	E								-1	1	-1				
C1C00402	E										1	-1			
C1C00502	E											1			
C1C00103	E												1	-1	1
C1C00203	E												-1	1	
C1C00403	E														
C1C00503	E														
C1C00104	E														
C1C00204	E														
C1C00304	E														
C1C00504	E														
C1C00105	E														
C1C00205	E														
C1C00305	E														
C1C00405	E														
L1C10200	E							1					1		
L1C20100	E	1												1	
L1C10300	E								1						1
L1C30200	E		1							1					
L1C20400	E			1											
L1C20500	E				1										
L1C40300	E					1					1				
L1C50400	E						1					1			

ROW	X1020403	X1020503	X1040303	X1050403	X1010204	X1020104	X1010304	X1030204	X1020404	X1020504	X1050404	X1010205	X1020105	X1010305
OBJ														
C1C00201														
C1C00301														
C1C00401														
C1C00501														
C1C00102														
C1C00302														
C1C00402														
C1C00502														
C1C00103														
C1C00203	1	1												
C1C00403	-1		1	-1										
C1C00503		-1		1										
C1C00104					1	-1	1							
C1C00204					-1	1		-1	1	1				
C1C00304							-1	1						
C1C00504										-1	1			
C1C00105												1	-1	1
C1C00205												-1	1	
C1C00305														-1
C1C00405														
L1C10200					1							1		
L1C20100						1							1	
L1C10300							1							1
L1C30200								1						
L1C20400	1								1					
L1C20500		1								1				
L1C40300			1											
L1C50400				1							1			

ROW	X1030205	X1020405	X1020505	X1040305	S1000001	S1000002	S1000003	S1000004	S1000005	S1000006	S1000007	S1000008	ALPHA	IAFL1
OBJ													1	
C1C00201														
C1C00301														
C1C00401														
C1C00501														
C1C00102														
C1C00302														
C1C00402														
C1C00502														
C1C00103														
C1C00203														A
C1C00403														
C1C00503														
C1C00104														A
C1C00204														
C1C00304														
C1C00504														
C1C00105														
C1C00205	-1	1	1											
C1C00305	1			-1										
C1C00405		-1		1										
L1C10200					1								-A	
L1C20100						1							-A	
L1C10300							1						-A	
L1C30200	1							1					-A	
L1C20400		1							1				-A	
L1C20500			1							1			-A	
L1C40300				1							1		-A	
L1C50400												1	-A	

EXAMPLE PROBLEM, 1ST RUN FOR SUCCESSIVE SAT

SECTION 1 - ROWS

	NUMBER	...ROW..	AT	...ACTIVITY...	SLACK ACTIVITY	..LOWER LIMIT.	..UPPER LIMIT.	.DUAL ACTIVITY
	1	OBJ	BS	.25000	.25000-	NONE	NONE	1.00000
A	2	C1000201	EQ	.	.	.	.	.
A	3	C1000301	EQ	.	.	.	.	.
	4	C1000401	EQ	.	.	.	.	.05000-
	5	C1000501	EQ	.	.	.	.	.05000-
	6	C1000102	BS	.	.	.	.	.
	7	C1000302	BS	.	.	.	.	.
	8	C1000402	BS	.	.	.	.	.
	9	C1000502	BS	.	.	.	.	.
	10	C1000103	EQ	.	.	.	.	.05000-
	11	C1000203	EQ	5.00000	.	5.00000	5.00000	.05000-
	12	C1000403	EQ	.	.	.	.	.05000-
	13	C1000503	EQ	.	.	.	.	.05000-
A	14	C1000104	EQ	2.00000	.	2.00000	2.00000	.
A	15	C1000204	EQ	.	.	.	.	.
A	16	C1000304	EQ	.	.	.	.	.
A	17	C1000504	EQ	.	.	.	.	.
A	18	C1000105	EQ	.	.	.	.	.
	19	C1000205	BS	.	.	.	.	.
A	20	C1000305	EQ	.	.	.	.	.
	21	C1000405	EQ	.	.	.	.	.05000-
A	22	L1010200	EQ	.	.	.	.	.
A	23	L1020100	EQ	.	.	.	.	.
	24	L1010300	EQ	.	.	.	.	.05000
A	25	L1030200	EQ	.	.	.	.	.
A	26	L1020400	EQ	.	.	.	.	.
A	27	L1020500	EQ	.	.	.	.	.
	28	L1040300	EQ	.	.	.	.	.05000
A	29	L1050400	EQ	.	.	.	.	.

EXAMPLE PROBLEM, 1ST RUN FOR SUCCESSIVE SAT

SECTION 2 - COLUMNS

	NUMBER	.COLUMN.	AT	...ACTIVITY...	..INPUT COST..	..LOWER LIMIT.	..UPPER LIMIT.	.REDUCED COST.
	30	X1020101	BS	.	.	.	NONE	.
	31	X1030201	BS	.	.	.	NONE	.
	32	X1020401	LL	.	.	.	NONE	.05000
	33	X1020501	LL	.	.	.	NONE	.05000
	34	X1040301	BS	.	.	.	NONE	.
	35	X1050401	BS	.	.	.	NONE	.
▲	36	X1010202	LL	.	.	.	NONE	.
	37	X1010302	LL	.	.	.	NONE	.05000
▲	38	X1030202	LL	.	.	.	NONE	.
	39	X1040302	LL	.	.	.	NONE	.05000
▲	40	X1050402	LL	.	.	.	NONE	.
	41	X1010203	BS	.	.	.	NONE	.
	42	X1020103	BS	2.50000	.	.	NONE	.
	43	X1010303	BS	2.50000	.	.	NONE	.
	44	X1020403	BS	2.00000	.	.	NONE	.
	45	X1020503	BS	.50000	.	.	NONE	.
	46	X1040303	BS	2.50000	.	.	NONE	.
	47	X1050403	BS	.50000	.	.	NONE	.
	48	X1010204	BS	2.00000	.	.	NONE	.
▲	49	X1020104	LL	.	.	.	NONE	.
	50	X1010304	LL	.	.	.	NONE	.05000
	51	X1030204	BS	.	.	.	NONE	.
▲	52	X1020404	LL	.	.	.	NONE	.
	53	X1020504	BS	2.00000	.	.	NONE	.
	54	X1050404	BS	2.00000	.	.	NONE	.
▲	55	X1010205	LL	.	.	.	NONE	.
	56	X1020105	BS	.	.	.	NONE	.
	57	X1010305	LL	.	.	.	NONE	.05000
	58	X1030205	BS	.	.	.	NONE	.
	59	X1020405	LL	.	.	.	NONE	.05000
▲	60	X1020505	LL	.	.	.	NONE	.
	61	X1040305	BS	.	.	.	NONE	.
	62	S1000001	BS	.50000	.	.	NONE	.
▲	63	S1000002	LL	.	.	.	NONE	.
	64	S1000003	LL	.	.	.	NONE	.05000
	65	S1000004	BS	2.50000	.	.	NONE	.
	66	S1000005	BS	.50000	.	.	NONE	.
▲	67	S1000006	LL	.	.	.	NONE	.
	68	S1000007	LL	.	.	.	NONE	.05000
	69	S1000008	BS	.	.	.	NONE	.
	70	ALPHA	BS	.25000	1.00000	.	NONE	.

		X1020101	X1030201	X1020401	X1020501	X1040301	X1050401	X1010202	X1010302	X1030202	X1040302	X1050402	X1010203	X1020103	X1010303
OBJ	N														
C1000201	E	1	-1	1	1										
C1000301	E		1			-1									
C1000401	E			-1		1	-1								
C1000501	E				-1		1								
C1000102	E							1	1						
C1000302	E								-1	1	-1				
C1000402	E										1	-1			
C1000502	E											1			
C1000103	E												1	-1	1
C1000203	E												-1	1	
C1000403	E														
C1000503	E														
C1000104	E														
C1000204	E														
C1000304	E														
C1000504	E														
C1000105	E														
C1000205	E														
C1000305	E														
C1000405	E														
L1010200	E							1					1		
L1020100	E	1												1	
L1010300	E								1						1
L1030200	E		1							1					
L1020400	E			1											
L1020500	E				1										
L1040300	E					1					1				
L1050400	E						1					1			

		X1020403	X1020503	X1040303	X1050403	X1010204	X1020104	X1010304	X1030204	X1020404	X1020504	X1050404	X1010205	X1020105	X1010305
OBJ	N														
C1000201	E														
C1000301	E														
C1000401	E														
C1000501	E														
C1000102	E														
C1000302	E														
C1000402	E														
C1000502	E														
C1000103	E														
C1000203	E	1	1												
C1000403	E	-1		1	-1										
C1000503	E		-1		1										
C1000104	E					1	-1	1							
C1000204	E					-1	1		-1	1	1				
C1000304	E							-1	1						
C1000504	E										-1	1			
C1000105	E												1	-1	1
C1000205	E												-1	1	
C1000305	E														-1
C1000405	E														
L1010200	E					1							1		
L1020100	E						1							1	
L1010300	E							1							1
L1030200	E								1						
L1020400	E	1								1					
L1020500	E		1								1				
L1040300	E			1											
L1050400	E				1							1			

		X1030205	X1020405	X1020505	X1040305	S1000001	S1000002	S1000003	S1000004	S1000005	S1000006	S1000007	S1000008	ALPHA	INPL1
OBJ	N													1	
C1000201	E														
C1000301	E														
C1000401	E														
C1000501	E														
C1000102	E														
C1000302	E														
C1000402	E														
C1000502	E														
C1000103	E														
C1000203	E														A
C1000403	E														
C1000503	E														
C1000104	E														A
C1000204	E														
C1000304	E														
C1000504	E														
C1000105	E														
C1000205	E	-1	1	1											
C1000305	E	1			-1										
C1000405	E		-1		1										
L1010200	E					1								-A	
L1020100	E						1							-A	
L1010300	E							1							A
L1030200	E	1							1					-A	
L1020400	E		1							1				-A	
L1020500	E			1							1			-A	
L1040300	E				1							1			A
L1050400	E												1	-A	

EXAMPLE PROBLEM, 2ND RUN FOR SUCCESSIVE SAT

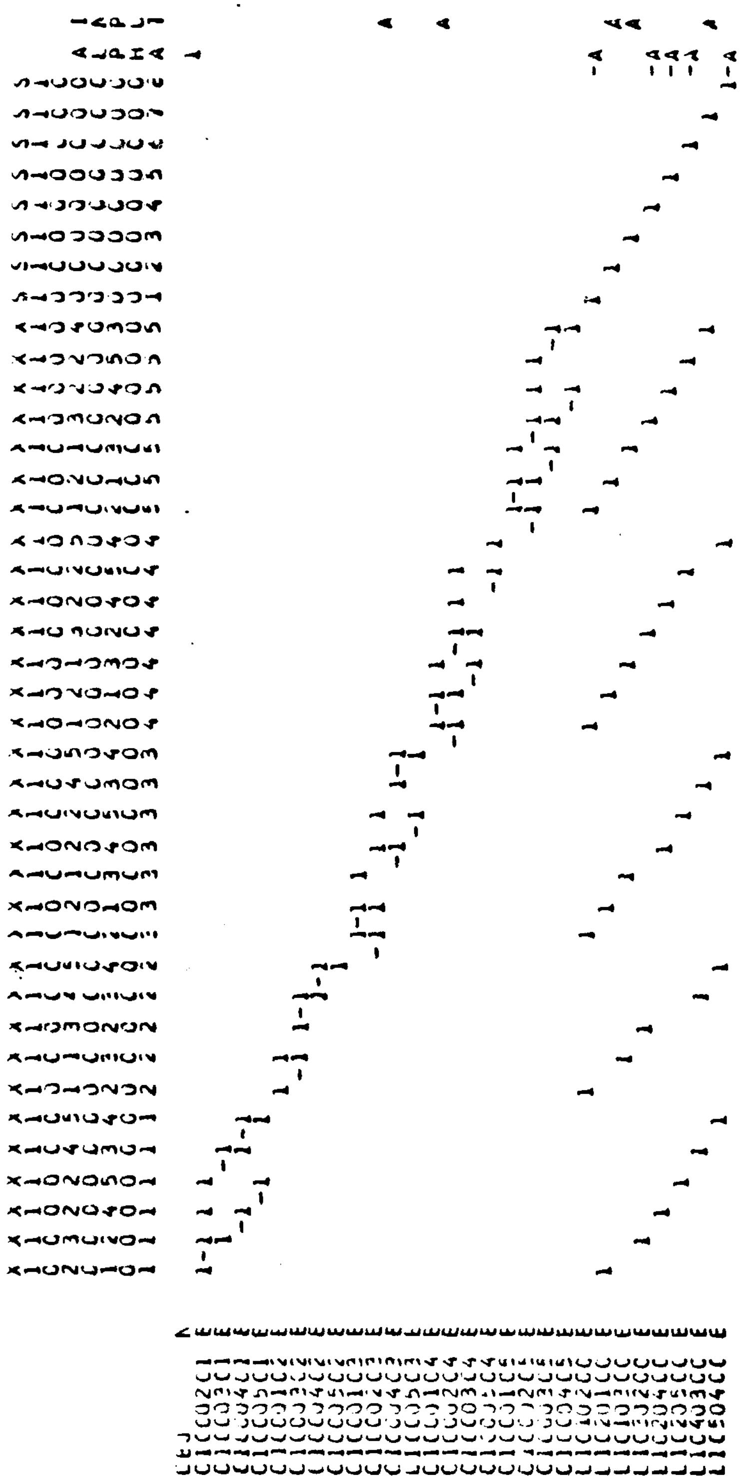
SECTION 1 - ROWS

	NUMBER	...ROW..	AT	...ACTIVITY...	SLACK ACTIVITY	..LOWER LIMIT.	..UPPER LIMIT.	.DUAL ACTIVITY
	1	OE.	BS	.25000	.25000-	NONE	NONE	1.00000
	2	C1000201	EQ	.	.	.	.	.10000-
	3	C1000301	EQ	.	.	.	.	.10000-
	4	C1000401	EQ	.	.	.	.	.20000-
	5	C1000501	EQ	.	.	.	.	.20000-
	6	C1000102	BS	.	.	.	.	.
	7	C1000302	BS	.	.	.	.	.
	8	C1000402	BS	.	.	.	.	.
	9	C1000502	BS	.	.	.	.	.
A	10	C1000103	EQ	.	.	.	.	.
	11	C1000203	EQ	5.00000	.	5.00000	5.00000	.10000-
	12	C1000403	EQ	.	.	.	.	.10000-
	13	C1000503	EQ	.	.	.	.	.10000-
A	14	C1000104	EQ	2.00000	.	2.00000	2.00000	.
A	15	C1000204	EQ	.	.	.	.	.
A	16	C1000304	EQ	.	.	.	.	.
A	17	C1000504	EQ	.	.	.	.	.
	18	C1000105	EQ	.	.	.	.	.10000
	19	C1000205	BS	.	.	.	.	.
A	20	C1000305	EQ	.	.	.	.	.
	21	C1000405	EQ	.	.	.	.	.10000-
A	22	L1010200	EQ	.	.	.	.	.
	23	L1020100	EQ	.	.	.	.	.10000
A	24	L1010300	EQ	2.50000	.	2.50000	2.50000	.
A	25	L1030200	EQ	.	.	.	.	.
A	26	L1020400	EQ	.	.	.	.	.
	27	L1020500	BS	.	.	.	.	.
	28	L1040300	EQ	2.50000	.	2.50000	2.50000	.10000
A	29	L1050400	EQ	.	.	.	.	.

EXAMPLE PROBLEM, 2ND RUN FOR SUCCESSIVE SAT

SECTION 2 - COLUMNS

	NUMBER	.COLUMN.	AT	...ACTIVITY...	..INPUT COST..	..LOWER LIMIT.	..UPPER LIMIT.	.REDUCED COST.
	30	X1020101	BS	.	.	.	NONE	.
	31	X1030201	BS	.	.	.	NONE	.
	32	X1020401	LL	.	.	.	NONE	.10000
	33	X1020501	LL	.	.	.	NONE	.10000
	34	X1040301	BS	.	.	.	NONE	.
	35	X1050401	BS	.	.	.	NONE	.
A	36	X1010202	LL	.	.	.	NONE	.
A	37	X1010302	LL	.	.	.	NONE	.
A	38	X1030202	LL	.	.	.	NONE	.
	39	X1040302	LL	.	.	.	NONE	.10000
A	40	X1050402	LL	.	.	.	NONE	.
	41	X1010203	LL	.	.	.	NONE	.10000
	42	X1020103	BS	2.50000	.	.	NONE	.
	43	X1010303	BS	2.50000	.	.	NONE	.
	44	X1020403	BS	2.00000	.	.	NONE	.
	45	X1020503	BS	.50000	.	.	NONE	.
	46	X1040303	BS	2.50000	.	.	NONE	.
	47	X1050403	BS	.50000	.	.	NONE	.
	48	X1010204	BS	2.00000	.	.	NONE	.
	49	X1020104	LL	.	.	.	NONE	.10000
A	50	X1010304	LL	.	.	.	NONE	.
	51	X1030204	BS	.	.	.	NONE	.
A	52	X1020404	LL	.	.	.	NONE	.
	53	X1020504	BS	2.00000	.	.	NONE	.
	54	X1050404	BS	2.00000	.	.	NONE	.
	55	X1010205	LL	.	.	.	NONE	.10000
	56	X1020105	BS	.	.	.	NONE	.
	57	X1010305	LL	.	.	.	NONE	.10000
	58	X1030205	BS	.	.	.	NONE	.
	59	X1020405	LL	.	.	.	NONE	.10000
A	60	X1020505	LL	.	.	.	NONE	.
	61	X1040305	BS	.	.	.	NONE	.
	62	S1000001	BS	.50000	.	.	NONE	.
	63	S1000002	LL	.	.	.	NONE	.10000
	64	S1000003	BS	.	.	.	NONE	.
	65	S1000004	BS	2.50000	.	.	NONE	.
	66	S1000005	BS	.50000	.	.	NONE	.
A	67	S1000006	LL	.	.	.	NONE	.
	68	S1000007	LL	.	.	.	NONE	.10000
A	69	S1000008	LL	.	.	.	NONE	.
	70	ALPHA	BS	.25000	1.00000	.	NONE	.



EXAMPLE PROBLEM, 3RD RUN FOR SUCCESSIVE SAT

SECTION 1 - ROWS

	NUMBER	...ROW..	AT	...ACTIVITY...	SLACK ACTIVITY	..LOWER LIMIT.	..UPPER LIMIT.	.DUAL ACTIVITY
	1	OBJ	BS	.22500	.22500-	NONE	NONE	1.00000
A	2	C1000201	EQ	.	.	.	.	.
A	3	C1000301	EQ	.	.	.	.	.
A	4	C1000401	EQ	.	.	.	.	.
	5	C1000501	EQ	.	.	.	.	.05000-
	6	C1000102	BS	.	.	.	.	.
	7	C1000302	BS	.	.	.	.	.
	8	C1000402	BS	.	.	.	.	.
	9	C1000502	BS	.	.	.	.	.
	10	C1000103	EQ	.	.	.	.	.05000-
	11	C1000203	EQ	5.00000	.	5.00000	5.00000	.05000-
A	12	C1000403	EQ	.	.	.	.	.
	13	C1000503	EQ	.	.	.	.	.05000-
	14	C1000104	EQ	2.00000	.	2.00000	2.00000	.05000-
	15	C1000204	EQ	.	.	.	.	.05000-
	16	C1000304	EQ	.	.	.	.	.05000-
	17	C1000504	EQ	.	.	.	.	.05000-
A	18	C1000105	EQ	.	.	.	.	.
	19	C1000205	BS	.	.	.	.	.
A	20	C1000305	EQ	.	.	.	.	.
A	21	C1000405	EQ	.	.	.	.	.
A	22	L1010200	EQ	.	.	.	.	.
A	23	L1020100	EQ	2.50000	.	2.50000	2.50000	.
	24	L1010300	EQ	2.50000	.	2.50000	2.50000	.05000
A	25	L1030200	EQ	.	.	.	.	.
	26	L1020400	EQ	.	.	.	.	.05000
	27	L1020500	BS	.	.	.	.	.
A	28	L1040300	EQ	2.50000	.	2.50000	2.50000	.
	29	L1050400	EQ	.	.	.	.	.05000

EXAMPLE PROBLEM, 3RD RUN FOR SUCCESSIVE SAT

SECTION 2 - COLUMNS

	NUMBER	.COLUMN.	AT	...ACTIVITY...	..INPUT COST..	..LOWER LIMIT.	..UPPER LIMIT.	.REDUCED COST.
	30	X1020101	BS	.	.	.	NONE	.
	31	X1030201	BS	.	.	.	NONE	.
	32	X1020401	LL	.	.	.	NONE	.05000
	33	X1020501	LL	.	.	.	NONE	.05000
	34	X1040301	BS	.	.	.	NONE	.
	35	X1050401	BS	.	.	.	NONE	.
▲	36	X1010202	LL	.	.	.	NONE	.
	37	X1010302	LL	.	.	.	NONE	.05000
▲	38	X1030202	LL	.	.	.	NONE	.
▲	39	X1040302	LL	.	.	.	NONE	.
	40	X1050402	LL	.	.	.	NONE	.05000
	41	X1010203	BS	.	.	.	NONE	.
	42	X1020103	BS	2.50000	.	.	NONE	.
	43	X1010303	BS	2.50000	.	.	NONE	.
	44	X1020403	BS	2.25000	.	.	NONE	.
	45	X1020503	BS	.25000	.	.	NONE	.
	46	X1040303	BS	2.50000	.	.	NONE	.
	47	X1050403	BS	.25000	.	.	NONE	.
	48	X1010204	BS	2.00000	.	.	NONE	.
▲	49	X1020104	LL	.	.	.	NONE	.
	50	X1010304	LL	.	.	.	NONE	.05000
	51	X1030204	BS	.	.	.	NONE	.
▲	52	X1020404	LL	.	.	.	NONE	.
	53	X1020504	BS	2.00000	.	.	NONE	.
	54	X1050404	BS	2.00000	.	.	NONE	.
▲	55	X1010205	LL	.	.	.	NONE	.
	56	X1020105	BS	.	.	.	NONE	.
	57	X1010305	LL	.	.	.	NONE	.05000
	58	X1030205	BS	.	.	.	NONE	.
	59	X1020405	LL	.	.	.	NONE	.05000
▲	60	X1020505	LL	.	.	.	NONE	.
	61	X1040305	BS	.	.	.	NONE	.
	62	S1000001	BS	.25000	.	.	NONE	.
▲	63	S1000002	LL	.	.	.	NONE	.
	64	S1000003	LL	.	.	.	NONE	.05000
	65	S1000004	BS	2.25000	.	.	NONE	.
	66	S1000005	LL	.	.	.	NONE	.05000
▲	67	S1000006	LL	.	.	.	NONE	.
	68	S1000007	BS	.	.	.	NONE	.
	69	S1000008	LL	.	.	.	NONE	.05000
	70	ALPHA	BS	.22500	1.00000	.	NONE	.

Row	Type	X1020101	X1030201	X1020401	X1020501	X1040301	X1050401	X1010202	X1010302	X1030202	X1040302	X1050402	X1010203	X1020103	X1010303	X1020403	X1020503	X1040303	X1050403	X1010204	X1020104	X1010304
LEJ	N																					
C1000201	E	1	-1	1	1																	
C1000301	E		1			-1																
C1000401	E			-1		1	-1															
C1000501	E				-1		1															
C1000102	E							1	1													
C1000302	E								-1	1	-1											
C1000402	E										1	-1										
C1000502	E											1										
C1000103	E												1	-1	1							
C1000203	E												-1	1		1	1					
C1000403	E															-1		1	-1			
C1000503	E																-1		1			
C1000104	E																			1	-1	1
C1000204	E																			-1	1	
C1000304	E																					-1
C1000504	E																					
C1000105	E																					
C1000205	E																					
C1000305	E																					
C1000405	E																					
L1010200	E							1					1							1		
L1020100	E	1												1							1	
L1010300	E								1						1							1
L1030200	E		1							1												
L1020400	E			1												1						
L1020500	E				1												1					
L1040300	E					1					1							1				
L1050400	E						1					1							1			

Row	Type	X1030204	X1020404	X1020504	X1050404	X1010205	X1020105	X1010305	X1030205	X1020405	X1020505	X1040305	S1000001	S1000002	S1000003	S1000004	S1000005	S1000006	S1000007	S1000008	ALPHA	INPUT
LEJ	N																				1	
C1000201	E																					
C1000301	E																					
C1000401	E																					
C1000501	E																					
C1000102	E																					
C1000302	E																					
C1000402	E																					
C1000502	E																					
C1000103	E																					
C1000203	E																					A
C1000403	E																					
C1000503	E																					
C1000104	E																					A
C1000204	E	-1	1	1																		
C1000304	E	1																				
C1000504	E			-1	1																	
C1000105	E					1	-1	1														
C1000205	E					-1	1		-1	1	1											
C1000305	E							-1	1			-1										
C1000405	E									-1		1										
L1010200	E					1							1								-A	
L1020100	E						1							1								A
L1010300	E							1							1							A
L1030200	E	1							1							1					-A	
L1020400	E		1							1							1					A
L1020500	E			1							1							1			-A	
L1040300	E											1							1			A
L1050400	E				1															1		A

EXAMPLE PROBLEM, 4TH RUN FOR SUCCESSIVE SAT

SECTION 1 - ROWS

	NUMBER	...COLUMN..	AT	...ACTIVITY...	SLACK ACTIVITY	..LOWER LIMIT.	..UPPER LIMIT.	.DUAL ACTIVITY
	1	OBJ	BS	.22500	.22500-	NONE	NONE	1.00000
A	2	C1000201	EQ	.	.	.	.	.
A	3	C1000301	EQ	.	.	.	.	.
A	4	C1000401	EQ	.	.	.	.	.
A	5	C1000501	EQ	.	.	.	.	.
	6	C1000102	BS	.	.	.	.	.
	7	C1000302	BS	.	.	.	.	.
	8	C1000402	BS	.	.	.	.	.
	9	C1000502	BS	.	.	.	.	.
	10	C1000103	EQ	.	.	.	.	.10000-
	11	C1000203	EQ	5.00000	.	5.00000	5.00000	.10000-
A	12	C1000403	EQ	.	.	.	.	.
A	13	C1000503	EQ	.	.	.	.	.
	14	C1000104	EQ	2.00000	.	2.00000	2.00000	.10000-
	15	C1000204	EQ	.	.	.	.	.10000-
	16	C1000304	EQ	.	.	.	.	.10000-
A	17	C1000504	EQ	.	.	.	.	.
A	18	C1000105	EQ	.	.	.	.	.
	19	C1000205	BS	.	.	.	.	.
A	20	C1000305	EQ	.	.	.	.	.
A	21	C1000405	EQ	.	.	.	.	.
A	22	L1010200	EQ	.	.	.	.	.
A	23	L1020100	EQ	2.50000	.	2.50000	2.50000	.
	24	L1010300	EQ	2.50000	.	2.50000	2.50000	.10000
A	25	L1030200	EQ	.	.	.	.	.
	26	L1020400	EQ	2.25000	.	2.25000	2.25000	.10000
	27	L1020500	EQ	.	.	.	.	.10000
A	28	L1040300	EQ	2.50000	.	2.50000	2.50000	.
A	29	L1050400	EQ	2.25000	.	2.25000	2.25000	.

EXAMPLE PROBLEM, 4TH RUN FOR SUCCESSIVE SAT

SECTION 2 - COLUMNS

	NUMBER	.COLUMN.	AT	...ACTIVITY...	..INPUT COST..	..LOWER LIMIT.	..UPPER LIMIT.	.REDUCED COST.
	30	X1020101	BS	.	.	.	NONE	.
	31	X1030201	BS	.	.	.	NONE	.
	32	X1020401	LL	.	.	.	NONE	.10000
	33	X1020501	LL	.	.	.	NONE	.10000
	34	X1040301	BS	.	.	.	NONE	.
	35	X1050401	BS	.	.	.	NONE	.
A	36	X1010202	LL	.	.	.	NONE	.
	37	X1010302	LL	.	.	.	NONE	.10000
A	38	X1030202	LL	.	.	.	NONE	.
A	39	X1040302	LL	.	.	.	NONE	.
A	40	X1050402	LL	.	.	.	NONE	.
A	41	X1010203	LL	.	.	.	NONE	.
	42	X1020103	BS	2.50000	.	.	NONE	.
	43	X1010303	BS	2.50000	.	.	NONE	.
	44	X1020403	BS	.25000	.	.	NONE	.
	45	X1020503	BS	2.25000	.	.	NONE	.
	46	X1040303	BS	2.50000	.	.	NONE	.
	47	X1050403	BS	2.25000	.	.	NONE	.
	48	X1010204	BS	2.00000	.	.	NONE	.
A	49	X1020104	LL	.	.	.	NONE	.
	50	X1010304	LL	.	.	.	NONE	.10000
	51	X1030204	BS	.	.	.	NONE	.
	52	X1020404	BS	2.00000	.	.	NONE	.
A	53	X1020504	LL	.	.	.	NONE	.
	54	X1050404	BS	.	.	.	NONE	.
A	55	X1010205	LL	.	.	.	NONE	.
	56	X1020105	BS	.	.	.	NONE	.
	57	X1010305	LL	.	.	.	NONE	.10000
	58	X1030205	BS	.	.	.	NONE	.
	59	X1020405	LL	.	.	.	NONE	.10000
	60	X1020505	LL	.	.	.	NONE	.10000
	61	X1040305	BS	.	.	.	NONE	.
	62	S1000001	BS	.25000	.	.	NONE	.
	63	S1000002	BS	.	.	.	NONE	.
	64	S1000003	LL	.	.	.	NONE	.10000
	65	S1000004	BS	2.25000	.	.	NONE	.
	66	S1000005	LL	.	.	.	NONE	.10000
	67	S1000006	LL	.	.	.	NONE	.10000
	68	S1000007	BS	.	.	.	NONE	.
	69	S1000008	BS	.	.	.	NONE	.
	70	ALPHA	BS	.22500	1.00000	.	NONE	.

ROW	TYPE	X1020101	X1030201	X1020401	X1020501	X1040301	X1050401	X1010202	X1010302	X1030202	X1040302	X1050402	X1010203	X1020103	X1010303	X1020403	X1020503	X1040303	X1050403
OBJ	N																		
C1000201	E	1	-1	1	1														
C1000301	E		1			-1													
C1000401	E			-1		1	-1												
C1000501	E				-1		1												
C1000102	E							1	1										
C1000302	E								-1	1	-1								
C1000402	E										1	-1							
C1000502	E											1							
C1000103	E												1	-1	1				
C1000203	E												-1	1		1	1		
C1000403	E															-1		1	-1
C1000503	E																-1		1
C1000104	E																		
C1000204	E																		
C1000304	E																		
C1000504	E																		
C1000105	E																		
C1000205	E																		
C1000305	E																		
C1000405	E																		
L1010200	E							1					1						
L1020100	E	1												1					
L1010300	E								1						1				
L1030200	E		1							1									
L1020400	E			1												1			
L1020500	E				1												1		
L1040300	E					1					1							1	
L1050400	E						1					1							1

ROW	X1010204	X1020104	X1010304	X1030204	X1020404	X1020504	X1050404	X1010205	X1020105	X1010305	X1030205	X1020405	X1020505	X1040305	S1000001	S1000002	S1000003	S1000004	S1000005	S1000006	S1000007	S1000008	ALPHA	INFLT
OBJ																							1	
C1000201																								
C1000301																								
C1000401																								
C1000501																								
C1000102																								
C1000302																								
C1000402																								
C1000502																								
C1000103																								
C1000203																								A
C1000403																								
C1000503																								
C1000104	1	-1	1																					A
C1000204	-1	1		-1	1	1																		
C1000304			-1	1																				
C1000504						-1	1																	
C1000105								1	-1	1														
C1000205								-1	1		-1	1	1											
C1000305										-1	1			-1										
C1000405												-1		1										
L1010200	1							1							1								-A	
L1020100		1							1							1								A
L1010300			1							1							1							A
L1030200				1							1							1					-A	
L1020400					1							1							1					A
L1020500						1							1							1				A
L1040300														1							1			A
L1050400							1															1		A

EXAMPLE PROBLEM, STP RUN FOR SUCCESSIVE SAT

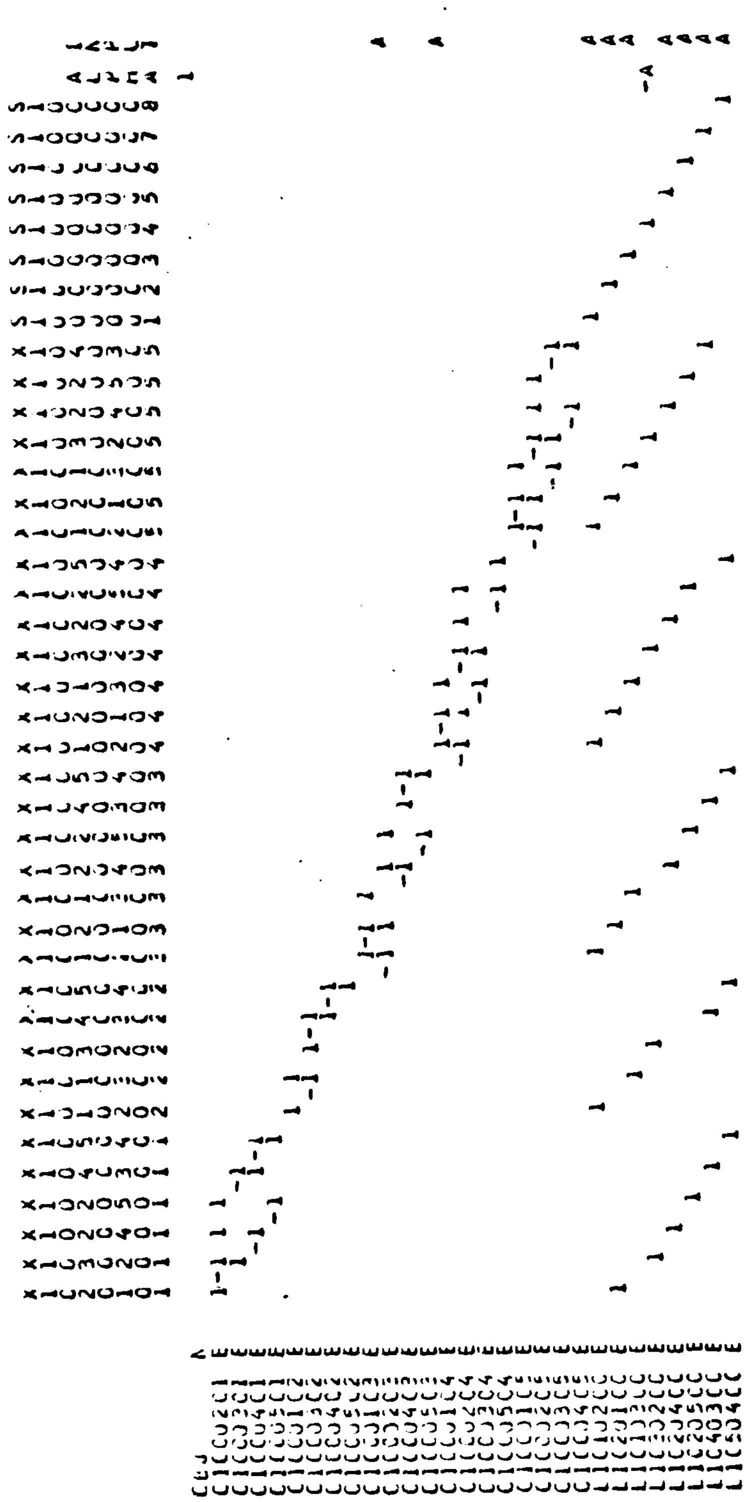
SECTION 1 - ROWS

	NUMBER	...ROW..	AT	...ACTIVITY...	SLACK ACTIVITY	..LOWER LIMIT.	..UPPER LIMIT.	.DUAL ACTIVITY
	1	OBJ	BS	.20000	.20000-	NONE	NONE	1.00000
A	2	C1000201	EQ	.	.	.	.	.
A	3	C1000301	EQ	.	.	.	.	.
	4	C1000401	EQ	.	.	.	.	.10000-
	5	C1000501	EQ	.	.	.	.	.10000-
	6	C1000102	BS	.	.	.	.	.
	7	C1000302	BS	.	.	.	.	.
	8	C1000402	BS	.	.	.	.	.
	9	C1000502	BS	.	.	.	.	.
	10	C1000103	EQ	.	.	.	.	.10000-
	11	C1000203	EQ	5.00000	.	5.00000	5.00000	.10000-
	12	C1000403	EQ	.	.	.	.	.10000-
	13	C1000503	EQ	.	.	.	.	.10000-
	14	C1000104	EQ	2.00000	.	2.00000	2.00000	.10000-
A	15	C1000204	EQ	.	.	.	.	.
A	16	C1000304	EQ	.	.	.	.	.
A	17	C1000504	EQ	.	.	.	.	.
A	18	C1000105	EQ	.	.	.	.	.
	19	C1000205	BS	.	.	.	.	.
A	20	C1000305	EQ	.	.	.	.	.
	21	C1000405	EQ	.	.	.	.	.10000-
	22	L1010200	EQ	.	.	.	.	.10000
A	23	L1020100	EQ	2.50000	.	2.50000	2.50000	.
	24	L1010300	EQ	2.50000	.	2.50000	2.50000	.10000
A	25	L1030200	EQ	.	.	.	.	.
A	26	L1020400	EQ	2.25000	.	2.25000	2.25000	.
A	27	L1020500	EQ	2.25000	.	2.25000	2.25000	.
	28	L1040300	EQ	2.50000	.	2.50000	2.50000	.10000
A	29	L1050400	EQ	2.25000	.	2.25000	2.25000	.

EXAMPLE PROBLEM, 5TH RUN FOR SUCCESSIVE SAT

SECTION 2 - COLUMNS

NUMBER	.COLUMN.	AT	...ACTIVITY...	..INPUT COST..	..LOWER LIMIT.	..UPPER LIMIT.	.REDUCED COST.
30	X1020101	BS	.	.	.	NONE	.
31	X1030201	BS	.	.	.	NONE	.
32	X1020401	LL	.	.	.	NONE	.10000
33	X1020501	LL	.	.	.	NONE	.10000
34	X1040301	BS	.	.	.	NONE	.
35	X1050401	BS	.	.	.	NONE	.
36	X1010202	LL	.	.	.	NONE	.10000
37	X1010302	LL	.	.	.	NONE	.10000
38	X1030202	LL	.	.	.	NONE	.
39	X1040302	LL	.	.	.	NONE	.10000
40	X1050402	LL	.	.	.	NONE	.
41	X1010203	LL	.	.	.	NONE	.10000
42	X1020103	BS	2.50000	.	.	NONE	.
43	X1010303	BS	2.50000	.	.	NONE	.
44	X1020403	BS	2.25000	.	.	NONE	.
45	X1020503	BS	.25000	.	.	NONE	.
46	X1040303	BS	2.50000	.	.	NONE	.
47	X1050403	BS	.25000	.	.	NONE	.
48	X1010204	BS	2.00000	.	.	NONE	.
49	X1020104	LL	.	.	.	NONE	.10000
50	X1010304	BS	.	.	.	NONE	.
51	X1030204	BS	.	.	.	NONE	.
52	X1020404	LL	.	.	.	NONE	.
53	X1020504	BS	2.00000	.	.	NONE	.
54	X1050404	BS	2.00000	.	.	NONE	.
55	X1010205	LL	.	.	.	NONE	.10000
56	X1020105	BS	.	.	.	NONE	.
57	X1010305	LL	.	.	.	NONE	.10000
58	X1030205	BS	.	.	.	NONE	.
59	X1020405	LL	.	.	.	NONE	.10000
60	X1020505	LL	.	.	.	NONE	.
61	X1040305	BS	.	.	.	NONE	.
62	S1000001	LL	.	.	.	NONE	.10000
63	S1000002	BS	.	.	.	NONE	.
64	S1000003	LL	.	.	.	NONE	.10000
65	S1000004	BS	2.00000	.	.	NONE	.
66	S1000005	BS	.	.	.	NONE	.
67	S1000006	LL	.	.	.	NONE	.
68	S1000007	LL	.	.	.	NONE	.10000
69	S1000008	BS	.	.	.	NONE	.
70	ALPHA	BS	.20000	1.00000	.	NONE	.



EXAMPLE PROBLEM, 6TH RUN FOR SUCCESSIVE SAT

SECTION 1 - ROWS

	NUMBER	...ROW..	AT	...ACTIVITY...	SLACK ACTIVITY	..LOWER LIMIT.	..UPPER LIMIT.	.DUAL ACTIVITY
	1	OBJ	BS	.	.	NONE	NONE	1.00000
A	2	C1000201	EQ	.	.	.	.	.
	3	C1000301	EQ	.	.	.	.	.10000-
	4	C1000401	EQ	.	.	.	.	.10000-
	5	C1000501	EQ	.	.	.	.	.10000-
	6	C1000102	BS	.	.	.	.	.
	7	C1000302	BS	.	.	.	.	.
	8	C1000402	BS	.	.	.	.	.
	9	C1000502	BS	.	.	.	.	.
A	10	C1000103	EQ	.	.	.	.	.
A	11	C1000203	EQ	5.00000	.	5.00000	5.00000	.
A	12	C1000403	EQ	.	.	.	.	.
A	13	C1000503	EQ	.	.	.	.	.
A	14	C1000104	EQ	2.00000	.	2.00000	2.00000	.
A	15	C1000204	EQ	.	.	.	.	.
	16	C1000304	EQ	.	.	.	.	.10000-
A	17	C1000504	EQ	.	.	.	.	.
	18	C1000105	BS	.	.	.	.	.
A	19	C1000205	EQ	.	.	.	.	.
	20	C1000305	EQ	.	.	.	.	.10000-
	21	C1000405	EQ	.	.	.	.	.10000-
A	22	L1010200	EQ	2.00000	.	2.00000	2.00000	.
A	23	L1020100	EQ	2.50000	.	2.50000	2.50000	.
A	24	L1010300	EQ	2.50000	.	2.50000	2.50000	.
	25	L1030200	EQ	.	.	.	.	.10000
A	26	L1020400	EQ	2.25000	.	2.25000	2.25000	.
A	27	L1020500	EQ	2.25000	.	2.25000	2.25000	.
A	28	L1040300	EQ	2.50000	.	2.50000	2.50000	.
A	29	L1050400	EQ	2.25000	.	2.25000	2.25000	.

EXAMPLE PROBLEM, ETF FUN FOR SUCCESSIVE SAT

SECTION 2 - COLUMNS

	NUMBER	.COLUMN.	AT	...ACTIVITY...	..INPUT COST..	..LOWER LIMIT.	..UPPER LIMIT.	.REDUCED COST.
	30	X1020101	BS	.	.	.	NONE	.
	31	X1030201	BS	.	.	.	NONE	.
	32	X1020401	LL	.	.	.	NONE	.10000
	33	X1020501	LL	.	.	.	NONE	.10000
	34	X1040301	BS	.	.	.	NONE	.
	35	X1050401	BS	.	.	.	NONE	.
▲	36	X1010202	LL	.	.	.	NONE	.
▲	37	X1010302	LL	.	.	.	NONE	.
	38	X1030202	LL	.	.	.	NONE	.10000
▲	39	X1040302	LL	.	.	.	NONE	.
▲	40	X1050402	LL	.	.	.	NONE	.
	41	X1010203	BS	.	.	.	NONE	.
	42	X1020103	BS	2.50000	.	.	NONE	.
	43	X1010303	BS	2.50000	.	.	NONE	.
	44	X1020403	BS	2.25000	.	.	NONE	.
	45	X1020503	BS	.25000	.	.	NONE	.
	46	X1040303	BS	2.50000	.	.	NONE	.
	47	X1050403	BS	.25000	.	.	NONE	.
	48	X1010204	BS	2.00000	.	.	NONE	.
▲	49	X1020104	LL	.	.	.	NONE	.
	50	X1010304	LL	.	.	.	NONE	.10000
	51	X1030204	BS	.	.	.	NONE	.
▲	52	X1020404	LL	.	.	.	NONE	.
	53	X1020504	BS	2.00000	.	.	NONE	.
	54	X1050404	BS	2.00000	.	.	NONE	.
	55	X1010205	BS	.	.	.	NONE	.
▲	56	X1020105	LL	.	.	.	NONE	.
	57	X1010305	LL	.	.	.	NONE	.10000
	58	X1030205	BS	.	.	.	NONE	.
	59	X1020405	LL	.	.	.	NONE	.10000
▲	60	X1020505	LL	.	.	.	NONE	.
	61	X1040305	BS	.	.	.	NONE	.
▲	62	S1000001	LL	.	.	.	NONE	.
▲	63	S1000002	LL	.	.	.	NONE	.
	64	S1000003	BS	.	.	.	NONE	.
	65	S1000004	LL	.	.	.	NONE	.10000
	66	S1000005	BS	.	.	.	NONE	.
▲	67	S1000006	LL	.	.	.	NONE	.
	68	S1000007	BS	.	.	.	NONE	.
	69	S1000008	BS	.	.	.	NONE	.
	70	ALPHA	BS	.	1.00000	.	NONE	.

APPENDIX E

OPTIMIZATION PROGRAM, INPUT DATA MATRIX PICTURE AND OUTPUT FOR MAX-SLACK APPROACH

```
//CHANG JOB  (1538,1808),'CHANG JOON WOONG',CLASS=A
//*
//*       MPSIII EXAMPLE PROBLEM
//*
//   EXEC  MSSMPS
//CPC.SYSIN  DD  *
          PROGRAM ('ND')
          INITIALZ
          TITLE ('EXAMPLE PROBLEM, 2ND RUN FOR MAX-SLACK')
          MOVE   (XOBJ,'OBJ')
          MOVE   (XRHS,'INPUT')
          MOVE   (XDATA,'LINEQS')
          MOVE   (XPBNAME,'PROJECT')
          CONVERT ('SUMMARY')
          SETUP ('MAX')
          BCDOUT
          PICTURE
          WHIZARD
          PRIMAL
          SOLUTION
          EXIT
          PEND
/*
//EXEC.SYSIN   DD  *
NAME          LINEQS
ROWS
 N  OBJ
 E  C1000201
 E  C1000301
 E  C1000401
 E  C1000501
 E  C1000102
 E  C1000302
 E  C1000402
 E  C1000502
 E  C1000103
 E  C1000203
 E  C1000403
 E  C1000503
 E  C1000104
 E  C1000204
 E  C1000304
 E  C1000504
 E  C1000105
 E  C1000205
 E  C1000305
 E  C1000405
 E  L1010200
 E  L1020100
 E  L1010300
 E  L1030200
 E  L1020400
 E  L1020500
 E  L1040300
 E  L1050400
COLUMNS
    X1020101   C1000201        1
    X1020101   L1020100        1
    X1030201   C1000201       -1
    X1030201   C1000301        1
    X1030201   L1030200        1
    X1020401   C1000201        1
    X1020401   C1000401       -1
```

X1020401	L1020400	1
X1020501	C1000201	1
X1020501	C1000501	-1
X1020501	L1020500	1
X1040301	C1000301	-1
X1040301	C1000401	1
X1040301	L1040300	1
X1050401	C1000401	-1
X1050401	C1000501	1
X1050401	L1050400	1
X1010202	C1000102	1
X1010202	L1010200	1
X1010302	C1000102	1
X1010302	C1000302	-1
X1010302	L1010300	1
X1030202	C1000302	1
X1030202	L1030200	1
X1040302	C1000302	-1
X1040302	C1000402	1
X1040302	L1040300	1
X1050402	C1000402	-1
X1050402	C1000502	1
X1050402	L1050400	1
X1010203	C1000103	1
X1010203	C1000203	-1
X1010203	L1010200	1
X1020103	C1000103	-1
X1020103	C1000203	1
X1020103	L1020100	1
X1010303	C1000103	1
X1010303	L1010300	1
X1020403	C1000203	1
X1020403	C1000403	-1
X1020403	L1020400	1
X1020503	C1000203	1
X1020503	C1000503	-1
X1020503	L1020500	1
X1040303	C1000403	1
X1040303	L1040300	1
X1050403	C1000403	-1
X1050403	C1000503	1
X1050403	L1050400	1
X1010204	C1000104	1
X1010204	C1000204	-1
X1010204	L1010200	1
X1020104	C1000104	-1
X1020104	C1000204	1
X1020104	L1020100	1
X1010304	C1000104	1
X1010304	C1000304	-1
X1010304	L1010300	1
X1030204	C1000204	-1
X1030204	C1000304	1
X1030204	L1030200	1
X1020404	C1000204	1
X1020404	L1020400	1
X1020504	C1000204	1
X1020504	C1000504	-1
X1020504	L1020500	1
X1050404	C1000504	1
X1050404	L1050400	1
X1010205	C1000105	1
X1010205	C1000205	-1
X1010205	L1010200	1
X1020105	C1000105	-1
X1020105	C1000205	1
X1020105	L1020100	1

```
          X1010305    C1000105         1
          X1010305    C1000305        -1
          X1010305    L1010300         1
          X1030205    C1000205        -1
          X1030205    C1000305         1
          X1030205    L1030200         1
          X1020405    C1000205         1
          X1020405    C1000405        -1
          X1020405    L1020400         1
          X1020505    C1000205         1
          X1020505    L1020500         1
          X1040305    C1000305        -1
          X1040305    C1000405         1
          X1040305    L1040300         1
          S1000001    OBJ              1             L1010200        1
          S1000002    OBJ              1             L1020100        1
          S1000003    OBJ              1             L1010300        1
          S1000004    OBJ              1             L1030200        1
          S1000005    OBJ              1             L1020400        1
          S1000006    OBJ              1             L1020500        1
          S1000007    OBJ              1             L1040300        1
          S1000008    OBJ              1             L1050400        1
          ALPHA       OBJ              1
*         ALPHA       L1010200    -0.10000E 02
*         ALPHA       L1020100    -0.10000E 02
*         ALPHA       L1010300    -0.10000E 02
*         ALPHA       L1030200    -0.10000E 02
*         ALPHA       L1020400    -0.10000E 02
*         ALPHA       L1020500    -0.10000E 02
*         ALPHA       L1040300    -0.10000E 02
*         ALPHA       L1050400    -0.10000E 02
RHS
          INPUT       C1000201         0
          INPUT       C1000301         0
          INPUT       C1000401         0
          INPUT       C1000501         0
          INPUT       C1000102         0
          INPUT       C1000302         0
          INPUT       C1000402         0
          INPUT       C1000502         0
          INPUT       C1000103         0
          INPUT       C1000203         5
          INPUT       C1000403         0
          INPUT       C1000503         0
          INPUT       C1000104         2
          INPUT       C1000204         0
          INPUT       C1000304         0
          INPUT       C1000504         0
          INPUT       C1000105         0
          INPUT       C1000205         0
          INPUT       C1000305         0
          INPUT       C1000405         0
          INPUT       L1010200         2.5
          INPUT       L1020100         2.5
          INPUT       L1010300         2.5
          INPUT       L1030200         2.5
          INPUT       L1020400         2.5
          INPUT       L1020500         2.5
          INPUT       L1040300         2.5
          INPUT       L1050400         2.5
ENDATA
/*
//
```

Row	Type	X1020101	X1030201	X1020401	X1020501	X1040301	X1050401	X1010202	X1010302	X1030202	X1040302	X1050402	X1010103	X1020103	X1010303
OBJ	N														
C1000201	E	1	-1	1	1										
C1000301	E		1			-1									
C1000401	E			-1		1	-1								
C1000501	E				-1		1								
C1000102	E							1	1						
C1000302	E								-1	1	-1				
C1000402	E										1	-1			
C1000502	E											1			
C1000103	E												1	-1	1
C1000203	E												-1	1	
C1000403	E														
C1000503	E														
C1000104	E														
C1000204	E														
C1000304	E														
C1000504	E														
C1000105	E														
C1000205	E														
C1000305	E														
C1000405	E														
L1010200	E							1					1		
L1020100	E	1												1	
L1010300	E								1						1
L1030200	E		1							1					
L1020400	E			1											
L1020500	E				1										
L1040300	E					1					1				
L1050400	E						1					1			

Row	Type	X1020403	X1020503	X1040303	X1050403	X1010204	X1020104	X1010304	X1030204	X1020404	X1020504	X1050404	X1010205	X1020105	X1010305
OBJ	N														
C1000201	E														
C1000301	E														
C1000401	E														
C1000501	E														
C1000102	E														
C1000302	E														
C1000402	E														
C1000502	E														
C1000103	E														
C1000203	E	1	1												
C1000403	E	-1		1	-1										
C1000503	E		-1		1										
C1000104	E					1	-1	1							
C1000204	E					-1	1		-1	1	1				
C1000304	E							-1	1						
C1000504	E										-1	1			
C1000105	E												1	-1	1
C1000205	E												-1	1	
C1000305	E														-1
C1000405	E														
L1010200	E					1							1		
L1020100	E						1							1	
L1010300	E							1							1
L1030200	E								1						
L1020400	E	1								1					
L1020500	E		1								1				
L1040300	E			1											
L1050400	E				1							1			

Row	Type	X1030205	X1020405	X1020505	X1040305	S1000001	S1000002	S1000003	S1000004	S1000005	S1000006	S1000007	S1000008	ALPHA	1AYL1
OBJ	N					1	1	1	1	1	1	1	1	1	
C1000201	E														
C1000301	E														
C1000401	E														
C1000501	E														
C1000102	E														
C1000302	E														
C1000402	E														
C1000502	E														
C1000103	E														
C1000203	E														A
C1000403	E														
C1000503	E														
C1000104	E														A
C1000204	E														
C1000304	E														
C1000504	E														
C1000105	E														
C1000205	E	-1	1	1											
C1000305	E	1			-1										
C1000405	E		-1		1										
L1010200	E					1									A
L1020100	E						1								A
L1010300	E							1							A
L1030200	E	1							1						A
L1020400	E		1							1					A
L1020500	E			1							1				A
L1040300	E				1							1			A
L1050400	E												1		A

EXAMPLE PROBLEM, 2ND RUN FOR SLACK-MAX

SECTION 1 - ROWS

NUMBER	...ROW..	AT	...ACTIVITY...	SLACK ACTIVITY	..LOWER LIMIT.	..UPPER LIMIT.	.DUAL ACTIVITY
1	OBJ	BS	4.00000	4.00000-	NONE	NONE	1.00000
2	C1000201	EQ	.	.	.	.	1.00000
3	C1000301	EQ	.	.	.	.	2.00000
4	C1000401	EQ	.	.	.	.	3.00000
5	C1000501	EQ	.	.	.	.	4.00000
6	C1000102	BS	.	.	.	.	.
7	C1000302	BS	.	.	.	.	.
8	C1000402	BS	.	.	.	.	.
9	C1000502	BS	.	.	.	.	.
10	C1000103	EQ	.	.	.	.	2.00000
11	C1000203	EQ	5.00000	.	5.00000	5.00000	3.00000
12	C1000403	EQ	.	.	.	.	1.00000
13	C1000503	EQ	.	.	.	.	2.00000
14	C1000104	EQ	2.00000	.	2.00000	2.00000	3.00000
15	C1000204	EQ	.	.	.	.	2.00000
16	C1000304	EQ	.	.	.	.	3.00000
17	C1000504	EQ	.	.	.	.	1.00000
18	C1000105	BS	.	.	.	.	.
19	C1000205	EQ	.	.	.	.	1.00000
20	C1000305	EQ	.	.	.	.	2.00000
21	C1000405	EQ	.	.	.	.	3.00000
22	L1010200	EQ	2.50000	.	2.50000	2.50000	1.00000-
23	L1020100	EQ	2.50000	.	2.50000	2.50000	1.00000-
24	L1010300	EQ	2.50000	.	2.50000	2.50000	2.00000-
25	L1030200	EQ	2.50000	.	2.50000	2.50000	1.00000-
26	L1020400	EQ	2.50000	.	2.50000	2.50000	2.00000-
27	L1020500	EQ	2.50000	.	2.50000	2.50000	1.00000-
28	L1040300	EQ	2.50000	.	2.50000	2.50000	1.00000-
29	L1050400	EQ	2.50000	.	2.50000	2.50000	1.00000-

EXAMPLE PROBLEM, 2ND RUN FOR SLACK-MAX

SECTION 2 - COLUMNS

NUMBER	.COLUMN.	AT	...ACTIVITY...	..INPUT COST..	..LOWER LIMIT.	..UPPER LIMIT.	.REDUCED COST.
30	X1020101	BS	.	.	.	NONE	.
31	X1030201	BS	.	.	.	NONE	.
32	X1020401	LL	.	.	.	NONE	4.00000-
33	X1020501	LL	.	.	.	NONE	4.00000-
34	X1040301	BS	.	.	.	NONE	.
35	X1050401	BS	.	.	.	NONE	.
36	X1010202	LL	.	.	.	NONE	1.00000-
37	X1010302	LL	.	.	.	NONE	2.00000-
38	X1030202	LL	.	.	.	NONE	1.00000-
39	X1040202	LL	.	.	.	NONE	1.00000-
40	X1050402	LL	.	.	.	NONE	1.00000-
41	X1010203	LL	.	.	.	NONE	2.00000-
42	X1020103	BS	2.50000	.	.	NONE	.
43	X1010303	BS	2.50000	.	.	NONE	.
44	X1020403	BS	2.50000	.	.	NONE	.
45	X1020503	LL	.	.	.	NONE	.
46	X1040303	BS	2.50000	.	.	NONE	.
47	X1050403	BS	.	.	.	NONE	.
48	X1010204	BS	2.00000	.	.	NONE	.
49	X1020104	LL	.	.	.	NONE	2.00000-
50	X1010304	LL	.	.	.	NONE	2.00000-
51	X1030204	BS	.	.	.	NONE	.
52	X1020404	BS	.	.	.	NONE	.
53	X1020504	BS	2.00000	.	.	NONE	.
54	X1050404	BS	2.00000	.	.	NONE	.
55	X1010205	LL	.	.	.	NONE	2.00000-
56	X1020105	BS	.	.	.	NONE	.
57	X1010305	LL	.	.	.	NONE	4.00000-
58	X1030205	BS	.	.	.	NONE	.
59	X1020405	LL	.	.	.	NONE	4.00000-
60	X1020505	LL	.	.	.	NONE	.
61	X1040305	BS	.	.	.	NONE	.
62	S1000001	BS	.50000	1.00000	.	NONE	.
63	S1000002	BS	.	1.00000	.	NONE	.
64	S1000003	LL	.	1.00000	.	NONE	1.00000-
65	S1000004	BS	2.50000	1.00000	.	NONE	.
66	S1000005	LL	.	1.00000	.	NONE	1.00000-
67	S1000006	BS	.50000	1.00000	.	NONE	.
68	S1000007	BS	.	1.00000	.	NONE	.
69	S1000008	BS	.50000	1.00000	.	NONE	.
70	ALPHA	LL	.	1.00000	.	NONE	1.00000

APPENDIX F

LINK FLOW FOR 9/36 NETWORK

Table F.1. Link Flow for 9/36 Network ( ); MAX-SLACK

LINK (NODES)	DESTINATION NODE 1	2	3	4	5	6	7	8	9	TOTAL FLOW
1-2		0.5			(4) 3.5					(4) 4
2-1										
1-3					(4) 4					(4) 4
3-1										
1-8					(4) 3.75					(4) 3.75
8-1		0.5								0.5
1-9					(3) 3.75					(3) 3.75
9-1										
2-3					1.63					1.63
3-2		(1) 2.25								(1) 2.25
2-4					(4) 1.88					(4) 1.88
4-2		(4) 2.25								(4) 2.25
2-9										
9-2		(4) 4								(4) 4
3-4					1.88					1.88
4-3		1.75								1.75
3-5					(4) 3.75					(4) 3.75
5-3		(1) 0.5								(1) 0.5
4-5					(4) 3.75					(4) 3.75
5-4										
4-6										

LINK	DESTINATION NODE									TOTAL
(NODES)	1	2	3	4	5	6	7	8	9	FLOW
6-4		(4) 4								(4) 4
5-6										
6-5					(4) 4					(4) 4
5-7										
7-5		(1) 0.5			(3) 3.5					(4) 4
6-7										
7-6		(4) 4								(4) 4
6-8										
8-6					(4) 4					(4) 4
7-8		2.25								2.25
8-7					1.63					1.63
7-9		(4) 2.25								(4) 2.25
9-7					(3) 1.88					(3) 1.88
8-9		1.75								1.75
9-8					1.88					1.88
REQ-MAT				$R_1(5)=15$, $R_5(2)=9$						

APPENDIX G

LINK FLOW FOR 13/60 NETWORK

Table G.1. Link Flow for 13/60 Network ( ); MAX-SLACK

LINK (NODES)	DESTINATION NODE 1	2	3	4	5	6	7	8	9	10	11	12	13	TOTAL FLOW
1-2														
2-1												(3.75) 3.75		(3.75) 3.75
1-3						(0.25)								(0.25)
3-1	(3.75) 3.33													(3.75) 3.33
1-4												1.39		1.39
4-1	(3.75) 3.33													(3.75) 3.33
1-5												(3.75) 2.36		(3.75) 2.36
5-1	(2.5) 3.33					(0.25)								(2.75) 3.33
2-3												(3.75) 3.75		(3.75) 3.75
3-2														
2-6												(3.75) 3.75		(3.75) 3.75
6-2														
2-9												(3.75) 3.75		(3.75) 3.75
9-2														

LINK	DESTINATION NODE													TOTAL
(NODES)	1	2	3	4	5	6	7	8	9	10	11	12	13	FLOW
3-4	0.73											1.88		2.61
4-3						1.56								1.56
3-6						(0.25) 1.97								(0.25) 1.97
6-3	(3.75) 2.92													(3.75) 2.92
3-7												(3.75) 1.88		(3.75) 1.88
7-3	1.15					0.41								1.56
4-5	0.97											1.39		2.36
5-4						2								2
4-7						0.64								0.64
7-4	2.6													2.6
4-8												1.89		1.89
8-4	0.97					0.2								1.17
5-8						(3.75) 2								(3.75) 2
8-5	(2.5) 2.36													(2.5) 2.36

LINK	DESTINATION NODE													TOTAL
(NODES)	1	2	3	4	5	6	7	8	9	10	11	12	13	FLOW
5-12												(3.75) 3.75		(3.75) 3.75
12-5														
6-7	0.95											(3.75) 1.97		(3.75) 2.92
7-6						(3.75) 1.97								(3.75) 1.97
6-9												0.47		0.47
9-6	1.9					0.07								1.97
6-10												1.3		1.3
10-6	(3.75) 1.97													(3.75) 1.97
7-8	0.73											(3.75) 1.88		(3.75) 2.6
8-7						(3.75) 1.17								(3.75) 1.17
7-10														
10-7	(3.75) 2.92													(3.75) 2.92
7-11												(3.75) 1.97		(3.75 1.97
11-7	0.61					0.56								1.17

LINK	DESTINATION NODE													TOTAL
(NODES)	1	2	3	4	5	6	7	8	9	10	11	12	13	FLOW
8-11						0.63								0.63
11-8	(2.5) 2.6													(2.5) 2.6
8-12												(3.75) 3.75		(3.75) 3.75
12-8														
9-10												1.3		1.3
10-9	1.9													1.9
9-13												(3.75) 2.92		(3.75 2.92
13-9						0.07								0.07
10-11	(2.5) 2.91													2.91
11-10														
10-13	0.3											2.62		2.92
13-10														
11-12												(3.75) 3.75		(3.75) 3.75
12-11														

LINK (NODES)	DESTINATION NODE 1	2	3	4	5	6	7	8	9	10	11	12	13	TOTAL FLOW
11-13						0.07								0.07
13-11	0.3											1.78		2.08
12-13														
13-12												(3.75) 3.75		(3.75) 3.75
REQ-MAT	$R_2(12)=15,\ R_5(6)=4,\ R_{10}(1)=10$													

APPENDIX H

LINK FLOW FOR SYMMETRIC REQ-MAT OF 13/60 NETWORK

Table H.1. Link Flow for Symmetric Req-Mat of 13/60 Network ( );MAX-SLACK

LINK	DESTINATION NODE													TOTAL
(NODES)	1	2	3	4	5	6	7	8	9	10	11	12	13	FLOW
1-2	(1/4)	1/2							(1/2)				(1/4) 1/4	(3/4) 3/4
2-1	1/4				1/2							(1/2)		(3/4) 3/4
1-3													(3/4) 3/4	(3/4) 3/4
3-1	(3/4) 3/4													(3/4) 3/4
1-4													(3/4) 3/4	(3/4) 3/4
4-1	(3/4) 3/4													(3/4) 3/4
1-5					1/2							(1/2)	(1/4) 1/4	(3/4) 3/4
5-1	(1/4) 1/4	1/2							(1/2)					(3/4) 3/4
2-3												(3/4) 3/4		(3/4) 3/4
3-2		(3/4) 3/4												(3/4) 3/4
2-6												(3/4) 3/4		(3/4) 3/4
6-2		(3/4) 3/4												(3/4) 3/4

LINK (NODES)	DESTINATION NODE 1	2	3	4	5	6	7	8	9	10	11	12	13	TOTAL FLOW
2-9									(1/2)			(1/4) 1/2		(3/4) 3/4
9-2	(1/4) 1/4	(1/2)			1/2									(3/4) 3/4
3-4					(3/4) 3/4									(3/4) 3/4
4-3		(3/4) 3/4												(3/4) 3/4
3-6													(3/4) 3/4	(3/4) 3/4
6-3					(3/4) 3/4									(3/4) 3/4
3-7													(3/4) 3/4	(3/4) 3/4
7-3	(3/4) 3/4													(3/4) 3/4
4-5					(3/4) 3/4									(3/4) 3/4
5-4									(3/4) 3/4					(3/4) 3/4
4-7									(3/4) 3/4					(3/4) 3/4
7-4	(3/4) 3/4													(3/4) 3/4

LINK	DESTINATION NODE													TOTAL
(NODES)	1	2	3	4	5	6	7	8	9	10	11	12	13	FLOW
4-8													(3/4) 3/4	(3/4) 3/4
8-4		(3/4) 3/4												(3/4) 3/4
5-8									(3/4) 3/4					(3/4) 3/4
8-5					(3/4) 3/4									(3/4) 3/4
5-12									1/2			(1/2)	(1/4) 1/4	(3/4) 3/4
12-5	(1/4) 1/4	1/2			(1/2)									(3/4) 3/4
6-7												(3/4) 3/4		(3/4) 3/4
7-6									(3/4) 3/4					(3/4) 3/4
6-9									(3/4) 3/4					(3/4) 3/4
9-6					(3/4) 3/4									(3/4) 3/4
6-10													(3/4) 3/4	(3/4) 3/4
10-6		(3/4) 3/4												(3/4) 3/4

LINK (NODES)	DESTINATION NODE 1	2	3	4	5	6	7	8	9	10	11	12	13	TOTAL FLOW
7-8												(3/4) 3/4		(3/4) 3/4
8-7									(3/4) 3/4					(3/4) 3/4
7-10									(3/4) 3/4					(3/4) 3/4
10-7	(3/4) 3/4													(3/4) 3/4
7-11												(3/4) 3/4		(3/4) 3/4
11-7	(3/4) 3/4													(3/4) 3/4
8-11													(3/4) 3/4	(3/4) 3/4
11-8					(3/4) 3/4									(3/4) 3/4
8-12												(3/4) 3/4		(3/4) 3/4
12-8	(3/4) 3/4													(3/4) 3/4
9-10					(3/4) 3/4									(3/4) 3/4
10-9									(3/4) 3/4					(3/4) 3/4

LINK (NODES)	DESTINATION NODE 1	2	3	4	5	6	7	8	9	10	11	12	13	TOTAL FLOW
9-13					(1/2)							(1/4) 1/2	1/4	(3/4) 3/4
13-9	(1/4) 1/4	(1/2)							(1/2)					(3/4) 3/4
10-11					(3/4) 3/4									(3/4) 3/4
11-10		(3/4) 3/4												(3/4) 3/4
10-13													(3/4) 3/4	(3/4) 3/4
13-10	(3/4) 3/4													(3/4) 3/4
11-12												(3/4) 3/4		(3/4) 3/4
12-11		(3/4) 3/4												(3/4) 3/4
11-13													(3/4) 3/4	(3/4) 3/4
13-11	(3/4) 3/4													(3/4) 3/4
12-13		(1/2)							1/2				(1/4) 1/4	(3/4) 3/4
13-12	(1/4) 1/4				(1/2)							1/4		(3/4) 3/4

REQ-MAT $R_1(13)=2$, $R_2(12)=2$, $R_5(9)=2$, $R_9(5)=2$, $R_{12}(2)=2$, $R_{13}(1)=2$

LIST OF REFERENCES

1. Arne, T., Computer Methods in Operations Research, p. 79, Academic Press, 1978.

2. Pooch, U. W., Green, W. H., and Moss, G. G., Telecommunications and Networking, Little, Brown Computer Systems Series, 1982.

3. Gass, S. I., Linear Programming Method and Application, 4th Ed., McGraw-Hill, 1975.

4. Kleinrock, L., Queueing Systems, V. 2, pp. 292-297, Wiley, 1976.

5. Davices, D. W., and Barber, D. A., Communication Networks for Computer, pp. 291-294, Wiley, 1973.

6. Carvis, H., Communications Network Analysis, pp. 103-104, Lexington, 1981.

7. Hiroshi, I. and Tadao, S., Theoretical Aspects in the Analysis and Synthesis of Packet Communication Networks, Proceedings of the IEEE, V. 66, No. 11, Nov 1978.

8. Kuo, F. F., Protocols and Techniques for Data Communications Networks, pp. 124-127, Prentice-Hall, 1981.

9. Ketron Inc., MPS III User Manual, 1980.

10. Phillips, D.T., and Alberto, G.D., Fundamentals of Network Analysis, Prentice Hall, 1981.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943	2
3.	Professor John W. Wozencraft, Code 62Wz Department of Electrical Engineering Naval Postgraduate School Monterey, CA 93943	3
4.	Professor Paul H. Moose, Code 62Me Department of Electrical Engineering Naval Postgraduate School Monterey, CA 93943	2
5.	Lieutenant Colonel Chang Joon Wong Korea MC Navy Headquarters (N-1) Seoul Korea	3

LMED

8