Technical Research Report 1165

AD

AD 748801

SIMPO-I GENERAL MATRIX MANIPULATOR (GMM)

Joanne M. Witt and Adele P. Narva
Pauline T. Olson, Task Leader
STATISTICAL RESEARCH AND ANALYSIS DIVISION

AF/RL

U. S. Army

Behavior and Systems Research Laboratory

January 1972

BEHAVIOR AND SYSTEMS RESEARCH LABORATORY

An activity of the Chief, Research and Development

J. E. UHLANER
Director

NOTICES

DISTRIBUTION: Primary distribution of this report has been made by BESRL. Please address correspondence concerning distribution of reports to: U. S. Army Behavior and Systems Research Laboratory, Attn: RDMR-BLZ, 1300 Wilson Boulevard, Arlington, Virginia 22209.

FINAL DISPOSITION: This report may be destroyed when it is no longer needed. Please do not return it to the Behavior and Systems Research Laboratory.

NOTE: The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)

U. S. Army Behavior and Systems Research Laboratory, Arlington, Virginia

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. REPORT TITLE

SIMPO-I GENERAL MATRIX MANIPULATOR (GMM)

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

5. AUTHOR(S) (First name, middle initial, last name)

Joanne M. Witt and Adele P. Narva

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
January 1972	83	7

8a. CONTRACT OR GRANT NO.

b. PROJECT NO.

DA R&D Proj. No. 2Q065101M711

c. SIMPO II

d. d-18

9a. ORIGINATOR'S REPORT NUMBER(S)

Technical Research Report 1165

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

Deputy Chief of Staff for Personnel, DA
Office of Personnel Operations, DA

13. ABSTRACT

In response to an operations research requirement established under the auspices of the Army Study Advisory Committee, BESRL's Statistical Research and Analysis Division has been engaged in study and evaluation of the Army's personnel subsystems with respect to effectiveness of its policies in assignment, training, utilization, and contingency readiness of specialized personnel. The SIMPO-I effort has been directed to development of a model simulation package which management can use to assess quantitatively the cumulative impact of personnel policy changes on the allocation, distribution, and utilization of Army personnel with special attention to effects of policies on deployability. The present Technical Research Report deals with the development of a computer model which can be used in evaluating many different military manpower systems. The model, the General Matrix Manipulator (GMM), is based on the movement of elements within or between matrices used to represent the different personnel categories according to supplied rules. The two dimensions of the matrices represent two time variables. Description of the routines, instructions for model use, and a sample problem are provided. The GMM has been used in developing DISTRO, a SIMPO-I model required by PRIMAR II, "Program to Improve Management of Army Resources". It has also been used to assist in the evaluation of policies on the overseas assignment sequence for officer personnel and on the reenlistment of members of the WAC for the Office of Plans and Programs, Office of Personnel Operations.

DD FORM 1 NOV 65 1473 REPLACES DD FORM 1473, 1 JAN 64, WHICH IS OBSOLETE FOR ARMY USE.

Unclassified
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
*Model simulation						
*Mass-flow model						
*Computer technology						
*Computerized routines						
Time-related variables						
*Modeling capabilities						
*Modeling techniques						
Statistical methodology						
*Matrix modeling						
Nodes						
Simulation tools; techniques						
Personnel - allocation, distribution, utilization						
Personnel - policy evaluation						

ii

Technical Research Report 1165

AD

SIMPO-I GENERAL MATRIX MANIPULATOR (GMM)

Joanne M. Witt and Adele P. Narva

Pauline T. Olson, Task Leader

STATISTICAL RESEARCH AND ANALYSIS DIVISION
Cecil D. Johnson, Chief

BEHAVIOR AND SYSTEMS RESEARCH LABORATORY

Office, Chief of Research and Development
Department of the Army

1300 Wilson Boulevard, Arlington, Virginia 22209

January 1972

Army Project Number
20065101M711

SIM II d-18

iii

BESRL Technical Research Reports and Technical Research Notes are intended for sponsors of R&D tasks and other research and military agencies. Any findings ready for implementation at the time of publication are presented in the latter part of the Brief. Upon completion of a major phase of the task, formal recommendations for official action normally are conveyed to appropriate military agencies by briefing or Disposition Form.

FOREWORD

The BESRL Work Unit, "Computerized Models for the Simulation of Policies and Operations of the Personnel Subsystem--SIMPO-I", was conducted by the Statistical Research and Analysis Division. The task constituted the initial undertaking of an operations research requirement described in the Army Master Study Program under the title, "A Simulation Model of Personnel Operations (SIMPO)" and is Project 2O065101M711, "Army Operations and Intelligence Analysis," under the auspices of the Army Study Advisory Committee. Sub-Work Units include: a) Operational Analysis of Personnel Subsystems; b) Cataloging and Integration of Existing Manpower Models; c) Development of Measures of System Effectiveness; d) Development of Modeling Techniques; e) Design and Programming of SIMPO-I; f) Application and Evaluation of Computerized Models; and g) Problem Oriented Language for Management.

The present Technical Research Report deals with the development of a computer model which can be used in evaluating many different military manpower systems. The model, the General Matrix Manipulator (GMM), is based on the movement of elements within or between matrices used to represent the different personnel categories according to supplied rules. The two dimensions of the matrices represent two time variables. Description of the routines, instructions for model use, and a sample problem are provided.

J. E. UHLANER, Director
Behavior and Systems
Research Laboratory

SIMPO-I GENERAL MATRIX MANIPULATOR (GMM)

BRIEF

Requirement:

To develop a generalized mass-flow model which can be used to simulate many different Army manpower systems by using an appropriately coded data deck.

Research Product:

The GMM is a modular set of computerized routines which move elements representing groups of persons within and among a variable number of matrices. Two time variables can be depicted--for example, time in tour and time in service, or time in grade and time in service. Movement is controlled by priority-of-fill cards input at simulation time and may be changed during the simulation period.

Utilization:

The GMM has been used in developing DISTRO, a SIMPO-I model required by PRIMAR II, "Program to Improve Management of Army Resources," Monitor Team of the SIMPO Steering Committee. It has also been used to assist in the evaluation of policies on the overseas assignment sequence for officer personnel and on the reenlistment of members of the WAC for the Office of Plans and Programs, Office of Personnel Operations.

SIMPO-I GENERAL MATRIX MANIPULATOR (GMM)

CONTENTS

	Page
INTRODUCTION AND BACKGROUND	1
THE GMM MODEL	2
TECHNICAL DISCUSSION OF MATRIX MODELING	3
Number and Relationship of Matrices	4
Transition from one Matrix to Another	4
Search Patterns	7
Updating	9
Uniform or Different-Sized Matrices	9
Matrix Capacitance	10
Model Input and Output	10
SIMPO-I GMM ROUTINES	11
Main Driver Program, MAINGMM	11
GMM SAMPLE PROBLEM	35
LITERATURE CITED	43
APPENDIX	45
DISTRIBUTION	75
DD Form 1473 (Document Control Data - R&D)	77

TABLES

			Page
Table	1.	Input to GMM	14
	2.	Variables input by user to GMM	17
	3.	Input to Program FILL	21
	4.	Definitions of variables for Program FILL	21

FIGURES

Figure	1.	Flows for Type A nodes	6
	2.	Node types in the GMM	6
	3.	Node Type B flows in the GMM	7
	4.	Search areas needed in the matrix manipulator flow model	8
	5.	Data deck setup for GMM	12
	6.	Data deck setup for program FILL	20
	7.	Logical flow of Subroutine TIMEUP	24
	8.	Storage of data by entry PACK	25
	9.	Flow chart for Subroutine SUMMARY	27
	10.	Flow chart for Subroutine FILLUP	29
	11.	Flow chart for Subroutine LOCK	30
	12.	Flow chart for Subroutine OUTSIDE	34

SIMPO-I GENERAL MATRIX MANIPULATOR (GMM)

The U. S. Army Behavior and Systems Research Laboratory Work Unit, "Simulation Models of Personnel Operations (SIMPO)," has developed a series of network flow models designed to help the managers of Army personnel make better estimates of accessions, training, promotion, assignment, and distribution needs and capabilities under a variety of policy considerations. The models developed are of two general types: mass flow models in which groups of similar persons are moved through a series of states in a deterministic manner, and entity flow models with which individuals and their attributes can be considered. In the SIMPO entity models, losses, gains, transfers, and other changes are stochastic (uncertain), the outcome of each simulated event depending upon the value of a random number generated by the computer. In the entity models, a number of different characteristics can be treated separately--each variable can be permitted to have many values and more variables can be considered. In bulk flow models, only a single value of a characteristic can be represented at a node. Additional time-related variables are monitored by using an array as the data file at the node.

The first mass flow model, DYNAMOD, was quite general in concept. There was to be, at each node represented in the model, a vector of numbers with position in the vector representing time at the node. During each simulated time period, the subsets of individuals then completing an assignment were transferred, losses were taken, the system vectors were updated, requirements were calculated, and new assignments were made. Four system variations were modeled using this concept, producing four separate computerized models (1).

New problems within the Army rotation system became so urgent with the development and subsequent replacement of large forces for Vietnam that the DYNAMOD concept was abandoned. Additional system complexity was modeled with matrix nodes and provision for greater flexibility in the selection of assignment priorities. Plans were made to develop the SIMPO-I "Grand" Model. The resulting model, now called the General Matrix Manipulator (GMM), has been especially designed to stress flexibility. It is particularly useful in evaluating the one-time problem as contrasted with problems that must be periodically reexamined and that can be handled more efficiently by models specially developed to simulate the appropriate system. Examples of special purpose SIMPO-I models are DYROM II (2) and ACGMOD (3). Another special purpose SIMPO-I mass flow model has received wide use with different rotation, training, and reenlistment problems. The Career-Noncareer Model (4) is a rotation model with the capability of evaluating transition from first-term to career status. The three special purpose models are more efficient for the system they represent than the GMM would be. The value of the GMM lies in its usefulness with diverse systems since it permits the analyst to make an early response to one-time questions on the effects of a policy change.

THE GMM MODEL

This generalized mass flow model was developed around the concept of maneuvering elements of square tabular data displays, corresponding to personnel categories, according to rules input at the time of model execution. Number of tables and interactions between the tables were to be flexible. Possible interactions were to be from a limited set of actions: all or a portion of an element, or several elements, of a table could be moved to the corresponding position in another table, or out of the system of tables; the movement could also be a shift to the first column and/or row of another table. The movements were to be made under constraint of an upper limit on the total number allowed in a table and a lower limit on the time of stay in the table. Updating the table from one simulated time interval to the next was to involve shifting each element of the table to the next row below and the next column to the right. System renewal elements were to be directed to the selected table(s) at time of model execution.

It was thought that problems in the functional areas of accessions, assignment, and reassignment including rotation, distribution, and promotion would be especially amenable to evaluation with such a model. Since the actual model of the system of interest would be dependent upon the flow rules supplied by the analyst/user of the programs, great flexibility for modeling different flow systems would be offered.

The modeling capabilities actually included in the present system of computer programs are these:

1. Monitoring of two different time measurements.

2. Variable number of main categories of personnel, i.e., groups of tabular displays.

3. Variable number of sub-categories of personnel--separate tabular displays.

4. Optional rules for transferring all or a portion of elements in one table to another table.

5. Variable sized tables--up to 48 rows by 48 columns.

The first of these capabilities, monitoring two time measurements, is accomplished by calling the two dimensions of the square table (array) time dimensions and by updating all elements in the array each time the system is advanced from one simulation period to the next. The particular time measure represented is optional but must be consistent through all parts of the simulated system. Thus, it is possible to simulate months, quarters, or years so long as all rates supplied are consistent with the same time period, and each row and column in the table represents a time step of the same order. Some time combinations relevant to Army personnel system problems are time-in-tour and time-in-system, time-in-grade and time-in-system, and time-in-tour and time-in-enlistment.

The next two capabilities are related to the number of classes or groups of people required to model a given system. An example of main categories would be short tour, long tour, and CONUS. Sub-categories simulated might be first, second, and third or subsequent short tours within the short tour main category; and before short tour, after one short tour, and after two or more short tours within the other two main categories. Another possibility might involve tour areas as main categories and grades as sub-categories. The model provides the option of upper limits on the total number in each separate sub-category. The model uses a symmetric arrangement of sub-categories within main categories; that is, if one main category requires 9 sub-categories, all main categories must have 9. (It is possible to have dummy sub-categories.) Since the model is disk based, the number of separate categories is limited by disk storage area rather than by computer main memory.

Capability 4, optional rules of transfer, is the most important feature of the model. Flow between the elements of any two tables is possible. The order in which the model considers the flows can be varied in any manner permitted by the set of rules used. Since each problem modeled requires its own set of rules, many different flow patterns can be described, making it possible to model quite different systems.

The fifth capability, variability in table size, makes it possible to consider different limits on tour lengths, or different limits on time in grade. In applying the size variation, a full-sized table is carried along in the computer by the model but a limited area of it is used. Maximum table size is 48 rows x 48 columns. The last row and column are used to accumulate elements advanced into them. Suppose tour length is only 12 months. The model would carry along the 48 x 48 area but would function within the 12 + 1, or 13, rows and 48 columns required by a 12-month tour. A rule of movement would need to be supplied to keep the 13th row empty if 12 months is the absolute upper limit of time allowed in the tour.

TECHNICAL DISCUSSION OF MATRIX MODELING

The GMM may be thought of as a collection of matrix nodes connected by a flow network. Development of routines that could be used flexibly to maneuver elements or combinations of elements from one matrix to another led to the examination of certain classes of problems. These are:

1. Number and relationship of the matrices
2. Transition from one matrix to another
3. Search patterns
4. Updating
5. Matrix size
6. Capacitance

In the following paragraphs, these problems are discussed and distinctions are drawn between the capabilities of the SIMPO-I GMM and that of a completely general matrix manipulator.

Number and Relationship of Matrices

In a personnel subsystem, there is usually a constraint on total number of members. This limit may apply to the lowest subset of individuals--as in the element of a matrix--or it may apply to the sum of several categories, the total number within a node or across several nodes. In a matrix manipulator, it should be possible to structure the system under either type of constraint. Either one or more matrices may be allowed in a cluster, and either the individual matrix or the cluster may be capacitated. The two matrix dimensions may be as large as the particular system requires but, in the interest of reducing simulation time, should be kept as small as possible. Also, the matrices may be as numerous as required by the system under consideration, limited only by disk area available and simulation time costs.

Transition from one Matrix to Another

In a time-dependent model, transition from a cell of one matrix to another cell of a second matrix can be accomplished by one or another of several flow types. Denoting the losing matrix node as C and the gaining matrix node as D, the flow types in terms of the element in D which may gain from an element in C are shown in the following table:

Flow Type	Element losing in C at time t	Element gaining in D at time t + 1
1	$c_{i,j}$	$d_{i+1,\ j+1}$
2	$c_{i,j}$	$d_{1,\ j+1}$
3	$c_{i,j}$	$d_{i+1,\ 1}$
4	$c_{i,j}$	d_{11}

The nodes in a GMM can be thought of as being located on a grid on which locations are specified by tour and level (grade, rank, or skill). When more than one node is located at a grid location (e.g., when several MOS's are interchangeable in one tour but not necessarily in other tours), these nodes are considered to be subtours of the same tour. The flow upward to higher levels within a tour constitutes the familiar feeder pattern, the flow across tours at the same level depicts the rotation phenomenon.

Thus, the GMM is a collection of nodes having one kind of flow (Type 1) within a command aggregation (the feeder pattern) and a second kind of flow (Type 2) across commands. These two kinds of flow are depicted in Figure 1 in connection with Type A nodes (matrices of time in command vs time in service). The two kinds of node envisaged for the GMM are shown in Figure 2. The second type of node (Type B) consists of matrices whose rows represent time in command and whose columns represent time in grade or skill level. Only one type of node will usually be present in a single flow pattern, although a third dimension could be depicted by increasing the number of matrices, in effect, combining the characteristics of A and B nodes in the same model.

Type 1 flows involve adding the amount taken from a feeder node to the equivalent cell of the gaining node. In Type A nodes, which are considered in the discussion that follows, neither time-in-service nor time-in-command changes as a result of the flow. Prescribed proportions can be taken from each row (representing those with equal time in service).

Type 2 flow is implemented by adding K persons to a cell in the gaining node that corresponds to the first column (least time in command and the row (time in service) from which the K persons were removed in the contributing node. Each cell in the matrix would then be advanced one row and one column. Nondeployability constraints relating to rotation status are imposed on this flow. In a distribution model having both Type 1 and Type 2 flows, more detailed distribution of assets to claimants could undoubtedly be made after the Type 1 flows and before the Type 2 flow. The rate of fill (constraints on flow into nodes) at each command aggregation could then be recomputed to reflect distribution priorities.

A Type 3 flow would represent a flow within a command via the feeder pattern among Type B nodes. These nodes have time in node (grade or skill level for an MOS or branch) as the vertical dimension of the matrix and time in command (tour) as the horizontal dimension. The Type 3 flows will have the effect of resetting time in node to zero and retaining the column location (time in command) of the receiving cell. This flow could be used to depict an officer promotion policy.

Type 4 flows are included here for the sake of completeness. The Type 4 flows would place all input into a node at the upper left-hand cell (i.e., with time in grade and time in command both set at zero). For a Type B node, this flow would represent promotion to fill a vacancy located in a different command. See Figure 3.

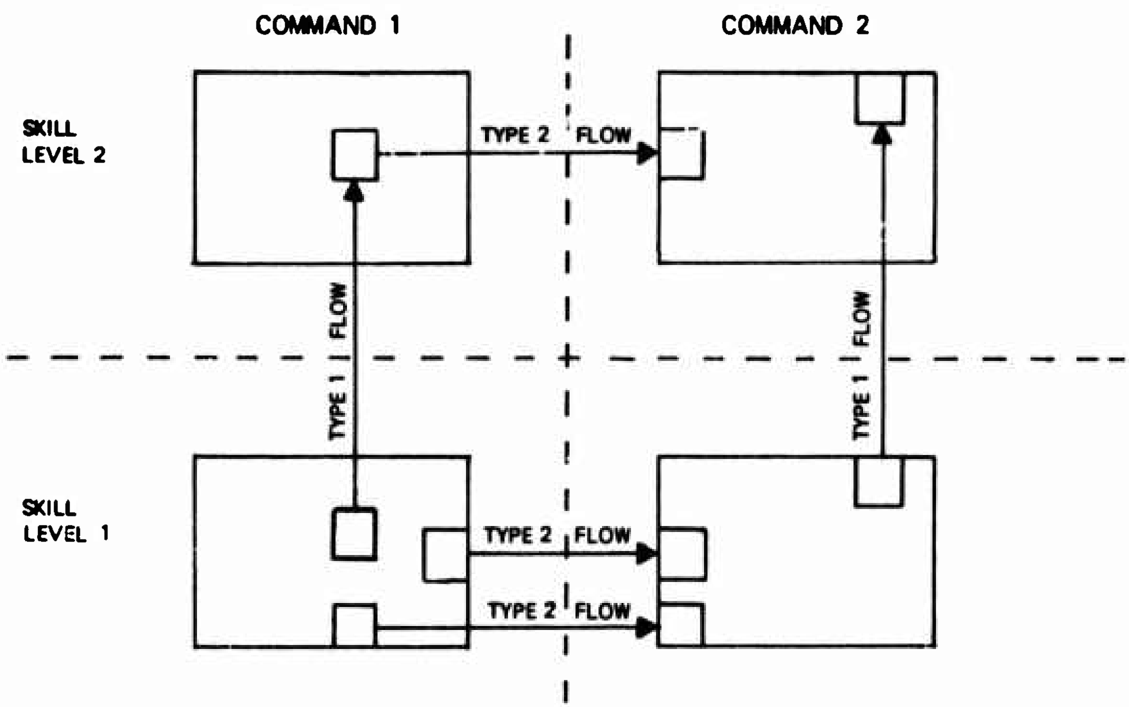

Figure 1. Flows for Type A Nodes

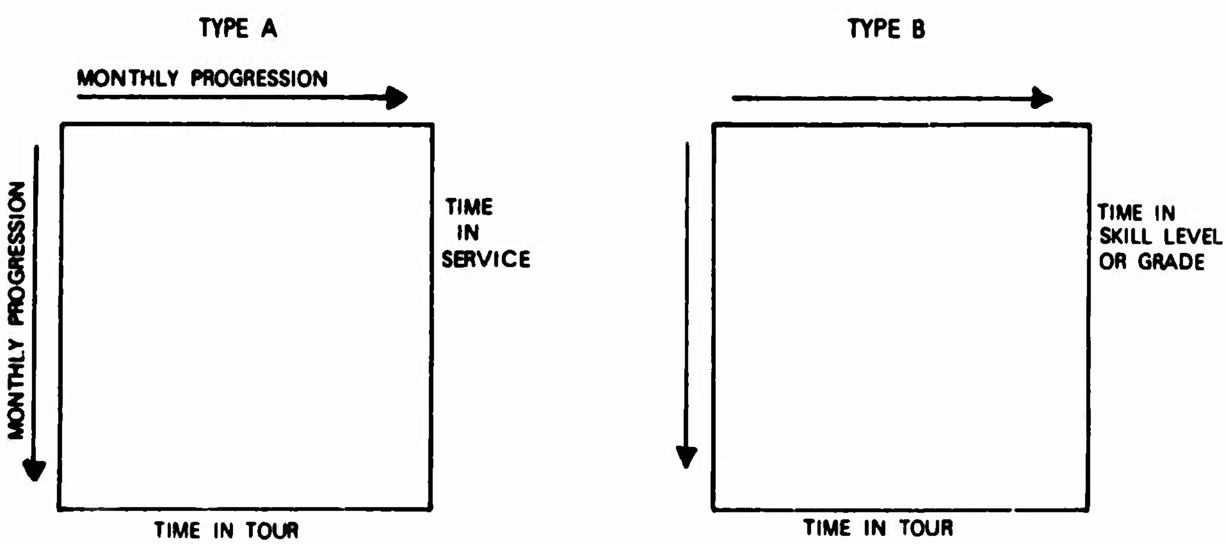

Figure 2. Node types in the GMM

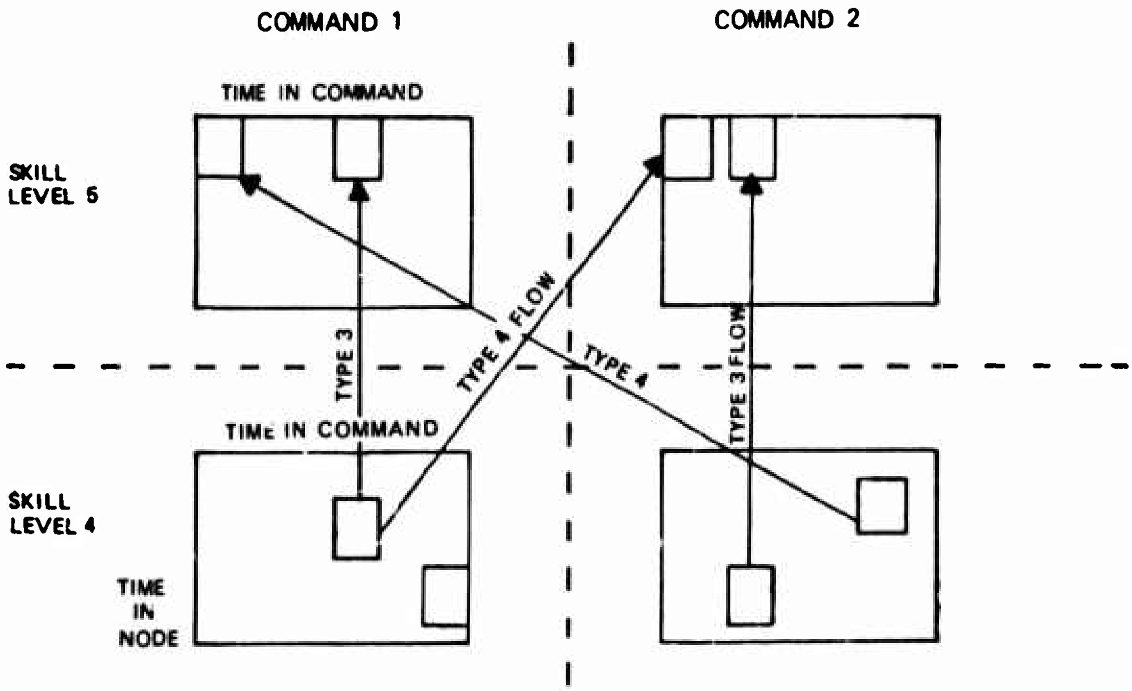

Figure 3. Node Type B flows in the GMM

Search Patterns

Flows between model nodes are under the control of priority-of-fill rules provided by the analyst. The order in which nodes are reached in extracting people to fill other nodes is determined by the sequence of the rules. Another aspect of patterns of search concerns the order in which elements of the same losing matrix are entered and removed. To simplify development of rules of flow in the manipulator, it is desirable that areas of the losing matrix be covered automatically by specifying appropriate transfer limits in terms of rows and columns (time) beyond which the cells of a particular matrix are not decremented to provide flow from the node. Direction of extraction from the transfer area is also important. In one instance, the search--and the successive emptying--begins with the highest numbered row (column) and works down, while in the other, search starts with the lowest numbered row and works up. Complete flexibility requires that the model user be given the option of covering specifically any of the shaded areas of the matrices in Figure 4, with search progressing in either direction. All these search areas can be covered by specifying upper and lower row and column boundaries (matrix (n) in the figure). Similarly, direction of search can be determined by specifying increasing or decreasing dimensionality for each matrix dimension.

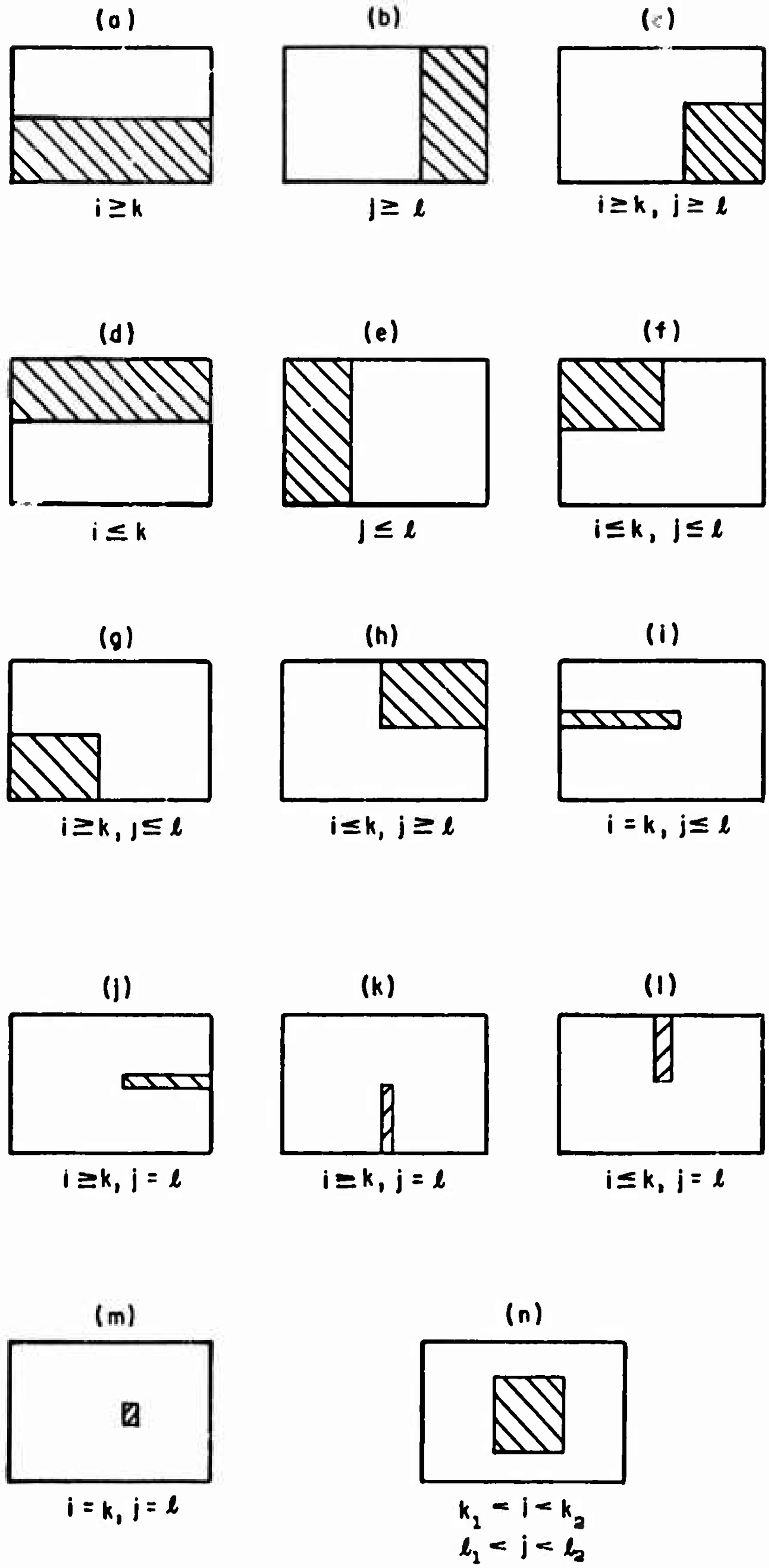

Figure 4. Search areas needed in the matrix manipulator flow model

As is often the case when plans for a model are being developed, the concepts of search patterns discussed above were not immediately apparent to the model developer. The SIMPO-I Matrix Manipulator makes possible flow Types 1, 2, and 3 with all searches starting at the upper time limit and progressing toward the lower.

Updating

Bringing time up one unit is a different problem in the mass flow models than in the entity models. In the mass flow models, position in the node array indicates time spent in the node, node cluster, or system. In the entity models, time in state is computable from state entry time and present simulation time. Moving the nth element of the mass flow node array to the (n+1)th position poses no problem until the array boundaries are encountered. This problem can be handled in the fixed length node by making certain the priority-of-fill rules move the element into the last position before updating takes place--or as a first step of the updating process. The node representing residual system members not in fixed-length nodes--or not in a high priority function--must either have safeguards that prevent the loss of the last element at updating or the node array must be so large that the last position always remains empty. In the GMM, the last element in each row or column is cumulative, and the time represented for this part of the matrix is "greater than t" where t is the nominal duration of the node on the single dimension.

Uniform or Different-Sized Matrices

The updating function is related to another basic concept for a matrix manipulator--that of size of array for each node. In modeling a personnel system, some nodes can be easily classified as a matrix, others are rightly a vector, still others an element (a pool). The Phase I GMM uses a matrix at each node. The matrices are square and all are the same size. Within the square matrix, it is possible to use a second limit (length of tour) on the number of rows in which an element other than a zero may appear. Suppose the CONUS tour is 24 months long and the maximum matrix size is 48 by 48. With 24 and 48 input as model parameters, individuals moving past 24 months in a CONUS assignment are accumulated in the 25th row. Updating on the other dimension continues during each subsequent period until the 48th position in the 25th row is reached, at which point further accumulation occurs. The effect is that the modeler can stipulate a 25 by 48 matrix at a particular node.

At the present time, it seems desirable to modify the existing mode to include the capability of handling variable dimensions for the nodes. The Phase I GMM uses a combination of maximum matrix size and matrix number sequence to reach the correct position on the disk where a matrix node is written. Changing the concept of the model from a symmetrical arrangement of sets and subsets of square matrices to a nonsymmetrical one composed of elements, vectors, and matrices as required by the system being simulated may mean major changes in all computer routines or complete restructuring of the model. While the SIMPO-II GMM will probably use variably dimensioned nodes, it is well to remember that the SIMPO-I GMM uses a square matrix at each node and accesses the disk by a combination of dimension and matrix sequence number.

In order to circumvent meaningless search of empty cells of the losing matrix, the SIMPO-I GMM keeps a record of the extreme matrix dimension at which non-zero elements are located. Search starts at the extreme non-zero element and works back toward row or column number one.

Matrix Capacitance

Just as provision must be made to fix an upper limit on the number of persons contained in a given matrix node (or set of matrix nodes), a truly general model must also consider the lower limit. In the present GMM, the lower limit is considered only in the special purpose DISTRO application, where distribution of available personnel to Army elements within a group of matrices is simulated by using a special computer routine with results of a GMM simulation period as input. Later developments (SIMPO-II) in the GMM will involve lower as well as upper limits for matrix capacity.

Model Input and Output

The GMM requires the starting data to be supplied to the model in the same configuration in which it is to be used. In simulating a complicated personnel system, data preparation is a tedious clerical task even when appropriate information is available. The necessary details are often not available in a form which allows the analyst to make the necessary summaries. Since the Army keeps its personnel records on tapes and makes extracts available for operational and study needs, it is feasible to provide routines which allow the GMM to obtain data without requiring intermediate hand processing. These routines are planned as a part of the SIMPO-II effort.

Planning appropriate model output for a general matrix manipulator is more difficult than for a specific model. System status for a specified time period is perhaps the most easily provided--and the most difficult to interpret, unless the simulated system is a very simple one. A summary based on flows between nodes or on the total number of persons in a given matrix node at a specified period is probably more useful information. The GMM package contains routines to obtain these two types of aggregated information. It is also possible to sum the elements in a given area of one or more matrices. For example, it would be possible to obtain the sum of all persons who have been in short tour less than 6 months, if time in tour were one of the time variables under consideration.

This capability is useful, too, when the first month of an assignment area is to be considered transient time and duty personnel are obtained from those there longer than a month.

SIMPO-I GMM ROUTINES

To insure flexibility and adaptability to special problem areas, the GMM is conceptualized as a network of matrix routines interwoven into a logical unit. As new specifications arise, this network of matrix routines can be expanded to include additional special function routines. The only major change necessary within the GMM, besides the development of special routines, will be the logical integration of the new routines into the main GMM driver program. This network concept has not completely materialized in the SIMPO-I version of the GMM, for the current main GMM program accomplishes some functions which will be accomplished by separate routines in the SIMPO-II GMM.

One main routine serves as a driver program for the entire GMM; it either performs a specific function or calls the appropriate subroutine which, in turn, performs this function. As a result, the main routine determines the logical basis of the simulation, and the validity of any model is directly related to the logic employed. The main routine driver and its interface with the individual subroutines is described in the following sections.

Main Driver Program, MAINGMM

MAINGMM, the driver program, is a logical group of statements which, using an iterative procedure, simulates each time period. These statements define the simulation logic by the order and manner in which they employ the separate GMM subroutines.

MAINGMM logic can be summarized in a series of steps, each representing a special function or event within the simulation process. These steps perform the following functions:

1. input simulation data
2. update the system
3. make initial transfers
4. calculate node assets
5. determine node requirements
6. fill node requirements
7. make final transfers
8. input additional personnel
9. increase time period

These steps are accomplished within MAINGMM in conjunction with the separate subroutines. In the SIMPO-I GMM, the MAINGMM driver performs steps 1, 2, 5, 8, and 9 without the aid of subroutines.

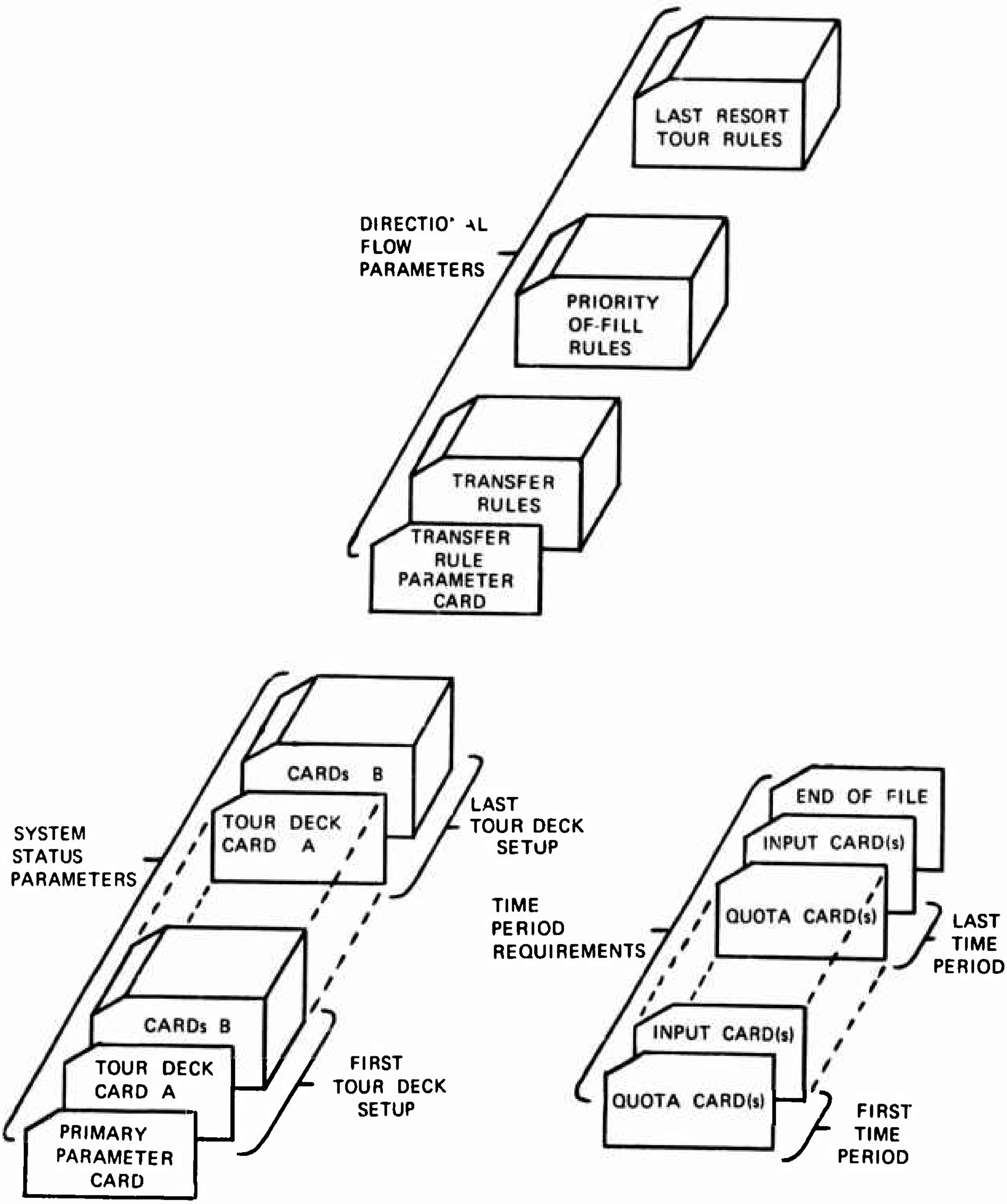

Figure 5. Data Deck Setup for GMM

Additional subroutines will take over these five functions in the SIMPO-II phase of the GMM. Subroutines TIMEUP, SUMMARY, FILUP, OUTSIDE, MX, X59B, and LOCK accomplish steps 3, 4, 6, and 7. Each of these nine steps and the individual subroutines are described in detail below in the order in which they occur in the iterative process.

Step 1. Input simulation data. The data input deck (Figure 5) can be divided into three sections, each defined according to its function within the simulation. These three input sections, outlined in Table 1 and described in Table 2, determine respectively: the status of the system, the directional flow of personnel within the system, and the special monthly system requirements.

Status of the system. The primary parameter card and the tour deck setup describe in detail the system that will be simulated. The primary parameter card sets the boundary conditions within which the system operates. That is, it determines the number of nodes and node clusters, the number of flow rules, and specific program options.

The tour deck setup cards describe each node, its characteristics, and personnel within that node. Tour deck card A data serve as parameters for each node, defining the node in relation to other nodes and describing its length and loss rate. Tour deck card B data describe personnel distributed within the node. Currently, an accurate distribution of personnel according to time served within a particular node is practically impossible to obtain. Therefore, a uniform distribution of time served in the node is usually assumed.

Since the tour deck setup card B data--the personnel distribution--comprises the bulk of the GMM data, card B data are stored on an on-line disk to save computer storage space. These data can easily be retrieved as needed. Inputting data in segments onto a permanent storage disk is definitely more practical than reading all the data from cards into the core storage area, saving computer time as well as computer storage space.

The MAINGMM has an option to read the card B data from either a permanent storage disk or from the cards directly. If the data have previously been stored on the permanent disk, the user inputs only ITT cards A; the program atuomatically retrieves the card B data from the permanent storage disk. Only if the user turns on sense switch 3 will the program read both the card A and card B data from cards. In either case, as the card B data are input in segments to the storage area within core, card B data are transferred to a temporary working storage disk. Since several computer runs are usually made to compare output from the different model options, the permanent disk storage prevents duplicative reading of the same data cards onto the temporary working disk. Instead, the data can be read directly from the permanent storage disk and transferred to the temporary disk.

Table 1

INPUT TO GMM

Card Description	No. of Cards	Variable	Columns	Format
		System Status Parameters		
Primary Parameter	1	ID	1-3	A3
		NTOUR	4-7	I4
		ITT	8-11	I4
		FIRST	12-15	I4
		LAST	16-19	I4
		IFILL	20-23	I4
		NP	24-27	I4
		MAXSUB	28-31	I4
		CIOS	32-35	I4
		JC	36-39	I4
		IC	40-43	I4
		MAXLEN	44-47	I4
		LRT	48-51	I4
		PDW	52-57	F6.4
		IDISTON	58	I1
		IPUNCH	59	I1

Tour Deck Setup consists of ITT (found in columns 8-11 of Primary Parameter Card) groups, each with one Card A followed by M Cards B (M = ((MAXLEN - 1)/20 = 1) truncated * MAXLEN).

Input Tour Deck Cards B only if Sense Switch 3 is on. Otherwise input ITT Cards A.

Tour Deck Card A	1	TYPE	1-10	I10
		SUB	11-20	I10
		ACTUAL	31-40	I10
		NUM	41-50	I10
		LENGTH	51-60	I10
		OUT	61-66	F6.3

Construct M (M = ((MAXLEN - 1)/20 = 1) truncated * MAXLEN) Cards B: (MAXLEN is found in columns 44-47 of the Primary Parameter Card). This means the matrix data are input row by row with MAXLEN rows and (MAXLEN - 1)/20 + 1 truncated cards per row.

Table 1 (Continued)

Card Description	No. of Cards	Variable	Columns	Format
Tour Deck Cards B	M	SYST(1,1)	1-4	I4
		SYST(1,2)	5-8	I4
		.		.
		.		.
		.		.
		SYST(1,20)	77-80	I4
		SYST(1,21)	1-4	I4
		.		.
		.		.
		.		.
		SYST(1,40)	77-80	I4
		SYST(1,41)	1-4	I4
		.		.
		.		.
		.		.
		SYST(1,MAXLEN)		I4
		SYST(2,1)	1-4	I4
		.		.
		.		.
		.		.
		SYST(MAXLEN, MAXLEN)		I4
		Directional Flow Parameters		
Transfer Rule Parameter Card	1	NOTT	1-4	I4
		NOFIRST	5-8	I4
		NOLAST	9-12	I4
Transfer Rules	NOTT	OUTTT	1-10	I10
		OUTST	11-20	I10
		OUTTO	21-30	I10
		OUTSO	31-40	I10
		PERD	41-50	I10
		PERDTO	51-60	I10
		PCT	61-66	F6.3
Priority-of-fill Rules	NP	INTOUR	1-8	I8
		INSUB	9-16	I8
		OUTTOUR	17-24	I8
		OUTSUB	25-32	I8
		AFTER	33-40	I8

Table 1 (Continued)

Card Description	No. of Cards	Variable	Columns	Format
		PEROUT	41-46	F6.3
		PER	47-52	F6.3
		REP	53-58	F6.3
		IOD	59-61	I3
		IGRADE	62-64	I3
		ITYPE	65-67	I3
Last Resort Tour Rules	LRT	LSTRSTO	1-5	I5
		LSTRSTT	6-10	I5
		LSTRSTS	11-15	I5

Monthly Requirements

Monthly requirements consist of one group of two sets of cards, one set of N Quota cards and one set of M Input cards, for each time period to be simulated.

(N = ((NTOUR - 1)/10 + 1 truncated; M = ((CIOS - 1)/10 + 1) truncated).

Monthly requirements consist of one group of two cards, one Quota card and one Input card, for each time period to be simulated

Card Description	No. of Cards	Variable	Columns	Format
Quota Card	N	NEEDS(1)	1-8	I8
		NEEDS(2)	9-16	I8
		.		.
		.		.
		.		.
		NEEDS(NTOUR)		I8
Input Card	M	IOS(1)	1-8	I8
		IOS(2)	9-16	I8
		.		.
		.		.
		.		.
		IOS(CIOS)		I8

Table 2

VARIABLES INPUT BY USER TO GMM

Variable	Definition
ID	Simulation Identification (3 characters)
NTOUR	Number of node clusters
MAXSUB	Maximum number of nodes within each node cluster (No. of subtours)
ITT	Maximum number of nodes (NTOUR * MAXSUB)
FIRST	First period to be simulated
LAST	Last period to be simulated
IFILL	Parameter governing filling of requirements: If IFILL less than 1--have NTOUR requirements, one for each node cluster. If IFILL .GE. 1--have ITT requirements, one for each node.
NP	Number of priority of fill rules
MAXLEN	Maximum length of system in time periods
CIOS	Number of categories input from outside the system
PDW	Percentage of requirements which may be filled from outside the system
ITT of the following:	
TYPE(I)[a]	Node cluster designation (Numbered one to NTOUR)
SUB(I)	Node category designation (Numbered one to MAXSUB)
ACTUAL(I)	Number of entities in node
NUM(I)	Number of nodes within the node cluster
LENGTH(I)	Length of node in time periods
OUT(I)	Percentage of entities lost from node at end of 12 time periods (yearly attrition rate)
NOTT	Number of possible horizontal node to node movements (transfers)
NOFIRST	Number of initial transfers or horizontal movements
NOLAST	Number of final transfers or horizontal movements

Table 2 (Continued)

Variable	Definition
NOTT of the following:	
OUTTT(J)[b]	Node cluster to which horizontal movement is made (gaining cluster)
OUTST(J)	Node within node cluster to which movement is made (gaining node)
OUTTO(J)	Node cluster from which horizontal movement is made (losing cluster)
OUTSO(J)	Node within node cluster from which movement is made (losing node)
PERD(J)	Row of node (particular time period) from which horizontal movement is made
PERDTO(J)	Row of node to which horizontal movement is made
PCT(J)	Percentage of entities within each cell of losing node which are available for horizontal movement
NP of the following:	
INTOUR(K)[c]	Node cluster to which movement is made (gaining cluster)
INSUB(K)	Node within node cluster to which movement is made (gaining node)
OUTTOUR(K)	Node cluster from which movement is made (losing cluster)
OUTSUB(K)	Node within node cluster from which movement is made (losing node)
AFTER(K)	Row (time in tour) of node from which movement is to be made (see explanatory note in IOD(K))
PER(K)	Percentage of requirements of node to which movement is possible which may be made from the node cluster (OUTTOUR) and node (OUTSUB) specified
PEROUT(K)	Percentage of node (OUTTOUR) that may be used as fill for (INTOUR) node to which movement is specified (Assumed to be 100% if blank on input)
IOD(K)	If IOD .EQ. 1--movement is only from time period (row) indicated (AFTER(K)) If IOD .NE. 1--movement is from time period MAXLEN through AFTER(K) i.e. if AFTER .EQ. 12, movement takes place from row 13 only if IOD .EQ. 1; if IOD .NE. 1, from MAXLEN to 12th rows
REP(K)	If REP .EQ. 1--may fill NEEDS with 100% of those available from outside the system If REP .NE. 1--NEEDS are filled with PDW% of those available from outside the system

Table 2 (Continued)

Variable	Definition
ITYPE(K)	Determines how data will be input into a node If ITYPE .LE. 2--input will be into row 1, column I. If ITYPE .EQ. 3--input will be into row I, column 1. If ITYPE .GE. 4--input will be into row 1, column 1.
LRT	Number of categories of personnel available as input to system which must be emptied and saved in the event they are not needed by algorithm
<u>LRT</u> of the following:	
LSTRSTT(L)[d]	Tour to which LSTRSTO(L) input is sent
LSTRSTO(L)	Designation of cell containing input to system
LSTRSTS(L)	Node within node cluster to which LSTRSTO(L) input is to be sent
NEEDS(M)[e]	If IFILL .EQ. 0--Read in NTOUR node cluster requirements for the period being simulated
NE(I)[a]	If IFILL .EQ. 1--Read ITT node requirements for period being simulated
IOS(N)[f]	Number input from outside available to the system

[a] I = 1,ITT; [b] J = 1,NOTT; [c] K = 1,NP; [d] L = 1,LRT; [e] M = 1,NTOUR; [f] N = 1,CIOS

To input the tour deck card B information onto a permanent storage disk prior to the simulations, the user employs <u>Program FILL</u>. Program FILL writes the card B data onto a disk in one of two ways: it either zeroes the entire matrix node and then enters the non-zero elements into the node, or it enters the elements into an existing matrix node. By entering elements into an existing node, the user holds computer time at a minimum when he is modifying small parts of the data. Figure 6 illustrates the data deck setup. Table 3 describes the specific data construction and Table 4 defines each of the input variables for Program FILL.

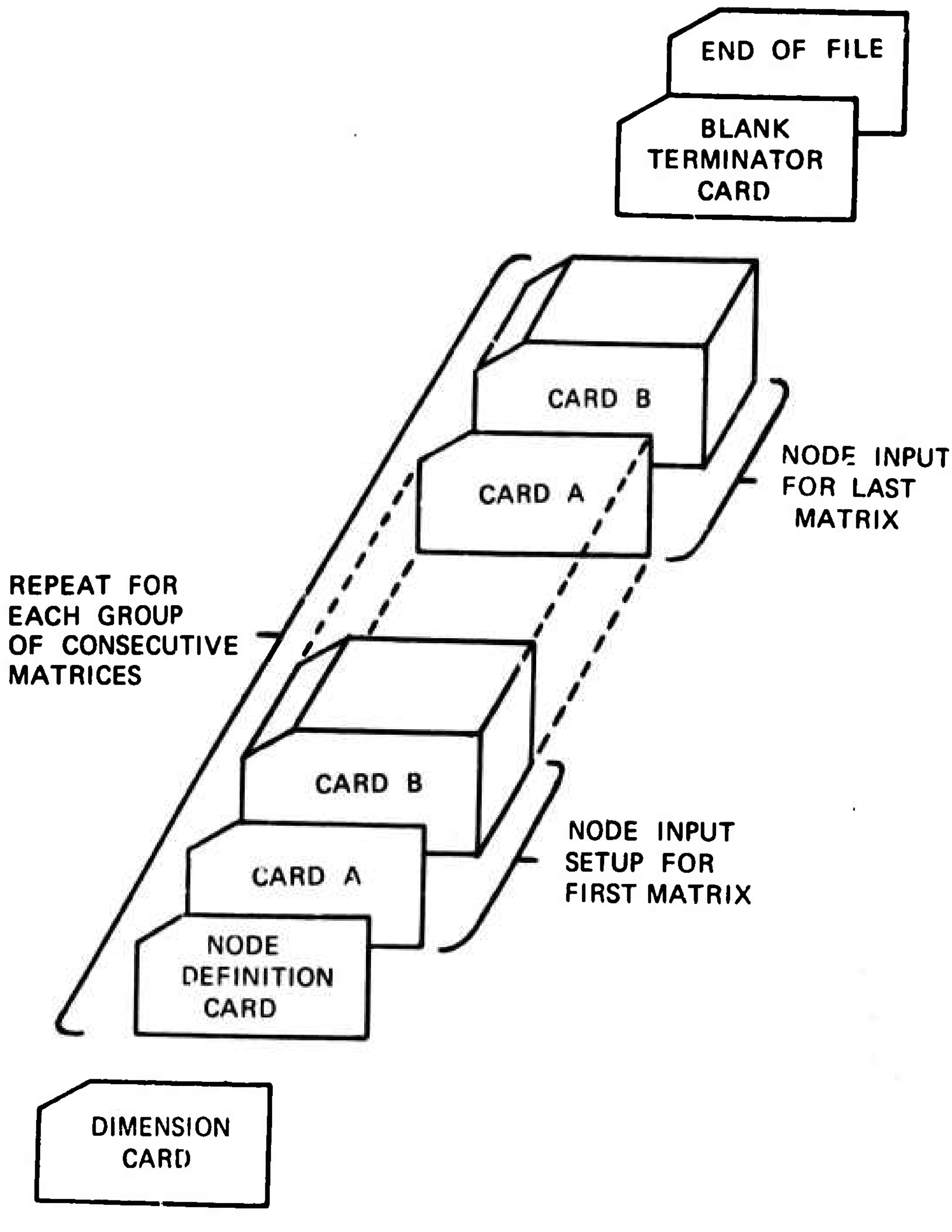

Figure 6. Data deck setup for Program FILL

Table 3

INPUT TO PROGRAM FILL

Card Description	No. of Card	Variable	Columns	Format
Dimension	1	M	1-4	I4
Node Definition	1	ISTART	1-4	I4
		ISTOP	5-8	I4
Node Input Setup consists of ISTOP-ISTART+1 groups, each with one Card A and K Cards B.				
Card A	1	K	1-4	I4
		IZERO	5-8	I4
Card B	K	N	1-6	F6.0
		I	7-8	I2
		J	9-10	I2
		PER	11-12	F2.2
Repeat Node Definition and Node Input Setup cards for each group of nodes to be input to system.				
Program Terminator	1	-none-	1-80	Blank

Table 4

DEFINITIONS OF VARIABLES FOR PROGRAM FILL

Variable	Definition
M	Dimension of a square matrix. In the GMM data, this is equal to the greatest time in the system.(MAXLEN).
ISTART	Number of starting matrix
ISTOP	Number of ending matrix
K	Number of data cards to be input to matrix
IZERO	If IZERO .EQ. 0, matrix will be zeroed, then data entered.
	If IZERO .EQ. 1, data will be entered into existing matrix.
N	Value of the matrix element
I	Starting row of the matrix entered
J	Starting column of the matrix entered
PER	Percent of N entered into each element of a diagonal vector starting at element I,J. If PER .EQ. 0, 100% enters element I,J.

Flows within the system. Three types of personnel flow are available within the GMM--direct movement between nodes, movement between nodes based on node requirements, and movement to an initial node position.

The initial and final transfer rules determine direct movement between nodes. This movement can be from any row and column of a losing node to any row and the same column of a gaining node. These data cards specify which elements of the losing and gaining nodes within the node clusters and which percentages of these elements will be moved from and to the respective nodes. Normally, this movement represents the Type 1 flow of personnel across skill levels within command elements prior to the assignment process. For example, a transfer rule might specify that 10 percent of the personnel in the eighth month of the ST node must move to the next skill level within the ST, or to the outside of the system.

Movement between nodes based on node requirements, described by the priority-of-fill rules, represents the Type 2 flow or the rotation phenomenon. These rules direct the search and assignment procedure by specifying the order in which the program searches certain nodes for personnel to fill other node requirements. For example, in order to fill the ST requirements, the rules might initially direct the search to personnel who have completed 24 months in a CONUS node, after which search might be directed to personnel who have completed 24 months in the LT.

Movement to an initial node position, i.e., to the first row and first column of a gaining node, is described by the Last Resort Tour Rules. This movement represents the Type 4 flow. The Last Resort Tour Rules currently specify where unassigned trainee input will go. For example, new trainees who are not assigned to fill the ST requirements will be assigned to a holding node in CONUS. Both data segments--status of the system and flows within the system--are input once at the start of the simulation and cannot be modified. In the SIMPO-II GMM, the user will be able to modify these parameters during the computer run.

Special monthly system requirements. The last section of input data consists of two cards for each time period, one card stating the node quota or authorized strength and one the input to the system from the outside. Based on the magnitude of these monthly node requirements, personnel flow through the system according to the priority-of-fill rules. If specified in the priority-of-fill rules, the quotas may be partially filled by the input to the system which represents new output from the training schools.

Step 2. Update the system. System updating simulates the passage of a single time period within the personnel system. Since both time in node and time in system locate personnel within the nodes, the updating procedure must include both these time dimensions. Therefore, the updating procedure moves all personnel over one row and down one column in the node corresponding to completion in the real system of another month in node and in system.

A reverse step-wise process accomplishes this system updating. All personnel move over one row and down one column starting with the last row and column and working backwards to the first row and column in the node. The latest months in the node and in the system equal the parameters LENGTH and MAXLEN, respectively. Initially, all personnel who have completed LENGTH months in the node accumulate, after loss rates have been applied, in the LENGTH + 1 row in the node and in a column corresponding to their time in the system + 1; personnel who have completed MAXLEN months in the system accumulate in the MAXLEN + 1 column in the node and in a row corresponding to their time in the node + 1. The LENGTH row and MAXLEN column in the node are then set equal to zero, after which the personnel in the LENGTH - 1 row and the MAXLEN - 1 column are updated to the LENGTH row and the MAXLEN column of the node. This reverse process continues moving all personnel diagonally within the node until the first row and first column equal zero. At this time the updating process is complete.

Step 3. Make initial transfers. Initial personnel transfers may represent losses from the system and/or movements between nodes within the system. These transfers usually represent a Type 1 flow across skill or grade levels within command aggregates. Since these initial transfers occur outside the regular assignment process, they are not a function of node requirements. Instead, the amount of flow is a direct function of the percentages specified in the transfer rules. To make these transfers, MAINGMM releases control to subroutine TIMEUP, which transfers personnel from any row and column of a losing node to any row and the same column of a gaining node. Subroutine TIMEUP removes PCT percent of the personnel who are in the PERD time period from the OUTSO node within the OUTTO node cluster (see Table 2 for definition of terms). These personnel will enter the PERDTO time period of the OUTST node within the OUTTT node cluster. In order to specify movements representing losses out of the system, the gaining node coordinates, OUTTT and OUTST, must equal zero. Figure 7 illustrates in detail how TIMEUP accomplishes these different transfers. In the SIMPO II GMM, a new subroutine with the capability of transferring personnel from any row and column of a losing node to any other row and column of a gaining node will replace subroutine TIMEUP.

To transform records of the initial transfer movements into usable form, Subroutine TIMEUP releases control to Entry Pack of Subroutine X59B. This routine PACK, written in COMPASS, compresses or packs a cumbersome data description of the transfer movements into a data description which can be easily handled and stored. Entry PACK packs eight fixed-point 24-bit computer words into three fixed-point 24-bit computer words (Figure 8). These three words contain data describing the node and node cluster from and to which the personnel were moved, the number of personnel input into the node and lost to the system, and the time periods in the input node and in the system.

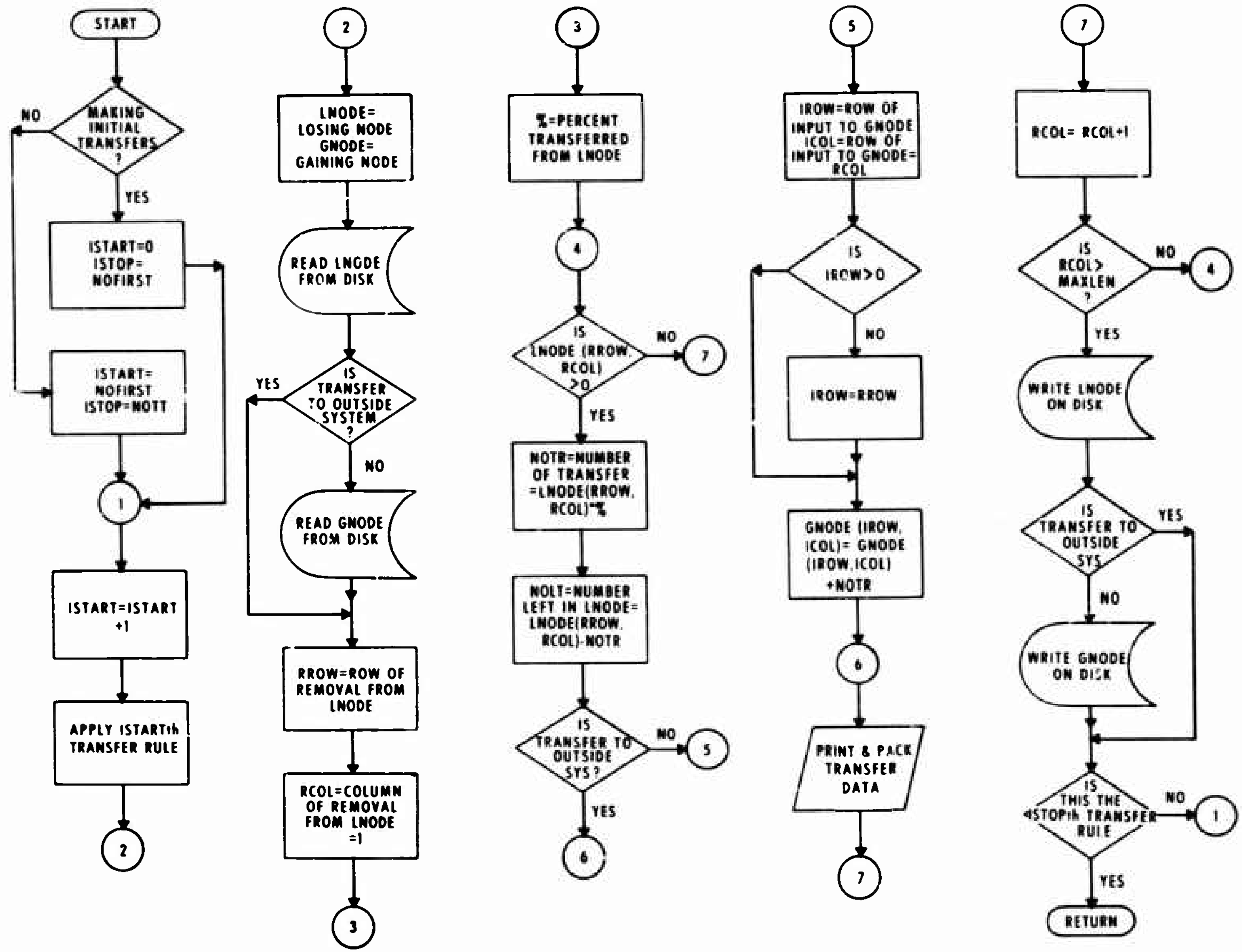

Figure 7. Logical Flow of Subroutine TIMEUP

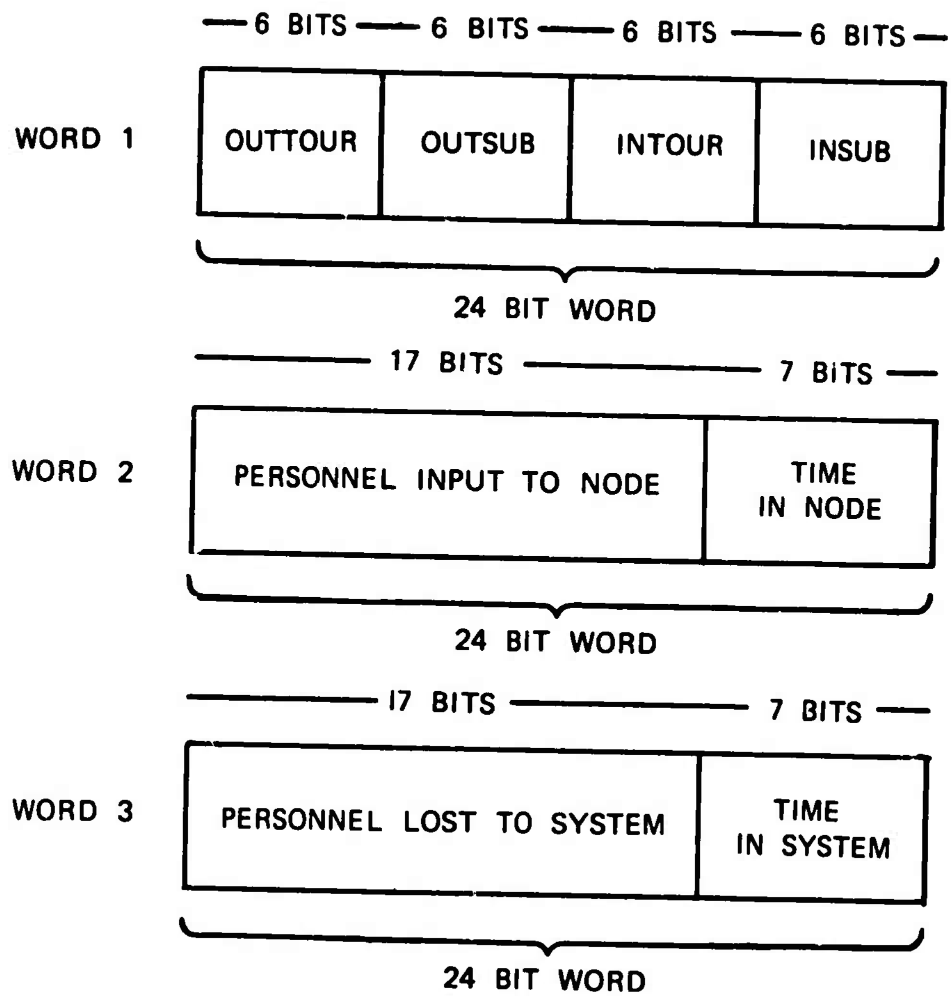

Figure 8. Storage of data by entry PACK

After PACK has compressed the data description into three computer words, TIMEUP regains control and stores these three words on a temporary disk. This cycle of first packing and then storing the data occurs after each transfer of personnel.

Step 4. Calculate node totals. All personnel assets within each node and node cluster are matched with node and node cluster requirements in Step 5 to determine the number of additional personnel needed. To calculate these asset totals, MAINGMM releases control to subroutine SUMMARY.

Subroutine SUMMARY has the capability of obtaining sums of partial and/or complete matrices and groups of matrices. Elements in the rows and columns of any one matrix may be added to elements in the rows and columns of any other matrix.

In the GMM, where only sums of complete nodes are calculated, subroutine SUMMARY parameters NUMELEM and INDIV determine how these sums will be calculated (Figure 9). MAINGMM sets NUMELEM equal to two and INDIV equal to the node to be summed. Personnel in rows one to LENGTH and in columns one to MAXLEN are summed for each node. This sum represents all personnel who are serving in the node during a particular time period. It does not, however, include personnel in row LENGTH + 1, who have completed LENGTH time periods in the node and are waiting for new assignments.

Subroutine SUMMARY can also obtain sums of partial nodes or groups of nodes. A MATGRPS vector, which has NUMELEM elements, controls the order in which the individual and aggregate node sums are calculated. The value of each element in the MATGRPS vector refers to the node to be summed. A zero value for an element directs the subroutine to obtain an aggregate sum of all node totals calculated since the last zero element. For example, if NUMELEM were equal to 3 and the values of the three MATGRPS elements were 1, 2, and 0, respectively, the subroutine would sum the first and second nodes separately and then add these two sums together. In this example, a total of nodes 1 and 2 could represent all AUS and RA personnel serving in the short combat tour area.

To obtain sums of partial nodes, other summation guidelines of inclusive rows and columns for each node must be input as elements of the following vectors: BEGROW, ENDROW, BEGCOL, and ENDCOL. Disregarding zero elements in the MATGRPS vector, the order of other elements within the four guidline vectors corresponds to the order of the elements in the MATGRPS vector. The first element in each of the BEGROW, ENDROW, BEGCOL, and ENDCOL vectors refers to the first row, last row, first column, and the last column to be summed in the first MATGRPS node. If the first element in BEGROW were equal to 2, in ENDROW to 12, in BEGCOL to 1, and in ENDCOL to 48, then the first node sum would contain all personnel in the second to the twelth time periods of the node and the first to the forty-eighth time periods in the system. This sum could represent AUS non-transients in the short tour area.

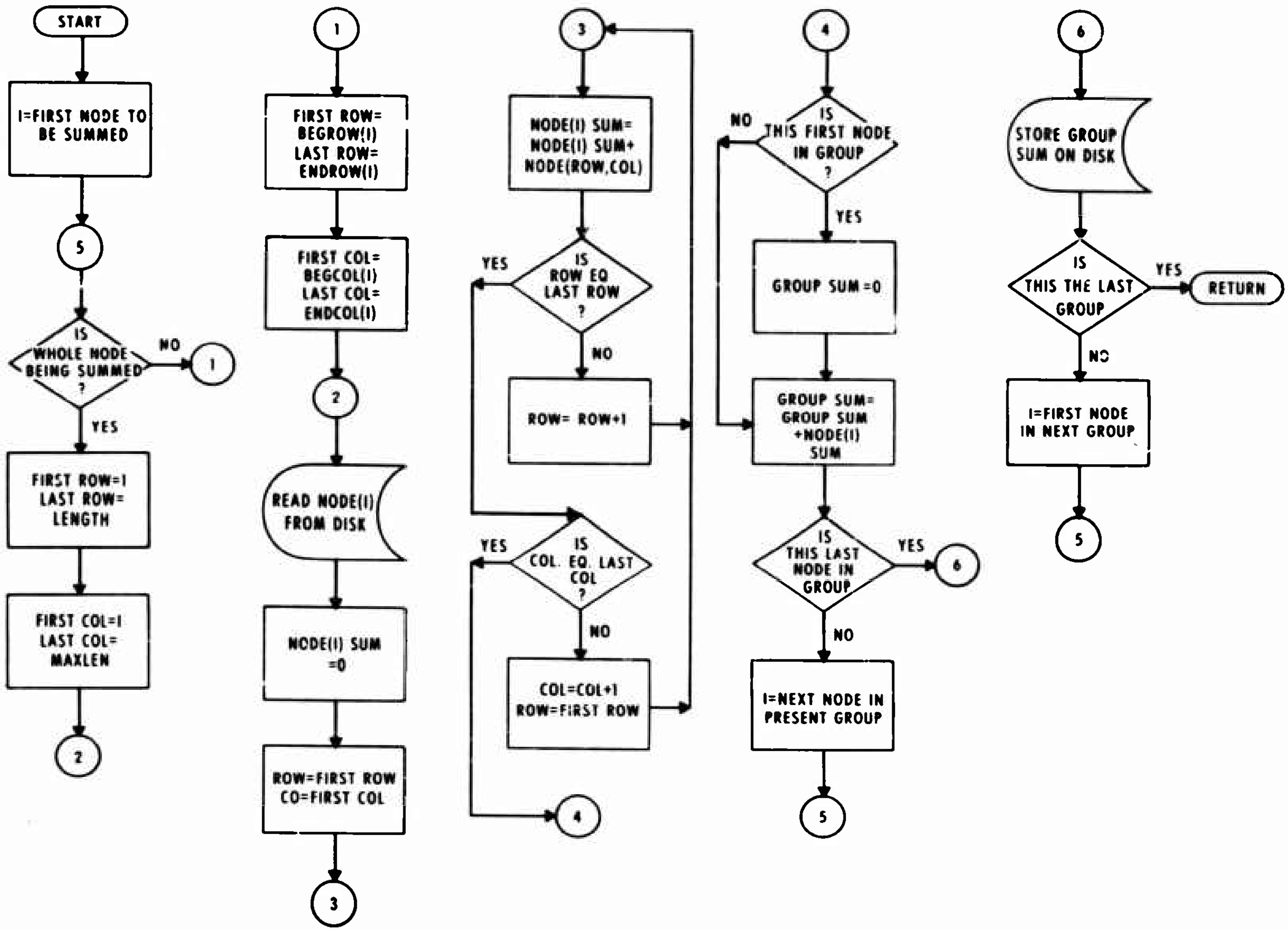

Figure 9. Flow chart for Subroutine SUMMARY

Step 5. Determine requirements. To determine node and node cluster requirements, MAINGMM subtracts the assets in each node (calculated in step 4 by subroutine SUMMARY) from the node requirements input for the same time period. If additional personnel are needed to fill these requirements, the program continues to step 6. Other, it executes step 7.

Step 6. Fill requirements. Once the node cluster requirements have been determined, MAINGMM relinquishes control to subroutine FILUP, which searches for personnel to fill these requirements.

Subroutine FILUP uses an identical procedure to apply each priority-of-fill rule in the order in which each rule is input to the program (Figure 10). Initially, FILUP checks to see if the gaining node has any requirements. When these requirements are positive, FILUP determines how the personnel search procedure will be handled. If the losing node is inside the system, subroutines FILUP and LOCK search for personnel within the node. If the personnel are from outside the system, subroutine OUTSIDE handles the search.

When the losing node is within the system, FILUP first seeks personnel who have completed LENGTH time periods in the losing node and are available for reassignment. If the requirements for the gaining node are still positive after using all of these available personnel, FILUP begins to remove personnel before they have completed LENGTH months in the losing node. In order to remove personnel within a node, FILUP calls subroutine LOCK, a routine which locates personnel within a losing node, removes them, and inputs them into a gaining node.

Initially subroutine LOCK searches for personnel within the LENGTH row and column MAXLEN to one of the losing node. After each group of personnel has been located and removed from the losing node, subroutine LOCK adds them to the gaining node. In subroutine LOCK, the time period in the system, or column, of the gaining node is identical to that of the losing node. The time period in the node, or row, however, becomes row one in the gaining node regardless of what it was in the losing node.

If there are not enough personnel in the LENGTH time period to fill the node requirements, LOCK searches in a similar manner in the LENGTH - 1 row. This reverse row-by-row procedure continues until all node requirements are met, or until LOCK has searched in the AFTER row (Figure 11). Control then returns to FILUP.

To decrease the amount of computer time needed for searching the entire node system column dimension of MAXLEN to one, LOCK uses a COMPASS function MX.

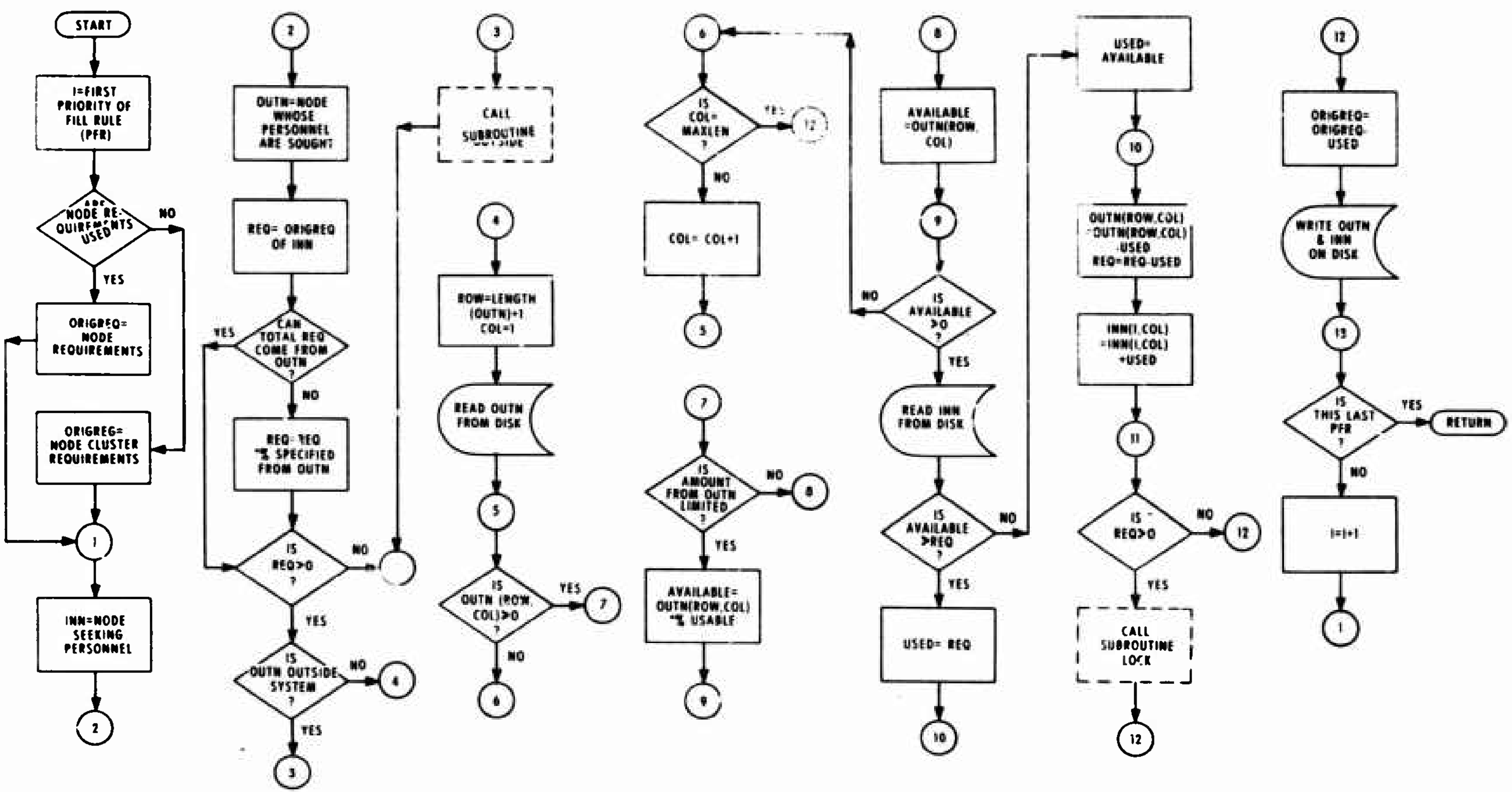

Figure 10. Flow chart for Subroutine FILLUP

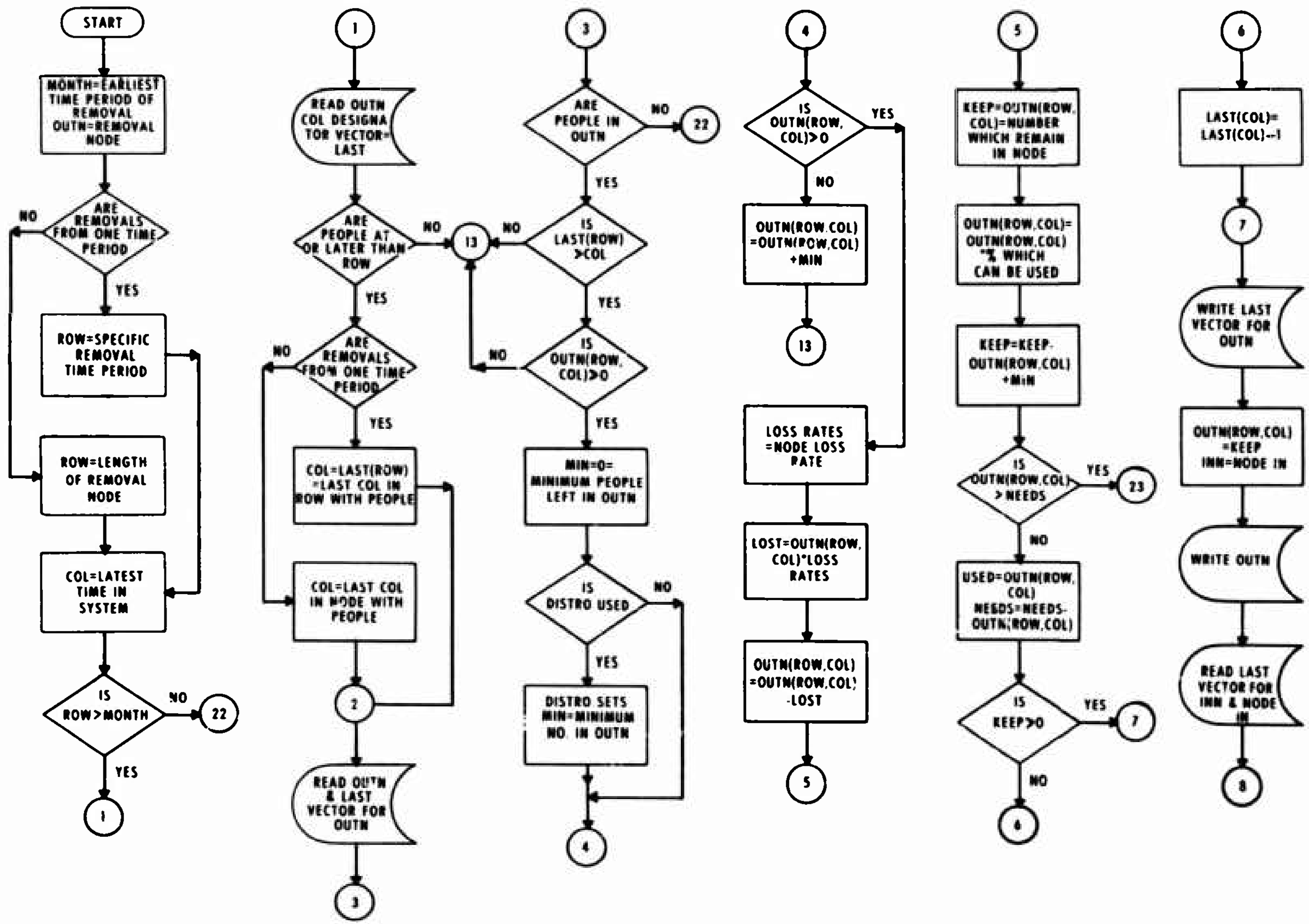

Figure 11. Subroutine LOCK

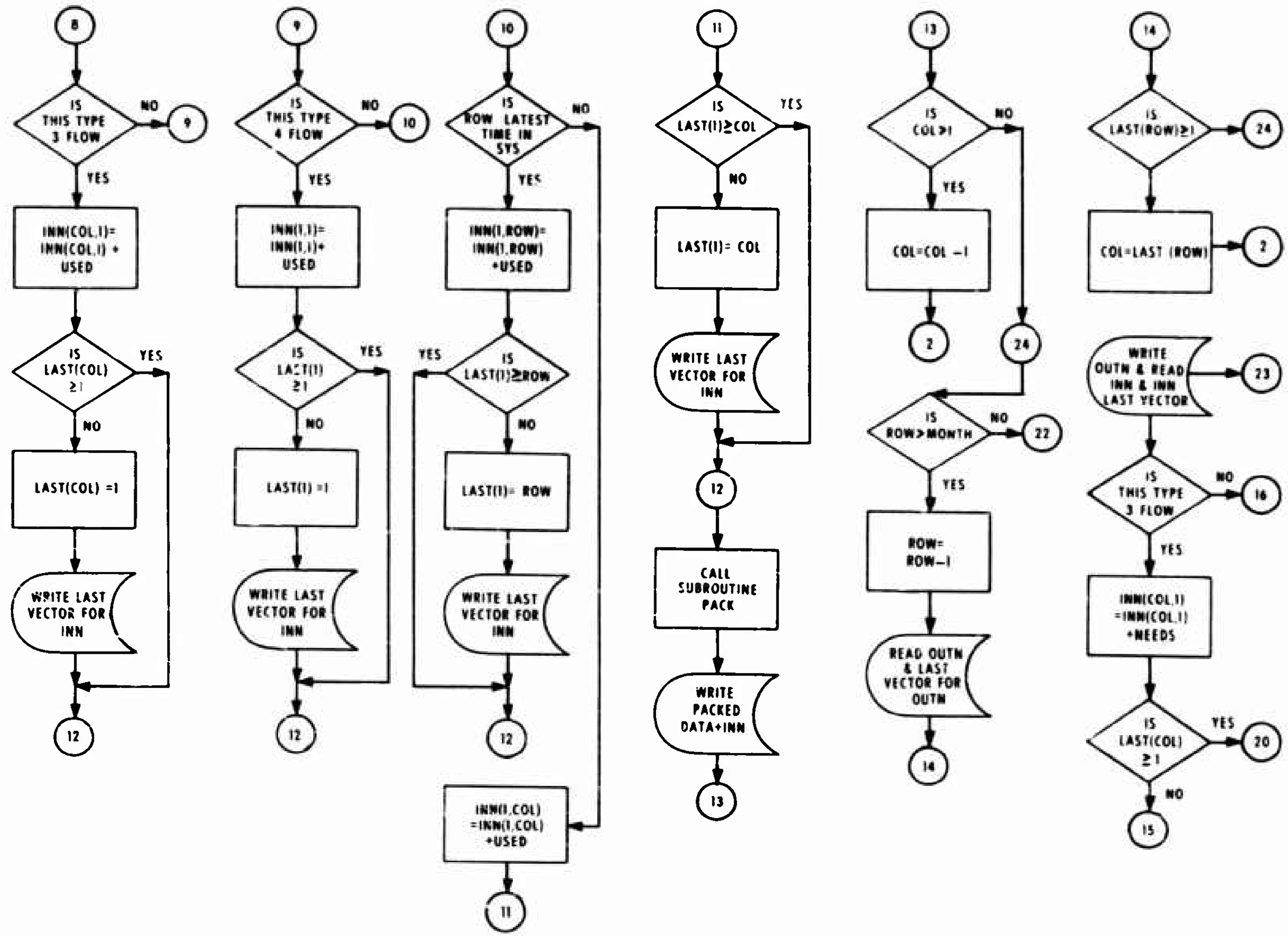

Figure 11 continued.

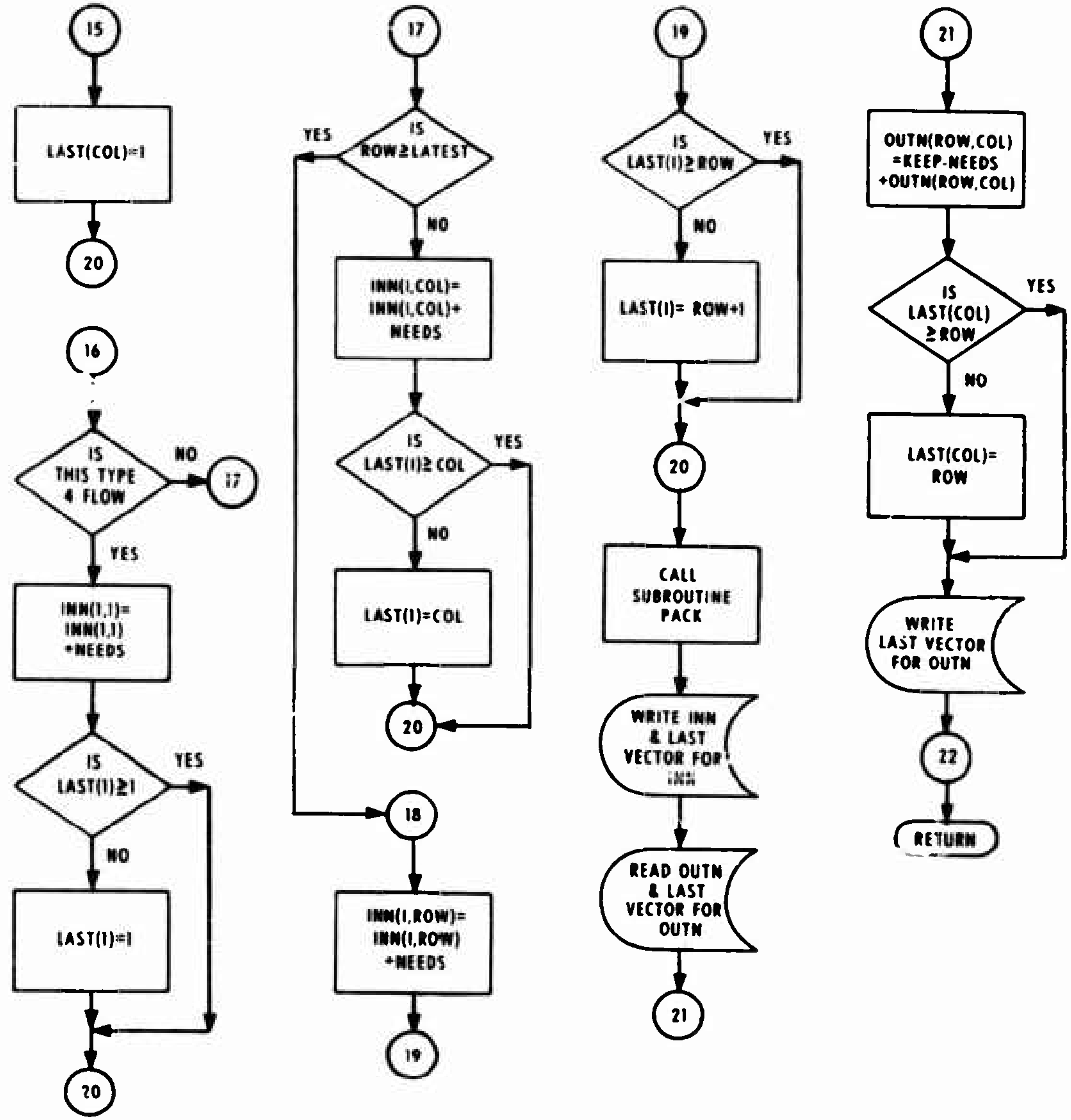

Figure 11 continued.

Function MX determines the latest time period in the system or the last column where there are personnel. For each node there is an INE vector whose positions correspond to specific rows in the node (MX sets each INE element) with elements equal to the last column in that row where personnel are found. For example, the element INE(X) refers to the X row and the INE(X) column in row X of the losing node. By searching backwards in row X from the INE(X) column instead of from the MAXLEN column, subroutine LOCK reduces search time significantly.

If the losing node represents personnel input from outside the system, FILUP relinquishes control to subroutine OUTSIDE, instead of to subroutine LOCK. As shown in Figure 12, subroutine OUTSIDE has an option of either fixed or variable input from outside the system. Under the fixed input option, the amount of input from the outside for this time period, specified in the monthly requirements input card, is the maximum which can be allocated against all gaining node requirements. Under the variable input option, however, the maximum amount of input from the outside is a function of the requirements for the individual gaining nodes. These two options can either test the operating level of a system with a fixed amount of input or determine the number of new personnel needed to maintain a certain operating level.

These new personnel enter the first row and first column of the gaining node, after which control returns to subroutine FILUP.

Data descriptions of each personnel movement within subroutines FILUP, LOCK, and OUTSIDE are packed, using subroutine X59B-Entry PACK, and then are stored on a temporary disk. This cycle of transferring control to FILUP at the beginning of each priority-of-fill rule and then to LOCK or OUTSIDE, if necessary, continues until every priority-of-fill rule has been applied.

Step 7. Make final transfers. Final transfers are handled identically to initial transfers by subroutine TIMEUP. The difference is merely one of timing. The final NOFIRST + 1 to NOTT transfers occur after all personnel assignments have been made against the requirements, whereas the initial 1 to NOFIRST transfers occur prior to the assignment process. These final transfers round out the system before the next time period begins by reassigning all remaining personnel who must be moved. For example, when a person completes a combat tour (ST), he must be reassigned regardless of the requirements in other tour areas. Thus, as a final transfer movement, personnel who have completed the ST and who have not been reassigned might be transferred to a CONUS tour.

Step 8. Input personnel from outside into the system. During each time period of the simulation, the user may input into the system a fixed number of new personnel representing new trainees. If this new input is not used within the time period to fill the node requirements, it must enter nodes within the system. These nodes are called Last Resort Tours.

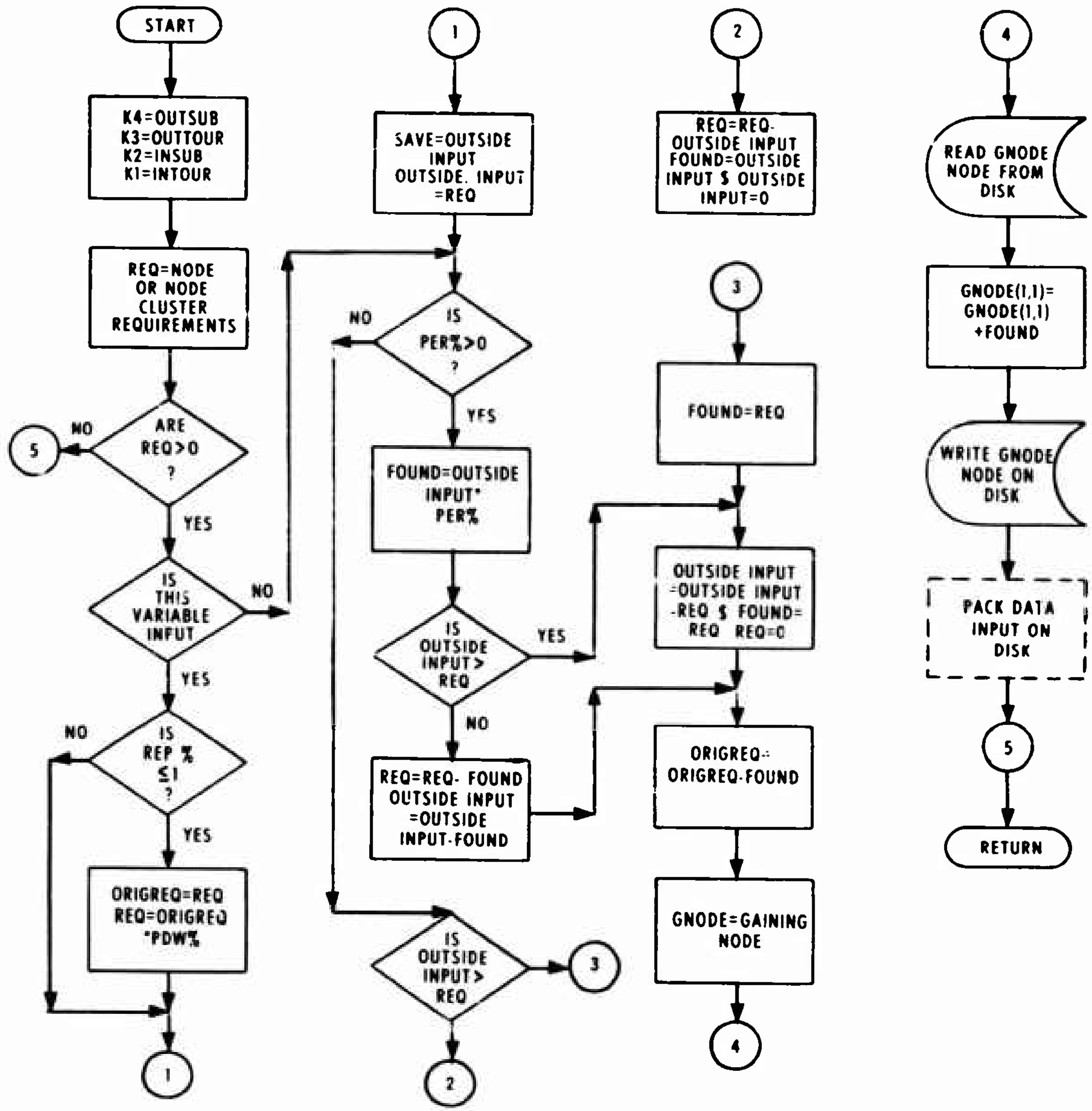

Figure 12. Flow chart for Subroutine OUTSIDE

MAINGMM inputs these unassigned new trainees into the first row and first column of the gaining nodes specified in the Last Resort Tour Rules. This position in the node represents their first time period in the node and their first time period in the system. As with other movements of personnel, these movements are packed and stored on a temporary disk.

Step 9. Increase time period. At the end of the eight previously mentioned simulation steps, the variable INTEG, representing time periods, is increased by one. A new time period within the personnel system begins.

After Step 9, the cycle repeats itself for the next time period. The cycle ends when INTEG equals LAST, the last time period simulated. At this time, MAINGMM calls Entry UNPACK of subroutine X59B, which reverses the process of PACK by unpacking the data descriptions of the flow movements within the system. These flow data and other data specified by the user are then output by an on-line printer.

GMM SAMPLE PROBLEM

A problem for GMM proposed by the Plans and Programs Office of the Office of Personnel Operations concerned the feasibility of management's proposed policy limiting overseas assignments. The proposal had been made to eliminate all third overseas assignments except for those persons volunteering for such duty or for individuals who had received an interjectory assignment in Continental United States. Very limited information was available on a data base for a sample subsystem. In a total inventory of 1078 persons, 310 were assigned to short tour, 119 to long tour, and 649 to CONUS. Of the 310 in short tour, 100 had served in long tour immediately prior to the short tour assignment while 210 had been in CONUS. The 119 in long tour were 107 from short tour and 12 from CONUS. The system was to be considered closed--no losses or gains. (Since an intermediate level personnel system was being considered, this assumption was not bad--losses to the real system would be replaced with persons having had similar assignment history and the total personnel inventory would remain the same.) The question posed was: Will it be possible to maintain the authorized overseas manning levels while at the same time allowing 18 months in the CONUS tours, if no persons are required to serve three consecutive overseas tours?

Considering the inadequacy of the furnished data base, the problem was not complex enough for a simulation model. No more information was available than that required by the steady-state models described in the early reports on manpower models by Sorenson and Olson (5,6). Pages 44 and 45 of the Nomogram report show estimates of the possible CONUS tours under two conditions: 1) with no sequential overseas assignments and 2) with all long tour assignees going to a short tour assignment while all short tour assignees return to CONUS for an assignment there. With no sequential overseas assignments, the nomogram shows that duration of the CONUS tour would fall somewhere between 18 and 24 months. Under the second condition--with two sequential assignments--the CONUS tour would be somewhat longer than 24 months, but less than 30 months.

Since SIMPO-I research analysts wished to test the adaptability of the GMM, the decision was made to set up a model of this problem with the GMM. For modeling purposes, the system was structured into three main tour areas, short tour, long tour, and CONUS. Subtours were determined by prior history: short tour after CONUS, short tour after long tour, long tour after CONUS, long tour after short tour, CONUS after short tour, CONUS after long tour, and CONUS before overseas assignment. Since the GMM uses a symmetrical arrangement of subtours within main tour areas, three subtours were needed in each main tour. Tour numbers were assigned by the following design:

		Previous Assignment		
		ST	LT	CONUS
Present	ST	1,1	1,2	1,3
Assignment	LT	2,1	2,2	2,3
	CONUS	3,1	3,2	3,3

Tours 1,1 and 2,2 are dummy tours. The starting data show them empty and no priorities are included for transferring persons to them during the simulation. Maximum, minimum, and actual starting tour durations in months are shown below:

	Maximum	Minimum	Actual
ST	12	12	12
LT	36	12	18
CONUS	36	12	18

Assignment priorities used were:

Into Subtour	From Subtour	After	Part of quota to be filled
1,3	3,3	36	100%
1,3	3,2	36	"
1,2	2,3	36	"
1,3	3,3	18	"
1,3	3,2	18	"
1,3	3,1	18	"
1,3	3,3	12	"
1,3	3,2	12	"
1,3	3,1	12	"
1,2	2,3	12	"

Into Subtour	From Subtour	After	Part of quota to be filled
2,3	3,3	18	100%
2,3	3,1	18	"
2,1	1,3	12	"
2,3	3,3	12	"
2,3	3,1	12	"
2,3	3,2	12	"
2,1	1,3	12	"
2,1	1,2	12	"
3,1	1,1	12	"
3,1	1,3	12	"
3,1	1,2	12	"
3,2	2,1	36	"
3,2	2,3	36	"

For the 24-month simulation, the manning levels for the three main tour areas were to be:

Month	Short Tour	Long Tour	CONUS
1	310	119	679
2	310	119	679
3	310	119	679
4	310	119	679
5	310	119	709
6	310	119	709
7	310	119	709
8	310	119	739
9	310	119	739
10	310	119	739
11	310	119	739
12-36	310	119	769

Changes in the designated manning level for CONUS were made to test observed shortages output by the model and to make sure persons ending overseas tours would be moved by the model to a different assignment.

Data Deck Description

Card 1	Columns	Variable[a]	Value in Sample Problem
(Parameter card describing the limits of the system being modeled)	1-3	(Identification of Sample)	MAJ
	4-7	NTOUR	3
	8-11	ITT	9
	12-15	FIRST	1
	16-19	LAST	20
	20-23	IFILL	0
	24-27	NP	23
	28-31	MAXSUB	3
	32-35	CIOS	0
	36-39	JC	6
	40-43	IC	0
	44-47	MAXLEN	48
	52-57	LRT	0
	58	IDISTON	0
	59	IPUNCH	1

[a] See Table 2 for definitions

Cards 2-10 Subtour Parameter Cards[a]

Variable	Cols	Card Number 2	3	4	5	6	7	8	9	10
Main tour area	1-10	1	1	1	2	2	2	3	3	3
Subtour	11-20	1	2	3	1	2	3	1	2	3
Quota	21-30	0	0	0	0	0	0	0	0	0
Actual	31-40	0	100	210	107	0	12	0	0	649
Number of Subtours	41-50	3	3	3	3	3	3	3	3	3
Maximum Length of Subtour	51-60	12	12	12	36	36	36	47	47	47
Loss rate	61-70	0	0	0	0	0	0	0	0	0

[a] In this problem the data matrices were already on a computer disk and were called in when needed.

Cards 11-33 Priority-of-Fill Rules

		Into Tour Area	Into Subtour	Out of Tour Area	Out of Subtour	After (months)	Percent[a] Available
Cols		1-8	9-16	17-24	25-32	33-40	41-46
Card Nr.	11	1	3	3	3	36	1000
	12	1	3	3	2	36	1000
	13	1	2	2	3	36	1000
	14	1	3	3	3	18	1000
	15	1	3	3	2	18	1000
	16	1	3	3	1	18	1000
	17	1	3	3	3	12	1000
	18	1	3	3	2	12	1000
	19	1	3	3	1	12	1000
	20	1	2	2	3	12	1000
	21	2	3	3	3	18	1000
	22	2	3	3	1	18	1000
	23	2	1	1	3	12	1000
	24	2	3	3	3	12	1000
	25	2	3	3	1	12	1000
	26	2	3	3	2	12	1000
	27	2	1	1	3	12	1000
	28	2	1	1	2	12	1000
	29	3	1	1	1	12	1000
	30	3	1	1	3	12	1000
	31	3	1	1	2	12	1000
	32	3	2	2	1	36	1000
	33	3	2	2	3	36	1000

[a] Read with three decimal places.

Cards 34-69 Tour Area Quotas

Card No.	Tour Area 1 (Cols 1-8)	2 (Cols 9-16)	3 (Cols 17-24)
34	310	119	679
35	310	119	679
36	310	119	679
37	310	119	679
38	310	119	709
39	310	119	709
40	310	119	709
41	310	119	709
42	310	119	709
43	310	119	709
44	310	119	709
45-69	310	119	769

Card 70 End of file card.

To put the data matrices on the computer disk, a special program called FILL was used. FILL develops a data matrix from input consisting of matrix maximum dimensions and individual cell values, together with the row and column for each cell.

Reassignments actually simulated by the model are shown below:

	To Short Tour		To Long Tour		To CONUS	
Month	from CONUS	from LT	from CONUS	ST	from ST	from LT
1	30				30	
2	30				30	
3	25				25	
4	25				25	
5	25				25	
6	25				25	
7	25				25	
8	25				25	
9	25				25	
10	25				25	
11	25				25	
12	25				25	
13	30				30	
14	30				30	
15	25				25	
16	25				25	
17	25				25	
18	25				25	
19	25		5		25	5
20	25		6		25	6
21	25		6		25	6
22	25		6		25	6
23	25		6		25	6
24	25		6		25	6
25	29	1	7		30	6
26	29	1	7		30	6
27	24	1	7		25	6
28	24	1	7		25	6
29	24	1	7		25	6
30	24	1	7		25	6
31	24	1	7		25	6
32	24	1	7		25	6
33	24	1	7		25	6
34	24	1	7		25	6
35	24	1	7		25	6
36	24	1	7		25	6

The average CONUS tour duration for overseas replacements was up from 18 months at the start of the simulation to over 21 months at the end of the first simulated year and to over 23 months at the end of the second year. No sequential overseas assignments were necessary during that time. During the third year simulated, the CONUS tour dropped somewhat, but not below 18 months unless some persons were assigned to two sequential overseas tours. These results are in agreement with those obtained from the Nomograms. It is possible to simulate replacements to short tour from long tour. Such runs were not made with the GMM, but results could be expected to follow the tour lengths computed for the Nomogram using the same options.

For another application of the GMM see the DISTRO model (7).

LITERATURE CITED

1. Witt, Joanne M. and Adele P. Narva. SIMPO-I Dynamic Army Model (DYNAMOD). BESRL Research Study 70-2. May 1970.

2. Olson, Pauline T. DYROM II: SIMPO-I Model representing Army upper enlisted grades. BESRL Research Study 69-2. March 1969.

3. Olson, Pauline T. ACCMOD: A SIMPO-I dynamic flow model to project enlisted accession needs. BESRL Research Study 68-2. April 1968.

4. McMullen, Robert L. SIMPO-I Career-Noncareer Model. BESRL Technical Research Report 1162. June 1970.

5. Sorenson, Richard C. and Pauline T. Olson. Manpower rotation policy models. BESRL Technical Research Note 172. June 1966.

6. Olson, Pauline T. Nomograms for Army manpower policy evaluation. BESRL Technical Research Report 1147. June 1966.

7. Witt, Joanne M. SIMPO-I DISTRO--Distribution Rotation Model. BESRL Research Memorandum _______. (In preparation)

APPENDIX

COMPUTER LISTINGS OF GMM ROUTINES

Preceding page blank

```
      PROGRAM MAINGMM
C          APRIL 1967          NARVA
C        FILUP          NARVA          APRIL 1967
C        OUTSIDE        NARVA          APRIL 1967
C        LOCK           NARVA          APRIL 1967
      COMMON        NOFIRST   ,NOLAST   ,NOTT     ,ITRTIME  ,INTEG
      COMMON        IP        ,INF(48)  ,IEND     ,OUTTT(200)
      COMMON        OUTST(200),OUTTO(200)         ,OUTSO(200)
      COMMON        PCT(200)  ,PERD(200),JK       ,SYST(48,48)
      COMMON        PA(100)   ,NAV(100) ,LENGTH(100)        ,OUT(100)
      COMMON        INTOUR(220)         ,INSUB(220)         ,OUTTOUR(220)
      COMMON        IOO(220)  ,OUTSUB(220)        ,AFTER(220)
      COMMON        PER(220)  ,RATE(100)          ,NEEDS(10),NE(10,10)
      COMMON        IOS(10)   ,NED(10)  ,NEE(100) ,IFILL    ,MAXLEN
      COMMON        NTOUR     ,NP       ,CIOS     ,PERDTO(200)
      COMMON        PDW       ,LSTRSTO(10)        ,LSTRSTT(10)
      COMMON        LSTRSTS(10)         ,ITT      ,MAXSUB   ,ACT(10)
      COMMON        NSTT(10)  ,INOS(10,10)        ,IGRADE(220)
      COMMON         REP(220) ,ITYPE(220)         ,PEROUT(220)
      COMMON        IDISION   ,ISUM     ,GRPSUM(100)        ,PRIO(100)
      COMMON        BEGROW(100)         ,ENDROW(100)        ,BEGCOL(100)
      COMMON        ENDCOL(100)         ,MATSUM(100)        ,MATGRPS(100)
      COMMON        TYPE(100),SUB(100)  ,NUM(100) ,ACTUAL(100)
      COMMON        NPRLEV    ,NT       ,IHOLD    ,LEN      ,LEVEL    ,M
      COMMON        NCRNODE(100)        ,MIN(100) ,GRPINPR(100)
      COMMON        MAXDEPL(100)
      COMMON IBI(1000)
      INTEGER       TYPE      ,SUB      ,ACTUAL   ,ENDCOL
      INTEGER       CIOS      ,SYST     ,OUTSUB   ,OUTTOUR  ,AFTER
      INTEGER       ACT       ,OUTTO    ,OUTSO    ,OUTTT    ,OUTST
      INTEGER       PERD      ,GRPSUM   ,BEGROW   ,ENDROW   ,BEGCOL
      INTEGER       FIRST     ,PERDTO
C          DEFINITIONS OF TERMS USED IN SIMULATION PROGRAM
C
C
C
C         ID= RUN IDENTIFICATION
C
C         NTOUR= NUMBER OF OVERALL TOUR TYPES
C
C         ITT= NUMBER OF TOURS(TOTAL NUMBER) TO BE SIMULATED
C
C         FIRST=BEGINNING TIME PERIOD
C
C         MAXSUB=MAXIMUM NUMBER OF SUBTOURS
C
C         NP= TOTAL NUMBER OF PRIORITIES INPUT
C
C         NOUT(NTOUR X MAXSUB) = TOTAL OUTPUT FROM EACH TOUR AND SUBTOUR
C         AVAILABLE FOR ANOTHER TOUR
C
C         LOSSES(NTOUR X MAXSUB) = TOTAL NUMBER OUTPUT FROM EACH TOUR,
C          SUBTOUR LOST FROM SYSTEM
C
C         MIDOUT( NTOUR X MAXSUB) = TOTAL OUTPUT FROM EACH TOUR, SUBTOUR
C         DURING MONTH AVAILABLE FOR ANOTHER TOUR
C
C         MIDLOSS( INTOUR X MAXSUB) = TOTAL OUTPUT FROM EACH TOUR, SUBTOUR
C         DURING MONTH LOST FROM SYSTEM
C
```

Preceding page blank

```
C           IFILL=  PARAMETER USED TO DETERMINE IF QUOTAS ARE FOR TOUR OR
C              IF= 0 HAVE NTOUR QUOTAS
C              IF LESS THEN OR GREATER THAN 0 HAVE ITT QUOTAS
C
C
C           C1=NUMBER OF CRITICAL TOURS
C
C           C2= NUMBER OF CRITICAL TOUR TYPES * 2
C
C           CIOS= NUMBER OF CATEGORIES INPUT FROM OUTSIDE
C           HOW=
C
C           MAXLEN=MAXIMUM LENGTH OF TIME IN SYSTEM
C
C           I= 1,ITT
C
C           TYPE(I)=TOUR TYPE
C
C           SUB(I)=SUBTOUR TYPE
C
C           QUOTA(I)=QUOTA OF TOUR OR SUBTOUR(DEPENDS ON IFILL PARAMETER)
C
C           ACTUAL(I)=ACTUAL NUMBER OF MEN IN TOUR OR SUBTOUR#
C           DEPENDENT UPON IFILL PARAMETER
C
C           NUM(I)=NUMBER OF SUBTOURS IN TOUR
C
C           LENGTH(I)= LENGTH OF SUBTOUR
C
C           OUT(I)=PERCENTAGE OUTPUT FORM SYSTEM AFTER 1 YEAR  FROM SURTOUR
C
C       PEROUT= PERCENTAGE OF OUTOUR AVAIL FOR FILL TO INTOUR NEEDS
C
C-----ITYPE = TYPE OF MOVEMENT INTO NEW TOUR
C------0,1,2 = (1,I)   INPUT
C----- 3 = (I,1)  INPUT
C----- 4 = (1,1)  INPUT
C
C
        N1=N2=N3=0
        READ 1, ID,NTOUR,ITT,FIRST,LAST,IFILL,NP,MAXSUR,JC,IC,CIOS,MAXLEN
      1,LRI,PDW,IDISTON,IPUNCH
C       IF IFILL=1,QUOTAS ARE FOR NODES,IF IFILL=0,QUOTAS ARE FOR NODE CLUSTERS.
        IF(IFILL-1)53,54
   53 NT=NTOUR
      GO TO 55
   54 NT=ITT
   55 CONTINUE
C     READS IN DISTRIBUTION PARAMETERS
      IF(IDISTON)2140,2140,2149
 2149 MATOUT=MATIN=1
      CALL INPUT(MATOUT,MATIN)
 2140 DO 2104 I=1,MAXLEN
 2104 INF(I)=0
        DO 2105 I=1,ITT
 2105 CALL RANWRITE(11,INE,MAXLEN,I)
        MONTH= FIRST
        JK=MAXLEN*MAXLEN
        DO 3210 I=1,ITT
C
C-----MATRIX DATA READ IN.
        READ 4, TYPE(I),SUB(I),IQUOTE  ,ACTUAL(I),NUM(I),LENGTH(I),OUT(I)
```

```
      IJK= TYPE(I)
      NSIT(IJK)=NUM(I)
C-----IF ON (EQUAL TO 1) RUN WILL RESTART--USE COMPUTER PUNCHED MATRIX DATA.
      IF(SSWTCHF(4).EQ.1) GO TO 3210
      GO TO (800,801),SSWTCHF(3)
  800 DO 2 K=1,MAXLEN
    2 READ 3, (SYST(J,K),J=1,MAXLEN)
      GO TO 802
  801 CALL RANREAD(12,SYST,JK,I)
  802 CALL RANWRITE(10,SYST,JK,I)
 3210 CONTINUE
C
C-----TRANSFER PRIORITIES READ IN.
      READ 3,NOTT,NOFIRST,NOLAST
      PRINT 70,NOTT,NOFIRST,NOLAST
C      NOTT=NUMBER OF POSSIBLE TRANSFERS BETWEEN MOS
C      NOFIRST=NUMBER OF TRANSFERS PRIOR TO OTHER ASSIGNMENTS.
C      NOLAST=NUMBER OF TRANSFERS FOLLOWING OTHER ASSIGNMENTS.
C      IF OUTTT OR TOUR TO WHICH PERSONNEL ARE SENT=0. REMOVALS=LOSSES TO SYSTEM
       DO 2113 I=1,NOTT
      READ 4, OUTTT(I),OUTST(I),OUTTO(I),OUTSO(I),PERD(I),PERDTO(I),PCT
     1(I)
 2113 PRINT 71, PCT(I),OUTTO(I),OUTSO(I),PERD(I),OUTTT(I),OUTST(I),
     1PERDTO(I)
       PRINT 7, ID
C
C-----PRIORITIES READ IN.
       DO 5 I=1,NP
       READ 6, INTOUR(I),INSUB(I),OUTTOUR(I),OUTSUB(I),AFTER(I),PER(I),
     1 PERPOUT(I), REP(I), TOD(I), IGRADE(I), ITYPE(I)
      NPER=PER(I)*100
      PRINT 8, NPER ,OUTTOUR(I),OUTSUB(I),AFTER(I),INTOUR(I),INSUB(I),
     1ITYPE(I)
    5 CONTINUE
C
C-----LAST RESORT TOURS READ IN.
       DO 500 I=1,LRT
      READ 515, (LSTRSTO(I),LSTRSTT(I),LSTRSTS(I))
  500 PRINT 20,(LSTRSTO(I),LSTRSTT(I),LSTRSTS(I))
C          PERCENT= VECTOR OF PERCENTAGES TO BE APPLIED TO CALCULATED INPUT
C
C-----PRINT SYSTEM AS IS.
      PRINT 1006
      DO 807 I=1,ITT
      CALL RANREAD(10 ,SYST,JK,I)
      CALL FLIP
      PRINT 1355, I
      IAT=0
      JTT=0
      DO 808 JI=1,JK,MAXLEN
      JTT=JTT+1
      IT=JI+MAXLEN-1
       DO 808 J=JI,IT
      IF(SYST(J)) 809,808,809
  809 IAT=IAT+3
      IBT(IAT-2)= J-JI+1
      IBT(IAT)=SYST(J)
      IBT (IAT-1)= JTT
  808 CONTINUE
      IF(IAT) 810,810,811
  811 PRINT 604, (IBT(J),J=1,IAT)
  810 CONTINUE
```

```
  807 CONTINUE
C
C-----BEGIN MONTH BY MONTH SIMULATION.
       DO 401  INTEG=MONTH,LAST
       IP=0
       DO 372  I=1,NTOUR
       ACT(I)=0
      NEEDS(I)=0
  372 CONTINUE
C
C      UPDATE ENTIRE SYSTEM
      PRINT 19,INTEG
      DO 10 I=1,ITT
      NE(I)=0
      CALL RANREAD(10,SYST,JK,I)
      O=LENGTH(I)
      PP=OUT(I)*O/12.0
      LA=LENGTH(I)+1
      IF(LA.GT.MAXLEN) LA=MAXLEN
      LENNODE=LENGTH(I)
      IF(LENNODE.GT.MAXLEN) LENNODE=MAXLEN
C
C      UPDATES PERSONNEL MATRICES BY COLLECTING ALL WHO HAVE COMPLETED THE NODE
C        IN THE LA NODE TIME PERIOD AND THE IL+1 SYSTEM TIME PERIOD.
C       REMOVE COMPLETED NODE PERSONNEL TO SYST(LA,IL+1)
      DO 14 IL=1,MAXLEN
      ILSUM=0
      DO 58 K=LENNODE,MAXLEN
      IF(SYST(K,IL))56,58,57
   56 SYST(K,IL)=0
      GO TO 58
   57 X1=SYST(K,IL)*PP
      Y=SYST(K,IL)-X1
      IJ=X1
      IF(X1-IJ)11,11,12
   12 IK=Y
      Y=IK+1
      X1=IJ
   11 IF(I.EQ.5) PRINT 425,I,K,IL,SYST(K,IL)
      SYST(K,IL)=0
      ILSUM=ILSUM+Y
   58  CONTINUE
      IF(IL.GT.1) SYST(LA,IL)=IRETAIN
      IRETAIN=ILSUM
C
C       ALL OTHERS ARE MOVED UP ONE MONTH IN THE NODE AND ONE MONTH IN THE SYSTEM
C       KOLD AND ILOLD=ROW AND COLUMN PRIOR TO UPDATING SYSTEM.
C       KNEW AND ILNEW=ROW AND COLUMN AFTER UPDATING SYSTEM.
      ILOLD=MAXLEN-IL
      ILNEW=MAXLEN-IL+1
      DO 14 K=1,LENNODE
      KNEW=LENNODE-K+1
      KOLD=LENNODE-K
      IF(KNEW.LE.1)59,60
   59 SYST(KNEW,ILNEW)=0
      GO TO 14
   60 IF(ILNEW.LE.1)61,62
   61 SYST(KNEW,ILNEW)=0
      GO TO 14
   62 SYST(KNEW,ILNEW)=SYST(KOLD,ILOLD)+SYST(KNEW,ILNEW)
      SYST(KOLD,ILOLD)=0
   14  CONTINUE
```

```
      SYST(1,1)=0
      CALL RANWRITE(10,SYST,JK,I)
   10 CONTINUE
C
C       TIMEUP ACCOMPLISHES TRANSFERS.  THESE TRANSFERS ARE EFFECTIVE PRIOR
C         TO APPLICATION OF FILL RULES
C       ITRTIME=TRANSFER TIME.  IF=1, ALL NOFIRST TRANSFERS ARE MADE.
C         IF=2, NOFIRST+1 TO NOTT TRANSFERS ARE MADE.
      ITRTIME=1
       CALL TIMEUP
C
C       SUBROUTINE SUMMARY CALCULATES ACTUAL FOR EACH MATRIX
      DO 64 I=1,ITT
      IACTUAL=ACTUAL(I)=0
      CALL SUMMARY(2,I,IACTUAL)
      ACTUAL(I)=IACTUAL
      KKK=TYPE(I)
      ACT(KKK)=ACT(KKK)+ACTUAL(I)
C
C       UPDATES INE VECTOR
       CALL RANREAD(11,INE,MAXLEN,I)
      DO 63 K=1,MAXLEN
      INE(K)=0
      DO 63 J=1,MAXLEN
      IF(SYST(K,J).GT.0)INE(K)=J
   63 CONTINUE
      CALL RANWRITE(11,INE,MAXLEN,I)
   64 CONTINUE
C
      PRINT 24
      MAXNODE=MINNODE=0
      DO 27 I=1,NTOUR
      DO 26 II=1,MAXSUB
   26 MAXNODE=MAXNODE+1
      MINNODE=MAXSUB*(I-1)+1
   27 PRINT 29,I,ACT(I),(II,ACTUAL(II),II=MINNODE,MAXNODE)
C
C-----CALCULATE MINIMUM PERSONNEL WITHIN TOUR AREA.
      IF(IDISTON)2142,2142,2143
 2143 MATOUT=MATIN=1
      CALL ISUMAR(MATOUT,MATIN)
C
C-----INPUT PERSONNEL REQUIREMENTS.
 2142 IF(IFILL-1) 31,32,32
   31 READ 33, (NEEDS(L),L=1,NTOUR)
      PRINT 48,(NEEDS(L),L=1,NT)
       GO TO 34
   32 READ 33, (NE(L),L=1,ITT)
       IS=0
       MO=1
       DO 35 L=1,NTOUR
       NX=NSIT(L)
       IS=IS+NX
       DO 36 K=MO,IS
   36 NEEDS(L)=NEEDS(L)+NE(K)
       MO=IS+1
   35 CONTINUE
   34 CONTINUE
C
C-----INPUT OUTSIDE DATA
       IF(CIOS-1) 610,611,611
  611 READ 33, (IOS(I),I=1,CIOS)
```

```
  610 CONTINUE
       DO 165 I=1,NTOUR
       NED(I)=NEEDS(I)
       NEEDS(I)=NEEDS(I)-ACT(I)
       IF(NEEDS(I)) 376,165,165
  376 NEEDS(I)=0
  165 CONTINUE
      PRINT 48,(NED(I),I=1,NTOUR)
      PRINT 30,(NEEDS(I),I=1,NTOUR)
       DO 166 I=1,ITT
       NEE(I)=NE(I)
      NE(I)=NE(I)-ACTUAL(I)
       IF(NE(I)) 377,166,166
  377 NE(I)=0
  166 CONTINUE
      PRINT 40,(NEE(I),I=1,ITT)
      PRINT 41,(NE(I),I=1,ITT)
C
C     CONTROLS NEEDS TO BE FILLED ACCORDING TO PRIORITY OF FILL LEVELS
      PRINT 52,IDISTON
      IF(IDISTON)2144,2144,2145
 2145 DO 2144 LEVEL=1,NPRLEV
      MATOUT=MATIN=1
      CALL MODIFY(MATOUT,MATIN)
 2144 CALL FILUP
C
C      TRANSFER ALL UNASSIGNED PERSONNEL
      ITRTIME=2
      CALL TIMEUP
C
       IF(CIOS) 612,612,613
  613 DO 510 I=1,LRT
       J=LSTRSTO(I)
      K=LSTRSTT(I)
       LL=LSTRSTS(I)
       IF(IOS(J)) 510,510,520
  520 IC=(K-1)*MAXSUB+LL
       CALL RANREAD (10,SYST,JK,IC)
       SYST(1,1)=SYST(1,1)+IOS(J)
      IP=IP+1
      PRINT 1005, K,LL,J,IOS(J),IP
       CALL PACK(K,LL,0,J,IOS(J),0,1,1,N1,N2,N3)
       CALL RANWRITE(13,N1,3,IP)
       CALL RANWRITE(10,SYST,JK,IC)
       ACTUAL(IC)=ACTUAL(IC)+IOS(J)
       ACT(K)=ACT(K)+IOS(J)
       IOS(J)=0
  510 CONTINUE
  612 CONTINUE
C
C-----CALCULATE DEPLOYABLE PERSONNEL FOR EACH TOUR AREA.
      IF(IDISTON.LE.0) GO TO 17
      MATOUT=MATIN=1
      CALL ISUMAR(MATOUT,MATIN)
   17 CONTINUE
C
      PRINT 356
      PRINT 1006
      DO 350 I=1,ITT
       PRINT 1355, I
       CALL RANREAD(10,SYST,JK,I)
      CALL FLIP
```

```
      IAI=0
      JII=0
       DO 600  JI=1,JK,MAXLEN
       JII=JII+1
      II=JI+MAXLEN-1
      DO 600  J=JI,II
      IF(SYST(J)) 601,600,601
  601 IAI=IAI+3
      IBI(IAI-2)= J-JI+1
      IBI(IAI)=SYST(J)
      IBI (IAI-1)= JII
  600 CONTINUE
      IF(IAI) 602,602,603
  603 PRINT 604, (IBI(J),J=1,IAI)
  602 CONTINUE
  350 CONTINUE
      DO 806  I=1,III
       CALL RANREAD(10,SYST,JK,I)
       DO 804 K=1,MAXLEN
       INF(K)=0
       DO 804 J=1,MAXLEN
       IF(SYST(K,J)) 804,804,805
  805 INF(K)=1
  804 CONTINUE
  806 CALL  RANWRITE(11,INF,MAXLEN,I)
       PRINT 902
       DO 900 J=1,IP
       CALL RANREAD(13,N1,3,J)
       CALL UNPACK(M1,M2,M3,M4,M5,M6,M7,M8,N1,N2,N3)
  900 PRINT 901, (M1,M2,M3,M4,M5,M6,M8,M7)
      PRINT 356
      GO TO (3211,401) SSWITCHF(5)
 3211 WRITE (58,3212)
      PAUSE
      IF (SSWITCHF(5) .EQ.1) GO TO 402
  401 CONTINUE
  402 PRINT  404,INIEG
      IF(IPUNCH.LE.0) GO TO 65
      DO 51 I=1,III
   51 PUNCH 50, TYPE(I), SUB(I), IQUOTE  , ACTUAL(I), NUM(I), LENGTH(I),
     1 OUT(I)
C
C     CALLS SUBROUTINE ALLOCATE WHICH ACTUALLY DISTRIBUTES PERSONNEL WITHI
C        TOUR AREAS TO COMMAND ELEMENTS
   65 IF(IDISTON)2146,2146,2147
 2147 MATOUT=MATIN=1
      CALL  TADD(MATOUT,MATIN)
C
C     X69A FORMATS
    1 FORMAT(A3,12I4,F6.4,2I1)
    3 FORMAT(20I4)
    4 FORMAT(6I10,F6.3)
    6 FORMAT   (5I8, 3F6.3,3I1)
    7 FORMAT(32H1FILL PRIORITIES FOR SIMULATION,A4//)
    8 FORMAT(3X,I3,1X,7HPERCENT,5X,6H FROM ,I2,1H.,I2,5X,7H AFTER I3,7HP
     1ERIODS,5X,6H INTO ,I2,1H.,I2,5X,15H MOVEMENT TYPE I2)
   18 FORMAT(//1X47H NUMBER AVAILABLE FOR ANOTHER TOUR AFTER MONTH, I4)
   20 FORMAT(1X,48HREMAINDER OF SYSTEM INPUT FROM OUTSIDE CATEGORY I3,5H
     1 INTO,I2,1H.,I2)
   23 FORMAT(//28H0PERSONNEL INDICATOR VECTORS /5X,34HVALUES = MONTH IN
     1SYSTEM OR COLUMN/5X,32HPOSITIONS = MONTH IN TOUR OR ROW)
   24 FORMAT(//29H0NODE CLUSTER AND NODE TOTALS)
```

```
   29 FORMAT(/5X,13HNODE CLUSTER ,I2,3H = ,I6/(13X,5HNODE ,I2,3H = ,I6))
   30 FORMAT(//25H0NODE CLUSTER SHORTAGES =,10I10)
   33 FORMAT(10I8)
   40 FORMAT(//14H0NODE QUOTAS =,3X,(10I10))
   41 FORMAT(//17H0NODE SHORTAGES =,(10I10))
   48 FORMAT(//22H0NODE CLUSTER QUOTAS =,3X,10I10)
   50 FORMAT  (6I10, F6.3)
   52 FORMAT(11H0IDISTON = ,I3)
   70 FORMAT(19H0TOTAL TRANSFERS = ,I3,2X,21H INITIAL TRANSFERS = ,I3,2X
     1,18HFINAL TRANSFERS = ,I3)
   71 FORMAT(1X,F6.3,6H FROM I1,1H,,I1,2X,6HAFTER ,I3,14H TIME PERIODS ,
     13HTO ,I1,1H,,I1,2X,6HAFTER I1,13H TIME PERIODS)
  356 FORMAT(///)
  404 FORMAT  (23H LAST PERIOD SIMULATED=   I4)
  425 FORMAT(8H0MATRIX ,I2,3X,4HROW ,I2,3X,7HCOLUMN ,I2,3X,2H =,I4)
  515 FORMAT(3I5)
  604 FORMAT(7(4X,2I4,I7))
  901 FORMAT(8I10)
  902 FORMAT(//4X6HTOURIN,5X5HSUBIN,3X7HTOUROUT,4X6HSUBOUT,2X8HN MEN IN,
     14X6HN LOST,1X9HM IN SYST,1X16H M IN TOUR (ROW)    )
 1005 FORMAT  (12H MAIN          2I8,8X,4I8)
 1006 FORMAT(//1X,15HPERSONNEL NODES/H022H     SYS TOUR(COL,ROW))
 1355 FORMAT(1X,I4)
 3212 FORMAT  (103H FOR ANOTHER PASS--TURN OFF SWITCH 5 AND PRESS MI /
     1 MI. TO TERMINATE PRESS MI / MI.                            )
 2146 STOP
      END
```

FORTRAN DIAGNOSTIC RESULTS FOR MAINGMM

NULL STATEMENT NUMBERS
23

```
      SUBROUTINE LOCK (I,J,MON,ILOSS,IGAIN,IOUT,IFF,MAX,ICC,IE,K1,K3   )
C-----ALL COMMON STATEMENTS IN LOCK ARE NOT IDENTICAL WITH OTHER SUBROUTINES
      COMMON        NOFIRST    ,NOLAST    ,NOTT       ,ITRTIME   ,INTEG
      COMMON        IP         ,INE(48)   ,IEND       ,OUTTT(200),
      COMMON        OUTST(200),OUTTO(200)             ,OUTSO(200)
      COMMON        PCT(200)   ,PERD(200),JK          ,SYST(48,48)
      COMMON        PA(100)    ,NAV(100)  ,LENGTH(100)            ,OUT(100)
      COMMON        INTOUR(220)           ,INSUB(220)             ,OUTTOUR(220)
      COMMON        IOD(220)   ,OUTSUB(220)           ,AFTER(220)
      COMMON        PER(220)   ,RATE(100)             ,NEEDS(10),NE(10,10)
      COMMON        IOS(10)    ,NED(10)   ,NEE(100)   ,IFILL     ,MAXLEN
      COMMON        NTOUR      ,NP        ,CTOS       ,PERDTO(200)
      COMMON        PDW        ,LSTRSTO(10)           ,LSTRSTT(10)
      COMMON        LSTRSTS(10)           ,IZT        ,MAXSUR    ,ACT(10)
      COMMON        NSTT(10)   ,INOS(10,10)           ,IGRADE(220)
      COMMON         REP(220)  ,ITYPE(220)            ,PEROUT(220)
      COMMON        IDISTON    ,ISUM      ,GRPSUM(100)            ,PRIO(100)
      COMMON        BEGROW(100)           ,ENDROW(100)            ,BEGCOL(100)
      COMMON        ENDCOL(100)           ,MATSUM(100)            ,MATGRPS(100)
      COMMON        TYPE(100),SUB(100)    ,NUM(100)   ,ACTUAL(100)
      COMMON        NPRLEV     ,NT        ,IHOLD      ,LEN       ,LEVEL     ,M
      COMMON        NCRNODE(100)          ,MIN(100)   ,GRPINPR(100)
      COMMON        MAXDEPL(100)
      INTEGER       TYPE       ,SUB       ,ACTUAL     ,ENDCOL
      INTEGER       CTOS       ,SYST      ,OUTSUB     ,OUTTOUR   ,AFTER
      INTEGER       ACT        ,OUTTO     ,OUTSO      ,OUTTT     ,OUTST
      INTEGER       PERD       ,GRPSUM    ,BEGROW     ,ENDROW    ,BEGCOL
      INTEGER       PERDTO
      INTEGER       OKK
      IF (IGRADE(IEND) .LE.0 ) GO TO 14
      PRINT 1005, IGRADE(IEND), IEND
      CALL TRNSPOSE
   14 CONTINUE
       N1=N2=N3=0
C-----MON = TIME PERIOD AFTER WHICH REMOVAL TAKES PLACE (MAXLEN TO MON+1)
       MONTH=MON+1
       ITT=0
          L=(I-1)*MAX+1
C
C-----IOD= IF ON, REMOVAL IS FROM PERIOD FOLLOWING MON  ONLY
      IF(IOD(IE) -1) 40,41,41
C-----M= ROW OF REMOVAL FROM THE MATRIX
   41 M=MONTH
       GO TO 42
   40 M=LENGTH(I)
   42 CONTINUE
       LEN=MAXLEN
       IF(M-MONTH) 1,152,152
  152 ISU=0
       CALL RANREAD([illegible],INE,MAXLEN,L)
C
C2- [illegible] OF COL S WITH + VALUE IN ALL ROWS GREATER THAN OR EQUAL
C2- [illegible] FF.
[illegible] 1,154,154
[illegible]
C- [illegible] OF MATRIX (TOUR).. [illegible]ALUE OF INE(X)= HIGHEST COL OF ROW
C------ WITH A + VALUE.
C-----ISU= FLAG OF ENTITY IN OR BEYOND REMOVAL ROW.
      IF (INE(LI)) 158,150
  158 IF ((LI)-MONTH) 150,151,151
```

```
  151 ISU=ISU+1
  150 CONTINUE
C1----FUNCTION MX DETERMINES HIGHEST COL IN MATRIX WHICH HAS A POSITIVE VALUE
C1----IF IOD IS OFF.
       NMN=MX(INE,MAXLEN)
      GO TO 155
C3----CHECK THE NUMBER OF COL S WITH + VALUE IN ALL ROWS EQUAL TO MONTH IF IODON
  154 DO 156 LI=1,MAXLEN
      IF (INE(LI)) 159,156
  159 IF ((LI)-MONTH) 156,157,156
  157 ISU=ISU+1
  156 CONTINUE
C-----NMN= STARTING ROW OF REMOVAL FROM THE MATRIX (NMN TO COL1 OF ROW M-REMVLPT
      NMN = INE(M)
  155 IF(ISU-1) 1,102,102
C
  102 LEN=NMN
       NMN=0
       NMLEN=0
      IPR=1
C4----SEARCH FOR HIGHEST POSITION OF A POSITIVE INE VALUE
C4----NMLEN IS THE HIGHEST POSITION
C
      DO 301 ILK=MONTH,MAXLEN
      IF(INE(ILK) .LE. 0) GO TO 301
      IF(LEN-INE(ILK))301,302,304
  302 NMN=ILK
      GO TO 301
  304 NMLEN=ILK
  301 CONTINUE
      IPR=IPR+1
      IF(NMN.EQ.0 .AND. NMLEN.EQ.0) GO TO 1
       IF(NMN) 303,303,306
  303 IF(IOD-1)306,1,1
  306 IF(NMN-NMLEN)307,305,305
  307 NMN=NMLEN
  305 M=NMN
       GO TO 322
C
  321 CALL RANREAD(10,SYST,JK,L)
       CALL RANREAD(11,INE,MAXLEN,L)
      IF(INE(M)-1)9,324,324
  324  LEN=INE(M)
       ISU=LEN
       GO TO 323
C
  322 CONTINUE
    2    CALL RANREAD(10,SYST,JK,L)
        CALL RANREAD(11,INE,MAXLEN,L)
  323 CONTINUE
      IPR=IPR+1
       IF(ACTUAL(L)) 1,1,34
   34 IF (INE(M)- LEN)   32,51,51
   51 CONTINUE
      IF(SYST(M,LEN)) 32,32,33
   33 IHOLD=0
      IF(IDISTON)61,61,60
C     DISTRO MAINTAINS MINIMUM LEVEL IN TOUR AREAS.
   60 IF(NCRNODE(K1).EQ.1 .AND. LEVEL.EQ.1) GO TO 61
      IF(MIN(I) .EQ.1) GO TO 62
      CALL MINIMUM(I,K1)
   61 IF(SYST(M,LEN))62,62,63
```

```
   62 SYST(M,LFN)=SYST(M,LFN)+IHOLD
      GO TO 32
   63 SOO=M/12.0*OUT(L)
      IF(SOO-1) 7,7,8
    8 SOO=1.0
    7 IFF=SYST(M,LFN)*SOO
      SYST(M,LFN)=SYST(M,LFN)-IFF
      ITT=IFF+ITT
C-----WHEN PERCENTAGE WAS APPLIED TO 1 ENTITY--THE ENTITY WAS LOST TO SYST
      KOM=SYST(M,LFN)
      IF (SYST(M,LFN) .EQ.1) GO TO 26
      IF(PEROUT(IE).GT.0) GO TO 12
      PEROUT(IE)=1.00
   12 SYST(M,LFN) = SYST(M,LFN)*PEROUT(IE)
C       RETAIN PERSONNEL NEEDED FOR MINIMUM FILL
   26 KOM=KOM-SYST(M,LFN)+IHOLD
      IF(SYST(M,LEN)-IOUT)3,4,4
    3 IGAIN=IGAIN+SYST(M,LEN)
      IOUT=IOUT-SYST(M,LEN)
      IS=SYST(M,LEN)
C     IF KOM.GT. 0,PERSONNEL REMAIN IN SYST(M,LFN) AND INE(LEN) REMAINS THE SAME
      IF(KOM.GT. 0) GO TO 64
      INE(LEN)=INE(LEN)-1
   64 CALL RANWRITE(11,INE,MAXLEN,L)
      SYST(M,LFN)=0
      SYST(M,LFN)=SYST(M,LEN)+KOM
      CALL RANWRITE(10,SYST,JK,L)
      OKK=M+1
      CALL RANREAD  (10,SYST,JK,ICC)
      CALL RANREAD(11,INE,MAXLEN,ICC)
C
C-----TYPE 3  INTO (I,1)
      LEN2=1
      GO TO  (1100,1100,1300,1400) ITYPE (IEND)
 1300 SYST(LEN,1) = SYST(LEN,1) + IS
      IJJ= LEN
      LEN2=1
      IF (INE(LEN)-LEN2) 1354,1355,1355
 1354 INE(LEN)=1
      CALL RANWRITE(11,INE,MAXLEN,ICC)
 1355 GO TO 25
C
C-----TYPE 4  INTO (1,1)
 1400 SYST(1,1) = SYST(1,1) + IS
      LEN2=IJJ=1
      IF (INE(1)-LEN2) 1454,1455,1455
 1454 INE(1)= 1
      CALL RANWRITE  (11,INE,MAXLEN, ICC)
 1455 GO TO 25
C
C-----TYPE 1 AND 2 (1,I)
 1100 IF (M-MAXLEN) 24,24,23
   23 SYST(1,M  )=SYST(1,M  )+IS
      IJJ=1
      LEN2=M
      IF(INE(1)-M) 52,52,53
   52 INE(1)=M
      CALL RANWRITE(11,INE,MAXLEN,ICC)
   53 CONTINUE
      GO TO 25
   24 SYST(1,LEN)= SYST(1,LEN)+IS
      IJJ=1
```

```
      LEN2=LEN
      IF(INE(1)-LEN2) 54,55,55
   54 INE(1)=LEN
      CALL RANWRITE(11,INE,MAXLEN,ICC)
   55 CONTINUE
   25 CONTINUE
       IP=IP+1
      M PRINT = M-1                                                     MAJ ONLY
      ITEM=5
      PRINT 1001, K1,K3,I,J,IS,IFF,IJJ,LEN2,IP,ITEM
      CALL PACK (K1,K3,I,J,IS,IFF, IJJ,LEN2,N1,N2,N3    )
       CALL RANWRITE(13,N1,3,IP)
       CALL RANWRITE  (10,SYST,JK,ICC)
C
   32 IF(LEN-1) 9,9,6
    9 IF(M-MONTH) 1,1,10
   10 M=M-1
       GO TO 321
    6 LEN=LEN-1
       GO TO 2
C
    4 IGAIN=IGAIN+IOUT
       CALL RANWRITE( 10,SYST,JK,L)
       CALL RANREAD  (10,SYST,JK,ICC)
      CALL RANREAD(11,INE,MAXLEN,ICC)
C
      GO TO (1110,1110,1310,1410) ITYPE(IEND)
C----- TYPE 3 INTO (I,1)
 1310 SYST(LEN,1) = SYST(LEN,1) + IOUT
      IJJ= LEN
      LEN2=1
      IF (INE(LEN)-LEN2) 1357,22,22
 1357 INE(LEN)=1
      GO TO 22
C
C-----TYPE 4  INTO (1,1)
 1410 SYST(1,1) = SYST(1,1)+ IOUT
      LEN2=IJJ=1
      IF (INE(1)-LEN2) 1457,22,22
 1457 INE(1)= 1
      GO TO 22
C
C-----TYPE 1 AND 2   (1,I)
 1110 IF (M-MAXLEN) 21,21,20
   21 SYST(1,LEN)=SYST(1,LEN)+IOUT
      IJJ=1
      LEN2=LEN
      IF(INE(1)-LEN) 57,22,22
   57 INE(1)=LEN
      GO TO 22
   20 SYST(1,M   )=SYST(1,M   )+IOUT
      IJJ=1
      LEN2=M
      IF(INE(1)-M) 56,56,22
   56 INE(1)=M+1
   22 CONTINUE
       IP=IP+1
      MPRINT=M-1
      ITEM=6
      PRINT 1001, K1,K3,I,J,IS,IFF,IJJ,LEN2,IP,ITEM
      CALL PACK (K1,K3,I,J, IOUT,IFF,IJJ,LEN2,N1,N2,N3  )
       CALL RANWRITE(13,N1,3,IP)
```

```
      CALL RANWRITE(11,INE,MAXLEN,ICC)
       CALL RANWRITE (10,SYST,JK,ICC)
       CALL RANREAD  (10,SYST,JK,L)
       SYST(M,LEN)=KOM-IOUT+SYST(M,LEN)
      CALL RANREAD(11,INE,MAXLEN,L)
      IF(INE(LEN)-M) 58,59,59
   58 INE(LEN)=M
      CALL RANWRITE(11,INE,MAXLEN,ICC)
   59  CALL RANWRITE (10,SYST,JK,L)
       OKK=M+1
       IOUT=0
    1 IFF=ITT
      IF (IGRADE(IEND) .LE.0 ) GO TO 15
      PRINT 1007, IGRADE(IEND), IEND
      CALL TRNSPOSE
   15 CONTINUE
C
C     LOCK FORMATS
 1001 FORMAT  (6H LOCK1  10I10)
 1002 FORMAT  (6H LOCK2 10I10)
 1003 FORMAT(11H0IDISTON = ,I2)
 1005 FORMAT  (8H IGRADE=  I4, 6H IEND= I4)
 1007 FORMAT  (8H IGRADE=  I4, 6H IEND= I4)
       RETURN
      END

                   FORTRAN DIAGNOSTIC RESULTS  FOR    LOCK


NULL STATEMENT NUMBERS
     1003          1002
```

```
      SUBROUTINE FILUP
      COMMON       NOFIRST    ,NOLAST    ,NOTT      ,ITRTIME    ,INTEG
      COMMON       IP         ,INE(48)   ,IEND      ,OUTTT(200)
      COMMON       OUTST(200),OUTTO(200)           ,OUTSO(200)
      COMMON       PCT(200)   ,PERD(200),JK         ,SYST(48,48)
      COMMON       PA(100)    ,NAV(100)  ,LENGTH(100)           ,OUT(100)
      COMMON       INTOUR(220)           ,INSUB(220)            ,OUTTOUR(220)
      COMMON       IOD(220)   ,OUTSUB(220)          ,AFTER(220)
      COMMON       PER(220)   ,RATE(100)            ,NEEDS(10),NE(10,10)
      COMMON       IOS(10)    ,NED(10)   ,NEE(100)  ,IFILL      ,MAXLEN
      COMMON       NTOUR      ,NP        ,CIOS      ,PERDTO(200)
      COMMON       POW        ,LSTRSTO(10)          ,LSTRSTT(10)
      COMMON       LSTRSTS(10)           ,ITT       ,MAXSUB     ,ACT(10)
      COMMON       NSIT(10)   ,INOS(10,10)          ,IGRADE(220)
      COMMON        REP(220)  ,ITYPE(220)           ,PEROUT(220)
      COMMON       IDISTON    ,ISUM      ,GRPSUM(100)           ,PRIO(100)
      COMMON       BEGROW(100)           ,ENDROW(100)           ,BEGCOL(100)
      COMMON       ENDCOL(100)           ,MATSUM(100)           ,MATGRPS(100)
      COMMON       TYPE(100),SUB(100)    ,NUM(100)  ,ACTUAL(100)
      COMMON       NPRLEV     ,NT        ,IHOLD     ,LEN        ,LEVEL     ,M
      COMMON       NCRNODE(100)          ,MIN(100)  ,GRPINPR(100)
      COMMON       MAXDEPL(100)
      INTEGER      TYPE       ,SUB       ,ACTUAL    ,ENDCOL
      INTEGER      CIOS       ,SYST      ,OUTSUB    ,OUTTOUR    ,AFTER
      INTEGER      ACT        ,OUTTO     ,OUTSO     ,OUTTT      ,OUTST
      INTEGER      PERD       ,GRPSUM    ,BEGROW    ,ENDROW     ,BEGCOL
      INTEGER      PERDTO
      INTEGER OK,OKK
       N1=N2=N3=0
       IF(IFILL-1) 5,10,10
    5 DO 15 I=1,ITT
   15 NE(I)=0
   10 ICOUNT=0
C
C-----CHECK EACH TOUR TO FILL REQUIREMENTS ACCORDING TO FILL RULES.
       DO 20 IEND=1,NP
        K1=INTOUR(IEND)
       K2=OUTTOUR(IEND)
       K3=INSUB(IEND)
        K4=OUTSUB(IEND)
       OK=0
      IF(IFILL-1) 35,40,40
   40 IFIND=NE(K1,K3)
      GO TO 45
   35 IFIND=NEEDS(K1)
   45 ICC=(K1-1)*MAXSUB+K3
      IF(PER(IEND)) 26,26,30
   30 IFIND=PER(IEND) * IFIND
   26 CONTINUE
       IF(IFIND) 140,141,142
  140 IFIND=0
  141 GO TO 20
  142 CONTINUE
      IF(OUTTOUR(IEND)) 80,80,75
   75 IC=(K2-1) * MAXSUB+K4
C      INITIALLY, SYSTEM INPUTS THOSE WHO HAVE COMPLETED THE TOURS.
      LA=LENGTH(IC)+1
        CALL RANREAD(10,SYST,JK,IC)
      DO 255 IM=1,MAXLEN
      IF(SYST(LA,IM).LE.0) GO TO 255
```

```
      IF(PEROUT(IEND).GT.0) GO TO 55
      PEROUT(IEND)=1.00
   55 KOM=SYST(LA,IM)*PEROUT(IEND)
      IF(SYST(LA,IM).EQ.1) KOM=1
       IF(KOM) 262,262,256
  256 IF(IFIND-KOM) 60,60,65
   60  CONTINUE
C
C-----NUMBER AVAILABLE IS NOT SUFFICIENT TO FILL REQUIREMENTS.
      SYST(LA,IM)=SYST(LA,IM)-IFIND
       CALL RANREAD(10,SYST,JK,ICC)
      PRINT 143,SYST(LA,IM),IM,LENGTH(IC),KOM
      SYST(1,IM   )=SYST(1,IM   )+IFIND
       IP=IP+1
      PRINT 1005,K1,K3,K2,K4,IFIND,IM,IP
      CALL PACK(K1,K3,K2,K4,IFIND,0,IM,1,N1,N2,N3)
       CALL RANWRITE( 13,N1,3,IP)
        CALL RANWRITE(10,SYST,JK,ICC)
      OK=IFIND
       IFIND=0
      ACT(K1)=ACT(K1)+OK
       ACTUAL(ICC)=ACTUAL(ICC)+OK
       GO TO 70
   65 CONTINUE
C
C-----NUMBER AVAILABLE EXCEEDS REQUIREMENTS.
       IFIND=IFIND-KOM
       OK=KOM
      SYST(LA,IM)=SYST(LA,IM)-KOM
        CALL RANREAD(10,SYST,JK,ICC)
      SYST(1,IM   )=SYST(1,IM   )+OK
       IP=IP+1
      PRINT 1006,K1,K3,K2,K4,OK,IM,IP
      CALL PACK(K1,K3,K2,K4,OK,0,IM,1,N1,N2,N3)
       CALL RANWRITE(13,N1,3,IP)
       CALL RANWRITE(10,SYST,JK,ICC)
      ACT(K1)=ACT(K1)+OK
       ACTUAL(ICC)=ACTUAL(ICC)+OK
       OKK=M+1
        GO TO 258
  255 CONTINUE
  258 CONTINUE
       IF(  IFIND) 260,70,262
  260 IFIND=0
       GO TO 70
C      THERE SHOULD NOT BE ANYONE BEYOND THE TOUR LIMITS OR LENGTH(IC).
  262 IF(AFTER(IEND)-LENGTH(IC))50,50,70
   50 IOUT=IFIND
       ILOSS=IGAIN=IFF=0
        MMMM=MAXSUB
      MY=MAXSUB
      IF=IEND
C     SEARCH FOR PERSONNEL IF THEY ARE AVAILABLE ABOVE MINIMUM LEVELS.  FOR
C        CRITICAL TOUR LEVELS IGNORE MINIMUM LEVELS.
      IF(IDISTON.EQ.0) GO TO 25
      IF(MIN(K2).EQ.0) GO TO 25
      IF(LEVEL .EQ.1 .AND. NCRNODE(K1) .EQ.1) GO TO 25
      GO TO 20
   25 CALL LOCK(K2,K4,AFTER(IEND),ILOSS,IGAIN,IOUT,IFF,MY,ICC,IE,K1,K3)
      PRINT 1002, K2,K4,AFTER(IEND),ILOSS,IGAIN,IOUT,IFF,MY,ICC,IE,K1,K3
       MAXSUB=MMMM
       ACT(K2)=ACT(K2)-IGAIN-IFF
```

```
          OK=IGAIN
          IFIND=IOUT
          ACTUAL(IC)=ACTUAL(IC)-IGAIN-IFF
          ACT(K1)=ACT(K1)+OK
          ACTUAL(ICC)=ACTUAL(ICC)+OK
          IF(IFILL-1) 150,151,151
  151 NE(K2,K4)=NE(K2,K4)+IGAIN+IFF
  150 NEEDS(K2)=NEEDS(K2)+IGAIN+IFF
   70  IF(IFILL-1) 85,90,90
   90 NE(K1,K3)=IFIND
   85 NEEDS(K1)=IFIND
          GO TO 120
C
C-----INPUT SOUGHT FROM OUTSIDE SYSTEM.
   80 CALL OUTSIDE(IEND)
          GO TO 120
  120 ICOUNT=ICOUNT+OK
   20 CONTINUE
C
C      FILUP FORMATS
  143 FORMAT(6H FILUP,4I10)
 1001 FORMAT  (12H BEFORE LOCK  12I8)
 1002 FORMAT  (12H AFTER LOCK  12I8)
 1005 FORMAT  (8H FILUP1  5I8,8X,2I8)
 1006 FORMAT  (8H FILUP2  5I8,8X,2I8)
C
          RETURN
        END

                    FORTRAN DIAGNOSTIC RESULTS  FOR    FILUP


NULL STATEMENT NUMBERS
      1001
```

```
      SUBROUTINE TIMEUP
      COMMON      NOFIRST    ,NOLAST    ,NOTT      ,ITRTIME   ,INTEG
      COMMON      IP         ,INE(48)   ,IEND      ,OUTTT(200)
      COMMON      OUTST(200),OUTTO(200)            ,OUTSO(200)
      COMMON      PCT(200)   ,PERD(200),JK         ,SYST(48,48)
      COMMON      PA(100)    ,NAV(100)  ,LENGTH(100)          ,OUT(100)
      COMMON      INTOUR(220)           ,INSUB(220)           ,OUTTOUR(220)
      COMMON      IOD(220)   ,OUTSUB(220)          ,AFTER(220)
      COMMON      PER(220)   ,RATE(100)            ,NEEDS(10),NE(10,10)
      COMMON      IOS(10)    ,NED(10)   ,NEE(100)  ,IFILL     ,MAXLEN
      COMMON      NTOUR      ,NP        ,CIOS      ,PERDTO(200)
      COMMON      PDW        ,LSTRSTO(10)          ,LSTRSTT(10)
      COMMON      LSTRSTS(10)           ,ITT       ,MAXSUB    ,ACT(10)
      COMMON      NSIT(10)   ,INOS(10,10)          ,IGRADE(220)
      COMMON       REP(220)  ,ITYPE(220)           ,PEROUT(220)
      COMMON      IDISTON    ,ISUM      ,GRPSUM(100)          ,PRIO(100)
      COMMON      BEGROW(100)           ,ENDROW(100)          ,BEGCOL(100)
      COMMON      ENDCOL(100)           ,MATSUM(100)          ,MATGRPS(100)
      COMMON      TYPE(100),SUB(100)    ,NUM(100)  ,ACTUAL(100)
      COMMON      NPRLEV     ,NT        ,IHOLD     ,LEN       ,LEVEL     ,M
      COMMON      NCRNODE(100)          ,MIN(100)  ,GRPINPR(100)
      COMMON      MAXDEPL(100)
      INTEGER     TYPE       ,SUB       ,ACTUAL    ,ENDCOL
      INTEGER     CIOS       ,SYST      ,OUTSUB    ,OUTTOUR   ,AFTER
      INTEGER     ACT        ,OUTTO     ,OUTSO     ,OUTTT     ,OUTST
      INTEGER     PERD       ,GRPSUM    ,BEGROW    ,ENDROW    ,BEGCOL
      INTEGER     PERDTO
       DIMENSION SYS(48,48)
      INTEGER SYS
      PRINT 13,ITRTIME
       JK=MAXLEN*MAXLEN
      IF(ITRTIME.EQ.2) GO TO 10
      ISTART=1
      ISTOP=NOFIRST
      GO TO 11
   10 ISTART=NOFIRST+1
      ISTOP=NOTT
   11 DO 1 I=ISTART,ISTOP
      N1=N2=N3=0
      K1=OUTTT(I)$K2=OUTTO(I)$K3=OUTST(I)$K4=OUTSO(I)
       IF(OUTTT(I)-1) 6,7,7
C
C-----TRANSFERS BEHAVE LIKE LOSSES TO SYSTEM.
    6  K=(OUTTO(I)-1) * MAXSUB+OUTSO(I)
      CALL RANREAD(10,SYST,JK,K)
       MSP=PERD(I)
       DO 8 L=1,MAXLEN
       IF(SYST(MSP,L)) 8,8,9
    9 NNN=SYST(MSP,L)*PCT(I)
       SYST(MSP,L)=SYST(MSP,L)-NNN
      PRINT 3, OUTTT(I),OUTST(I),OUTTO(I),OUTSO(I),PCT(I),PERD(I),
     1SYST(MSP,L)
       IP=IP+1
       CALL PACK(K1,K3,K2,K4,0,NNN,MSP,L ,N1,N2,N3)
       CALL RANWRITE(13,N1,3,IP)
    8 CONTINUE
       GO TO 1
C
C-----TRANSFERS ARE FROM ONE TOUR TO ANOTHER WITHIN THE SYSTEM.
    7  J=(OUTTT(I)-1)*MAXSUB+OUTST(I)
```

```
      K=(OUTTO(I)-1)*MAXSUB+OUTSO(I)
      CALL RANREAD(10,SYST,JK,K)
      CALL RANREAD(10,SYS,JK,J)
      MSP=PERD(I)
      DO 2 L=1,MAXLEN
      IF(SYST(MSP,L)) 2,2,5
    5 CONTINUE
      NNN=SYST(MSP,L)*PCT(I)
      SYST(MSP,L)=SYST(MSP,L)-NNN
      MSPTO=MSP
      IF(PERDTO(I).GT.0) MSPTO=PERDTO(I)
      SYS(MSPTO,L)=SYS(MSPTO,L)+NNN
      PRINT 3, OUTTT(I),OUTST(I),OUTTO(I),OUTSO(I),PCT(I),PERD(I) ,
     1SYST(MSP,L),SYS(MSPTO,L)
      IP=IP+1
      CALL PACK(K1,K3,K2,K4,NNN,0,MSPTO,L,N1,N2,N3)
      CALL RANWRITE(13,N1,3,IP)
    2 CONTINUE
      CALL RANWRITE(10,SYS,JK,J)
    1 CALL RANWRITE(10,SYST,JK,K)
C
C     TIMEUP FORMATS
    3 FORMAT(4I10,F8.4,4I10)
   13 FORMAT(7H0TIMEUP,2X,10HITRTIME = ,I4)
C
      RETURN
      END
```

FORTRAN DIAGNOSTIC RESULTS FOR TIMEUP

NO ERRORS

```
      SUBROUTINE OUTSIDE(I)
      COMMON      NOFIRST    ,NOLAST     ,NOTT       ,ITRTIME    ,INTEG
      COMMON      IP         ,INE(48)    ,IEND       ,OUTTT(200)
      COMMON      OUTST(200) ,OUTTO(200)             ,OUTSO(200)
      COMMON      PCT(200)   ,PFRD(200)  ,JK         ,SYST(48,48)
      COMMON      PA(100)    ,NAV(100)   ,LENGTH(100)            ,OUT(100)
      COMMON      INTOUR(220)            ,INSUB(220)             ,OUTTOUR(220)
      COMMON      IOD(220)   ,OUTSUB(220)            ,AFTER(220)
      COMMON      PER(220)   ,RATE(100)              ,NEEDS(10)  ,NE(10,10)
      COMMON      IOS(10)    ,NED(10)    ,NEE(100)   ,IFILL      ,MAXLEN
      COMMON      NTOUR      ,NP         ,CIOS       ,PERDTO(200)
      COMMON      PDW        ,LSTRSTO(10)            ,LSTRSTT(10)
      COMMON      LSTRSTS(10)            ,ITT        ,MAXSUB     ,ACT(10)
      COMMON      NSIT(10)   ,INOS(10,10)            ,IGRADE(220)
      COMMON       REP(220)  ,ITYPE(220)             ,PEROUT(220)
      COMMON      IDISTON    ,ISUM       ,GRPSUM(100)            ,PRIO(100)
      COMMON      BEGROW(100)            ,ENDROW(100)            ,BEGCOL(100)
      COMMON      ENDCOL(100)            ,MATSUM(100)            ,MATGRPS(100)
      COMMON      TYPE(100),SUB(100)     ,NUM(100)   ,ACTUAL(100)
      COMMON      NPRLEV     ,NT         ,IHOLD      ,LEN        ,LEVEL      ,M
      COMMON      NCRNODE(100)           ,MIN(100)   ,GRPTNPR(100)
      COMMON      MAXDEPL(100)
      INTEGER     TYPE       ,SUB        ,ACTUAL     ,ENDCOL
      INTEGER     CIOS       ,SYST       ,OUTSUB     ,OUTTOUR    ,AFTER
      INTEGER     ACT        ,OUTTO      ,OUTSO      ,OUTTT      ,OUTST
      INTEGER     PFRD       ,GRPSUM     ,BEGROW     ,ENDROW     ,BEGCOL
      INTEGER     PERDTO
      INTEGER OK
      N1=N2=N3=0
      K4=OUTSUB(I)
      K3=OUTTOUR(I)
      K2=INSUB(I)
      K1=INTOUR(I)
C       IFK4=0 THIS MEANS THAT PROGRAM IS USING VARIABLE INPUT
C       IOS(K4) = VECTOR OF NEW INPUT DURING THIS TIME PERIOD
      IF(IFILL-1) 5,5,10
   10 IFIND=NE(K1,K2)
      GO TO 87
    5 IFIND=NEEDS(K1)
   87 IF(IFIND.LE.0) GO TO 91
      IF(K4-1) 82,83,83
   82 IF(REP(I)-1.0) 84,85,85
   84 SSAVE=IFIND
      IFIND=PDW*SSAVE
      GO TO 86
   85 IFIND=SSAVE
   86 KK=IOS(1)
      IOS(1)=IFIND
      KJ=K4
      K4=1
   83 IF(PER(I)) 20,20,81
   81 OK=IOS(K4)*PER(I)
      IF(IOS(K4)- IFIND) 92,31,31
   92 IFIND=IFIND-OK
      IOS(K4)=IOS(K4)-OK
      GO TO 35
   20 IF(IOS(K4)-IFIND) 25,30,30
   25 IFIND=IFIND-IOS(K4)
      OK=IOS(K4)
      IOS(K4)=0
```

```
      GO TO 35
   30 OK=IFIND
   31 IOS(K4) = IOS(K4)-IFIND
      OK=IFIND
      IFIND=0
   35 IF(IFILL-1) 40,50,50
   50 NE(K1,K2)=NE(K1,K2)-OK
   40 NEEDS(K1)=NEEDS(K1)-OK
      INOS(K1,K2)=OK
      ICC=(K1-1)*MAXSUB+K2
      CALL RANREAD(10,SYST,JK,ICC)
      SYST(1,1)=SYST(1,1)+OK
      CALL RANWRITE(10,SYST,JK,ICC)
      IP=IP+1
      PRINT 1005,  K1,K2,K3,K4,OK,IP
      CALL PACK(K1,K2,K3,K4,OK,0,1,1,N1,N2,N3)
      CALL RANWRITE(13,N1,3,IP)
      ACT(K1)=ACT(K1)+OK
      ACTUAL(ICC)=ACTUAL(ICC)+OK
      IF(KJ-1) 89,89,88
   89 SSAVE=IFIND
      IOS(1)=KK
      K4=KJ
   88 CONTINUE
      CALL RANREAD(11,INE,MAXLEN,ICC)
      IF(INE(1)-1) 90,91,91
   90 INE(1)=1
      CALL RANWRITE(11,INE,MAXLEN,ICC)
   91 CONTINUE
C
C     OUTSIDE FORMATS
 1005 FORMAT  (12H OUTSIDE=--  12I8)
C
      RETURN
      END
```

FORTRAN DIAGNOSTIC RESULTS FOR OUTSIDE

NO ERRORS

```
      SUBROUTINE TRNSPOSE
      COMMON       NOFIRST    ,NOLAST    ,NOTT      ,ITRTIME   ,INTEG
      COMMON       IP         ,INE(48)   ,IEND      ,OUTTT(200)
      COMMON       OUTST(200),OUTTO(200)            ,OUTSO(200)
      COMMON       PCT(200)   ,PERD(200),JK         ,SYST(48,48)
      COMMON       PA(100)    ,NAV(100)  ,LENGTH(100)          ,OUT(100)
      COMMON       INTOUR(220)           ,INSUB(220)           ,OUTTOUR(220)
      COMMON       IOD(220)   ,OUTSUB(220)          ,AFTER(220)
      COMMON       PER(220)   ,RATE(100)            ,NEEDS(10),NE(10,10)
      COMMON       IOS(10)    ,NED(10)   ,NEE(100) ,IFILL     ,MAXLEN
      COMMON       NTOUR      ,NP        ,CIOS      ,PERDTO(200)
      COMMON       POW        ,LSTRSTO(10)          ,LSTRSTT(10)
      COMMON       LSTRSTS(10)           ,ITT       ,MAXSUB    ,ACT(10)
      COMMON       NSIT(10)   ,INOS(10,10)          ,IGRADE(220)
      COMMON        REP(220)  ,ITYPE(220)           ,PEROUT(220)
      COMMON       IDISTON    ,ISUM      ,GRPSUM(100)          ,PRIO(100)
      COMMON       BEGROW(100)           ,ENDROW(100)          ,BEGCOL(100)
      COMMON       ENDCOL(100)           ,MATSUM(100)          ,MATGRPS(100)
      COMMON       TYPE(100),SUB(100)    ,NUM(100) ,ACTUAL(100)
      COMMON       NPRLEV     ,NT        ,IHOLD     ,LEN       ,LEVEL     ,M
      COMMON       NCRNODE(100)          ,MIN(100) ,GRPINPR(100)
      COMMON       MAXDEPL(100)
      INTEGER      TYPE       ,SUB       ,ACTUAL    ,ENDCOL
      INTEGER      CIOS       ,SYST      ,OUTSUB    ,OUTTOUR   ,AFTER
      INTEGER      ACT        ,OUTTO     ,OUTSO     ,OUTTT     ,OUTST
      INTEGER      PERD       ,GRPSUM    ,BEGROW    ,ENDROW    ,BEGCOL
      INTEGER      PERDTO
      IF (IGRADE(IEND) .LE. 0 ) GO TO 999
      ENTRY FLIP
      DO 10 I=1,MAXLEN
      DO 10 J=I,MAXLEN
      ISAVE= SYST(I,J)
      SYST(I,J)= SYST(J,I)
   10 SYST(J,I)= ISAVE
  999 RETURN
      END
```

FORTRAN DIAGNOSTIC RESULTS FOR TRNSPOSE

NO ERRORS

```
      SUBROUTINE SUMMARY(NUMELEM,INDIV,INTOT)
C     APRIL 1969-PROGRAMMER WITT
C     SUMMARY SUBROUTINE-OBTAINS SUMS OF PARTIAL AND/OR COMPLETE MATRICE
C        S AND GROUPS OF MATRICES
C      SUBROUTINE PARAMETERS
C      IF INDIV .GT. 0.TOTALS FOR WHOLE MATRICES ARE OBTAINED.  VALUE OF INDIV
C        DETERMINES MATRIX TO BE SUMMED
C      INTOT CONTAINS MATRIX TOTAL
C
      COMMON        NOFIRST   ,NOLAST   ,NOTT      ,ITRTIME  ,INTEG
      COMMON        IP        ,INE(48)  ,IEND      ,OUTTT(200)
      COMMON        OUTST(200),OUTTO(200)          ,OUTSO(200)
      COMMON        PCT(200)  ,PERD(200),JK        ,SYST(48,48)
      COMMON        PA(100)   ,NAV(100) ,LENGTH(100)          ,OUT(100)
      COMMON        INTOUR(220)         ,INSUB(220)          ,OUTTOUR(220)
      COMMON        IOD(220)  ,OUTSUB(220)         ,AFTER(220)
      COMMON        PER(220)  ,RATE(100)           ,NEEDS(10),NE(10,10)
      COMMON        IOS(10)   ,NED(10)  ,NEE(100)  ,IFILL    ,MAXLEN
      COMMON        NTOUR     ,NP       ,CIOS      ,PERDTO(200)
      COMMON        PDW       ,LSTRSTO(10)         ,LSTRSTT(10)
      COMMON        LSTRSTS(10)         ,ITT       ,MAXSUB   ,ACT(10)
      COMMON        NSIT(10)  ,INOS(10,10)         ,IGRADE(220)
      COMMON         REP(220) ,ITYPE(220)          ,PEROUT(220)
      COMMON        IDISTON   ,ISUM     ,GRPSUM(100)          ,PRIO(100)
      COMMON        BEGROW(100)         ,ENDROW(100)          ,BEGCOL(100)
      COMMON        ENDCOL(100)         ,MATSUM(100)          ,MATGRPS(100)
      COMMON        TYPE(100),SUB(100)  ,NUM(100)  ,ACTUAL(100)
      COMMON        NPRLEV    ,NT       ,IHOLD     ,LEN      ,LEVEL     ,M
      COMMON        NCRNODE(100)        ,MIN(100)  ,GRPINPR(100)
      COMMON        MAXDEPL(100)
      INTEGER       TYPE      ,SUB      ,ACTUAL    ,ENDCOL
      INTEGER       CIOS      ,SYST     ,OUTSUB    ,OUTTOUR  ,AFTER
      INTEGER       ACT       ,OUTTO    ,OUTSO     ,OUTTT    ,OUTST
      INTEGER       PERD      ,GRPSUM   ,BEGROW    ,ENDROW   ,BEGCOL
      INTEGER       PERDTO
C
C
C     SECTION 1
C     SUMS INDIVIDUAL MATRICES
C     NUMELEM=NUMBER OF ELEMENTS IN MATGRPS VECTOR
C     MATGRPS=VECTOR WHICH DETERMINES WHICH MATRICES OR VECTORS ARE TO
C        BE SUMMED AND THE AGGREGATE SUMS TO BE OBTAINED
C
      IF(INDIV.GE.1)GO TO 4
      PRINT 100
      PRINT 103
    4 ISTART=1$ISUM=0
      J=0
      DO 10 I=1,NUMELEM
      MATSUM(I)=0
      IF(INDIV.GE.1)GO TO 2
      IF(MATGRPS(I).EQ.0)GO TO 5
      J=J+1
      MAT=MATGRPS(I)
      IBEGR=BEGROW(J)
      IENDR=ENDROW(J)
      IBEGC=BEGCOL(J)
      IENDC=ENDCOL(J)
      GO TO 3
    2 IBEGR=IBEGC=1
```

```
      IENDR=LENGTH(INDIV)
      IENDC=MAXLEN
      MAT=INDIV
    3 CALL RANREAD(10,SYST,JK,MAT)
      DO 1 K=IBEGR,IENDR
      DO 1 I=IBEGC,IENDC
    1 MATSUM(I)=MATSUM(I)+SYST(K,L)
      IF(INDIV.GE.1)GO TO 11
      PRINT 101,INTEG,MATGRPS(I),MATSUM(I),IBEGR,IENDR,IBEGC,IENDC
      GO TO 10
C
C
C     SECTION 2
C     SUMS GROUPS OF MATRICES AND/OR VECTORS
C     GRPSUM=SUMS OF GROUPS OF MATRICES OR VECTORS DESIGNATED IN MATGRPS
C        VECTOR
C
    5 IIEND=I-1
      ISUM=ISUM+1
      GRPSUM(ISUM)=0
      DO 8 K=ISTART,IIEND
    8 GRPSUM(ISUM)=GRPSUM(ISUM)+MATSUM(K)
      PRINT 102,     GRPSUM(ISUM)
      ISTART=I
   10 CONTINUE
      CALL RANWRITE(14,GRPSUM,ISUM,INTEG)
      GO TO 12
C
   11 INTOT=MATSUM(I)
   12 CONTINUE
C
C     SUBROUTINE SUMMARY FORMATS
  100 FORMAT(/31H0OUTPUT FROM SUBROUTINE SUMMARY/)
  101 FORMAT(I5,I8,I12,4X,I2,1H-,I2,3X,I3,1H-,I2)
  102 FORMAT(47X,I10)
  103 FORMAT(7H0PERIOD,3X,4HNODE,3X,10HNODE TOTAL,3X,4HROWS,3X,7HCOLUMNS
     13X,13HCLUSTER TOTAL)
C
      RETURN
      END

                 FORTRAN DIAGNOSTIC RESULTS  FOR    SUMMARY

NO ERRORS
```

```
      PROGRAM FILL
      COMMON A(48,48), NB(48,48)
      COMMON IBI(2000)
      REAL PER,N
      INTEGER A
C-----M= MATRIX SIZE(SQUARE)
      READ 101,M
      ISIZE=M*M
C-----ISTART = STARTING MATRIX----ISTOP = ENDING MATRIX(INCLUSIVE)
    1 READ 101, ISTART, ISTOP
      IF (ISTART-1) 999,2,2
    2 DO 602 IA= ISTART, ISTOP
      PRINT 107
      CALL RANREAD (10,A,ISIZE,IA)
      READ 101,K,IZERO
C-----K= NUMBER OF DATA CARDS
C-----IF IZERO EQ 0  MATRIX WILL BE ZEROED THEN DATA ENTERED
C-----IF IZERO IS 1 DATA WILL BE ENTERED INTO EXISTING MATRIX
      IF (IZERO-1) 3,4,4
    3 DO 10 I=1,M
      DO 10 J=1,M
   10 A(I,J) = 0
    4 II=N=I=J=PER=0.0
      DO 20 KA=1,K
      READ 100, N,I,J,PER
      IF (I.LE.0) GO TO 20
      IF (J.LE.0) GO TO 20
C-----N= NUMBER  I4 FORMAT
C-----I= STARTING ROW
C-----J= STARTING COLUMN
C-----PER = % OF N IN EACH ELEMENT
C-----IF PER = 0,100 PERCENT IS ASSUMED
      IF (PER.EQ.0) 15,16
   15 PER =1
   16 II=100./ (PER*100.)
      DO 20 L=1,II
      NA= N* PER
      IF (I.GT.M) 23,24
   24 IF (J.GT.M) 23,25
   25 IF (IZERO-1) 26,21,21
   26 IF  (A(I,J)) 22,21
   21 A(I,J) = NA
      I=I+1
      J=J+1
      GO TO 20
   22 PRINT 103,I,J
      L=II
      GO TO 20
   23 PRINT 104, I,J
      L=II
   20 CONTINUE
   30 CALL RANWRITE  (10,A,ISIZE,IA)
      PRINT 106,IA
      CALL RANREAD  (10,A,ISIZE,IA)
      DO 31 I= 1,M
      DO 31 J=I,M
      ISAVE= A(I,J)
      A(I,J)= A(J,I)
   31 A(J,I)= ISAVE
      IAI=0
```

```
      JII=0
      DO 600  JI=1,ISIZE,M
      JII= JII+1
      IIA= JI+M-1
      DO 600 J= JI,IIA
      IF (A(J)) 601,600,601
  601 IAI= IAI+3
      IBI (IAI-1)= JII
      IBI(IAI)= A(J)
      IBI(IAI-2)= J-JI+1
  600 CONTINUE
      IF (IAI) 602,602,603
  603 PRINT 604, (IBI(J), J=1,IAI)
  604 FORMAT  (7(4X,2I4,I7))
  602 CONTINUE
      GO TO 1
C-----DATA FORMAT
  100 FORMAT  (F6.0,2I2,F2.2)
  101 FORMAT  (2I4)
  103 FORMAT (14H DATA FOR ROW I2, 5H COL I2,/ 38H NOT ENTERED    PRIOR D
     XATA STORED THERE)
  104 FORMAT  (14H DATA FOR ROW I2,5H COL I2,/ 34H ENTERED BUT   DATA EXC
     XEEDS MATRIX )
  106 FORMAT  (1X,I3)
  107 FORMAT  (///5X, 3HCOL ,4H ROW   )
  999 END
```

FORTRAN DIAGNOSTIC RESULTS FOR FILL

```
 NULL STATEMENT NUMBERS
      30

ERROR TYPE 1046 DETECTED AT      1 STATEMENT  BEYOND STATEMENT NO.    22
 THE RUNNING INDEX IN A DO MAY BE CHANGED WITHIN THE LOOP.

ERROR TYPE 1046 DETECTED AT      1 STATEMENT  BEYOND STATEMENT NO.    23
 THE RUNNING INDEX IN A DO MAY BE CHANGED WITHIN THE LOOP.
COMPASS.P.L.X.
```

```
                                                      ENTRY   PACK,UNPACK
00000  01077777 01 0   77777 0  PACK                  UJP     **
00001  47100107 47 0  P00107 1                        STI     XRT,1
00002  14600000 14 1   00000 2                        ENA     0B
00003  14700000 14 1   00000 3                        ENQ     0B
00004  54100000 54 0  P00000 1                        LDI     PACK,1
00005  21500006 21 1   00006 1                        LDQ,I   6,1
00006  13000030 13 0   00030 0                        SHAQ    24
00007  21500007 21 1   00007 1                        LDQ,I   7,1
00010  12400022 12 1   00022 0                        SHQ     18
00011  13000006 13 0   00006 0                        SHAQ    6
00012  21500010 21 1   00010 1                        LDQ,I   8,1
00013  12400022 12 1   00022 0                        SHQ     18
00014  13000006 13 0   00006 0                        SHAQ    6
00015  21500011 21 1   00011 1                        LDQ,I   9,1
00016  12400022 12 1   00022 0                        SHQ     18
00017  13000006 13 0   00006 0                        SHAQ    6
00020  40500012 40 1   00012 1                        STA,I   10,1
00021  21500004 21 1   00004 1                        LDQ,I   4,1
00022  13000030 13 0   00030 0                        SHAQ    24
00023  21500000 21 1   00000 1                        LDQ,I   0,1
00024  12400024 12 1   00024 0                        SHQ     20
00025  13000004 13 0   00004 0                        SHAQ    4
00026  21500001 21 1   00001 1                        LDQ,I   1,1
00027  12400024 12 1   00024 0                        SHQ     20
00030  13000004 13 0   00004 0                        SHAQ    4
00031  40500013 40 1   00013 1                        STA,I   11,1
00032  21500005 21 1   00005 1                        LDQ,I   5,1
00033  13000030 13 0   00030 0                        SHAQ    24
00034  21500002 21 1   00002 1                        LDQ,I   2,1
00035  12400024 12 1   00024 0                        SHQ     20
00036  13000004 13 0   00004 0                        SHAQ    4
00037  21500003 21 1   00003 1                        LDQ,I   3,1
00040  12400024 12 1   00024 0                        SHQ     20
00041  13000004 13 0   00004 0                        SHAQ    4
00042  40500014 40 1   00014 1                        STA,I   12,1
00043  01000105 01 0  P00105 0                        UJP     BACK
00044  01077777 01 0   77777 0  UNPACK                UJP     **
00045  47100107 47 0  P00107 1                        STI     XRT,1
00046  14700000 14 1   00000 3                        ENQ     0B
00047  54100044 54 0  P00044 1                        LDI     UNPACK,1
00050  20500012 20 1   00012 1                        LDA,I   10,1
00051  13000006 13 0   00006 0                        SHAQ    6
00052  41500006 41 1   00006 1                        STQ,I   6,1
00053  14700000 14 1   00000 3                        ENQ     0B
00054  13000006 13 0   00006 0                        SHAQ    6
00055  41500007 41 1   00007 1                        STQ,I   7,1
00056  14700000 14 1   00000 3                        ENQ     0B
00057  13000006 13 0   00006 0                        SHAQ    6
00060  41500010 41 1   00010 1                        STQ,I   8,1
00061  12077755 12 0   77755 0                        SHA     -18
00062  40500011 40 1   00011 1                        STA,I   9,1
00063  20500013 20 1   00013 1                        LDA,I   11,1
00064  14700000 14 1   00000 3                        ENQ     0B
```

```
00065  13000020 13 0  00020 0           SHAQ      16
00066  41500004 41 1  00004 1           STQ,I     4,1
00067  14700000 14 1  00000 3           ENQ       0
00070  13000004 13 0  00004 0           SHAQ      4
00071  41500000 41 1  00000 1           STQ,I     0,1
00072  12000004 12 0  00004 0           SHA       4
00073  40500001 40 1  00001 1           STA,I     1,1
00074  20500014 20 1  00014 1           LDA,I     12,1
00075  14700000 14 1  00000 3           ENQ       0B
00076  13000020 13 0  00020 0           SHAQ      16
00077  41500005 41 1  00005 1           STQ,I     5,1
00100  14700000 14 1  00000 3           ENQ       0
00101  13000004 13 0  00004 0           SHAQ      4
00102  41500002 41 1  00002 1           STQ,I     2,1
00103  12000004 12 0  00004 0           SHA       4
00104  40500003 40 1  00003 1           STA,I     3,1
00105  15100015 15 0  00015 1   BACK    INI       13,1
00106  47100110 47 0 P00110 1           STI       **+2,1
00107  14177777 14 0  77777 1   XRI     ENI       **,1
00110  01077777 01 0  77777 0           UJP       **
                                        END

                                  NUMBER OF LINES WITH DIAGNOSTICS
```


3300 COMPASS/MSOS-COMPASS V   X59B

```
ENTRY-POINT SYMBOLS
    PACK            00000
    UNPACK          00044

LENGTH OF SUBPROGRAM       00111
LENGTH OF COMMON           00000
LENGTH OF DATA             00000
```

3300 COMPASS/MSOS-COMPASS V MX

ENTRY-POINT SYMBOLS

```
    MX                  00001
```

```
LENGTH OF SUBPROGRAM      00023
LENGTH OF COMMON          00000
LENGTH OF DATA            00000
```


3300 COMPASS/MSOS-COMPASS V MX

```
                                                      ENTRY    MX
00000  14177777 14 0  77777 1  BACK                   ENI      **,1
00001  01077777 01 0  77777 0  MX                     UJP      **
00002  47100000 47 0 P00000 1                         STI      BACK,1
00003  54100001 54 0 P00001 1                         LDI      MX,1
00004  14600002 14 1  00002 2                         ENA      2
00005  34000001 34 0 P00001 0                         RAD      MX
00006  20100000 20 0  00000 1                         LDA      0,1
00007  44000014 44 0 P00014 0                         SWA      I1
00010  44000016 44 0 P00016 0                         SWA      I2
00011  20500001 20 1  00001 1                         LDA,I    1,1
00012  53500000 53 1  00000 1                         TAI      1
00013  15177776 15 0  77776 1                         INI      -1,1
00014  20177777 20 0  77777 1  I1                     LDA      **,1
00015  15177776 15 0  77776 1                         INI      -1,1
00016  21177777 21 0  77777 1  I2                     LDQ      **,1
00017  03600021 03 1 P00021 2                         AQJ,GE   **2
00020  13000030 13 0  00030 0                         SHAQ     24
00021  02500016 02 1 P00016 1                         IJD      I2,1
00022  01000000 01 0 P00000 0                         UJP      BACK
                                                      END

                                                    NUMBER OF LINES WITH DIAGNOSTICS
```