AD-A061 389 DYNAMICS RESEARCH CORP WILMINGTON MASS F/G 5/9
DIGITAL AVIONICS INFORMATION SYSTEM (DAIS). VOLUME II. TRAINING--ETC(U)
SEP 78 A J CZUCHRY, K M DOYLE, J T FRUEH F33615-75-C-5218
UNCLASSIFIED AFHRL-TR-78-58(II) NL

1 OF 1
AD
A061 389

END
DATE
FILMED
2-79
DDC

(18) (19) AFHRL-TR-78-58(II)

LEVEL II (2)

AIR FORCE HUMAN RESOURCES LABORATORY

DDC FILE COPY AD A061389

(6) DIGITAL AVIONICS INFORMATION SYSTEM (DAIS). Volume II.
TRAINING REQUIREMENTS ANALYSIS MODEL USERS GUIDE

By

(10) Andrew J. Czuchry
Kristy M. Doyle
Jonathan T. Fruehling
Dynamics Research Corporation
60 Concord Street
Wilmington, Massachusetts 01887

H. Anthony Baran
Duncan L. Dieterly, Major, USAF

ADVANCED SYSTEMS DIVISION
Wright-Patterson Air Force Base, Ohio 45433

(11) Sept[illegible] 1978

(12) 79 P.

(9) Final Report, [illegible] May 1975 – Jan[illegible] 1978

(15) F33615-75-C-5218

(16) 2051 (17) 44

AD56 520

388 902

AIR FORCE SYSTEMS COMMAND
BROOKS AIR FORCE BASE, TEXAS 78235

78 11 13 054

NOTICE

When U.S. Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This final report was submitted by Dynamics Research Corporation, 60 Concord Street, Wilmington, Massachusetts 01887, under contract F33615-75-C-5218, project 2051, with Advanced Systems Division, Air Force Human Resources Laboratory (AFSC), Wright-Patterson Air Force Base, Ohio 45433. Mr. H. Anthony Baran was the contract monitor.

This report has been reviewed and cleared for open publication and/or public release by the appropriate Office of Information (OI) in accordance with AFR 190-17 and DoDD 5230.9. There is no objection to unlimited distribution of this report to the public at large, or by DDC to the National Technical Information Service (NTIS).

This technical report has been reviewed and is approved for publication.

GORDON A. ECKSTRAND, Director
Advanced Systems Division

RONALD W. TERRY, Colonel, USAF
Commander

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER AFHRL-TR-78-58(II)	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE *(and Subtitle)* DIGITAL AVIONICS INFORMATION SYSTEM (DAIS): TRAINING REQUIREMENTS ANALYSIS MODEL USERS GUIDE	5. TYPE OF REPORT & PERIOD COVERED Final May 1975 – January 1978
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Andrew J. Czuchry, H. Anthony Baran, Kristy M. Doyle, Duncan L. Dieterly, Jonathan T. Frueh	8. CONTRACT OR GRANT NUMBER(s) F33615-75-C-5218
9. PERFORMING ORGANIZATION NAME AND ADDRESS Dynamics Research Corporation 60 Concord Street Wilmington, Massachusetts 01887	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 63243F 20510001
11. CONTROLLING OFFICE NAME AND ADDRESS HQ Air Force Human Resources Laboratory (AFSC) Brooks Air Force Base, Texas 78235	12. REPORT DATE September 1978
	13. NUMBER OF PAGES 80
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)* Advanced Systems Division Air Force Human Resources Laboratory Wright-Patterson Air Force Base, Ohio 45433	15. SECURITY CLASS. *(of this report)* Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

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

18. SUPPLEMENTARY NOTES

The research reported herein was sponsored jointly by Air Force Human Resources Laboratory, Air Force Avionics Laboratory, and Air Force Logistics Command. It was performed and funded as part of the Digital Avionics Information System Advanced Development Program.

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

digital avionics information system
life cycle costing
task characteristic parameter
training block
training plan
training program
training requirements analysis model

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

The training requirements analysis model (TRAMOD) described in this user's guide represents an important portion of the larger effort called the Digital Avionics Information System (DAIS) Life Cycle Cost (LCC) Study. TRAMOD is the second of three models that comprise a LCC modeling system for use in the early stages of system development. As part of the overall modeling system, the training model is an efficient tool for developing training programs on the basis of task, time, and resource criteria. A data base containing information associated with these criteria is also included. The interactive nature of TRAMOD affords the user great flexibility in structuring its operation while retaining the capability of addressing specific training problems in depth. This guide explains the available options and illustrates the manner in which user/model interaction is accomplished.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

SUMMARY

This volume is a user's guide to the operation of the training requirements analysis model (TRAMOD) described in Volume I of this report. It introduces TRAMOD by describing its functions, the logical operations and subroutines, input data requirements, and available outputs. The reader is guided through a typical working session with the model which includes examples of the ways in which user/model interaction is accomplished. Primarily intended to provide the user with the necessary information to maximize the effectiveness of the session, the guide explains the available user options and illustrates the interactive choices which may be made along the option paths. Use of TRAMOD requires a remote terminal and familiarity with the material covered in Volume I of this report. That volume provides a more detailed description of TRAMOD's design, inputs, purpose, and application.

TRAMOD is a computerized analytical model, one of a group which comprises a life cycle cost (LCC) modeling system for use in the early stages of system design. It is, however, operable in a stand-alone mode and is a powerful tool for use in a wide variety of situations calling for the selection of training approaches and programs on the basis of task, time, and resource criteria. General applicability is achieved by allowing the user complete control of task definition. Because of this feature, its extreme flexibility, and its interactive nature, the most suitable operator of TRAMOD is either a behavioral scientist or a training analyst, i.e., a person familiar with at least the fundamentals of human learning theory.

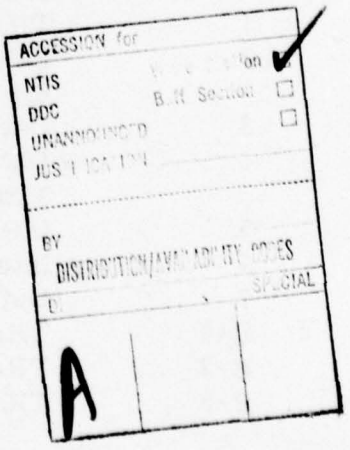
TRAMOD consists of three components which are operated in iterative sequences. They are a task block generator, a training plan generator, and a training program generator. Each component is operable independently, performing certain subfunctions to produce the product for which it is named. The data bank, which is input to the task block generator, consists of tasks which are described in terms of a series of parameters which reflect task training characteristics. TRAMOD currently makes use of five. The number of these task characteristic parameters, as well as their specification and assigned values, are user-defined options in data bank construction. The data bank also contains training time and cost information. The model affords the user a mechanism to exercise interactive control over these data through their selection or deletion in any given operation.

PREFACE

This two volume report describes the Training Requirements Analysis Model (TRAMOD). Volume I describes the model and its development. This volume is a user's guide to its operation and potential use. The report is one of a series of technical reports, data banks, and computer programs produced under contract no. F33615-75-C-5218, "DAIS Life Cycle Costing Study." This study, in conjunction with present Air Force capabilities is to provide the means for assessing the life cycle cost impact of the operational implementation of the Digital Avionics Information System (DAIS).

The study was directed by the Advanced Systems Division, Air Force Human Resources Laboratory, Wright-Patterson Air Force Base, Ohio, and is documented under Work Unit 20510001, "DAIS Life Cycle Costing Study." It was performed under Air Force Avionics Laboratory Program Element 63243F, "Digital Avionics Information System," Project 2051. Project 2051, "Impact of DAIS on Life Cycle Costs," is jointly sponsored by the Air Force Human Resources Laboratory, Air Force Avionics Laboratory, and the Air Force Logistics Command. Contract funds were provided by the Air Force Avionics Laboratory. The DAIS Program Manager is Lt. Col. Robert A. Dessert, the Air Force Human Resource Laboratory Project Scientist is Mr. H. Anthony Baran, and the Air Force Logistics Command Project Officer is Capt. Ronald Hahn. The latter two are DAIS Deputy Directors. The Contractor Program Manager is Mr. John Goclowski.

TABLE OF CONTENTS



LIST OF ILLUSTRATIONS

LIST OF TABLES

DIGITAL AVIONICS INFORMATION SYSTEM (DAIS): TRAINING REQUIREMENTS ANALYSIS MODEL USER'S GUIDE

1. GENERAL INFORMATION AND APPLICATION

1.1 INTRODUCTION

The training model (TRAMOD) described in this report is a powerful tool for use in deciding which among a group of tasks should be trained, and how (given a series of constraints) to best accomplish a well defined set of training objectives. Its adaptability stems from its ability to allow a user to structure its operation and decision criteria to a wide variety of situations while retaining the capability of addressing specific training problems in depth. Designed for use early in the systems acquisition process to facilitate the identification and timely consideration of potential training problems, its primary objective is to enhance Air Force capability to avoid potential training problem situations through action within the design process itself. Its extreme flexibility and amenability to user interaction throughout its operation make it appropriate for use in an extremely wide variety of training related investigations.

1.2 MODEL DESCRIPTION

TRAMOD was designed to operate in conjunction with a Reliability and Maintainability (R&M) model and a system cost model to form a life cycle cost impact modeling system (LCCIM). That system was developed as part of and will be applied within the Digital Avionics Information System advanced development program. Each model within the LCC modeling system addresses a component function of life cycle costing. Together they form a tool capable of assessing and combining the elements of system LCC in a way which can enhance its consideration throughout the systems acquisition process, especially during its early phases. This is accomplished by providing information needed to perform trade-offs within the design process which can result in reducing the LCC of new systems. TRAMOD addresses the training requirements element of system LCC. The R&M model addresses other elements such as system operation and maintenance, and system support resource utilization. The system cost model provides the cost data integration and computational capability of the LCCIM.

Figure 1 shows a block diagram of TRAMOD indicating its three analytic components, the required inputs, and the resulting outputs. The major input to the model is the data bank consisting of tasks and their behaviors evaluated in terms of selected job-related characteristics. The remaining inputs are entered interactively during the execution of TRAMOD. They identify the user's policy criteria and resource constraints and thus serve to direct the entire analytic process. The major output of TRAMOD is the training program; the selected task blocks and the training plan are intermediate outputs which are important aids in understanding and controlling the execution of the model. Table 1 identifies the functions performed in each component.

1.3 MODEL OPERATION

The user, preferably a behavioral scientist or training analyst, first selects the criteria for determining which tasks are to be trained and inputs this information to the first component of the model: the task block generator. This component tests the training characteristics of each task, presently quantified in the training data bank in terms of five parameters, against the user-specified selection criteria. These criteria are essentially a set of parameter values which the model uses as cut-off points. The task blocks to be trained are then displayed along with descriptions of the selected tasks in terms of the parameters: taxonomic grouping and level (cognitive and psychomotor); learning difficulty; criticality; frequency; and a parameter which serves to collate tasks that are best trained as a block. The user has the option of iteratively exercising this component and varying the selection criteria upon which it operates.

The selected task blocks become the input data set for the second component of the model: the training plan generator. The user must also specify the requirements, a dollar cost expense limit, and a time limit in months. The training plan generator component then uses a combination of optimizing and mapping techniques to generate a training plan. This output consists of the specific task blocks to be trained with recommendations concerning: training mode (school or OJT); method of instruction (simulation, performance, lecture, demonstration, etc.); and media (simulator, mockups, actual equipment, etc.). It also includes a time estimate and a cost estimate.

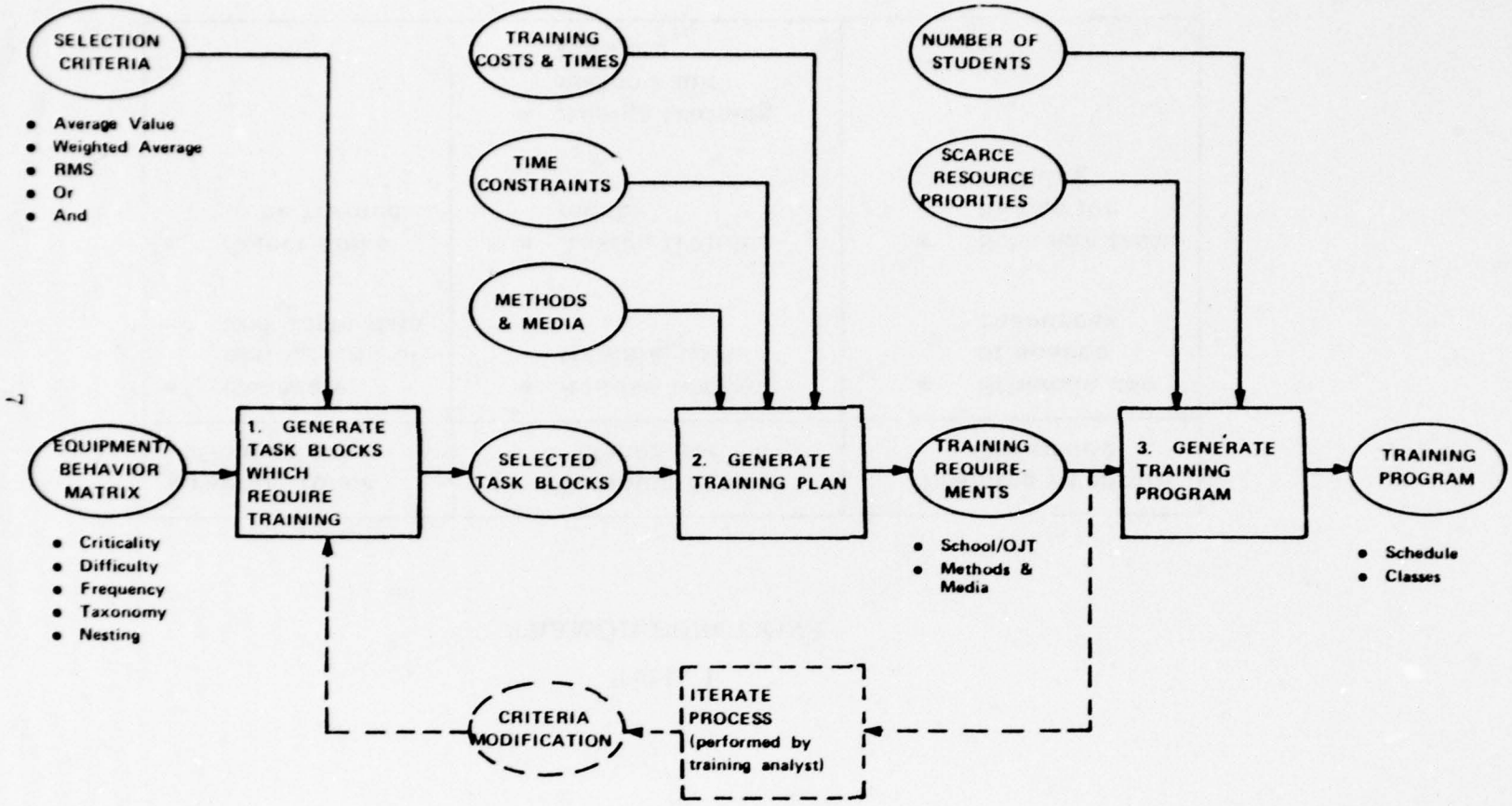


Figure 1 TRAINING REQUIREMENTS ANALYSIS MODEL

Table 1

TRAMOD FUNCTIONS

Training Block Generator	Training Plan Generator	Training Program Generator
• Generate histograms of task input data	• Assess training times & costs	• Schedule use of scarce resources
• Select tasks to be trained	• Assign training mode	• Schedule task blocks for training
	• Assign training method and medium	

After reviewing the initial training plan, the user has the option of selecting different sets of training criteria and iteratively exercising this component to generate alternative training plans. This is quickly accomplished, affording a means for rapid hypothesis testing and the comparative analysis of the effects of varying training plan constraints.

Once a training plan has been generated which is acceptable to the user, it becomes the input data set for the training program generator component. This component affords the user the options of selecting a task characteristic to govern the training sequence of the task blocks and designating one of the training media as a scarce resource requiring prioritized use. A scheduler within the component then generates the schedule arrangements that are possible and applies any optional constraints which the user may have specified. The final output is a training program which consists of task blocks to be trained, appropriate training methods and media, training modes and training times. At that point, TRAMOD is programmed to return to the beginning of the entire operation to process tasks for the next trainee group.

1.4 HOST COMPUTER CONSIDERATIONS

TRAMOD is programmed for operation on the CDC-6600 computer. Consequently, the FORTRAN programming language used by the model responds to the syntax of the CDC 6600 FORTRAN compiler. Once the program has been loaded into the host computer with all data files, all that is needed for execution is a remote terminal. The program is interactive and is run on-line utilizing the CDC 6600 time-sharing system. Details concerning terminal interaction with the model will be discussed later in this report. Applicable FORTRAN syntax statements, as used in the program, will be illustrated.

1.5 DATA BANK

Operation of TRAMOD requires a task data bank containing an array of user defined tasks described in terms of parameters which reflect task training characteristics. Task data are structured in matrix form as shown in Figure 2. Tasks and subtasks are coded in terms of the equipment (system, subsystem, line replaceable unit), the maintenance event that is required to restore the equipment to operational readiness (or the operational activity required), and the behavior(s) to be acquired. Each behavior will have been previously quantified by a training analyst in terms of the task characteristic parameters (see Table 2).

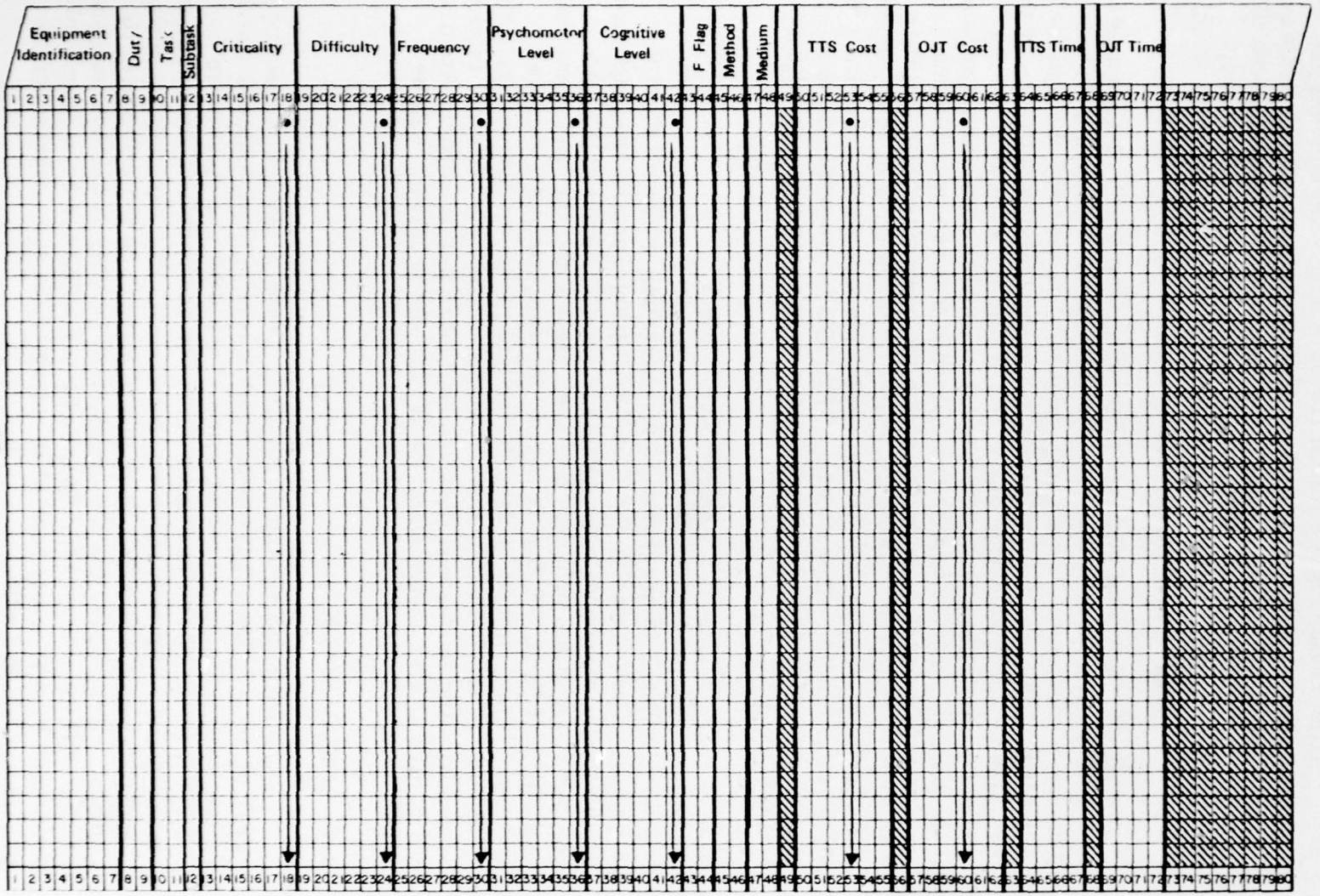


Figure 2 TRAMOD – CARD LAYOUT FORM

II. SYNTAX FOR RESPONSES TO PROGRAM TRAMOD

2.1 INITIATION

Once the user has set up the terminal and established a data link with the host computer, he is ready to call up program TRAMOD. The following is a sample initiation procedure. The first three commands call up the task dictionary, data bank, and model files; the fourth command begins execution of TRAMOD.

Example Initiation Procedure

```
COMMAND- ATTACH,TRAMOD

 PFN IS
 TRAMOD
 PF CYCLE NO. = 001
COMMAND- ATTACH,TASKS

 PFN IS
 TASKS
 PF CYCLE NO. = 001
COMMAND- ATTACH,HDB

 PFN IS
 HDB
 PF CYCLE NO. = 002
COMMAND- EDITOR

..EDIT,TRAMOD

..RUN,FTN

    45000B    CM STORAGE USED
       5.692 CP SECONDS COMPILATION TIME

ENTER AFSC-
```

At this point the program has been initiated and the input data file is being read.

2.2 INPUTS

There are two types of inputs to TRAMOD: off-line (batch) and on-line (terminal). Maintained in a single file, off-line input data are contained in task data records, each of which pertains to a single task within the group from which those to be trained will be selected. Each task data record is comprised of previously determined values for each of the task characteristic parameters (presently five) upon which TRAMOD operates. Those parameters are: (1) criticality; (2) learning difficulty; (3) frequency; (4) psychomotor level; and (5) cognitive level. When appropriate, the input data also include flags used to force a task to demand a specific training mode, method, or medium.

Also included on each task data record is the estimated time in hours required to train a person to perform that task. These times are provided for both Technical Training School (TTS) training and on-the-job training (OJT). The estimated cost per hour for each type of training is also listed.

Sources of and methods for assigning data values are described in Volume I of this report. With the exception of frequency, the values assigned to the task characteristic parameters should be based on the judgement of engineers and technicians familiar with the equipment upon which the tasks will be performed. The frequency should be obtained from actual reliability and maintainability (R&M) data for that equipment.

In very early stages of the design process of new systems, data values could be obtained by comparing the new system's equipment to operational equipment for which actual data is available, computing the necessary factor(s) to reflect equipment differences, and applying the factor(s) to the operational equipment data to generate data applicable to the new system. Once this is done, the DAIS R&M model [3] can be used to advantage in organizing that data for direct input to TRAMOD.

Course length and cost data for TTS training are available from actual Air Force training course outlines and cost estimates produced by the Air Training Command. OJT training times and associated costs must be estimated until more definitive information becomes available. As was the case for R&M data, course length and cost data for new systems must be obtained by means of a comparative analysis and subsequent extrapolation from the old to the new.

2.2.1 Data File Inputs

Table 2 is a description of all the fields for the off-line file (batch) input data elements. They are explained in sufficient detail to allow the model user to format the data required to be included in that file. Table 3 provides the user with duty/task descriptor codes, and is to be used in conjunction with Table 2. Table 4 further describes the fields in terms of card column layout and justification. Figures 2 and 3 are, respectively, an example of a card layout work sheet showing how the data should be prepared for keypunch and a sample card from an input file deck.

2.2.2 Interactive Inputs

Prior to operating the model, the user should be aware of certain responses (see Table 5) that the model prompts the user to make during the course of its operation. Footnoted items are required inputs. Inclusion of the others is discretionary.

2.3 OUTPUTS

All outputs of the model are generated during the course of operation in its interactive mode and (see Table 6) are displayed on the user's terminal. The sample program run provided in Appendix A shows the locations of these displays within the interactive process.

Table 2

DESCRIPTION OF INPUT DATA ELEMENTS KEY FIELDS

Columns	Identifier - Definition
1-7	Equipment Identification Number - defines the equipment in a series of codes showing (Column 1) the type of weapon system, (2) the major system within the weapon system, (3) functional groupings of the major system, and (4-7) provides a numerical breakdown by operational function (e.g., radar navigation, radio navigation, or bombing navigation), subsystem, line replaceable unit, and shop replaceable unit. The following applies for DAIS: Column 2 - major system A - avionics Column 3 - functional groups A - air-ground attack C - communications I - instruments M - miscellaneous N - navigation Z - core
8-12	Duty/Task/Subtask Identification - defines the job in a series of codes showing (8-9) the duties of the trainee, (10-11) tasks to be performed and, (12) the subtasks that further define elements and/or knowledge aspects of the task required for their successful completion. (See Table 3 for assigned codes.)
13-18	Task Criticality - it is necessary to evaluate each task/function within a maintenance event with respect to criticality. Inevitably, almost all tasks are of a high level of criticality in insuring the ultimate success of a mission. However, as the training of personnel to perform tasks is an end in itself, the tasks may be assigned individual levels of criticality within the content of each event. There are three levels of task criticality used for TRAMOD operation. They are defined as follows (column 18 is decimal filled): (1) Non-critical - tasks that, if not performed correctly and/or to standards, would not degrade the overall effectiveness of the event, but which might affect the efficiency of the performance.

(3) Semi-critical - tasks that, if not performed correctly and/or to standards, would not substantially degrade the effectiveness of the maintenance event, but which, if performed correctly and to standards, would lead to an efficient and effective overall maintenance event performance.

(5) Critical - tasks that, if not performed correctly and to standards, would seriously affect the effectiveness and success of the maintenance event.

(0) Not-applicable - tasks that are not applicable to a given equipment or system. This designation may be used by the computer to allow it to bypass the other input characteristic parameters.

19-24 Learning Difficulty - the learning difficulty of a task/function may be expressed as a function of the time it takes to learn to perform the task relative to the population of learning times across all tasks associated with the same system. This relationship is convenient and sufficient for the purpose at hand. However, its limitation is realized as is the fact that the time involved in the learning process is a function of the interaction of many variables including effort, complexity, and practice. Although levels of task difficulty provided by recent USAF occupational surveys are based on a scale of one to nine, for the purposes of this data base, five levels were used. They are defined as follows (column 24 is decimal filled):

(1) Extremely low - very much less than the mean value for learning times across all tasks associated with the subsystem studies.

(2) Low - somewhat less than the mean value for learning times across all tasks associated with the subsystem studied.

(3) Average - approximating the mean value for learning times across all tasks associated with the subsystem studied

(4) High - somewhat more than the mean value for learning times across all tasks associated with the subsystem studied

(5) Extremely high - very much more than the mean value for learning times across all tasks associated with the subsystem studied.

25-30 Task Frequency - frequency of task/function occurrence is a measure of the exposure time of a trainee to each task he encounters when performing his duties. For the shop, flightline, and support equipment maintenance duties, the exposure time is obtained by exercising the following equation using reliability and maintainability data/estimates for the subsystem studied.

$$MI = \frac{\text{MTTR by Maintenance Event}}{\text{MFHBMA}}$$

where:

MI is the maintenance index of the time taken to perform a maintenance action on a given subsystem for each flight hour of operation.

MTTR is the mean time to repair given that a maintenance action is required. This value is calculated by multiplying the average time it takes to perform a task event by the probability of occurrence of that event.

MFHBMA is the mean flight hours between maintenance actions.

Five levels are used to record task frequency for TRAMOD operations. These values are obtained from the maintenance index values (MIs) of the subsystems for each maintenance event combination that requires the tasks of interest to be performed. The scale used to obtain the levels represents the linearly partitioned relative weighting of the logarithmic plots of the MIs. In other words, the logarithmic values of the MIs were divided into five discrete increments to obtain their relative level as defined below (column 30 is decimal filled):

(1) Extremely low - task is performed infrequently
(2) Low - task is performed at a rate less than the average
(3) Average - task is performed at the average rate
(4) High - task is performed at a rate above the average
(5) Extremely high - task is performed frequently

31-36 Task Psychomotor Level - each task/function of a maintenance event entails some level of conscious, physical action in response to sensory inputs. The degree of visual acuity, reaction time, manual dexterity, multilimb coordination, finger dexterity, arm-hand steadiness, control, precision or interactions of any of the above psychomotor factors, as measured by the amount of practice required to learn and apply each task, were chosen to serve as bases for evaluating task levels. As no suitable specific taxonomy exists for either defining or measuring the psychomotor levels of a maintenance task, a measurement criterion, presented below, was constructed to serve the needs of TRAMOD operation. An attempt is made to clarify the definitions by example (column 36 is decimal filled):

(1) Imitation - task demands little or no practice to perform. Only routine motor skills and perceptual discriminations are needed. Task performance may require instruction and illustration for a few simple parts.

(2) Manipulation - task requires some practice either to integrate routine motor skills and perceptions or to perfect certain motor coordinations or perceptual discriminations. Task performance may be completed for the most part without assistance other than reference material. Speed is not critical.

(3) Precision - task requires moderate practice to perfect or integrate the perceptual motor skills. Task performance demands the ability to do all parts of the task (at minimum recommended level) unassisted with reasonable speed and accuracy. Inspection/verification of performance may be necessary.

(4) Articulation - task requires much practice to acquire the motor coordination and/or perceptual discriminations necessary for proficient performance in all activities of all parts of the task. High accuracy but not necessarily high speed is needed.

(5) Naturalization - task requires a great deal of practice to acquire the motor coordinations and perceptual discriminations necessary for proficient performance. Task requires highest speed and accuracy with maximum skill production without the use of reference materials.

37-42 Task Cognitive Level - each task/function within a maintenance event may be described in terms of the relative cognitive (knowledge) level needed by a person to learn or perform it. As no suitable specific taxonomy exists for measuring the cognitive levels required of a technician performing a maintenance task, a measurement criteria, presented below, was constructed to serve the needs of TRAMOD. These definitions are derived from a combination of the Specialty Training Standards (STS) proficiency code definitions and Bloom's [1] cognitive level definitions. An attempt is made to clarify the definitions by example. The five levels assigned are:

(1) Comprehension - task requires that basic facts and nomenclature be known for successful performance.

(2) Application - task requires that the principles and procedures involved be known and used for successful performance.

(3) Analysis - task requires that operating principles be understood and an ability to draw rudimentary conclusions concerning the subject matter. Technician should be able to evaluate the relevancy of data.

(4) Synthesis - task requires that considerable theory be known and an ability to evaluate conditions.

(5) Evaluation - task requires all of the above levels plus that of making predictions or decisions requiring a complete understanding of underlying theory.

43-44 F Flag - training mode output forcing code

(1) Technical Training School (TTS)
(2) On-the-Job Training (OJT)
(0) Let model decide

45-46 Method - training method output forcing code

(1) information lecture
(2) discussion
(3) demonstration
(4) simulation
(5) performance
(0) let model decide

47-48 Medium - training medium output forcing code

(1) transparencies
(2) training films
(3) mock-ups
(4) simulator
(5) actual equipment
(0) let model decide

50-55 TTS Cost - technical training school cost/hour for the current task (column 53 is decimal filled)

57-62 OJT Cost - on-the-job training cost/hour for the current task (column 60 is decimal filled)

64-67 TTS Time - time in hours for TTS training for current task

69-72 OJT Time - time in hours for OJT training for current tasks

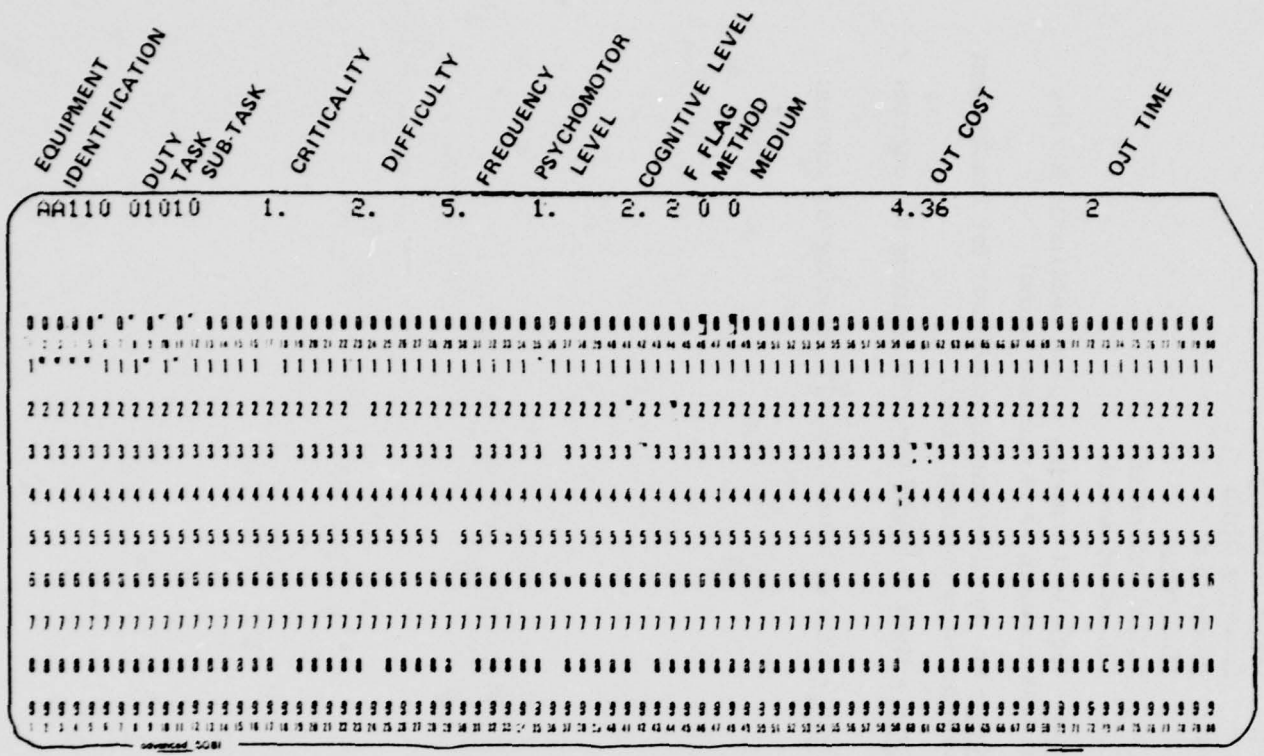


Figure 3 SAMPLE INPUT DATA CARD

Table 3

DUTY/TASK DESCRIPTOR FILE

DUTIES

ID	01	FLIGHTLINE DUTIES
ID	02	SHOP DUTIES
ID	03	AVIONICS SUPPORT EQUIPMENT REPAIR DUTIES
ID	04	FLIGHTLINE SUPPORT EQUIPMENT DUTIES
ID	05	GENERAL NON-TECHNICAL DUTIES
ID	06	GENERAL TECHNICAL DUTIES

TASKS

		01	02	03	04	05	06	(APPLICABLE DUTIES)
ID	01	x	x	x	x			IDENTIFY NECESSARY MAINTENANCE AIDS
ID	02	x	x	x	x			OBTAIN AND RETURN MAINTENANCE AIDS
ID	03	x	x	x				GAIN/CLOSE ACCESS TO EQUIPMENT
ID	04	x	x	x				CONNECT/DISCONNECT TEST EQUIPMENT
ID	05	x	x	x				VERIFY MALFUNCTION
ID	06	x	x	x				EVALUATE DISC. REPORT/CHECK PREV. HIST.
ID	07	x	x	x				ISOLATE MALFUNCTION/LOCATE FAULT
ID	08	x	x	x				DETERMINE ACTION TO BE TAKEN
ID	09	x	x	x				PERFORM REPAIR MAINTENANCE
ID	10	x	x	x				SERVICE(LUBRICATE,CLEAN,PRESSURIZE,ETC.)
ID	11	x	x	x				CALIBRATE/ALIGN
ID	12	x	x	x				ADJUST
ID	13	x	x					OBTAIN/RETURN REPLACEMENT UNIT (LRU,SRU)
ID	14	x						REMOVE/REPLACE SAFETY WIRES/BOND. STRAPS
ID	15	x	x					DISCONNECT/REMOVE/INSTALL/CONNECT
ID	16	x	x	x	x			RECORD MAINTENANCE ACTIONS/RESULTS
ID	17		x					TEST OPERATION OF HENCH CHECK EQUIP/T.S.
ID	18		x	x				VERIFY REP. OF MALFUNCTION/MAINT.PROC.
ID	19		x					INITIATE DISPOSITION OF EQUIPMENT
ID	20				x			OPERATE SUPPORT EQUIPMENT
ID	50					x		OBSERVE SAFETY PRECAUTIONS AND REQUIREM.
ID	51					x		OBSERVE SECURITY PRECAUTIONS AND REQUIR.
ID	52					x		USE DATA DOCUMENTATION SYSTEMS
ID	53					x		UNDERSTAND/USE SUPPLY SYSTEM
ID	54					x		DEMO.FAMIL.WITH O.MAINT. POL/PROC/CONC.
ID	55					x		DEMONSTRATE FAMILIARTY WITH AIRCRAFT
ID	56					x		KNOW RELATED TRAINING
ID	60						x	KNOW AND USE GENERAL TEST EQUIPMENT
ID	61						x	KNOW BASIC PRINCIPLES OF ELECTRONICS
ID	62						x	UNDERSTAND & USE TROOBLESHOOTING TECHNI.
ID	63						x	DEMO. REPAIR&MAINT. TECHNIQUES/PROCEDUR.
ID	64						x	UNDERSTANDING SUBSYSTEM INTERFACING
ID	65						x	KNOW & USE TECHNICAL PUBLICATIONS
ID	66						x	DEMO.KNOWLEDGE OF SPEC.SUBSYSTEMS,LRUS
ID	67						x	DEMO.KNOWLEDGE OF SPEC.SUBSYSTEMS,LRUS
ID	68						x	DEMO.KNOWLEDGE OF SPEC.SUBSYSTEMS,LRUS
ID	69						x	DEMO.KNOWLEDGE OF SPEC.SUBSYSTEMS,LRUS

Table 3 (continued)

```
ID      70      x DEMO.KNOWLEDGE OF SPEC.SUBSYSTEMS,LRUS
ID      71      x DEMO.KNOWLEDGE OF SPEC.SUBSYSTEMS,LRUS
ID      72      x DEMO.KNOWLEDGE OF SPEC.SUBSYSTEMS,LRUS
ID      73      x DEMO.KNOWLEDGE OF SPEC.SUBSYSTEMS,LRUS
ID      74      x DEMO.KNOWLEDGE OF SPEC.SUBSYSTEMS,LRUS
ID      75      x DEMO.KNOWLEDGE OF SPEC.SUBSYSTEMS,LRUS
ID      76      x DEMO.KNOWLEDGE OF SPEC.SUBSYSTEMS,LRUS
ID      77      x DEMO.KNOWLEDGE OF SPEC.SUBSYSTEMS,LRUS
ID      78      x DEMO.KNOWLEDGE OF SPEC.SUBSYSTEMS,LRUS
ID      79      x DEMO.KNOWLEDGE OF SPEC.SUBSYSTEMS,LRUS
ID      80      x DEMO.KNOWLEDGE OF SPEC. AUTO. T.S.
ID      81      x DEMO.KNOWLEDGE OF SPEC. AUTO. T.S.
ID      82      x DEMO.KNOWLEDGE OF SPEC. AUTO. T.S.
ID      83      x DEMO.KNOWLEDGE OF SPEC. AUTO. T.S.
ID      84      x DEMO.KNOWLEDGE OF SPEC. AUTO. T.S.
ID      85      x DEMO.KNOWLEDGE OF SPEC. AUTO. T.S.
ID      86      x DEMO.KNOWLEDGE OF SPEC. AUTO. T.S.
ID      87      x DEMO.KNOWLEDGE OF SPEC. AUTO. T.S.
ID      88      x DEMO.KNOWLEDGE OF SPEC. AUTO. T.S.
ID      89      x DEMO.KNOWLEDGE OF SPEC. AUTO. T.S.
ID      90      x DEMO.KNOW. OF SPEC. MANUAL TEST STATIONS
ID      91      x DEMO.KNOW. OF SPEC. MANUAL TEST STATIONS
ID      92      x DEMO.KNOW. OF SPEC. MANUAL TEST STATIONS
ID      93      x DEMO.KNOW. OF SPEC. MANUAL TEST STATIONS
ID      94      x DEMO.KNOW. OF SPEC. MANUAL TEST STATIONS
ID      95      x DEMO.KNOW. OF SPEC. MANUAL TEST STATIONS
ID      96      x DEMO.KNOW. OF SPEC. MANUAL TEST STATIONS
ID      97      x DEMO.KNOW. OF SPEC. MANUAL TEST STATIONS
ID      98      x DEMO.KNOW. OF SPEC. MANUAL TEST STATIONS
ID      99      x DEMO.KNOW. OF SPEC. MANUAL TEST STATIONS
```

Table 4

DATA CARD COLUMN DESCRIPTION FOR TRAINING DATA ELEMENTS

Column	Title	Length	Type[a]	Justification[b]
1-7	EQUIPMENT ID NO.			
1	Weapon System	1	A	F
2	Major Aircraft System	1	A	F
3	Functional Group	1	A	F
4	Organizational Function	1	N	F
5	Subsystem	1	N	F
6	Line Replaceable Unit	1	N	F
7	Shop Replaceable Unit	1	N	F
8-12	JOB ID NO.			
8-9	Duty	2	N	F
10-11	Task	2	N	F
12	*Subtask*	1	N	F
13-17	Criticality	5	N	R
18	Decimal	1	X	F
19-23	Difficulty	5	N	R
24	Decimal	1	X	F
25-29	Frequency	5	N	R
30	Decimal	1	X	F
31-35	*Psychomotor Level*	5	N	R
36	Decimal	1	X	F
37-41	Cognitive Level	5	N	R
42	Decimal	1	X	F
43-44	F Flag	2	N	R
45-46	Method	2	N	R
47-48	Medium	2	N	R
49	Blank	1	X	F
50-55	TTS Cost (col. 53 is decimal)	6	N	R
56	Blank	1	X	F
57-62	OJT Cost (col. 60 is decimal)	6	N	R
63	Blank	1	X	F
64-67	TTS Time	4	N	R
68	Blank	1	X	F
69-72	OJT Time	4	N	R
73-80	Blank	8	X	F

[a]Type – A = alpha, N = numeric, X = either.

[b]Justification – R = right, L = left, F = fixed.

Table 5

INTERACTIVE INPUTS

1. Threshold cutoff levels for the task characteristic parameter values[a]
   - (a) Criticality
   - (b) Learning difficulty
   - (c) Frequency
   - (d) Psychomotor level
   - (e) Cognitive level
2. Task selection algorithm choice[a]
3. Number of trainees to be trained in each Air Force Specialty Code[a]
4. Regression coefficients for derivation of cost and time data for each task block
5. Training time constraint[a]
6. Training cost constraint
7. TTS/OJT split
8. Alternative training objective mapping
9. Alternative methods and medium for training
10. Scarce TTS resource to be optimally scheduled
11. Minimum class size[a]
12. Maximum class size[a]

---

[a]required input

Table 6

INTERACTIVE OUTPUTS

1. Histogram of complete task input data
2. Histogram of tasks to be trained
3. Histogram of tasks not be be trained
4. Total times and costs for TTS and OJT training
*5. Student split between TTS and OJT training (optional output)
*6. Total times and costs for feasible combinations of TTS/OJT training (optional output)
7. Training plan
8. Training program (schedule)

---

*These two optional outputs are provided to accommodate the possibility that personnel may receive a combination of TTS and OJT instruction. While not done under present Air Force policy, such dual instruction is an option which might be desirable in the future.

III. DETAILED MODEL DESCRIPTION

3.1 TASK BLOCK GENERATOR

The interactive nature of TRAMOD affords the user great flexibility in directing its operation. Its first analytical component consists of a set of five screening algorithms which test the task characteristic parameter values of the input data against the user's criterion, to select those tasks to be trained (Figure 4). The user is first prompted to select threshold values for each of the task characteristics, and then one of the five algorithms which will govern the screening process. The threshold value for each parameter is then registered as the lowest value which TRAMOD is to use as a criterion, in any of five specific tests, to determine whether a task should be trained. An initial histogram of the input task data set is immediately generated so that the user may scan the characteristics of the task population being tested as an aid in selecting appropriate thresholds. The various screening algorithms compare these thresholds (N_i) with the actual values associated with each task under examination (C_i). Each algorithm performs a unique test representing a different degree of restrictiveness:

"All" Decision Tree - all the characteristics must meet their individual thresholds

e.g. $C_i \geq N_i$ for all i, i = 1, ., 5

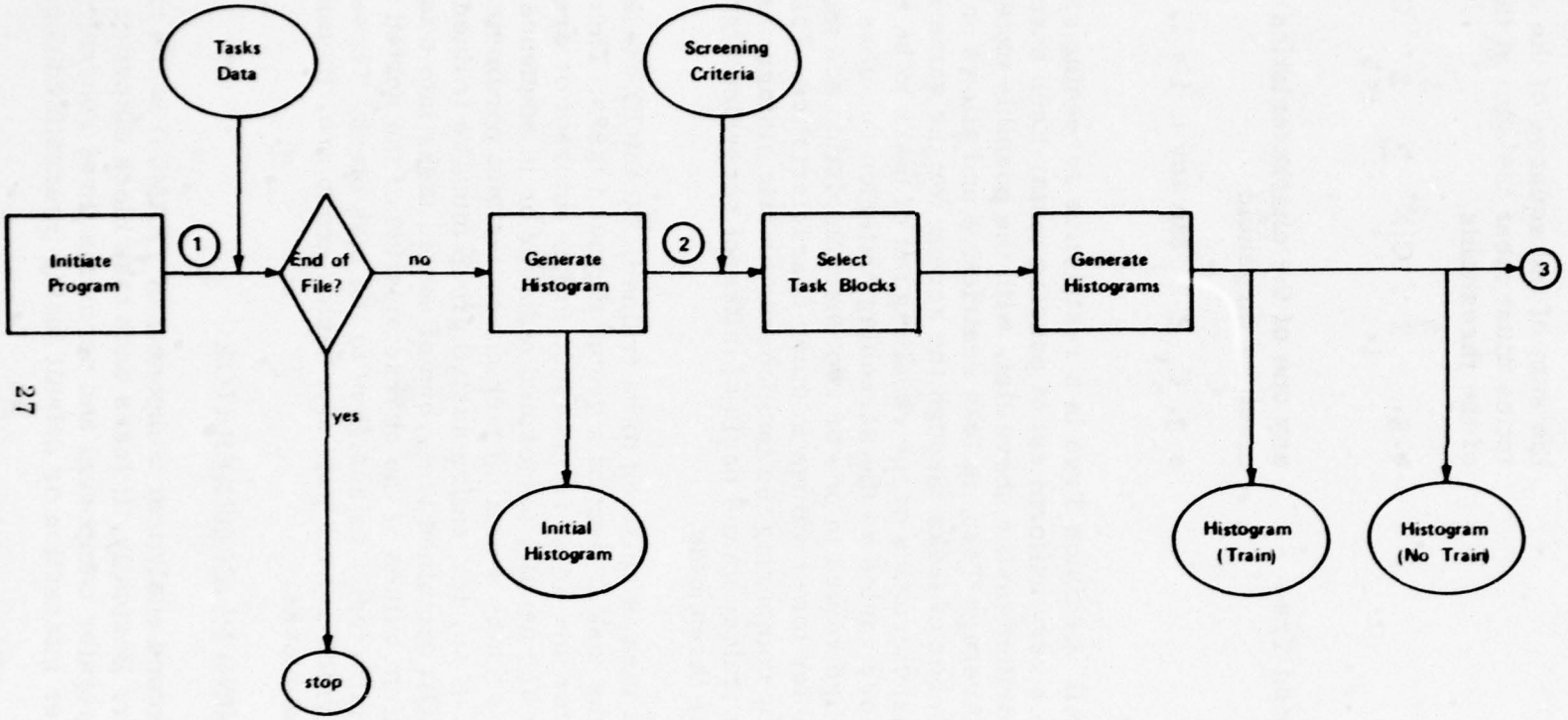
Pure Average Test - the average of the characteristics must meet the average of the thresholds

e.g. $$\sum_{i=1}^{5} C_i \geq \sum_{i=1}^{5} N_i$$

Weighted Average Test - with thresholds normalized and used as weighting on the characteristics, the weighted average must meet the average of the thresholds

e.g. $$\sum_{i=1}^{5} C_i \bullet W_i \geq \frac{1}{5} \bullet \sum_{i=1}^{5} N_i$$

where $$W_i = \frac{6-N_i}{\sum_{i=1}^{5} (6-N_i)} \quad \text{for } i = 1, ., 5$$



GENERATE SELECTED TASK BLOCKS

Figure 4

RMS Test - the sum of the squares of the characteristics must meet the sum of the squares of the thresholds

e.g. $\sum_{i=1}^{5} (C_i)^2 \geq \sum_{i=1}^{5} (N_i)^2$

"Any" Decision Tree - any one of the characteristics must meet its threshold

e.g. $C_i \geq N_i$ for any i, i = 1, ., 5

The "All" Decision Tree is a restrictive screening criterion and results in a very uniform set of passing tests. Each succeeding screening algorithm on the above list, with the possible exception of the Weighted Average Test, is less restrictive and allows an increasing number of tasks through the screen for the same set of N_i's. The characteristics of the resulting set of tasks to be trained become far more varied as the screening criterion becomes less restrictive. High values in one or two characteristics are more able to compensate for lower values in other characteristics. This can be readily seen by comparing the task characteristic histograms of tasks selected to be trained on the basis of different screening algorithms using the same thresholds.

Once a task is selected to be trained, TRAMOD checks to determine if the task is part of a group of nested tasks. There are tasks which, for one or more reasons, either interact or are related such that they are usually performed together or in sequence, and are designated as such by a nesting task characteristic parameter value in the data bank. If so, the entire nested group must be trained as a whole. TRAMOD combines the group of nested tasks into a task block and the maximum values of the characteristics of the nested tasks become the characteristics assigned to the task block. Those tasks to be trained which are not part of a nested group are, themselves, treated as task blocks.

3.2 TRAINING PLAN GENERATOR

The second analytical component of TRAMOD is the training plan generator. Basically, it takes each task block outputted by the task block generator component and performs three general functions in either a user interaction or default (using internalized constant

parameter values) mode. These functions are: (1) the assessment of training time and cost; (2) the assignment of a training mode; and (3) the assignment of training methods and media.

3.2.1 Assessment of Training Times and Costs

Once the training blocks have been determined through the user's selection criteria, TRAMOD prints a series of questions to which the user must respond to indicate how the training times and costs for the task blocks are to be derived (Figure 5). Three options are available to the user:

- Direct input of training times and costs for each task block in the input data set
- Calculation of training times and costs with user-selected regression coefficients; times and costs being linear combinations of the task characteristics, e.g.,

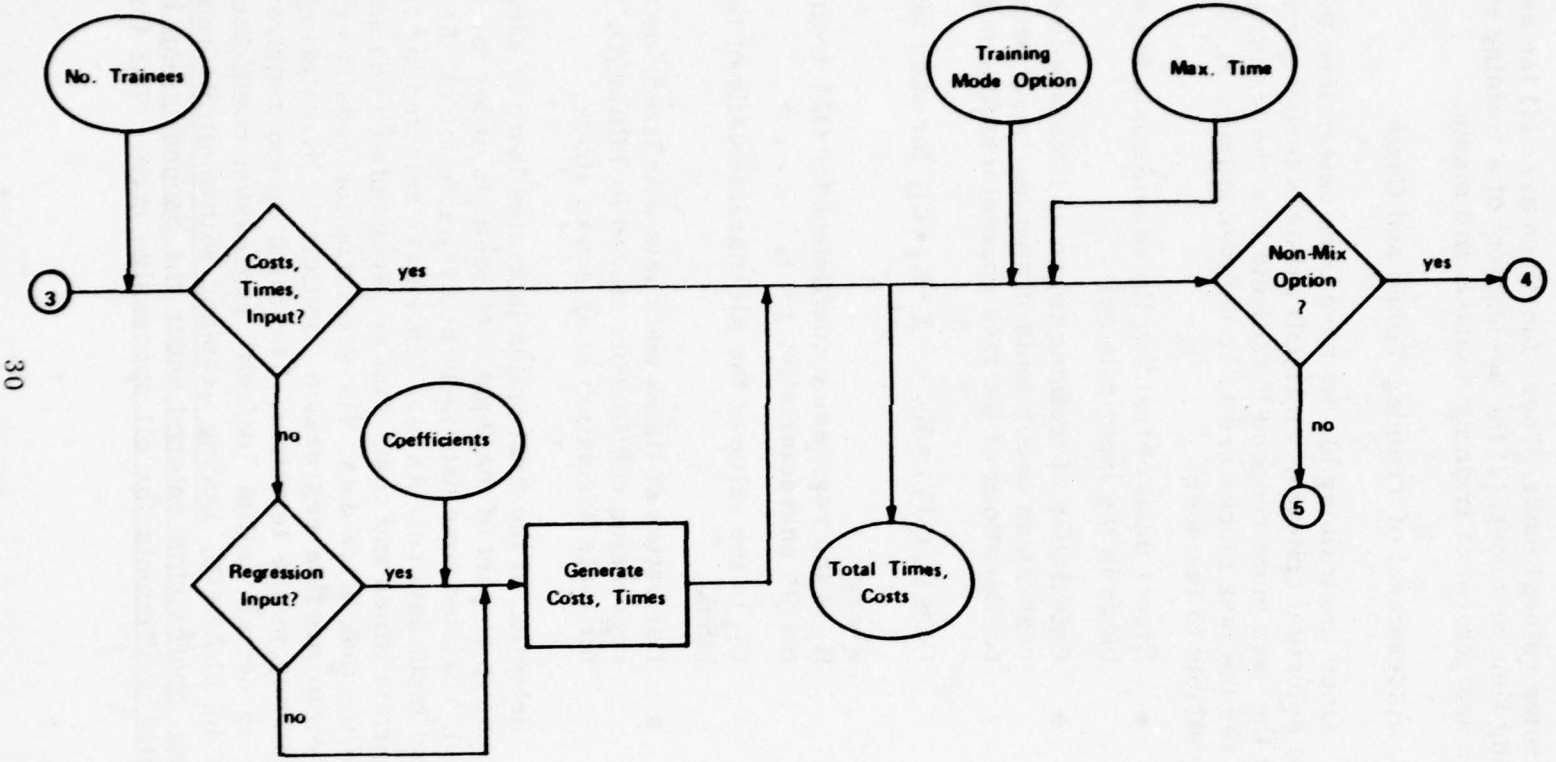
  $$\text{Cost}(i, \text{OJT}) = K_0 + \sum_{j=1}^{5} K_j \bullet C_{ij} \quad \text{for each task block } i$$

  where

  K_j is the regression coefficient for OJT training costs for the j^{th} characteristic, $j = 1, \ldots, 5$

  C_{ij} is the value of the j^{th} characteristic of the i^{th} task block
- Derivation of times and costs with fixed "default" regression coefficients stored in TRAMOD. The equation for their derivation is as shown above.

Selection of the first option indicates that the data have already been read in as part of the input data base. In order for this option to be used, cost and time data must be available for all the tasks in the original input data set. As task blocks are selected for training, the appropriate times and costs data are accumulated and stored with the rest of the task block data. The second option requires (and prompts) the user to provide regression coefficients by entering them interactively at the terminal. The third option requires no data input, as certain initial "default" regression coefficients are stored in TRAMOD. During sessions with multiple runs, regression coefficients entered under the second option become the "default" coefficients for all successive runs. The initial



OBTAIN TRAINING TIMES AND COSTS

Figure 5

TABLE 7
DEFAULT REGRESSION COEFFICIENTS

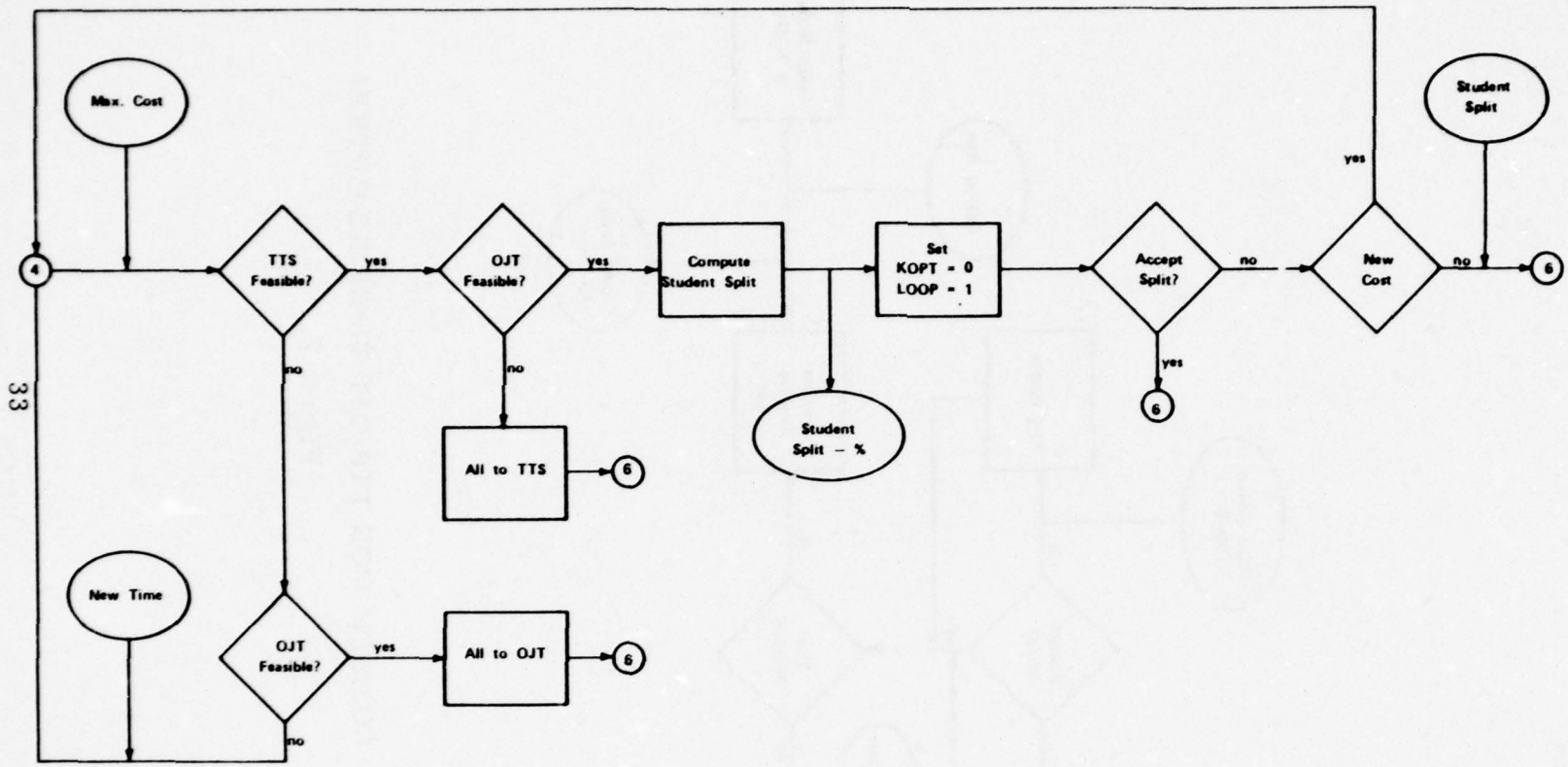
	TTS TIME	OJT TIME	TTS COST	OJT COST
CONSTANT (K_o)	617.4578	-7.725949	15.0	4.36
CRITICALITY (K_j)	- .2408859	1.137375	0	0
DIFFICULTY (K_j)	- 1.130284	6.137375	0	0
FREQUENCY (K_j)	- 124.5503	-1.487537	0	0
PSYCHOMOTOR (K_j)	- 2.232280	2.534994	0	0
COGNITIVE (K_j)	14.86655	1.387448	0	0

the times and costs data arrays needed for the training mode selection process in which these values are summed over all task blocks to be trained and outputs generated indicating the total times and student costs required for both TTS and OJT training.

3.2.2 Assignment of Training Mode

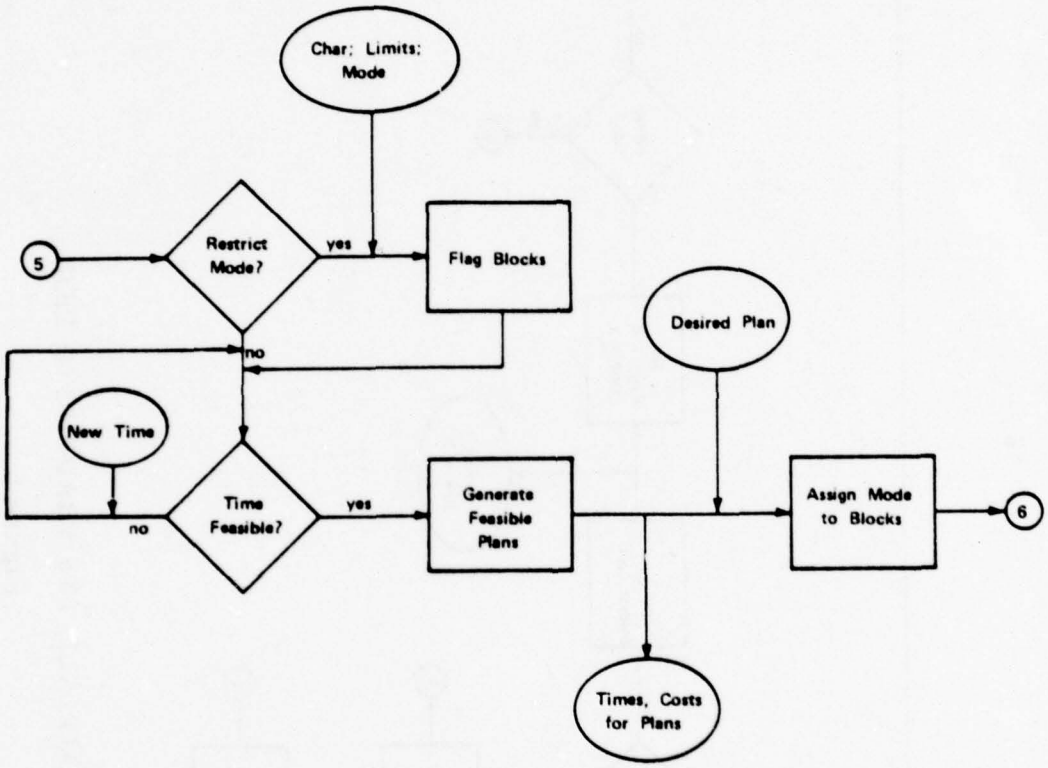
TRAMOD will determine assignments for training, to TTS or OJT, in either of two ways, i.e., on a "mixed" or "non-mixed" basis, at the discretion of the user. A "non-mix" option assumes each student goes to either school or OJT, but not both (Figure 6). The user inputs time and per student cost restrictions for the entire training program, and, if both TTS and OJT training requirements are feasible. TRAMOD calculates the percentage split among the students between the two training modes. If the user does not input a maximum time which allows both modes of training, and a maximum student cost large enough to include one of the modes, a percentage split will not be calculated. Instead, both OJT and TTS training requirements will be singly tested against the user's time and cost restrictions to determine if either, alone, is a feasible mode. If so, TRAMOD will assign all the students to that training mode. Otherwise, the user is prompted to input revised maximums for training time and/or cost which will allow training through at least one of the two modes. If TRAMOD splits the students between school and OJT, the remainder of the training plan generator and the training program generator module are executed twice; once for the all-TTS training program student group and then again for the student group to receive OJT training. In the latter case, certain non-applicable options are ignored, such as multiple schedule arrangements for scarce resource optimization.

The second manner in which TRAMOD may assign training modes is the "mixed" option (Figure 7). If the user selects this option, the assignments will combine both school and OJT training such that each student receives a mix of school and OJT training. The user inputs a maximum time for the training program, and task blocks are distributed between TTS and OJT training in a way which results in a minimum cost training plan. Distribution decisions are made by a dynamic programming algorithm on the sole basis of times and costs data. However, the user has the option of overriding individual decisions and forcing tasks to be assigned to the user's choice of TTS or OJT training. After the algorithm is completed, TRAMOD prints all feasible times and costs for training which meet



ALL OJT/ALL TTS TRAINING OPTION

Figure 6



COMBINATION TTS/OJT TRAINING OPTION

Figure 7

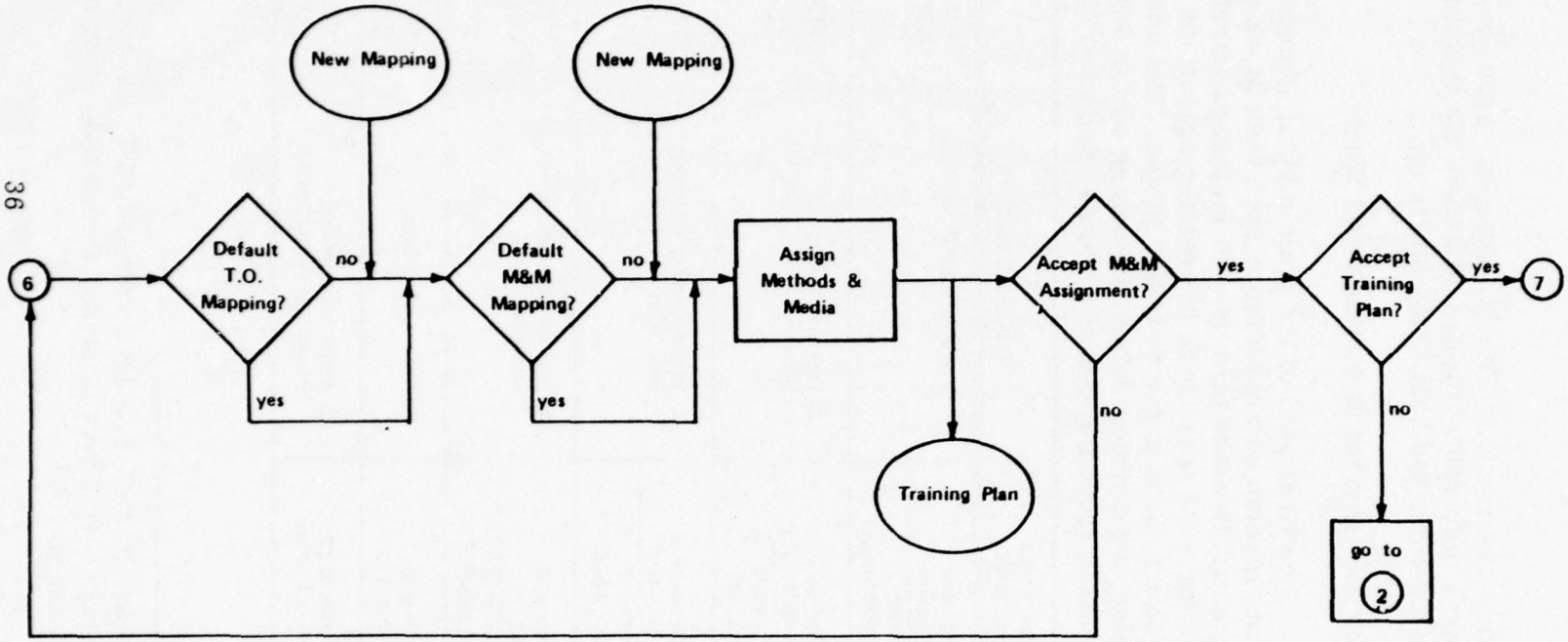
the time restriction. The user can compare the costs associated with different time restrictions and select the training mode mix based solely on time and cost considerations.

3.2.3 Assignment of Methods and Media

The final part of the training plan generator module assigns a training method and medium to each task block (Figure 8). The assignment is made through two mappings: first, from the taxonomic levels for each task block to training objectives (T.O.'s); second, from the T.O.'s to methods and media. The user may input his own mapping; otherwise, default mappings within TRAMOD are used. These were taken from Parker and Downs[1] and are illustrated below.

Taxonomic Description	Training Objective	Method/Media
Psychomotor 1 (Imitation) Cognitive 1 (Comprehensive)	1. Learning Identifications	TTS: Discussion/Transparencies OJT: Informal Lecture/ Transparencies
Psychomotor 2 (Manipulation) Psychomotor 3 (Precision)	2. Learning Perceptual Discrimination	TTS: Simulation/Training Film OJT: Demonstration/Training Film
Psychomotor 4 (Articulation) Cognitive 2 (Application)	3. Understanding Principles and Relationships	TTS: Simulation/Simulator OJT: Performance/Mock-Ups
Cognitive 3 (Analysis) Cognitive 4 (Synthesis)	4. Learning Procedural Sequence	TTS: Performance/Simulator OJT: Performance/Training Film
Cognitive 5 (Evaluation)	5. Making Decisions	TTS: Simulation/Simulator OJT: Performance/Training Film
Psychomotor 5 (Naturalization)	6. Performing Skilled Perceptual Motor Arts	TTS: Performance/Simulator OJT: Performance/On-Equipment

[1] Parker, James F., Jr., and Judith E. Downs, "Selection of Training Media," Air Force Systems Command, Wright-Patterson AFB, Ohio, September 1961.



ASSIGN METHODS AND MEDIA

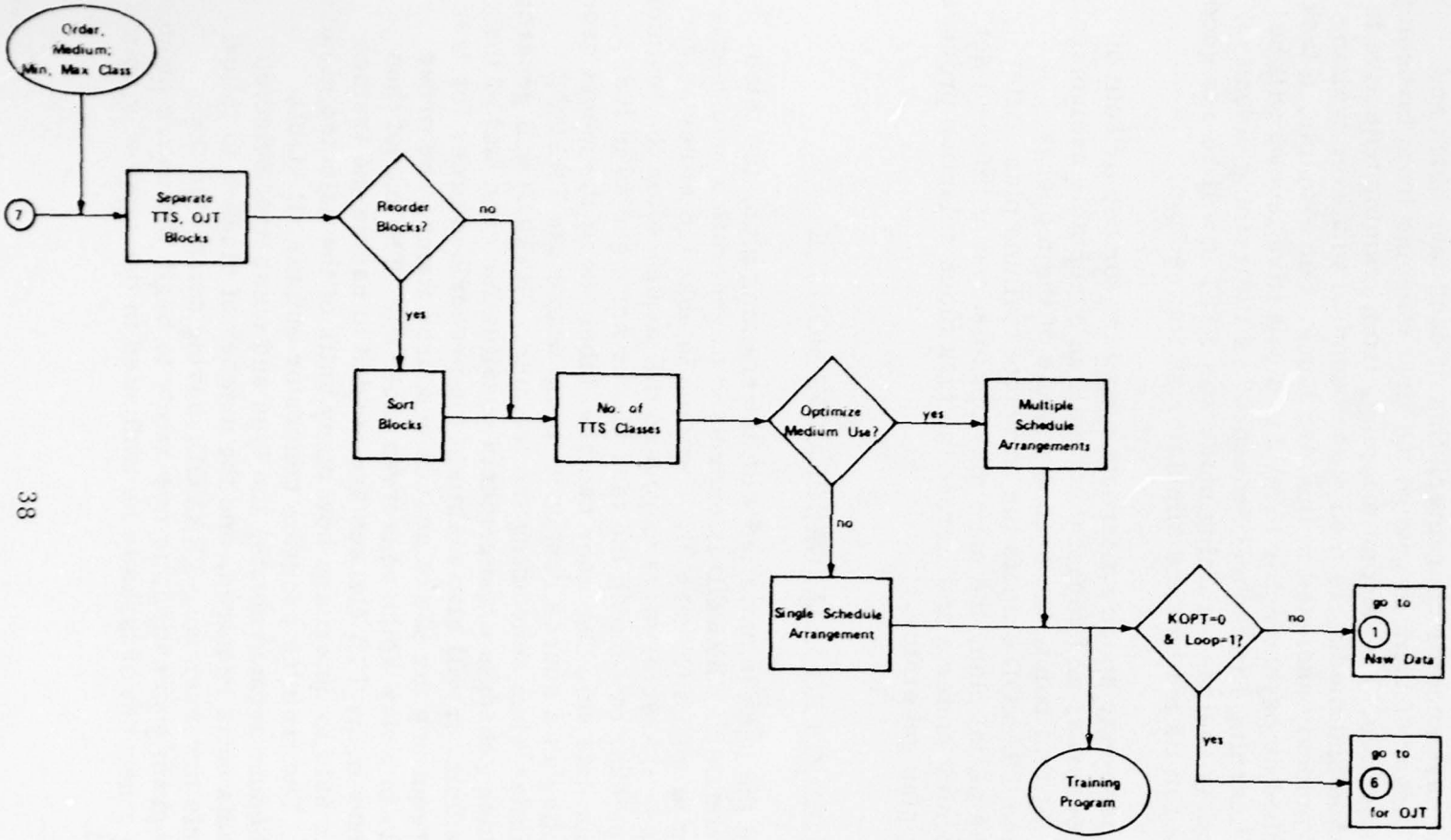
Figure 8

The model examines the parameter values associated with the cognitive and psychomotor characteristics of each task block and chooses the larger of them to govern the first mapping from taxonomy to training objective. The second mapping, from training objective to training method and media, is then made based on whatever training mode was previously assigned to the task block. For example, a task block described as psychomotor level 2 and cognitive level 3 will be mapped to "Learning Procedural Sequence" as its training objective. If the previously assigned training mode was TTS, it will be assigned performance for its method and simulator as its medium.

The user may specify alternate mappings for one or both of the above processes, and repeat them until an acceptable assignment is obtained. At this point, the second module of the model is completed and TRAMOD outputs the complete training plan. After reviewing the initial plan, the user may opt to select a different set of task screening criteria and iterate the task block selection process and training plan generator.

3.3 TRAINING PROGRAM GENERATOR MODULE

Once the user is satisfied with the training plan, the third analytical module of TRAMOD is exercised to generate a representative training program (Figure 9). The user is asked to either select one of the task characteristics to govern the sequence for the training of the task blocks, or to keep the task blocks arranged as in the original input data set. The user can also label one of the media used in TTS training as a scarce resource item, whose use TRAMOD should minimize when scheduling the training. TRAMOD will generate all the training schedule arrangements possible for each unit of the selected medium. It will also maintain the prescribed order for task blocks not requiring the use of scarce resource items, and order those that do in a way which conserves their use. The model then calculates how many TTS classes are needed to handle the trainee group and is able to determine how many units of the selected medium are needed. The training program generator outputs all viable training schedule arrangements, the type and number of selected training media units required, and the number of classes to follow each schedule arrangement. TRAMOD, having completed the training program generation, is now ready to begin the entire process again using a new set of tasks to be evaluated in the same way as the original set.



GENERATE TRAINING SCHEDULE

Figure 9

REFERENCES

1. Czuchry, A. J., K. M. Doyle, J. T. Frueh, H. A. Baran, and D. L. Dieterly. Digital avionics information system (DAIS): Training model technical report. AFHRL-TR-78-58(I), Wright-Patterson AFB, OH: Advanced Systems Division, Air Force Human Resources Laboratory, December 1978.

2. Parker, Jr., J. F. and J. E. Downs. Selection of training media. ASD-TR-61-473, Psychological Research Associates, September 1961.

3. Czuchry, A. J., J. M. Glasier, R. H. Kistler, M. A. Bristol, H. A. Baran, and D. L. Dieterly. Digital avionics information system (DAIS): Reliability and maintainability model. AFHRL-TR-78-2(I), Wright-Patterson AFB, OH: Advanced Systems Division, Air Force Human Resources Laboratory, April 1978. AD-A056 530.

4. Czuchry, A. J., J. M. Glasier, R. H. Kistler, M. A. Bristol, H. A. Baran, and D. L. Dieterly. Digital avionics information system (DAIS): Reliability and maintainability model, volume 2: User's guide. AFHRL-TR-78-2(II), Wright-Patterson AFB, OH: Advanced Systems Division, Air Force Human Resources Laboratory, September 1978.

APPENDIX A

SAMPLE TRAINING MODEL
COMPUTER TERMINAL SESSION
(with User Interaction shown)

```
ENTER AFSC-   423X5

  4 TASKS BEFORE SCREENING

RESPOND Y (DISPLAY) OR N (INHIBIT) TO HISTOGRAM TITLES

** HISTOGRAM OF COMPLETE TASK DATA FOR 423X5  **   Y                    <--

     I                                        5
     I                                        5
     I    1              2                    5      2          1 2
     I    1   3   5     1 2 3                5    1 2 3        1 2
    0I-------------------------------------------------------------------
       CRITICALITY   DIFFICULTY    FREQUENCY   PSYCHOMOTOR   COGNITIVE


ENTER CUTOFF LEVEL (0 THRU 5) FOR:
CRITICALITY -  1                                                    <--

DIFFICULTY  -  2                                                    <--

FREQUENCY   -  5                                                    <--

PSYCHOMOTOR -  1                                                    <--

COGNITIVE   -  1                                                    <--


ENTER SCREENING OPTION
1='ANY' DECISION TREE
2='ALL' DECISION TREE
3=RMS TEST
4=WEIGHTED AVERAGE
5=PURE AVERAGE
=    2                                                              <--

DO YOU WANT A LISTING OF REJECTED TASKS?   Y                        <--

 KEY EQUIP   TASK   NAME                                        C D F P C
   2  AGEFL  04020  OBTAIN AND RETURN MAINTENANCE AIDS          1 1 5 2 1

** HISTOGRAM OF TASKS TO BE TRAINED FOR 423X5  **   Y                 <--

     I                                        5
     I                   2                    5                    2
     I    1   3   5      2 3                  5    1 2 3         1 2
    0I-------------------------------------------------------------------
       CRITICALITY   DIFFICULTY    FREQUENCY   PSYCHOMOTOR   COGNITIVE


** HISTOGRAM OF REJECTED TASKS FOR 423X5   **   Y                    <--

     I    1              1                    5      2          1
    0I-------------------------------------------------------------------
       CRITICALITY   DIFFICULTY    FREQUENCY   PSYCHOMOTOR   COGNITIVE
```

```
  3 TASKS AFTER SCREENING

DO YOU WANT TO SCREEN THE TASKS OVER AGAIN?   N                    <--

ENTER NUMBER OF TRAINEES IN GROUP 423X5   -   26                   <--

DO YOU WISH TO USE REGRESSION COEFFICIENTS FOR:
TTS COSTS?    N                                                    <--

OJT COSTS?    N                                                    <--

TTS TIMES?    N                                                    <--

OJT TIMES?    N                                                    <--

TIME TO TRAIN IS SCHOOL:    4.8 DAYS TIME TO TRAIN IN OJT:    4.8 DAYS
COST TO TRAIN IN SCHOOL:      570.0 COST TO TRAIN IN OJT:       165.7

INPUT TRAINING MODE OPTION:
1=NON-MIX
2=SCHOOL/OJT TRAINING MIX
=   1                                                              <--

ENTER TIME RESTRICTION FOR TRAINING PROGRAM-    5                  <--

ENTER MAXIMUM TOTAL COST FOR TRAINING PROGRAM-    10000            <--

A TRAINING PROGRAM FOR   26 STUDENTS, MAXIMUM TOTAL COST OF 10000.:
              14 STUDENTS TO SCHOOL              12 STUDENTS TO OJT
PERCENT SPLIT:  53.8% SCHOOL,  46.2% OJT

YOU CAN ACCEPT THE CURRENT SPLIT, INPUT A NEW VALUE FOR COST LIMIT
AND RECOMPUTE THE SPLIT, OR INPUT THE SPLIT DIRECTLY.
KEEP THE ABOVE SPLIT?    Y                                         <--

** TRAINING PROGRAM FOR TTS STUDENTS **

DO YOU WANT TO ENTER AN ALTERNATIVE MAPPING FROM TAXONOMY
TO TRAINING OBJECTIVE?   Y                                         <--

ENTER THE NUMBER OF THE TRAINING OBJECTIVE ASSOCIATED WITH EACH OF THE
FOLLOWING TAXONOMIC LEVELS, WHERE:
T.O. 1 = LEARNING IDENTIFICATIONS
T.O. 2 = LEARNING PERCEPTUAL DISCRIMINATIONS
T.O. 3 = UNDERSTANDING PRINCIPLES AND RELATIONSHIPS
T.O. 4 = LEARNING PROCEDURAL SEQUENCE
T.O. 5 = MAKING DECISIONS
T.O. 6 = PERFORMING SKILLED PERCEPTUAL MOTOR ACTS

PSYCHOMOTOR =1; T.O.=   3                                          <--

PSYCHOMOTOR =2; T.O.=   2                                          <--

PSYCHOMOTOR =3; T.O.=   4                                          <--
```

```
PSYCHOMOTOR =4; T.O.=   4
PSYCHOMOTOR =5; T.O.=   6
COGNITIVE   =1; T.O.=   1
COGNITIVE   =2; T.O.=   4
COGNITIVE   =3; T.O.=   3
COGNITIVE   =4; T.O.=   5
COGNITIVE   =5; T.O.=   5

DO YOU WISH TO ENTER AN ALTERNATIVE MAPPING
FROM TRAINING OBJECTIVE TO METHODS AND MEDIA?   Y

ENTER THE NUMBERS OF THE TRAINING METHOD AND TRAINING MEDIUM
ASSOCIATED WITH EACH OF THE FOLLOWING TRAINING OBJECTIVES, WHERE:
METHOD 1 = INFORMAL LECTURE          MEDIUM 1 = TRANSPARENCIES
METHOD 2 = DISCUSSION                MEDIUM 2 = TRAINING FILMS
METHOD 3 = DEMONSTRATION             MEDIUM 3 = MOCK-UPS
METHOD 4 = SIMULATION                MEDIUM 4 = SIMULATOR
METHOD 5 = PERFORMANCE               MEDIUM 5 = ACTUAL EQUIPMENT
METHOD 6 = OTHER (LEFT BLANK)        MEDIUM 6 = OTHER (LEFT BLANK)

TRAINING OBJECTIVE=1; METHOD FOR TTS =   2
                      MEDIUM FOR TTS =   1
TRAINING OBJECTIVE=1; METHOD FOR OJT =   1
                      MEDIUM FOR OJT =   3
TRAINING OBJECTIVE=2; METHOD FOR TTS =   1
                      MEDIUM FOR TTS =   3
TRAINING OBJECTIVE=2; METHOD FOR OJT =   3
                      MEDIUM FOR OJT =   2
TRAINING OBJECTIVE=3; METHOD FOR TTS =   2
                      MEDIUM FOR TTS =   2
TRAINING OBJECTIVE=3; METHOD FOR OJT =   3
                      MEDIUM FOR OJT =   3
TRAINING OBJECTIVE=4; METHOD FOR TTS =   2
                      MEDIUM FOR TTS =   1
TRAINING OBJECTIVE=4; METHOD FOR OJT =   4
                      MEDIUM FOR OJT =   4
```

```
TRAINING OBJECTIVE=5; METHOD FOR TTS =   2
                      MEDIUM FOR TTS =   2
TRAINING OBJECTIVE=5; METHOD FOR OJT =   5
                      MEDIUM FOR OJT =   4
TRAINING OBJECTIVE=6; METHOD FOR TTS =   5
                      MEDIUM FOR TTS =   4
TRAINING OBJECTIVE=6; METHOD FOR OJT =   5
                      MEDIUM FOR OJT =   5
```

423X5 - TRAINING PLAN GENERATOR OUTPUT

I SUBSYSTEM- AGEFL	MODE	DAYS METH	MED I
I 04 FLIGHTLINE SUPPORT EQUIPMENT DUTIES			I
I 010 IDENTIFY NECESSARY MAINTENANCE AIDS	TTS	.3 DSCUS	TRANSI
I 160 RECORD MAINTENANCE ACTIONS/RESULTS	TTS	1.5 DSCUS	TRANSI
I 200 OPERATE SUPPORT EQUIPMENT	TTS	3.0 DSCUS	TRANSI

```
DO YOU WISH TO GENERATE ANOTHER METHODS AND MEDIA ASSIGNMENT?   N

DO YOU WISH TO REGENERATE THE TRAINING PLAN?   N

ENTER NUMBER INDICATING CHARACTERISTIC TO  GOVERN ORDERING OF TASKS
FOR TRAINING.  THIS TASK HIERARCHY ORDERS FROM LOW TO HIGH
1=CRITICALITY
2=DIFFICULTY
3=FREQUENCY
4=PSYCHOMOTOR
5=COGNITIVE
0=KEEP INITIAL ORDERING OF TASKS
=   5

ENTER NUMBER INDICATING MEDIUM TO BE OPTIMALLY SCHEDULED
1=TRANSPARENCIES
2=TRAINING FILMS
3=MOCK-UPS
4=SIMULATORS
5=ACTUAL EQUIPMENT
0=IGNORE OPTION
=   4
```

```
ENTER MINIMUM CLASS SIZE FOR TTS-   2                                    <---

ENTER MAXIMUM CLASS SIZE FOR TTS-   5                                    <---


   3 TTS CLASSES ARE NEEDED FOR   14 STUDENTS

** SCHEDULE ARRANGEMENT  1 **

SUBSYS  TASK  MODE  DAYS   COST   METHOD      MEDIUM
 AGEFL  04010 TTS     .3   30.0 DISCUSSION XPARENCIES
 AGEFL  04160 TTS    1.5  180.0 DISCUSSION XPARENCIES
 AGEFL  04200 TTS    3.0  360.0 DISCUSSION XPARENCIES



** TRAINING PROGRAM FOR OJT STUDENTS **

DO YOU WANT TO ENTER AN ALTERNATIVE MAPPING FROM TAXONOMY
TO TRAINING OBJECTIVE?   N                                               <---


DO YOU WISH TO ENTER AN ALTERNATIVE MAPPING
FROM TRAINING OBJECTIVE TO METHODS AND MEDIA?   N                        <---



              423X5 - TRAINING PLAN GENERATOR OUTPUT
-------------------------------------------------------------------------
I SUBSYSTEM- AGEFL                                 MODE  DAYS METH   MED I
I-----------------------------------------------------------------------I
I 04  FLIGHTLINE SUPPORT EQUIPMENT DUTIES                                I
I    010  IDENTIFY NECESSARY MAINTENANCE AIDS      OJT    .3 LECTR MOCK I
I    160  RECORD MAINTENANCE ACTIONS/RESULTS       OJT   1.5 SIMUL SIMULI
I    200  OPERATE SUPPORT EQUIPMENT                OJT   3.0 SIMUL SIMULI
-------------------------------------------------------------------------


DO YOU WISH TO GENERATE ANOTHER METHODS AND MEDIA ASSIGNMENT?   N         <---

DO YOU WISH TO REGENERATE THE TRAINING PLAN?   N                          <---

ENTER NUMBER INDICATING CHARACTERISTIC TO  GOVERN ORDERING OF TASKS
FOR TRAINING.  THIS TASK HIERARCHY ORDERS FROM LOW TO HIGH
1=CRITICALITY
2=DIFFICULTY
3=FREQUENCY
4=PSYCHOMOTOR
5=COGNITIVE
0=KEEP INITIAL ORDERING OF TASKS
=   0                                                                    <---
```

```
** SCHEDULE ARRANGEMENT  1 **

SUBSYS   TASK   MODE   DAYS    COST    METHOD      MEDIUM
 AGEFL   04010  OJT      .3     8.7   LECTURE     MOCK-UPS
 AGEFL   04160  OJT     1.5    52.3  SIMULATION  SIMULATORS
 AGEFL   04200  OJT     3.0   104.6  SIMULATION  SIMULATORS



DO YOU WANT TO GENERATE A TRAINING PLAN FOR
ANOTHER AFSC?   N                                    ←

     STOP.
      1.848 CP SECONDS EXECUTION TIME
```

APPENDIX B

TRAINING MODEL PROGRAM LISTING

```
100=      PROGRAM TRAMOD  (INPUT,TAPE5=INPUT,OUTPUT,
110=     & TASKS,TAPE2=TASKS,HDB,TAPE10=HDB)
120=C
130=C   - - PROGRAM FOR TRAINING MODEL - -
140=C
150=      DIMENSION LRO(5,5),NRO(5,5),TVAL(2),CVAL(2),NUSE(10)
160=      DIMENSION CODE(2),MED(6),MET(6),CHS(5),FLS(3),JMAT(6,2)
170=C CHS AND FLS ARE TEMPORARY ARRAYS TO HOLD PORTIONS OF
180=C EACH INPUT RECORD, NAMELY THE CHARACTERISTICS AND FLAGS.
190=C
200=      COMMON/BLOC1/ NBLOC,TRPLAN(350),MAXES(350),JCOSTS(350),
210=     & JTIMES(350),INONE(10)
220=C NBLOC = NUMBER OF TASK BLOCKS TO BE TRAINED
230=C TRPLAN= SUBSYSTEM ID, SUCH AS 'AM110' OR 'TRAIN'. MAX 7 CHAR.
240=C MAXES = PACKED TO HOLD IN EACH DIGIT (FROM RIGHT TO LEFT):
250=C               1 = CRITICALITY CHARACTERISTIC
260=C               2 = DIFFICULTY CHARACTERISTIC
270=C               3 = FREQUENCY CHARACTERISTIC
280=C               4 = PSYCHOMOTOR CHARACTERISTIC
290=C               5 = COGNITIVE CHARACTERISTIC
300=C               6 = MODE
310=C               7 = METHOLD
320=C               8 = MEDIA
330=C               9-10 = DUTY NUMBER
340=C               11-13= TASK CODE
350=C JCOSTS= TRAINING COSTS. FIRST 8 DIGITS TTS, SECOND 8 OJT.
360=C JTIMES= TRAINING TIMES (HOURS). STORED SIMILARLY TO JCOSTS,
370=C         IN THOUSANDTHS, SUCH THAT 12345678 MEANS 12345.678
380=C INONE = TEMPORARY ARRAY TO HOLD THE 10 VALUES PACKED IN MAXES.
390=C
400=      COMMON/BLOC2/ MTHMAP(6,2),MEDMAP(6,2),MAPPSY(5),MAPCOG(5)
410=C MTHMAP= MAPPING FROM TRAINING OBJECTIVE I TO METHOD PER TRAINING MODE
420=C MEDMAP= MAPPING FROM TRAINING OBJECTIVE I TO MEDIA PER TRAINING MODE
430=C MAPPSY= MAPPING FOM PSYCHOMOTOR LEVL TO TRAINING OBJECTIVE
440=CMAPCOG= MAPPING FROM COGNITIVE LEVEL TO TRAININF OBJECTIVE
450=C
460=      COMMON/BLOC3/ NTR,NST(2),NHIER,NMEDIA,NSCH,MIN,MAX,NCL,NSIZE(30)
470=C SEE SUBROUTINE SCHEDULE FOR DESCRIPTIONS.
480=C
490=      COMMON/SHARE/ TASKS(350),KASTS(350),CS(350,2),TS(350,2)
510=C TASKS = SUBSYSTEM ID, SUCH AS 'AM110' OR 'LEARN' FOR INPUT DATA.
520=C KASTS = PACKED STORAGE OF INPUT DATA AS IN MAXES.
530=C         SET NEGATIVE IF TASK IS TO BE FORCED.
540=C CS    = TTS AND OJT COSTS
550=C TS    = TTS AND OJT TIMES
590=C
595=      COMMON/LABELS/LABEL(2,6)
600=      COMMON/MINT/CODE
610=C CODE = LITERAL TTS OR OJT
630=C
640=      COMMON/NAMES/ DUT(4,6),TSK(4,62),KDUT(6),KTSK(62),NDUT,NTSK
650=C DUT   = DUTY NAMES, 40 CHARACTERS LONG
660=C TSK   = TASK NAMES, 40 CHARACTER LONG
670=C KDUT  = DUTY NUMBER, 01 TO 99, STORED LITERAL
680=C KTSK  = TASK NUMBER, 000 TO 99 STORED LITERAL
690=C NDUT  = NUMBER OF DUTIES
700=C NTSK  = NUMBER OF TASKS
710=C
720=      DIMENSION NSCHED(350,4)
730=      EQUIVALENCE (TASKS(1),NSCHED(1,1))
```

```
C NSCHED (DESCRIBED IN SUBROUTINE SCHEDULE) SHARES SPACE WITH
C COMMON BLOC 'SHARES'.
C
      DATA MTHMAP/2,4,4,5,4,5,1,3,5,5,5,5/
      DATA MEDMAP/1,2,4,4,4,4,1,2,3,2,2,5/
      DATA JMAT/1,5,0,3,4,0,2,2,2,3,4,0/
      DATA MAPPSY/1,2,2,3,6/
      DATA MAPCOG/1,3,4,4,5/
      DATA CODE/'TTS','OJT'/
      DATA MET/' LECTURE','DISCUSSION','DEMNSTATN.',
     & 'SIMULATION','PERFRMNCE',' '/
      DATA MED/'XPARENCIES','TRNG FILMS',' MOCK-UPS',
     & 'SIMULATORS','EQUIPMENT',' '/
      DATA BLANK,END,ID,NINES,ZERO/' ','END','ID','99','0'/
C
C READ NAMES - FIRST DUTIES. CARDS WITHOUT 'ID' IN COL 1-2 ARE COMMENTS
      REWIND 2
      REWIND 10
      READ(2,10) NDUT,NTSK
   10 FORMAT(2I3)
      J=1
   20 READ(2,30) ICH,KDUT(J),(DUT(K,J),K=1,4)
   30 FORMAT(A2,4XA2,22X4A10)
      IF (ICH.NE.ID) GO TO 20
      J=J+1
      IF (J.LE.NDUT) GO TO 20
C
C NOW TASK NAMES
      J=1
   40 READ(2,50) ICH,KTSK(J),(TSK(K,J),K=1,4)
   50 FORMAT(A2,6XA2,20X4A10)
      IF (ICH.NE.ID) GO TO 40
      J=J+1
      IF (J.LE.NTSK) GO TO 40
C
C GET AFSC OF INTEREST.  BLANK ENTRY GETS NEXT IN FILE
   60 PRINT 70
   70 FORMAT('0ENTER AFSC-    ')
      READ(5,80) AFSC
   80 FORMAT(A6)
      IEOF=0
   90 READ(10,80) GRID
      IF (EOF(10).NE.0) GO TO 100
      IF (AFSC.EQ.BLANK.OR.AFSC.EQ.GRID) GO TO 130
      GO TO 90
C
C END OF FILE REACHED. IF WE'VE BEEN THROUGH THE FILE ALREADY,
C THE REQUESTED AFSC IS NOT ON THE FILE. IF NOT, REWIND AND RE-TRY.
  100 IF (IEOF.EQ.0) GO TO 120
      PRINT 110,AFSC
  110 FORMAT(1XA6,' NOT FOUND')
      REWIND 10
      GO TO 60
  120 REWIND 10
      IEOF=1
      GO TO 90
C
C NK IS NUMBER OF INPUT DATA TASKS. PRINT AFSC.
  130 NK=1
      DO 135  J=1,5
      DO 135  K=1,5
  135 LRO(J,K)=0
      IF (AFSC.EQ.BLANK) PRINT 140,GRID
  140 FORMAT(1X11('*')/1X'AFSC-',A6/1X11('*'))
C
C READ NEXT TASK.  NUM1 AND NUM2 ARE THE DUTY AND TASK.
```

```
1390=   150 READ(10,160) TNK,NUM1,NUM2,SUB,CHS,FLS,KEEP,C1,C2,T1,T2
1400=   160 FORMAT(A7,2A2,A1,5F6.0,3F2.0,I1,F6.2,F7.2,2F5.0)
1410=       IF (NUM2.EQ.NINES) GO TO 200
1420=       CALL CONV(NUM1,NUM2)
1430=       IF (NK.GT.350) GO TO 180
1440=C
1450=C CONVERT TIMES AND COSTS
1460=       TS(NK,1)=T1
1470=       TS(NK,2)=T2
1480=       CS(NK,1)=C1
1490=       CS(NK,2)=C2
1500=C
1510=C PACK TASK, 5 CHARACTERISTICS, 3 FLAGS, DUTY AND TASK
1515=       ENCODE(9,227,TASKS(NK)) TNK,SUB
1520=       DO 170  J=1,5
1530=       JCH=CHS(J)
1540=       INONE(J)=JCH
1550=       LRO(JCH,J)=LRO(JCH,J)+1
1560=   170 IF (J.LE.3) INONE(J+5)=FLS(J)
1570=       INONE(9)=NUM1
1580=       INONE(10)=NUM2
1590=       CALL PACK(INONE,KAS)
1600=C
1610=C IF THE TASK IS FORCED THROUGH SCREENING, KASTS IS SET NEGATIVE
1620=       IF (KEEP.GT.0) KAS=-KAS
1630=       KASTS(NK)=KAS
1640=       NK=NK+1
1650=       GO TO 150
1660=C
1670=C ERROR
1680=   180 PRINT 190
1690=   190 FORMAT('0DATA BASE ERROR. MORE THAN 350 TASKS FOR THIS AFSC')
1692=   192 READ (10,160) TNK,NUM1,NUM2
1694=       IF (NUM2.EQ.NINES) GO TO 60
1696=       GO TO 192
1710=C
1720=C PRINT HISTOGRAM OF ALL TASKS
1730=   200 NK=NK-1
1740=       PRINT 210,NK,GRID
1745=   210 FORMAT(/I4,' TASKS BEFORE SCREENING'/
1750=      * 'ORESPOND Y (DISPLAY) OR N (INHIBIT) TO HISTOGRAM TITLES'
1760=      $//' ** HISTOGRAM OF COMPLETE TASK DATA FOR',1XA6,1X'**    ')
1770=       CALL INP(2,0,0,IANS),RETURNS(760)
1780=       IF (IANS.EQ.1) CALL HIST(LRO)
1785=       GO TO 228
1790=C
1800=C SCREEN TASKS AND PREPARE FOR LISTING
1810=   220 DO 225  I=1,NK
1820=       DECODE(9,245,TASKS(I)) TEST,TNK,SUB
1830=   225 ENCODE(9,227,TASKS(I)) TNK,SUB
1840=   227 FORMAT('0',A7,A1)
1850=   228 CALL SCREEN(NK),RETURNS(760)
1860=       PRINT 230
1870=   230 FORMAT(' DO YOU WANT A LISTING OF REJECTED TASKS?    ')
1880=       CALL INP(2,0,0,IAN),RETURNS(760)
1890=       IF (IAN.EQ.1) PRINT 240
1900=   240 FORMAT('0 KEY EQUIP    TASK    NAME',36X'C D F P C')
1905=       DO 242  J=1,5
1910=       DO 242  K=1,5
1915=       LRO(J,K)=0
1920=   242 NRO(J,K)=0
1930=C
1940=C LOOP THROUGH TASKS, SETTING UP LRO AND NRO TO PASS TO HISTOGRAM
1950=C SUBROUTINE.  REJECTED TASKS ARE PRINTED (IF IAN IS SET).
1960=C SELECTED TASKS HAVE A 0 IN TASKS.
1970=       DO 270  I=1,NK
```

```
1980=       KAS=IABS(KASTS(I))
1990=       DECODE(9,245,TASKS(I)) TEST,TNK,SUB
2000=   245 FORMAT(A1,A7,A1)
2010=       IF (TEST.EQ.ZERO) GO TO 250
2030=       CALL UNPACK(KAS,INONE)
2033=       DO 248  J=1,5
2036=       JCH=INONE(J)
2039=   248 LRO(JCH,J)=LRO(JCH,J)+1
2040=       GO TO 270
2060=   250 CALL UNPACK(KAS,INONE)
2063=       DO 255  J=1,5
2066=       JCH=INONE(J)
2069=   255 NRO(JCH,J)=NRO(JCH,J)+1
2070=       IF (IAN.EQ.0) GO TO 270
2080=       NUM1=INONE(9)
2090=       NUM2=INONE(10)
2100=       PRINT 260,I,TNK,KDUT(NUM1),KTSK(NUM2),SUB,(TSK(K,NUM2),
2110=      & K=1,4),(INONE(K),K=1,5)
2120=   260 FORMAT(I5,1XA7,1X2A2,A1,1X4A10,5(1XI1))
2130=   270 CONTINUE
2140=C
2150=C PRINT HISTOGRAMS OF SELECTED AND REJECTED TASKS
2160=       PRINT 280,GRID
2170=   280 FORMAT('0** HISTOGRAM OF TASKS TO BE TRAINED FOR ',
2180=      & A6,1X'**    ')
2190=       CALL INP(2,0,0,IANS),RETURNS(760)
2200=       IF (IANS.EQ.1) CALL HIST(LRO)
2210=       PRINT 290,GRID
2220=   290 FORMAT('0** HISTOGRAM OF REJECTED TASKS FOR ',A6,1X' **    ')
2230=       CALL INP(2,0,0,IANS),RETURNS(760)
2240=       IF (IANS.EQ.1) CALL HIST(NRO)
2250=C
2260=C
2270=       PRINT 300,NBLOC
2275=   300 FORMAT(/I4,' TASKS AFTER SCREENING'/
2280=      * '0DO YOU WANT TO SCREEN THE TASKS OVER AGAIN?    ')
2290=       CALL INP(2,0,0,IANS),RETURNS(760)
2300=       IF (IANS.EQ.0) GO TO 360
2310=C
2320=C IF REJECTED TASKS WERE LISTED, USER CAN SELECTIVELY FORCE
2330=C ANY REJECTED THROUGH THE NEXT SCREENING.
2340=       IF (IAN.EQ.0) GO TO 220
2350=       PRINT 310
2360=   310 FORMAT(' DO YOU WANT ANY TASKS LISTED ABOVE FORCED',
2370=      & ' THROUGH SUBSEQUENT SCREENING?    ')
2380=       CALL INP(2,0,0,IANS),RETURNS(760)
2390=       IF (IANS.EQ.0) GO TO 220
2400=       PRINT 320
2410=   320 FORMAT(' ENTER KEY NUMBER OF TASKS TO BE FORCED. END WITH 0     ')
2420=   330 CALL INP(0,0,NK,JK),RETURNS(760)
2430=       IF (JK.EQ.0) GO TO 220
2440=       DECODE(9,245,TASKS(JK)) TEST
2450=       IF (TEST.NE.ZERO) GO TO 340
2460=       KASTS(JK)=-KASTS(JK)
2470=       GO TO 330
2480=   340 PRINT 350,JK
2490=   350 FORMAT(' TASK WITH KEY',I4,' WAS NOT REJECTED')
2500=       GO TO 330
2510=C
2520=C
2525=C DETERMINE TRAINING MODE FOR SELECTED TASK BLOCKS
2530=   360 PRINT 370,GRID
2540=   370 FORMAT('0ENTER NUMBER OF TRAINEES IN GROUP ',A6,2X'-    ')
2550=       CALL INP(0,1,7607607609,NTR),RETURNS(760)
2560=       CALL TRMODE(KOPT),RETURNS(760)
2570=C
```

```
2580=C
2610=       NLOOP=0
2620=       NITER=0
2630=C KOPT IS ZERO IF BOTH MODES WILL EVENTUALLY BE USED IN TWO DIFFERE
2640=C TRAINING PROGRAMS. KOPT IS 1 IF ALL TASKS WERE FORCED TO ONE MODE
2650=C KOPT IS 2 IF THE MIX OPTION WAS SELECTED (ONE PLAN).
2660=       IF (KOPT.NE.0) GO TO 410
2670=  380 NLOOP=NLOOP+1
2680=C FOR NLOOP = 1 AND THEN 2, WE WILL TRAIN ALL TASKS IN THAT MODE (EXCEPT
2690=C THOSE FORCED THE OTHER WAY).
2700=       PRINT 390,CODE(NLOOP)
2710=  390 FORMAT('0** TRAINING PROGRAM FOR ',A3,' STUDENTS **')
2720=       DO 400 I=1,NBLOC
2730=       CALL UNPACK(MAXES(I),INONE)
2740=       M=INONE(6)
2750=       INONE(6)=JMAT(M+1,NLOOP)
2760=  400 IF (M.NE.3.AND.M.NE.4) CALL PACK(INONE,MAXES(I))
2770=C
2780=C THE ABOVE MAGIC SETS THE MODE OF EACH TASK ACCORDING TO THE FOLLOWING
2790=C  1) IF ZERO, GETS SET TO 1 OR 2 PER NLOOP.
2800=C  2) IF 3 OR 4 (FORCED BOTH WAYS), NO CHANGE IS MADE.
2810=C  3) IF 1, GETS SET TO 5 IF NLOOP IS 1.  WHEN NLOOP IS 2, THE FIVE WILL
2820=C     WILL BE RESET TO ZERO.  TTS WILL BE PROCESSED, BUT NOT OJT.
2830=C  4) IF 2, GETS SET TO 0 FOR NLOOP=1, AND AS USUAL, 2 FOR NLOOP=2.  THI
2840=C     PROHIBITS TTS AND ALLOWS OJT PROCESSING.
2850=C  5) JMAT IS THE MATRIX SHOWN BELOW.  FOR EACH NLOOP AND EACH POSSIBLE
2860=C     MODE NUMBER, JMAT IS THE RESULTANT MODE SETTING. ( EXCEPT THAT MOD
2870=C     CANNOT EQUAL 5 WHEN NLOOP = 1)  IF THERE IS NO CHANGE TO BE MADE,
2880=C
2890=C           M  I  NLOOP=1  I  NLOOP=2
2900=C          ----I----------I----------
2910=C              I          I
2920=C           0  I    1     I   2
2930=C              I          I
2940=C           1  I    5     I   0
2950=C              I          I
2960=C           2  I    0     I   -
2970=C              I          I
2980=C           3  I    3     I   3
2990=C              I          I
3000=C           4  I    4     I   4
3010=C              I          I
3020=C           5  I    -     I   0
3030=C
3040=  410 PRINT 420
3050=  420 FORMAT('0DO YOU WANT TO ENTER AN ALTERNATIVE MAPPING',
3060=     $ ' FROM TAXONOMY'/' TO TRAINING OBJECTIVE?   ')
3070=       CALL INP(2,0,0,IANS),RETURNS(760)
3080=       IF (IANS.EQ.0) GO TO 465
3100=       PRINT 430
3110=  430 FORMAT('0ENTER THE NUMBER OF THE TRAINING OBJECTIVE ASSOCIATED',
3120=     $ ' WITH EACH OF THE'/' FOLLOWING TAXONOMIC LEVELS, WHERE:'/
3130=     $ ' T.O. 1 = LEARNING IDENTIFICATIONS'/' T.O. 2 = LEARNING PERCEP'
3140=     $ ,'TUAL DISCRIMINATIONS'/' T.O. 3 = UNDERSTANDING PRINCIPLES',
3150=     $ ' AND RELATIONSHIPS'/' T.O. 4 = LEARNING PROCEDURAL SEQUENCE'/
3160=     $ ' T.O. 5 = MAKING DECISIONS'/' T.O. 6 = PERFORMING SKILLED',
3170=     $ ' PERCEPTUAL MOTOR ACTS'/)
3180=       DO 450  I=1,5
3190=       PRINT 440,LABEL(1,4),LABEL(2,4),I
3200=  440 FORMAT(1XA10,A2,'=',I1,' T.O.=   ')
3210=  450 CALL INP(0,1,6,MAPPSY(I)),RETURNS(760)
3220=       DO 460  I=1,5
3230=       PRINT 440,LABEL(1,5),LABEL(2,5),I
3240=  460 CALL INP(0,1,6,MAPCOG(I)),RETURNS(760)
3250=C
3260=C
```

```
3270=   465 PRINT 470
3280=   470 FORMAT('ODO YOU WISH TO ENTER AN ALTERNATIVE MAPPING',
3290=      & /' FROM TRAINING OBJECTIVE TO METHODS AND MEDIA?   ')
3300=       CALL INP(2,0,0,IANS),RETURNS(760)
3310=       IF (IANS.EQ.0) GO TO 515
3330=       PRINT 480
3340=   480 FORMAT('OENTER THE NUMBERS OF THE TRAINING METHOD AND TRAINING',
3350=      & ' MEDIUM'/' ASSOCIATED WITH EACH OF THE FOLLOWING TRAINING',
3360=      & ' OBJECTIVES, WHERE:'/' METHOD 1 = INFORMAL LECTURE',T40,
3370=      & 'MEDIUM 1 = TRANSPARENCIES'/' METHOD 2 = DISCUSSION',T40,
3380=      & 'MEDIUM 2 = TRAINING FILMS'/' METHOD 3 = DEMONSTRATION',T40,
3390=      & 'MEDIUM 3 = MOCK-UPS'/' METHOD 4 = SIMULATION',T40,
3400=      & 'MEDIUM 4 = SIMULATOR'/' METHOD 5 = PERFORMANCE',T40,
3410=      & 'MEDIUM 5 = ACTUAL EQUIPMENT'/' METHOD 6 = OTHER (LEFT BLANK)',
3420=      & T40,'MEDIUM 6 = OTHER (LEFT BLANK)'/)
3430=       DO 510  I=1,6
3440=       PRINT 490,I,CODE(1)
3450=   490 FORMAT(' TRAINING OBJECTIVE=',I1,'; METHOD FOR ',A3,' =    ')
3460=       CALL INP(0,1,6,MTHMAP(I,1)),RETURNS(760)
3470=       PRINT 500,CODE(1)
3480=   500 FORMAT(23X,'MEDIUM FOR ',A3,' =    ')
3490=       CALL INP(0,1,6,MEDMAP(I,1)),RETURNS(760)
3500=       PRINT 490,I,CODE(2)
3510=       CALL INP(0,1,6,MTHMAP(I,2)),RETURNS(760)
3520=       PRINT 500,CODE(2)
3530=   510 CALL INP(0,1,6,MEDMAP(I,2)),RETURNS(760)
3540=C
3550=C
3555=C USE TAXONOMY-TO-TRAINING OBJECTIVE-TO-METHODS AND MEDIA MAPPINGS
3556=C TO ASSIGN EACH TASK BLOCK A TRAINING METHOD AND MEDIUM.
3560=   515 CALL METMED(NITER,NLOOP)
3570=       CALL REPT(GRID,NLOOP)
3580=C
3590=C
3600=       PRINT 520
3610=   520 FORMAT('ODO YOU WISH TO GENERATE ANOTHER METHODS AND MEDIA',
3620=      & ' ASSIGNMENT?    ')
3630=       CALL INP(2,0,0,IANS),RETURNS(760)
3640=       NITER=NITER+1
3650=       IF (IANS.EQ.1) GO TO 410
3660=C
3670=       PRINT 530
3680=   530 FORMAT('ODO YOU WISH TO REGENERATE THE TRAINING PLAN?    ')
3690=       CALL INP(2,0,0,IANS),RETURNS(760)
3700=       IF (IANS.EQ.0) GO TO 540
3710=       NBLOC=0
3720=       GO TO 220
3730=   540 NMEDIA=0
3731=C
3732=C PREPARE TO SCHEDULE TRAINING.  SELECT CHARACTERISTIC TO ORDER
3733=C TRAINING SEQUENCE AND HIGH COST MEDIUM TO BE ASSIGNED OPTIMALLY.
3740=       PRINT 550, (J,LABEL(1,J),LABEL(2,J),J=1,5)
3750=   550 FORMAT('OENTER NUMBER INDICATING CHARACTERISTIC TO  GOVERN',
3760=      & ' ORDERING OF TASKS'/' FOR TRAINING.  THIS TASK HIERARCHY',
3770=      & ' ORDERS FROM LOW TO HIGH'.5(/I2,'=',A10,A2)/
3780=      & ' 0=KEEP INITIAL ORDERING OF TASKS'/' =    ')
3790=       CALL INP(0,0,5,NHIER),RETURNS(760)
3800=       IF (NLOOP.EQ.2) GO TO 590
3810=       PRINT 560
3820=   560 FORMAT('OENTER NUMBER INDICATING MEDIUM TO BE OPTIMALLY',
3830=      & ' SCHEDULED'/' 1=TRANSPARENCIES'/' 2=TRAINING FILMS'/
3840=      & ' 3=MOCK-UPS'/' 4=SIMULATORS'/' 5=ACTUAL EQUIPMENT'/
3850=      & ' 0=IGNORE OPTION'/' =    ')
3860=       CALL INP(0,0,5,NMEDIA),RETURNS(760)
3870=       PRINT 570
3880=   570 FORMAT(' ENTER MINIMUM CLASS SIZE FOR ITS-    ')
```

```
3890=       CALL INP(0,0,0,MIN),RETURNS(760)
3900=       PRINT 580
3910=   580 FORMAT(' ENTER MAXIMUM CLASS SIZE FOR TTS-    ')
3920=       CALL INP(0,MIN,999999999999999,MAX),RETURNS(760)
3930=C
3940=C
3950=   590 CALL SCHEDU(NLOOP,NUMB,MEDTOT,KODE)
3960=       IF (KODE-2) 620,600,630
3970=   600 PRINT 610,NCL,NST(1)
3980=   610 FORMAT(/I5,' TTS CLASSES ARE NEEDED FOR',I5,' STUDENTS')
3990=   620 NUSE(1)=1
4000=       GO TO 700
4001=C
4002=C  DETERMINE NUMBER OF CLASSES TO FOLLOW EACH SCHEDULE ARRANGEMENT.
4003=C  NR=REMAINDER CLASSES TO BE ASSIGNED AS NEEDED. NUSE(I)= NUMBER
4004=C  OF CLASSES FOLLOWING SCHEDULE ARRANGEMENT I.
4010=   630 NPER=NCL/NSCH
4020=       NR=NCL-(NPER*NSCH)
4030=       DO 640  I=1,NSCH
4040=   640 NUSE(I)=NR
4050=       IF (NR.EQ.0) GO TO 660
4060=       DO 650  I=1,NR
4070=   650 NUSE(I)=NUSE(I)+1
4080=   660 DO 670  I=1,NSCH
4090=   670 IF (NUSE(I).GT.0) PRINT 680,NUSE(I),I
4100=   680 FORMAT(I4,' CLASSES FOLLOW SCHEDULE ARRANGEMENT',I3)
4110=       PRINT 690,MEDTOT,MED(NMEDIA)
4120=   690 FORMAT('0THE TRAINING PROGRAM REQUIRES',I3,1XA10)
4130=C
4140=C
4150=   700 DO 750  J=1,NSCH
4160=       IF (NUSE(J).LE.0) GO TO 750
4170=       PRINT 710,J
4180=   710 FORMAT('0** SCHEDULE ARRANGEMENT',I3,' **'/
4190=      & '0SUBSYS  TASK  MODE  DAYS   COST',3X'METHOD     MEDIUM')
4200=       DO 730  I=1,NUMB
4210=       KN=NSCHED(I,J)
4220=       CALL UNPACK(MAXES(KN),INONE)
4230=       M=INONE(6)
4240=       IF (M.EQ.0) GO TO 730
4250=       M2=INONE(7)
4260=       M3=INONE(8)
4270=       CALL DECIDE(M,NLOOP,MM)
4280=       NUM1=INONE(9)
4290=       NUM2=INONE(10)
4300=       CALL HALVES(JTIMES(KN),TVAL)
4310=       CALL HALVES(JCOSTS(KN),CVAL)
4320=       PRINT 720,TRPLAN(KN),KDUT(NUM1),KTSK(NUM2),CODE(MM),TVAL(M)/8.,
4330=      & CVAL(M),MET(M2),MED(M3)
4340=   720 FORMAT(1XA7,1X2A2,'0',1XA3,2F7.1,2(1XA10))
4350=   730 CONTINUE
4360=       PRINT 740
4370=   740 FORMAT(///)
4380=   750 CONTINUE
4390=C
4400=C
4410=       IF (KOPT.EQ.0.AND.NLOOP.EQ.1) GO TO 380
4420=   760 PRINT 770
4430=   770 FORMAT('0DO YOU WANT TO GENERATE A TRAINING PLAN FOR'/
4440=      & ' ANOTHER AFSC?    ')
4450=       CALL INP(2,0,0,IANS),RETURNS(780)
4460=       IF (IANS.EQ.1) GO TO 60
4470=   780 STOP
4480=       END
4490=       SUBROUTINE SCREEN(NK),RETURNS(ABORT)
4500=       DIMENSION LABEL(2,4),N(5),LN(5),TVAL(2),CVAL(2),IX(10)
```

```
4510=       COMMON/BLOC1/ NBLOC,TRPLAN(350),MAXES(350),JCOSTS(350),
4520=      & JTIMES(350),INONE(10)
4530=       COMMON/SHARE/ TASKS(350),KASTS(350),CS(350,2),TS(350,2)
4550=       COMMON/LABELS/LABEL
4560=       COMMON/NAMES/ DUT(4,6),TSK(4,62),KDUT(6),KTSK(62),NDUT,NTSK
4570=       DATA LABEL/'CRITICALIT','Y','DIFFICULTY',' ','FREQUENCY',
4580=      & ' ','PSYCHOMOTO','R','COGNITIVE',' ','CONSTANT',' '/
4590=       IPAIR(TVAL)=IFIX(TVAL(1)*100) + IFIX(TVAL(2)*100)*100000000
4600=       DATA ONE/'1'/
4610=C
4615=C SELECT TASK DECISION ALGORITHM AND THRESHOLDS
4620=    10 PRINT 20
4630=    20 FORMAT('0ENTER CUTOFF LEVEL (0 THRU 5) FOR:')
4640=       DO 40  I=1,5
4650=       PRINT 30,LABEL(1,I),LABEL(2,I)
4660=    30 FORMAT(1XA10,A2,'-  ')
4670=    40 CALL INP (0,0,5,N(I)),RETURNS(320)
4680=       PRINT 50
4690=    50 FORMAT ('0ENTER SCREENING OPTION'/* 1='ANY' DECISION TREE*/
4700=      & * 2='ALL' DECISION TREE*/' 3=RMS TEST'/' 4=WEIGHTED AVERAGE'/
4710=      & ' 5=PURE AVERAGE'/' =     ')
4720=       CALL INP (0,1,5,NOPT),RETURNS(320)
4730=       NBLOC=0
4740=       NR=1
4750=C
4760=C SET UP CONSTANTS TO BE USED IN LOOP BELOW
4770=       IF (NOPT.LT.3) GO TO 70
4780=       ISUM=0
4790=       SUM=0.
4800=       DO 60  I=1,5
4810=       NI=N(I)
4820=       LN(I)=6-NI
4830=       ISUM=ISUM+NI*NI
4840=    60 SUM=SUM+NI
4850=       IF (NOPT.EQ.5) GO TO 70
4860=       SUM=30.0-SUM
4870=       SUM2=SUM/5.0
4880=C
4890=C LOOP THROUGH TASKS. IF KASTS IS NEGATIVE, TASK IS TO BE FORCED
4900=C THROUGH THE SCREENING.  IF THE FIRST CHARACTER OF TASKS IS '1',
4910=C THE TASKS IS FLAGGED TO HAVE PASSED THE SCREENING.  IT IS INITIALLY
4920=C SET TO ZERO BEFORE THE SCREENING PROCESS.
4930=    70 KAS=KASTS(NR)
4940=       CALL UNPACK(IABS(KAS),IX)
4950=       DECODE(9,75,TASKS(NR)) TEST,SYS,SUB
4960=       IF (TEST.EQ.ONE) GO TO 300
4970=    75 FORMAT(A1,A7,A1)
4980=       IF (KAS.LT.0) GO TO 200
4990=       GO TO (90,110,130,150,180),NOPT
5000=C
5010=C ANY DECISION TREE
5020=    90 DO 100  I=1,5
5030=       IF (IX(I).GE.N(I)) GO TO 200
5040=   100 CONTINUE
5050=       GO TO 300
5060=C
5070=C ALL DECISION TREE
5080=   110 DO 120  I=1,5
5090=       IF (IX(I).LT.N(I)) GO TO 300
5100=   120 CONTINUE
5110=       GO TO 200
5120=C
5130=C RMS TEST
5140=   130 IT=0
5150=       DO 140  I=1,5
5160=   140 IT=IT+IX(I)**2
```

```
5170=       IF (IT-ISUM) 300,200,200
5180=C
5190=C WEIGHTED AVERAGE
5200=  150 AVE=0.
5210=       DO 160  I=1,5
5220=  160 AVE=AVE+IX(I)*LN(I)/SUM
5230=       IF (AVE-SUM2) 300,200,200
5240=C
5250=C PURE AVERAGE
5260=  180 AVE=0.
5270=       DO 190  I=1,5
5280=  190 AVE=AVE+IX(I)
5290=       IF (AVE.LT.SUM) GO TO 300
5300=C
5310=C SELECTED TASK.  TRAIN ALL SUBTASKS WITHIN TASK BLOCK.
5320=  200 NBLOC=NBLOC+1
5330=       IJ=NR
5340=       IBASE=IX(10)
5350=       IDUTY=IX(9)
5360=C
5370=C SEE IF PREDECESSOR IS SAME SYSTEM/DUTY/TASK
5380=  210 IJ=IJ-1
5390=       IF (IJ.LT.1) GO TO 220
5400=       DECODE(9,75,TASKS(IJ)) TEST,SYSNEW,SUBNEW
5410=       IF (SYS.NE.SYSNEW) GO TO 220
5420=       CALL UNPACK(IABS(KASTS(IJ)),IX)
5430=       IF (IX(9).EQ.IDUTY.AND.IBASE.EQ.IX(10)) GO TO 210
5440=C
5450=C IJ WILL MARK THE FIRST SUBTASK OF A TASK BLOCK.  WE SET MODE
5460=C AND METHOD AND MEDIA.  INITIALIZE TIMES AND COSTS AND PREPARE
5470=C FOR FINDING CHARACTERISTIC MAXIMA.
5480=  220 IJ=IJ+1
5490=       JTIMES(NBLOC)=0
5500=       JCOSTS(NBLOC)=0
5510=       TRPLAN(NBLOC)=SYS
5520=       CALL UNPACK(IABS(KASTS(IJ)),IX)
5530=       MODE=IX(8)
5540=       DO 230  J=1,5
5550=  230 INONE(J)=0
5560=       DO 235  J=6,10
5570=  235 INONE(J)=IX(J)
5580=       TVAL(1)=0.
5590=       TVAL(2)=0.
5600=       CVAL(1)=0.

            CVAL(2)=0.
5620=C
5630=C NOW EACH SUBTASK
5640=  240 DO 250  J=1,2
5650=       IF (MODE.EQ.3-J) GO TO 250
5660=       TVAL(J)=TVAL(J)+TS(IJ,J)
5670=       CVAL(J)=CVAL(J)+CS(IJ,J)
5680=  250 CONTINUE
5690=       DO 260  J=1,5
5700=       MM=IX(J)
5710=  260 IF (MM.GT.INONE(J)) INONE(J)=MM
5720=       ENCODE(9,265,TASKS(IJ)) SYS,SUB
5730=  265 FORMAT('1',A7,A1)
5740=C
5750=C SEE IF NEXT TASK IS IN SAME GROUP OF SUBTASKS
5760=       IJ=IJ+1
5770=       IF (IJ.LE.NR) GO TO 270
5780=       IF (IJ.GT.NK) GO TO 280
5790=       DECODE(9,75,TASKS(IJ)) TEST,SYSNEW,SUBNEW
5800=       IF (SYSNEW.NE.SYS) GO TO 280
5810=  270 CALL UNPACK(IABS(KASTS(IJ)),IX)
```

```
5820=       IF (IJ.LE.NR.OR.(IX(9).EQ.IDUTY.AND.IX(10).EQ.IBASE))GOTO240
5830=C
5840=C FINALIZE TASK BLOCK PARAMETERS
5850=  280 CALL PACK(INONE,MAXES(NBLOC))
5860=       JTIMES(NBLOC)=IPAIR(TVAL)
5870=       JCOSTS(NBLOC)=IPAIR(CVAL)
5880=       TRPLAN(NBLOC)=SYS
5890=C
5900=C NEXT TASK
5910=  300 NR=NR+1
5920=       IF (NR.LE.NK) GO TO 70
5930=       IF (NBLOC.GT.0) RETURN
5940=       PRINT 310
5950=  310 FORMAT('ONO TASKS PASSED SCREENING. INPUT NEW CHOICES')
5960=       GO TO 10
5970=  320 RETURN ABORT
5980=       END
5990=       SUBROUTINE TRMODE(NOPT),RETURNS(ABORT)
6000=       DIMENSION TTSOJT(4),COEFF(4,6),INPUT1(4),INPUT2(4),INPUT3(4)
6010=C
6020=C INPUT1 - 0 TO USE TTS AND OJT TIMES AND COSTS AS THEY ARE
6030=C          1 TO USE REGRESSION COEFFICIENTS TO DERIVE SOME OR ALL
6040=C INPUT2 - 0 TO ENTER YOUR OWN COEFFICIENTS FOR THIS CATEGORY
6050=C          1 TO USE THE DEFAULT VALUES (OR VALUES ENTERED PREVIOUSLY)
6060=C INPUT3 - 0 TO DERIVE VALUES FOR ALL TASK BLOCKS FOR THIS CATEGORY
6070=C          1 TO DERIVE VALUES ONLY IF CURRENTLY ZERO
6080=C
6090=       DIMENSION T(2),C(2),TVAL(2),CVAL(2)
6100=       COMMON/BLOC1/ NBLOC,TRPLAN(350),MAXES(350),JCOSTS(350),
6110=      & JTIMES(350),INONE(10)
6120=       COMMON/BLOC3/ NTR,NST(2),DUM(36)
6130=       COMMON/LABELS/LABEL(2,6)
6140=       COMMON/MINT/CODE(2)
 6150=       DATA COEFF/0.,0.,-.240886,1.137375,0.,0.,-1.130284,6.14645,
6160=      & 0.,0.,-124.5503,-1.487537,0.,0.,-2.23228,2.534994,
6170=      & 0.,0.,14.86655,1.1387448,15.,4.36,617.4578,-7.725949/
6180=       DATA TTSOJT/'TTS COSTS','OJT COSTS','TTS TIMES','OJT TIMES'/
6190=       IPAIR(TVAL)=IFIX(TVAL(1)*100) + IFIX(TVAL(2)*100)*100000000
6200=C
6210=C INPUT REGRESSION CHOICES
6220=       NSUM=0
6230=       PRINT 20
6240=   20 FORMAT('0DO YOU WISH TO USE REGRESSION COEFFICIENTS FOR:')
6250=       DO 40  K=1,4
6260=       PRINT 30,TTSOJT(K)
6270=   30 FORMAT(1XA9,'?     ')
6280=       CALL INP(2,0,0,IANS),RETURNS(330)
6290=       INPUT1(K)=IANS
6300=   40 IF (IANS.EQ.0) NSUM=NSUM+1
6310=       IF (NSUM.EQ.4) GO TO 200
6320=C
6330=       MSUM=0
6340=       DO 70  K=1,4
6350=       IF (INPUT1(K).EQ.0) GO TO 70
6360=       PRINT 50,TTSOJT(K)
6370=   50 FORMAT(' DO YOU WISH TO DERIVE ',A9,' FOR ONLY THOSE ',
6380=      & 'WHICH ARE CURRENTLY ZERO?     ')
6390=       CALL INP(2,0,0,INPUT3(K)),RETURNS(330)
6400=       PRINT 60,TTSOJT(K)
6410=   60 FORMAT(' WILL YOU USE DEFAULT REGRESSION COEFFICIENTS FOR',
6420=      & 1XA9,'?     ')
6430=       CALL INP(2,0,0,IANS),RETURNS(330)
6440=       INPUT2(K)=IANS
6450=       IF (IANS.EQ.0) MSUM=MSUM+1
6460=   70 CONTINUE
6470=C
```

```
6480=C INPUT REGRESSION COEFFICIENTS
6490=       IF (MSUM.EQ.0) GO TO 120
6500=       DO 110  I=1,4
6510=       IF (INPUT1(I).EQ.0) GO TO 110
6520=       IF (INPUT2(I).EQ.1) GO TO 110
6530=       PRINT 80,TTSOJT(I)
6540=    80 FORMAT(' TO DERIVE ',A9,', ENTER REGRESSION COEFFICIENTS FOR:')
6550=       DO 100  J=1,6
6560=       PRINT 90,LABEL(1,J),LABEL(2,J)
6570=    90 FORMAT(1XA10,A2,'=     ')
6580=   100 CALL INP(1,0,0,COEFF(I,J)),RETURNS(330)
6590=   110 CONTINUE
6600=C
6610=C UPDATE TASK DATA BASED ON NEW CHOICES
6620=   120 DO 180  I=1,NBLOC
6630=       CALL UNPACK(MAXES(I),INONE)
6650=       DO 180  K=1,4
6660=       IF (INPUT1(K).EQ.0) GO TO 180
6670=       MODE=MOD(K,2)
6680=C DISALLOW ENTERING DATA FOR OPPOSITE OF FORCED MODE
6690=       IF (MODE.EQ.3-INONE(6)) GO TO 180
6700=C
6710=C EXTRACT CURRENT VALUE
6720=       IF (K.GT.2) GO TO 130
6730=       JTEMP=JCOSTS(I)
6740=       GO TO 140
6750=   130 JTEMP=JTIMES(I)
6760=   140 CALL HALVES(JTEMP,TVAL)
6770=       TEMP=TVAL(MODE)
6780=C
6790=C TEMP IS THE TTS OR OJT COST OR TIME INITIALLY SUPPLIED ONE
6800=C THE TASK DATA INPUT.  IF ZERO, WE WILL REGRESS FOR IT. OF NOT WE
6810=C CHECK INPUT3 TO SEE IF WE SHOULD OVERRIDE OR RETAIN IT.
6820=       IF (TEMP.EQ.0.0) GO TO 170
6830=       IF (INPUT3(K).EQ.1) GO TO 180
6840=       TEMP=0.0
6860=   150 DO 160  J=1,5
6870=   160 TEMP=TEMP+COEFF(K,J)*INONE(J)
6880=       TVAL(MODE)=TEMP+COEFF(K,6)
6890=       JTEMP=IPAIR(TVAL)
6900=       IF (K.GT.2) GO TO 170
6910=       JCOSTS(I)=JTEMP
6920=       GO TO 180
6930=   170 JTIMES(I)=JTEMP
6940=   180 CONTINUE
6950=C
6960=C SUM UP COSTS AND TIMES FOR BOTH TRAINING MODES
6970=   200 DO 210  J=1,2
6980=       C(J)=0.
6990=   210 T(J)=0.
7000=       DO 230  I=1,NBLOC
7010=       CALL HALVES(JTIMES(I),TVAL)
7020=       CALL HALVES(JCOSTS(I),CVAL)
7030=       DO 220  J=1,2
7040=       TEMP=TVAL(J)
7050=       T(J)=T(J)+TEMP
7060=       CEMP=CVAL(J)*TEMP
7070=       CVAL(J)=CEMP
7080=   220 C(J)=C(J)+CEMP
7090=   230 JCOSTS(I)=IPAIR(CVAL)
7100=       PRINT 240,(T(J)/8.,J=1,2),(C(J),J=1,2)
7110=   240 FORMAT('0TIME TO TRAIN IS SCHOOL: ',F5.1,' DAYS',1X
7120=      & 'TIME TO TRAIN IN OJT: ',F5.1,' DAYS'/
7130=      & ' COST TO TRAIN IN SCHOOL: '
7140=      & ,F10.1,' COST TO TRAIN IN OJT: ',F10.1)
7150=C
```

```
7160=C GET TRAINING OPTION AND TIME LIMIT
7170=      PRINT 250
7180=  250 FORMAT ('0INPUT TRAINING MODE OPTION:'/' 1=NON-MIX'/
7190=     & ' 2=SCHOOL/OJT TRAINING MIX'/' =    ')
7200=      CALL INP(0,1,2,NOPT),RETURNS(330)
7210=      PRINT 260
7220=  260 FORMAT('0ENTER TIME RESTRICTION FOR TRAINING PROGRAM-     ')
7230=      CALL INP(1,0,0,TM),RETURNS(330)
7240=      TNTIME=8.*TM
7250=      IF (NOPT.EQ.2) GO TO 620
7260=C
7270=C NON-MIX TRAINING OPTION.  DETERMINE LIMITATION BASED ON TIME & COST
7280=  265 PRINT 270
7290=  270 FORMAT(' ENTER MAX PER STUDENT COST FOR TRAINING PROGRAM-     ')
7300=      CALL INP(1,0,0,COST),RETURNS(330)
7310=  280 KT=0
7320=      KC=0
7330=      DO 300  J=1,2
7340=      IF (T(J).GT.TNTIME) GO TO 290
7350=      KT=KT+J
7360=  290 IF (C(J).GT.COST) GO TO 300
7370=      KC=KC+J
7380=  300 CONTINUE
7390=      KKT=KT+1
7400=      GO TO (310,340,440,520),KKT
7410=C
7420=C NEITHER SCHOOL NOR OJT MEETS TIME CONSTRAINT
7430=  310 PRINT 320
7440=  320 FORMAT('0TIME CONSTRAINT NOT SATISFIED BY SCHOOL OR OJT'/
7450=     & ' INPUT NEW VALUE FOR TOTAL TIME TO TRAIN-     ')
7460=      CALL INP(1,0,0,TM),RETURNS(330)
7470=      TNTIME=TM*8.
7480=      IF (TNTIME.GT.0.0) GO TO 280
7490=  330 RETURN ABORT
7500=C
7510=C ONLY SCHOOL MEETS TIME CONSTRAINT - CHECK FOR COST CONSTRAINT
7520=  340 IF (KC.EQ.1.OR.KC.EQ.3) GO TO 380
7530=      PRINT 350
7540=  350 FORMAT(' SCHOOL MEETS TIME CONSTRAINT, BUT COST TOO HIGH')
7545=C ALLOW NEW MAX COST SELECTION
7550=      GO TO 265
7560=C
7570=C FORCE TO SCHOOL. IF ALREADY FORCED OJT, SET TO ZERO.]
7580=C IF 'BOTH' FLAG IS SET (MODE IS 3 OR 4), LEAVE IT ALONE.
7590=  380 PRINT 390
7600=  390 FORMAT('0TIME AND COST CONSTRAINTS FORCE ALL STUDENTS TO SCHOOL')
7610=      NST(1)=NTR
7620=      DO 430  I=1,NBLOC
7630=      CALL UNPACK(MAXES(I),INONE)
7640=      M=INONE(6)+1
7650=      GO TO (400,430,410,430,430),M
7660=  400 INONE(6)=1
7670=      GO TO 420
7680=  410 INONE(6)=0
7690=  420 CALL PACK(INONE,MAXES(I))
7700=  430 CONTINUE
7710=      RETURN
7720=C
7730=C ONLY OJT MEETS TIME CONSTRAINT
7735=C CHECK FOR COST CONSTRAINT
7740=  440 IF (KC.GE.2) GO TO 460
7750=      PRINT 450
7760=  450 FORMAT('0OJT MEETS TIME CONSTRAINT, BUT COST TOO HIGH')
7765=C ALLOW NEW MAX COST SELECTION
7770=      GO TO 265
7780=C
```

```
7790=C FORCE TO OJT.  IF ALREADY FORCED TTS, SET TO ZERO.  IF
7800=C 'BOTH' FLAG IS SET, LEAVE IT ALONE.
7810=   460 PRINT 470
7820=   470 FORMAT('0TIME AND COST CONSTRAINTS FORCE ALL STUDENTS TO OJT'
7830=       NST(1)=0
7840=       DO 510  I=1,NBLOC
7850=       CALL UNPACK(MAXES(I),INONE)
7860=       M=INONE(6)
7870=       IF (M-1) 480,490,510
7880=   480 INONE(6)=2
7890=       GO TO 500
7900=   490 INONE(6)=0
7910=   500 CALL PACK(INONE,MAXES(I))
7920=   510 CONTINUE
7930=       RETURN
7940=C
7950=C BOTH MODES MEET TIME CONSTRAINT
7955=C CHECK BOTH AGAINST COST CONSTRAINT
7960=   520 IF (KC.GT.0) GO TO 540
7970=       PRINT 530
7980=   530 FORMAT(' NEITHER SCHOOL NOR OJT MEETS COST CONSTRAINT')
7990=       GO TO 265
8000=C
8010=C BOTH MODES MEET BOTH CONSTRAINTS
8015=   540 GO TO (380,460,545),KC
8016=C
8017=C BOTH MODES MEET BOTH CONSTRAINTS
8018=C COMPUTE TRAINEE SPLIT BETWEEN 2 MODES.
8020=   545 NOPT=0
8050=       XNS=(COST-C(2)*NTR)/(C(1)-C(2))
8060=       IF (XNS.GT.0.) GO TO 550
8070=       XNS=0.
8080=       GO TO 560
8090=   550 IF (IFIX(XNS).GT.NTR) XNS=NTR
8100=C
8110=   560 NST(1)=XNS
8120=       NST(2)=NTR-NST(1)
8130=       PNS=NST(1)*100.0/NTR
8140=       PNO=NST(2)*100.0/NTR
8150=       PRINT 570,NTR,COST,NST,PNS,PNO
8160=   570 FORMAT('0A TRAINING PROGRAM FOR ',I4,' STUDENTS, MAX PER '
8170=      & ,'STUDENT COST OF ',F6.0,':'/10XI8,' STUDENTS TO SCHOOL',9X
8180=      & I8,' STUDENTS TO OJT'/' PERCENT SPLIT: ',F5.1,'% SCHOOL, '
8190=      & ,F5.1,'% OJT '//' YOU CAN ACCEPT THE CURRENT SPLIT,'
8200=      & ,' INPUT A NEW VALUE FOR COST LIMIT'/' AND RECOMPUTE THE'
8210=      & ,', OR INPUT THE SPLIT DIRECTLY.'/' KEEP THE ABOVE SPLIT?    ')
8220=       CALL INP(2,0,0,IANS),RETURNS(330)
8230=       IF (IANS.EQ.1) GO TO 615
8240=C
8250=C RECOMPUTE NEW SPLIT
8260=       PRINT 580
8270=   580 FORMAT('0DO YOU WISH TO ENTER A NEW COST LIMIT AND',
8280=      & ' RECOMPUTE THE SPLIT?    ')
8290=       CALL INP(2,0,0,IANS),RETURNS(330)
8300=       IF (IANS.EQ.0) GO TO 605
8301=       NOPT=0
8302=       GO TO 265
8310=C
8320=C INPUT SPLIT DIRECTLY
8330=   605 PRINT 610
8340=   610 FORMAT(' INPUT PERCENT STUDENTS TO SCHOOL-    ')
8350=       CALL INP(1,0.,100.,SCH),RETURNS(330)
8360=       NST(1)=SCH*.01*NTR
8370=       NST(2)=NTR-NST(1)
8371=C
8372=C IF EITHER MODE HAS NO STUDENTS, FORCE THE MODEL ACCORDINGLY
```

```
8373=  615 IF (NST(1).EQ.0) GO TO 460
8374=      IF (NST(2).EQ.0) GO TO 380
8380=      RETURN
8390=C
8400=C MIX OPTION
8405=C TRAINING MODE SELECTED TO MINIMIZE COST SUBJECT TO TIME CONSTRAINT
8410=  620 NST(1)=NTR
8420=      PRINT 630
8430=  630 FORMAT('0DO YOU WANT THE OPTION TO SEND TASKS WITH',
8440=     & ' SPECIFIC LEVELS'/' OF A PRIMARY CHARACTERISTIC TO',
8450=     & ' EITHER TTS OR OJT ?     ')
8460=      CALL INP(2,0,0,IANS),RETURNS(330)
8470=      IF (IANS.EQ.0) GO TO 710
8480=      PRINT 640,(J,LABEL(1,J),LABEL(2,J),J=1,5)
8490=  640 FORMAT(' ENTER NUMBER INDICATING THE PRIMARY CHARACTERISTIC',
8500=     & /' FOR THE SCHOOL/OJT DECISION:',5(/I2,'=',A10,A2)/' =    ')
8510=      CALL INP(0,1,5,NCHAR),RETURNS(330)
8520=      PRINT 650
8530=  650 FORMAT('0ALL TASKS WITH PRIMARY CHARACTERISTIC VALUES',
8540=     & ' BETWEEN N1 AND N2'/' WILL BE TRAINED BY X:'/
8550=     & ' ENTER 1,2,3,4, OR 5 FOR N1-     ')
8560=      CALL INP(0,1,5,N1),RETURNS(330)
8570=      IF (N1.EQ.5) GO TO 670
8580=      PRINT 660
8590=  660 FORMAT(' ENTER 1,2,3,4 OR 5 FOR N2 SUCH THAT N2>=N1 -     ')
8600=      CALL INP(0,N1,5,N2),RETURNS(330)
8610=      GO TO 680
8620=  670 N2=5
8630=  680 PRINT 690
8640=  690 FORMAT(' ENTER X, WHERE 1=TTS, 2=OJT -     ')
8650=      CALL INP(0,1,2,MFLAG),RETURNS(330)
8660=C
8670=C SET MODE FOR TASKS WHICH MEET ABOVE CRITERIA
8680=      DO 700  I=1,NBLOC
8690=      CALL UNPACK(MAXES(I),INONE)
8700=      M=INONE(6)
8710=      IF (M.GT.0) GO TO 700
8720=      NCH=INONE(NCHAR)
8730=      IF (NCH.GT.N2.OR.NCH.LT.N1) GO TO 700
8740=      INONE(6)=MFLAG
8750=      CALL PACK(INONE,MAXES(I))
8760=  700 CONTINUE
8770=  710 CALL DYPROG(TNTIME),RETURNS(330)
8780=      RETURN
8790=      END
8800=      SUBROUTINE CONV(I,J)
8810=      COMMON/NAMES/DUT(4,6),TSK(4,62),KDUT(6),KTSK(62),NDUT,NTSK
8820=C
8830=C THIS ROUTINE CONVERTS DUTY LITERAL 'I' AND TASK LITERAL 'J'
8840=C TO INDICIES OF ARRAYS IN COMMON/NAMES/
8850=      DO 10  II=1,NDUT
8860=      IF (I.EQ.KDUT(II)) GO TO 20
8870=   10 CONTINUE
8880=      II=1
8890=   20 I=II
8900=C
8910=      DO 30  JJ=1,NTSK
8920=      IF (J.EQ.KTSK(JJ)) GO TO 40
8930=   30 CONTINUE
8940=      JJ=1
8950=   40 J=JJ
8960=      RETURN
8970=      END
8980=      SUBROUTINE PACK(JRAY,J)
8990=      DIMENSION JRAY(10),MULT(8)
9000=      DATA MULT/1,10,100,1000,10000,100000,1000000,10000000/
```

```
9010=C
9020=C THIS ROUTINE PACKS 10 INTEGERS IN JRAY(1) THRU JRAY(10)
9030=C INTO THE LAST 13 DIGITS OF WORD J. 1-8 ARE SINGLE DIGIT INTEGERS,
9040=C 9 IS 2 DIGITS, 10 IS 3 DIGITS.
9050=       J=0
9060=       DO 10  K=1,8
9070=    10 J=J+JRAY(K)*MULT(K)
9080=C
9090=       J=J+JRAY(9)*100000000+JRAY(10)*10000000000
9100=       RETURN
9110=       END
9120=       SUBROUTINE UNPACK(J,JRAY)
9130=       DIMENSION JRAY(10)
9140=C THIS ROUTINE REVERSES 'PACK'
9150=       JTOT=J
9160=       DO 10  K=1,8
9170=       JNEW=JTOT/10
9180=       JRAY(K)=JTOT - (JNEW*10)
9190=    10 JTOT=JNEW
9200=C
9210=       JNEW=JTOT/100
9220=       JRAY(9)=JTOT - (JNEW*100)
9230=       JRAY(10)=JNEW
9240=       RETURN
9250=       END
9260=       SUBROUTINE HALVES(INT,VAL)
9270=       DIMENSION VAL(2)
9280=C
9290=C THIS ROUTINE SPLITS 'INT' INTO 2 8-DIGIT INTEGERS
9300=C AND CONVERTS EACH FROM THOUSANDTHS TO A REAL NUMBER
9310=       I2=INT/100000000
9320=       I1=INT- (I2*100000000)
9330=       VAL(1)=I1*.01
9340=       VAL(2)=I2*.01
9350=       RETURN
9360=       END
9370=       SUBROUTINE INP(JFLAG,MIN,MAX,INT),RETURNS(ABORT)
9380=       EQUIVALENCE(JMIN,AMIN),(JMAX,AMAX),(JINT,AINT)
9390=       DIMENSION CH(80)
9400=       DATA BL,J,COM/' ',0,','/
9410=     1 IF (J.EQ.0) READ(5,5) CH
9420=     5 FORMAT(80A1)
9430=       IF (JFLAG-1) 20,100,500
9440=C
9450=C READ FIRST CHARACTER OF INTEGER. IGNORE BLANK OR ONE PLUS.
9460=C ERROR ON - OR . OR , OR ANY LETTER.
9470=    20 KP=0
9480=    30 J=J+1
9490=       IF (J.GT.80) GO TO 540
9500=       CALL FIND(CH(J),KODE),RETURNS(530)
9510=       IF (KODE.LT.1.OR.KODE.GT.14) GO TO 540
9520=       IF (KODE.LT.11) GO TO 40
9530=       KODE=KODE-10
9540=       GO TO (30,540,540,35),KODE
9550=C
9560=C ALLOW ONLY ONE PLUS SIGN
9570=    35 IF (KP.EQ.1) GO TO 540
9580=       KP=1
9590=       GO TO 30
9600=C
9610=C INITIALIZE NUMBER. THEN GET NEXT DIGIT. BLANK OR COMMA IS DELIMITER.
9620=    40 INT=KODE-1
9630=    50 J=J+1
9640=       IF (J.GT.80) GO TO 540
9650=       CALL FIND(CH(J),KODE),RETURNS(530)
9660=       IF (KODE.LT.1.OR.KODE.GT.13) GO TO 540
```

```
9670=       IF (KODE.LT.11) GO TO 60
9680=       GO TO 70
9690=C
9700=C CONVERT DIGIT.  GET NEXT NUMBER
9710=    60 INT=INT*10 + KODE-1
9720=       GO TO 50
9730=C
9740=C ALL THROUGH.  CHECK RANGE AND ASSURE BLANKS TO END.
9750=    70 IF (MIN.EQ.0.AND.MAX.EQ.0) GO TO 80
9760=       IF (INT.LT.MIN.OR.INT.GT.MAX) GO TO 540
9770=    80 IF (CH(J).EQ.COM) RETURN
9780=    82 J=J+1
9790=    85 IF (J.GT.80) GO TO 90
9800=    86 IF (CH(J).EQ.COM) RETURN
9810=       IF (CH(J).NE.BL) GO TO 540
9820=       GO TO 82
9830=    90 J=0
9840=       RETURN
9850=C
9860=C READ FIRST REAL CHARACTER. , IS ERROR. SET FLAGS FOR OTHERS
9870=   100 KD=0
9880=       KM=0
9890=       KP=0
9900=       FACT=.1
9910=   110 J=J+1
9920=       IF (J.GT.80) GO TO 540
9930=       CALL FIND(CH(J),KODE),RETURNS(530)
9940=       IF (KODE.LT.1.OR.KODE.GT.15) GO TO 540
9950=       IF (KODE.LT.11) GO TO 160
9960=       KODE=KODE-10
9970=       GO TO (150,540,120,140,130),KODE
9980=C
9990=C SET UP FOR FRACTION
10000=   120 IF (KD.EQ.1) GO TO 540
10010=       KD=1
10020=       GO TO 110
10030=C
10040=C SET UP NEGATIVE NUMBER
10050=   130 IF (KD+KP+KM.NE.0) GO TO 540
10060=       KM=1
10070=       GO TO 110
10080=C
10090=C PLUS IS NOT ALLOWED AFTER ANYTHING ELSE
10100=   140 IF (KP+KD+KM.NE.0) GO TO 540
10110=       KP=1
10120=       GO TO 110
10130=C
10140=C NO BLANKS AFTER A DECIMAL
10150=   150 IF(KD) 110,110,540
10160=C
10170=C SET UP FIRST DIGIT
10180=   160 AINT=KODE-1
10190=       IF (KD.EQ.0) GO TO 170
10200=       AINT=AINT*FACT
10210=       FACT=FACT*.1
10220=C
10230=C GET NEXT CHARACTER
10240=   170 J=J+1
10250=       IF (J.GT.80) GO TO 540
10260=       CALL FIND(CH(J),KODE),RETURNS(530)
10270=       IF (KODE.LT.1.OR.KODE.GT.13) GO TO 540
10280=       IF (KODE.LT.11) GO TO 190
10290=       IF (KODE.EQ.13) GO TO 180
10300=       GO TO 300
10310=C
10320=C FIX DECIMAL POINT
```

```
10330=  180 IF (KD.EQ.1) GO TO 540
10340=      KD=1
10350=      GO TO 170
10360=C
10370=C INSERT NEXT NUMBER
10380=  190 IF (KD.EQ.1) GO TO 200
10390=      AINT=AINT*10.0 + KODE-1.0
10400=      GO TO 170
10410=  200 AINT=AINT + (KODE-1)*FACT
10420=      FACT=FACT*.1
10430=      GO TO 170
10440=C
10450=C CHECK RANGE
10460=  300 IF (KM.EQ.1)  AINT=-AINT
10470=      INT=JINT
10480=      JMAX=MAX
10490=      JMIN=MIN
10500=      IF (AMAX.EQ.0.0.AND.AMIN.EQ.0.0) GO TO 80
10510=      IF (AINT.LT.AMIN.OR.AINT.GT.AMAX) GO TO 540
10520=      GO TO 80
10530=C
10540=C CHECK FOR Y OR N
10550=  500 J=J+1
10560=      IF (J.GT.80) GO TO 540
10570=      IF (CH(J).EQ.BL) GO TO 500
10580=      CALL FIND(CH(J),KODE),RETURNS(530)
10590=      J=J+1
10600=      IF (KODE-16) 540,510,520
10610=  510 INT=1
10620=      GO TO 86
10630=  520 INT=0
10640=      GO TO 86
10650=  530 J=0
10660=      RETURN ABORT
10670=C
10680=C PRINT ERROR MESSGAE
10690=  540 PRINT 550
10700=  550 FORMAT(' INPUT ERROR. RE-ENTER.     ')
10710=      J=0
10720=      GO TO 1
10730=      END
10740=      SUBROUTINE FIND(CH,KODE),RETURNS(ABORT)
10750=      DIMENSION TAB(17)
10760=      DATA EX/'X'/
10770=      DATA TAB/'0','1','2','3','4','5','6','7','8','9',
10780=     & ' ',',','.','.','+','-','Y','N'/
10790=C
10800=      IF (CH.EQ.EX) RETURN ABORT
10810=      DO 10  KODE=1,17
10820=      IF (CH.EQ.TAB(KODE)) RETURN
10830=   10 CONTINUE
10840=      KODE=0
10850=      RETURN
10860=      END
10870=      SUBROUTINE HIST(LRO)
10880=C THIS ROUTINE PRODUCES 5 HISTOGRAMS OF 5 COLUMNS EACH
10890=      DIMENSION AA(5),LRO(5,5),XS(5,5)
10900=      COMMON/LABELC/LABEL(2,6)
10910=      DATA BL,AA/' ','1','2','3','4','5'/
10911=C
10912=C FIRST FIND MAX BAR
10913=      N=0
10914=      DO 5  M=1,5
10915=      DO 5  J=1,5
10916=      K=LRO(J,M)
10917=    5 IF (K.GT.N) N=K
```

```
10918=       IF (N.EQ.0) RETURN
10930=C
10940=C FIND REASONABLE LABEL RATE AND LINE FEED RATE (JSTEP)
10950=       JSTEP=1
10960=       IF (N.LT.25) GO TO 10
10970=       JSTEP=N/25
10980=    10 JLAB=JSTEP*5
10990=       JFIRST=JSTEP*25
11000=C
11010=C BLANK OUT PRINT BUFFER XS ZERO OUT DATA ARRAY LRO
11020=       DO 30  M=1,5
11030=       DO 30  J=1,5
11040=    30 XS(J,M)=BL
11110=C
11120=C START AT MAX Y-GRID
11130=       JROW=N
11140=    40 DO 50  J=1,5
11150=       DO 50  M=1,5
11160=C SET PRINT BUFFER TO APPROPRIATE CHARACTER
11170=    50 IF (LRO(J,M).GE.JROW) XS(J,M)=AA(J)
11180=C
11190=C SEE IF IT'S TIME FOR A LABEL
11200=       IF (MOD(JROW,JLAB).EQ.0) GO TO 70
11210=       PRINT 60,XS
11220=    60 FORMAT(6X'I',5(3X5(1XA1)))
11230=       GO TO 90
11240=C
11250=C PRINT LINE WITH LABEL
11260=    70 PRINT 80,JROW,XS
11270=    80 FORMAT(I6,'I',5(3X5(1XA1)))
11280=C FOR THE FIRST LABEL, MAKE SURE NEXT ONES ARE 'EVEN'
11290=    90 IF (JROW.LE.JFIRST) GO TO 100
11300=       JROW=JROW-JSTEP
11310=       IF (JROW.LT.JFIRST) JROW=JFIRST
11320=       GO TO 40
11330=C
11340=C DECREMENT ONE LINE
11350=   100 JROW=JROW-JSTEP
11360=       IF (JROW.GT.0) GO TO 40
11370=C
11380=C PRINT HORIZONTAL LABEL
11390=       PRINT 110,(LABEL(1,J),LABEL(2,J),J=1,5)
11400=   110 FORMAT(5X'0I',65('-')/8XA10,A4,A10,A4,A9,A3,A10,A4,A9,A1///)
11410=       RETURN
11420=       END
11430=       SUBROUTINE DYPROG(TNTIME),RETURNS(ABORT)
11440=       DIMENSION T(2),C(2)
11450=       COMMON/BLOC1/ NBLOC,TRPLAN(350),MAXES(350),JCOSTS(350),
11460=      * JTIMES(350),INONE(10)
11470=       COMMON/SHARE/ ICNS(350),QUOT(350),ISW(350),DUM(1050)
11480=C
11490=C
11500=    10 J=0
11510=       OC=0.0
11520=       OT=0.0
11530=       CSUM=0.0
11540=       TSUM=0.0
11550=       XSUM=0.0
11560=C
11570=C
11580=       DO 80  I=1,NBLOC
11590=       CALL UNPACK(MAXES(I),INONE)
11600=       M=INONE(6)
11610=       CALL HALVES(JTIMES(I),T)
11620=       CALL HALVES(JCOSTS(I),C)
11630=       IF (M.NE.0) GO TO 60
```

```
11640=C
11650=C INCREASE ARRAY OF TASKS TO BE OPTIMIZED
11660=       J=J+1
11670=       ICNS(J)=I
11680=       IF (C(1).GT.C(2)) GO TO 30
11690=C
11700=C TTS CHEAPER.  IF SHORTER, FORCE TTS.  ELSE INITIALIZE TTS
11710=       IF (T(1).LE.T(2)) GO TO 20
11720=       ISW(I)=1
11730=       QUOT(J)=(C(2)-C(1))/(T(1)-T(2))
11740=       XSUM=XSUM+T(2)
11750=       OT=OT+T(1)
11760=       GO TO 80
11770=C
11780=C FORCE TTS ... ITS SHORTER AND CHEAPER.
11790=    20 M=1
11800=       GO TO 50
11810=C
11820=C OJT CHEAPER.  IF SHORTER, FORCE OJT.  ELSE INITIALIZE OJT.
11830=    30 IF (T(2).LE.T(1)) GO TO 40
11840=       ISW(I)=2
11850=       QUOT(J)=(C(1)-C(2))/(T(2)-T(1))
11860=       XSUM=XSUM+T(1)
11870=       OT=OT+T(2)
11880=       GO TO 80
11890=C
11900=C FORCE OJT... ITS SHORTER AND CHEAPER
11910=    40 M=2
11920=    50 J=J-1
11930=       INONE(6)=M
11940=       CALL PACK(INONE,MAXES(I))
11950=C
11960=C ADD UP PRESET TASKS
11970=    60 IF (M.GT.2) M=M-2
11980=       TSUM=TSUM+T(M)
11990=       CSUM=CSUM+C(M)
12000=    80 CONTINUE
12010=C
12020=C ASSURE TIME IS SUFFICIENT
12030=       NBK=J
12040=       TNT=XSUM+TSUM
12045=       TNTTT=TNT/8.0
12050=    90 IF (TNTIME.GE.TNT) GO TO 110
12070=    95 PRINT 100,TNTTT
12080=   100 FORMAT('0MINIMUM TIME REQUIRED IS',F8.2,' DAYS. NEW TIME=    ')
12090=       CALL INP(1,TNTTT,1.E30,TNEW),RETURNS(220)
12100=       TNTIME=TNEW*8.0
12110=       GO TO 10
12120=C
12130=C SORT ON QUOT
12140=   110 IF (NBK-1) 200,145,115
12150=   115 NBKM1=NBK-1
12160=       DO 140  I=1,NBKM1
12170=C
12180=C FIND SMALLEST
12190=       X=1.E30
12200=       DO 120  J=I,NBK
12210=       IF (QUOT(J).GE.X) GO TO 120
12220=       K=J
12230=       X=QUOT(J)
12240=   120 CONTINUE
12250=       IC=ICNS(K)
12260=C
12270=C PLACE IN ITH AND SLIDE DOWN (BACKWARDS)
12280=       IF (K.EQ.I) GO TO 140
12290=       KMI=K-I
```

```
12300=       DO 130  L=1,KMI
12310=       KS=K+1-L
12320=       QUOT(KS)=QUOT(KS-1)
12330=  130  ICNS(KS)=ICNS(KS-1)
12340=C
12350=       QUOT(I)=X
12360=       ICNS(I)=IC
12370=  140  CONTINUE
12380=C
12390=C COMPUTE TIME TO MAKE UP
12400=  145  J=0
12410=  150  TTM=TSUM+OT-TNTIME
12420=       IF (TTM.LE.0.0) GO TO 170
12430=       J=J+1
12440=       IF (J.GT.NBK) STOP
12450=C
12460=C FLIP
12470=       I=ICNS(J)
12480=       MOLD=ISW(I)
12490=       M=3-MOLD
12500=       ISW(I)=M
12510=       CALL HALVES(JTIMES(I),T)
12520=       OT=OT-T(MOLD)+T(M)
12530=       GO TO 150
12540=C
12550=C BACK UP, SWITCHING WHEN POSSIBLE
12560=  170  J=J-1
12570=       IF (J.LE.0) GO TO 180
12580=       I=ICNS(J)
12590=       MOLD=ISW(I)
12600=       M=3-MOLD
12610=       CALL HALVES(JTIMES(I),T)
12620=       DIFF=T(MOLD)-T(M)
12630=       IF (TTM.GT.DIFF) GO TO 170
12640=       ISW(I)=M
12650=       OT=OT-DIFF
12660=       TTM=TTM-DIFF
12670=       GO TO 170
12680=C
12690=C PRINT IT
12700=  180  DO 190  J=1,NBK
12710=       I=ICNS(J)
12720=       M=ISW(I)
12730=       CALL HALVES(JCOSTS(I),C)
12740=  190  OC=OC+C(M)
12750=  200  TOT=(TSUM+OT)/8.0
12760=       COT=CSUM+OC
12770=       PRINT 210,TOT,COT
12780=  210  FORMAT('0TRAINING PLAN   WILL'/3X'OF LENGTH',5X'COST'
12790=      * /F10.1,F11.1/'0IS THAT ACCEPTABLE ?   ')
12800=       CALL INP(2,0,0,IANS),RETURNS(220)
12810=       IF (IANS.EQ.0) GO TO 95
12820=C
12830=       IF (NBK.LE.1) RETURN
12840=       DO 215  J=1,NBK
12850=       I=ICNS(J)
12860=       CALL UNPACK(MAXES(I),INONE)
12870=       INONE(6)=ISW(I)
12880=  215  CALL PACK(INONE,MAXES(I))
12890=       RETURN
12900=  220  RETURN ABORT
13730=       END
13740=       SUBROUTINE METMED(NITER,NLOOP)
13750=       COMMON/BLOC1/ NBLOC,TRPLAN(350),MAXES(350),JCOSTS(350),
13760=      % JTIMES(350),INONE(10)
13770=       COMMON/BLOC2/ MTHMAP(6,2),MEDMAP(6,2),MAPPSY(5),MAPCOG(5)
```

```
13780=C
13790=       DO 10  I=1,NBLOC
13800=       CALL UNPACK(MAXES(I),INONE)
13810=       M=INONE(6)
13820=       IF (M.EQ.0) GO TO 10
13830=       CALL DECIDE(M,NLOOP,MM)
13835=C CHECK FOR PRE-ASSIGNED METHODS AND MEDIA
13840=       IF (INONE(7).GT.0.AND.NITER.EQ.0) GO TO 10
13850=       NT1=INONE(4)
13860=       NT2=INONE(5)
13865=C HIGHER OF COG,PSYCH LEVELS DIRECTS MAPPING INTO TRAINING OBJECTIVE
13870=       NTO=MAPCOG(NT2)
13880=       IF (NT1.GT.NT2) NTO=MAPPSY(NT1)
13885=C MAPPING FROM TRAINING OBJECTIVE TO METHOD AND MEDIUM
13890=       INONE(7)=MTHMAP(NTO,M)
13900=       INONE(8)=MEDMAP(NTO,M)
13910=       CALL PACK(INONE,MAXES(I))
13920=    10 CONTINUE
13930=       RETURN
13940=       END
13950=       SUBROUTINE DECIDE(M,NLOOP,M2)
13960=C THIS ROUTINE CONVERTS M AND M2 TO EITHER 1 OR 2.
13970=C M IS THE TTS/OJT INDEX TO COSTS, TIMES, ETC.
13980=C M2 IS THE ARTIFICIAL MODE WHICH REPRESENTS HOW THE
13990=C TASK IS ACTUALLY TAUGHT.  EVEN IF ALL STUDENTS ARE
14000=C GOING TTS, M2 COULD BE '2' IF M CAME IN '4', WHICH
14010=C MEANS THIS TASK IS TAUGHT TO ALL STUDENTS, WHETHER
14020=C GOING TTS OR OJT, BUT THE TASK IS TAUGHT OJT.
14030=C NLOOP IS 1 OR 2 IF ALL STUDENTS ARE GOING TO TTS (1) OR
14040=C OJT (2).   IF NLOOP IS ZERO, THERE IS A MIX AND M WILL BE
14050=C SET TO M-2 IF THE ABOVE OPTION IS USED.
14060=C M=5 WAS SET WITH NLLOP=1 TO FLAG AS A TTS FORCE. WE SET IT
14070=C TO 1 HERE.  IT WILL NOT BE SET TO 2 LATER (WHEN NLOOP=2)
14080=C AS WILL MOST OTHERS (ALL WHICH ARE NOW M=1).
14090=       IF (M.NE.5) GO TO 10
14100=       M=1
14110=       GO TO 20
14120=    10 IF (M.GT.2) GO TO 30
14130=    20 M2=M
14140=       RETURN
14150=    30 M=M-2
14160=       M2=M
14170=       IF (NLOOP.GT.0) M=NLOOP
14180=       RETURN
14190=       END
14200=       SUBROUTINE SCHEDU(NLOOP,NUMB,MEDTOT,KODE)
14210=       DIMENSION TVAL(2)
14220=       COMMON/BLOC1/NBLOC,TRPLAN(350),MAXES(350),JCOSTS(350),
14230=      & JTIMES(350),INONE(10)
14240=       COMMON/BLOC3/NTR,NST(2),NHIER,NMEDIA,NSCH,MIN,MAX,NCL,NSIZE(30)
14241=C NTR=TOTAL NUMBER OF TRAINEES. NST(I)=NUMBER OF TRAINEES ASSIGNED
14242=C  TO MODE I.  NHIER IS THE IDENTIFYING NUMBER OF THE CHARACTERISTIC
14243=C GOVERNING TRAINING SEQUENCE OF TASK BLOCKS.  NMEDIA IDENTIFIES THE
14244=C HIGH COST MEDIUM TO BE OPTIMALLY SCHEDULED.  NSCH IS THE NUMBER OF
14245=C SCHEDULE ARRANGEMENTS.  MIN AND MAX ARE THE CLASS SIZE LIMITS.  NCL=
14246=C TOTAL NUMBER OF CLASSES NEEDED AND NSIZE(I) IS THE SIZE OF CLASS I.
14250=       COMMON/SHARE/ NSCHED(350,4),JSIM(350),KSIM(350)
14261=C NSCHED IS A TASK BLOCK POINTER FOR EACH OF UP TO 4 SCHEDULES.
14262=C KSIM IS A TASK BLOCK POINTER WHICH IS SORTED TO CONTAIN FIRST
14263=C THE HIGH COST MEDIUM TASKS, THEN OTHER TTS TASKS, AND FINALLY THE
14264=C OJT TASKS.  JSIM CONTAINS THE GOVERNING CHARACTERISTIC VALUE IN
14265=C A PARALLEL ARRAY. BELOW, NSIM, NCHR, AND NOJT ARE THE SIZES OF
14266=C THE RESPECTIVE PORTIONS OF EACH ARRAY.
14270=C
14280=C CREATE ARRAYS FOR ORDERING BLOCKS FOR TRAINING
14290=       NSIM=0
```

```
14300=       NCHR=0
14310=       NOJT=0
14320=       MEDTOT=0
14330=       KODE=1
14340=       STIME=0.0
14341=       OTIME=0.0
14342=       IAS1=0
14343=       IAS2=0
14344=C
14345=C FIRST FIND BIASES
14346=       DO 10  I=1,NBLOC
14347=       CALL UNPACK(MAXES(I),INONE)
14348=       M=INONE(6)
14349=       IF (M.EQ.0.OR.M.EQ.2.OR.M.EQ.4) GO TO 10
14350=       IAS2=IAS2+1
14351=       IF (INONE(7).EQ.NMEDIA)  IAS1=IAS1+1
14352=    10 CONTINUE
14353=C
14360=       DO 40  I=1,NBLOC
14370=       CALL UNPACK(MAXES(I),INONE)
14380=       M=INONE(6)
14390=       IF (M.EQ.0) GO TO 40
14400=       CALL DECIDE(M,NLOOP,M2)
14410=       CALL HALVES(JTIMES(I),TVAL)
14420=       IF (M.EQ.2) GO TO 30
14425=C STIME HOLDS SCHOOL TRAINING TIME
14430=       STIME=STIME+TVAL(1)
14440=       IF (INONE(8).EQ.NMEDIA) GO TO 20
14450=       NCHR=NCHR+1
14460=       JSIM(NCHR+IAS1)=INONE(NHIER)
14470=       KSIM(NCHR+IAS1)=I
14480=       GO TO 40
14490=    20 NSIM=NSIM+1
14500=       JSIM(NSIM)=INONE(NHIER)
14510=       KSIM(NSIM)=I
14520=       GO TO 40
14525=C OTIME HOLDS OJT TRAINING TIME
14530=    30 OTIME=OTIME+TVAL(2)
14540=       NOJT=NOJT+1
14550=       JSIM(NOJT+IAS2)=INONE(NHIER)
14560=       KSIM(NOJT+IAS2)=I
14570=    40 CONTINUE
14580=       IF (NHIER.EQ.0) GO TO 50
14589=C SORT TASK BLOCKS ACCORDING TO SELECTED CHARACTERISTIC
14590=       CALL DBLSOR(JSIM,KSIM,NSIM)
14600=       CALL DBLSOR(JSIM(IAS1+1),KSIM(IAS1+1),NCHR)
14610=       CALL DBLSOR(JSIM(IAS2+1),KSIM(IAS2+1),NOJT)
14620=C
14630=    50 NUMB=NSIM+NCHR+NOJT
14640=       IF (NOJT.EQ.NUMB) GO TO 200
14645=C DETERMINE NUMBER AND SIZE OF CLASSES - NCL AND NSIZE(I)
14650=       KODE=2
14660=       NHOLD=NST(1)
14670=       NCL=NHOLD/MAX
14680=       NREM=NHOLD-(NCL*MAX)
14690=       DO 60  I=1,NCL
14700=    60 NSIZE(I)=MAX
14710=       IF (NREM.GE.MIN) GO TO 80
14720=       IF (NREM.EQ.0) GO TO 90
14730=       K=0
14740=       DO 70  I=1,NREM
14750=       K=K+1
14760=       NSIZE(K)=NSIZE(K)+1
14770=    70 IF (K.EQ.NCL) K=0
14780=       GO TO 90
14790=    80 NCL=NCL+1
```

```
14800=       NSIZE(NCL)=NREM
14810=    90 IF (NSIM.EQ.0) GO TO 180
14820=C
14830=C SCHEDULE TTS BLOCKS, WITH OPT. MEDIUM OPTION
14831=C FORM ORIGINAL SCHEDULE BY ORDERING IAS1 TASK BLOCKS FIRST AND
14832=C THEN IAS2 BLOCKS TO FOLLOW.  KEEP FIRST IAS1 TASKS (HIGH COST
14833=C MEDIUM) TOGETHER AS A BLOCK, AND DETERMINE POSSIBLE ARRANGEMENTS
14834=C OF ORIGINAL SCHEDULE BY SHIFTING THIS BLOCK POSITION IN SEQUENCE.
14835=C CAN'T BEGIN TRAINING THIS BLOCK IN ARRANGEMENT NSCH (AT TIME RHO)
14836=C UNTIL ITS TRAINING IN ARRANGEMENT NSCH-1 IS OVER (AT TIME ETA).
14840=       KODE=3
14850=       TAU=0.
14860=       DO 100  K=1,NSIM
14870=       I=KSIM(K)
14880=       CALL UNPACK(MAXES(I),INONE)
14890=       M=INONE(6)
14900=       CALL DECIDE(M,NLOOP,M2)
14910=       CALL HALVES(JTIMES(I),TVAL)
14920=   100 TAU=TAU+TVAL(M)
14930=       ETA=TAU
14940=C
14950=       K=0
14960=       NSCH=1
14970=       NITEM=1
14980=   110 DO 120  J=1,NSIM
14990=       K=K+1
15000=   120 NSCHED(K,NSCH)=KSIM(J)
15010=       DO 130  J=NITEM,NCHR
15020=       K=K+1
15030=   130 NSCHED(K,NSCH)=KSIM(IAS1+J)
15040=       K=0
15050=       NSCH=NSCH+1
15060=       RHO=0.0
15070=   140 K=K+1
15080=       IF (K.GT.NCHR.OR.NSCH.GT.4.OR.NSCH.GT.NCL) GO TO 150
15090=       I=KSIM(K+IAS1)
15100=       NSCHED(K,NSCH)=I
15110=       CALL UNPACK(MAXES(I),INONE)
15120=       M=INONE(6)
15130=       CALL DECIDE(M,NLOOP,M2)
15140=       CALL HALVES(JTIMES(I),TVAL)
15150=       RHO=RHO+TVAL(M)
15160=       IF (RHO.LT.ETA) GO TO 140
15170=       NITEM=K+1
15180=       ETA=TAU+RHO
15190=       IF (ETA.LT.STIME) GO TO 110
15200=   150 NSCH=NSCH-1
15210=C
15220=C NOW HAVE TOTAL POSSIBLE SCHEDULE ARRANGEMENTS (MAX=4)
15221=C WHICH = NSCH.  I.E., ONE UNIT OF SELECTED MEDIUM CAN BE USED FOR
15222=C NSCH CLASSES.  NOW DETERMINE HOW MANY UNITS NEEDED FOR NCL CLASSES.
15230=       MEDTOT=NCL/NSCH
15240=       IF (NCL.GT.MEDTOT*NSCH) MEDTOT=MEDTOT+1
15250=C ADD OJT BLOCKS TO CLASS SCHEDULES
15260=       NTS=NSIM+NCHR
15270=       IF (NTS.EQ.NUMB) RETURN
15280=       NTS1=NTS+1
15290=       DO 170  J=1,NSCH
15300=       DO 160  K=NTS1,NUMB
15310=       L=K-NTS
15320=   160 NSCHED(K,J)=KSIM(L)
15330=   170 CONTINUE
15340=       RETURN
15350=C
15360=C NO HIGH COST MEDIUM OPTIMIZATION IN SCHEDULES
15361=C ONLY NEED ONE SCHEDULE ARRANGEMENT AND ALL NCL CLASSES
```

```
15362=C FOLLOW IT.  SCHEDULE TTS BLOKS.
15370=  180 NSCH=1
15380=      DO 190  I=1,NCHR
15390=  190 NSCHED(I,1)=KSIM(I+IAS1)
15400=      IF (NCHR.EQ.NUMB) RETURN
15410=      NITEM=NCHR+1
15420=      GO TO 210
15430=C
15440=C SCHEDULE OJT BLOCKS
15450=  200 NSCH=1
15460=      NITEM=1
15470=  210 K=0
15480=      DO 220  I=NITEM,NUMB
15490=      K=K+1
15500=      NSCHED(I,1)=KSIM(K+IAS2)
15510=  220 CONTINUE
15520=      RETURN
15530=      END
15540=      SUBROUTINE DBLSOR(J,K,N)
15550=      DIMENSION J(N),K(N)
15551=C SORTS N VALUES IN ARRAY J IN ASCENDING ORDER.
15552=C ARRAY K IS A POINTER FOR BLOCKS WHOSE VALUES ARE IN ARRAY J
15553=C AND KEEPS TRACK OF RE-ORDERNG.
15560=      IF (N.LE.1) RETURN
15570=      N1=N-1
15580=      DO 30  I1=1,N1
15590=      I1P=I1+1
15600=      DO 20  I2=I1P,N
15610=      IF (J(I1).LE.J(I2)) GO TO 20
15620=      JTEMP=J(I2)
15630=      J(I2)=J(I1)
15640=      J(I1)=TEMP
15650=      KTEMP=K(I2)
15660=      K(I2)=K(I1)
15670=      K(I1)=KTEMP
15680=   20 CONTINUE
15690=   30 CONTINUE
15700=      RETURN
15710=      END
15720=      SUBROUTINE REPT(GRID,NLOOP)
15730=      DIMENSION TVAL(2)
15740=      COMMON/BLOC1/ NBLOC,TRPLAN(350),MAXES(350),JCOSTS(350),
15750=     & JTIMES(350),INONE(10)
15760=      COMMON/NAMES/DUT(4,6),TSK(4,62),KDUT(6),KTSK(62),NDUT,NTSK
15770=      COMMON/MINT/CODE(2)
15780=      DIMENSION AMETH(6),AMED(6)
15790=      DATA AMETH/'LECTR','DSCUS','DEMO','SIMUL','PERFM',' '/
15800=      DATA AMED/'TRANS','FILMS','MOCK','SIMUL','EQUIP',' '/
15810=C
15820=C BEGIN TRAINING PLAN OUTPUT
15830=   10 PRINT 20,GRID
15840=   20 FORMAT(///16XA5,'   TRAINING PLAN GENERATOR OUTPUT')
15850=      OLDSYS=0.
15860=C
15870=C LOOP THROUGH TASKS
15880=      DO 90  I=1,NBLOC
15890=      CALL UNPACK(MAXES(I),INONE)
15900=      I1=INONE(6)
15910=      IF (I1.EQ.0) GO TO 90
15920=      CALL DECIDE(I1,NLOOP,I1M)
15930=      SYS=TRPLAN(I)
15940=      IF (SYS.EQ.OLDSYS) GO TO 50
15950=      IF (OLDSYS.NE.0.) PRINT 30
15960=   30 FORMAT(1X72('-')///)
15970=      PRINT 40,SYS
15980=   40 FORMAT(1X72('-')/' I SUBSYSTEM-',A6,31X'MODE  DAYS METH   MED I')
```

```
15990=       NOLDT=0
16000=       OLDSYS=SYS
16010=C
16020=    50 NEWDT=INONE(9)
16030=       IF (NEWDT.EQ.NOLDT) GO TO 70
16040=       PRINT 60,KDUT(NEWDT),(DUT(J,NEWDT),J=1,4)
16050=    60 FORMAT(' I',70('-'),'I'/' I',1XA2,1X4A10,26X'I')
16060=       NOLDT=NEWDT
16070=C
16080=C
16090=    70 NEWTSK=INONE(10)
16100=       I2=INONE(7)
16110=       I3=INONE(8)
16120=       CALL HALVES(JTIMES(I),TVAL)
16130=       PRINT 80,KTSK(NEWTSK),(TSK(J,NEWTSK),J=1,4),CODE(I1M),
16140=      * TVAL(I1)/8.,AMETH(I2),AMED(I3)
16150=    80 FORMAT(' I',4XA2,'0',1X4A10,1XA3,1XF5.1,1XA5,1XA5,'I')
16160=    90 CONTINUE
16170=       PRINT 30
16180=       RETURN
16190=       END
16191=*EOR
16192=*EOF
16193=*EOF
```

Table B-1

TRAMOD - Subroutine List

TRAMOD	-	reads input tasks data set; directs execution of the model
SCALE	-	maps variable scaled input data into 1-5 scale needed for model; optional
HIST	-	generates histogram of tasks over five task characteristics
SCREEN	-	selects task blocks to be trained according to user criteria
TRMODE	-	assigns training mode for task blocks and student population
METMED	-	assigns methods and media for training task blocks
REPT	-	generates training plan output
SCHEDULE	-	generates training schedule arrangements; determines number of classes and scarce media items for TTS
DBLSORT	-	sorts primary array in ascending order and rearranges elements of corresponding array in parallel
INP FIND	-	read interactive input data and check against acceptable range

Table B-2

TRAMOD - VARIABLE LIST

COGMAP	-	array containing mapping from cognitive levels to training objectives
COSTS	-	array containing TTS, OJT costs for task blocks
GRID	-	trainee group identification
MAX	-	maximum TTS class size
MAXES	-	array containing characteristics for selected task blocks
MEDMAP	-	array containing mapping from training objectives to media
MEDTOT	-	total number of scarce resource media needed for training program
MIN	-	minimum TTS class size
MTHMAP	-	array containing mapping from training objectives to methods
NBLOC	-	number of task blocks to be trained
NCL	-	number of TTS classes needed for training program
NHIER	-	characteristic which governs task block training sequence
NMEDIA	-	scarce resource medium to be optimized
NSCH	-	number of schedule arrangements for training program
NSCHED	-	array containing schedule arrangements for training program
NSIZE	-	size of TTS classes
NST	-	array containing students for TTS and OJT
NTIME	-	length of training program
NTR	-	total number of trainees

Table B-2 (continued)

PSYMAP	-	array containing mapping from psychomotor levels to training objectives
SYS	-	equipment identification associated with tasks
TASKS	-	array containing tasks from input data set
TIMES	-	array containing TTS, OJT times for task blocks
TRPLAN	-	array containing task blocks to be trained

Table B-3

TRAMOD - COMMON STORAGE

BLOC 1

NBLOC	-	integer
TRPLAN (100, 7)	-	real
MAXES (100,7)	-	real

BLOC 2

MTHMAP (6, 2)	-	integer
MEDMAP (6 2)	-	integer
PSYMAP (5)	-	integer
COGMAP (5)	-	integer

BLOC 3

NTR	-	integer
NST (2)	-	integer

BLOC 4

NHIER	-	integer
NMEDIA	-	integer
NSCH	-	integer
NSCHED (100, 10)	-	integer

BLOC 5

MIN	-	integer
MAX	-	integer
NCL	-	integer
NSIZE (30)	-	integer

SUB 1

COSTS (100, 2)	-	real
TIMES (100, 2)	-	real

☆U.S. GOVERNMENT PRINTING OFFICE:1978- 671-056/74