T-1622

SCHEDULING WORK IN

AN

ENGINEERING ORGANIZATION

By

Duane F. Nelson

8426/100 DH

T-1622

A Thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Mathematics).

Signed: Duane F. Nelson
Duane F. Nelson
Student

Golden, Colorado

Date: April 17, 1974

Approved: H. S. Swanson
Dr. Huntington S. Swanson
Thesis Advisor

Robert A. Walsh
Dr. Robert A. Walsh
Head, Mathematics Department

Golden, Colorado

Date: Apr 24, 1974

ABSTRACT

This paper presents an automated method of scheduling work in an engineering organization. The method utilizes a set of heuristic procedures which schedule resources (men) for various activities on a weekly basis. A critical path is initially calculated to determine the total slack for each activity. Activities are sorted according to total slack. Critical activities are examined first. These activities are scheduled at a resource level which will meet the required completion dates if possible. Then, noncritical activities are scheduled with normal resources if available. If previously defined minimum resources are not available for certain activities, their scheduling is postponed. After attempting to schedule all activities that may be started, surplus resources are added to appropriate scheduled activities. Various routines assign several different skills to one activity, require that certain activities be accomplished concurrently, and provide for utilization of secondary skills.

CONTENTS

ILLUSTRATIONS

ACKNOWLEDGEMENTS

The author wishes to express his appreciation to Dr. J. W. Hilf, Chief, Division of Design, Bureau of Reclamation, Engineering and Research Center, Denver, Colorado, for his support and encouragement in the development of a proposed method for scheduling work in an engineering organization. Acknowledgement is due H. W. Anderson, Assistant to Chief, Division of Design, for his guidance and direction in understanding and developing procedures that are consistent with good engineering organizational practices. The assistance of F. A. Clarke, Structural Engineer, Structural and Architectural Branch, in planning and developing the computer programs is especially appreciated.

The author is grateful to Dr. H. S. Swanson, Colorado School of Mines, for his guidance in preparation of the manuscript and to Dr. R.E.D. Woolsey and W. R. Astle for their review.

INTRODUCTION

Efficient planning and control of work in large engineering organizations is complex and difficult. A good work schedule is essential if deadlines are to be met.

Some of the distinguishing factors of an engineering organization are:

1. An essentially stable work force, consisting primarily of professional employees
2. Fluctuations in work loads which at times lead to activities being constrained by limited resources (men)
3. Activities which require several different skills
4. Utilization of more than one skill by many employees
5. The need for some activities to be accomplished concurrently at equal resource levels
6. A requirement for the work schedule to project scheduling information several months into the future
7. Activities of long duration which may be measured in weeks
8. The requirement for coordination by many segments of the organization on certain activities

Oftentimes, scheduling is primarily accomplished by middle managers who use different techniques that vary greatly in quality. These managers are in a position to observe only a portion of the organizational scheduling problem. Top management, being in a position to observe the overall scheduling problem, does not have the time to be concerned with

all of the details required to prepare the best schedule. A computer-assisted approach appears to provide the best method of counteracting this weakness.

Efforts toward using the computer to assist in controlling work were instituted in certain organizations in the late 1950's. Two network methods which have received wide-spread acceptance are PERT (Program Evaluation Review Task) and CPM (Critical Path Method). PERT (1) was originally developed as a management control technique for use on the Fleet Ballistic Missile (Polaris) Program. CPM (2) was developed by E. I. Dupont de Nemours Company and has since received wide-spread usage in the construction industry. PERT and CPM are similar methods, the major difference being in formulation of time estimates for activity durations. PERT uses a three-point probabilistic time estimate, whereas CPM uses a single deterministic estimate. The original models are based on the assumption that unlimited resources are available. Various smoothing techniques have been used with PERT in an attempt to level the resources over the given project duration.

Some of the advantages which may be obtained in using PERT or CPM are:

1. A better definition of a work program and objectives
2. A division of the program into its component parts, illustrating the various activity interactions, yet maintaining a systems view of the program
3. Better discipline in planning and control

4. An orderly method of monitoring and reporting progress of the program
5. A capability of readily identifying critical elements in the program so that quick action may be taken
6. An illustration of the effect of technical and procedural changes in the program
7. Improved cost and schedule predictions
8. An overall view of allocation of resources

Basic PERT and CPM have not been generally acceptable for programs having numerous resource constraints. Other methods of solving the basic scheduling problem have been attempted where resources are limited. Several authors have reviewed numerous proposed solutions (3, 4). Optimization has been the goal of many at these attempts. Linear programing models have been developed. When small numbers of resources are utilized by activities it is desirable to use an integer programing solution. Bounded enumeration and implicit enumeration approaches have also been tried. None of the above methods have proven satisfactory on projects having a large number of activities because of excessive computer run time, large storage requirements, nonavailability of suitable integer programing codes, or lack of flexibility.

A model to schedule work in an engineering organization based on heuristic procedures is described in this thesis. A heuristic is considered as a decision rule which might be used by an organization in its day-to-day operation. It may not lead to an optimal schedule but will

aid in creating a good schedule. Others have developed models which use heuristic procedures (3, 4). Many of the scheduling programs offered by computer vendors are probably based on a heuristic approach. One non-proprietary program is SPAR-1 described by Jerome Weist (5). This program has been used successfully on certain projects, in particular those involving skilled labor.

A heuristic approach allows an approximation of the better organizational procedures, while eliminating the poorer ones. Thus, implementation of a scheduling system is simplified. Modifying some procedures while emphasing the existing good ones, allows for a smooth transition from the present system to a new one.

The method which is presented in this thesis schedules work at one-week periods, first using a critical path program to determine the total slack of each activity with unlimited resources. Next, a set of heuristics are employed to allocate resources within the prescribed limits according to priorities acceptable to an engineering organization.

DEFINITIONS

Activity----------	Smallest division of work (block on network diagram).
Concurrent activities---	Activities which must begin and end on the same dates and must be accomplished at the same resource levels.
Multiskill activities---	Activities requiring more than one skill. Resources representing the various skills must be applied at the same level.
Job----------	Collection of activities represented by lowest level network.
Job network----------	Network representing a job.
Multijob network--------	Any network representing a collection of job networks (two or more).
Master network----------	Collection of all current job networks.
Resource----------	Man possessing a certain skill.
Resource level----------	The percentage of normal crew size at which resources are assigned to an activity.
Skill group----------	A group of men (one or more) possessing a given skill.
Crew----------	The men (one or more) assigned to an activity utilizing a given skill.
Activity team----------	Collection of crews (one or more) required to complete an activity.

Minimum crew size---------- Crew size based on assignment of the lowest feasible level of resources to an activity which will accomplish orderly progress toward completion.

Normal crew size---------- Crew size based on a level of resources which would normally be assigned if resources were plentiful and the completion date was not constraining.

Maximum crew size---------- Crew size based on the highest level of resources which can be assigned to an activity without a significant reduction of efficiency.

Critical crew size---------- Crew size required to complete a critical activity on its due date.

Primary skill---------- The skill which a man utilizes in the majority of assignments.

Secondary skill---------- A skill that may be utilized if men having that skill as a primary skill are not available.

Scheduling period---------- Division of time between scheduling changes (e.g., 1 week).

Scheduling interval---------- Total length of time over which scheduling is accomplished.

Early start time----------	The earliest time at which an activity may begin if the preceding activities are started as early as possible.
Late start time----------	The latest time at which an activity may begin without delaying completion of the job.
Critical path----------	Path through a network, from beginning to terminal node, for which the sum of the activity durations is greater than or equal to that for any other paths (longest path).
Total slack----------	Number of periods an activity may be delayed without affecting any activity on the critical path. The total slack for an activity is equal to its late start time minus its early start time. These times are based on remaining duration for activities which have been active.

METHOD

Introduction

The underlying philosophy in development of the scheduling method was that the system should help line managers schedule their work, without concentrating on all of the minute details involved. To be useful, it is necessary to give information to a manager that shows the relationship between his work and that of other managers who are working on the same job in order that the affect of his actions on others may be foreseen. The scheduling period was chosen as one week, in order to allow flexibility for the managers to schedule the details within the period. The scheduling interval will probably span several months or perhaps years. The schedule most likely will be calculated over a shorter interval, and will be updated frequently.

The procedure which was developed consists of two parts: (1) Calculating the total slack of each activity using a critical path program, and (2) Applying the scheduling heuristics with a scheduling program. Figure 1 gives a diagram of the procedure.

Critical Path Program

The critical path program calculates the total slack for each activity. The standard forward pass-backward pass algorithm is used to determine early start and late start times. The total slack for an activity is equal to the late start time minus the early start time. The information required for these calculations includes activity precedent relations, activity durations, and the jobs to which the

COMMAND PROGRAM: SPREAD

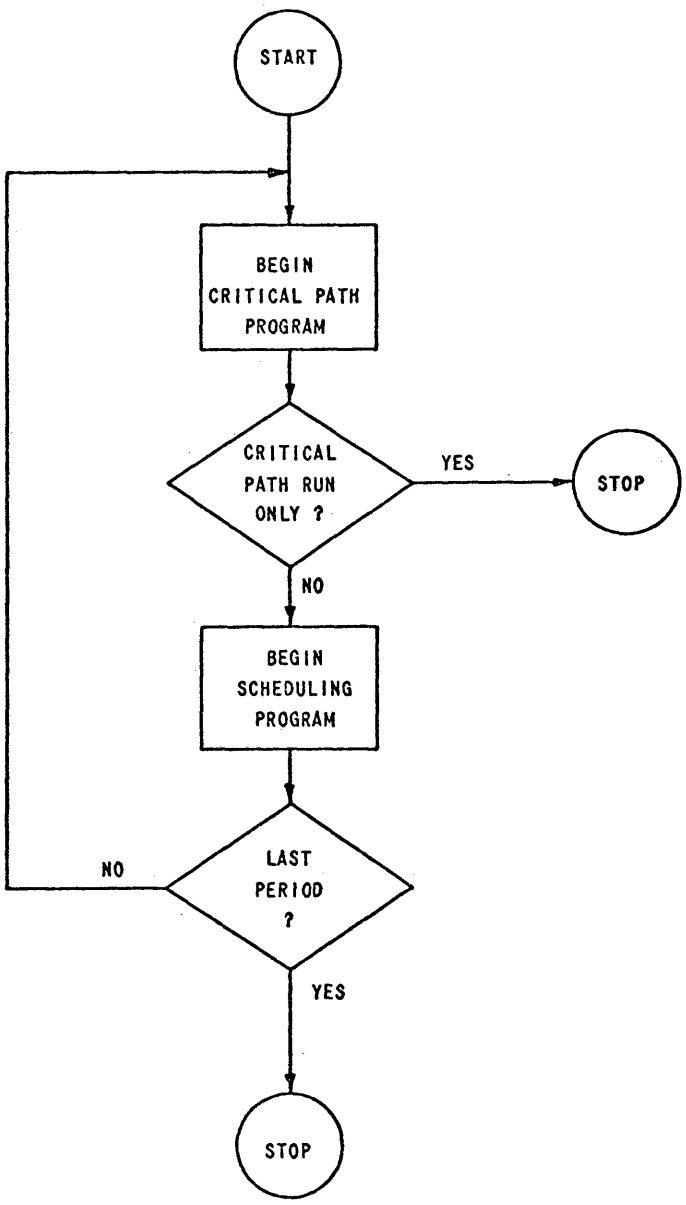


Figure 1

activities belong. The information is received in order of decreasing job priority. Since total slack is the only item of interest, the calculation of other values common for a typical CPM program are not made. The general procedure for accomplishing these calculations has been well documented by many authors (6).

Scheduling Program

The scheduling program uses the output from the critical path program and additional data pertaining to resource availability, required skills, required resource levels, and concurrent activities. The program allocates the available resources for a given period according to a set of heuristic procedures simulating an orderly processing of work. These procedures are described later.

Resource Routines: Two basic routines are used for scheduling and adding resources. The first (LISTM) is a routine for scheduling minimum size crews. Figure 2 shows a flow chart of the routine. Activities are initially scheduled at minimum crew size and augmented later. The second routine (LISTAD) increases crew sizes above that which has already been scheduled by LISTM. LISTAD is used, in different instances, to increase crew sizes from minimum to normal, from minimum to critical, from existing to maximum or any portion of the foregoing, depending on availability of resources. Figure 3 shows the flow chart of LISTAD. The routines schedule resources for each multiskill activity and for concurrent activities at equal levels in order that progress toward completion may progress at a uniform rate.

LISTM

ROUTINE FOR SCHEDULING MINIMUM CREWS

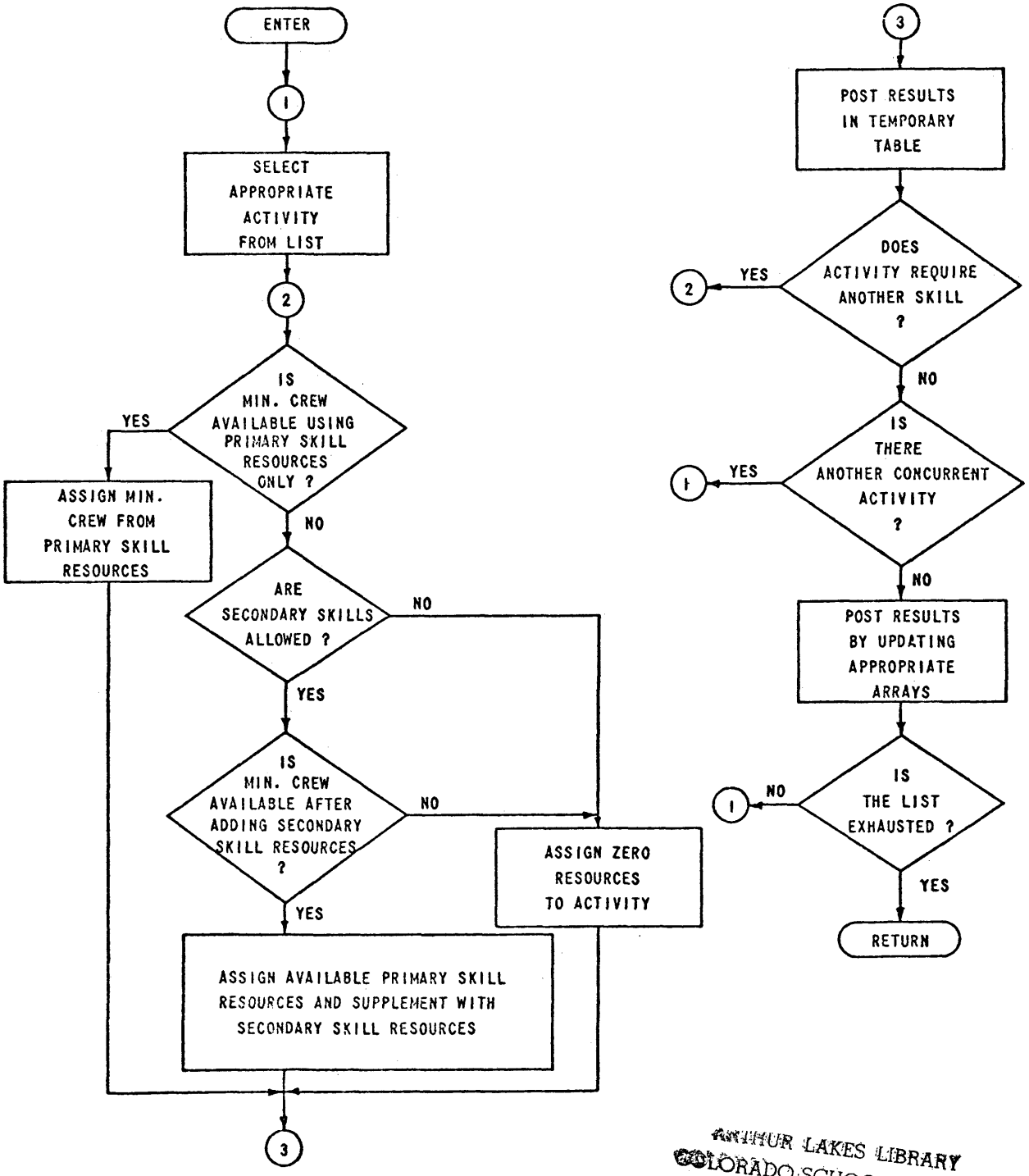


Figure 2

LISTAD

ROUTINE FOR INCREASING CREW SIZES

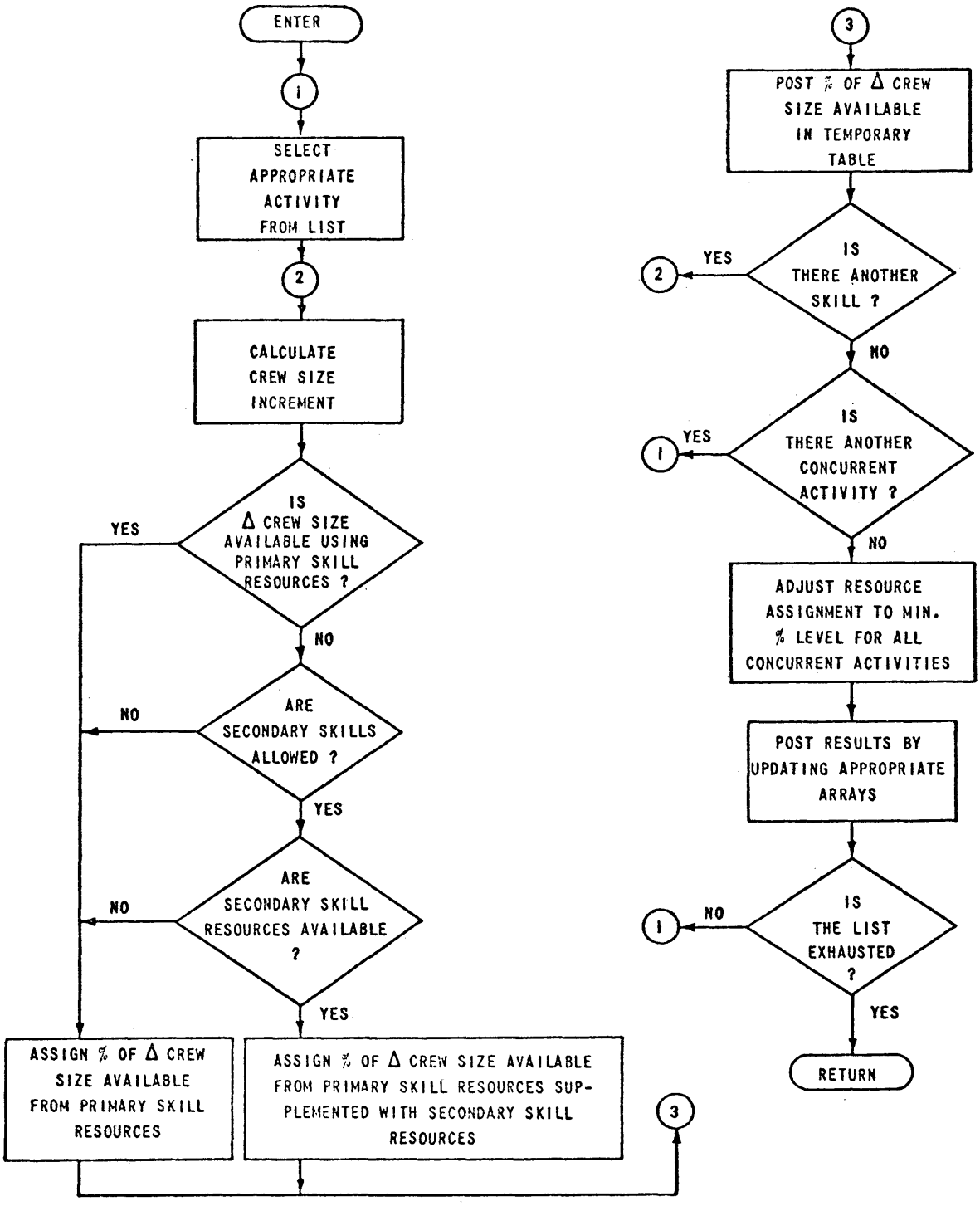


Figure 3

Priority Classification: Activities are segregated into four classes: (1) Critical-active, (2) Critical-nonactive, (3) Noncritical-active, and (4) Noncritical-nonactive. An activity is classified critical if its total slack is less than or equal to some predetermined value (possibly 0). This value may be selected by top management to produce a schedule that will best meet organizational needs. An activity is active if it was scheduled during the last time period. The four classes of activities are treated differently by the scheduling heuristics.

Scheduling Heuristics: The first class of activities, in order of increasing total slack, to be considered by the scheduling program are the critical-active ones. These activities are scheduled at minimum crew size. Secondary skills are utilized if resources possessing equivalent primary skills are not available. Secondary skills are not utilized if they are not possessed by resources or not allowed by management. The critical-nonactive activities are considered next. These activities are scheduled at minimum crew size if resources are available; otherwise, they are not scheduled. Again, secondary skills may be used. At this point, the activities which have been scheduled are merged into one critical-active list and sorted first according to total slack and second according to job priority. Subject to availability, resources are now added to the scheduled activities up to critical crew size, using secondary skills if necessary.

Attention is turned next to the noncritical activities in order of increasing total slack. The active-noncritical activities are scheduled at minimum crew size, using secondary skills if required. Hereafter, secondary skills are not considered. The active-noncritical activities are scheduled up to normal crew size, if possible. Next, scheduling nonactive-noncritical activities at minimum crew size is attempted. The crew size for these activities are increased up to normal crew size if resources are available. Then, scheduled noncritical activities are merged into one active-noncritical list and sorted first according to total slack and second according to job priority.

All possible activities have now been scheduled. The critical activities have been scheduled at critical crew size, and the noncritical activities have been scheduled at normal crew size if sufficient resources were available. If minimum resources were not available for an activity, it has not been scheduled for the present time period.

Although resources possessing certain skills may be scarce at a given time, resources possessing other skills may be plentiful. Thus, it is likely that all resources have not been assigned. If this is true, the surplus resources may be added to the scheduled activities. First, the remaining resources are assigned to critical activities, if possible, in order to increase the resource level up to maximum crew size. Then, if resources remain available, they are assigned to noncritical activities to increase their resource level up to maximum crew

size. The activity priorities for assigning surplus resources remain unchanged from those previously established.

All resources which may be used have been assigned. The schedule has been completed for the present time period. The next step is to return to the critical path program and repeat the above sequence of calculations for the following time period.

COMPUTER PROGRAM INFORMATION

General Layout

The computer programs were written for Computer Sciences Corporation INFONET system. This system enabled the total program package to be separated into two complete programs without an appreciable increase in computer-run time. The two programs, the critical path program (CPATH) and the scheduling program (SCHEDL) may be loaded one at a time as each is used, thereby greatly increasing the potential network size without resorting to the usual overlaying procedures. Command strings of coding (SPREAD) load and initiate execution of the programs in proper sequence. Either time sharing or batch modes may be used. More detail may be found in the Appendix.

Calculations

Integer arithmetic is used almost exclusively by the programs. Thus, roundoff of quantities occurs at each period. It was believed that this would not significantly affect the results and would be consistent with the reliability of the input data.

Input/Output

It is assumed that other programs will be developed to prepare input and generate output. These programs should be especially written to satisfy the organizational requirements and utilize the data processing equipment that is available. For example, the input system might consist of a program that uses card input to prepare files containing

the bulk of the scheduling information which in turn could be stored by a data base management system. Other programs using time sharing terminals could update specific information pertaining to local organizational levels and also initiate the scheduling run. The input system could also utilize an electromagnetic plotter to draw precedence diagrams. A CRT (cathode ray tube) terminal could be used for viewing portions of the precedence diagrams for correctness and initiating changes in an interactive mode.

The output system could use the files generated by the scheduling and critical path programs to prepare reports and charts to satisfy the requirements at various organizational levels. An electromagnetic plotter or CRT printer equipment would be useful for output in addition to computer printer listings.

SAMPLE PROBLEM

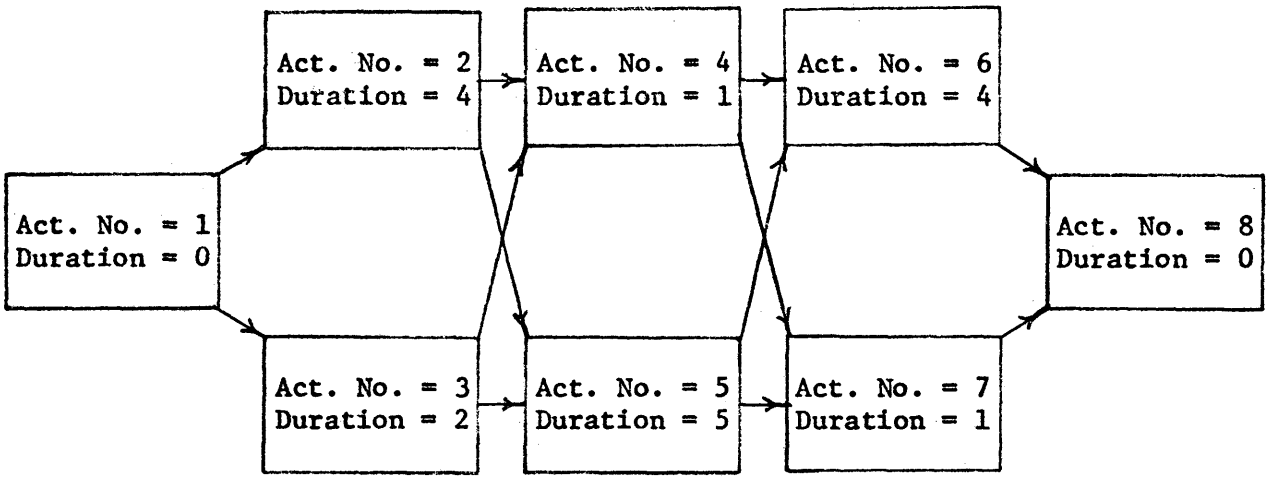
Description

This sample problem is based on 2 jobs, each having 8 activities. Figure 4 presents the precedence diagrams of the 2 jobs. Three of the activities have 0 duration. This feature was included so that the precedence diagrams could always start and finish with one activity if desired. Activity durations are based on the use of normal resources, without any constraints due to competing activities. The first job is due in 14 periods, and the second job is due in 12 periods. There are 5 activities, one group of 3 and another group of 2, which must be performed concurrently.

Table 1 gives resource information for the various activities. Activities 2 and 6 in Job 2 require multiskills. Skill Group 1 possesses a secondary Skill 6. The variations among maximum, normal, and minimum crew sizes are small, and have little affect on activity durations under the resource constraints imposed on competing activities.

The resource availability for each skill group, over the scheduling interval, is shown in Table 2. It is necessary to prepare this table for the total number of periods in the interval if a complete schedule is desired. The interval consists of 17 periods. In an actual application of the program, the scheduling interval would probably be much shorter than the job durations.

JOB 1: DUE DATE = 14



JOB 2: DUE DATE = 12

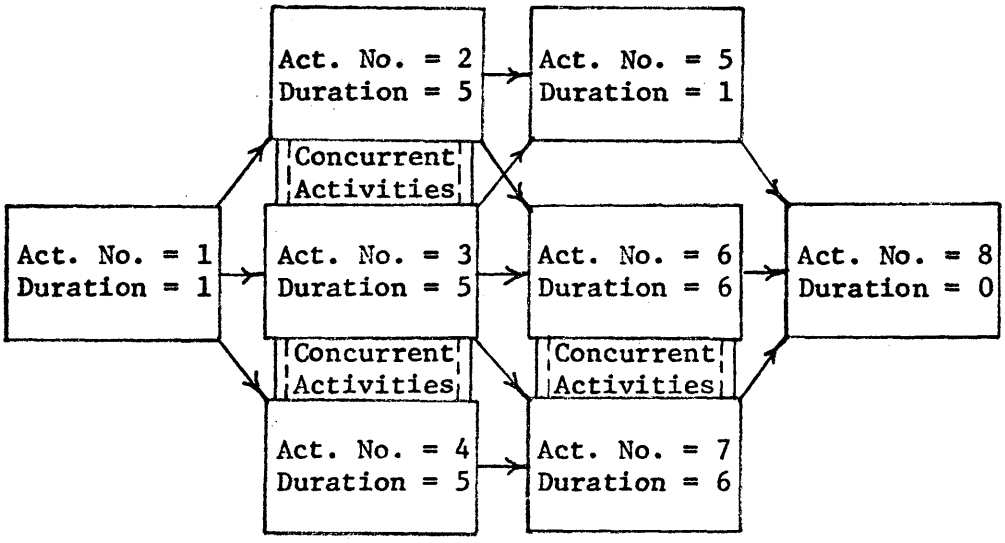


Figure 4. Precedence Diagram

Job	Activity	Primary Skills	Secondary Skills	Max. Crew	Normal Crew	Min. Crew
1	1	-	-	-	-	-
	2	1	6	8	6	4
	3	2	-	6	4	3
	4	6	-	5	3	2
	5	3	-	6	4	2
	6	4	-	3	2	2
	7	5	-	10	7	5
	8	-	-	-	-	-
2	1	2	-	10	8	5
	2	3, 1	-, 6	4, 2	4, 2	3, 1
	3	4	-	4	3	2
	4	5	-	4	2	2
	5	6	-	9	6	4
	6	1, 3, 5	6, -, -	10, 6, 9	7, 4, 6	4, 2, 3
	7	2	-	6	4	4
	8	-	-	-	-	-

Table 1. Activity Resource Information

Skill Group No.	No. of Men Available for Following Period																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	4	4	4	3	4	5	5	5	5	5	5	9	9	9	9	9	9
2	5	5	5	4	4	5	5	5	5	5	5	9	9	9	9	9	9
3	4	4	5	5	5	5	5	5	5	5	5	9	9	9	9	9	9
4	4	5	4	4	5	5	5	5	5	5	5	9	9	9	9	9	9
5	4	5	3	4	5	5	5	5	5	5	5	9	9	9	9	9	9
6	3	5	3	3	5	5	5	5	5	5	5	9	9	9	9	9	9

Skill Group No.	No. of Men Assigned for Following Period																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	4	4	4	3	2	2	0	0	0	0	5	9	9	9	9	9	0
2	5	5	5	0	0	0	0	0	0	0	5	6	6	6	6	6	0
3	0	4	4	4	4	4	5	5	5	5	9	6	6	6	6	6	0
4	0	3	3	3	3	3	0	0	0	0	0	3	3	3	3	0	0
5	0	2	2	2	2	2	0	0	0	0	7	9	9	9	9	9	9
6	0	2	2	3	5	0	5	0	0	0	3	1	1	1	1	1	0

Table 2. Resource Tables

Solution

The resource assignments for each skill group over the scheduling interval are indicated in Table 2. It may be noted that resources were abundant during certain periods, and that the resource availability is not realistic for an actual scheduling problem.

Table 3 shows the slack in each activity over the scheduling interval. Job 1 has been extended 3 periods past its due date and Job 2 has been extended 4 periods past its due date. Activities 5, 6, 7, and 8 of Job 1 have negative slack. Activities 6, 7, and 8 of Job 2 have negative slack. The negative slack in both activities numbered 8 is unimportant, since these activities are artificial and have 0 duration. The column corresponding to Period 1 shows that a run, based on the critical path method only, indicates minimums of 1 period of slack in Job 1 and 0 periods of slack in Job 2. The critical path may be identified by the 0 values.

The activity schedule in Figure 5 shows the relationship of the progress toward completion of the various activities. The concurrent activities in Job 2 are easily identified, since they are active during the same periods. Job 1 was delayed 3 periods, and Job 2 was delayed 4 periods in meeting the due dates. The activities which were competing for resources and thus delayed because of lower priorities may be identified by comparing Figure 5 with Figure 6, which presents an activity schedule that is based on availability of normal resources.

Job	Activity	Period																
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1																	
	2	1	1	1	1													
	3	3	2	2														
	4	5	5	5	5	5												
	5	1	1	1	1	1	0	-1	-1	-1	-1	-1						
	6	1	1	1	1	1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1		
	7	4	4	4	4	4	3	2	2	2	2	2	2	1	0	-1	-2	-3
	8	1	1	1	1	1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-2	-3
2	1	0																
	2	0	0	0	0	0	0											
	3	0	0	0	0	0	0											
	4	0	0	0	0	0	0											
	5	5	5	5	5	5	5	5										
	6	0	0	0	0	0	0	0	-1	-2	-3	-4	-4	-4	-4	-4	-4	
	7	0	0	0	0	0	0	0	-1	-2	-3	-4	-4	-4	-4	-4	-4	
	8	0	0	0	0	0	0	0	-1	-2	-3	-4	-4	-4	-4	-4	-4	

Table 3. Total Slack

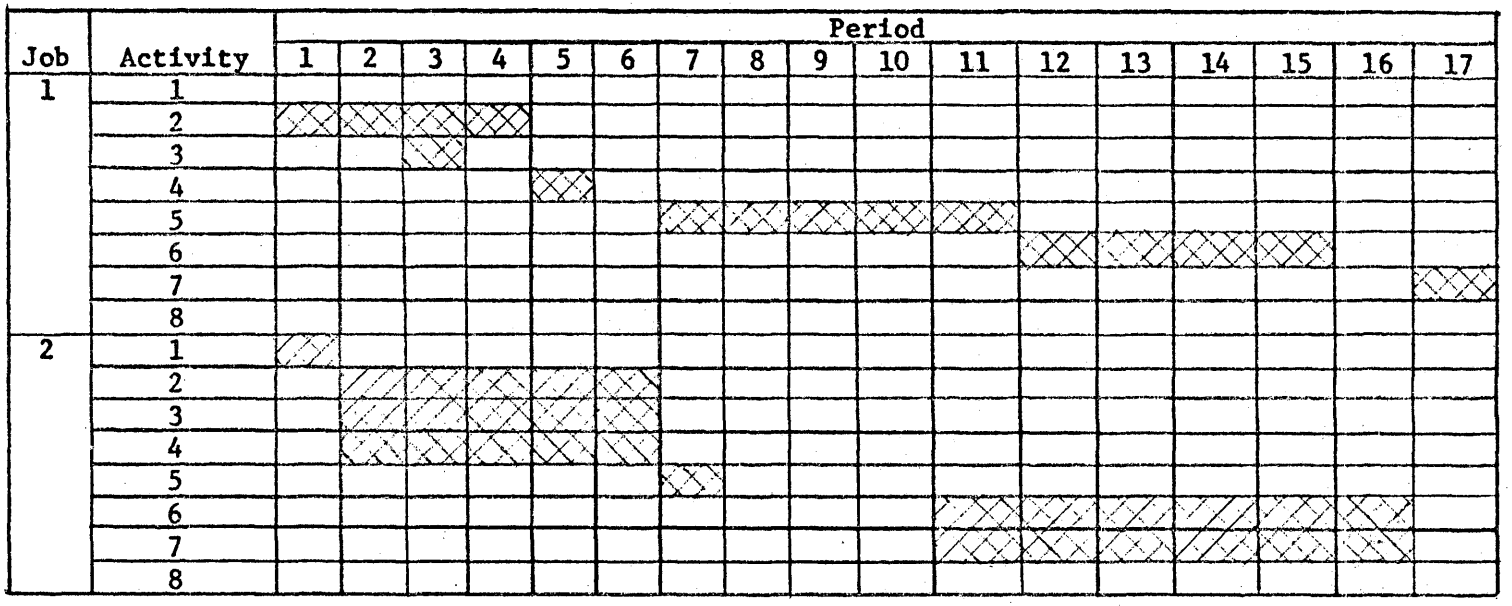


Figure 5. Activity Schedule for Limited Resources

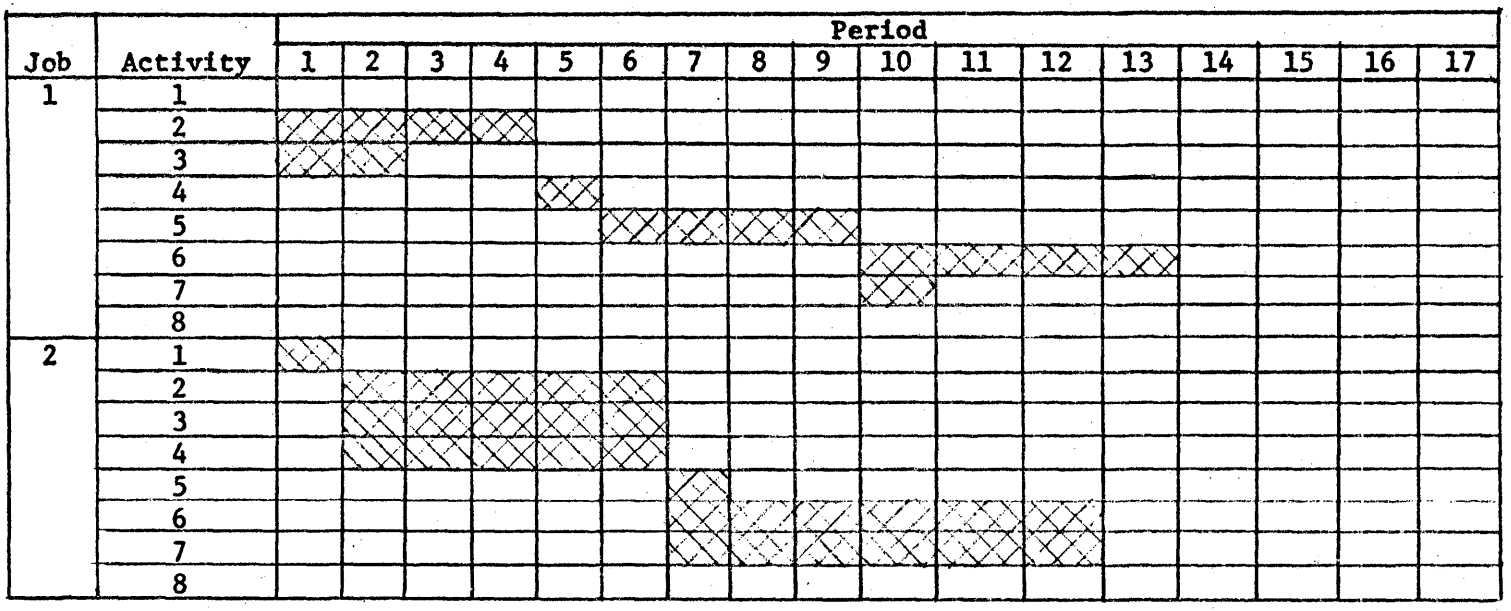


Figure 6. Activity Schedule for Unlimited Resources

Other reports and charts, that illustrate the computer output information in different relationships and detail, should be made available to satisfy a particular organization's needs. A report giving more detailed information on activities competing for particular resources or a report showing activity completion percentages are examples.

It is apparent that only a few skills at certain times were required by more than one activity in quantities of such magnitude that all of the competing activities could not be scheduled. Several approaches might be taken by an organization to relieve this situation. Hiring new men, training employees in secondary skills, transferring employees permanently from one skill group to another, or working overtime are possible remedies. If there are no remedies available, realistic job completion dates will be known in advance, which may tend to alleviate a crisis at a later date. It will also be known that work may proceed in an efficient-orderly manner.

CONCLUSIONS

A heuristic method of scheduling work in an engineering organization has been presented. Computer programs have been developed to apply the method to practical situations. Scheduling in real organizations has not been attempted at this time; however, a sample problem has been included.

The programs were written with a particular organization in mind. Implementation in any organization might suggest modifications in the heuristics to best serve the organization's needs. The scheduling program was structured to facilitate modifications regarding treatment of priority classifications and use of secondary skills. Other modifications which might be considered are: (1) Weighing rather than ranking in establishing job priority, (2) Imposing limits on the number and length of activity schedule interruptions, (3) Accounting for the effect of holidays on the work schedule, (4) Executing the critical path program at intervals greater than one period, should the computer time become excessive, and (5) Sorting on late finish or early start times rather than total slack.

Serious thought should be given to the implementation of any scheduling program in an engineering organization. Cooperation and understanding at all levels in the managerial hierarchy is essential. The line supervisor is particularly important. Without tactful enlistment of his support, inaccurate input data for the scheduling

system may be a consequence. The scheduling system should be proposed as an aid to help each manager schedule the work of his subordinates, and not as an automated system that will direct him "how to run his shop." The success of the system will depend on an intensive training program and a smooth transition from present methods to automated scheduling.

APPENDIX

The appendix contains details directly related to the computer programs. The following topics are included:

1. General Information
2. Program Layout
3. Data File Descriptions
4. List of Routines
5. Subroutine Calling Sequence
6. Selected Variable Meanings
7. Arrays Used for Scheduling
8. Steps in Program SCHEDL
9. Listing of SPREAD
10. Listing of CPATH
11. Listing of SCHEDL
12. Sample Listing of Input Data Files

General Information

1. Three activity numbering systems are used by the program. The first system is used by CPATH and is required for DAT15 data. The activities of each job are numbered consecutively beginning with the number 1, and are called "job identification numbers." The second system is used by SCHEDL and is required for DAT9 and DAT10 data. All of the activities of the master network are numbered consecutively beginning with the number 1 and in order of the priority of the jobs comprising the network. These numbers are called "Master network identification numbers."

The numbers in the third system are called "serial numbers." Serial numbers are assigned each period by the scheduling routine for schedulable activities. They appear in the output of the scheduling lists.

2. Each group of concurrent activities must be numbered consecutively.

3. Within each group of concurrent activities resources may not be assigned from any given skill group to more than one activity. Also, the use of secondary skills is not allowed if the corresponding primary skills are required.

Program Layout

The scheduling program, SPREAD, consists of two command program strings; a critical path routine, named CPATH; and a scheduling routine, named SCHEDL. The command program strings are presently located on files SPREAD and SP2. The critical path routine is on files CPATH, PODYN1, and INITJB. The scheduling routine is on files SCHEDL, SUB1, SUB2, SUB3, and SUB4. Both CPATH and SCHEDL are complete programs in the sense that they are loaded and executed individually one at a time. CALL OBEY statements in each program initiates the loading of the other program. The two command strings contain control statements required to initiate and terminate the execution of the program. The command strings will need to be revised for adaption to the total scheduling system when it becomes operational. If only the critical path routine is run (NPER = 0), the program is terminated through a CALL OBEY statement in CPATH.

It is assumed that the relocatable versions of the files will be saved and the executable versions will be dropped. CPATH-XQT is dropped by the program if scheduling is accomplished, however SCHEDL-XQT is not dropped and must be done externally. If only the critical path routine is run, CPATH-XQT must be dropped externally.

DAT23 and DAT24 were used to restore DAT13 and DAT14 with the original data (See SP2). This feature should eventually be removed.

Data Files

DAT8, DAT9, DAT10, DAT11, DAT13, DAT14, and DAT15 are input files.

DAT8 - Activity information for master network according to ascending order of identification numbers which are assigned by the program. A record consists of the following: ACTIV(I,2), ACTIV(I,3), ACTIV(I,4), ACTIV(I,5), ACTIV(I,6), ACTIV(I,13).

DAT9 - Multiskill information for master network according to ascending order of identification numbers. A record consists of the following: ACTIVITY ID,MSKILL(I,1), MSKILL(I,2).

DAT10 - Concurrent activity information for master network according to ascending order of identification numbers. The smallest number of the group appears as the first item of the two item record. An example follows:

5,6

5,7

5,8

21,22

21,23

DAT11 - Skill group information. There is one item per record. The file consists of RPØØL(I,2) items followed by RPØØL(I,1) items for successive scheduling periods.

DAT13 - General information. The file consists of one 7-item record as follows:

(1) Scheduling interval

(2) Number of skill groups

(3) Number of records on DAT11

(4) Critical slack value

(5) 0

(6) 0

(7) 0

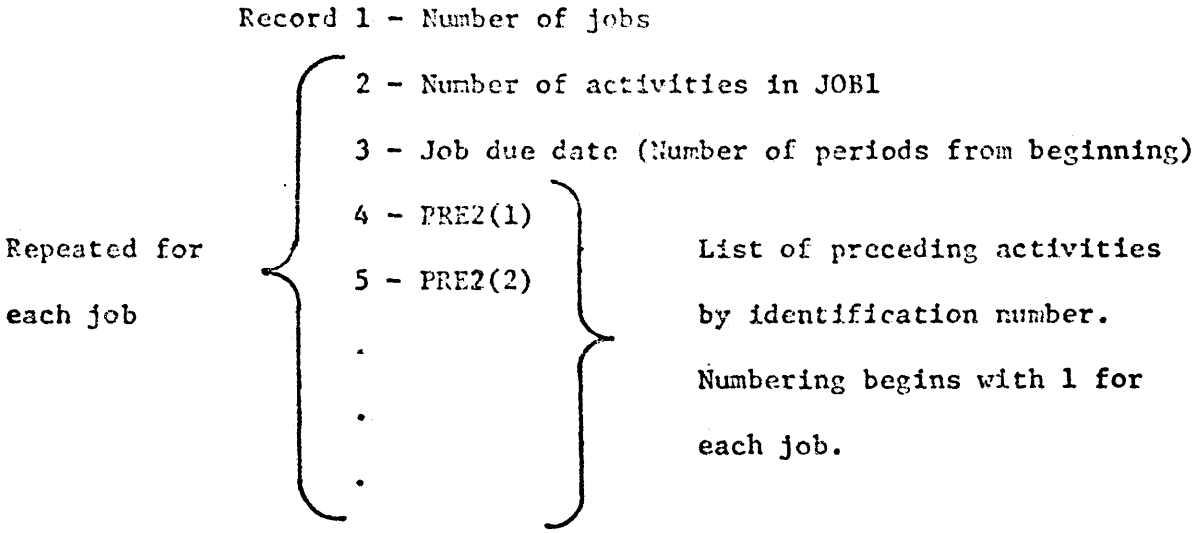
DAT14 - Network description information for critical path program according to ascending order of job identification numbers. Each 3-item record consists of the following:

(1) Number of precedent activities

(2) Location of first precedent activity in PRE2(I) (See DAT15)

(3) Duration of activity.

DAT15 - Network description information. The file consists of one item per record. An example for one job follows. Similar information (except for Item 1) for other jobs would directly follow according to job priority.



Files DAT7, 12 and 16 are created by the programs exclusively for transfer of data between CPATH and SCHEDL.

DAT7 - Critical path program information. A record consists of the following for a given activity:

(1) Slack

(2) Early start

(3) Number of periods remaining for completion.

12 - Activity information. A record consists of the following:

ACTIV(I,1), ACTIV(I,7), ACTIV(I,10)

16 - This file is used for updating other files

File 6 is the output file.

6(RESLT) - This file contains the output for each scheduling period as follows:

(1) Critical path routine output-slack, early start and remaining duration

(2) ACTIV array activities which are scheduable for the given period.

(3) LISTAC array

(4) LISTA array

(5) RPØØL array

LIST OF ROUTINES

1.	SPREAD	Initiates execution of the scheduling program.
2.	SP2	Terminates execution of the scheduling program.
3.	CPATH	Accomplishes critical path analysis of master network.
4.	PODYN1	Accomplishes dynamic storage for CPATH.
5.	ININFØ	Initializes location arrays KPINF1, KPINF2, and KPINF3.
6.	INITPR	Initializes location arrays for PRE1 and PRE2.
7.	CØMIN	Stores data dynamically.
8.	CØMØUT	Retrieves dynamically stored data.
9.	INITJB	Initializes dynamic storage for CPATH.
10.	SCHEDL	Accomplishes scheduling of activities.
11.	REPØØL	Updates RPØØL array for current scheduling period.
12.	RACTIV	Reads ACTIV array data from previous scheduling period.
13.	UPDATL	Adds data for new activities to ACTIV array.
14.	SØRT	Sorts ACTIV array according to ascending order of identification numbers.
15.	RDFIL	Adds additional data to ACTIV array.
16.	PLIST	- Places activities in appropriate scheduling lists.
17.	MULTIS	Sets up MSKILL array.
18.	CØNAC	- Sets up CØNACT array.
19.	CREWSZ	Adjusts maximum and minimum crew sizes for concurrent activities.
20.	UPSTAT	- Updates status of activities Column 10 in ACTIV array.
21.	UPDUR	Updates activity durations.
22.	RESULT	Writes scheduling information on output file.
23.	PACK	Deletes completed activities and packs remaining data in ACTIV array.
24.	LISTM	Schedules a list of activities at minimum crew size.
25.	MERGE	Merges scheduled activities from two lists into one list.
26.	LISTAD	Adds resources to activities in a list.
27.	SSKIL	Determines if resources are available by using secondary skills.
28.	TPØST	Posts data in TEMPRT array.
29.	PØST	Posts resources and adjusts resource pool.
30.	LPACK	Deletes nonschedulable activities and packs remaining activities in a list.
31.	DELCS	Calculates crew size increment for adding resources.
32.	PERAV	Finds percent of resources available without using secondary skills.
33.	SSKILA	Adds resources by utilizing secondary skills.
34.	ADJUST	Adjusts resources to same level in TEMPRT.
35.	SØRTL	Sorts scheduling list according to slack.
36.	SØRTCA	Sorts CØNACT array according to slack.
37.	SØRTEK	Sorts CØNACT array according to serial numbers.

SUBROUTINE CALLING SEQUENCE

I. CPATH: Critical path program

A. PODYNl

B. ININFØ

C. INITPR

D. INITJB

E. CØMIN (called several times)

F. CØMØUT (called several times)

G. OBEY (loads SCHEDL)

II. SCHEDL: Scheduling program

A. REPØØL

B. RACTIV

C. UPDATL

D. SØRT

E. RDFIL

F. PLIST

1. SØRTL
2. SØRTL
3. SØRTL
4. SØRTL

G. MULTIS

H. CØNAC

1. SØRTCA

I CREWSZ

J. LISTM

1. SSKIL
2. TPOST
3. PØST
4. LPACK

K. LISTM

1. SSKIL
2. TPØST
3. PØST
4. LPACK

L. MERGE

M. LISTAD

1. DELCS
2. PERAV
3. SSKILA
4. TPØST
5. ADJUST
6. PØST

N. LISTM

1. SSKIL
2. TPØST
3. PØST
4. LPACK

Ø. LISTAD

1. DELCS
2. PERAV
3. SSKILA
4. TPØST
5. ADJUST
6. PØST

P. LISTM

1. SSKIL
2. TPØST
3. PØST
4. LPACK

Q. LISTAD

1. DELCS
2. PERAV
3. SSKILA
4. TPØST
5. ADJUST
6. PØST

R. MERGE

S. LISTAD

1. DELCS
2. PERAV
3. SSKILA
4. TPØST
5. ADJUST
6. PØST

T. LISTAD

1. DELCS
2. PERAV
3. SSKILA
4. TPØST
5. ADJUST
6. PØST

U. UPSTAT

V. UPDUR

1. SØRTBK

W. RESULT

X. PACK

Y. ØBEY (loads CPATH)

Z. ØBEY (transfers control to SP2)

Selected Variable Meanings

ICA	Number of items in CØNACT array
IDUR	Activity time remaining
IMS	Number of items in MSKILL array
IPER	Present period
IPØST	Number of items in TEMPRT array
IS	Serial number of activity
ISG	Skill group number
ISSKIL	Secondary skill group number
JØBT	Planned job completion date (period)
KPINF1	Total slack
KPINF2	Early start
KPINF3	Time remaining to complete activity
KS	1: (Normal minimum) crew size increment
	- 2: (Critical minimum) crew size increment
	- 3: (Maximum posted) crew size increment
KSLACK	Number of periods of slack for which activities are considered critical
MSSKIL	Number of men assigned with secondary skill
NAS	Number of scheduable activities (number of items in ACTIV array)
NAN	Number of activities in master network
NCA	Number of items in concurrent activity file
NDUR	Change in activity time remaining
NJT	Total number of jobs
NL1	Number of items in LISTAC array
NL2	- Number of items in LISTBC array
NL3	Number of items in LISTA array
NL4	Number of items in LISTB array
NLIST	Number of items in a particular scheduling list
NPER	Number of periods in the scheduling interval
NSG	Number of skill groups
NSSKIL	0: No secondary skills allowed
	1: Secondary skills allowed
PRE1 (1)	Number of precedence relations
" (2)	Location of first precedence relation in preactivity list
" (3)	Duration
" (4)	Early start
" (5)	Late start, replaced later by total slack
PRE2	Preactivity identification number

ARRAYS USED FOR SCHEDULING

ACTIV (I, J)

I = SERIAL NO.

J

I	1	2	3	4	5	6	7	8	9	10	11	12	13
1	MASTER NETWORK IDENTIFICATION NO. OF ACTIVITY (IN ASCENDING ORDER)	CONTROLLING SKILL GROUP	MIN. CREW (NO. OF MEN)	NORMAL CREW (NO. OF MEN)	MAX. CREW (NO. OF MEN)	OVERTIME (HR./WK.)	ACTIVITY TIME REMAINING	NO. OF MEN CURRENTLY ASSIGNED	NO. OF MEN WITH SECONDARY SKILL	STATUS INDICATOR (1) CRITICAL-ACTIVE (2) CRITICAL-NONACTIVE (3) NONCRITICAL-ACTIVE (4) NONCRITICAL-NONACTIVE	TOTAL SLACK	LOCATION OF FIRST APPROPRIATE ITEM IN MSKILL	NO. OF ITEMS IN MSKILL
2													
3													
NAS													

RPOOL (I, J)

I = SKILL GROUP NO.

J

I	1	2
1	NO. OF MEN NOT ASSIGNED	SECONDARY SKILL GROUP WHICH MAY BE UTILIZED
2		
3		
NSG		

MSKILL (I, J)

I

J

I	1	2	3	4
1	SKILL GROUP NO.	MAN POWER RATIO (%) TO CONTROLLING SKILL	NO. OF MEN CURRENTLY ASSIGNED	NO. OF MEN WITH SECONDARY SKILL
2				
3				
IMS				

CONACT (I, J)

I

J

I	1	2
1	SERIAL NO. OF ACTIVITY (IN SCHEDULING ORDER)	SERIAL NO. OF OTHER ACTIVITY
2		
3		
ICA		

LISTAC (I); LISTBC (I)
LISTA (I); LISTB (I)

I

I	1
1	SERIAL NO. OF ACTIVITY
2	
3	
NL1; NL2; NL3; NL4	

TEMPRT (I, J)

USE 1: POSTED FROM LISTM

I

I	1	2	3	4	5
1	SERIAL NO. OF ACTIVITY	SKILL GROUP NO.	NO. OF MEN WITH SPECIFIED SKILL	SECONDARY SKILL GROUP NO.	NO. OF MEN ASSIGNED WITH SECONDARY SKILL
2					
3					
IPOST					

TEMPRT (I, J)

USE 2: POSTED FROM LISTAD

I

I	1	2	3	4	5
1	SERIAL NO. OF ACTIVITY	SKILL GROUP NO.	Δ CREW SIZE	RESOURCE AVAILABILITY WITHOUT USING SECONDARY SKILL	RESOURCE AVAILABILITY USING SECONDARY SKILL
2					
3					
IPOST					

TEMPRT (I, J)

USE 3: POSTED FROM CREWSZ

I

I	1	2	3	4	5
1	SERIAL NO. OF CONCURRENT ACTIVITY	MIN. CREW	NORMAL CREW	MAX. CREW	NOT USED
2					
3					
IT					

Program SCHEDL
Accomplishes Scheduling of Activities

1. Read general information.

2. Update the Resource Pool (RPØØL) array for the current scheduling period.

3. Read Activity (ACTIV) array data saved from the previous scheduling period.

4. Add new schedulable activity data to the ACTIV array. (Based on early start time = 0)

5. Sort the ACTIV array according to ascending order of activity identification numbers.

6. Add remaining activity data to the ACTIV array.

7. Place activities in appropriate scheduling lists according to slack. (LISTAC, LISTBC, LISTA, LISTB)

8. Set up the Multi-Skill (MSKILL) array according to ascending order of activity identification numbers.

9. Set up the Concurrent Activity (CØNACT) array according to ascending order of slack.

10. Adjust maximum and minimum crew sizes for concurrent activities so that the same resource level is assigned to all.

11. Schedule activities appearing in the active, critical (LISTAC) array at minimum crew size. (Include secondary skills.)

12. Schedule activities appearing in the non-active, critical (LISTBC) array at minimum crew size. (Include secondary skills.)

13. Merge scheduled LISTBC activities into the LISTAC array.

14. Add resources to LISTAC activities up to critical crew size. (Include secondary skills.)

15. Schedule activities appearing in the active, non-critical (LISTA) array at minimum crew size. (Include secondary skills.)

16. Add resources to scheduled LISTA activities up to normal crew size.

17. Schedule activities appearing in the non-active, non-critical (LISTB) array at minimum crew size.

18. Add resources to scheduled LISTB activities up to normal crew size.

19. Merge scheduled LISTB activities into the LISTA array.

20. Add surplus resources to LISTAC activities up to maximum crew size.

21. Add surplus resources to LISTA activities up to maximum crew size.

22. Update status of active activities.

23. Update activity durations.

24. Write results on output file.

25. Remove activities that have been completed from the previous scheduling period in the ACTIV array and pack remaining data.

26. Update general information.

27. END (control reverts to critical path program)

```
  100       %                        PROGRAM SPREAD
  110       %
  120       %
  130       %
  140       % SCHEDULING PROGRAM FOR RESOURCE ALLOCATION IN THE DIVISION
  150       % OF ENGINEERING
  160       %
  170       !SWITCH OUT$:RESLT
  180       !LIMIT USS:10
  190       !EQUATE 7 DAT7
  200       !EQUATE 8 DAT8
  210       !EQUATE 9 DAT9
  220       !EQUATE 10 DAT10
  230       !EQUATE 11 DAT11
  240       !EQUATE 13 DAT13
  250       !EQUATE 14 DAT14
  260       !EQUATE 15 DAT15
  270       !LINK CPATH
  280       !LINK SCHEDL
  290       !LOAD CPATH
  300       !START
↑Q
SRU'S:,7
!EDIT SP2
↑L
  100       !SWITCH OUT$:;CLOSE
  110       !DROP CPATH-XQT
  120       !DROP 16
  130       !DROP 12
  140       !DROP DAT7
  150       !COPY;REPL FROM:DAT24 TO:DAT14
  160       !COPY;REPL FROM:DAT23 TO:DAT13
  170       !SAVE RESLT
  180       !OFF
```

```
100       %                          PROGRAM CPATH
110       %
120       %
130       %
140       % CALCULATES EARLY START TIMES, LATE START TIMES AND
150       % TOTAL SLACK FOR AN ACTIVITY ORIENTED JOB NETWORK
160       %
170       COMMON POOL(500)
180       COMMON/JOBCE/LNAT(3),LJT(3),LPRE1(3),%
190       LPRE2(3),LINF1(3),LINF2(3),LINF3(3)
200       DIMENSION PRE1(5)
210       INTEGER PRE1, PRE2
220       %
230       % INITIALIZE DYNAMIC STORAGE POOL
240       %
250       CALL P0DYN1(500)
260       READ(15) NJT
270       %
280       % READ GENERAL INFORMATION
290       %
300       CALL ININFO
310       CALL INITPR
320       READ(13) NPER,NEG,NCA,KSLACK,NAN,NAS,IPER
330       REWIND 13
340       NAN=0
350       DO 100 IJ=1,NJT
360       CALL INITJB
370       %
380       % READ NUMBER OF ACTIVITIES
390       %
400       READ(15) NAT
410       CALL COMIN(LNAT,NAT,IJ)
420       LNAT(2) = IJ
430       %
440       % READ JOB COMPLETION DATE
450       %
460       READ(15) JOBT
470       CALL COMIN(LJT,JOBT,IJ)
480       LJT(2) = IJ
490       %
500       % READ PRECEDENCE AND DURATION DATA
510       %
520       NPRE2 = 0
530       CALL COMIN(LPRE1,PRE1,NAT)
540       LPRE1(2) = NAT
550       DO 10 IA = 1,NAT
560       READ(14) PRE1(1),PRE1(2),PRE1(3)
570       NPRE2 = NPRE2+PRE1 1)
580       PRE1(4) = 0
590       PRE1(5) = 9999
600       CALL COMIN(LPRE1,PRE1,IA)
610       10 CONTINUE
620       %
```

```
      % READ PRE-ACTIVITY LIST
      %
      CALL COMIN(LPRE2,PRE2,NPRE2)
      LPE2(2) = NPRE2
      DO 20 IP2 = 1,NPRE2
      READ(15) PRE2
      CALL COMIN(LPRE2,PRE2,IP2)
   20 CONTINUE
      %
      % CALCULATE EARLY START TIMES
      %
      DO 50 NA = 2,NAT
      CALL COMOUT(LPRE1,PRE1,NA)
      IF(PRE1(1).EQ.0) GO TO 50
      I1 = PRE1(2)
      I2 = PRE1(2)+PRE1(1)-1
      DO 40 I = I1,I2
      CALL COMOUT(LPRE2,PRE2,I)
      IA = PRE2
      CALL COMOUT(LPRE1,PRE1,IA)
      I3 = PRE1(4)+PRE1(3)
      CALL COMOUT(LPRE1,PRE1,NA)
      IF(I3.LE.PRE1(4)) GO TO 30
      PRE1(4) = I3
      CALL COMIN(LPRE1,PRE1,NA)
   30 IC = IC+1
   40 CONTINUE
   50 CONTINUE
      %
      % CALCULATE LATE START TIMES
      %
      CALL COMOUT(LPRE1,PRE1,NAT)
      PRE1(5) = PRE1(4)
      CALL COMIN(LPRE1,PRE1,NAT)
      DO 75 NN = 2,NAT
      IF(IC.LE.0) GO TO 80
      NA = NAT+2-NN
      CALL COMOUT(LPRE1,PRE1,NA)
      I2A = PRE1(5)
      IF(PRE1(1).LE.0)  GO TO 75
      I1 = PRE1(1)
      DO 70 I=1,I1
      CALL COMOUT(LPRE2,PRE2,IC)
      IA = PRE2
      CALL COMOUT(LPRE1,PRE1,IA)
      I2 = I2A-PRE1(3)
      IF(I2.GE.PRE1(5)) GO TO 60
      PRE1(5) = I2
      CALL COMIN(LPRE1,PRE1,IA)
   60 IC = IC-1
   70 CONTINUE
   75 CONTINUE
      %
```

```
      % CALCULATE TOTAL SLACK
      %
   80 CALL COMOUT(LPRE1,PRE1,NAT)
      JDD = JOBT-PRE1(4)
      DO 90 NA = 1,NAT
      IA = NA+NAN
      CALL COMOUT(LPRE1,PRE1,NA)
      KPINF1 = PRE1(5)-PRE1(4)+JDD-IPER
      KPINF2 = PRE1(4)
      KPINF3 = PRE1(3)
      CALL COMIN(LINF1,KPINF1,IA)
      LINF1(2) = IA
      CALL COMIN(LINF2,KPINF2,IA)
      LINF2(2) = IA
      CALL COMIN(LINF3,KPINF3,IA)
      LINF3(2) = IA
   90 CONTINUE
      NAN = NAN+NAT
  100 CONTINUE
      %
      % WRITE CRITICAL PATH ROUTINE OUTPUT
      %
      DO 110 I = 1,NAN
      CALL COMOUT(LINF1,KPINF1,I)
      CALL COMOUT(LINF2,KPINF2,I)
      CALL COMOUT(LINF3,KPINF3,I)
      WRITE(6,*) KPINF1,KPINF2,KPINF3
      WRITE(7) KPINF1,KPINF2,KPINF3
  110 CONTINUE
      WRITE(13) NPER,NEG,NCA,KSLACK,NAN,NAS,IPER
      REWIND 13
      REWIND 7
      REWIND 14
      %
      % CRITICAL PATH RUN ONLY
      %
      IF(NPER.EQ.0)CALL OBEY('SWITCH OUT$:;CLOSE$SAVE RESLT$%
      DROP SCHEDL-XQT$OFF ',13)
      %
      % CALL SCHEDULING PROGRAM
      %
      CALL OBEY('LOAD SCHEDL$START ',5)
      STOP
      END
```

```
!LINK,XDEF CPATH
CPATH  .    04/17/73    11:27:12    XTN SYMS
                                                PAGE PAGE
COMIN   000000001703   CPATH   COMOUT  000000002153   CPATH
DREAD$  000000013215   CPATH   DREDB$  000000013704   CPATH
DWRIT$  000000012447   CPATH   FENRY$  000000005502   CPATH
FCNTL$  000000005337   CPATH   FERR$   000000002716   CPATH
FERR2$  000000002752   CPATH   FERR3$  000000002754   CPATH
FERR4$  000000002756   CPATH   FERR5$  000000002760   CPATH
FERR6$  000000002762   CPATH   FINIT$  000000003670   CPATH
FIOA1$  000000005713   CPATH   FIOA2$  000000006033   CPATH
FIOA3$  000000006617   CPATH   FIOA4$  000000007016   CPATH
FIOA5$  000000007070   CPATH   FIOA6$  000000007166   CPATH
FIOA7$  000000007243   CPATH   FIOA8$  000000007372   CPATH
FIOA9$  000000007426   CPATH   FIOAA$  000000007453   CPATH
FIOAB$  000000007462   CPATH   FIOAC$  000000007507   CPATH
FIOAH$  000000007653   CPATH   FIOAI$  000000007235   CPATH
FIOAJ$  000000006071   CPATH   FIOAK$  000000007706   CPATH
FIOAV$  000000010306   CPATH   FIOAW$  000000010314   CPATH
FIOAX$  000000010315   CPATH   FIOCI$  000000011400   CPATH
FIOCO$  000000010443   CPATH   FIOER$  000000007765   CPATH
FIOGN$  000000004622   CPATH   FIOGW$  000000004621   CPATH
FIOJ1$  000000006455   CPATH   FIOJ2$  000000010205   CPATH
FIOJ3$  000000006320   CPATH   FIOJ4$  000000010163   CPATH
FIOJ5$  000000010224   CPATH   FIOL1$  000000006457   CPATH
FKLDS$  000000006264   CPATH   FLIST$  000000006512   CPATH
FPAUS$  000000002557   CPATH   FSTOP$  000000002553   CPATH
FTERM$  000000006523   CPATH   FWKBK$  000000002750   CPATH
FWRTG$  000000004606   CPATH   IDDCD   000000002255   CPATH
ININFO  000000002467   CPATH   INITJB  000000002434   CPATH
INITPR  000000002451   CPATH   LDMHB$  000000102464   CPATH
LFMD1$  000000014446   CPATH   LFMDM$  000000014450   CPATH
LIDEL$  000000003555   CPATH   LIDIS$  000000004434   CPATH
LIENJ$  000000103421   CPATH   LIHAI$  000000003203   CPATH
LIHFG$  000000103260   CPATH   LIHOC$  000000103257   CPATH
LIHPR$  000000103261   CPATH   LINBL$  000000004443   CPATH
LIOUT$  000000004313   CPATH   LIRED$  000000003451   CPATH
LIREJ$  000000004325   CPATH   LIRSQ$  000000003450   CPATH
LIRSS$  000000003452   CPATH   LPTDL$  000000003034   CPATH
LPTER$  000000003072   CPATH   LPTNR$  000000003112   CPATH
LPTRE$  000000003000   CPATH   LPTRN$  000000003006   CPATH
OBEY    000000002523   CPATH   P0DYN1  000000001665   CPATH
USEPFR  000000000000   ABS

SRU'S:5.5
!
```

```
 10'      SUBROUTINE P0DYN1(MAXP)
 12       COMMON POOL(100)
 13       COMMON/POOLCT  NOCOMP,MAXCOM,NOARRAY,LOCARY(100)
 14       NOCOMP=0
 15       MAXCOM=MAXP
 16       NOARRAY=0
 17       RETURN
100       ENTRY COMIN(LOC,BLOCK,N)
101       % STORE BLOCK DYNAMICALLY
103       INTEGER BLOCK,POOL
104       DIMENSION LOC(3),BLOCK(10)
105       4 NO=LOC(1)
106       M=N-LOC(2)
107       IF(M.GT.0)GO TO 10
108       % STORE ARRAY
109       5 I=LOC(3)
110       L=LOCARY(I)+NO*(N-1)
111       DO 6 I=1,NO
112       POOL(L)=BLOCK(I)
113       6 L=L+1
114       RETURN
115       % ALLOCATE SPACE
116       10 NOADD=NO*M
117       MAX=NOCOMP+NOADD
118       IF(MAXCOM.GE.MAX)GO TO 20
119       15 PRINT*,'***TOO BIG'
120       RETURN
121       20 IF(LOC(2).GT.0)GO TO 30
122       % ADD TO END OF POOL
123       L1=NOCOMP+1
124       L2=MAX
125       IF(NOARRAY.GE.100)GO TO 15
126       NOARRAY=NOARRAY+1
127       LOC(3)=NOARRAY
128       LOCARY(NOARRAY)=L1
129       25 NOCOMP=MAX
130       LOC(2)=N
131       DO 28 I=L1,L2
132       28 POOL(I)=-0.
133       GO TO 5
134       %
135       30 I=LOC(3)
136       L1=LOCARY(I)+NO*LOC(2)
137       L2=L1+NOADD-1
138       MOVE=NOCOMP-L1+1
139       IF(MOVE.LE.0)GO TO 25
```

```
      % SHIFT POOL
      I=MA)
      J=NOCOMP
      DO 35 K=1,MOVE
      POOL(I)=POOL(J)
      I=I-1
      35 J=J-1
      %CHANGE LOCARY
      DO 45 I=1,NOARRAY
      IF(LOCARY(I .LT.L1)GO TO 45
      LOCARY(I)=LOCARY(I)+NOADD
      45 CONTINUE
      GO TO 25
      ENTRY COMOUT(LOC,BLOCK,N)
      % GET BLOCK
      NO=LOC(1)
      IF(LOC(2).LT.N)GO TO 220
      I=LOC(3)
      L=LOCARY(I)+NO*(N-1)
      DO 210 I=1,NO
      BLOCK(I)=POOL(L)
      210 L=L+1
      RETURN
      % RETURN ZEROS
      220 DO 221 I=1,NO
      221 BLOCK(I)=-0.
      RETURN
      ENTRY IDDCD(ITYPE,BLOCK,LOC,NOID,N)
      % DECODE ID
301.1 ID=BLOCK(1)
      NOID=LOC(2)
      IF(NOID.LE.0) GO TO 320
      I=LOC(3)
      L=LOCARY(I)
      DO 310 I=1,NOID
      IF(ID.EQ.POOL(L))GO TO 330
      310 L=L+1
      320 N=0
      GO TO 340
      330 N=I
      340 GO TO(350,360,370),ITYPE
      % OLD
      350 RETURN
      % OLD OR NEW
      360 IF(N.GT.0)RETURN
      361 NOID=NOID+1
      N=NOID
      GO TO 4
      % NEW ONLY
      370 IF(N.EQ.0) GO TO 361
      N=0
      RETURN
      END
```

```
100       SUBROUTINE INITJB
110       %
120       %INITIALIZE JOB DYNAMIC STORAGE
130       %
140       COMMON/JOBCB/LNAT(3),LJT(3),LPRE1(3),%
150       LPRE2(3),LINF1(3),LINF2(3),LINF3(3)
160       LNAT(1)=1
170       LNAT(2)=0
180       LJT(1)=1
190       LJT(2)=0
200       RETURN
210       ENTRY INITPR
220       LPRE1(1)=5
230       LPRE1(2)=0
240       LPRE2(1)=1
250       LPRE2(2)=0
260       RETURN
270       ENTRY ININFO
280       LINF1(1)=1
290       LINF1(2)=0
300       LINF2(1)=1
310       LINF2(2)=0
320       LINF3(1)=1
330       LINF3(2)=0
340       RETURN
350       END
```

```
100       %                    PROGRAM SCHEDL
110       %
120       %
130       %
140       % ACCOMPLISHES SHEDULING OF ACTIVITIES
150       %
160       COMMON ACTIV(1000,13),MSKILL(1000,4),CONACT(50,2),%
170       RPOOL(100,2),TEMPRT(20,5),NAN,NAS,IS,IMS,ICA,NCA,ISG,NSG,%
180       IPOST,NPER,KSLACK,LISTAC(100),LISTBC(100),LISTA(100),%
190       LISTB(100),NL1,NL2,NL3,NL4,IPER
200       INTEGER ACTIV,RPOOL,CONACT,TEMPRT
210       ICA=0
220       %
230       % READ GENERAL INFORMATION
240       %
250       READ(13) NPER,NSG,NCA,KSLACK,NAN,NAS,IPER
260       REWIND 13
270       %
280       % UPDATE RPOOL ARRAY
290       %
300       CALL REPOOL
310       IF(NAS.EQ.0) GO TO 10
320       %
330       % READ ACTIV ARRAY DATA SAVED FROM PREVIOUS SCHEDULING PERIOD
340       %
350       CALL RACTIV
360       %
370       % ADD NEW SCHEDULABLE ACTIVITIES TO ACTIV ARRAY
380       %
390       10 CALL UPDATL
400       IF(NAS-1) 100,15,12
410       %
420       % SORT ACTIV ARRAY ACCORDING TO ASCENDING ORDER OF
430       % INDENTIFICATION NUMBERS
440       %
450       12 CALL SORT
460       %
470       % ADD REMAINING DATA TO ACTIV ARRAY
480       %
490       15 CALL RDFIL
500       %
510       % PLACE ACTIVITIES IN APPROPRIATE SCHEDULING LISTS SORTED ON
520       % TOTAL SLACK
530       %
540       CALL PLIST
550       %
560       % SET UP MSKILL ARRAY ACCORDING TO ASCENDING ORDER OF
570       % ACTIVITY INDENTIFICATION NUMBERS
580       %
590       CALL MULTIS
600       IF(NCA.EQ.0) GO TO 20
610       %
```

```
      % SET UP CONACT ARRAY ACCORDING TO ASCENDING ORDER OF SLACK
      %
      CALL CONAC
      %
      % ADJUST MINIMUM AND MAXIMUM CREW SIZES FOR CONCURRENT
      % ACTIVITIES
      %
      CALL CREWSZ
   20 IF(NL1.EQ.0) GO TO 30
      %
      % SCHEDULE ACTIVITIES IN LISTAC ARRAY AT MINIMUM CREW SIZE
      %
      CALL LISTM(LISTAC,NL1,1,1)
   30 IF(NL2.EQ.0) GO TO 40
      %
      % SCHEDULE ACTIVITIES IN LISTBC ARRAY AT MINIMUM CREW SIZE
      %
      CALL LISTM(LISTBC,NL2,1,2)
      %
      % MERGE SCHEDULED ACTIVITIES IN LISTBC ARRAY INTO
      % LISTAC ARRAY
      %
      CALL MERGE(LISTAC,LISTBC,NL1,NL2,1)
   40 IF(NL1.EQ.0) GO TO 50
      %
      % SCHEDULE ACTIVITIES IN LISTAC ARRAY AT CRITICAL CREW SIZE
      %
      CALL LISTAD(LISTAC,NL1,1,2,1)
   50 IF(NL3.EQ.0) GO TO 60
      %
      % SCHEDULE ACTIVITIES IN LISTA ARRAY AT MINIMUM CREW SIZE
      %
      CALL LISTM(LISTA,NL3,1,3)
      IF(NL3.EQ.0) GO TO 60
      %
      % SCHEDULE ACTIVITIES IN LISTA ARRAY AT NORMAL CREW SIZE
      %
      CALL LISTAD( LISTA,NL3,0,1,3)
   60 IF(NL4.EQ.0) GO TO 70
      %
      % SCHEDULE ACTIVITIES IN LISTB ARRAY AT MINIMUM CREW SIZE
      %
      CALL LISTM(LISTB,NL4,0,4)
      IF(NL4.EQ.0) GO TO 70
      %
      % SCHEDULE ACTIVITIES IN LISTB ARRAY AT NORMAL CREW SIZE
      %
      CALL LISTAD(LISTB,NL4,0,1,4)
      %
```

```
1110      % MERGE SCHEDULED ACTIVITIES IN LISTB ARRAY INTO
1120      % LISTA ARRAY
1130      %
1140      CALL MERGE (LISTA,LISTB,NL3,NL4,3)
1150      70 IF(NL1.EQ.0) GO TO 80
1160      %
1170      % ADD SURPLUS RESOURCES TO LISTAC ARRAY
1180      %
1190      CALL LISTAD (LISTAC,NL1,0,3,1)
1200      80 IF(NL3.EQ.0) GO TO 90
1210      %
1220      % ADD SURPLUS RESOURCES TO LISTA ARRAY
1230      %
1240      CALL LISTAD (LISTA,NL3,0,3,3)
1250      %
1260      % UPDATE STATUS OF ACTIVE ACTIVITIES
1270      %
1280      90 CALL UPSTAT
1290      %
1300      % UPDATE ACTIVITY DURATIONS
1310      %
1320      CALL UPDUR
1330      %
1340      % WRITE RESULTS ON FILE
1350      %
1360      CALL RESULT
1370      %
1380      % PACK DATA FROM LAST PERIOD IN ACTIV ARRAY
1390      %
1400      CALL PACK
1410      %
1420      % UPDATE GENERAL INFORMATION FILE
1430      %
1440      IPER=IPER+1
1450      IF(IPER.EQ.NPER) GO TO 100
1460      WRITE(13) NPER,NSG,NCA,KSLACK,NAN,NAS,IPER
1470      REWIND 13
1480      %
1490      % RETURN TO CRITICAL PATH PROGRAM
1500      %
1510      CALL OBEY('LOAD CPATH;START ',4)
1520      100 CALL OBEY('DO ZP2',2)
1530      STOP
1540      END
```

```
LINK.XDEF SCHEDL
SCHEDL .    04.17/73    11:29:30    XTN SYMS
                                              PAGE PAGE
ADJUST  000000005340    SCHEDL    CONAC   000000002213    SCHEDL
CREWSZ  000000002345    SCHEDL    DELCS   000000005040    SCHEDL
DREAD$  000000016527    SCHEDL    DREDB$  000000017216    SCHEDL
DWRIT$  000000015761    SCHEDL    FENRY$  000000013503    SCHEDL
FCNTL$  000000013340    SCHEDL    FERP$   000000006230    SCHEDL
FERR2$  000000006264    SCHEDL    FERR3$  000000006266    SCHEDL
FERR4$  000000006270    SCHEDL    FERR5$  000000006272    SCHEDL
FERR6$  000000006274    SCHEDL    FINIT$  000000006202    SCHEDL
FIOA1$  000000010100    SCHEDL    FIOA2$  000000010220    SCHEDL
FIOA3$  000000011004    SCHEDL    FIOA4$  000000011203    SCHEDL
FIOA5$  000000011255    SCHEDL    FIOA6$  000000011353    SCHEDL
FIOA7$  000000011430    SCHEDL    FIOA8$  000000011557    SCHEDL
FIOA9$  000000011613    SCHEDL    FIOAA$  000000011640    SCHEDL
FIOAB$  000000011647    SCHEDL    FIOAC$  000000011674    SCHEDL
FIOAH$  000000012040    SCHEDL    FIOAI$  000000011422    SCHEDL
FIOAJ$  000000010256    SCHEDL    FIOAK$  000000012073    SCHEDL
FIOAV$  000000012473    SCHEDL    FIOAW$  000000012501    SCHEDL
FIOAX$  000000012502    SCHEDL    FIOCI$  000000014712    SCHEDL
FIOCO$  000000013755    SCHEDL    FIOER$  000000012152    SCHEDL
FIOGN$  000000012623    SCHEDL    FIOSW$  000000012622    SCHEDL
FIOJ1$  000000010642    SCHEDL    FIOJ2$  000000012372    SCHEDL
FIOJ3$  000000010505    SCHEDL    FIOJ4$  000000012350    SCHEDL
FIOJ5$  000000012411    SCHEDL    FIOL1$  000000010644    SCHEDL
FKLDG$  000000010451    SCHEDL    FLIST$  000000010677    SCHEDL
FPAUS$  000000006071    SCHEDL    FSTOP$  000000006065    SCHEDL
FTERM$  000000010710    SCHEDL    FWKSK$  000000006262    SCHEDL
FWRTS$  000000012607    SCHEDL    LDMHB$  000000144310    SCHEDL
LFMD1$  000000017760    SCHEDL    LFMDM$  000000017762    SCHEDL
LIDEL$  000000007067    SCHEDL    LIDIS$  000000007746    SCHEDL
LIENJ$  000000145245    SCHEDL    LIHAI$  000000006515    SCHEDL
LIHFG$  000000145104    SCHEDL    LIHOC$  000000145103    SCHEDL
LIHPA$  000000145105    SCHEDL    LINBL$  000000007755    SCHEDL
LIOUT$  000000007625    SCHEDL    LIRED$  000000006763    SCHEDL
LIREJ$  000000007637    SCHEDL    LIREQ$  000000006762    SCHEDL
LIRES$  000000006764    SCHEDL    LISTAD  000000004161    SCHEDL
LISTM   000000003541    SCHEDL    LPACK   000000004757    SCHEDL
LPTDL$  000000006346    SCHEDL    LPTER$  000000006404    SCHEDL
LPTNR$  000000006424    SCHEDL    LPTRE$  000000006312    SCHEDL
LPTRN$  000000006320    SCHEDL    MERGE   000000004004    SCHEDL
MULTIS  000000002110    SCHEDL    OBEY    000000006035    SCHEDL
PACK    000000003257    SCHEDL    PERAV   000000005224    SCHEDL
PLIST   000000001752    SCHEDL    POST    000000004602    SCHEDL
RACTIV  000000001417    SCHEDL    RDFIL   000000001665    SCHEDL
REPOOL  000000001334    SCHEDL    RESULT  000000003057    SCHEDL
SORT    000000001555    SCHEDL    SORTBK  000000005713    SCHEDL
SORTCR  000000005620    SCHEDL    SORTL   000000005523    SCHEDL
SSKIL   000000004501    SCHEDL    SSKILA  000000005266    SCHEDL
SUB1    000000001327    SCHEDL    SUB2    000000003534    SCHEDL
SUB3    000000004474    SCHEDL    SUB4    000000005516    SCHEDL
TPOST   000000004544    SCHEDL    UPDATL  000000001456    SCHEDL
UPDUR   000000002650    SCHEDL    UPSTAT  000000002563    SCHEDL
USERFR  000000000000    ABS

SRU'S:6.4
```

```
100       SUBROUTINE SUB1
110       %
120       %
130       %
140       DIMENSION IROW(6)
150       COMMON ACTIV(1000,13),MSKILL(1000,4),CONACT(50,2),%
160       RPOOL(100,2),TEMPRT(20,5),NAN,NAS,IS,IMS,ICA,NCA,ISG,NSG,%
170       IPOST,NPER,KSLACK,LISTRC(100),LISTSC(100),LISTA(100),%
180       LISTS(100),NL1,NL2,NL3,NL4,IPER
190       INTEGER ACTIV,RPOOL,CONACT,TEMPRT
200       %
210       %
220       %-----------------------------------------------------------
230       ENTRY REPOOL
240       %
250       % UPDATES RESOURCE DATA FOR NEW PERIOD
260       %
270       NSKIP=NSG*IPER
280       DO 66 I=1,NSG
290       READ(11) RPOOL(I,2)
300       66 CONTINUE
310       IF(NSKIP.EQ.0) GO TO 69
320       DO 68 I=1,NSKIP
330       READ(11)
340       68 CONTINUE
350       69 DO 70 I=1,NSG
360       READ(11) RPOOL(I,1)
370       70 CONTINUE
380       RETURN
390       %
```

```
400       %
410       %-------------------------------------------------------------
420       ENTRY RACTIV
430       %
440       % READ ACTIVITY ARRAY
450       %
460       DO 2 N=1,NAS
470       READ(12) ACTIV(N,1),ACTIV(N,7),ACTIV(N,10)
480       2 CONTINUE
490       REWIND 12
500       RETURN
510       %
520       %
530       %-------------------------------------------------------------
540       ENTRY UPDATL
550       %
560       % ADD NEW ACTIVITIES TO ACTIV ARRAY
570       %
580       IS=1
590       DO 8 NA=1,NAN
600       READ(7)ITS,IES,IDUR
610       IF(IES.NE.0.OR.IDUR.LE.0) GO TO 8
620       IF(ACTIV(IS,1).NE.NA) GO TO 6
630       ACTIV(IS,7)=IDUR
640       ACTIV(IS,11)=ITS
650       IS=IS+1
660       GO TO 8
670       6 NAS=NAS+1
680       ACTIV(NAS,1)=NA
690       ACTIV(NAS,7)=IDUR
700       ACTIV(NAS,10)=0
710       ACTIV(NAS,11)=ITS
720       ACTIV(NAS,13)=0
730       8 CONTINUE
740       REWIND 7
750       RETURN
760       %
```

```
      %
      %-------------------------------------------------------------
      ENTRY SORT
      %
      % SORTS ACTIV ARRAY ON COLUMN 1 ACCORDING TO ASCENDING ORDE
      % OF INDENTIFICATION NUMBERS
      %
      NM1=NAS-1
      DO 10 I=1,NM1
      IP1=I+1
      DO 10 J=IP1,NAS
      IF(ACTIV(I,1).LE.ACTIV(J,1)) GO TO 10
      ITEMP1=ACTIV(I,1)
      ITEMP2=ACTIV(I,7)
      ITEMP3=ACTIV(I,10)
      ITEMP4=ACTIV(I,11)
      ACTIV(I,1)=ACTIV(J,1)
      ACTIV(I,7)=ACTIV(J,7)
      ACTIV(I,10)=ACTIV(J,10)
      ACTIV(I,11)=ACTIV(J,11)
      ACTIV(J,1)=ITEMP1
      ACTIV(J,7)=ITEMP2
      ACTIV(J,10)=ITEMP3
      ACTIV(J,11)=ITEMP4
      10 CONTINUE
      RETURN
      %
      %
      %-------------------------------------------------------------
      ENTRY RDFIL
      %
      % FILLS ACTIV ARRAY WITH INPUT DATA
      %
      IS=1
      DO 12 NA=1,NAN
      READ(8)(IRON(I),I=1,6)
      IF(ACTIV(IS,1).NE.NA) GO TO 12
      ACTIV(IS,2)=IRON(1)
      ACTIV(IS,3)=IRON(2)
      ACTIV(IS,4)=IRON(3)
      ACTIV(IS,5)=IRON(4)
      ACTIV(IS,6)=IRON(5)
      ACTIV(IS,13)=IRON(6)
      IF(IS.GE.NAS) GO TO 13
      IS=IS+1
      12 CONTINUE
      13 RETURN
      %
```

```
1250      %
1260      %-----------------------------------------------------------------
1270      ENTRY PLIST
1280      %
1290      % PLACES AND SORTS ACTIVITIES IN SCHEDULING LISTS
1300      %
1310      NL1=0
1320      NL2=0
1330      NL3=0
1340      NL4=0
1350      %
1360      % PLACE ACTIVITIES IN SCHEDULING LISTS AND ASSIGN STATUS
1370      % TO ACTIV(I,10)
1380      %
1390      DO 26 I=1,NAS
1400      IF(ACTIV(I,10).GE.1) GO TO 20
1410      IF(ACTIV(I,11).GT.KSLACK) GO TO 18
1420      %
1430      % NONACTIVE CRITICAL
1440      %
1450      NL2=NL2+1
1460      LISTBC(NL2)=I
1470      ACTIV(I,10)=2
1480      GO TO 26
1490      %
1500      % NONACTIVE NON-CRITICAL
1510      %
1520      18 NL4=NL4+1
1530      LISTB(NL4)=I
1540      ACTIV(I,10)=4
1550      GO TO 26
1560      20 IF(ACTIV(I,11).GT.KSLACK) GO TO 24
1570      %
1580      % ACTIVE CRITICAL
1590      %
1600      NL1=NL1+1
1610      LISTAC(NL1)=I
1620      GO TO 26
1630      %
1640      % ACTIVE NON-CRITICAL
1650      %
1660      24 NL3=NL3+1
1670      LISTA(NL3)=I
1680      26 CONTINUE
1690      %
1700      % SORT THE FOUR SCHEDULABLE LISTS ACCORDING TO SLACK
1710      %
1720      IF(NL1.LE.1) GO TO 27
1730      CALL SORTL(LISTAC,NL1)
1740      27 IF(NL2.LE.1) GO TO 28
1750      CALL SORTL(LISTBC,NL2)
1760      28 IF(NL3.LE.1  GO TO 29
1770      CALL SORTL(LISTA,NL3)
1780      29 IF(NL4.LE 1) GO TO 30
1790      CALL SORTL(LISTB,NL4)
1800      30 RETURN
1810      %
```

```
1820      %
1830      %-----------------------------------------------------------------
1840      ENTRY MULTIS
1850      %
1860      % SETS UP MSKILL ARRAY ACCORDING TO ASCENDING ORDER OF
1870      % IDENTIFICATION NUMBERS
1880      %
1890      IMS=0.
1900      DO 36 I=1,NAS
1910      IF(ACTIV(I,13).EQ.0) GO TO 36
1920      31 READ(9)(IROW(J),J=1,3)
1930      IF (IROW(1) -ACTIV(I,1)) 31,32,36
1940      32 NMS=ACTIV(I,13)
1950      ACTIV(I,12)=IMS+1
1960      DO 34 MS=1,NMS
1970      IMS=IMS+1
1980      MSKILL(IMS,1)=IROW(2)
1990      MSKILL(IMS,2)=IROW(3)
2000      MSKILL(IMS,3)=0
2010      MSKILL(IMS,4)=0
2020      IF(MS.EQ.NMS) GO TO 34
2030      READ(9) (IROW(J),J=1,3)
2040      34 CONTINUE
2050      36 CONTINUE
2060      RETURN
2070      %
```

```
2080      %
2090      %-------------------------------------------------------------
2100      ENTRY CONAC
2110      %
2120      % SET UP CONACT ARRAY IN ASCENDING ORDER OF SLACK
2130      %
2140      NR=1
2150      ICA=0
2160      READ(10) (IROW(J),J=1,2)
2170      DO 50 I=1,NAS
2180      IF(IROW(1)-ACTIV(I,1)) 44,46,50
2190      40 READ(10)(IROW(J),J=1,2)
2200      NR=NR+1
2210      42 IF(IROW(1)-ACTIV(I,1)) 44,46,50
2220      44 IF(NR.GE.NCA) GO TO 52
2230      GO TO 40
2240      46 ICA=ICA+1
2250      CONACT(ICA,1)=I
2260      IB=I+1
2270      DO 48 IC=IB,NAS
2280      IF(IROW(2).NE.ACTIV(IC,1)) GO TO 48
2290      CONACT(ICA,2)=IC
2300      GO TO 49
2310      48 CONTINUE
2320      ICA=ICA+1
2330      49 IF (NR.GE.NCA) GO TO 52
2340      GO TO 40
2350      50 CONTINUE
2360      52 IF(ICA.LT.1) RETURN
2370      %
2380      % SORT CONACT ARRAY
2390      %
2400      CALL SORTCA
2410      RETURN
2420      %
```

```
2430      %
2440      %-----------------------------------------------------------------
2450      ENTRY CREW2
2460      %
2470      % ADJUST MINIMUM AND MAXIMUM CREW SIZES FOR CONCURRENT
2480      % ACTIVITIES TO SAME RESOURCE LEVELS
2490      %
2500      %
2510      IT=0
2520      DO 64 I=1,ICA
2530      %
2540      % STORE CONCURRENT ACTIVITIES IN TEMPRT
2550      %
2560      57 IT=IT+1
2570      IA=CONACT(I,1)
2580      IF(IT.GT.1) IA=CONACT(I,2)
2590      TEMPRT(IT,1)=IA
2600      TEMPRT(IT,2)=ACTIV(IA,3)
2610      TEMPRT(IT,3)=ACTIV(IA,4)
2620      TEMPRT(IT,4)=ACTIV(IA,5)
2630      IF(IT.EQ.1) GO TO 57
2640      IF(I.EQ.ICA) GO TO 58
2650      IF(CONACT(I,1).EQ.CONACT(I+1,1)) GO TO 64
2660      %
2670      % CALCULATE MAXIMUM AND MINIMUM CREW SIZE RATIOS
2680      %
2690      58 GMIN=0
2700      GMAX=9999
2710      DO 60 J=1,IT
2720      IS=TEMPRT(J,1)
2730      IF(ACTIV(IS,4).EQ.0) GO TO 60
2740      RATIO=100.0*FLOAT(ACTIV(IS,3))/FLOAT(ACTIV(IS,4))
2750      IF(GMIN.LT.RATIO) GMIN=RATIO
2760      RATIO=100.0*FLOAT(ACTIV(IS,5))/FLOAT(ACTIV(IS,4))
2770      IF(GMAX.GT.RATIO)GMAX=RATIO
2780      60 CONTINUE
2790      %
2800      % ADJUST AND RECORD CREW SIZES IN ACTIV ARRAY
2810      %
2820      DO 62 J=1,IT
2830      IS=TEMPRT(J,1)
2840      ACTIV(IS,3)=0.01*FLOAT(ACTIV(IS,4))*GMIN+0.5
2850      ACTIV(IS,5)=0.01*FLOAT(ACTIV(IS,4))*GMAX+0.5
2860      62 CONTINUE
2870      IT=0
2880      64 CONTINUE
2890      RETURN
2900      %
```

```
2910      %
2920      %------------ ----------------------------------------------
2930      ENTRY UPSTAT
2940      %
2950      % UPDATES STATUS IN ACTIV ARRAY FOR ACTIVE ACTIVITIES
2960      %
2970      DO 135 N=1,NAS
2980      ACTIV(N,10)=0
2990      135 CONTINUE
3000      IF(NL1.EQ.0) GO TO 137
3010      %
3020      % ACTIVE CRITICAL
3030      %
3040      DO 136 I=1,NL1
3050      IS=LISTAC(I)
3060      ACTIV(IS,10)=1
3070      136 CONTINUE
3080      137 IF(NL3.EQ.0) GO TO 139
3090      %
3100      % ACTIVE NON-CRITICAL
3110      %
3120      DO 138 I=1,NL3
3130      IS=LISTA(I)
3140      ACTIV(IS,10)=3
3150      138 CONTINUE
3160      139 RETURN
3170      %
```

```
      %
      %-----------------------------------------------------------------
      ENTRY UPDUR
      %
      % UPDATES ACTIVITY DURATIONS
      %
      IS=1
      JCA=1
      IF(ICA.LE.1) GO TO 140
      %
      % SORT CONACT ARRAY IN ASCENDING ORDER OF SERIAL NUMBERS
      %
      CALL SORTSK
      %
      % CALCULATE CHANGE IN DURATIONS
      %
  140 DO 149 NA=1,NAN
      READ(14) IROW(1),IROW(2),IDUR
      IF(ACTIV(IS,1).NE.NA) GO TO 148
      IF(ACTIV(IS,10).EQ.0) GO TO 147
      IF(ACTIV(IS,4).NE.0) GO TO 142
      NDUR=1
      GO TO 146
  142 IF(JCA.GT.ICA) GO TO 144
      IF(CONACT(JCA,2)-IS) 143,146,144
  143 JCA=JCA+1
      GO TO 142
  144 NDUR=FLOAT(ACTIV(IS,8))*(1.+.025*FLOAT(ACTIV(IS,6)))/%
      FLOAT(ACTIV(IS,4))+0.49
      %
      % UPDATE ACTIV ARRAY
      %
  146 IDUR=IDUR-NDUR
      IF(IDUR.LT.0)IDUR=0
      ACTIV(IS,7)=IDUR
  147 IS=IS+1
      %
      % RECORD UPDATED INFORMATION ON SCRATCH FILE
      %
  148 WRITE(16) IROW(1),IROW(2),IDUR
  149 CONTINUE
  150 REWIND 14
      REWIND 16
      %
      % UPDATE CRITICAL PATH PROGRAM INPUT
      %
      DO 151 NA=1,NAN
      READ(16) IROW(1),IROW(2),IROW(3)
      WRITE(14) IROW(1),IROW(2),IROW(3)
  151 CONTINUE
      REWIND 14
      REWIND 16
      RETURN
      %
```

```
3720      %
3730      %-----------------------------------------------------------------
3740      ENTRY RESULT
3750      %
3760      % WRITES RESULTS FOR PERIOD ON FILE
3770      %
3780      DO 152 N=1,NAS
3790      WRITE(6,*)(ACTIV(N,I),I=1,13)
3800      152 CONTINUE
3810      IF(IMS.EQ.0) GO TO 154
3820      DO 153 N=1,IMS
3830      WRITE(6,*)(MSKILL(N,I),I=1,4)
3840      153 CONTINUE
3850      154 IF(NL1.EQ.0) GO TO 156
3860      DO 155 N=1,NL1
3870      WRITE(6,*) LISTAC(N)
3880      155 CONTINUE
3890      156 IF(NL3.EQ.0) GO TO 158
3900      DO 157 N=1,NL3
3910      WRITE(6,*) LISTA(N)
3920      157 CONTINUE
3930      158 CONTINUE
3940      DO 159 N=1,NSG
3950      WRITE(6,*)(RPOOL(N,I),I=1,2)
3960      159 CONTINUE
3970      RETURN
3980      %
3990      %
4000      %-----------------------------------------------------------------
4010      ENTRY PACK
4020      %
4030      % PACKS DATA IN ACTIV ARRAY AFTER REMOVING NONACTIVE
4040      % ACTIVITIES
4050      %
4060      IN=1
4070      DO 4 I=1,NAS
4080      IF(ACTIV(I,7).LE.0) GO TO 4
4090      ACTIV(IN,1)=ACTIV(I,1)
4100      ACTIV(IN,7)=ACTIV(I,7)
4110      ACTIV(IN,8)=0
4120      ACTIV(IN,9)=0
4130      ACTIV(IN,10)=ACTIV(I,10)
4140      WRITE(12)ACTIV(IN,1),ACTIV(IN,7),ACTIV(IN,10)
4150      IN=IN+1
4160      4 CONTINUE
4170      REWIND 12
4180      NAS=IN-1
4190      RETURN
4200      END
```

```
      SUBROUTINE SUB2
      %
      %
      %
      DIMENSION LIST1(1000),LIST2(1000)
      COMMON ACTIV(1000,13),MSKILL(1000,4),CONACT(50,2),%
      RPOOL(100,2),TEMPRT(20,5),NAN,NAE,IS,IME,ICA,NCA,ISG,NSG,%
      IPOST,NPER,NSLACK
      INTEGER ACTIV,RPOOL,CONACT,TEMPRT,PERCTA,PERCTB
      %
      %
      %-------------------------------------------------------------
      ENTRY LISTM(LIST1,NLIST,NSSKIL,NC)
      %
      % SCHEDULES ACTIVITIES ON A SPECIFIED LIST AT MINIMUM CREW
      % SIZE
      %
      JCA=0
      IPOST=0
      DO 88 I=1,NLIST
      IS= LIST1(I)
      %
      % IS MINIMUM CREW SIZE AVAILABLE?
      %
      NSKILL=ACTIV(IS,13)+1
      DO 82 N=1,NSKILL
      MSSKIL=0
      IF (N.GT.1) GO TO 72
      ISG=ACTIV(IS,2)
      MINCS=ACTIV(IS,3)
      GO TO 74
      72 LOC=ACTIV(IS,12)+N-2
      ISG=MSKILL(LOC,1)
      MINCS=0.01*FLOAT (MSKILL(LOC,2))*FLOAT(ACTIV(IS,3))+0.5
      74 IF(MINCS.LE.RPOOL(ISG,1)) GO TO 76
      IF (NSSKIL.EQ.0) GO TO 78
      MAVAIL=RPOOL(ISG,1)
      %
      % IS MINIMUM CREW SIZE AVAILABLE BY USING SECONDARY SKILLS?
      %
      CALL SSKIL (MAVAIL,ISSKIL,MSSKIL,MINCS)
      IF(MSSKIL.EQ.0) GO TO 78
      76 MAVAIL=MINCS-MSSKIL
      GO TO 80
      %
```

```
      % POST IN TEMPORARY TABLE
      %
      78 MAVAIL=0
      80 IPOST=IPOST+1
      CALL TPOST(MAVAIL,ISSKIL,MSSKIL)
      82 CONTINUE
      %
      % IS THERE ANOTHER CONCURRENT ACTIVITY?
      %
      IF(JCA.GT.ICA.OR.ICA.EQ.0) GO TO 87
      IF(JCA.EQ.0) JCA=1
      83 JS=CONACT(JCA,1)
      IF(ACTIV(JS,10).EQ.NC) GO TO 84
      JCA=JCA+1
      IF(JCA.LT.ICA) GO TO 83
      84 IF(IS.NE.CONACT(JCA,2)) GO TO 85
      JCA=JCA+1
      IF(JCA.GT.ICA) GO TO 87
      GO TO 86
      85 IF(IS.NE.CONACT(JCA,1)) GO TO 87
      GO TO 88
      86 IF(CONACT(JCA-1,1).EQ.CONACT(JCA,1)) GO TO 88
      %
      % POST RESOURCES IF SUFFICIENT?
      %
      87 CALL POST
      IPOST=0
      88 CONTINUE
      %
      % PACK SCHEDULING LIST
      %
      CALL LPACK(LIST1,NLIST)
      RETURN
      %
```

```
      %
      %-----------------------------------------------------------------
      ENTRY MERGE(LIST1,LIST2,N1,N2,NC)
      %
      % MERGES ACTIVE ACTIVITIES FROM TWO LISTS INTO ONE LIST
      % IN ORDER OF ASCENDING SLACK
      %
      I1=1
      I2=1
      I3=N2+1
      IF(N1.EQ.0) GO TO 103
      %
      % ADD LIST1 ITEMS TO LIST2
      %
      DO 102 N=1,N1
      J=N+N2
      LIST2(J)=LIST1(N)
  102 CONTINUE
  103 N3=N1+N2
  104 IF(N1.EQ.0) GO TO 106
      %
      % SELECT ITEMS FROM LIST2 TO CONSTRUCT A NEW MERGED LIST1
      %
      IS1=LIST2(I3)
      IS2=LIST2(I2)
      IF(I2.GT.N2) GO TO 105
      IF(I3.GT.N3) GO TO 106
      IF(ACTIV(IS1,11).GT.ACTIV(IS2,11)) GO TO 106
  105 LIST1(I1)=LIST2(I3)
      I3=I3+1
      GO TO 108
  106 LIST1(I1)=LIST2(I2)
      IS2=LIST1(I1)
      ACTIV(IS2,10)=NC
      I2=I2+1
  108 I1=I1+1
      IF(I1.LE.N3) GO TO 104
      N1=N3
      N2=0
      RETURN
      %
```

```
1300      %
1310      %-----------------------------------------------------------------
1320      ENTRY LISTAD(LIST1,NLIST,NSSKIL,KS,NC)
1330      %
1340      % SCHEDULES ACTIVITIES AT GREATER THAN MINIMUM CREW SIZE
1350      %
1360      JCA=0
1370      IPOST=0
1380      DO 126 I=1,NLIST
1390      MINPER=999999
1400      IS=LIST1(I)
1410      NSKILL=ACTIV(IS,13)+1
1420      DO 120 N=1,NSKILL
1430      MSSKIL=0
1440      %
1450      % DETERMINE SKILL GROUP TO BE USED
1460      %
1470      IF(N.GT.1) GO TO 112
1480      ISG=ACTIV(IS,2)
1490      GO TO 114
1500      112 LOC=ACTIV(IS,12)+N-2
1510      ISG=MSKILL(LOC,1)
1520      %
1530      % CALCULATE CREW SIZE INCREMENT
1540      %
1550      114 CALL DELCS(KSDEL,KS,N,LOC)
1560      IF(KSDEL.NE.0) GO TO 115
1570      PERCTA=0
1580      GO TO 116
1590      %
1600      % DETERMINE PERCENT OF CREW SIZE AVAILABLE USING PRIMARY
1610      % SKILL
1620      %
1630      115 CALL PERAV(KSDEL,PERCTA)
1640      IF (NSSKIL.EQ.0) GO TO 116
1650      IF(RPOOL(ISG,2).EQ.0) GO TO 116
1660      %
1670      % INCREASE CREW SIZE PERCENTAGE BY USING SECONDARY SKILLS
1680      % IF POSSIBLE
1690      %
1700      CALL SSKILA(PERCTA,PERCTB,KSDEL)
1710      GO TO 118
1720      %
```

```
      % POST IN TEMPORARY TABLE
      %
      116 PERCTB=PERCTA
      118 IPOST=IPOST+1
      CALL TPOST(MODEL,PERCTA,PERCTB)
      IF(MINPER.GT.TEMPRT(IPOST,5))MINPER=TEMPRT(IPOST,5)
      120 CONTINUE
      %
      % IS THERE ANOTHER CONCURRENT ACTIVITY?
      %
      IF(JCA.GT.ICA.OR.ICA.EQ.0) GO TO 125
      IF(JCA.EQ.0) JCA=1
      121 JS=CONACT(JCA,1)
      IF(ACTIV(JS,10).EQ.NC) GO TO 122
      JCA=JCA+1
      IF(JCA.LT.ICA) GO TO 121
      122 IF(IS.NE.CONACT(JCA,2)) GO TO 123
      JCA=JCA+1
      IF(JCA.GT.ICA) GO TO 125
      GO TO 124
      123 IF(IS.NE.CONACT(JCA,1)) GO TO 125
      GO TO 126
      124 IF(CONACT(JCA-1,1).EQ.CONACT(JCA,1)) GO TO 126
      %
      % ADJUST RESOURCES TO SAME LEVEL
      %
      125 CALL ADJUST(MINPER)
      %
      % POST RESOURCES
      %
      CALL POST
      IPOST=0
      126 CONTINUE
      RETURN
      END
```

```
100       SUBROUTINE SUB3
110       %
120       %
130       %
140       DIMENSION LIST(1000)
150       COMMON ACTIV(1000,13),MSKILL(1000,4),CONACT(50,2),%
160       RPOOL(100,2),TEMPRT(20,5),NAN,NAS,IS,IMS,ICA,NCA,ISG,NSG,%
170       IPOST,NPER,KSLACK
180       INTEGER ACTIV,RPOOL,CONACT,TEMPRT,PSSKIL,PCTA
190       %
200       %
210       %-----------------------------------------------------------------
220       ENTRY SSKIL(MAVAIL,ISSKIL,MSSKIL,KSZ)
230       %
240       % DETEMINES IF RESOURCES ARE AVAILABLE THROUGH
250       % USING SECONDARY SKILLS
260       %
270       ISSKIL=RPOOL(ISG,2)
280       IF(ISSKIL.EQ.0) GO TO 88
290       MTOTAL=MAVAIL+RPOOL(ISSKIL,1)
300       IF(MTOTAL.LT.KSZ) GO TO 88
310       MSSKIL=KSZ-MAVAIL
320       GO TO 90
330       88 MSSKIL=0
340       90 RETURN
350       %
360       %
370       %-----------------------------------------------------------------
380       ENTRY TPOST(I1,I2,I3)
390       %
400       % POSTS RESOURCE INFORMATION IN TEMPORARY TABLE
410       %
420       TEMPRT(IPOST,1)=IS
430       TEMPRT(IPOST,2)=ISG
440       TEMPRT(IPOST,3)=I1
450       TEMPRT(IPOST,4)=I2
460       TEMPRT(IPOST,5)=I3
470       RETURN
480       %
```

```
      %
      %-----------------------------------------------------------------
      ENTRY POST
      %
      % POSTS RESOURCES AND ADJUSTS RESOURCE POOL
      %
      % DETERMINE IF RESOURCES ARE AVAILABLE FOR POSTING
      %
      DO 92 I= 1,IPOST
      IF(TEMPRT(I,3).EQ.0.AND.TEMPRT(I,5).EQ.0) RETURN
      92 CONTINUE
      %
      % ASSIGN RESOURCES IN ACTIV ARRAY AND ADJUST RPOOL ARRAY
      % FOR MAIN SKILL
      %
      I=1
      93 IS=TEMPRT(I,1)
      ACTIV(IS,8)=ACTIV(IS,8)+TEMPRT(I,3)+TEMPRT(I,5)
      ACTIV(IS,9)=ACTIV(IS,9)+TEMPRT(I,5)
      ISG=TEMPRT(I,2)
      RPOOL(ISG,1)=RPOOL(ISG,1)-TEMPRT(I,3)
      ISG=TEMPRT(I,4)
      RPOOL(ISG,1)=RPOOL(ISG,1)-TEMPRT(I,5)
      I=I+1
      %
      % ASSIGN RESOURCES IN MSKILL ARRAY AND ADJUST RPOOL ARRAY
      % FOR OTHER SKILLS
      %
      NMS=ACTIV(IS,13)
      N=1
      94 IF(N.GT.NMS) GO TO 96
      L=ACTIV(IS,12)+N-1
      MSKILL(L,3)=MSKILL(L,3)+TEMPRT(I,3)+TEMPRT(I,5)
      MSKILL(L,4)=MSKILL(L,4)+TEMPRT(I,5)
      ISG=TEMPRT(I,2)
      RPOOL(ISG,1)=RPOOL(ISG,1)-TEMPRT(I,3)
      ISG=TEMPRT(I,4)
      RPOOL(ISG,1)=RPOOL(ISG,1)-TEMPRT(I,5)
      I=I+1
      N=N+1
      GO TO 94
      96 IF(I.LE.IPOST) GO TO 93
      RETURN
      %
```

```
      %
      %-------------------------------------------------------------
      ENTRY LPACK(LIST,NLIST)
      %
      % DELETES ACTIVITIES FROM SCHEDULING LIST IF RESOURCES CANNO
      % BE OBTAINED AND PACKS REMAINING ACTIVITIES
      %
      ISHIFT=0
      DO 100 N=1,NLIST
      IS=LIST(N)
      J=N-ISHIFT
      LIST(J)=LIST(N)
      IF(ACTIV(IS,8).NE.0) GO TO 100
      ISHIFT=ISHIFT+1
  100 CONTINUE
      IF(ACTIV(IS,8).EQ.0) J=J-1
      NLIST=J
      RETURN
      %
```

```
      %
      %-----------------------------------------------------------------
      ENTRY DELCS(KSDEL1,KS,MS,LOC)
      %
      % CALCULATE CREW SIZE INCREMENT FOR ADDING RESOURCES
      %
      I4=ACTIV(IS,4)
      I5=ACTIV(IS,5)
      I8=ACTIV(IS,8)
      IF(MS.EQ.1) GO TO 125
      %
      % DETERMINE CREW SIZE BASED ON MULTISKILL MANPOWER RATIO
      %
      I4=.01*FLOAT(MSKILL(LOC,2)*I4)+.5
      I5=.01*FLOAT(MSKILL(LOC,2)*I5)+.5
      I8=MSKILL(LOC,3)
  125 IF(KS-2) 126,128,131
      %
      % INCREMENT REQUIRED FOR NORMAL CREW SIZE
      %
  126 KSDEL1=I4-I8
      GO TO 132
      %
      % INCREMENT REQUIRED FOR CRITICAL CREW SIZE
      %
  128 FACTOR=ACTIV(IS,7)+ACTIV(IS,11)
      IF(FACTOR.GT.0.0) GO TO 129
      KCS=I5
      GO TO 130
  129 KCS=FLOAT(I4*ACTIV(IS,7))/((1.+.025*%
      FLOAT(ACTIV(IS,6)))*FACTOR)+0.5
      IF(KCS.GT.I5) KCS=I5
  130 KSDEL1=KCS-I8
      GO TO 132
      %
      % INCREMENT REQUIRED FOR MAXIMUM CREW SIZE
      %
  131 KSDEL1=I5-I8
  132 RETURN
      %
```

```
1520        %
1530        %-------------------------------------------------------------
1540        ENTRY PERAV (KSDEL2,PCTA)
1550        %
1560        % FINDS PERCENT OF RESOURCES AVAILABLE WITHOUT
1570        % USING SECONDARY SKILLS
1580        %
1590        PCTA=100.0*FLOAT(RPOOL(ISG,1))/FLOAT(KSDEL2)+0.5
1600        IF(PCTA.GT.100) PCTA=100
1610        RETURN
1620        %
1630        %
1640        %-------------------------------------------------------------
1650        ENTRY SSKILA(IPCTA,IPCTB,KSDEL3)
1660        %
1670        % ADDS RESOURCES BY USING SECONDARY SKILLS
1680        %
1690        IB=RPOOL(ISG,2)
1700        PSSKIL=100.0*FLOAT(RPOOL(IB,1))/FLOAT(KSDEL3)+0.5
1710        IPCTB=IPCTA+PSSKIL
1720        IF(IPCTB.GT.100)IPCTB=100
1730        RETURN
1740        %
1750        %
1760        %-------------------------------------------------------------
1770        ENTRY ADJUST(MINPER)
1780        %
1790        % ADJUSTS RESOURCES TO SAME LEVEL IN TEMPRT
1800        %
1810        DO 134 I=1,IPOST
1820        NMEN=0.01*FLOAT(MINPER*TEMPRT(I,3))+0.5
1830        IF(TEMPRT(I,4).GT.MINPER)TEMPRT(I,4)=MINPER
1840        TEMPRT(I,3)=0.01*FLOAT(TEMPRT(I,3)*TEMPRT(I,4))+0.5
1850        TEMPRT(I,5)=NMEN-TEMPRT(I,3)
1860        ISG=TEMPRT(I,2)
1870        TEMPRT(I,4)=RPOOL(ISG,2)
1880        134 CONTINUE
1890        RETURN
1900        END
```

```
100       SUBROUTINE SUB4
110       %
120       %
130       %
140       DIMENSION LIST1(1000)
150       COMMON ACTIV(1000,13),MSKILL(1000,4),CONACT(50,2),%
160       RPOOL(100,2),TEMPRT(20,5),NAN,NAS,IS,IMS,ICA,NCA,ISG,NSG,%
170       IPOST,NPER,KSLACK
180       INTEGER ACTIV,CONACT
190       %
200       %
210       %-----------------------------------------------------------
220       ENTRY SORTL(LIST1,N)
230       %
240       % SORTS A LIST OF ACTIVITES IN ASCENDING ORDER OF SLACK FOR
250       % SCHEDULING LISTS
260       %
270       NM1=N-1
280       DO 28 I=1,NM1
290       IP1=I+1
300       DO 28 J=IP1,N
310       IA=LIST1(I)
320       JA=LIST1(J)
330       IF(ACTIV(IA,11).LE.ACTIV(JA,11)) GO TO 28
340       ITEMP=LIST1(I)
350       LIST1(I)=LIST1(J)
360       LIST1(J)=ITEMP
370       28 CONTINUE
380       RETURN
390       %
```

```
400       %
410       %-----------------------------------------------------------------
420       ENTRY SORTCA
430       %
440       % SORTS CONACT ARRAY IN ASCENDING ORDER OF SLACK
450       %
460       ICAM1=ICA-1
470       DO 54 I=1,ICAM1
480       IP1=I+1
490       DO 54 J=IP1,ICA
500       IA=CONACT(I,1)
510       JA=CONACT(J,1)
520       IF(ACTIV(IA,11).LE.ACTIV(JA,11)) GO TO 54
530       ITEMP1=CONACT(I,1)
540       ITEMP2=CONACT(I,2)
550       CONACT (I,1)=CONACT(J,1)
560       CONACT(I,2)=CONACT(J,2)
570       CONACT(J,1)=ITEMP1
580       CONACT(J,2)=ITEMP2
590       54 CONTINUE
600       RETURN
610       %
620       %
630       %-----------------------------------------------------------------
640       ENTRY SORTSK
650       %
660       % SORTS CONACT ARRAY IN ASCENDING ORDER OF SERIAL NUMBERS
670       %
680       ICAM1=ICA-1
690       DO 55 I=1,ICAM1
700       IP1=I+1
710       DO 55 J=IP1,ICA
720       IF(CONACT(J,1).GE.CONACT(I,1)) GO TO 55
730       ITEMP1=CONACT(I,1)
740       ITEMP2=CONACT(I,2)
750       CONACT(I,1)=CONACT(J,1)
760       CONACT(I,2)=CONACT(J,2)
770       CONACT(J,1)=ITEMP1
780       CONACT(J,2)=ITEMP2
790       55 CONTINUE
800       RETURN
810       END
```

```
EDIT DAT8
↑L
     1        0,0,0,0,0,0
     2        1,4,6,8,0,0
     3        2,3,4,6,0,0
     4        6,2,3,5,0,0
     5        3,2,4,6,0,0
     6        4,2,2,3,0,0
     7        5,5,7,10,0,0
     8        0,0,0,0,0,0
     9        2,5,8,10,0,0
    10        3,3,4,4,0,1
    11        4,2,3,4,0,0
    12        5,2,2,4,0,0
    13        6,4,6,9,0,0
    14        1,4,7,10,0,2
    15        2,4,4,6,0,0
    16        0,0,0,0,0,0
↑Q
SRU'S:.4
!EDIT DAT9
↑L
     1        10,1,50
     2        14,3,57
     3        14,5,86
↑Q
SRU'S:.3
!EDIT DAT10
↑L
     1        10,11
     2        10,12
     3        14,15
↑Q
SRU'S:.3
!EDIT DAT11
↑L
     1        6
     2        0
     3        0
     4        0
     5        0
     6        0
     7        4
     8        5
     9        4
    10        4
    11        4
    12        3
    13        4
    14        5
    15        4
    16        5
    17        5
    18        5
↑Q
SRU'S:.4
```

NOTE: THIS PROGRAM REQUIRES THE PREPARATION OF UNFORMATTED INPUT DATA FILES

```
!EDIT DAT13
↑L
     1       2,6,3,0,0,0,0
↑Q
SRU'S:.3
!EDIT DAT14
↑L
     1       0,0,0
     2       1,1,4
     3       1,2,2
     4       2,3,1
     5       2,5,5
     6       2,7,4
     7       2,9,1
     8       2,11,0
     9       0,0,1
    10       1,1,5
    11       1,2,5
    12       1,3,5
    13       2,4,1
    14       2,6,6
    15       2,8,6
    16       3,10,0
↑Q
SRU'S:.4
!EDIT DAT15
↑L
     1       2
     2       8
     3       14
     4       1
     5       1
     6       2
     7       3
     8       2
     9       3
    10       4
    11       5
    12       4
    13       5
    14       6
    15       7
    16       8
    17       12
    18       1
    19       1
    20       1
    21       2
    22       3
    23       2
    24       3
    25       3
    26       4
    27       5
    28       6
    29       7
↑Q
SRU'S:.5
```

NOTE: THIS PROGRAM REQUIRES THE PREPARATION OF UNFORMATTED INPUT DATA FILES

REFERENCES

1. Malcolm, D. G., Roseboom, J. H., Clark, C. E., and Frazar, W., 1959, Application of a technique for research and development program evaluation: Operation Research, v. 7, p. 646-669.

2. Kelley, James E. Jr., and Walker, Morgan R., 1959, Critical path planning and scheduling: Proc. of the Eastern Joint Computer Conference, p. 160-173.

3. Paulson, Boyd C. Jr., 1971, Man-computer concepts for project management: Stanford Univ., Construction Inst. Tech. Rept., 148, p. 43-53.

4. Herroelen, Willy S., 1972, Resource-constrained project scheduling - the state of the art: Operations Research Quart., v. 23, no. 3, p. 261-275.

5. Wiest, Jerome D., 1967, A heuristic model for scheduling large projects with limited resources: Management Sci., v. 13, no. 6, p. B359-B377.

6. Moder, Joseph J., and Phillips, Cecil R., 1964, Project management with CPM and PERT: New York, Reinhold, 283 p.