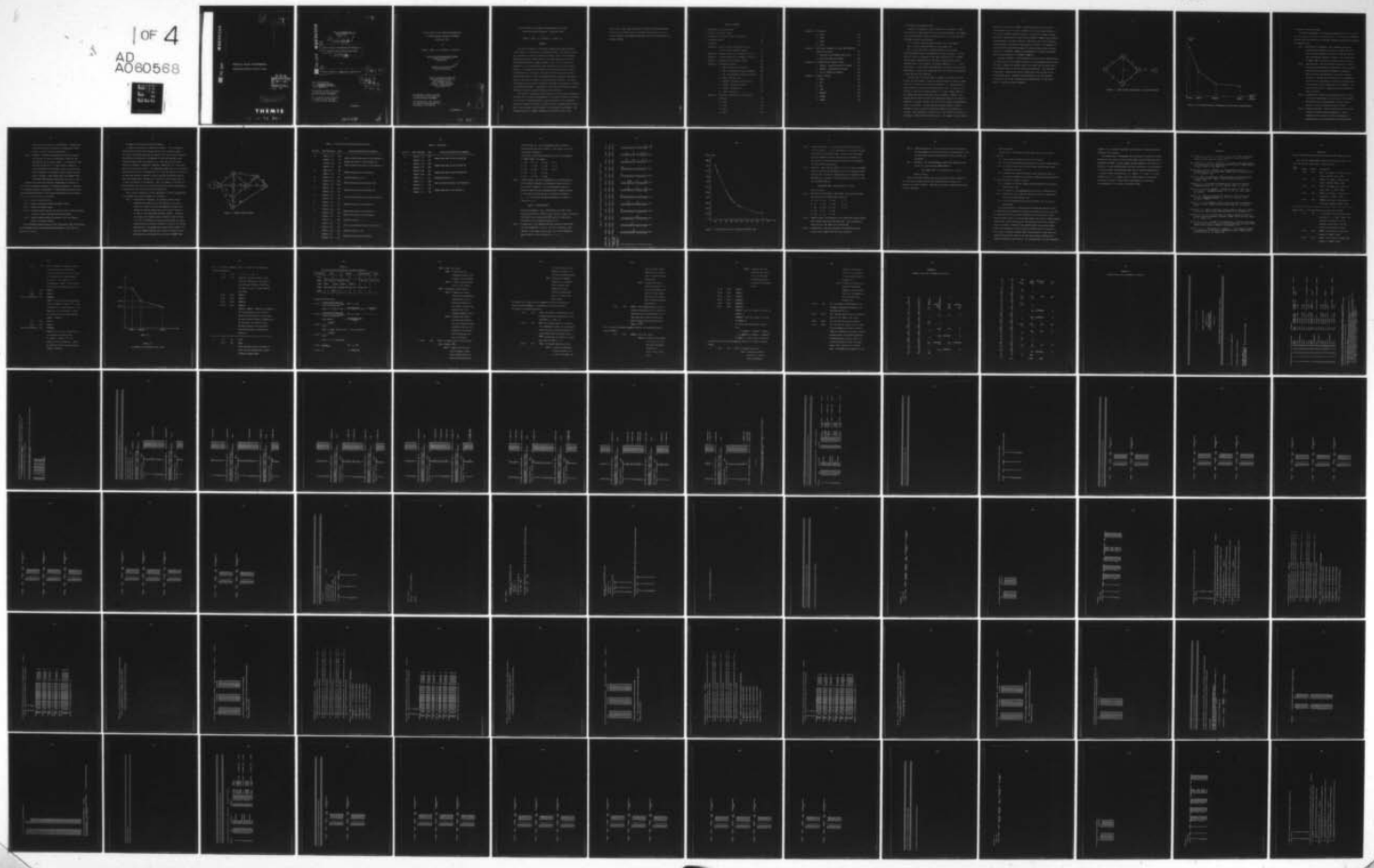
AD-A060 568 TEXAS A AND M UNIV COLLEGE STATION INST OF STATISTICS F/G 9/2
A USER'S GUIDE TO THE COMPUTER IMPLEMENTATION OF THE NEW PROJEC--ETC(U)
AUG 78 T C BAKER, R L SIELKEN
UNCLASSIFIED THEMIS-TR-57 N00014-78-C-0426
NL

1 OF 4
AD A060568



AD A060568

DDC FILE COPY

(12)

LEVEL II

TEXAS A&M UNIVERSITY

COLLEGE STATION, TEXAS 77843

DDC
RECEIVED
NOV 1 1978
A

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

THEMIS

78 10 31 026

AD A060568

DDC FILE COPY

TEXAS A&M UNIVERSITY
PROJECT THEMIS - TR-

(14) (9) Technical Report, 57

(6) A USER'S GUIDE TO THE COMPUTER IMPLEMENTATION OF THE NEW PROJECT SCHEDULING PROCEDURE: STATISTICAL PERT.

by

(10) Thomas C. Baker, Jr. and Robert L. Sielken Jr.

(11) Aug 78

(12) 303p.

Texas A&M University
Office of Naval Research
Contract (15) N00014-78-C-0426
Project NR047-179

Reproduction in whole or in part
is permitted for any purpose of
the United States Government.

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

DDC
RECEIVED
NOV 1 1978
A

ATTACHMENT I

347 380

LB

A USER'S GUIDE TO THE COMPUTER IMPLEMENTATION OF THE NEW PROJECT SCHEDULING PROCEDURE: STATISTICAL PERT

by

Thomas C. Baker, Jr. and Robert L. Sielken Jr.

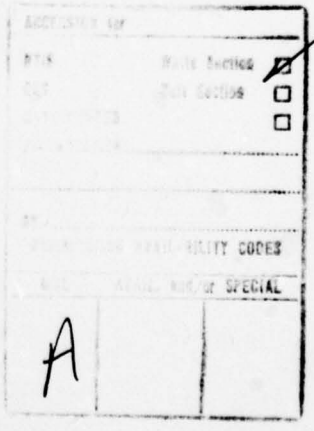
THEMIS OPTIMIZATION RESEARCH PROGRAM
Technical Report No. 57
August 1978

INSTITUTE OF STATISTICS
Texas A&M University

Research conducted through the
Texas A&M Research Foundation
and sponsored by the
Office of Naval Research
Contract N00014-78-C-0426
Project NR047-179

Reproduction in whole or in part
is permitted for any purpose of
the United States Government.

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



ATTACHMENT II

31 026

A User's Guide to the Computer Implementation of the New Project Scheduling Procedure: Statistical PERT

by

Thomas C. Baker, Jr. and Robert L. Sielken Jr.

ABSTRACT

This report documents a new project scheduling procedure developed at the Institute of Statistics, Texas A&M University. The project scheduling algorithm is a five step iterative procedure capable of determining a minimum cost project schedule when the activities making up the project have durations which are random variables. The cost of an activity is assumed to be a convex piecewise linear function of the activity's mean duration. The problem is to determine the activity mean durations which both minimize the total project cost and insure that the mean (or some specified percentile) of the corresponding project completion time distribution is less than or equal to a specified project deadline. The entire distribution of the project's completion time under the minimum cost schedule is a valuable by-product. Information on the trade-off between the project's minimum cost and its specified deadline is also provided.

This report is intended as a user's guide to the new project scheduling procedure and its computer implementation. The report includes a description of the project scheduling problem, a general overview of the scheduling procedure including references to technical reports documenting the development of the procedure, and an example of the procedure's performance. The documentation of the computer implementation includes specific input

instructions; sample input and output; flowcharts; individual program descriptions; technical details concerning temporary data sets, job control language, and program interruption and restart procedures; and program listings.

TABLE OF CONTENTS

1. The Project Scheduling Problem 1

2. Outline of Statistical PERT 5

3. An Example of Statistical PERT's Performance 7

4. Concluding Remarks 16

References 18

Appendix A. Specific Input Instructions for the Statistical PERT Computer System 19

Appendix B. Computer Input for the Example in Section 3 29

Appendix C. Computer Output for the Example in Section 3 31

Appendix D. Flowchart of the Computer System 107

Appendix E. Program Descriptions 108

1. MAIN: Original Input Program 108

2. DPS: Determinisitc Project Scheduling 109

3. DSR: Deterministic Schedule Resolution 110

4. SIMP: Simplification 111

5. MODSIMP: Subsequent Simplification 115

6. SUBNET: Subnetwork Analysis 116

7. DECOMP: Decomposition 136

8. SYNTH: Synthesis 139

Appendix F. Structure and Contents of the Temporary Data Sets 142

1. FILE0 143

2. FILE1 145

3. FILE2 147

4. FILE3 148

Appendix F. (Continued)

5. FILE4 150

6. FILE5 151

7. FILE6 153

8. FILE7 154

Appendix G. Job Control Language for Texas A&M University Computer Facilities 157

1. Execution using FORTRAN source decks 157

2. Execution using load modules 160

Appendix H. Interrupting the Computer System 162

1. SAVEFIL: System Interruption Program 164

2. READFIL: System Restart Program 165

3. JCL for SAVEFIL and READFIL 166

Appendix I. Program Listings 168

1. MAIN 169

2. DPS 175

3. DSR 187

4. SIMP 192

5. MODSIMP 200

6. DECOMP 227

7. SUBNET 242

1. The Project Scheduling Problem

A project is composed of several "tasks" or "activities." These activities can be represented by arcs in a directed network. For example, a small project might consist of activities A, B, C, D, and E with the following precedence relationships:

(i) A must be completed before either C or D can be begun;

(ii) B must be completed before D can be begun; and

(iii) C and D must both be completed before E can be begun.

The corresponding network representation is shown in Figure 1. The arc labeled F does not correspond to any "real" activity but is a "dummy" activity merely representing the precedence relation that A must be completed before D can be begun. The points numbered 1, 2, ..., 5 are called nodes. In the network representation of a project the activities originating at a node can be begun only after all activities terminating at that node have been completed.

The time that it actually takes to complete an activity once that activity has been begun is called the activity's duration and is a random variable. The cost of an activity is assumed to be a convex piecewise linear function of the activity's mean duration time. An example of an activity's cost curve is given in Figure 2. In this example TIME(1) is the minimum mean duration time that can be scheduled. TIME(4) is the cheapest mean duration and hence the maximum mean duration that would be scheduled. Of course a linear cost curve is the simplest convex piecewise linear cost function. The more general piecewise behavior, however, frequently arises if there are alternative methods of performing an activity. These methods do not differ in the end result but do differ in the amount of time they take and their cost. For example, to have a mean

duration in the interval [TIME(1), TIME(2)] might require the use of a very expensive piece of special equipment while having a mean duration in the interval [TIME(2), TIME(3)] requires only specially trained personnel and having a mean duration in the interval [TIME(3), TIME(4)] just requires varying amounts of standard resources. The form of the activity duration distribution may vary from one time interval to another. For example, the activity duration distribution might be a beta distribution when the mean duration is in [TIME(1), TIME(3)] and approximately a normal distribution when the mean duration is in [TIME(3), TIME(4)].

A project schedule is a specification of each activity's mean duration. The total project cost is simply the sum of the corresponding activity costs. The time to complete the entire project is a random variable whose distribution depends upon the activity duration distributions. The objective is to determine a minimum cost project schedule such that the mean of the corresponding project completion time distribution (or some specified percentile of the project completion time distribution) is less than or equal to a specified project deadline.

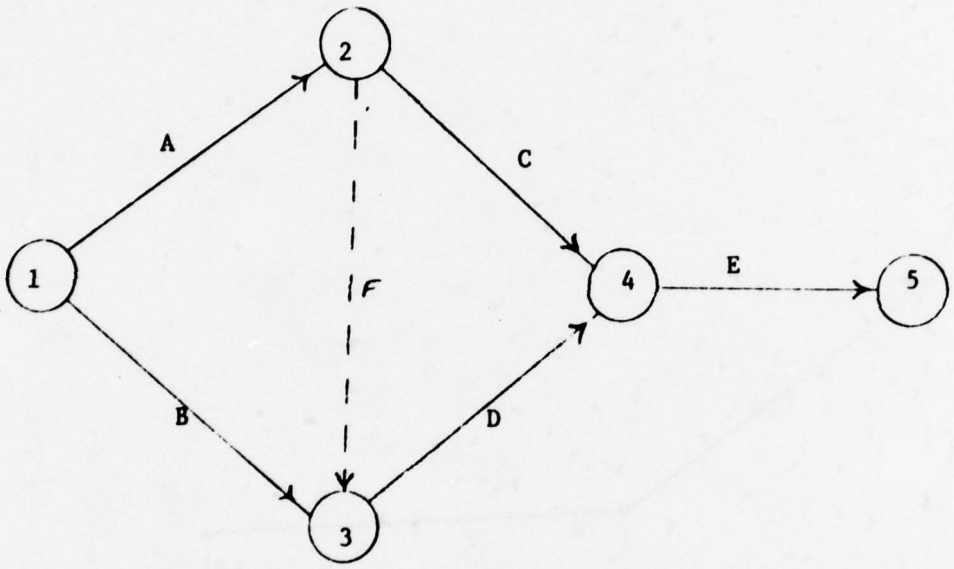


Figure 1. A Small Project Represented as a Directed Network

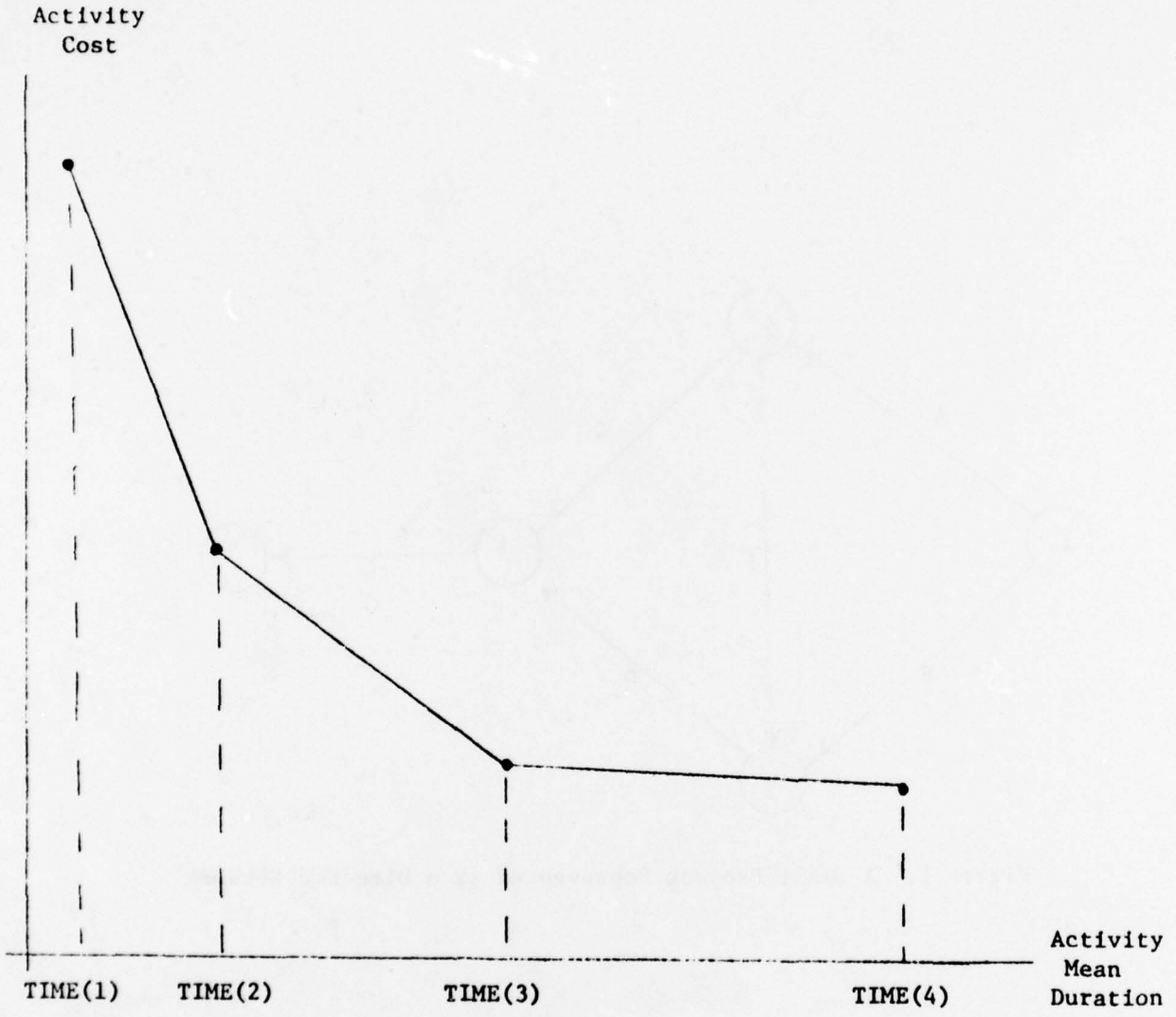


Figure 2. An Activity's Cost as a Function of the Activity's Mean Duration

2. Outline of Statistical PERT

In 1974 the development of a new project scheduling procedure was begun with the support of the Office of Naval Research. The new project scheduling procedure that has resulted is an iterative algorithm involving the following five general steps:

Step 1. Deterministic Scheduling: Find a minimum cost project schedule which completes the project by TARGET TIME when each activity's duration is exactly its mean duration and hence deterministic instead of random. (The initial value of TARGET TIME is usally the specified project deadline.)

Step 2. Simplification: Let each activity's duration be a random variable with distribution corresponding to that activity's mean duration chosen during Deterministic Scheduling. Replace various configurations of activities by single activities. The duration distribution for a replacement activity is the distribution of the time to complete all of the activities in the configuration it is replacing. The result of this step is a simplified project network with fewer activities.

Step 3. Decomposition: Partition the simplified project network into several subnetworks in such a way that the resultant subnetworks can be linked together in either series or parallel to form the simplified project network.

Step 4. Subnetwork Analysis: Analyze separately each of the subnetworks determined during Decomposition. Within a subnetwork each activity's duration distribution is approximated by a two-point discrete distribution with

matching mean, variance, and third moment. Determine the subnetwork duration distribution corresponding to these discrete activity duration distributions.

Step 5. Synthesis: Combine the approximate subnetwork duration distributions to obtain an approximate completion time distribution for the entire project. If the mean (or some specified percentile), $\hat{T}$, of this project completion time distribution is sufficiently close to the specified project deadline, the "optimal" project schedule has been found. Otherwise, reset TARGET TIME to New TARGET TIME = Old TARGET TIME* (Project Deadline/$\hat{T}$) and return to Step 1.

A general discussion and relatively nonmathematical overview of this project scheduling procedure is contained in Appendix D, "Flowchart of the Computer System," and Appendix E, "Program Descriptions"(see also Sielken and Hartley (1977)). The detailed documentation of the development thus far of each step is as follows:

Step 1. Dunn and Sielken (1977);

Step 2. Hartley and Wortham (1966) and Ringer (1969);

Step 3. Sielken and Fisher (1976);

Step 4. Sielken, Ringer, Hartley, and Arseven (1974), Sielken, Hartley, and Spoeri (1976), and Baker and Sielken (1978);

Step 5. Sielken, Ringer, Hartley and Arseven (1974) and Sielken, Hartley, and Spoeri (1976).

The forthcoming Technical Report No. 60, "Statistical PERT: The Precision of the Estimated Project Completion Time Distribution," will also be of interest to the user.

3. An Example of Statistical PERT's Performance

A small project network is depicted in Figure 3. The relationship between each activity's mean duration and its cost is given in Table 1. The project scheduling algorithm also requires that the activity's duration distribution be specified at the midpoint of each time interval on the convex piecewise linear cost function, i.e., when the activity's mean duration is [TIME(1) + TIME(2)]/2, [TIME(2) + TIME(3)]/2, etc. This information is also given in Table 1. The algorithm assumes that if an activity's mean duration is not at the midpoint but at c times the midpoint and still in the same time interval, then the activity's duration distribution has the same form (Normal, Beta, Constant, etc.) but with a new variance equal to c^2 times the variance at the midpoint. Thus, for example, if activity A's mean duration is 28, its cost is 34, and activity A's duration distribution is Beta [10,40] with mean 28 and variance $(28/25)^2 36$.

With a project deadline of 125 the algorithm's iterative determination of the minimum cost project schedule is as follows:

Step 1. Deterministic Scheduling: The shortest feasible project completion time when each activity duration is its mean duration is found to be 90 by determining the longest path through the project network when each activity's duration is equal to its minimum mean duration, TIME(1). Similarly, the longest such feasible project completion time is found to be 135 by determining the longest path through the project network when each activity's duration is equal to its maximum mean duration. The minimum cost schedule which completes the project by TARGET TIME when each activity's duration is its mean duration is determined for each value of TARGET TIME

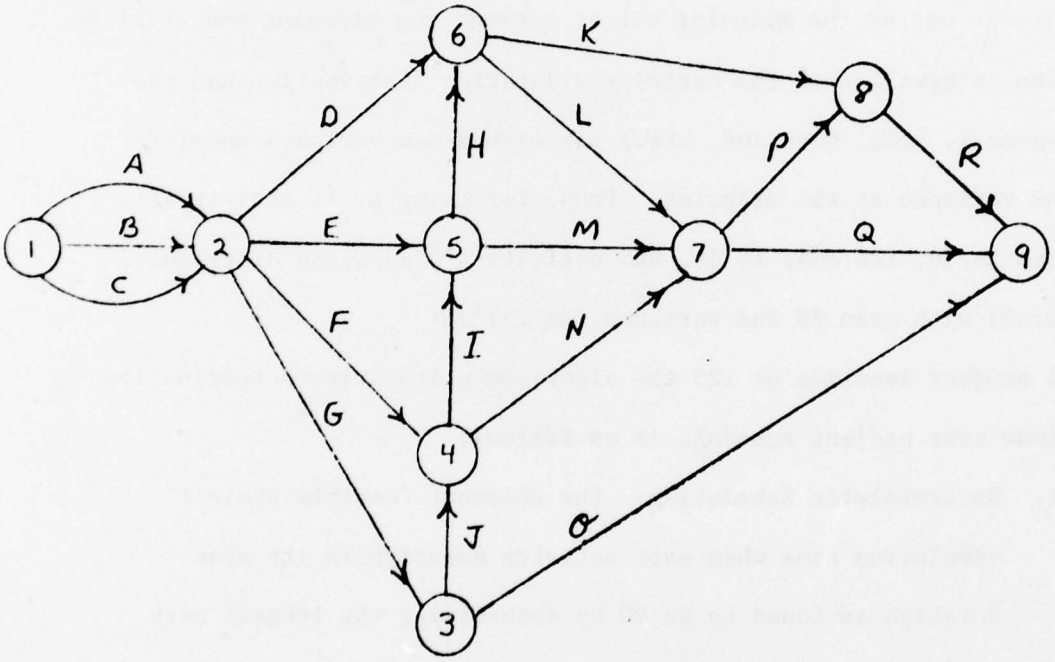


Figure 3. Example Project Network

Table 1. Activity Costs and Duration Distributions

Activity	Mean Durations	Cost	Duration Distribution at Midpoint
A	TIME(1) = 10	100	Beta on [5,20] with mean 12.5 and variance 16
	TIME(2) = 15	70	Normal with mean 17.5 and variance 25
	TIME(3) = 20	50	Beta on [10,40] with mean 25 and variance 36
	TIME(4) = 30	30	
B	TIME(1) = 8	60	Normal with mean 11 and variance 9
	TIME(2) = 14	45	
C	TIME(1) = 10	50	Constant Duration = 10
D	TIME(1) = 50	200	Normal with mean 60 and variance 100
	TIME(2) = 70	160	
E	TIME(1) = 32	45	Normal with mean 36 and variance 18
	TIME(2) = 40	40	
F	TIME(1) = 20	64	Beta on [10,50] with mean 26 and variance 70
	TIME(2) = 32	48	
G	TIME(1) = 13	30	Normal with mean 16 and variance 7
	TIME(2) = 19	25	
H	TIME(1) = 20	60	Normal with mean 25 and variance 12
	TIME(2) = 30	50	Normal with mean 32.5 and variance 9
	TIME(3) = 35	48	
I	TIME(1) = 5	60	Constant Duration = 5
J	TIME(1) = 18	62	Beta on [10,40] with mean 22 and variance 6
	TIME(2) = 26	49	
K	TIME(1) = 10	40	Constant Duration = 10
L	TIME(1) = 6	75	Normal with mean 8 and variance 5
	TIME(2) = 10	50	

Table 1. (Continued)

Activity	Mean Durations	Cost	Duration Distribution at Midpoint
M	TIME(1) = 36	225	Normal with mean 40 and variance 80
	TIME(2) = 44	175	
N	TIME(1) = 30	400	Normal with mean 35 and variance 25
	TIME(2) = 40	300	
O	TIME(1) = 60	250	Normal with mean 70 and variance 144
	TIME(2) = 80	210	
P	TIME(1) = 4	52	Constant Duration = 4
Q	TIME(1) = 2	100	Beta on [1,6] with mean 3 and variance 1
	TIME(2) = 4	90	
R	TIME(1) = 4	110	Normal with mean 5 and variance 2
	TIME(2) = 6	80	

between 90 and 135. The corresponding optimal activity mean durations are given in Table 2. The project cost curve is depicted in Figure 4.

The initial activity mean durations are those corresponding to TARGET TIME = 125; namely,

A = 30,	F = 32,	K = 10,	P = 4,
B = 14,	G = 13,	L = 10,	Q = 4,
C = 10,	H = 34,	M = 44,	R = 6.
D = 70,	I = 5,	N = 40,	
E = 40,	J = 23,	O = 80,	

Step 2. Simplification: The only configuration of activities which can be readily replaced by an equivalent single activity is A, B, and C in parallel. If the replacement activity is denoted by ABC, then the simplified project network consists of the single activity ABC and the original activities D, E, ..., R. The initial duration distribution for ABC as a function of t is

$$F_{ABC}(t) = F_A(t)F_B(t)F_C(t)$$

where $F_A(t)$ denotes a beta distribution with mean 30 and variance $(30/25)^2[(40-30)^2/36]$, $F_B(t)$ denotes a normal distribution with mean 14 and variance $(14/11)^2 9$, and $F_C(t)$ denotes the distribution for a constant duration of 10.

Step 3. Decomposition: The simplified project network is partitioned into two subnetworks in series. The first subnetwork, SUB_1, consists of the single activity ABC. The second subnetwork, SUB_2, consists of the activities D, E, ..., R.

Table 2. Optimal Activity Mean Durations for All Feasible TARGET TIME's

TARGET TIME	135-D	134-D	128-D	124-D	114-D	113-D	108-D	106-D	105-D	104-D	100-D	96-D	92-D
Range of D	$0 \le D \le 1$	$0 \le D \le 6$	$0 \le D \le 4$	$0 \le D \le 10$	$0 \le D \le 1$	$0 \le D \le 5$	$0 \le D \le 2$	$0 \le D \le 1$	$0 \le D \le 1$	$0 \le D \le 4$	$0 \le D \le 4$	$0 \le D \le 4$	$0 \le D \le 2$
Project Cost When D=0	1552.0	1552.4	1557.4	1563.9	1583.9	1586.1	1606.1	1614.6	1620.2	1626.2	1660.2	1702.8	1747.8
Activity													
A	30	30	30	30-D	20	20-D	15	15	15-D	14-D	10	10	10
B	14	14	14	14	14	14	14	14	14	14-D	10	10	10
C	10	10	10	10	10	10	10	10	10	10	10	10	10
D	70	70	70	70	70	70	70	70	70	70	70-D	66-D	62
E	40	40	40	40	40-D	39	39-D	37-D	36	36	36	36	36
F	32	32	32	32	32	32	32	32-D	31	31	31	31	31
G	19	19-D	13	13	13	13	13	13	13	13	13	13	13
H	35-D	34	34	34	34	34	34	34	34	34	34-D	30-D	26
I	5	5	5	5	5	5	5	5	5	5	5	5	5
J	26	26	26-D	22	22-D	21	21-D	19-D	18	18	18	18	18
K	10	10	10	10	10	10	10	10	10	10	10	10	10
L	10	10	10	10	10	10	10	10	10	10	10	10	10
M	44	44	44	44	44	44	44	44	44	44	44-D	40-D	36
N	40	40	40	40	40	40	40	40	40	40	40	40	40
O	80	80	80	80	80	80	80-D	78-D	77	77	77-D	73-D	69-D
P	4	4	4	4	4	4	4	4	4	4	4	4	4
Q	4	4	4	4	4	4	4	4	4	4	4	4	4
R	6	6	6	6	6	6	6	6	6	6	6	6	6-D

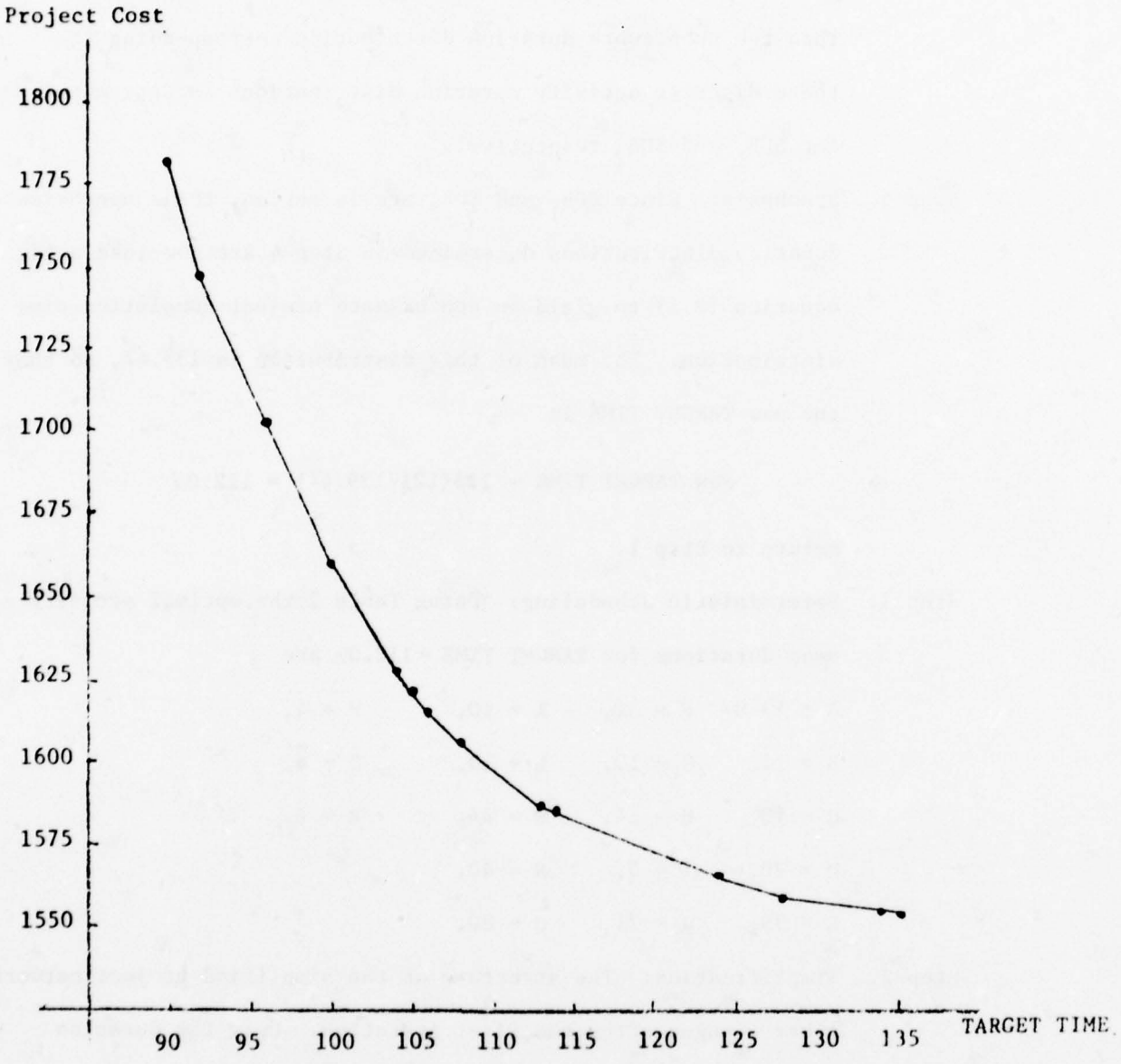


Figure 4. Total Project Cost as a Function of TARGET TIME

Step 4. Subnetwork Analysis: The activity duration distributions for activities ABC,D, ..., R are approximated by two-point discrete distributions with matching mean, variance, and third moment. Then the subnetwork duration distribution corresponding to these discrete activity duration distributions is determined for SUB_1 and SUB_2 respectively.

Step 5. Synthesis: Since SUB_1 and SUB_2 are in series, their approximate duration distributions determined in Step 4 are combined using equation (8.1) to yield an approximate project completion time distribution. The mean of this distribution is 139.42, so that the new TARGET TIME is

$$\text{New TARGET TIME} = 125(125/139.42) = 112.07$$

Return to Step 1.

Step 1. Deterministic Scheduling: Using Table 2 the optimal activity mean durations for TARGET TIME = 112.09 are

A = 19.07	F = 32,	K = 10,	P = 4,
B = 14,	G = 13,	L = 10,	Q = 4,
C = 10,	H = 34,	M = 44,	R = 6,
D = 70,	I = 5,	N = 40,	
E = 39,	J = 21,	O = 80,	

Step 2. Simplification: The structure of the simplified project network never changes after the first iteration. Only the duration distribution of ABC needs to be redetermined.

Step 3. Decomposition: The partitioning of the simplified project network never changes after the first iteration.

Step 4. Subnetwork Analysis: The new activity duration distributions are approximated by new two-point discrete distributions. Then new approximate duration distributions for SUB_1 and SUB_2 are determined.

Step 5. Synthesis: The new approximate project mean completion time is 128.42, so that the new TARGET TIME is

$$\text{New TARGET TIME} = 112.07(125/128.42) = 109.09.$$

Return to Step 1.

If 128.42 was sufficiently close to the specified project deadline of 125, then these optimal activity mean durations would constitute the minimum cost project schedule. Otherwise, the project scheduling algorithm could be continued.

4. Concluding Remarks

The new project scheduling procedure allows the project scheduler to specify

(i) the precedences among the project's activities,

(ii) the relationship between an activity's cost and its mean duration,

(iii) the manner in which an activity's actual duration varies about its mean duration, and

(iv) a deadline for either the project's mean completion time or a prescribed percentile of the project completion time distribution.

In return the project scheduler receives

(i) a minimum cost project schedule which delineates each activity's mean duration time,

(ii) an *estimate* of the distribution of the project completion time,

(iii) information on the trade-off between the project's minimum cost and its specified deadline, and

(iv) a tool for monitoring the project's progress and, if need be, rescheduling.

An exciting feature of this new project scheduling procedure is that it simultaneously incorporates the desire to minimize the project cost and the realization that an activity's duration is not necessarily a fixed quantity exactly equal to its prescribed duration but rather a random quantity varying about a prescribed duration. No longer must the project scheduler either (i) choose a reasonable cost schedule which heuristically hedges against the randomness in the activities he guesses will be critical, or (ii) choose a reasonable schedule which should probably finish before the deadline and then guess where he can save money without disturbing the suspected completion time too much. By considering both cost and randomness

together in one systematic algorithm, the new project scheduling procedure eliminates this guesswork.

The authors wish to acknowledge their gratitude to the Office of Naval Research for the support of this research under contracts N00014-68-A-0140 and N00014-76-C-0038. Several present and past members of the Institute of Statistics at Texas A&M University have also contributed to the development of this new project scheduling procedure and its computer implementation: E. Arseven (Lederle Laboratories), P. P. Biemer, C. S. Dunn, N. E. Fisher (Compucon Inc.), L. J. Ringer, and R. K. Spoeri (Bureau of the Census). The authors also want to particularly acknowledge the considerable contributions of H. O. Hartley to Statistical PERT.

References

Baker, Thomas C. Jr. and R. L. Sielken Jr., "Statistical PERT: Improvements in the Determination of the Project Completion Time Distribution", THEMIS Technical Report 58, August 1978.

Baker, Thomas C. Jr. and R. L. Sielken Jr., "Statistical PERT: The Precision of the Estimated Project Completion Time Distribution", THEMIS Project Technical Report No. 60, August 1978.

Biemer, Paul P., and R. L. Sielken Jr., "Incorporating Project Cost Considerations into Stochastic PERT", THEMIS Project Technical Report No. 52, November, 1975.

Dunn, C. S. and R. L. Sielken Jr., "Statistical PERT: An Improved Project Scheduling Algorithm", THEMIS Project Technical Report No. 55, February, 1977.

Fulkerson, D. R., "A Network Flow Computation for Project Cost Curves", Management Science, Vol. 7, No. 2, (January, 1961), pp. 167-178.

Hartley, H. O. and A. W. Wortham, "A Statistical Theory for PERT Critical Path Analysis," Management Science, Vol. 12, No. 10, (June, 1966), pp. 469-481.

Ringer, L. J., "Numerical Operators for Statistical PERT Critical Path Analysis", Management Science, Vol. 16, No. 2, (October, 1969), pp. B-136 - B-143.

Sielken, R. L. Jr. and Norman E. Fisher, "Statistical PERT: Decomposing a Project Network", THEMIS Project Technical Report No. 50, January, 1976.

Sielken, R. L. Jr. and H. O. Hartley, "A New Statistical Approach to Project Scheduling", THEMIS Project Technical Report No. 56, December 1977.

Sielken, R. L. Jr., H. O. Hartley and R. K. Spoeri, "Statistical PERT: An Improved Subnetwork Analysis Procedure", THEMIS Project Technical Report No. 51, January 1976.

Sielken, R. L. Jr., L. J. Ringer, H. O. Hartley and E. Arseven, "Statistical Critical Path Analysis in Acyclic Stochastic Networks: Statistical PERT", THEMIS Project Technical Report No. 48, November, 1974.

Spoeri, R. K., L. J. Ringer and R. L. Sielken Jr., "A Statistical Procedure for Optimization of PERT Network Scheduling Systems", THEMIS Project Technical Report No. 53, April, 1976.

Appendix A

Specific Input Instructions for the Statistical PERT Computer System

Only the main program (MAIN) requires any input data cards. The order and setup of those cards is indicated below.

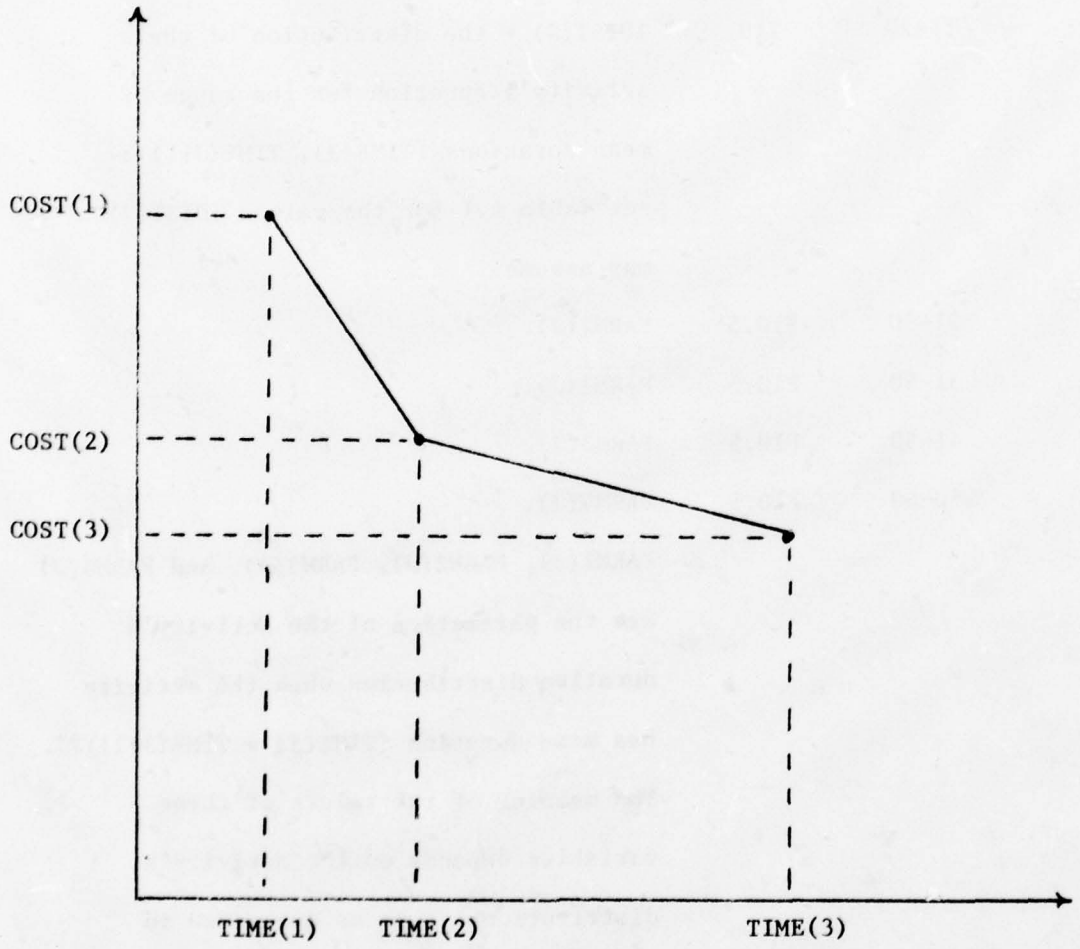
Card Order	Column	Format	Description
1	1-10	I10	NACT = the number of activities in the project network: NACT $\leq$ 1000.
	11-20	I10	NODES = the number of nodes in the project network; NODES $\leq$ 1000.
	21-30	I10	NSRCE = the number of the project's source node; 1 $\leq$ NSRCE $\leq$ 9999.
	31-40	I10	NSINK = the number of the project's sink node; 1 $\leq$ NSINK $\leq$ 9999.
	41-50	I10	LNODEN = the largest node number in the project network; LNODEN $\leq$ 9999.

* *

Repeat cards 2 - 5 for each activity in the project network.

2	1-10	I10	I = the number of the activity being described; 1 $\leq$ I $\leq$ 9999. (Note: The network's activities need not be numbered consecutively, and they need not be described in order.)
	11-20	I10	NODEO = the activity's origin node number; 1 $\leq$ NODEO $\leq$ 9999.
	21-30	I10	NODET = the activity's terminal node number; 1 $\leq$ NODET $\leq$ 9999.

	31-40	I10	NCT = the number of completion times and costs needed to specify this activity's piecewise linear cost curve as a function of its mean duration; $2 \leq NCT \leq 6$. See Figure A.1 for clarification. (Note: For activities with constant or fixed duration, NCT ≡ 2.)
3	1-10	I10	TIME(1).
	11-20	I10	TIME(2).
	⋮		⋮
(NCT-1)*10+1	-NCT*10	I10	TIME(NCT)
			TIME(J) = the J-th activity completion time; $0 \leq TIME(J) < 32768 = 2^{15}$. These times must be in increasing order. See Figure A.1 for clarification. (Note: For activities with constant duration TIME(1) ≡ TIME(2).)
4	1-10	I10	COST(1).
	11-20	I10	COST(2).
	⋮		⋮
(NCT-1)*10+1	-NCT*10	I10	COST(NCT).
			COST(J) = the cost associated with the J-th activity completion time; $0 \leq COST(J) < 32768 = 2^{15}$. See Figure A.1 for clarification. (Note: For activities with constant duration, COST(1) ≡ COST(2).)



(NCT = 3)

Figure A.1

An example of an activity's cost curve.

Card 5 is a "packet" composed of NCT - 1 cards each containing the following information.

5	1-10	I10	J = 1, 2, ..., NCT - 1.
	11-20	I10	IDIST(J) = the distribution of the activity's duration for the range of mean durations (TIME(J), TIME(J+1)); see Table A.1 for the values IDIST(J) may assume.
	21-30	F10.5	PARM1(J).
	31-40	F10.5	PARM2(J).
	41-50	F10.5	PARM3(J).
	50-60	F10.5	PARM4(J). PARM1(J), PARM2(J), PARM3(J), and PARM4(J) are the parameters of the activity's duration distribution when the activity has mean duration (TIME(J) + TIME(J+1))/2. The meaning of the values of these variables depends on the activity's distributional type as described in Table A.1.

* *

6	1-10	I10	TEST1.
	11-20	I10	TEST2. TEST1 and TEST2 control the amount of output from the deterministic project scheduling program (DPS).

Table A.1

Activity Duration Distributions and Their Parameters

Distribution	Beta		Normal		Rectangular	Dummy
IDIST	-1		0		1	2
PARM1	Min Time	Min Time	Min Time	....	Min Time	Fixed Time
PARM2	Mode	Alpha, α	Mean, μ	Mean, μ	....	....
PARM3	Max Time	Max Time	Max Time	St Dev., σ	Max Time	....
PARM4	-1	Beta, β	-1	1	....	....
	a	b	c	d	e	f

Interpreted Distribution:

a) $f(x) = \dfrac{\Gamma(a+b+2)(x-\text{Min})^{a}(\text{Max}-x)^{b}}{\Gamma(a+1)\Gamma(b+1)(\text{Max}-\text{Min})^{a+b+1}}$ $\quad \text{Min} \leq x \leq \text{Max}$

where a, b chosen so that $\mu = \dfrac{\text{Min}+4*\text{Mode}+\text{Max}}{6}$ and $\sigma^2 = \dfrac{(\text{Max}-\text{Min})^2}{36}$

b) $f(x) = \dfrac{\Gamma(\alpha+\beta+2)(x-\text{Min})^{\alpha}(\text{Max}-x)^{\beta}}{\Gamma(\alpha+1)\Gamma(\beta+1)(\text{Max}-\text{Min})^{\alpha+\beta+1}}$, $\quad \text{Min} \leq x \leq \text{Max}$, $\quad \alpha > -1 \quad \beta > -1$

which implies $\quad \mu = \dfrac{\text{Min}(\beta+1)+\text{Max}(\alpha+1)}{(\alpha+\beta+2)}$

c) $f(x) = \dfrac{1}{\sqrt{2\pi}\ s} e^{-\frac{1}{2}\left(\frac{x-\mu}{s}\right)^2}$,

where $s = (\text{Max} - \text{Min})/6$ and $\mu \geq 3s$ is reasonable

d) $f(x) = \dfrac{1}{\sqrt{2\pi}\ \sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$,

where $\mu \geq 3\sigma$ is reasonable

e) $f(x) = \dfrac{1}{(\text{Max}-\text{Min})}$, $\quad \text{Min} \leq x \leq \text{Max}$

f) $f(x) = 1$, $\quad x = \text{FIXED TIME}$

TEST1 = input data option.

TEST1 = 0 implies all the information input to the program is to be printed.

TEST1 = 1 implies the input data is not to be printed.

TEST2 = intermediate output option.

TEST2 = 0 implies the output arising from intermediate calculations is to be printed in addition to the optimal activity mean durations for every feasible completion time.

TEST2 = 1 implies the optimal activity mean durations for every feasible project completion time are to be printed but the intermediate calculations are not to be printed.

21-30	I10	JMAT = an output option in the simplification program (SIMP).

JMAT = 1 implies that only the activity number, origin node, terminal node, and duration distribution for

for each activity in the simplified network is to be printed (minimum output).

JMAT = 0 implies the minimum output plus a listing of which activities were simplified and what activity configurations are to be printed (complete output).

The quantities on card 7 are all parameters used by the subnetwork analysis program (SUBNET). See Appendix D for a more detailed description of Subnetwork Analysis.

7	1-10	I10	IEDF = the number of subdivisions in the estimated duration distribution for each subnetwork; $2 \leq IEDF \leq 20$.
	11-20	I10	NMAX = the maximum number of activities in a subnetwork for which the subnetwork's entire discrete duration distribution is to be automatically enumerated; since 2^{NMAX} calculations are required, a reasonable value for NMAX is ≤ 12.
	21-30	I10	IPOOL = the bound computation option. IPOOL = 0 implies the maximum-cluster procedure is to be used to determine the

upper and lower bounds (F_2^+, F_2^-) on each subnetwork's discrete duration distribution.

IPOOL = 1 implies the union-cluster procedure is to be used to determine the upper and lower bounds (F_1^+, F_2^-) on each subnetwork's discrete duration distribution.

31-40	I10	SAMSIZ = the maximum number of activity duration configurations to be explicitly considered in the determination of the upper and lower bounds on each subnetwork's discrete duration distribution; SAMSIZ $\geq 2^{NMAX}$.

Card 8 contains additional parameters used by the subnetwork analysis program (SUBNET).

8	1-10	I10	THELAM = (θ,λ) pair option.

THELAM = 0 implies the standard (θ,λ) pairs are to be used when determining clusters, i.e., $(\theta,\lambda) = (1,1), (2,2), (3,2)$.

THELAM = 1 implies the three specified (θ, λ) pairs are to be used when determining clusters instead of the standard pairs.

11-20	F10.5	THETA(1).
21-30	F10.5	THETA(2).
31-40	F10.5	THETA(3).
41-50	F10.5	LAMBDA(1).
51-60	F10.5	LAMBDA(2).
61-70	F10.5	LAMBDA(3).

THETA(I) = the I-th θ-value to be used if THELAM = 1.

LAMBDA(I) = the I-th λ-value to be used if THELAM = 1.

The following relations must always be true:

$$0 \leq \text{THETA}(1) \leq \text{THETA}(2) \leq \text{THETA}(3)$$

$$0 \leq \text{LAMBDA}(1) \leq \text{LAMBDA}(2) \leq \text{LAMBDA}(3).$$

If THELAM = 0, these values are ignored.

The quantities on card 9 are parameters used by the synthesis program (SYNTH).

9	1-10	I10	IOPT = a computation option.

IOPT = 0 implies the PCT-th percentile of the network's approximate

completion time distribution is to be compared to the specified project deadline, PD.

IOPT = 1 implies the mean of the network's approximate completion time distribution is to be compared to the specified project deadline, PD.

11-20	I10	NT = the number of subdivisions in the network's synthesized completion time distribution; NT $\leq$ 49.
21-30	F10.5	PCT = the percentile value to be used if IOPT = 0; for example, PCT = 90.0.
31-40	F10.5	PD = the specified project deadline.
41-50	F10.5	TT = the initial target time to be used by the deterministic project scheduler (DPS) in attempting to generate a project schedule whose corresponding completion time distribution has mean (IOPT = 1) or PCT-th percentile (IOPT = 0) equal equal to PD. Usually TT $\leq$ PD works best. A standard value would be TT = PD.

APPENDIX B

Computer Input for the Example in Section 3

```
 18          9          1          9          9
  1          1          2          4
 10         15         20         30
100         70         50         30
  1         -1         5.      .257812      20.      .257812
  2          0                 17.5          5.         1.
  3         -1        10.        25.        40.        -1.
  2          1          2          2
  8         14
 60         45
  1          0                    11.         3.         1.
  3          1          2          2
 10         10
 50         50
  1          2        10.
  4          2          6          2
 50         70
200        160
  1          0                    60.        10.         1.
  5          2          5          2
 32         40
 45         40
  1          0                    36.     4.242641       1.
  6          2          4          2
 20         32
 64         48
  1         -1        10.      .794285      50.      1.691428
  7          2          3          2
 13         19
 30         25
  1          0                    16.     2.645751       1.
  8          5          6          3
 20         30         35
```

```
60          50          48
 1           0                      25.    3.464102          1.
 2           0        23.5          32.5        41.5         -1.
 9           4           5             2
 5           5
60          60
 1           2          5.
10           3           4             2
18          26
62          49
 1          -1         10.           13.         40.          20.
11           6           8             2
10          10
40          40
 1           2         10.
12           6           7             2
 6          10
75          50
 1           0                      8.     2.236068          1.
13           5           7             2
36          44
225         175
 1           0                      40.    8.944272          1.
14           4           7             2
30          40
400         300
 1           0          20.           35.         50.          -1.
15           3           9             2
60          80
250         210
 1           0                      70.         12.          1.
16           7           8             2
 4           4
52          52
 1           2          4.
17           7           9             2
 2           4
100          90
 1          -1          1.            1.          6.           2.
18           8           9             2
 4           6
110          80
 1           0                      5.     1.414214          1.
 1           1           1
20           9           1           500
 0
 1          49                     125.        125.
```

Appendix C

Computer Output for the Example in Section 3

****

S T A T I S T I C A L P E R T

A PROJECT SCHEDULING PROCEDURE

WRITTEN AT THE

INSTITUTE OF STATISTICS
TEXAS A&M UNIVERSITY
COLLEGE STATION, TEXAS 77843

AUGUST 1978

INQUIRIES AND COMMENTS SHOULD BE ADDRESSED TO: ROBERT L. SIELKEN JR.

****

THIS IS THE OUTPUT FROM THE MAIN PROGRAM: MAIN

THE PROJECT NETWORK AS IT WAS READ IN:

THE NUMBER OF NODES IS 9.
THE NUMBER OF ACTIVITIES IS 18.
THE SOURCE NODE IS NUMBERED 1 AND THE SINK NODE IS NUMBERED 9.
THE LARGEST NODE NUMBER IS 9.

ACTIVITY	CRIG	TERM	J	TIME	COST	DISTRIBUTION	PARAMETERS						
1	1	2	1	10	100	BETA	MIN=	5.00000,	MAX=	20.00000,	ALPHA=	0.25781,	BETA= 0.25781
1	1	2	2	15	70	NORMAL	MEAN=	17.50000,	ST.DEV=	5.00000			
1	1	2	3	20	50	BETA	MIN=	10.00000,	MODE=	25.00000,	MAX=	40.00000	
1	1	2	4	30	30								
2	1	2	1	8	60	NORMAL	MEAN=	11.00000,	ST.DEV=	3.00000			
2	1	2	2	14	45								
3	1	2	1	10	50	FIXED(CONSTANT)	TIME=	10.00000					
3	1	2	2	10	50								
4	2	6	1	50	200	NORMAL	MEAN=	60.00000,	ST.DEV=	10.00000			
4	2	6	2	70	160								
5	2	5	1	32	45	NORMAL	MEAN=	36.00000,	ST.DEV=	4.24264			
5	2	5	2	40	40								
6	2	4	1	20	64	BETA	MIN=	10.00000,	MAX=	50.00000,	ALPHA=	0.79428,	BETA= 1.69143
6	2	4	2	32	48								
7	2	3	1	13	30	NORMAL	MEAN=	16.00000,	ST.DEV=	2.64575			
7	2	3	2	19	25								
8	5	6	1	20	60	NORMAL	MEAN=	25.00000,	ST.DEV=	3.46410			
8	5	6	2	30	50	NORMAL	MIN=	23.50000,	MEAN=	32.50000,	MAX=	41.50000	
8	5	6	3	35	48								
9	4	5	1	5	60	FIXED(CONSTANT)	TIME=	5.00000					
9	4	5	2	5	60								
10	3	4	1	18	62	BETA	MIN=	10.00000,	MAX=	40.00000,	ALPHA=	13.00000,	BETA= 20.00000
10	3	4	2	26	49								
11	6	8	1	10	40	FIXED(CONSTANT)	TIME=	10.00000					
11	6	8	2	10	40								
12	6	7	1	6	75	NORMAL	MEAN=	8.00000,	ST.DEV=	2.23607			
12	6	7	2	10	50								
13	5	7	1	36	225	NORMAL	MEAN=	40.00000,	ST.DEV=	8.94427			
13	5	7	2	44	175								
14	4	7	1	30	400	NORMAL	MIN=	20.00000,	MEAN=	35.00000,	MAX=	50.00000	
14	4	7	2	40	300								
15	3	9	1	60	250	NORMAL	MEAN=	70.00000,	ST.DEV=	12.00000			
15	3	9	2	80	210								
16	7	8	1	4	52	FIXED(CONSTANT)	TIME=	4.00000					
16	7	8	2	4	52								
17	7	9	1	2	100	BETA	MIN=	1.00000,	MAX=	6.00000,	ALPHA=	1.00000,	BETA= 2.00000
17	7	9	2	4	90								
18	8	9	1	4	110	NORMAL	MEAN=	5.00000,	ST.DEV=	1.41421			
18	8	9	2	6	80								

THE INFORMATION INPUT TO THE DETERMINISTIC PROJECT SCHEDULING PROGRAM WILL NOT BE PRINTED.
THE INTERMEDIATE OUTPUT GENERATED BY THE DETERMINISTIC PROJECT SCHEDULING PROGRAM WILL NOT BE PRINTED.

THE SIMPLIFICATION PROGRAM WILL PRINT ONLY THE MINIMUM OUTPUT.

DURING THE SUBNETWORK ANALYSIS PROGRAM,
THE NUMBER OF SUBDIVISIONS IN THE ESTIMATED DURATION DISTRIBUTION FOR EACH SUBNETWORK WILL BE 20;
THE LARGEST NUMBER OF ACTIVITIES IN A SUBNETWORK FOR WHICH THE SUBNETWORK'S EXACT DISCRETE DURATION DISTRIBUTION WILL BE ENUMERATED IS 9;
THE MAXIMUM NUMBER OF ACTIVITY DURATION CONFIGURATIONS EXPLICITLY CONSIDERED IN THE DETERMINATION OF THE UPPER AND LOWER BOUNDS ON EACH SUBNETWORK'S DISCRETE DURATION DISTRIBUTION WILL BE 500;
THE UNION-CLUSTER PROCEDURE WILL BE USED IN THE DETERMINATION OF THE BOUNDS;
THE FOLLOWING (THETA,LAMBDA) PAIRS WILL BE USED IN THE FORMATION OF THE CLUSTERS:
(1.00000, 1.00000),(2.00000, 2.00000),(3.00000, 2.00000).

```
THE NUMBER OF SUBDIVISIONS IN THE NETWORK'S SYNTHESIZED COMPLETION TIME DISTRIBUTION WILL BE 49.

THE MEAN OF THE NETWORK'S APPROXIMATE COMPLETION TIME DISTRIBUTION WILL BE COMPARED TO THE
         SPECIFIED PROJECT DEADLINE OF  125.00000.

THE TARGET TIME FOR THE FIRST ITERATION IS  125.00000.


THIS IS THE OUTPUT FROM SUBPROGRAM LOOP. THE PROGRAM SEARCHES THE GIVEN PROJECT NETWORK FOR LOOPS (CYCLES).
A VALID PROJECT NETWORK SHOULD CONTAIN NO LOOPS.


NODE     1 IS NOT PART OF A LOOP
NODE     2 IS NOT PART OF A LOOP
NODE     3 IS NOT PART OF A LOOP
NODE     4 IS NOT PART OF A LOOP
NODE     5 IS NOT PART OF A LOOP
NODE     6 IS NOT PART OF A LOOP
NODE     7 IS NOT PART OF A LOOP
NODE     8 IS NOT PART OF A LOOP
NODE     9 IS NOT PART OF A LOOP
THERE ARE NO LOOPS IN THIS NETWORK
```

```
**********************************************************************************************************************************
**********************************************************************************************************************************

THIS IS THE OUTPUT FROM THE DETERMINISTIC PROJECT SCHEDULER: DPS

**********************************************************************************************************************************
**********************************************************************************************************************************


THE ENTIRE PROJECT COST CURVE WILL BE DETERMINED.

LAMBDA = PROJECT COMPLETION TIME

THE STARTING VALUE OF LAMBDA IS         135.

THE CORRESPONDING TOTAL PROJECT COST IS       0.15520E 04.


DELTA (REPRESENTED BY "D") RANGES FROM 0 TO    1.
LAMBDA RANGES FROM        135 TO        134.
THE MINIMUM COST PROJECT SCHEDULE FOR PROJECT DEADLINE =         135-D:


PROJECT COMPLETION TIME =         135-D.

  ACTIVITY #: I    NEW VALUE: XACT(I)          ACTIVITY COST
        1                     30                0.30000E 02
        2                     14                0.45000E 02
        3                     10                0.50000E 02
        4                     70                0.16000E 03
        5                     40                0.40000E 02
        6                     32                0.48000E 02
        7                     19                0.25000E 02
        8                     35-D              0.48000E 02 + (  0.40000E 00*D)
        9                      5                0.60000E 02
       10                     26                0.49000E 02
       11                     10                0.40000E 02
       12                     10                0.50000E 02
       13                     44                0.17500E 03
       14                     40                0.30000E 03
       15                     80                0.21000E 03
       16                      4                0.52000E 02
       17                      4                0.90000E 02
       18                      6                0.80000E 02

THE CURRENT VALUE OF THE PROJECT COST IS      0.15520E 04 + (  0.40015E 00*D).


DELTA (REPRESENTED BY "D") RANGES FROM 0 TO    6.
LAMBDA RANGES FROM        134 TO        128.
THE MINIMUM COST PROJECT SCHEDULE FOR PROJECT DEADLINE =         134-D:


PROJECT COMPLETION TIME =         134-D.

  ACTIVITY #: I    NEW VALUE: XACT(I)          ACTIVITY COST
        1                     30                0.30000E 02
        2                     14                0.45000E 02
        3                     10                0.50000E 02
        4                     70                0.16000E 03
        5                     40                0.40000E 02
```

```
         6                    32                    0.48000E 02
         7                    19-D                  0.25000E 02 + (  0.83333E 00*D)
         8                    34                    0.48400E 02
         9                     5                    0.60000E 02
        10                    26                    0.49000E 02
        11                    10                    0.40000E 02
        12                    10                    0.50000E 02
        13                    44                    0.17500E 03
        14                    40                    0.30000E 03
        15                    80                    0.21000E 03
        16                     4                    0.52000E 02
        17                     4                    0.90000E 02
        18                     6                    0.80000E 02

THE CURRENT VALUE OF THE PROJECT COST IS      0.15524E 04 + (  0.83333E 00*D).


DELTA (REPRESENTED BY "D") RANGES FROM 0 TO   4.
LAMBDA RANGES FROM       128 TO       124.
THE MINIMUM COST PROJECT SCHEDULE FOR PROJECT DEADLINE =        128-D:


PROJECT COMPLETION TIME =        128-D.

  ACTIVITY #: I   NEW VALUE: XACT(I)          ACTIVITY COST
         1                    30                    0.30000E 02
         2                    14                    0.45000E 02
         3                    10                    0.50000E 02
         4                    70                    0.16000E 03
         5                    40                    0.40000E 02
         6                    32                    0.48000E 02
         7                    13                    0.30000E 02
         8                    34                    0.48400E 02
         9                     5                    0.60000E 02
        10                    26-D                  0.49000E 02 + (  0.16250E 01*D)
        11                    10                    0.40000E 02
        12                    10                    0.50000E 02
        13                    44                    0.17500E 03
        14                    40                    0.30000E 03
        15                    80                    0.21000E 03
        16                     4                    0.52000E 02
        17                     4                    0.90000E 02
        18                     6                    0.80000E 02

THE CURRENT VALUE OF THE PROJECT COST IS      0.15574E 04 + (  0.16250E 01*D).


DELTA (REPRESENTED BY "D") RANGES FROM 0 TO  10.
LAMBDA RANGES FROM       124 TO       114.
THE MINIMUM COST PROJECT SCHEDULE FOR PROJECT DEADLINE =        124-D:


PROJECT COMPLETION TIME =        124-D.

  ACTIVITY #: I   NEW VALUE: XACT(I)          ACTIVITY COST
         1                    30-D                  0.30000E 02 + (  0.20000E 01*D)
         2                    14                    0.45000E 02
         3                    10                    0.50000E 02
         4                    70                    0.16000E 03
         5                    40                    0.40000E 02
         6                    32                    0.48000E 02
```

```
        7                    13                  0.30000E 02
        8                    34                  0.48400E 02
        9                     5                  0.60000E 02
       10                    22                  0.55500E 02
       11                    10                  0.40000E 02
       12                    10                  0.50000E 02
       13                    44                  0.17500E 03
       14                    40                  0.30000E 03
       15                    80                  0.21000E 03
       16                     4                  0.52000E 02
       17                     4                  0.90000E 02
       18                     6                  0.80000E 02

THE CURRENT VALUE OF THE PROJECT COST IS      0.15639E 04 + (  0.20000E 01*D).

DELTA (REPRESENTED BY "D") RANGES FROM 0 TO   1.
LAMBDA RANGES FROM        114 TO        113.
THE MINIMUM COST PROJECT SCHEDULE FOR PROJECT DEADLINE =         114-D:


PROJECT COMPLETION TIME =         114-D.

  ACTIVITY #: I   NEW VALUE: XACT(I)          ACTIVITY COST
        1                    20                  0.50000E 02
        2                    14                  0.45000E 02
        3                    10                  0.50000E 02
        4                    70                  0.16000E 03
        5                    40-D                0.40000E 02 + (  0.62500E 00*D)
        6                    32                  0.48000E 02
        7                    13                  0.30000E 02
        8                    34                  0.48400E 02
        9                     5                  0.60000E 02
       10                    22-D                0.55500E 02 + (  0.16250E 01*D)
       11                    10                  0.40000E 02
       12                    10                  0.50000E 02
       13                    44                  0.17500E 03
       14                    40                  0.30000E 03
       15                    80                  0.21000E 03
       16                     4                  0.52000E 02
       17                     4                  0.90000E 02
       18                     6                  0.80000E 02

THE CURRENT VALUE OF THE PROJECT COST IS      0.15839E 04 + (  0.22500E 01*D).

DELTA (REPRESENTED BY "D") RANGES FROM 0 TO   5.
LAMBDA RANGES FROM        113 TO        108.
THE MINIMUM COST PROJECT SCHEDULE FOR PROJECT DEADLINE =         113-D:


PROJECT COMPLETION TIME =         113-D.

  ACTIVITY #: I   NEW VALUE: XACT(I)          ACTIVITY COST
        1                    20-D                0.50000E 02 + (  0.40000E 01*D)
        2                    14                  0.45000E 02
        3                    10                  0.50000E 02
        4                    70                  0.16000E 03
        5                    39                  0.40625E 02
        6                    32                  0.48000E 02
        7                    13                  0.30000E 02
```

```
         8                    34                  0.48400E 02
         9                     5                  0.60000E 02
        10                    21                  0.57125E 02
        11                    10                  0.40000E 02
        12                    10                  0.50000E 02
        13                    44                  0.17500E 03
        14                    40                  0.30000E 03
        15                    80                  0.21000E 03
        16                     4                  0.52000E 02
        17                     4                  0.90000E 02
        18                     6                  0.80000E 02

THE CURRENT VALUE OF THE PROJECT COST IS      0.15861E 04 + (  0.40000E 01*D).

DELTA (REPRESENTED BY "D") RANGES FROM 0 TO   2.
LAMBDA RANGES FROM       108 TO       106.
THE MINIMUM COST PROJECT SCHEDULE FOR PROJECT DEADLINE =        108-D:


PROJECT COMPLETION TIME =        108-D.

  ACTIVITY #: I   NEW VALUE: XACT(I)          ACTIVITY COST
         1                    15                  0.70000E 02
         2                    14                  0.45000E 02
         3                    10                  0.50000E 02
         4                    70                  0.16000E 03
         5                    39-D                0.40625E 02 + (  0.62500E 00*D)
         6                    32                  0.48000E 02
         7                    13                  0.30000E 02
         8                    34                  0.48400E 02
         9                     5                  0.60000E 02
        10                    21-D                0.57125E 02 + (  0.16250E 01*D)
        11                    10                  0.40000E 02
        12                    10                  0.50000E 02
        13                    44                  0.17500E 03
        14                    40                  0.30000E 03
        15                    80-D                0.21000E 03 + (  0.20000E 01*D)
        16                     4                  0.52000E 02
        17                     4                  0.90000E 02
        18                     6                  0.80000E 02

THE CURRENT VALUE OF THE PROJECT COST IS      0.16061E 04 + (  0.42500E 01*D).

DELTA (REPRESENTED BY "D") RANGES FROM 0 TO   1.
LAMBDA RANGES FROM       106 TO       105.
THE MINIMUM COST PROJECT SCHEDULE FOR PROJECT DEADLINE =        106-D:


PROJECT COMPLETION TIME =        106-D.

  ACTIVITY #: I   NEW VALUE: XACT(I)          ACTIVITY COST
         1                    15                  0.70000E 02
         2                    14                  0.45000E 02
         3                    10                  0.50000E 02
         4                    70                  0.16000E 03
         5                    37-D                0.41875E 02 + (  0.62500E 00*D)
         6                    32-D                0.48000E 02 + (  0.13333E 01*D)
         7                    13                  0.30000E 02
         8                    34                  0.48400E 02
```

```
         9                    5                    0.60000E 02
        10                   19-D                  0.60375E 02 + (  0.16250E 01*D)
        11                   10                    0.40000E 02
        12                   10                    0.50000E 02
        13                   44                    0.17500E 03
        14                   40                    0.30000E 03
        15                   78-D                  0.21400E 03 + (  0.20000E 01*D)
        16                    4                    0.52000E 02
        17                    4                    0.90000E 02
        18                    6                    0.80000E 02

THE CURRENT VALUE OF THE PROJECT COST IS      0.16146E 04 + (  0.55833E 01*D).


DELTA (REPRESENTED BY "D") RANGES FROM 0 TO   1.
LAMBDA RANGES FROM       105 TO       104.
THE MINIMUM COST PROJECT SCHEDULE FOR PROJECT DEADLINE =        105-D:


PROJECT COMPLETION TIME =        105-D.

  ACTIVITY #: I   NEW VALUE: XACT(I)          ACTIVITY COST
        1                    15-D                 0.70000E 02 + (  0.60000E 01*D)
        2                    14                   0.45000E 02
        3                    10                   0.50000E 02
        4                    70                   0.16000E 03
        5                    36                   0.42500E 02
        6                    31                   0.49333E 02
        7                    13                   0.30000E 02
        8                    34                   0.48400E 02
        9                     5                   0.60000E 02
       10                    18                   0.62000E 02
       11                    10                   0.40000E 02
       12                    10                   0.50000E 02
       13                    44                   0.17500E 03
       14                    40                   0.30000E 03
       15                    77                   0.21600E 03
       16                     4                   0.52000E 02
       17                     4                   0.90000E 02
       18                     6                   0.80000E 02

THE CURRENT VALUE OF THE PROJECT COST IS      0.16202E 04 + (  0.60000E 01*D).


DELTA (REPRESENTED BY "D") RANGES FROM 0 TO   4.
LAMBDA RANGES FROM       104 TO       100.
THE MINIMUM COST PROJECT SCHEDULE FOR PROJECT DEADLINE =        104-D:


PROJECT COMPLETION TIME =        104-D.

  ACTIVITY #: I   NEW VALUE: XACT(I)          ACTIVITY COST
        1                    14-D                 0.76000E 02 + (  0.60000E 01*D)
        2                    14-D                 0.45000E 02 + (  0.25000E 01*D)
        3                    10                   0.50000E 02
        4                    70                   0.16000E 03
        5                    36                   0.42500E 02
        6                    31                   0.49333E 02
        7                    13                   0.30000E 02
        8                    34                   0.48400E 02
        9                     5                   0.60000E 02
```

10	18	0.62000E 02
11	10	0.40000E 02
12	10	0.50000E 02
13	44	0.17500E 03
14	40	0.30000E 03
15	77	0.21600E 03
16	4	0.52000E 02
17	4	0.90000E 02
18	6	0.80000E 02

THE CURRENT VALUE OF THE PROJECT COST IS 0.16262E 04 + (0.85000E 01*D).

DELTA (REPRESENTED BY "D") RANGES FROM 0 TO 4.
LAMBDA RANGES FROM 100 TO 96.
THE MINIMUM COST PROJECT SCHEDULE FOR PROJECT DEADLINE = 100-D:

PROJECT COMPLETION TIME = 100-D.

ACTIVITY #: I	NEW VALUE: XACT(I)	ACTIVITY COST
1	10	0.10000E 03
2	10	0.55000E 02
3	10	0.50000E 02
4	70-D	0.16000E 03 + (0.20000E 01*D)
5	36	0.42500E 02
6	31	0.49333E 02
7	13	0.30000E 02
8	34-D	0.48400E 02 + (0.40000E 00*D)
9	5	0.60000E 02
10	18	0.62000E 02
11	10	0.40000E 02
12	10	0.50000E 02
13	44-D	0.17500E 03 + (0.62500E 01*D)
14	40	0.30000E 03
15	77-D	0.21600E 03 + (0.20000E 01*D)
16	4	0.52000E 02
17	4	0.90000E 02
18	6	0.80000E 02

THE CURRENT VALUE OF THE PROJECT COST IS 0.16602E 04 + (0.10650E 02*D).

DELTA (REPRESENTED BY "D") RANGES FROM 0 TO 4.
LAMBDA RANGES FROM 96 TO 92.
THE MINIMUM COST PROJECT SCHEDULE FOR PROJECT DEADLINE = 96-D:

PROJECT COMPLETION TIME = 96-D.

ACTIVITY #: I	NEW VALUE: XACT(I)	ACTIVITY COST
1	10	0.10000E 03
2	10	0.55000E 02
3	10	0.50000E 02
4	66-D	0.16800E 03 + (0.20000E 01*D)
5	36	0.42500E 02
6	31	0.49333E 02
7	13	0.30000E 02
8	30-D	0.50000E 02 + (0.10000E 01*D)
9	5	0.60000E 02
10	18	0.62000E 02

```
        11                      10                      0.40000E 02
        12                      10                      0.50000E 02
        13                      40-D                    0.20000E 03 + (  0.62500E 01*D)
        14                      40                      0.30000E 03
        15                      73-D                    0.22400E 03 + (  0.20000E 01*D)
        16                       4                      0.52000E 02
        17                       4                      0.90000E 02
        18                       6                      0.80000E 02

THE CURRENT VALUE OF THE PROJECT COST IS        0.17028E 04 + (  0.11250E 02*D).


DELTA (REPRESENTED BY "D") RANGES FROM 0 TO    2.
LAMBDA RANGES FROM          92 TO          90.
THE MINIMUM COST PROJECT SCHEDULE FOR PROJECT DEADLINE =          92-D:


PROJECT COMPLETION TIME =          92-D.

  ACTIVITY #: I    NEW VALUE: XACT(I)          ACTIVITY COST
        1                       10                      0.10000E 03
        2                       10                      0.55000E 02
        3                       10                      0.50000E 02
        4                       62                      0.17600E 03
        5                       36                      0.42500E 02
        6                       31                      0.49333E 02
        7                       13                      0.30000E 02
        8                       26                      0.54000E 02
        9                        5                      0.60000E 02
       10                       18                      0.62000E 02
       11                       10                      0.40000E 02
       12                       10                      0.50000E 02
       13                       36                      0.22500E 03
       14                       40                      0.30000E 03
       15                       69-D                    0.23200E 03 + (  0.20000E 01*D)
       16                        4                      0.52000E 02
       17                        4                      0.90000E 02
       18                        6-D                    0.80000E 02 + (  0.15000E 02*D)

THE CURRENT VALUE OF THE PROJECT COST IS        0.17478E 04 + (  0.17000E 02*D).


                        * * * * *


THE SINK HAS REACHED WITH INFINITE CAPACITY IMPLYING AN INFEASIBLE SOLUTION TO THE PRIMAL PROBLEM
                IF LAMBDA DROPS BELOW ITS CURRENT VALUE.          90.
```

****
****

THIS IS THE OUTPUT FROM THE DETERMINISTIC SCHEDULE RESOLUTION PROGRAM: DSR

****
****

THE MINIMUM COST PROJECT SCHEDULE FOR A TARGET TIME OF 125.00000 IS AS FOLLOWS:

ACTIVITY:	MEAN	DISTRIBUTION	PARAMETERS							
1	30.00000	BETA	MIN=	12.00000,	MAX=	48.00000,	ALPHA=	3.00000,	BETA=	3.00000
2	14.00000	NORMAL	MEAN=	14.00000,	ST.DEV=	3.81818				
3	10.00000	FIXED(CONSTANT)	TIME=	10.00000						
4	70.00000	NORMAL	MEAN=	70.00000,	ST.DEV=	11.66666				
5	40.00000	NORMAL	MEAN=	40.00000,	ST.DEV=	4.71404				
6	32.00000	BETA	MIN=	12.30770,	MAX=	61.53848,	ALPHA=	0.79428,	BETA=	1.69143
7	13.00000	NORMAL	MEAN=	13.00000,	ST.DEV=	2.14967				
8	34.00000	NORMAL	MEAN=	34.00000,	ST.DEV=	3.13846				
9	5.00000	FIXED(CONSTANT)	TIME=	5.00000						
10	23.00000	BETA	MIN=	10.45455,	MAX=	41.81821,	ALPHA=	12.99998,	BETA=	20.00002
11	10.00000	FIXED(CONSTANT)	TIME=	10.00000						
12	10.00000	NORMAL	MEAN=	10.00000,	ST.DEV=	2.79508				
13	44.00000	NORMAL	MEAN=	44.00000,	ST.DEV=	9.83869				
14	40.00000	NORMAL	MEAN=	40.00000,	ST.DEV=	5.71428				
15	80.00000	NORMAL	MEAN=	80.00000,	ST.DEV=	13.71428				
16	4.00000	FIXED(CONSTANT)	TIME=	4.00000						
17	4.00000	BETA	MIN=	1.33333,	MAX=	8.00000,	ALPHA=	1.00000,	BETA=	2.00000
18	6.00000	NORMAL	MEAN=	6.00000,	ST.DEV=	1.69705				

```
**********************************************************************************************************************************
**********************************************************************************************************************************

THIS IS THE OUTPUT FROM THE SIMPLIFICATION PROGRAM: SIMP

**********************************************************************************************************************************
**********************************************************************************************************************************
```

THE FOLLOWING ACTIVITIES CANNOT BE FURTHER COMBINED:

ACTIVITY	ORIGIN	TERMINAL	ACTIVITY NUMBER ASSIGNED BY DECOMP
2	1	2	1
4	2	6	2
5	2	5	3
6	2	4	4
7	2	3	5
8	5	6	6
9	4	5	7
10	3	4	8
11	6	8	9
12	6	7	10
13	5	7	11
14	4	7	12
15	3	9	13
16	7	8	14
17	7	9	15
18	8	9	16

****
****

THIS IS THE OUTPUT FROM THE MODIFIED SIMPLIFICATION PROGRAM: MODSIMP

****
****

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
2	1	2	30.8	6.40

F(T)	T
0.0 --	15.6000
0.05 --	20.0877
0.15 --	23.3456
0.25 --	25.5891
0.35 --	27.4363
0.45 --	29.1490
0.55 --	33.6000
0.65 --	33.6000
0.75 --	37.2000
0.85 --	37.2000
0.95 --	40.8000
1.00 --	44.4000

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
4	2	6	70.0	10.94

F(T)	T
0.0 --	35.0000
0.05 --	50.8092
0.15 --	57.9081
0.25 --	62.1310
0.35 --	65.5046
0.45 --	68.5340
0.55 --	71.4660
0.65 --	74.4954
0.75 --	77.8690
0.85 --	82.0919
0.95 --	89.1908
1.00 --	105.0000

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
5	2	5	40.0	4.42

F(T)		T
0.0	--	25.8579
0.05	--	32.2457
0.15	--	35.1141
0.25	--	36.8204
0.35	--	38.1836
0.45	--	39.4076
0.55	--	40.5924
0.65	--	41.8164
0.75	--	43.1796
0.85	--	44.8859
0.95	--	47.7543
1.00	--	54.1421

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
6	2	4	31.9	10.03

F(T)		T
0.0	--	12.3077
0.05	--	16.5614
0.15	--	20.5854
0.25	--	23.8162
0.35	--	26.8016
0.45	--	29.7300
0.55	--	32.7243
0.65	--	35.9089
0.75	--	39.4651
0.85	--	43.7664
0.95	--	50.1376
1.00	--	61.5385

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
7	2	3	13.0	2.02

F(T)		T
0.0	--	6.5510
0.05	--	9.4639
0.15	--	10.7720
0.25	--	11.5501
0.35	--	12.1717
0.45	--	12.7299
0.55	--	13.2701
0.65	--	13.8283
0.75	--	14.4499
0.85	--	15.2280
0.95	--	16.5361
1.00	--	19.4490

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
8	5	6	34.0	2.94

F(T)	T
0.0 --	24.5846
0.05 --	28.8375
0.15 --	30.7471
0.25 --	31.8831
0.35 --	32.7907
0.45 --	33.6056
0.55 --	34.3944
0.65 --	35.2093
0.75 --	36.1169
0.85 --	37.2529
0.95 --	39.1625
1.00 --	43.4154

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
9	4	5	5.0	0.00

F(T)	T
0.0 --	5.0000
0.05 --	5.0000
0.15 --	5.0000
0.25 --	5.0000
0.35 --	5.0000
0.45 --	5.0000
0.55 --	5.0000
0.65 --	5.0000
0.75 --	5.0000
0.85 --	5.0000
0.95 --	5.0000
1.00 --	5.0000

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
10	3	4	23.0	2.42

F(T)	T
0.0 --	10.4546
0.05 --	18.8855
0.15 --	20.3169
0.25 --	21.2104
0.35 --	21.9427
0.45 --	22.6122
0.55 --	23.2690
0.65 --	23.9550
0.75 --	24.7250
0.85 --	25.6942
0.95 --	27.3216
1.00 --	41.8182

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
11	6	8	10.0	0.00

F(T)		T
0.0	--	10.0000
0.05	--	10.0000
0.15	--	10.0000
0.25	--	10.0000
0.35	--	10.0000
0.45	--	10.0000
0.55	--	10.0000
0.65	--	10.0000
0.75	--	10.0000
0.85	--	10.0000
0.95	--	10.0000
1.00	--	10.0000

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
12	6	7	10.0	2.62

F(T)		T
0.0	--	1.6147
0.05	--	5.4023
0.15	--	7.1030
0.25	--	8.1147
0.35	--	8.9230
0.45	--	9.6488
0.55	--	10.3512
0.65	--	11.0770
0.75	--	11.8853
0.85	--	12.8970
0.95	--	14.5977
1.00	--	18.3853

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
13	5	7	44.0	9.23

F(T)		T
0.0	--	14.4839
0.05	--	27.8160
0.15	--	33.8027
0.25	--	37.3639
0.35	--	40.2090
0.45	--	42.7637
0.55	--	45.2363
0.65	--	47.7910
0.75	--	50.6361
0.85	--	54.1973
0.95	--	60.1840
1.00	--	73.5161

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
14	4	7	40.0	5.36

F(T)		T
0.0	--	22.8572
0.05	--	30.6004
0.15	--	34.0774
0.25	--	36.1458
0.35	--	37.7982
0.45	--	39.2819
0.55	--	40.7181
0.65	--	42.2018
0.75	--	43.8542
0.85	--	45.9226
0.95	--	49.3996
1.00	--	57.1428

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
15	3	9	80.0	12.86

F(T)		T
0.0	--	38.8572
0.05	--	57.4410
0.15	--	65.7858
0.25	--	70.7499
0.35	--	74.7156
0.45	--	78.2767
0.55	--	81.7233
0.65	--	85.2844
0.75	--	89.2501
0.85	--	94.2142
0.95	--	102.5590
1.00	--	121.1428

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
16	7	8	4.0	0.00

F(T)		T
0.0	--	4.0000
0.05	--	4.0000
0.15	--	4.0000
0.25	--	4.0000
0.35	--	4.0000
0.45	--	4.0000
0.55	--	4.0000
0.65	--	4.0000
0.75	--	4.0000
0.85	--	4.0000
0.95	--	4.0000
1.00	--	4.0000

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
17	7	9	4.0	1.30

F(T)		T
0.0	--	1.3333
0.05	--	1.9840
0.15	--	2.5292
0.25	--	2.9535
0.35	--	3.3398
0.45	--	3.7155
0.55	--	4.0979
0.65	--	4.5038
0.75	--	4.9579
0.85	--	5.5103
0.95	--	6.3427
1.00	--	8.0000

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
18	8	9	6.0	1.59

F(T)		T
0.0	--	0.9088
0.05	--	3.2085
0.15	--	4.2411
0.25	--	4.8554
0.35	--	5.3461
0.45	--	5.7667
0.55	--	6.2133
0.65	--	6.6539
0.75	--	7.1446
0.85	--	7.7589
0.95	--	8.7915
1.00	--	11.0912

```
**********************************************************************************************************************************
**********************************************************************************************************************************

THIS IS THE OUTPUT FROM THE DECOMPOSITION PROGRAM: DECOMP

**********************************************************************************************************************************
**********************************************************************************************************************************

     INPUT STAGE

     THE SIMPLIFIED NETWORK HAS                16 ARCS

     THE SOURCE IS NODE NUMBER                  1

     THE SINK IS NODE NUMBER                    9

     THE LARGEST NODE IS NODE NUMBER            9

     THE SIMPLIFIED NETWORK AS READ IN IS:

     ARC NUMBER      ORIGIN NODE      TERMINAL NODE
          1                1                2
          2                2                6
          3                2                5
          4                2                4
          5                2                3
          6                5                6
          7                4                5
          8                3                4
          9                6                8
         10                6                7
         11                5                7
         12                4                7
         13                3                9
         14                7                8
         15                7                9
         16                8                9
```

```
STAGE  1 BREAKUP

SUBNETWORK   1 IS COMPOSED OF SUBNETWORKS:

  2.  3.

IN SERIES
```

STAGE 2 BREAKUP

SUBNETWORK 2 IS A NON-DECOMPOSABLE NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 1

SINK NODE = 2

THE SUBNETWORK AS IT IS READ IN:

ARC	S(ARC)	T(ARC)
1	1	2

THE FOLLOWING REPRESENTATION OF THE SUBNETWORK ABOVE WILL BE USED BY THE SUBNETWORK ANALYSIS PROGRAM:

ACTIVITY	TAIL	HEAD
1	1	2

SUBNETWORK 3 IS A NON-DECOMPOSABLE NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 2

SINK NODE = 9

THE SUBNETWORK AS IT IS READ IN:

ARC	S(ARC)	T(ARC)
2	2	6
3	2	5
4	2	4
5	2	3
9	6	8
10	6	7
6	5	6
11	5	7
7	4	5
12	4	7
8	3	4
13	3	9
16	8	9
14	7	8
15	7	9

THE FOLLOWING REPRESENTATION OF THE SUBNETWORK ABOVE WILL BE USED BY THE SUBNETWORK ANALYSIS PROGRAM:

ACTIVITY	TAIL	HEAD
1	1	5
2	1	4
3	1	3
4	1	2
5	5	7
6	5	6
7	4	5
8	4	6
9	3	4
10	3	6
11	2	3
12	2	8
13	7	8
14	6	7
15	6	8

THE NUMBER OF NCA-DECOMPOSABLE SUBNETWORKS IS 2

****
****

THIS IS THE OUTPUT FROM THE SUBNETWORK ANALYSIS PROGRAM: SUBNET

****
****

EACH SUBNETWORK'S APPROXIMATE DURATION DISTRIBUTION IS DETERMINED.

SUBNETWORK 2

INITIAL INPUT

ACTIVITY	ORIGIN	TERMINAL	LOWER POINT	UPPER POINT	MEAN	STANDARD DEVIATION	PROB. LOWER PT.
1	1	2	24.0785	36.8868	30.8008	6.3962	0.47516

THE DURATION DISTRIBUTION FOR SUBNETWORK 2

TIME	DISTRIBUTION
0.156000 02	0.0
0.200880 02	0.500000-01
0.233460 02	0.150000 00
0.255840 02	0.250000 00
0.274360 02	0.350000 00
0.291480 02	0.450000 00
0.336000 02	0.550000 00
0.336000 02	0.650000 00
0.372000 02	0.750000 00
0.372000 02	0.850000 00
0.408000 02	0.950000 00
0.444000 02	0.100000 01

SUBNETWORK 3

INITIAL INPUT

ACTIVITY	ORIGIN	TERMINAL	LOWER POINT	UPPER POINT	MEAN	STANDARD DEVIATION	PROB. LOWER PT.
1	1	5	59.0567	80.9433	70.0000	10.9433	0.50000
2	1	4	35.5782	44.4218	40.0000	4.4218	0.50000
3	1	3	23.0185	43.2217	31.9697	10.0336	0.55793
4	1	2	10.9836	15.0164	13.0000	2.0164	0.50000
5	5	7	10.0000	10.0000	10.0000	0.0	1.00000
6	5	6	7.3782	12.6218	10.0000	2.6218	0.50000
7	4	5	31.0561	36.9439	34.0000	2.9439	0.50000
8	4	6	34.7713	53.2287	44.0000	9.2287	0.50000
9	3	4	5.0000	5.0000	5.0000	0.0	1.00000
10	3	6	34.6400	45.3600	40.0000	5.3600	0.50000
11	2	3	20.6713	25.5124	22.9933	2.4185	0.52037
12	2	8	67.1360	92.8640	80.0000	12.8640	0.50000
13	7	8	4.4082	7.5918	6.0000	1.5918	0.50000
14	6	7	4.0000	4.0000	4.0000	0.0	1.00000
15	6	8	2.8330	5.4402	3.9934	1.2957	0.55491

THE ORDER IN WHICH TO CONSIDER THE ACTIVITIES TO DETERMINE THE LONGEST PATH TIME:

ORDER	ACTIVITY
1	1
2	2
3	3
4	4
5	11
6	12
7	9
8	10
9	7
10	8
11	5
12	6
13	14
14	15
15	13

THE CRITICAL PATH TIME WHEN EACH ACTIVITY'S COMPLETION TIME IS SET EQUAL TO ITS MEAN IS = 0.949930 02

THE 8 NODES ON THE CRITICAL PATH ARE AS FOLLOWS BEGINNING WITH THE TERMINAL NODE:
8. 7. 6. 5. 4. 3. 2. 1.

THE 7 CRITICAL ACTIVITIES ARE AS FOLLOWS BEGINNING WITH THE TERMINAL ACTIVITY:
13. 14. 6. 7. 9. 11. 4.

A LOWER BOUND ON THE SUBNETWORK DURATION TIME IS = 0.834970 02

THE 8 NODES ON THE LONGEST PATH ARE AS FOLLOWS BEGINNING WITH THE TERMINAL NODE:
8. 7. 6. 5. 4. 3. 2. 1.

THE 7 ACTIVITIES ON THE LONGEST PATH ARE AS FOLLOWS BEGINNING WITH THE TERMINAL ACTIVITY:
13. 14. 6. 7. 9. 11. 4.

A UPPER BOUND ON THE SUBNETWORK DURATION TIME IS = 0.113040 03

THE 6 NODES ON THE LONGEST PATH ARE AS FOLLOWS BEGINNING WITH THE TERMINAL NODE:
8. 7. 6. 4. 3. 1.

THE 5 ACTIVITIES ON THE LONGEST PATH ARE AS FOLLOWS BEGINNING WITH THE TERMINAL ACTIVITY:
13. 14. 8. 9. 3.

```
THE ASSOCIATES ARE NOW IDENTIFIED FOR LAMBDA =      0.100000D 01

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   1-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY  13. IS =   0

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   2-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY  14. IS =   0

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   3-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY   6. IS =   1

          THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS
           8.

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   4-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY   7. IS =   1

          THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS
           8.

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   5-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY   9. IS =   0

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   6-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY  11. IS =   1

          THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS
           2.

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   7-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY   4. IS =   1

          THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS
           2.

THERE ARE   5 NONEMPTY CLUSTERS AFTER POOLING ON THE BASIS OF ASSOCIATES ONLY.

     THE ACTIVITIES IN THE   1-TH CLUSTER ARE AS FOLLOWS:
          13.

     THE ACTIVITIES IN THE   2-TH CLUSTER ARE AS FOLLOWS:
          14.

     THE ACTIVITIES IN THE   3-TH CLUSTER ARE AS FOLLOWS:
           6.  8.  7.

     THE ACTIVITIES IN THE   5-TH CLUSTER ARE AS FOLLOWS:
           9.

     THE ACTIVITIES IN THE   6-TH CLUSTER ARE AS FOLLOWS:
          11.  2.  4.

     THERE ARE   6 ACTIVITIES NOT IN ANY CLUSTER YET.
```

```
THE ELIMINANTS OF EACH NON-CRITICAL-PATH ACTIVITY ARE NOW DETERMINED FOR THETA =      0.10000D 01

     THERE ARE    8 ACTIVITIES NOT ON THE CRITICAL PATH.  THEY ARE AS FOLLOWS:
             1.      1
             2.      2
             3.      3
             4.      5
             5.      8
             6.     10
             7.     12
             8.     15

     THERE ARE    4 ELIMINANTS CORRESPONDING TO ACTIVITY    1
         THE    1-TH ELIMINANT CORRESPONDING TO ACTIVITY    1 IS ACTIVITY    7
         THE    2-TH ELIMINANT CORRESPONDING TO ACTIVITY    1 IS ACTIVITY    9
         THE    3-TH ELIMINANT CORRESPONDING TO ACTIVITY    1 IS ACTIVITY   11
         THE    4-TH ELIMINANT CORRESPONDING TO ACTIVITY    1 IS ACTIVITY    4

     THERE ARE    3 ELIMINANTS CORRESPONDING TO ACTIVITY    2
         THE    1-TH ELIMINANT CORRESPONDING TO ACTIVITY    2 IS ACTIVITY    9
         THE    2-TH ELIMINANT CORRESPONDING TO ACTIVITY    2 IS ACTIVITY   11
         THE    3-TH ELIMINANT CORRESPONDING TO ACTIVITY    2 IS ACTIVITY    4

     THERE ARE    2 ELIMINANTS CORRESPONDING TO ACTIVITY    3
         THE    1-TH ELIMINANT CORRESPONDING TO ACTIVITY    3 IS ACTIVITY   11
         THE    2-TH ELIMINANT CORRESPONDING TO ACTIVITY    3 IS ACTIVITY    4

     THERE ARE    0 ELIMINANTS CORRESPONDING TO ACTIVITY    5

     THERE ARE    2 ELIMINANTS CORRESPONDING TO ACTIVITY    8
         THE    1-TH ELIMINANT CORRESPONDING TO ACTIVITY    8 IS ACTIVITY    6
         THE    2-TH ELIMINANT CORRESPONDING TO ACTIVITY    8 IS ACTIVITY    7

     THERE ARE    0 ELIMINANTS CORRESPONDING TO ACTIVITY   10

     THERE ARE    6 ELIMINANTS CORRESPONDING TO ACTIVITY   12
         THE    1-TH ELIMINANT CORRESPONDING TO ACTIVITY   12 IS ACTIVITY   13
         THE    2-TH ELIMINANT CORRESPONDING TO ACTIVITY   12 IS ACTIVITY   14
         THE    3-TH ELIMINANT CORRESPONDING TO ACTIVITY   12 IS ACTIVITY    6
         THE    4-TH ELIMINANT CORRESPONDING TO ACTIVITY   12 IS ACTIVITY    7
         THE    5-TH ELIMINANT CORRESPONDING TO ACTIVITY   12 IS ACTIVITY    9
         THE    6-TH ELIMINANT CORRESPONDING TO ACTIVITY   12 IS ACTIVITY   11

     THERE ARE    0 ELIMINANTS CORRESPONDING TO ACTIVITY   15
```

THERE ARE 1 NONEMPTY CLUSTERS AFTER POOLING ON THE BASIS OF BOTH ASSOCIATES AND ELIMINANTS.

THERE ARE 12 ACTIVITIES IN THE 1-TH CLUSTER. THEY ARE AS FOLLOWS:
13. 12. 14. 6. 8. 7. 1. 9. 11. 2. 4. 3.

7 CLUSTERS HAVE BEEN POOLED TO MAKE THIS CLUSTER. THEY WERE AS FOLLOWS:
1. 2. 3. 4. 5. 6. 7.

BOUNDS ON THE DURATION DISTRIBUTION FOR SUBNETWORK 3 WHEN (THETA,LAMBDA) = (0.10000D 01 , 0.10000D 01)

TIME	LOWER BOUND	UPPER BOUND
0.85646D 02	0.23221D-02	0.23221D-02
0.87795D 02	0.23221D-02	0.69664D-02
0.89944D 02	0.11611D-01	0.13751D-01
0.92092D 02	0.18213D-01	0.18213D-01
0.94241D 02	0.20536D-01	0.20536D-01
0.96390D 02	0.22858D-01	0.30374D-01
0.98539D 02	0.75167D-01	0.80398D-01
0.10069D 03	0.11936D 00	0.11936D 00
0.10284D 03	0.18818D 00	0.18818D 00
0.10498D 03	0.35409D 00	0.35409D 00
0.10713D 03	0.50588D 00	0.50588D 00
0.10779D 03	0.52300D 00	0.52300D 00
0.10845D 03	0.67962D 00	0.67962D 00
0.10910D 03	0.67962D 00	0.67962D 00
0.10976D 03	0.75374D 00	0.75374D 00
0.11042D 03	0.88884D 00	0.88884D 00
0.11107D 03	0.88884D 00	0.88884D 00
0.11173D 03	0.88884D 00	0.88884D 00
0.11239D 03	0.88884D 00	0.88884D 00
0.11304D 03	0.10000D 01	0.10000D 01

THE BOUNDS WERE DETERMINED USING THE UNION-CLUSTER PROCEDURE.

ONLY 500 OF THE POSSIBLE ACTIVITY DURATION CONFIGURATIONS WERE CONSIDERED.

SYSTEMATIC SAMPLING WAS USED.

```
THE ASSOCIATES ARE NOW IDENTIFIED FOR LAMBDA =      0.20000D 01

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   1-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY  13, IS =   1

          THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS
          12.

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   2-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY  14, IS =   0

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   3-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY   6, IS =   1

          THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS
           8.

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   4-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY   7, IS =   1

          THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS
           8.

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   5-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY   9, IS =   0

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   6-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY  11, IS =   1

          THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS
           2.

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   7-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY   4, IS =   1

          THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS
           2.

THERE ARE   5 NONEMPTY CLUSTERS AFTER POOLING ON THE BASIS OF ASSOCIATES ONLY.

     THE ACTIVITIES IN THE   1-TH CLUSTER ARE AS FOLLOWS:
          13. 12.

     THE ACTIVITIES IN THE   2-TH CLUSTER ARE AS FOLLOWS:
          14.

     THE ACTIVITIES IN THE   3-TH CLUSTER ARE AS FOLLOWS:
           6.  8.  7.

     THE ACTIVITIES IN THE   5-TH CLUSTER ARE AS FOLLOWS:
           9.

     THE ACTIVITIES IN THE   6-TH CLUSTER ARE AS FOLLOWS:
          11.  2.  4.

     THERE ARE   5 ACTIVITIES NOT IN ANY CLUSTER YET.
```

```
THE ELIMINANTS OF EACH NON-CRITICAL-PATH ACTIVITY ARE NOW DETERMINED FOR THETA =      0.200000D 01

     THERE ARE    8 ACTIVITIES NOT ON THE CRITICAL PATH.  THEY ARE AS FOLLOWS:
            1.      1
            2.      2
            3.      3
            4.      5
            5.      8
            6.     10
            7.     12
            8.     15

     THERE ARE    4 ELIMINANTS CORRESPONDING TO ACTIVITY    1
         THE    1-TH ELIMINANT CORRESPONDING TO ACTIVITY    1 IS ACTIVITY    7
         THE    2-TH ELIMINANT CORRESPONDING TO ACTIVITY    1 IS ACTIVITY    9
         THE    3-TH ELIMINANT CORRESPONDING TO ACTIVITY    1 IS ACTIVITY   11
         THE    4-TH ELIMINANT CORRESPONDING TO ACTIVITY    1 IS ACTIVITY    4

     THERE ARE    3 ELIMINANTS CORRESPONDING TO ACTIVITY    2
         THE    1-TH ELIMINANT CORRESPONDING TO ACTIVITY    2 IS ACTIVITY    9
         THE    2-TH ELIMINANT CORRESPONDING TO ACTIVITY    2 IS ACTIVITY   11
         THE    3-TH ELIMINANT CORRESPONDING TO ACTIVITY    2 IS ACTIVITY    4

     THERE ARE    2 ELIMINANTS CORRESPONDING TO ACTIVITY    3
         THE    1-TH ELIMINANT CORRESPONDING TO ACTIVITY    3 IS ACTIVITY   11
         THE    2-TH ELIMINANT CORRESPONDING TO ACTIVITY    3 IS ACTIVITY    4

     THERE ARE    0 ELIMINANTS CORRESPONDING TO ACTIVITY    5

     THERE ARE    2 ELIMINANTS CORRESPONDING TO ACTIVITY    8
         THE    1-TH ELIMINANT CORRESPONDING TO ACTIVITY    8 IS ACTIVITY    6
         THE    2-TH ELIMINANT CORRESPONDING TO ACTIVITY    8 IS ACTIVITY    7

     THERE ARE    3 ELIMINANTS CORRESPONDING TO ACTIVITY   10
         THE    1-TH ELIMINANT CORRESPONDING TO ACTIVITY   10 IS ACTIVITY    6
         THE    2-TH ELIMINANT CORRESPONDING TO ACTIVITY   10 IS ACTIVITY    7
         THE    3-TH ELIMINANT CORRESPONDING TO ACTIVITY   10 IS ACTIVITY    9

     THERE ARE    6 ELIMINANTS CORRESPONDING TO ACTIVITY   12
         THE    1-TH ELIMINANT CORRESPONDING TO ACTIVITY   12 IS ACTIVITY   13
         THE    2-TH ELIMINANT CORRESPONDING TO ACTIVITY   12 IS ACTIVITY   14
         THE    3-TH ELIMINANT CORRESPONDING TO ACTIVITY   12 IS ACTIVITY    6
         THE    4-TH ELIMINANT CORRESPONDING TO ACTIVITY   12 IS ACTIVITY    7
         THE    5-TH ELIMINANT CORRESPONDING TO ACTIVITY   12 IS ACTIVITY    9
         THE    6-TH ELIMINANT CORRESPONDING TO ACTIVITY   12 IS ACTIVITY   11

     THERE ARE    0 ELIMINANTS CORRESPONDING TO ACTIVITY   15
```

THERE ARE 1 NONEMPTY CLUSTERS AFTER POOLING ON THE BASIS OF BOTH ASSOCIATES AND ELIMINANTS.

THERE ARE 13 ACTIVITIES IN THE 1-TH CLUSTER. THEY ARE AS FOLLOWS:
13. 12. 14. 6. 8. 7. 1. 9. 11. 2. 4. 3. 10.

7 CLUSTERS HAVE BEEN POOLED TO MAKE THIS CLUSTER. THEY WERE AS FOLLOWS:
1. 2. 3. 4. 5. 6. 7.

BOUNDS ON THE DURATION DISTRIBUTION FOR SUBNETWORK 3 WHEN (THETA,LAMBDA) = (0.20000D 01 , 0.20000D 01)

TIME	LOWER BOUND	UPPER BOUND
0.85646D 02	0.23213D-02	0.23213D-02
0.87755D 02	0.69638D-02	0.69638D-02
0.89944D 02	0.92850D-02	0.92850D-02
0.92052D 02	0.11606D-01	0.11606D-01
0.94241D 02	0.24989D-01	0.24989D-01
0.96350D 02	0.34861D-01	0.34861D-01
0.98538D 02	0.81552D-01	0.81552D-01
0.10069D 03	0.13154D 00	0.13154D 00
0.10284D 03	0.18928D 00	0.18928D 00
0.10498D 03	0.34177D 00	0.34177D 00
0.10713D 03	0.49998D 00	0.49998D 00
0.10779D 03	0.52137D 00	0.52137D 00
0.10845D 03	0.64707D 00	0.64707D 00
0.10910D 03	0.64707D 00	0.64707D 00
0.10976D 03	0.73837D 00	0.73837D 00
0.11042D 03	0.88491D 00	0.88491D 00
0.11107D 03	0.88491D 00	0.88491D 00
0.11173D 03	0.88491D 00	0.88491D 00
0.11239D 03	0.88491D 00	0.88491D 00
0.11304D 03	0.10000D 01	0.10000D 01

THE BOUNDS WERE DETERMINED USING THE UNION-CLUSTER PROCEDURE.

ONLY 500 OF THE POSSIBLE ACTIVITY DURATION CONFIGURATIONS WERE CONSIDERED.

SYSTEMATIC SAMPLING WAS USED.

```
THE ASSOCIATES ARE NOW IDENTIFIED FOR LAMBDA =      0.20000D 01

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   1-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY  13, IS =   1

          THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS
          12.

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   2-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY  14, IS =   0

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   3-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY   6, IS =   1

          THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS
          8.

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   4-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY   7, IS =   1

          THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS
          8.

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   5-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY   9, IS =   0

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   6-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY  11, IS =   1

          THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS
          2.

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   7-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY   4, IS =   1

          THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS
          2.

THERE ARE   5 NONEMPTY CLUSTERS AFTER POOLING ON THE BASIS OF ASSOCIATES ONLY.

     THE ACTIVITIES IN THE   1-TH CLUSTER ARE AS FOLLOWS:
          13. 12.

     THE ACTIVITIES IN THE   2-TH CLUSTER ARE AS FOLLOWS:
          14.

     THE ACTIVITIES IN THE   3-TH CLUSTER ARE AS FOLLOWS:
          6.  8.  7.

     THE ACTIVITIES IN THE   5-TH CLUSTER ARE AS FOLLOWS:
          9.

     THE ACTIVITIES IN THE   6-TH CLUSTER ARE AS FOLLOWS:
          11.  2.  4.

     THERE ARE   5 ACTIVITIES NOT IN ANY CLUSTER YET.
```

```
THE ELIMINANTS OF EACH NON-CRITICAL-PATH ACTIVITY ARE NOW DETERMINED FOR THETA =      0.300000 01

     THERE ARE    8 ACTIVITIES NOT ON THE CRITICAL PATH.  THEY ARE AS FOLLOWS:
            1.      1
            2.      2
            3.      3
            4.      5
            5.      8
            6.     10
            7.     12
            8.     15

     THERE ARE    4 ELIMINANTS CORRESPONDING TO ACTIVITY    1
         THE    1-TH ELIMINANT CORRESPONDING TO ACTIVITY    1 IS ACTIVITY    7
         THE    2-TH ELIMINANT CORRESPONDING TO ACTIVITY    1 IS ACTIVITY    9
         THE    3-TH ELIMINANT CORRESPONDING TO ACTIVITY    1 IS ACTIVITY   11
         THE    4-TH ELIMINANT CORRESPONDING TO ACTIVITY    1 IS ACTIVITY    4

     THERE ARE    3 ELIMINANTS CORRESPONDING TO ACTIVITY    2
         THE    1-TH ELIMINANT CORRESPONDING TO ACTIVITY    2 IS ACTIVITY    9
         THE    2-TH ELIMINANT CORRESPONDING TO ACTIVITY    2 IS ACTIVITY   11
         THE    3-TH ELIMINANT CORRESPONDING TO ACTIVITY    2 IS ACTIVITY    4

     THERE ARE    2 ELIMINANTS CORRESPONDING TO ACTIVITY    3
         THE    1-TH ELIMINANT CORRESPONDING TO ACTIVITY    3 IS ACTIVITY   11
         THE    2-TH ELIMINANT CORRESPONDING TO ACTIVITY    3 IS ACTIVITY    4

     THERE ARE    0 ELIMINANTS CORRESPONDING TO ACTIVITY    5

     THERE ARE    2 ELIMINANTS CORRESPONDING TO ACTIVITY    8
         THE    1-TH ELIMINANT CORRESPONDING TO ACTIVITY    8 IS ACTIVITY    6
         THE    2-TH ELIMINANT CORRESPONDING TO ACTIVITY    8 IS ACTIVITY    7

     THERE ARE    3 ELIMINANTS CORRESPONDING TO ACTIVITY   10
         THE    1-TH ELIMINANT CORRESPONDING TO ACTIVITY   10 IS ACTIVITY    6
         THE    2-TH ELIMINANT CORRESPONDING TO ACTIVITY   10 IS ACTIVITY    7
         THE    3-TH ELIMINANT CORRESPONDING TO ACTIVITY   10 IS ACTIVITY    9

     THERE ARE    6 ELIMINANTS CORRESPONDING TO ACTIVITY   12
         THE    1-TH ELIMINANT CORRESPONDING TO ACTIVITY   12 IS ACTIVITY   13
         THE    2-TH ELIMINANT CORRESPONDING TO ACTIVITY   12 IS ACTIVITY   14
         THE    3-TH ELIMINANT CORRESPONDING TO ACTIVITY   12 IS ACTIVITY    6
         THE    4-TH ELIMINANT CORRESPONDING TO ACTIVITY   12 IS ACTIVITY    7
         THE    5-TH ELIMINANT CORRESPONDING TO ACTIVITY   12 IS ACTIVITY    9
         THE    6-TH ELIMINANT CORRESPONDING TO ACTIVITY   12 IS ACTIVITY   11

     THERE ARE    0 ELIMINANTS CORRESPONDING TO ACTIVITY   15
```

```
THERE ARE    1 NONEMPTY CLUSTERS AFTER POOLING ON THE BASIS OF BOTH ASSOCIATES AND ELIMINANTS.

     THERE ARE  13 ACTIVITIES IN THE    1-TH CLUSTER.  THEY ARE AS FOLLOWS:
          13.  12.  14.   6.   8.   7.   1.   9.  11.   2.   4.   3.  10.

          7 CLUSTERS HAVE BEEN POOLED TO MAKE THIS CLUSTER. THEY WERE AS FOLLOWS:
               1.   2.   3.   4.   5.   6.   7.
```

BOUNDS ON THE DURATION DISTRIBUTION FOR SUBNETWORK 3 WHEN (THETA,LAMBDA) = (0.300000D 01 , 0.200000D 01)

TIME	LOWER BOUND	UPPER BOUND
0.856460 02	0.232130-02	0.232130-02
0.877950 02	0.696380-02	0.696380-02
0.899440 02	0.928500-02	0.928500-02
0.920920 02	0.116060-01	0.116060-01
0.942410 02	0.249890-01	0.249890-01
0.963900 02	0.348610-01	0.348610-01
0.985380 02	0.815520-01	0.815520-01
0.100690 03	0.131540 00	0.131540 00
0.102840 03	0.189280 00	0.189280 00
0.104980 03	0.341770 00	0.341770 00
0.107130 03	0.459980 00	0.459980 00
0.107790 03	0.521370 00	0.521370 00
0.108450 03	0.647070 00	0.647070 00
0.109100 03	0.647070 00	0.647070 00
0.109760 03	0.738370 00	0.738370 00
0.110420 03	0.884910 00	0.884910 00
0.111070 03	0.884910 00	0.884910 00
0.111730 03	0.884910 00	0.884910 00
0.112390 03	0.884910 00	0.884910 00
0.113040 03	0.100000 01	0.100000 01

THE BOUNDS WERE DETERMINED USING THE UNION-CLUSTER PROCEDURE.

ONLY 500 OF THE POSSIBLE ACTIVITY DURATION CONFIGURATIONS WERE CONSIDERED.

SYSTEMATIC SAMPLING WAS USED.

THE UPPER AND LOWER BOUNDS ON THE DISCRETE SUBNETWORK DURATION DISTRIBUTION ARE EQUAL.

THUS AN ESTIMATE OF THE DISCRETE DURATION DISTRIBUTION FOR SUBNETWORK 3 IS

TIME	DISTRIBUTION
0.856460 02	0.23213D-02
0.877950 02	0.69638D-02
0.899440 02	0.92850D-02
0.920920 02	0.11606D-01
0.942410 02	0.24989D-01
0.963900 02	0.34861D-01
0.985380 02	0.81552D-01
0.100690 03	0.13154D 00
0.102840 03	0.18928D 00
0.104980 03	0.34177D 00
0.107130 03	0.49958D 00
0.107790 03	0.52137D 00
0.108450 03	0.64707D 00
0.109100 03	0.64707D 00
0.109760 03	0.73837D 00
0.110420 03	0.88491D 00
0.111070 03	0.88491D 00
0.111730 03	0.88491D 00
0.112390 03	0.88491D 00
0.113040 03	0.10000D 01

****
****

THIS IS THE OUTPUT FROM THE SYNTHESIS PROGRAM: SYNTH

****
****

THE CURRENT PROJECT SCHEDULE IS THE ONE MOST RECENTLY LISTED BY THE DETERMINISTIC SCHEDULE RESOLUTION PROGRAM.

THE PROJECT'S CORRESPONDING APPROXIMATE COMPLETION TIME DISTRIBUTION IS DETERMINED BELOW:

FOR THIS PROBLEM

THE NUMBER OF C.D.F. SUBDIVISIONS USED THROUGHOUT SYNTHESIS IS 49.
THE TARGET TIME USED BY THE DETERMINISTIC SCHEDULER WAS 125.00000
THE MEAN OF THE PROJECT'S APPROXIMATE COMPLETION TIME DISTRIBUTION WILL BE COMPARED TO THE
SPECIFIED PROJECT DEADLINE TIME OF 125.00000

THE INSTRUCTIONS TO BE PERFORMED ARE

INSTRUCTION NO.	SUBNETWORK	SERIES=0 PARALLEL=1	SUBNETWORKS TO BE SYNTHESIZED
1	1	0	2 3

SUBNETWORK APPROXIMATE DISTRIBUTION DETERMINED USING SUBNETWORK ANALYSIS

2

X	F(X)
15.6000	0.0
20.0876	0.05000
23.3455	0.15000
25.5891	0.25000
27.4362	0.35000
29.1490	0.45000
33.6000	0.55000
33.6000	0.65000
37.2000	0.75000
37.2000	0.85000
40.8000	0.95000
44.4000	1.00000

3

X	F(X)
85.6461	0.00232
87.7948	0.00696
89.9435	0.00929
92.0922	0.01161
94.2409	0.02499
96.3896	0.03486
98.5384	0.08155
100.6871	0.13154
102.8358	0.18928
104.9845	0.34177
107.1332	0.49998
107.7898	0.52137
108.4463	0.64707
109.1029	0.64707
109.7594	0.73837
110.4160	0.88491
111.0725	0.88491
111.7291	0.88491
112.3856	0.88491
113.0422	1.00000

THE DISTRIBUTION FOR THE SYNTHESIZED NETWORK IS

T	F(T)
[illegible]	0.0
[illegible]	0.0
[illegible]	0.0
[illegible]	0.0
[illegible]	0.00012
[illegible]	0.00012
[illegible]	0.00035
[illegible]	0.00058
[illegible]	0.00070
[illegible]	0.00139
[illegible]	0.00151
[illegible]	0.00244
[illegible]	0.00334
[illegible]	0.00427
[illegible]	0.00522
[illegible]	0.00703
[illegible]	0.01029
[illegible]	0.01262
[illegible]	0.02228
[illegible]	0.02662
124.18331	0.03508
125.33017	0.04763
126.47703	0.06476
127.62389	0.07313
128.77075	0.11432
129.91761	0.12435
131.06447	0.17084
132.21133	0.20357
133.35819	0.24976
134.50505	0.30646
135.65191	0.33489
136.79876	0.41044
137.94562	0.45992
139.09248	0.51173
140.23934	0.54260
141.38620	0.58624
142.53306	0.66266
143.67992	0.68903
144.82678	0.74598
145.97364	0.79715
147.12050	0.83843
148.26736	0.88644
149.41422	0.90878
150.56108	0.94093
151.70794	0.96349
152.85480	0.97084
154.00166	0.98235
155.14852	0.99425
156.29538	0.99425
157.44224	1.00000

THE MEAN OF THE SYNTHESIZED DISTRIBUTION IS 139.41685

THE DIFFERENCE BETWEEN 139.41685 AND THE PROJECT DEADLINE IS 11.53 PERCENT OF THE PROJECT DEADLINE.
HENCE, THE NEW TARGET TIME IS 112.07397

```
**********************************************************************************************************************
THIS COMPLETES ITERATION      1
**********************************************************************************************************************
```

THIS IS THE OUTPUT FROM THE DETERMINISTIC SCHEDULE RESOLUTION PROGRAM: DSR

THE MINIMUM COST PROJECT SCHEDULE FOR A TARGET TIME OF 112.07396 IS AS FOLLOWS:

ACTIVITY:	MEAN	DISTRIBUTION	PARAMETERS						
1	19.07396	NORMAL	MEAN=	19.07396,	ST.DEV=	5.44970			
2	14.00000	NORMAL	MEAN=	14.00000,	ST.DEV=	3.81818			
3	10.00000	FIXED(CONSTANT)	TIME=	10.00000					
4	70.00000	NORMAL	MEAN=	70.00000,	ST.DEV=	11.66666			
5	39.00000	NORMAL	MEAN=	39.00000,	ST.DEV=	4.59619			
6	32.00000	BETA	MIN=	12.30770,	MAX=	61.53848,	ALPHA=	0.79428,	BETA= 1.69143
7	13.00000	NORMAL	MEAN=	13.00000,	ST.DEV=	2.14967			
8	34.00000	NORMAL	MEAN=	34.00000,	ST.DEV=	3.13846			
9	5.00000	FIXED(CONSTANT)	TIME=	5.00000					
10	21.00000	BETA	MIN=	9.54546,	MAX=	38.18184,	ALPHA=	12.99998,	BETA= 20.00002
11	10.00000	FIXED(CONSTANT)	TIME=	10.00000					
12	10.00000	NORMAL	MEAN=	10.00000,	ST.DEV=	2.79508			
13	44.00000	NORMAL	MEAN=	44.00000,	ST.DEV=	9.83869			
14	40.00000	NORMAL	MEAN=	40.00000,	ST.DEV=	5.71428			
15	80.00000	NORMAL	MEAN=	80.00000,	ST.DEV=	13.71428			
16	4.00000	FIXED(CONSTANT)	TIME=	4.00000					
17	4.00000	BETA	MIN=	1.33333,	MAX=	8.00000,	ALPHA=	1.00000,	BETA= 2.00000
18	6.00000	NORMAL	MEAN=	6.00000,	ST.DEV=	1.69705			

****
****

THIS IS THE OUTPUT FROM THE MODIFIED SIMPLIFICATION PROGRAM: MODSIMP

****
****

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
2	1	2	20.8	5.13

F(T)		T
0.0	--	10.0000
0.05	--	12.7844
0.15	--	15.2036
0.25	--	16.6132
0.35	--	17.8740
0.45	--	18.9984
0.55	--	22.7115
0.65	--	22.7115
0.75	--	25.2538
0.85	--	25.2538
0.95	--	30.3384
1.00	--	32.8808

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
4	2	6	70.0	10.94

F(T)		T
0.0	--	35.0000
0.05	--	50.8092
0.15	--	57.9081
0.25	--	62.1310
0.35	--	65.5046
0.45	--	68.5340
0.55	--	71.4660
0.65	--	74.4954
0.75	--	77.8690
0.85	--	82.0919
0.95	--	89.1908
1.00	--	105.0000

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
5	2	5	39.0	4.31

F(T)		T
0.0	--	25.2114
0.05	--	31.4396
0.15	--	34.2363
0.25	--	35.8999
0.35	--	37.2290
0.45	--	38.4224
0.55	--	39.5776
0.65	--	40.7710
0.75	--	42.1001
0.85	--	43.7637
0.95	--	46.5604
1.00	--	52.7886

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
6	2	4	31.9	10.03

F(T)		T
0.0	--	12.3077
0.05	--	16.5614
0.15	--	20.5854
0.25	--	23.8162
0.35	--	26.8016
0.45	--	29.7300
0.55	--	32.7243
0.65	--	35.9089
0.75	--	39.4651
0.85	--	43.7664
0.95	--	50.1376
1.00	--	61.5385

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
7	2	3	13.0	2.02

F(T)		T
0.0	--	6.5510
0.05	--	9.4639
0.15	--	10.7720
0.25	--	11.5501
0.35	--	12.1717
0.45	--	12.7299
0.55	--	13.2701
0.65	--	13.8283
0.75	--	14.4499
0.85	--	15.2280
0.95	--	16.5361
1.00	--	19.4490

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
8	5	6	34.0	2.94

F(T)		T
0.0	--	24.5846
0.05	--	28.8375
0.15	--	30.7471
0.25	--	31.8831
0.35	--	32.7907
0.45	--	33.6056
0.55	--	34.3944
0.65	--	35.2093
0.75	--	36.1169
0.85	--	37.2529
0.95	--	39.1625
1.00	--	43.4154

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
9	4	5	5.0	0.00

F(T)		T
0.0	--	5.0000
0.05	--	5.0000
0.15	--	5.0000
0.25	--	5.0000
0.35	--	5.0000
0.45	--	5.0000
0.55	--	5.0000
0.65	--	5.0000
0.75	--	5.0000
0.85	--	5.0000
0.95	--	5.0000
1.00	--	5.0000

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
10	3	4	21.0	2.21

F(T)		T
0.0	--	9.5455
0.05	--	17.2433
0.15	--	18.5502
0.25	--	19.3660
0.35	--	20.0346
0.45	--	20.6459
0.55	--	21.2456
0.65	--	21.8719
0.75	--	22.5750
0.85	--	23.4600
0.95	--	24.9458
1.00	--	38.1818

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
11	6	8	10.0	0.00

F(T)		T
0.0	--	10.0000
0.05	--	10.0000
0.15	--	10.0000
0.25	--	10.0000
0.35	--	10.0000
0.45	--	10.0000
0.55	--	10.0000
0.65	--	10.0000
0.75	--	10.0000
0.85	--	10.0000
0.95	--	10.0000
1.00	--	10.0000

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
12	6	7	10.0	2.62

F(T)		T
0.0	--	1.6147
0.05	--	5.4023
0.15	--	7.1030
0.25	--	8.1147
0.35	--	8.9230
0.45	--	9.6488
0.55	--	10.3512
0.65	--	11.0770
0.75	--	11.8853
0.85	--	12.8970
0.95	--	14.5977
1.00	--	18.3853

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
13	5	7	44.0	9.23

F(T)		T
0.0	--	14.4839
0.05	--	27.8160
0.15	--	33.8027
0.25	--	37.3639
0.35	--	40.2090
0.45	--	42.7637
0.55	--	45.2363
0.65	--	47.7910
0.75	--	50.6361
0.85	--	54.1973
0.95	--	60.1840
1.00	--	73.5161

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
14	4	7	40.0	5.36

F(T)		T
0.0	--	22.8572
0.05	--	30.6004
0.15	--	34.0774
0.25	--	36.1458
0.35	--	37.7982
0.45	--	39.2819
0.55	--	40.7181
0.65	--	42.2018
0.75	--	43.8542
0.85	--	45.9226
0.95	--	49.3996
1.00	--	57.1428

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
15	3	9	80.0	12.86

F(T)		T
0.0	--	38.8572
0.05	--	57.4410
0.15	--	65.7858
0.25	--	70.7499
0.35	--	74.7156
0.45	--	78.2767
0.55	--	81.7233
0.65	--	85.2844
0.75	--	89.2501
0.85	--	94.2142
0.95	--	102.5590
1.00	--	121.1428

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
16	7	8	4.0	0.00

F(T)		T
0.0	--	4.0000
0.05	--	4.0000
0.15	--	4.0000
0.25	--	4.0000
0.35	--	4.0000
0.45	--	4.0000
0.55	--	4.0000
0.65	--	4.0000
0.75	--	4.0000
0.85	--	4.0000
0.95	--	4.0000
1.00	--	4.0000

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
17	7	9	4.0	1.30

F(T)		T
0.0	--	1.3333
0.05	--	1.9840
0.15	--	2.5292
0.25	--	2.9535
0.35	--	3.3398
0.45	--	3.7155
0.55	--	4.0979
0.65	--	4.5038
0.75	--	4.9579
0.85	--	5.5103
0.95	--	6.3427
1.00	--	8.0000

ACTIVITY	ORIGIN	TERMINAL	MEAN	STANDARD DEVIATION
18	8	9	6.0	1.59

F(T)		T
0.0	--	0.9088
0.05	--	3.2085
0.15	--	4.2411
0.25	--	4.8554
0.35	--	5.3461
0.45	--	5.7867
0.55	--	6.2133
0.65	--	6.6539
0.75	--	7.1446
0.85	--	7.7589
0.95	--	8.7915
1.00	--	11.0912

```
**********************************************************************************************************************************
**********************************************************************************************************************************

THIS IS THE OUTPUT FROM THE SUBNETWORK ANALYSIS PROGRAM: SUBNET

**********************************************************************************************************************************
**********************************************************************************************************************************


EACH SUBNETWORK'S APPROXIMATE DURATION DISTRIBUTION IS DETERMINED.
```

SUBNETWORK 2

INITIAL INPUT

ACTIVITY	ORIGIN	TERMINAL	LOWER POINT	UPPER POINT	MEAN	STANDARD DEVIATION	PROB. LOWER PT.
1	1	2	16.1595	26.4703	20.7743	5.1270	0.55243

THE CUMULATIVE DISTRIBUTION FOR SUBNETWORK 2

TIME	DISTRIBUTION
0.100000D 02	0.0
0.127840 02	0.500000-01
0.152040 02	0.150000 00
0.166130 02	0.250000 00
0.178740 02	0.350000 00
0.189580 02	0.450000 00
0.227120 02	0.550000 00
0.227120 02	0.650000 00
0.252540 02	0.750000 00
0.252540 02	0.850000 00
0.303380 02	0.950000 00
0.328810 02	0.100000 01

SUBNETWORK 3

INITIAL INPUT

ACTIVITY	ORIGIN	TERMINAL	LOWER POINT	UPPER POINT	MEAN	STANDARD DEVIATION	PROB. LOWER PT.
1	1	5	59.0567	80.9433	70.0000	10.9433	0.50000
2	1	4	34.6888	43.3112	39.0000	4.3112	0.50000
3	1	3	23.0185	43.2217	31.9497	10.0336	0.55793
4	1	2	10.9836	15.0164	13.0000	2.0164	0.50000
5	5	7	10.0000	10.0000	10.0000	0.0	1.00000
6	5	6	7.3782	12.6218	10.0000	2.6218	0.50000
7	4	5	31.0561	36.9439	34.0000	2.9439	0.50000
8	4	6	34.7713	53.2287	44.0000	9.2287	0.50000
9	3	4	5.0000	5.0000	5.0000	0.0	1.00000
10	3	6	34.6400	45.3600	40.0000	5.3600	0.50000
11	2	3	18.8738	23.2939	20.9938	2.2082	0.52037
12	2	8	67.1360	92.8640	80.0000	12.8640	0.50000
13	7	8	4.4082	7.5918	6.0000	1.5918	0.50000
14	6	7	4.0000	4.0000	4.0000	0.0	1.00000
15	6	8	2.6330	5.4402	3.9934	1.2957	0.55491

THE ORDER IN WHICH TO CONSIDER THE ACTIVITIES TO DETERMINE THE LONGEST PATH TIME:

ORDER	ACTIVITY
1	1
2	2
3	3
4	4
5	11
6	12
7	9
8	10
9	7
10	8
11	5
12	6
13	14
14	15
15	13

THE CRITICAL PATH TIME WHEN EACH ACTIVITY'S COMPLETION TIME IS SET EQUAL TO ITS MEAN IS = 0.930000 02

THE 3 NODES ON THE CRITICAL PATH ARE AS FOLLOWS BEGINNING WITH THE TERMINAL NODE:
8. 2. 1.

THE 2 CRITICAL ACTIVITIES ARE AS FOLLOWS BEGINNING WITH THE TERMINAL ACTIVITY:
12. 4.

A LOWER BOUND ON THE SUBNETWORK DURATION TIME IS = 0.817000 02

THE 8 NODES ON THE LONGEST PATH ARE AS FOLLOWS BEGINNING WITH THE TERMINAL NODE:
8. 7. 6. 5. 4. 3. 2. 1.

THE 7 ACTIVITIES ON THE LONGEST PATH ARE AS FOLLOWS BEGINNING WITH THE TERMINAL ACTIVITY:
13. 14. 6. 7. 9. 11. 4.

A UPPER BOUND ON THE SUBNETWORK DURATION TIME IS = 0.113040 03

THE 6 NODES ON THE LONGEST PATH ARE AS FOLLOWS BEGINNING WITH THE TERMINAL NODE:
8. 7. 6. 4. 3. 1.

THE 5 ACTIVITIES ON THE LONGEST PATH ARE AS FOLLOWS BEGINNING WITH THE TERMINAL ACTIVITY:
13. 14. 8. 9. 3.

AD-A060 568 TEXAS A AND M UNIV COLLEGE STATION INST OF STATISTICS F/G 9/2
A USER'S GUIDE TO THE COMPUTER IMPLEMENTATION OF THE NEW PROJEC--ETC(U)
AUG 78 T C BAKER, R L SIELKEN N00014-78-C-0426
UNCLASSIFIED THEMIS-TR-57 NL

2 OF 4
AD
A060568

```
THE ASSOCIATES ARE NOW IDENTIFIED FOR LAMBDA =      0.100000D 01

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   1-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY  12. IS =   5

          THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS
          13.  14.  6.  7.  2.

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   2-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY   4. IS =   5

          THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS
          13.  14.  6.  7.  2.


THERE ARE   1 NONEMPTY CLUSTERS AFTER POOLING ON THE BASIS OF ASSOCIATES ONLY.

     THE ACTIVITIES IN THE   1-TH CLUSTER ARE AS FOLLOWS:
          12.  13.  14.  6.  7.  2.  4.

     THERE ARE   8 ACTIVITIES NOT IN ANY CLUSTER YET.
```

```
THE ELIMINANTS OF EACH NON-CRITICAL-PATH ACTIVITY ARE NOW DETERMINED FOR THETA =      0.100000D 01

     THERE ARE  13 ACTIVITIES NOT ON THE CRITICAL PATH.  THEY ARE AS FOLLOWS:
           1.      1
           2.      2
           3.      3
           4.      5
           5.      6
           6.      7
           7.      8
           8.      9
           9.     10
          10.     11
          11.     13
          12.     14
          13.     15

     THERE ARE   2 ELIMINANTS CORRESPONDING TO ACTIVITY   1
         THE   1-TH ELIMINANT CORRESPONDING TO ACTIVITY   1 IS ACTIVITY  12
         THE   2-TH ELIMINANT CORRESPONDING TO ACTIVITY   1 IS ACTIVITY   4

     THERE ARE   2 ELIMINANTS CORRESPONDING TO ACTIVITY   2
         THE   1-TH ELIMINANT CORRESPONDING TO ACTIVITY   2 IS ACTIVITY  12
         THE   2-TH ELIMINANT CORRESPONDING TO ACTIVITY   2 IS ACTIVITY   4

     THERE ARE   2 ELIMINANTS CORRESPONDING TO ACTIVITY   3
         THE   1-TH ELIMINANT CORRESPONDING TO ACTIVITY   3 IS ACTIVITY  12
         THE   2-TH ELIMINANT CORRESPONDING TO ACTIVITY   3 IS ACTIVITY   4

     THERE ARE   0 ELIMINANTS CORRESPONDING TO ACTIVITY   5

     THERE ARE   2 ELIMINANTS CORRESPONDING TO ACTIVITY   6
         THE   1-TH ELIMINANT CORRESPONDING TO ACTIVITY   6 IS ACTIVITY  12
         THE   2-TH ELIMINANT CORRESPONDING TO ACTIVITY   6 IS ACTIVITY   4

     THERE ARE   2 ELIMINANTS CORRESPONDING TO ACTIVITY   7
         THE   1-TH ELIMINANT CORRESPONDING TO ACTIVITY   7 IS ACTIVITY  12
         THE   2-TH ELIMINANT CORRESPONDING TO ACTIVITY   7 IS ACTIVITY   4

     THERE ARE   2 ELIMINANTS CORRESPONDING TO ACTIVITY   8
         THE   1-TH ELIMINANT CORRESPONDING TO ACTIVITY   8 IS ACTIVITY  12
         THE   2-TH ELIMINANT CORRESPONDING TO ACTIVITY   8 IS ACTIVITY   4

     THERE ARE   0 ELIMINANTS CORRESPONDING TO ACTIVITY   9

     THERE ARE   0 ELIMINANTS CORRESPONDING TO ACTIVITY  10

     THERE ARE   1 ELIMINANTS CORRESPONDING TO ACTIVITY  11
         THE   1-TH ELIMINANT CORRESPONDING TO ACTIVITY  11 IS ACTIVITY  12

     THERE ARE   2 ELIMINANTS CORRESPONDING TO ACTIVITY  13
         THE   1-TH ELIMINANT CORRESPONDING TO ACTIVITY  13 IS ACTIVITY  12
         THE   2-TH ELIMINANT CORRESPONDING TO ACTIVITY  13 IS ACTIVITY   4

     THERE ARE   0 ELIMINANTS CORRESPONDING TO ACTIVITY  14

     THERE ARE   0 ELIMINANTS CORRESPONDING TO ACTIVITY  15
```

THERE ARE 1 NONEMPTY CLUSTERS AFTER POOLING ON THE BASIS OF BOTH ASSOCIATES AND ELIMINANTS.

THERE ARE 11 ACTIVITIES IN THE 1-TH CLUSTER. THEY ARE AS FOLLOWS:
12. 13. 14. 6. 7. 2. 4. 1. 3. 8. 11.

2 CLUSTERS HAVE BEEN POOLED TO MAKE THIS CLUSTER. THEY WERE AS FOLLOWS:
1. 2.

BOUNDS ON THE DURATION DISTRIBUTION FOR SUBNETWORK 3 WHEN (THETA,LAMBDA) = (0.10000D 01 , 0.10000D 01)

TIME	LOWER BOUND	UPPER BOUND
0.839790 02	0.23224D-02	0.23224D-02
0.862590 02	0.23224D-02	0.23224D-02
0.885380 02	0.46447D-02	0.46447D-02
0.908180 02	0.13934D-01	0.18215D-01
0.930970 02	0.22678D-01	0.22678D-01
0.953770 02	0.36067D-01	0.43139D-01
0.976560 02	0.90297D-01	0.90297D-01
0.999350 02	0.12941D 00	0.13479D 00
0.102210 03	0.20336D 00	0.20336D 00
0.104490 03	0.38347D 00	0.38347D 00
0.106770 03	0.50814D 00	0.50814D 00
0.107470 03	0.50814D 00	0.50814D 00
0.108170 03	0.75049D 00	0.75049D 00
0.108860 03	0.75049D 00	0.75049D 00
0.109560 03	0.77906D 00	0.77906D 00
0.110260 03	0.90395D 00	0.90395D 00
0.110950 03	0.90395D 00	0.90395D 00
0.111650 03	0.90395D 00	0.90395D 00
0.112350 03	0.90395D 00	0.90395D 00
0.113040 03	0.10000D 01	0.10000D 01

THE BOUNDS WERE DETERMINED USING THE UNION-CLUSTER PROCEDURE.

ONLY 500 OF THE POSSIBLE ACTIVITY DURATION CONFIGURATIONS WERE CONSIDERED.

SYSTEMATIC SAMPLING WAS USED.

```
THE ASSOCIATES ARE NOW IDENTIFIED FOR LAMBDA =      0.20000D 01

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   1-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY  12. IS =   5

          THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS
          13.  14.   6.   7.   2.

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   2-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY   4. IS =   5

          THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS
          13.  14.   6.   7.   2.


THERE ARE   1 NONEMPTY CLUSTERS AFTER POOLING ON THE BASIS OF ASSOCIATES ONLY.

     THE ACTIVITIES IN THE   1-TH CLUSTER ARE AS FOLLOWS:
          12.  13.  14.   6.   7.   2.   4.

     THERE ARE   8 ACTIVITIES NOT IN ANY CLUSTER YET.
```

```
THE ELIMINANTS OF EACH NON-CRITICAL-PATH ACTIVITY ARE NOW DETERMINED FOR THETA =      0.200000 01

     THERE ARE  13 ACTIVITIES NOT ON THE CRITICAL PATH.  THEY ARE AS FOLLOWS:
           1.      1
           2.      2
           3.      3
           4.      5
           5.      6
           6.      7
           7.      8
           8.      9
           9.     10
          10.     11
          11.     13
          12.     14
          13.     15

     THERE ARE   2 ELIMINANTS CORRESPONDING TO ACTIVITY   1
         THE   1-TH ELIMINANT CORRESPONDING TO ACTIVITY   1 IS ACTIVITY  12
         THE   2-TH ELIMINANT CORRESPONDING TO ACTIVITY   1 IS ACTIVITY   4

     THERE ARE   2 ELIMINANTS CORRESPONDING TO ACTIVITY   2
         THE   1-TH ELIMINANT CORRESPONDING TO ACTIVITY   2 IS ACTIVITY  12
         THE   2-TH ELIMINANT CORRESPONDING TO ACTIVITY   2 IS ACTIVITY   4

     THERE ARE   2 ELIMINANTS CORRESPONDING TO ACTIVITY   3
         THE   1-TH ELIMINANT CORRESPONDING TO ACTIVITY   3 IS ACTIVITY  12
         THE   2-TH ELIMINANT CORRESPONDING TO ACTIVITY   3 IS ACTIVITY   4

     THERE ARE   0 ELIMINANTS CORRESPONDING TO ACTIVITY   5

     THERE ARE   2 ELIMINANTS CORRESPONDING TO ACTIVITY   6
         THE   1-TH ELIMINANT CORRESPONDING TO ACTIVITY   6 IS ACTIVITY  12
         THE   2-TH ELIMINANT CORRESPONDING TO ACTIVITY   6 IS ACTIVITY   4

     THERE ARE   2 ELIMINANTS CORRESPONDING TO ACTIVITY   7
         THE   1-TH ELIMINANT CORRESPONDING TO ACTIVITY   7 IS ACTIVITY  12
         THE   2-TH ELIMINANT CORRESPONDING TO ACTIVITY   7 IS ACTIVITY   4

     THERE ARE   2 ELIMINANTS CORRESPONDING TO ACTIVITY   8
         THE   1-TH ELIMINANT CORRESPONDING TO ACTIVITY   8 IS ACTIVITY  12
         THE   2-TH ELIMINANT CORRESPONDING TO ACTIVITY   8 IS ACTIVITY   4

     THERE ARE   0 ELIMINANTS CORRESPONDING TO ACTIVITY   9

     THERE ARE   1 ELIMINANTS CORRESPONDING TO ACTIVITY  10
         THE   1-TH ELIMINANT CORRESPONDING TO ACTIVITY  10 IS ACTIVITY  12

     THERE ARE   1 ELIMINANTS CORRESPONDING TO ACTIVITY  11
         THE   1-TH ELIMINANT CORRESPONDING TO ACTIVITY  11 IS ACTIVITY  12

     THERE ARE   2 ELIMINANTS CORRESPONDING TO ACTIVITY  13
         THE   1-TH ELIMINANT CORRESPONDING TO ACTIVITY  13 IS ACTIVITY  12
         THE   2-TH ELIMINANT CORRESPONDING TO ACTIVITY  13 IS ACTIVITY   4

     THERE ARE   0 ELIMINANTS CORRESPONDING TO ACTIVITY  14

     THERE ARE   0 ELIMINANTS CORRESPONDING TO ACTIVITY  15
```

```
THERE ARE   1 NONEMPTY CLUSTERS AFTER POOLING ON THE BASIS OF BOTH ASSOCIATES AND ELIMINANTS.

     THERE ARE  12 ACTIVITIES IN THE   1-TH CLUSTER.  THEY ARE AS FOLLOWS:
          12.  13.  14.   6.   7.   2.   4.   1.   3.   8.  10.  11.

          2 CLUSTERS HAVE BEEN POOLED TO MAKE THIS CLUSTER. THEY WERE AS FOLLOWS:
               1.   2.
```

BOUNDS ON THE DURATION DISTRIBUTION FOR SUBNETWORK 3 WHEN (THETA,LAMBDA) = (0.20000D 01 , 0.20000D 01)

TIME	LOWER BOUND	UPPER BOUND
0.839790 02	0.23178D-02	0.23178D-02
0.862590 02	0.23178D-02	0.23178D-02
0.885380 02	0.46355D-02	0.46355D-02
0.908180 02	0.20316D-01	0.20316D-01
0.930970 02	0.22633D-01	0.22633D-01
0.953770 02	0.35214D-01	0.35214D-01
0.976560 02	0.91031D-01	0.91031D-01
0.999350 02	0.13791D 00	0.13791D 00
0.102210 03	0.21670D 00	0.21670D 00
0.104490 03	0.37580D 00	0.37580D 00
0.106770 03	0.51085D 00	0.51085D 00
0.109470 03	0.51085D 00	0.51085D 00
0.102170 03	0.76481D 00	0.76481D 00
0.110860 03	0.76481D 00	0.76481D 00
0.109560 03	0.78091D 00	0.78091D 00
0.111260 03	0.90824D 00	0.90824D 00
0.110950 03	0.90824D 00	0.90824D 00
0.111650 03	0.90824D 00	0.90824D 00
0.112350 03	0.90824D 00	0.90824D 00
0.113040 03	0.10000D 01	0.10000D 01

THE BOUNDS WERE DETERMINED USING THE UNION-CLUSTER PROCEDURE.

ONLY 500 OF THE POSSIBLE ACTIVITY DURATION CONFIGURATIONS WERE CONSIDERED.

SYSTEMATIC SAMPLING WAS USED.

```
THE ASSOCIATES ARE NOW IDENTIFIED FOR LAMBDA =      0.200000 01

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   1-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY  12, IS =   5

          THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS
          13,  14,   6,   7,   2,

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   2-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY   4, IS =   5

          THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS
          13,  14,   6,   7,   2,


THERE ARE   1 NONEMPTY CLUSTERS AFTER POOLING ON THE BASIS OF ASSOCIATES ONLY.

     THE ACTIVITIES IN THE   1-TH CLUSTER ARE AS FOLLOWS:
          12,  13,  14,   6,   7,   2,   4,

     THERE ARE   8 ACTIVITIES NOT IN ANY CLUSTER YET.
```

```
THE ELIMINANTS OF EACH NON-CRITICAL-PATH ACTIVITY ARE NOW DETERMINED FOR THETA =      0.300000 01

     THERE ARE  13 ACTIVITIES NOT ON THE CRITICAL PATH.  THEY ARE AS FOLLOWS:
           1.      1
           2.      2
           3.      3
           4.      5
           5.      6
           6.      7
           7.      8
           8.      9
           9.     10
          10.     11
          11.     13
          12.     14
          13.     15

     THERE ARE    2 ELIMINANTS CORRESPONDING TO ACTIVITY    1
         THE   1-TH ELIMINANT CORRESPONDING TO ACTIVITY    1 IS ACTIVITY   12
         THE   2-TH ELIMINANT CORRESPONDING TO ACTIVITY    1 IS ACTIVITY    4

     THERE ARE    2 ELIMINANTS CORRESPONDING TO ACTIVITY    2
         THE   1-TH ELIMINANT CORRESPONDING TO ACTIVITY    2 IS ACTIVITY   12
         THE   2-TH ELIMINANT CORRESPONDING TO ACTIVITY    2 IS ACTIVITY    4

     THERE ARE    2 ELIMINANTS CORRESPONDING TO ACTIVITY    3
         THE   1-TH ELIMINANT CORRESPONDING TO ACTIVITY    3 IS ACTIVITY   12
         THE   2-TH ELIMINANT CORRESPONDING TO ACTIVITY    3 IS ACTIVITY    4

     THERE ARE    0 ELIMINANTS CORRESPONDING TO ACTIVITY    5

     THERE ARE    2 ELIMINANTS CORRESPONDING TO ACTIVITY    6
         THE   1-TH ELIMINANT CORRESPONDING TO ACTIVITY    6 IS ACTIVITY   12
         THE   2-TH ELIMINANT CORRESPONDING TO ACTIVITY    6 IS ACTIVITY    4

     THERE ARE    2 ELIMINANTS CORRESPONDING TO ACTIVITY    7
         THE   1-TH ELIMINANT CORRESPONDING TO ACTIVITY    7 IS ACTIVITY   12
         THE   2-TH ELIMINANT CORRESPONDING TO ACTIVITY    7 IS ACTIVITY    4

     THERE ARE    2 ELIMINANTS CORRESPONDING TO ACTIVITY    8
         THE   1-TH ELIMINANT CORRESPONDING TO ACTIVITY    8 IS ACTIVITY   12
         THE   2-TH ELIMINANT CORRESPONDING TO ACTIVITY    8 IS ACTIVITY    4

     THERE ARE    0 ELIMINANTS CORRESPONDING TO ACTIVITY    9

     THERE ARE    1 ELIMINANTS CORRESPONDING TO ACTIVITY   10
         THE   1-TH ELIMINANT CORRESPONDING TO ACTIVITY   10 IS ACTIVITY   12

     THERE ARE    1 ELIMINANTS CORRESPONDING TO ACTIVITY   11
         THE   1-TH ELIMINANT CORRESPONDING TO ACTIVITY   11 IS ACTIVITY   12

     THERE ARE    2 ELIMINANTS CORRESPONDING TO ACTIVITY   13
         THE   1-TH ELIMINANT CORRESPONDING TO ACTIVITY   13 IS ACTIVITY   12
         THE   2-TH ELIMINANT CORRESPONDING TO ACTIVITY   13 IS ACTIVITY    4

     THERE ARE    0 ELIMINANTS CORRESPONDING TO ACTIVITY   14

     THERE ARE    0 ELIMINANTS CORRESPONDING TO ACTIVITY   15
```

THERE ARE 1 NONEMPTY CLUSTERS AFTER POOLING ON THE BASIS OF BOTH ASSOCIATES AND ELIMINANTS.

THERE ARE 12 ACTIVITIES IN THE 1-TH CLUSTER. THEY ARE AS FOLLOWS:
12. 13. 14. 6. 7. 2. 4. 1. 3. 8. 10. 11.

2 CLUSTERS HAVE BEEN POOLED TO MAKE THIS CLUSTER. THEY WERE AS FOLLOWS:
1. 2.

BOUNDS ON THE DURATION DISTRIBUTION FOR SUBNETWORK 3 WHEN (THETA,LAMBDA) = (0.300000 01 , 0.200000 01)

TIME	LOWER BOUND	UPPER BOUND
0.839790 02	0.23178D-02	0.23178D-02
0.862590 02	0.23178D-02	0.23178D-02
0.885380 02	0.46355D-02	0.46355D-02
0.908180 02	0.20316D-01	0.20316D-01
0.930970 02	0.22633D-01	0.22633D-01
0.953770 02	0.35214D-01	0.35214D-01
0.976560 02	0.91031D-01	0.91031D-01
0.999350 02	0.13791D 00	0.13791D 00
0.102210 03	0.21670D 00	0.21670D 00
0.104490 03	0.37580D 00	0.37580D 00
0.106770 03	0.51085D 00	0.51085D 00
0.107470 03	0.51085D 00	0.51085D 00
0.108170 03	0.76481D 00	0.76481D 00
0.108860 03	0.76481D 00	0.76481D 00
0.109560 03	0.78091D 00	0.78091D 00
0.110260 03	0.90824D 00	0.90824D 00
0.110950 03	0.90824D 00	0.90824D 00
0.111650 03	0.90824D 00	0.90824D 00
0.112350 03	0.90824D 00	0.90824D 00
0.113040 03	0.10000D 01	0.10000D 01

THE BOUNDS WERE DETERMINED USING THE UNION-CLUSTER PROCEDURE.

ONLY 500 OF THE POSSIBLE ACTIVITY DURATION CONFIGURATIONS WERE CONSIDERED.

SYSTEMATIC SAMPLING WAS USED.

THE UPPER AND LOWER BOUNDS ON THE DISCRETE SUBNETWORK DURATION DISTRIBUTION ARE EQUAL.

THUS AN ESTIMATE OF THE DISCRETE DURATION DISTRIBUTION FOR SUBNETWORK 3 IS

TIME	DISTRIBUTION
0.839790 C2	0.231780-02
0.862590 02	0.231780-02
0.885380 02	0.463550-02
0.908180 02	0.203160-01
0.930970 02	0.226330-01
0.953770 02	0.352140-01
0.976560 C2	0.910310-01
0.999350 02	0.137910 00
0.102210 03	0.216700 00
0.104490 03	0.375600 00
0.106770 03	0.510850 00
0.107470 03	0.510850 00
0.108170 03	0.764810 00
0.108860 03	0.764880 00
0.109560 03	0.780910 00
0.110260 03	0.900840 00
0.110950 03	0.900840 00
0.111650 03	0.900840 00
0.112350 03	0.900840 00
0.113040 03	0.100000 01

****
****

THIS IS THE OUTPUT FROM THE SYNTHESIS PROGRAM: SYNTH

****
****

THE CURRENT PROJECT SCHEDULE IS THE ONE MOST RECENTLY LISTED BY THE DETERMINISTIC SCHEDULE RESOLUTION PROGRAM.

THE PROJECT'S CORRESPONDING APPROXIMATE COMPLETION TIME DISTRIBUTION IS DETERMINED BELOW:

FOR THIS PROBLEM

THE NUMBER OF C.D.F. SUBDIVISIONS USED THROUGHOUT SYNTHESIS IS 49.
THE TARGET TIME USED BY THE DETERMINISTIC SCHEDULER WAS 112.07396
THE MEAN OF THE PROJECT'S APPROXIMATE COMPLETION TIME DISTRIBUTION WILL BE COMPARED TO THE
SPECIFIED PROJECT DEADLINE TIME OF 125.00000

THE INSTRUCTIONS TO BE PERFORMED ARE

INSTRUCTION NO.	SUBNETWORK	SERIES=0 PARALLEL=1	SUBNETWORKS TO BE SYNTHESIZED
1	1	0	2 3

SUBNETWORK	APPROXIMATE DISTRIBUTION DETERMINED USING SUBNETWORK ANALYSIS	
2	X	F(X)
	10.0000	0.0
	12.7843	0.05000
	15.2036	0.15000
	16.6132	0.25000
	17.8740	0.35000
	18.9984	0.45000
	22.7115	0.55000
	22.7115	0.65000
	25.2538	0.75000
	25.2538	0.85000
	30.3384	0.95000
	32.6807	1.00000
3	X	F(X)
	83.9793	0.00232
	86.2587	0.00232
	88.5382	0.00464
	90.8176	0.02032
	93.0971	0.02263
	95.3765	0.03521
	97.6560	0.09103
	99.9354	0.13791
	102.2148	0.21670
	104.4943	0.37580
	106.7737	0.51085
	107.4702	0.51085
	108.1667	0.76481
	108.8632	0.76481
	109.5597	0.78091
	110.2562	0.90824
	110.9527	0.90824
	111.6492	0.90824
	112.3457	0.90824
	113.0422	1.00000

THE DISTRIBUTION FOR THE SYNTHESIZED NETWORK IS

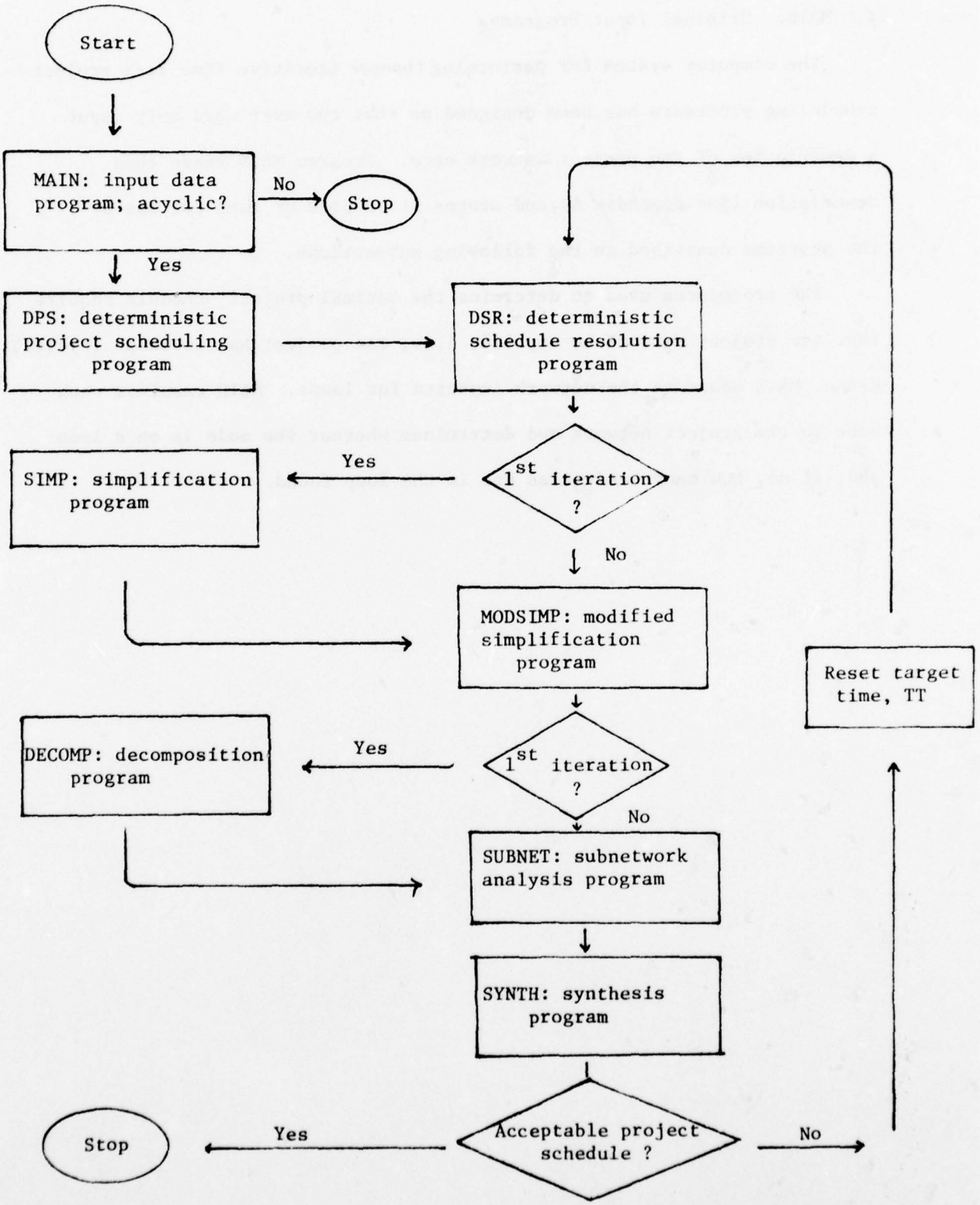
T	F(T)
93.97934	0.0
95.03941	0.0
96.09949	0.0
97.15956	0.00012
98.21964	0.00012
99.27971	0.00035
100.33979	0.00035
101.39986	0.00070
102.45993	0.00093
103.52001	0.00116
104.58008	0.00217
105.64016	0.00241
106.70023	0.00475
107.76031	0.00659
108.82038	0.00901
109.88046	0.01128
110.94053	0.01533
112.00061	0.01728
113.06068	0.02544
114.12076	0.03030
115.18083	0.04599
116.24090	0.05518
117.30098	0.07352
118.36105	0.08906
119.42113	0.10186
120.48120	0.14357
121.54128	0.18425
122.60135	0.21447
123.66143	0.29642
124.72150	0.31071
125.78158	0.39035
126.84165	0.42206
127.90172	0.50937
128.96180	0.53910
130.02187	0.61984
131.08195	0.68729
132.14202	0.72347
133.20210	0.76238
134.26217	0.81318
135.32225	0.83625
136.38232	0.88006
137.44240	0.90152
138.50247	0.91987
139.56255	0.94527
140.62262	0.96637
141.68269	0.97906
142.74277	0.97987
143.80284	0.99541
144.86292	0.99541
145.92299	1.00000

THE MEAN OF THE SYNTHESIZED DISTRIBUTION IS 128.42171

THE DIFFERENCE BETWEEN 128.42171 AND THE PROJECT DEADLINE IS 2.74 PERCENT OF THE PROJECT DEADLINE.
HENCE, THE NEW TARGET TIME IS 109.08782

```
**********************************************************************************************************************
THIS COMPLETES ITERATION      2
**********************************************************************************************************************
```

Appendix D. Flowchart of the Computer System



Appendix E

Program Descriptions

1. MAIN: Original Input Program

The computer system for performing the new iterative five step project scheduling procedure has been designed so that the user need only input a description of the project network once. Program MAIN reads this description (See Appendix A) and stores it on disk or tape for use by the programs described in the following subsections.

The procedures used to determine the optimal project schedule require that the project network be acyclic; i.e., the project has no loops (cycles). Hence, MAIN searches the network inputted for loops. MAIN examines each node in the project network and determines whether the node is on a loop and, if so, how many activities are in the loop found.

2. DPS: Deterministic Project Scheduling

The problem of finding a minimum cost project schedule which completes the project by TARGET TIME when each activity's duration is exactly its mean duration can be formulated as a linear programming problem. However, due to the large number of variables and constraints involved, a straightforward linear programming solution would be impractical. Instead the dual of this linear programming problem is considered, further reformulated, and then solved using the very efficient network-flow algorithm described in Dunn and Sielken (1977). This network-flow algorithm is a generalization of D. R. Fulkerson's algorithm (1961) for solving similar problems with linear activity cost functions. The generalized network-flow algorithm iteratively generates the minimum cost project schedule for every feasible deterministic completion deadline. The corresponding deterministic project cost curve is a convex piecewise linear function of TARGET TIME and a valuable description of the relationship between a project's cost and its deadline. The optimal activity mean durations are linear functions of the TARGET TIME on each linear piece of the project cost curve.

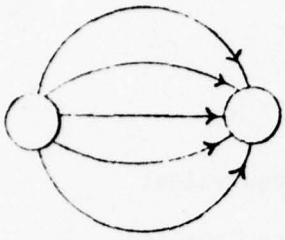
3. DSR: Deterministic Schedule Resolution

Since the deterministic project scheduler iteratively generates the optimal activity mean durations for all feasible completion deadlines, Step 1 is essentially only performed once. When the general iterative algorithm returns to Step 1 with a new TARGET TIME, finding the optimal activity mean durations is essentially a simple table look-up procedure. For a more complete detailed discussion of Deterministic Scheduling see Dunn and Sielken (1977).

4. SIMP: Simplification

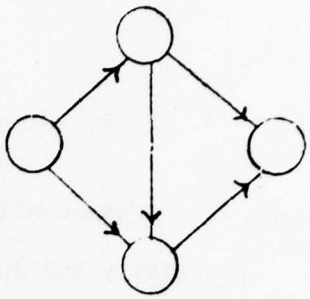
Five configurations of activities for which a single equivalent activity and duration distribution are readily available are depicted in Figure E.1. The equivalent single activity duration distributions for the parallel, series, and Wheatstone Bridge configurations were originally identified by Hartley and Wortham (1966) and for the Double Wheatstone Bridge and Criss-Cross configurations by Ringer (1969).

Simplification is an iterative procedure as illustrated in Figure E.2. In the special case where Simplification reduces the project network down to just one activity as in Figure E.2, the project completion time distribution is directly determined so that Steps 3 and 4 are skipped. Although a reduction to one activity is a very special case, reductions of over 50% are quite common.

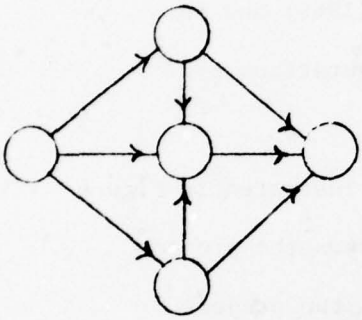


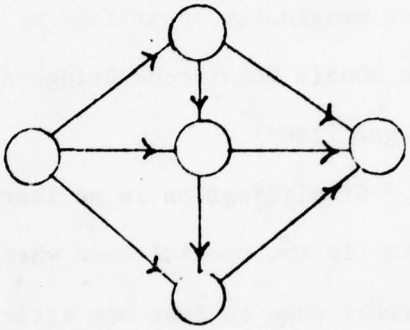
Activities in Parallel

Two Activities in Series

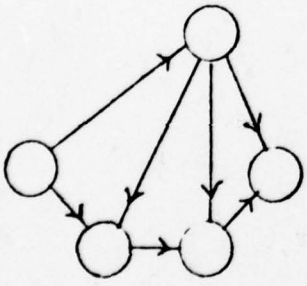


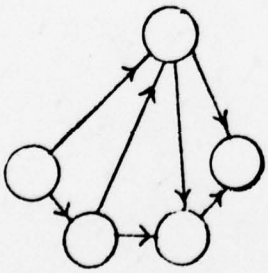
Wheatstone Bridge

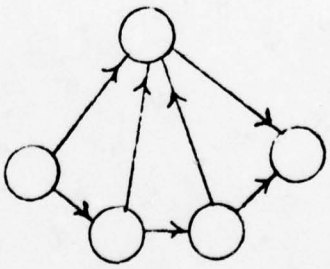




Double Wheatstone Bridges



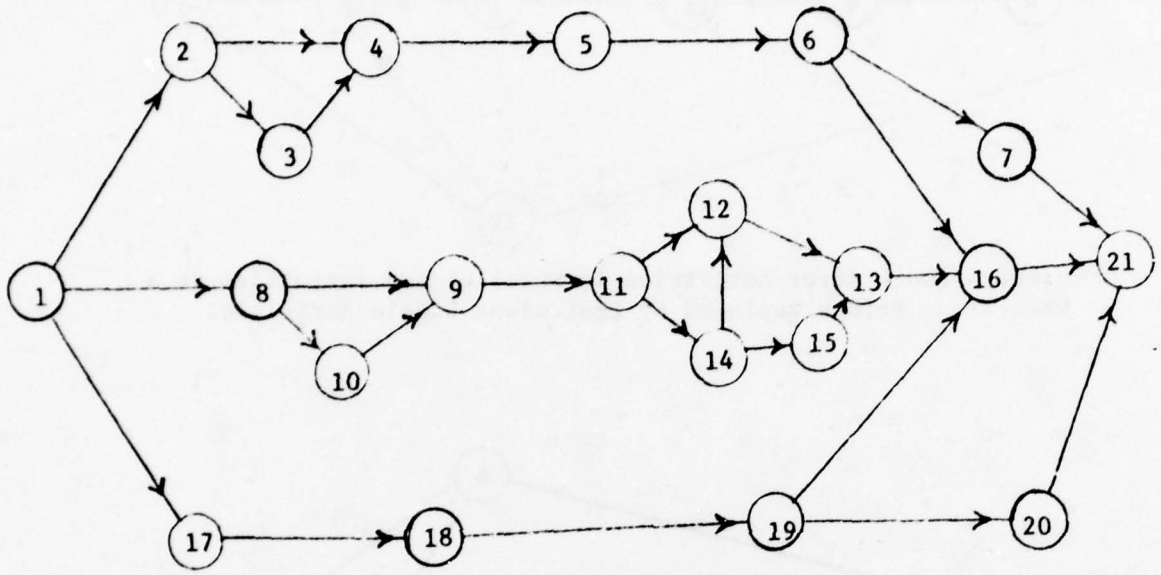




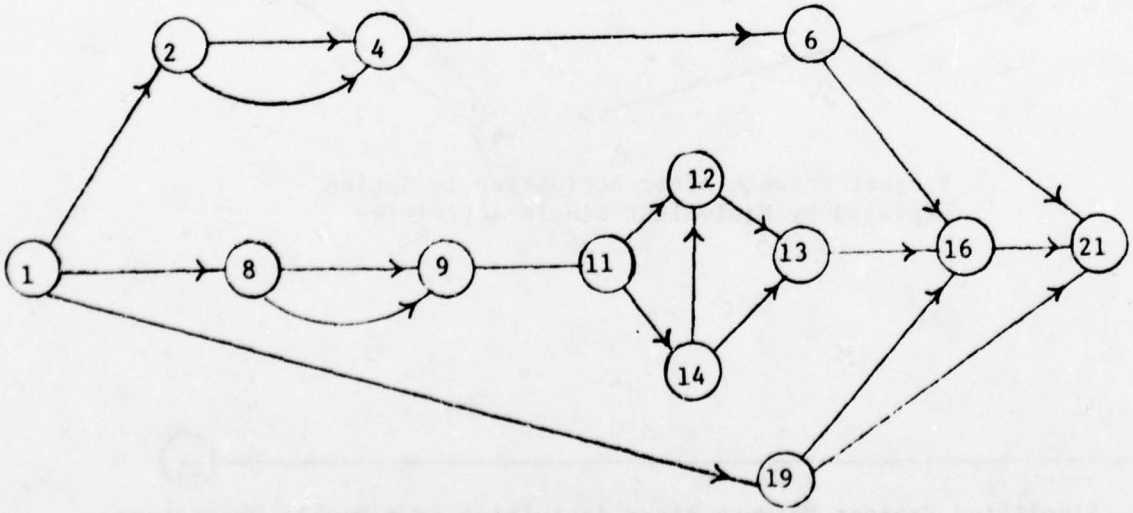
Criss - Crosses

Figure E.1 Activity Configurations Which Can be Readily Replaced by a Single Equivalent Activity

Figure E.2 The Iterative Simplification of a Project Network

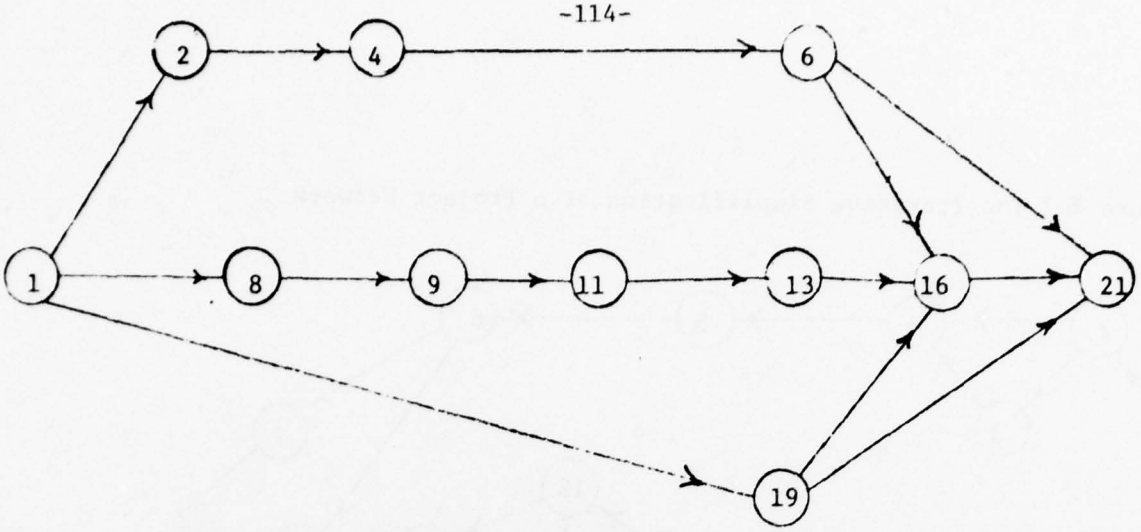


Original Project Network

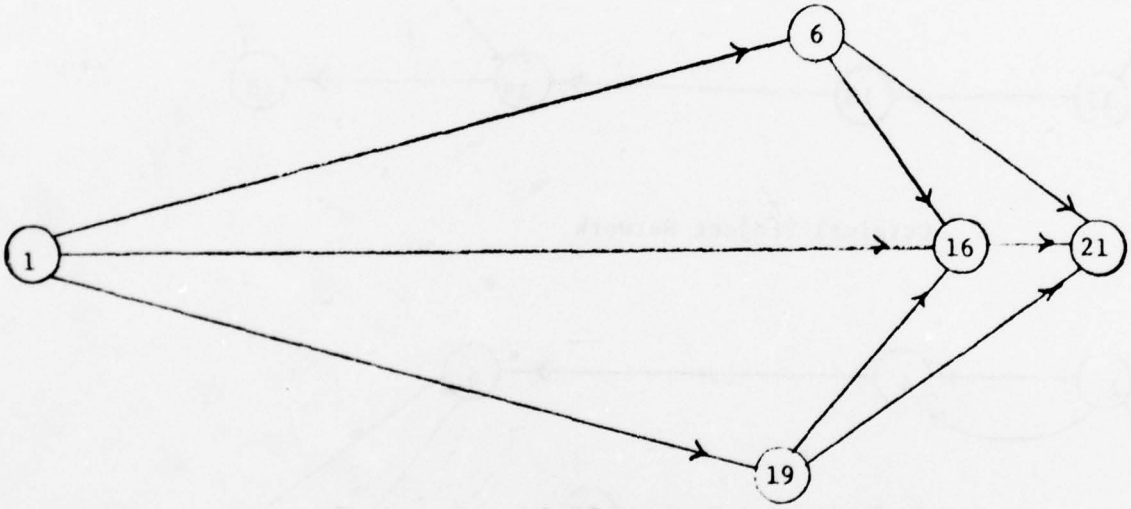


Project Network After Activities in Series
Replaced by Equivalent Single Activities

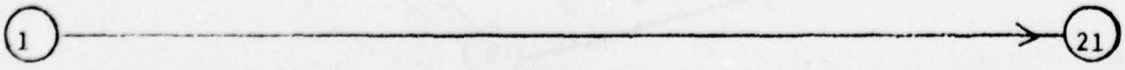
Figure E.2(Continued)



Project Network After Activities in Parallel and Activities in a Wheatstone Bridge Replaced by Equivalent Single Activities



Project Network After Activities in Series Replaced by Equivalent Single Activities



Simplified Project Network After Activities in a Double Wheatstone Bridge Replaced by an Equivalent Single Activity

5. MODSIMP: Subsequent Simplification

The structure of the simplified network is identified by SIMP. MODSIMP determines the distribution of the activity durations in the simplified network.

6. SUBNET: Subnetwork Analysis

6.1 Introduction

The objective of Subnetwork Analysis is to determine each subnetwork's duration distribution.

At the end of Step 2 each activity in the subnetwork has a specified duration distribution. This distribution is now approximated by a two-point discrete distribution. In particular, an activity, say A, is now conceptualized as having two possible duration times, say ℓ_A for a lower duration and u_A for an upper duration. The probability that the activity duration is ℓ_A is assumed to be P_A, and correspondingly the probability that the activity duration is u_A is assumed to be $Q_A = 1 - P_A$. The values of ℓ_A, u_A, and P_A are chosen so that the mean, variance, and third moment of the discrete distribution are the same as the mean, variance, and third moment of activity A's specified duration distribution.

Let n be the number of activities in the subnetwork. Let $v = 1, 2, \ldots, 2^n$ index the 2^n possible configurations of activity durations when each activity is either at its upper duration or at its lower duration. Let

$$p_v = \text{probability of the v-th activity duration configuration}$$

$$= \prod_{i=1}^{n} [P_i(1 - \delta_{v,i}) + Q_i\delta_{v,i}] \qquad (6.1)$$

where

$\delta_{v,i} = 1$ if the duration for the i-th activity is u_i in the v-th activity duration configuration

$= 0$ if the duration for the i-th activity is ℓ_i in the v-th activity duration configuration. (6.2)

Then the subnetwork duration distribution when each activity has its two-point discrete distribution is

$$F(t) = \sum_{v=1}^{2^n} p_v I_t(t_v) \tag{6.3}$$

where

t_v = the subnetwork duration when the activity durations are in the v-th configuration (6.4)

and

$$I_t(t_v) = 1 \quad \text{if } t_v \leq t ,$$
$$= 0 \quad \text{if } t_v > t . \tag{6.5}$$

The discrete distribution function F is an approximation to the subnetwork's exact duration distribution.

The goal of Subnetwork Analysis is to determine F.

Since the number, n, of activities in the subnetwork may be fairly large, the complete enumeration of the 2^n discrete subnetwork durations may sometimes be impractical. When this happens, the discrete subnetwork duration distribution F must be approximated. The approximation of F will be based on the activities which are mostly likely to influence the subnetwork duration. The identification of these important activities and their interrelationships is discussed in the next subsection which is a review of the procedures originating in Sielken,

Ringer, Hartley, and Arseven (1974) and Sielken, Hartley, and Spoeri (1976).

Each subnetwork is assumed to be an acyclic network with one source, one sink, and no cut vertices.

6.2 Formation of Clusters

The mean duration for activity A is

$$m_A = P_A \ell_A + Q_A u_A \ , \tag{6.6}$$

and the standard deviation of activity A's duration is

$$s_A = [P_A \ell_A^2 + Q_A u_A^2 - m_A^2]^{\frac{1}{2}} \ . \tag{6.7}$$

When each activity duration takes on a fixed (nonrandom) value, the subnetwork's duration is the duration of the longest path through the subnetwork where the "length" of an activity is its duration. For example, consider the subnetwork described in Table E.1 and displayed in Figure E.3. When each activity duration is its mean duration, then the subnetwork's duration is 32, corresponding to the path consisting of activities 2, 7, and 9.

Definition 1: A critical activity is an activity on the longest path when all the subnetwork's activity durations are set to their means.

Thus in the example the critical activities are 2, 7, and 9.

The search for the activities which are most likely to influence the subnetwork duration begins with the critical activities. Each critical activity initiates a separate set of activities called a

TABLE E.1

Activity Durations for the Subnetwork in Figure 2

Activity	ℓ_A	u_A	P_A	m_A	s_A
1	0.00	2.00	.5	1	1
2	8.00	10.50	.2	10	1
3	9.55	15.67	.6	12	3
4	1.50	4.00	.8	2	1
5	3.35	5.52	.7	4	1
6	4.00	6.00	.5	5	1
7	8.73	16.90	.6	12	4
8	12.00	14.50	.2	14	1
9	5.00	15.00	.5	10	5

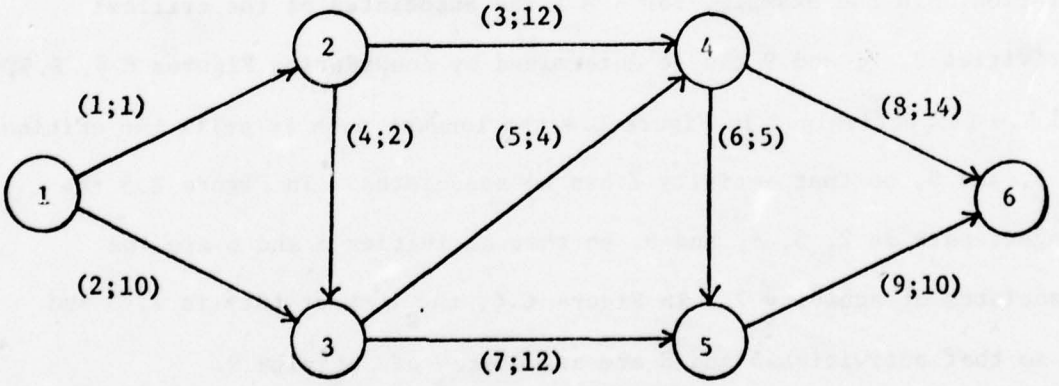


Figure E.3

Subnetwork with activities labeled (activity number; mean duration).

"cluster". Initially there are several clusters. In the example the initial clusters are

$$C_1 = \{2\}, \quad C_2 = \{7\}, \quad \text{and } C_3 = \{9\} . \tag{6.8}$$

Some of the non-critical activities may influence the subnetwork's duration when not all of the activity durations are at their mean values.

Definition 2: An associate of a critical activity A is a non-critical activity which is on the longest path when all activity durations are set to their mean except for the critical activity A which has its duration reduced from m_A to $\max(m_A - \lambda s_A, 0)$ where λ is a nonnegative parameter.

Thus the associates of a critical activity A are those activities whose effect on the subnetwork's duration are related to activity A's duration. In the example, for $\lambda = 1$ the associates of the critical activities 2, 7, and 9 can be determined by considering Figures E.4, E.5, and E.6 respectively. In Figure E.4 the longest path is still the critical path 2, 7, and 9, so that activity 2 has no associates. In Figure E.5 the longest path is 2, 5, 6, and 9, so that activities 5 and 6 are the associates of activity 7. In Figure E.6, the longest path is 2, 5 and 8, so that activities 5 and 8 are associates of activity 9.

The associates of each critical activity are determined and added to the cluster containing that critical activity. Thus, in the example the clusters are expanded to

$$C_1 = \{2\}, \quad C_2 = \{7, 5, 6\}, \quad \text{and } C_3 = \{9, 5, 8\} . \tag{6.9}$$

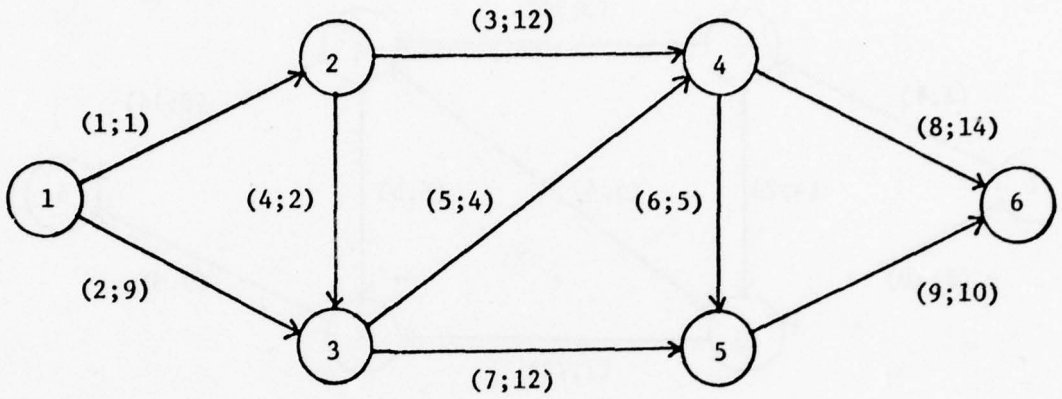


Figure E.4

Subnetwork for determining the associates of Activity 2 when $\lambda = 1$.

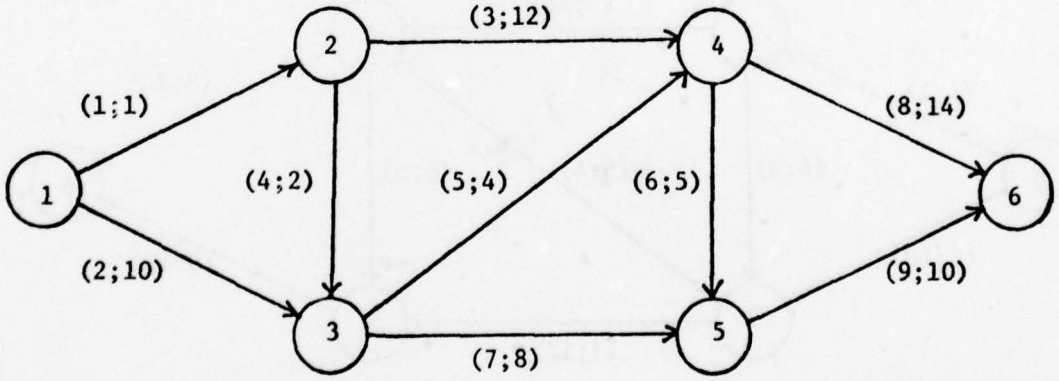


Figure E.5

Subnetwork for determining the associates of Activity 7 when $\lambda = 1$.

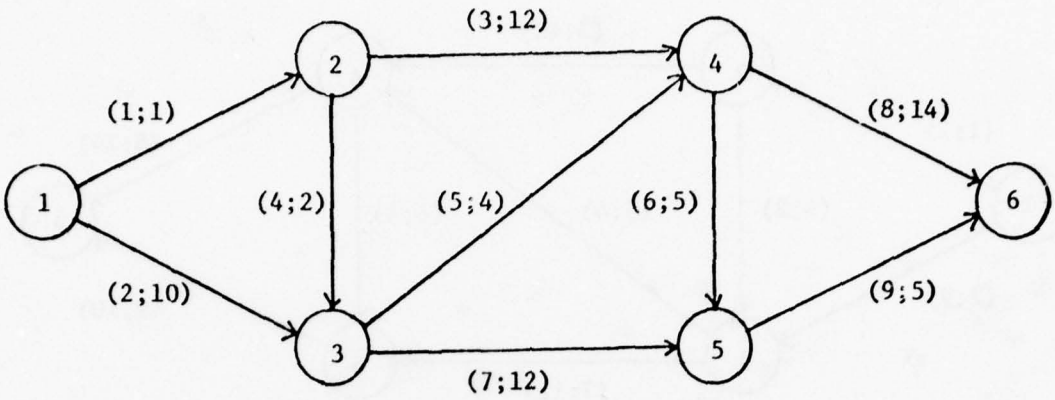


Figure E.6

Subnetwork for determining the associates of Activity 9 when $\lambda = 1$.

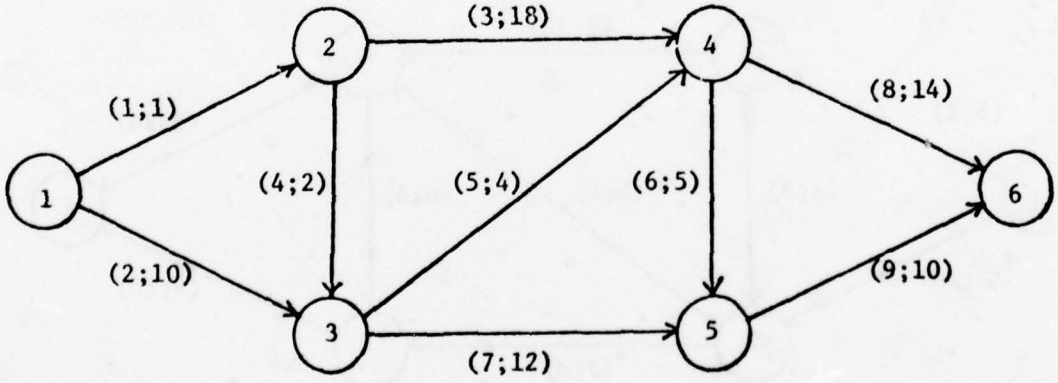


Figure E.7

Subnetwork for determining the eliminants of Activity 3 when $\theta = 2$.

The idea underlying the clusters is that they should be sets of activities whose effects on the subnetwork's duration are interrelated. Thus, if two clusters contain any activities in common, the activities in these two clusters all have an interrelated effect on the subnetwork's duration, so the two clusters are combined into one cluster. In the example clusters C_2 and C_3 both contain activity 5, so they are combined. The resulting clusters are

$$C_1 = \{2\} \text{ and } C_2 = \{5, 6, 7, 8, 9\} . \tag{6.10}$$

A non-critical activity may also influence the subnetwork's duration if its duration exceeds its mean.

Definition 3: An eliminant of a non-critical activity A is a critical activity which is not on the longest path when all activity durations are set to their means except for activity A which has its duration increased from m_A to $m_A + \theta s_A$ where θ is a nonnegative parameter.

For instance, if $\theta = 2$, the eliminants of the non-critical activity 3 in the example can be determined from Figure E.7. There the longest path is 1, 3, 6, and 9, so that the eliminants of activity 3 are the critical activities 2 and 7. In the example, when $\theta = 2$, none of the other non-critical activities (1, 4, 5, 6, and 8) have any eliminants. For a specified value of θ the eliminants of every non-critical activity are determined. If a non-critical activity A has eliminants, then the effect of A's eliminants on the subnetwork duration is related to A's duration, so A is added to every cluster containing at least one

of its eliminants. Thus in the example the clusters become

$$C_1 = \{2, 3\} \text{ and } C_2 = \{3, 5, 6, 7, 8, 9\} . \qquad (6.11)$$

After the clusters have been expanded on the basis of eliminants, any two clusters containing common elements are combined. Therefore in the example, C_1 and C_2 are combined to form a single cluster

$$C_1 = \{2, 3, 5, 6, 7, 8, 9\} . \qquad (6.12)$$

In general, after the determination of associates and eliminants for specified values of λ and θ and the subsequent combining of clusters, there may still be more than one cluster and some of the non-critical activities may not be in any cluster. Usually the larger the values of λ and θ the greater the number of activities in the clusters and the smaller the number of clusters. The clusters that remain represent sets of activities such that the effects on the subnetwork's duration of the activity durations for the activities within a set are all interrelated. Activities in different clusters have roughly independent effects on the subnetwork's duration. Activities not in any cluster have essentially no effect on the subnetwork's duration.

The consideration of critical activities, associates, eliminants, and the formation of clusters of related activities is obviously only one way of identifying the activities which have an important effect on the subnetwork's duration and their interrelationships. However, this particular procedure does have the following desirable properties:

Property 1: If $\lambda_2 > \lambda_1$, then any activity which would be an associate of a critical activity A when $\lambda = \lambda_1$ would also be an associate of A when $\lambda = \lambda_2$.

Property 2: If $\theta_2 > \theta_1$, then any critical activity which would be an eliminant of a non-critical activity A when $\theta = \theta_1$ would also be an eliminant of A when $\theta = \theta_2$.

Property 3: For any fixed value of λ, the set of activities in the union of the clusters is monotically nondecreasing as $\theta \to \infty$.

Property 4: The number of clusters is nonincreasing as $\theta \to \infty$.

Property 5: If $s_A > 0$ for a non-critical activity A, then there exists $\theta_A < \infty$ such that A will have some eliminants for any $\theta \geq \theta_A$.

Property 6: If $s_A > 0$ for every non-critical activity A and

$$\theta^* = \max\{\theta_A;\ A \text{ non-critical}\} ,$$

then for $\theta \geq \theta^*$ all activities will be in one cluster.

Most of these properties are fairly straightforward; however, Property 6 requires some special justification. This justification is based on the following definition and theorem which is proven in Sielken, Ringer, Hartley, and Arseven (1974).

Definition 4: In any acyclic network a bridge over any two consecutive arcs A_1 and A_2 is any arc A_3 such that all paths from the source to the sink passing through A_3 do not pass through either A_1 or A_2.

Theorem 1: In any acyclic network with no cut vertices there is at least one bridge for any pair of consecutive arcs.

Property 5 implies that all activities will belong to some cluster if $\theta \geq \theta^*$. Now consider any two consecutive activities A_1 and A_2 on the critical path. Theorem 1 implies that there is a bridge over A_1 and A_2, say A_3. Since the critical path passes through A_1 and A_2, A_3 cannot be on the critical path. Therefore, if $\theta \geq \theta^* \geq \theta_{A_3}$, A_1 and A_2 will be eliminants of A_3 and hence will be in the same cluster as A_3. Thus, since each cluster contains at least one original critical activity and any two consecutive critical path activities belong to the same cluster when $\theta \geq \theta^*$, there is only one cluster when $\theta \geq \theta^*$ and Property 6 is established.

6.3 Bounding the Discrete Subnetwork Duration Distribution F

6.3.1 Upper Bounds on F

Suppose that the cluster formation procedure described in subsection 6.2 has been carried out on a subnetwork for some specified values of θ and λ and yielded K clusters. For each cluster C so determined, let n_c be the number of activities in the cluster and let $v = 1, \ldots, 2^{n_c}$ index the 2^{n_c} configurations of activity durations corresponding to

(a) the duration for each activity A not in C being equal to its lower point ℓ_A, and

(b) the durations for the activities in C being at each of the 2^{n_c} possible combinations of their upper and lower points.

Then define

$$F^{+}(C;\ t) = \sum_{v=1}^{n_c} p_v I_t(t_v) \qquad (6.13)$$

where p_v, t_v, and $I_t(t_v)$ are defined in (6.1), (6.4), and (6.5) respectively. The distribution function $F^{+}(C;\ t)$ is an upper bound on F. This can be shown by considering the following:

Theorem 2: For any cluster C, any t, and any activity A not in C,

$$F^{+}(C \cup \{A\};\ t) \leq F^{+}(C;\ t) .$$

(For the proof of this theorem, see Sielken, Hartley, and Spoeri (1975).) A straightforward application of Theorem 2 yields

Theorem 3: For any two clusters C_1 and C_2 and any t,

$$F^{+}(C_1 \cup C_2;\ t) \leq \min\{F^{+}(C_1;\ t),\ F^{+}(C_2;\ t)\} .$$

If C* represents the set (cluster) of all activities in the subnetwork, then

$$F(t) = F^{+}(C^*;\ t) . \qquad (6.14)$$

Since C is a subset of C*, either Theorem 2 or Theorem 3 implies

$$F^{+}(C;\ t) \geq F(t) \qquad (6.15)$$

for any cluster C.

Theorems 2 and 3 can also be used to define some tighter upper bounds on the subnetwork's duration distribution than $F^{+}(C;\ t)$. Historically, two different improved bounds have been employed, and both have been incorporated into the current subnetwork analysis

procedure. They are

$$F_1^+(t;\ \theta,\ \lambda) = F^+\left(\bigcup_{i=1}^{K} C_i;\ t\right) \tag{6.16}$$

and

$$F_2^+(t;\ \theta,\ \lambda) = \min_{1 \le i \le K} F^+(C_i;\ t)\ . \tag{6.17}$$

Let $F^+(t;\ \theta,\ \lambda)$ denote either $F_1^+(t;\ \theta,\ \lambda)$ or $F_2^+(t;\ \theta,\ \lambda)$. Then, since Property 2 of the cluster formation procedure implies that as θ increases the clusters expand or are combined, Theorems 2 and 3 imply that $F^+(t;\ \theta,\ \lambda)$ is a nonincreasing function of θ for every t and any λ. Property 6 and (6.14) imply that for $\theta \ge \theta^*$

$$F^+(t;\ \theta,\ \lambda) = F(t) \tag{6.18}$$

for every t and any λ. Also (6.14) along with the definitions (6.16) and (6.17) imply

$$F^+(t;\ \theta,\ \lambda) \ge F(t) \tag{6.19}$$

for all t, θ, and λ. These results are summarized in Theorem 4.

Theorem 4: (a) $F^+(t;\ \theta,\ \lambda)$ is a nonincreasing function of θ for every t and any λ;

(b) there exists a finite value θ^* such that $\theta \ge \theta^*$ implies $F^+(t;\ \theta,\ \lambda) = F(t)$ for every t and λ; and

(c) for any θ, λ, and t

$$F^+(t;\ \theta,\ \lambda) \ge F(t)\ .$$

6.3.2 Lower Bounds on F

Let n_c denote the number of activities in cluster C, and let $v = 1, \ldots, 2^{n_c}$ index the 2^{n_c} configuration of activity durations corresponding to

(a) the duration for each activity A not in the cluster being equal to its upper point u_A, and

(b) the durations for activities in the cluster being at each of the 2^{n_c} possible combinations of the upper and lower points.

Then define

$$F^-(C;\ t) = \sum_{v=1}^{2^{n_c}} p_v I_t(t_v) \tag{6.20}$$

where p_v, t_v, and $I_t(t_v)$ are as previously defined. Also define

$$F_1^-(t;\ \theta,\ \lambda) = F^-\left(\bigcup_{i=1}^{K} C_i;\ t\right) \tag{6.21}$$

and

$$F_2^-(t;\ \theta,\ \lambda) = \max_{1 \le i \le K} F^-(C_i;\ t)\ . \tag{6.22}$$

Using an argument completely analagous to that used to prove Theorem 4, Sielken, Hartley, and Spoeri (1975) also proved

Theorem 5: (a) $F^-(t;\ \theta,\ \lambda)$ is a nondecreasing function of θ for any fixed value of λ;

(b) there exists a finite value θ^* such that $\theta \ge \theta^*$ implies

$$F^-(t;\ \theta,\ \lambda) = F(t)$$

for every t and any λ; and

(c) for any θ, λ, and t

$$F^{-}(t;\ \theta,\ \lambda) \leq F(t) \ .$$

(Again, $F^{-}(t;\ \theta,\ \lambda)$ is a generic term used to denote either $F_1^{-}(t;\ \theta,\ \lambda)$ or $F_2^{-}(t;\ \theta,\ \lambda)$.) Thus, $F^{-}(t;\ \theta,\ \lambda)$ is a valid lower bound on F.

6.3.3 The Tightness of the Bounds on F

That the F_1-bounds are tighter than the F_2-bounds can be seen as follows. The evaluation of $F_2^{-}(t;\ \theta,\ \lambda)$ involves the determination of $F^{-}(C_i;\ t)$ for each i whereas $F_1^{-}(t;\ \theta,\ \lambda) = F^{-}(\underset{i=1}{\overset{K}{U}} C_i,\ t)$. Let L_i be the length of the longest path when

1) the activities in C_i are at a particular configuration of their upper and lower durations and

2) all activities not in C_i have their upper durations.

Let L_U be the length of the longest path when

1) the configuration of upper and lower durations for the activities in C_i is the same as in the determination of L_i,

2) the activities in $\underset{j=1}{\overset{K}{U}} C_j - C_i$ are at <u>any</u> combination of their upper and lower durations, and

3) all activities not in $\underset{j=1}{\overset{K}{U}} C_j$ have their upper durations.

Then $L_i \geq L_U$ since every activity duration in the determination of L_i is greater than or equal to its corresponding duration in the determination of L_U. Since $L_i \geq L_U$ for any configuration of upper and lower durations for the activities in C_i,

$$F^{-}(\underset{j=1}{\overset{K}{U}} C_j;\ t) \geq F^{-}(C_i;\ t) \qquad (6.23)$$

and

$$F_1^-(t;\ \theta,\ \lambda) = F^-\left(\bigcup_{j=1}^{K} C_j;\ t\right) \geq \max_{1\leq i\leq K} F^-(C_i;\ t) = F_2^-(t;\ \theta,\ \lambda)\ . \qquad (6.24)$$

A similar argument can be used to show

$$F_1^+(t;\ \theta,\ \lambda) = F^+\left(\bigcup_{j=1}^{K} C_j;\ t\right) \leq \min_{1\leq i\leq K} F^+(C_i;\ t) = F_2^+(t;\ \theta,\ \lambda)\ . \qquad (6.25)$$

The extent of the differences between the two upper bounds and two lower bounds depends heavily on the structure of the particular subnetwork being analyzed and is a topic that should be considered in future empirical studies.

6.4 Using Sampling to Estimate the Upper and Lower Bounds on F

The only instance in which upper and lower bounds on F are computed rather than F itself is when it is computationally impractical to determine the longest path for each of the 2^n activity duration configurations.

For given θ and λ, the evaluation of $F_1^+(t;\ \theta,\ \lambda)$ only requires the determination of the longest path for each of 2^{n_U} activity configurations where n_U is the number of activities in the union of the clusters $C_o = \bigcup_{j=1}^{K} C_j$; i.e.,

$$n_U = \sum_{j=1}^{K} n_j\ . \qquad (6.26)$$

The evaluation of $F_1^-(t;\ \theta,\ \lambda)$ also entails only 2^{n_U} longest path determinations. Likewise, the evaluation of $F_2^+(t;\ \theta,\ \lambda)$ or $F_2^-(t;\ \theta,\ \lambda)$ only requires the determination of the longest path for each of

$$n_s = \sum_{i=1}^{K} 2^{n_i} \qquad (6.27)$$

activity configurations. Since 2^{n_U} is always greater than or equal to

n_s, $F_2^+(t;\ \theta,\ \lambda)$ and $F_2^-(t;\ \theta,\ \lambda)$ are the most economical bounds to compute in terms of the number of longest path determinations required. However, for any given θ and λ, $F_1^+(t;\ \theta,\ \lambda)$ and $F_1^-(t;\ \theta,\ \lambda)$ are tighter bounds than $F_2^+(t;\ \theta,\ \lambda)$ and $F_2^-(t;\ \theta,\ \lambda)$, respectively. Thus, in making the choice of which one of the two sets of bounds to compute, there is a trade-off between the accuracy of the bounds and the effort required to compute them.

Since the cluster formation procedure is such that the clusters expand or are pooled as θ increases, it may happen that for particular θ and λ, 2^{n_U} and 2^{n_i} for some i are both quite large even though θ is only moderately large. In this case it again becomes impractical to examine <u>all</u> the required activity configurations involved in determining either the F_1-bounds or the F_2-bounds. Consequently, if for the specified values of θ and λ, 2^{n_U} (or 2^{n_i} for some i, as the case may be) is excessively large, Subnetwork Analysis will compute estimates of the corresponding upper and lower bounds based on only a sample of the total number of possible configurations. The actual estimators of $F^+(C;t)$ or $F^-(C;t)$ based on sample values $x_1 \leq x_2 \leq \cdots \leq x_m$ for cluster C is

$$G(t) = \frac{\sum_{i=1}^{m} p_i I_t(x_i)}{\sum_{i=1}^{m} p_i} \tag{6.28}$$

where p_i and $I_t(\cdot)$ are as in (6.1) and (6.5) respectively. The sample values are determined by finding the longest path for each member of a systematic sample of the possible activity duration configurations for

the cluster. However, if the cluster contains more than 31 activities, the sample values are determined by random sampling. The estimator, G(t), and the sampling procedures used in its evaluation are developed and discussed in Baker and Sielken (1978).

6.5 Estimating F by Extrapolating Between the Upper and Lower Bounds of F

Theorems 4 and 5 of subsection 6.3 imply that if $\theta_{i+1} \geq \theta_i$ and $\lambda_{i+1} \geq \lambda_i$ for all i = 1, ..., I then

$$F^+(t; \theta_1, \lambda_1) \geq F^+(t; \theta_2, \lambda_2) \geq \ldots \geq F^+(t; \theta_I, \lambda_I) \geq F(t) \geq$$

$$F^-(t; \theta_I, \lambda_I) \geq F^-(t; \theta_{I-1}, \lambda_{I-1}) \geq \ldots \geq F^-(t; \theta_1, \lambda_1) \qquad (6.29)$$

for all t. Thus, if $F^+(t; \theta, \lambda)$ and $F^-(t; \theta, \lambda)$ have been calculated for I pairs (θ_i, λ_i) i = 1, ..., I $(\theta_{i+1} \geq \theta_i, \lambda_{i+1} \geq \lambda_i)$, then F(t) may be estimated by extrapolating between $F^+(t; \theta_I, \lambda_I)$ and $F^-(t; \theta_I, \lambda_I)$. As currently written, Subnetwork Analysis calculates upper and lower bounds on the subnetwork's approximate duration distribution for a sequence of three (θ, λ) pairs, $(\theta, \lambda) = (1, 1)$, (2, 2), (3, 2). An extrapolation procedure is then used to obtain an estimate of F. The procedure that has been developed for this purpose is documented in Baker and Sielken (1978).

6.6 A Summary of the Subnetwork Analysis Procedure

The following is a step-by-step description of the subnetwork analysis procedure in summary form. Recall that the objective of

Subnetwork Analysis is to determine an "approximation", say $\hat{F}$, to the subnetwork's duration distribution.

(a) If n = 1, let $\hat{F}$ be the actual activity duration distribution for the one activity comprising the subnetwork, and stop. Otherwise, go to Step b.

(b) Identify the two-point discrete distribution (ℓ_A, u_A, P_A, Q_A) for every activity A in the subnetwork.

(c) Ascertain the user's choice of

(1) NMAX, the maximum value of m for which all 2^m activity duration configurations are to be explicitly considered,

(2) the (θ, λ) pairs to be considered if not the standard pairs (1, 1), (2, 2), and (3, 2),

(3) whether the bounds on F are to be (F_1^-, F_1^+) or (F_2^-, F_2^+) if n > NMAX, and

(4) SAMSIZ, the sample size to be taken if, in the determination of bounds on F for some (θ, λ) pair, the number of activity configurations in the cluster being considered exceeds 2^{NMAX}.

(d) If the number of activities in the subnetwork doesn't exceed NMAX, compute the subnetwork's discrete duration distribution, F, explicitly, let $\hat{F} = F$, and stop. Otherwise, go to Step e.

(e) Do Steps f - i for every (θ, λ) pair. Then go to Step j.

(f) Form the clusters corresponding to (θ, λ). If the bounds are to be (F_1^-, F_1^+), go to Step g. If the bounds are to be (F_2^-, F_2^+), go to Step h.

(g) Form the union of the clusters and determine n_U. If $n_U \leq$ NMAX, evaluate the bounds (F_1^-, F_1^+) on the basis of all 2^{n_U} activity duration configurations. If $n_U >$ NMAX, take a sample of size SAMSIZ from the 2^{n_U} activity duration configurations and form both F_1^- and F_1^+ on the basis of this single sample. Go to Step e.

(h) Do the following for each cluster, C_i. Let n_i denote the number of activities in the cluster. If $n_i \leq$ NMAX, evaluate $F^-(C_i; t)$ and $F^+(C_i; t)$ on the basis of all 2^{n_i} activity duration configurations. If $n_i >$ NMAX, take a sample of size SAMSIZ from the 2^{n_i} activity duration configurations and form both $F^-(C_i; t)$ and $F^+(C_i; t)$ on the basis of this single sample.

(i) Form F_2^- and F_2^+ from the $F^-(C_i; t)$'s and $F^+(C_i; t)$'s respectively. Go to Step e.

(j) Form $\hat{F}$ by extrapolating the (F^-, F^+) bounds determined for the (θ, λ) pairs. Stop.

This process is repeated for every subnetwork in the simplified project network.

7. DECOMP: Decomposition

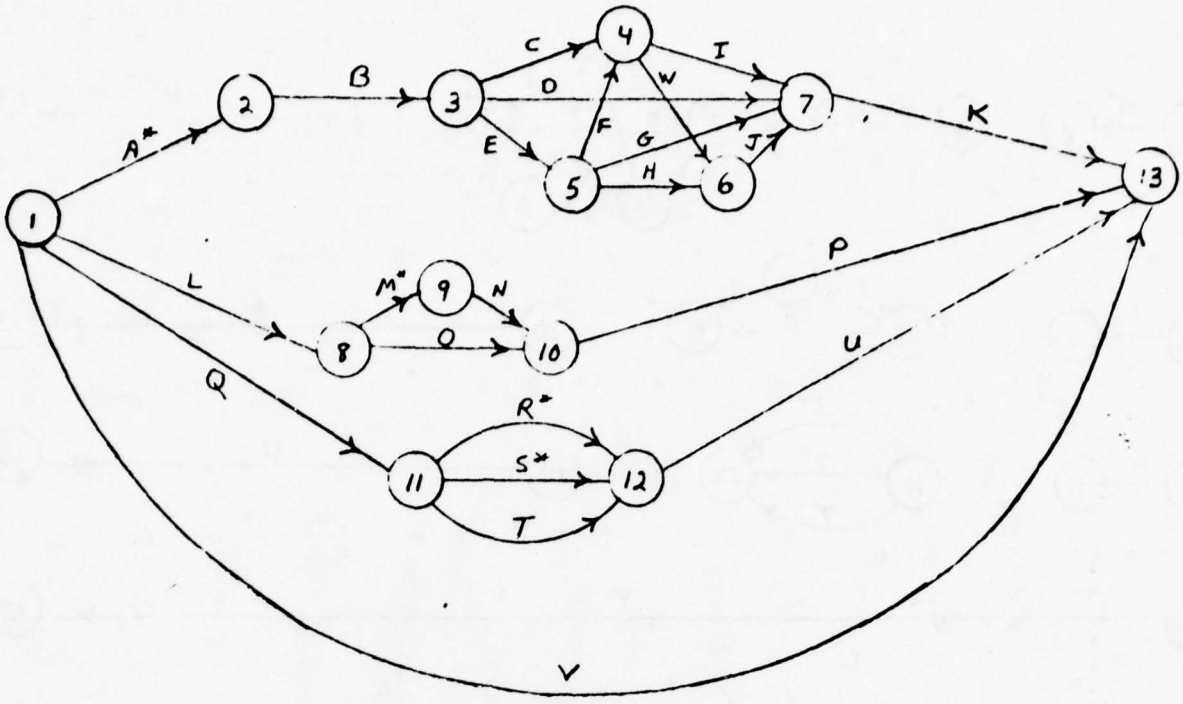
The objective of Deocmposition is to partition the simplified project network into the simplest possible subnetworks subject to the constaint that the subnetwork duration distributions can be easily combined to yield the project completion time distribution.

The simplified project network can be suitably partitioned using the following iterative procedure. First the simplified network is searched for subnetworks that begin at the beginning of the simplified network, end at the end of the simplified network, and are in parallel. Each parallel subnetwork is then subdivided into a sequence of smaller subnetworks that are in series. Each series subnetwork is then searched for parallel subnetworks. The partitioning into parallel and series subnetworks continues until no subnetwork can be further partitioned. This iterative partitioning procedure is illustrated in Figure E.8.

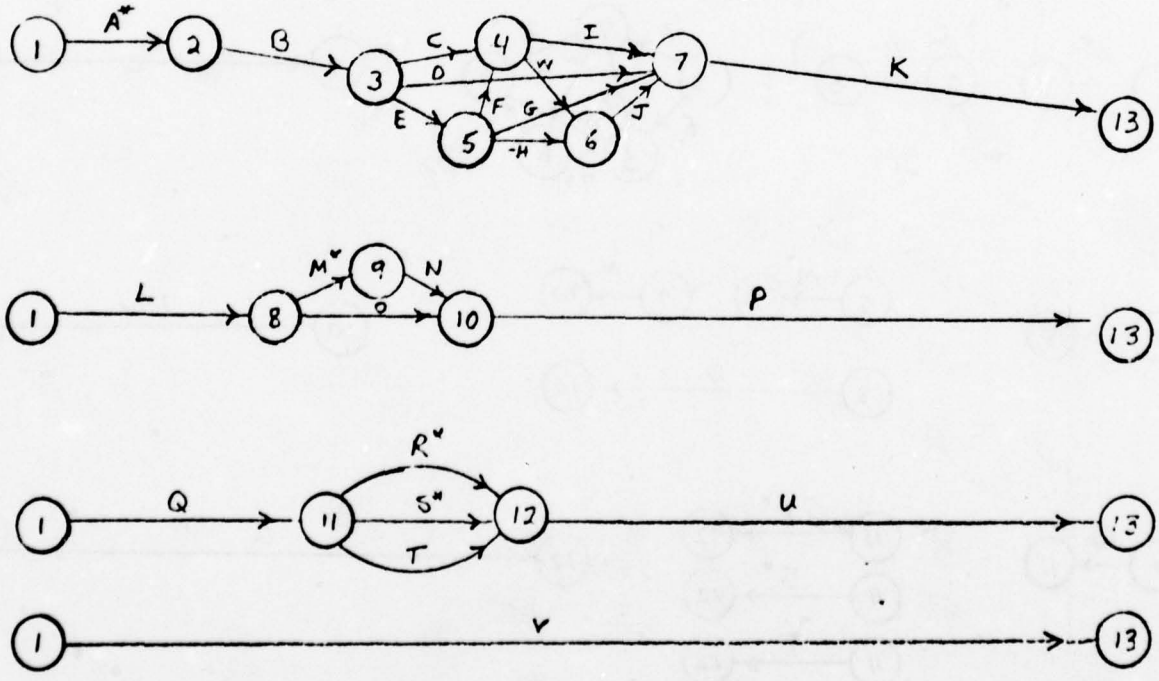
As in Simplification this partitioning of the simplified project network into subnetworks could have also included subnetwork configurations of the Wheatstone Bridge, Double Wheatstone Bridge, and Criss-Cross forms; however, the apparent frequency of these subnetwork configurations does not seem to justify the additional programming effort.

Since Steps 1 and 2 do not change the structure of the simplified project network and the partitioning of that network does not depend on the activity duration distributions, Decomposition is only done once and is really skipped when the general iterative algorithm returns to Step 3. Decomposition is documented in Sielken and Fisher (1976).

Figure E.8 The Decomposition of a Simplified Project Network*



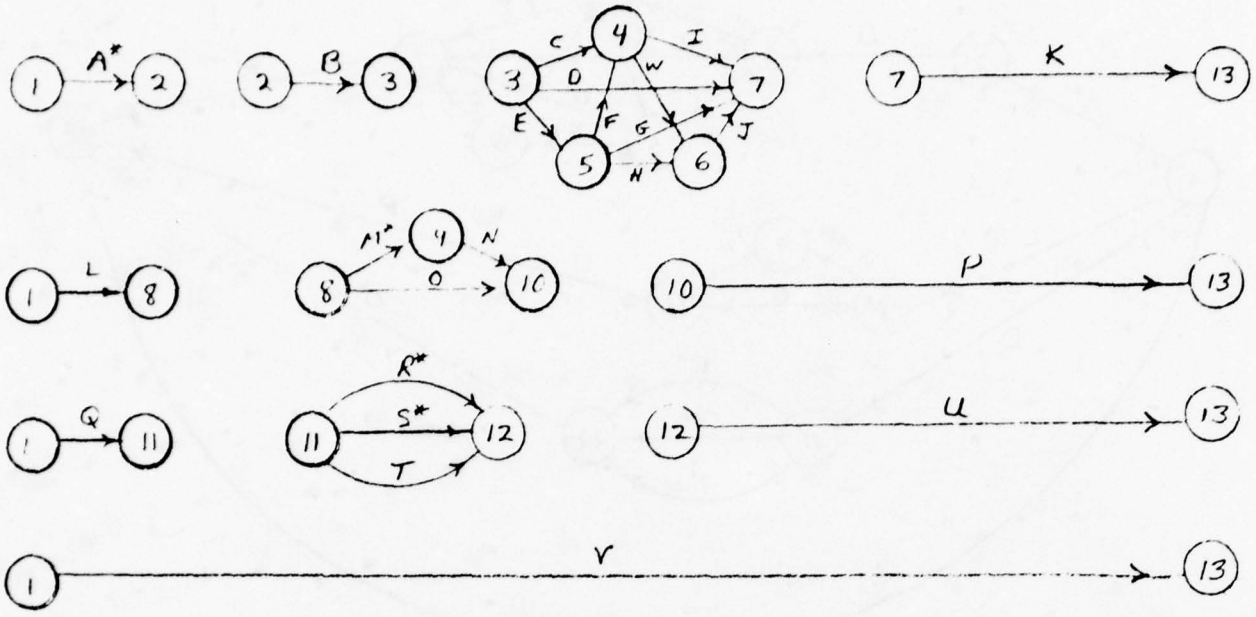
Simplified Project Network



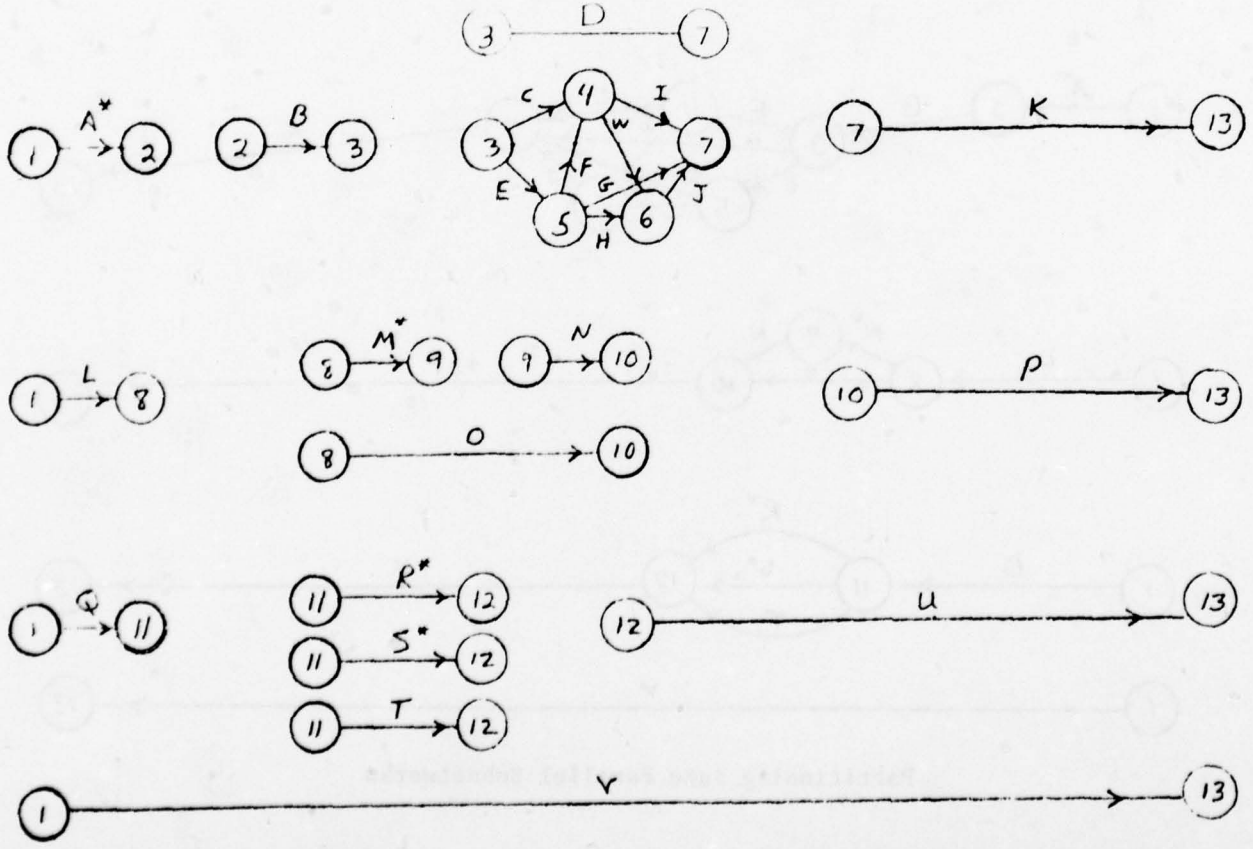
Partitioning into Parallel Subnetworks

The arcs A, M*, R*, and S* each represent non-simplifiable subnetworks and not single activities. Otherwise this "Simplified Project Network" could be further simplified.

Figure E.8 (Continued)



Partitioning into Series Subnetworks



Final Partitioned Simplified Project Network

8. SYNTH: Synthesis

The project's approximate completion time distribution can be determined by combining the approximate subnetwork duration distributions. When the project network is decomposed in Step 3, the result is a network of subnetworks with any two connected subnetworks being either in series or in parallel. Let SUB_1, and SUB_2 be any two such subnetworks, and let the corresponding approximate subnetwork duration distributions be $\hat{F}_1$ and $\hat{F}_2$. If SUB_1 and SUB_2 are in series, then the approximate duration distribution for SUB_1 and SUB_2 combined is

$$\hat{F}(t) = \sum_{s \leq t} \hat{F}_2(t - s)\hat{f}_1(s) = \sum_{s \leq t} \hat{F}_1(t - s)\hat{f}_2(s) \tag{8.1}$$

where $\hat{f}_1$ and $\hat{f}_2$ are the discrete probability density functions corresponding to $\hat{F}_1$ and $\hat{F}_2$ respectively. If SUB_1 and SUB_2 are in parallel, then the approximate duration distribution for SUB_1 and SUB_2 combined is

$$\hat{F}(t) = \hat{F}_1(t) \cdot \hat{F}_2(t). \tag{8.2}$$

By repeatedly combining subnetworks that are connected either in series or in parallel, the project's approximate completion time distribtuion is obtained.

Once the project's approximate completion time distribution has been determined, the project's approximate mean completion time, $\hat{T}$, can be calculated and compared with the project deadline. If the

project manager feels that $\hat{T}$ is sufficiently close to the project dealine, say within 5%, then the project schedule just determined in Step 1 is considered "optimal". Otherwise, a new project schedule must be determined by returning to Step 1 with a new TARGET TIME.

Step 1 only requires that the project would be completed by TARGET TIME if each activity's duration was exactly its mean duration. On the other hand, $\hat{T}$ takes into consideration the random nature of an activity's duration and hence will generally exceed TARGET TIME. The difficulty is in deciding how much less than the project deadline should TARGE TIME be in order that the corresponding $\hat{T}$ be sufficiently close to the project deadline. The algorithm iteratively updates its estimate of this TARGET TIME by

New TARGET TIME = Old TARGET TIME*(Project Deadline/$\hat{T}$).

The initial TARGET TIME would usually be the project deadline but could be chosen somewhat less than the project deadline. A typical sequence of TARGET TIMEs and $\hat{T}$'s for a project deadline of 400 might be

TARGET TIME = 400, $\hat{T}$ = 500,

TARGET TIME = 320, $\hat{T}$ = 360,

TARGET TIME = 356, $\hat{T}$ = 408.

Since the algorithm approximates the project's entire completion time distribution for each tentative schedule determined in Step 1,

the quantity $\hat{T}$ in the above discussion could just as easily be a specified percentile of the project completion time distribution. For example, the project manager might wish a minimum cost project schedule such that the probability of the project being completed before the project deadline is .90. In this case, the new TARGET TIMEs would be determined with $\hat{T}$ being the 90th percentile of the project's approximate completion time distribution instead of the project's approximate mean completion time.

Of course, each time Steps 1-5 are performed a new $\hat{T}$, approximate project completion time distribtion, and project cost are generated. This sequence of $\hat{T}$'s and project costs supplements the deterministic project cost curve in desribing the impact of the project deadline on the project cost.

Appendix F

Structure and Contents of the Temporary Data Sets

The project scheduling system requires eight temporary data sets which it uses to pass information from one program to the next. In the documentation, these data sets are called FILE0, FILE1, FILE2, ..., FILE7. The particular physical devices (tape, disk, etc.) which the file names represent are determined at execution time by the program's associated JCL-statements (Job Control Language statements) and by the user-defined unit numbers. The unit numbers corresponding to the eight data sets are defined symbolically in the FORTRAN code as F0, F1, F2, ..., F7, respectively. The particular values of the variables F0, ..., F7 are determined by DATA-statements which are inserted at the beginning of each program (see Appendix G for an example). Currently, F0 = 8, F1 = 9, F2 = 10, F3 = 11, F4 = 12, F5 = 13, F6 = 14, and F7 = 15. The particular form of the JCL-statements determine whether the unit numbers reference, tape, disk, or drum. The exact form of the JCL is highly dependent on the particular computer installation at which the system is being run. An example of the JCL used at Texas A&M is given in Appendix G.

The structure and contents of each of the data sets is outlined in the subsections that follow. In these subsections, variable names used in the input instructions (Appendix A) are not redefined. Any new variable names are defined when they are first used.

1. FILE0

Written by: MAIN(A,B), DPS(C); i.e., parts A and B of FILE0 are written by MAIN and part C is written by DPS

Modified by: none

Used by: DPS(A), DSR(A,C), SIMP(A,B), MODSIMP(A,B): i.e., DPS uses only part A of FILE0, etc.

A. Record 1: $(4 \times MMAX + 5) \times 4$ bytes

1. NACT; (Integer, I * 4)
2. NODES: (I * 4)
3. NSRCE; (I * 4)
4. NSINK; (I * 4)
5. LNODEN; (I * 4)
6. S(MMAC,4); (two-dimensional array of I * 4 integers)

For I = 1, ..., NACT,

S(I,1) = the number of the I-th activity input, NA(I)

S(I,2) = the I-th activity's origin node, NODE0(I)

S(I,3) = the I-th activity's terminal node, NODET(I)

S(I,4) = the number of times and costs for the I-th activity, NCT(I)

For I > NACT,

S(I,1) = S(I,2) = S(I,3) = S(I,4) ≡ 0

Currently, MMAX = maximum number of activites in the project network = 1000.

B. Record 1: 4 bytes

1. JMAT; (I * 4)

C. Record 1: (NBREAK + 1) × 2 bytes

1. NBREAK; (Half-word integer, I * 2)

NBREAK = the number of times and costs needed to specify the entire project's time-cost curve

2. BREAK(NBREAK); (a one-dimensional array of I * 2 integers)

For I = 1, ..., NBREAK,

BREAK(I) = the I-th largest time in the entire project's time-cost curve

Currently, NBREAK $\leq$ 3000.

2. FILE1

Written by: MAIN(A)

Modified by: none

Used by: DPS(A), DSR(A)

A. Record 1: 8 bytes

1. TEST1; (I * 2)

2. TEST2; (I * 2)

3. TEST3; (I * 2)

TEST3 = a computation option in DPS that is defined internally by MAIN ≡ 0

4. STIME; (I * 2)

STIME = a parameter associated with TEST3 which is also defined internally by MAIN ≡ 0

* *

The remaining portion of FILE1 consists of one "block" of records for <u>each</u> original activity.

For I = 1, ..., NACT

Record (I + 1) . 1: (2 × NCT(I)) × 2 bytes

1. TIME(1); (I * 2)

2. TIME(2); (I * 2)

.
.
.

NCT(I). TIME(NCT(I)); (I * 2)

NCT(I) + 1. COST(1); (I * 2)

NCT(I) + 2. COST(2); (I * 2)

.
.
.

NCT(I) × 2. COST(NCT(I)); (I * 2)

For J = 1, ..., NCT(I) - 1

Record (I + 1). (J + 1); 20 bytes

1. IDIST; (I * 4)
2. PARM1; (Real, R * 4)
3. PARM2; (R * 4)
4. PARM3; (R * 4)
5. PARM4; (R * 4)

IDIST, PARM1, PARM2, PARM3, and PARM4 define the distribution (and its parameters) of the I-th activity on the J-th segment of the time-cost curve

* * * *

3. FILE2

Written by: DSR(A)

Modified by: DSR(A)

Used by: MODSIMP(A)

A. One record for each original activity

For I = 1, ..., NACT,

Record I: 20 bytes

1. JDIST; (I * 4)

JDIST = the distributional type of the I-th activity for the current TARGET TIME, TT

2. TPARM1; (R * 4)

3. TPARM2; (R * 4)

4. TPARM3; (R * 4)

5. TPARM4; (R * 4)

TPARM1, TPARM2, TPARM3, and TPARM4 are the I-th activity's distributional parameters for the current TARGET TIME, TT

4. FILE3

Written by: MAIN(A), SIMP(B)

Modified by: DECOMP(C)

Used by: DECOMP(B), SUBNET(A,C)

A. Record 1: 44 bytes

1. IFDF; (I * 4)
2. NMAX; (I * 4)
3. IPOOL; (I * 4)
4. SAMSIZ; (I * 4)
5. THELAM; (I * 4)
6. THETA(1); (R * 4)
7. THETA(2); (R * 4)
8. THETA(3); (R * 4)
9. LAMBDA(1); (R * 4)
10. LAMBDA(2); (R * 4)
11. LAMBDA(3); (R * 4)

B. Record 1: $(4 \times MMAX + 5) \times 4$ bytes

This record is an exact duplicate of FILE0 - part A except that the information refers to the simplified project network rather than the original project network. This record is <u>destroyed</u> by the execution of DECOMP and is replaced by part C.

C. Two records for <u>each</u> subnetwork identified by DECOMP

For K = 1, ..., number of subnetworks,

Record K.1: 4 bytes

1. M = the number of activities in the K-th subnetwork

Record K.2: $(3 \times M + 2) \times 4$ bytes

1. NSUB; (I * 4)

   NSUB = the number of the K-th subnetwork

2. NMM; (I * 4)

   NMM = the number of nodes in the K-th subnetwork

3. NET(3,M); (a two-dimensional array of I * 4 integers)

   For I = 1, ..., M,

   NET(1,I) = the origin node of the I-th activity in the K-th subnetwork

   NET(2,I) = the terminal node of the I-th activity in the K-th subnetwork

   NET(3,I) = the number assigned in the simplified project network to the K-th subnetwork's I-th activity

5. FILE4

Written by: MAIN(A), MODSIMP(B)

Modified by: MODSIMP(B), SYNTH(A), READFIL(A)

Used by: DSR(A), SUBNET(B), SYNTH(A)

A. Record 1: 12 bytes

1. NCYC; (I * 4)

NCYC = the number of the current iteration

2. TT; (R * 4)

TT = TARGET TIME for the current iteration

3. NFLAG; (I * 4)

NFLAG = internally defined parameter used by DSR ≡ 0 or 1

B. Record 1: 4 bytes

1. JNACT; (I * 4)

JNACT = the number of activities in the simplified project network

For I = 1, ..., JNACT,

Record I + 1: 96 bytes

1. CDFJ(12);(a one-dimenstional array of double word reals, R * 8)

CDFJ(1), ..., CDFJ(12) = $F^{-1}(.0)$, $F^{-1}(.05)$, $F^{-1}(.15)$, $F^{-1}(.25)$, ..., $F^{-1}(.95)$, $F^{-1}(1.00)$ where F is the cumulative distribution function for the I-th activity in the simplified project network.

6. FILE5

Written by: MAIN(A), DECOMP(B), SUBNET(C)

Modified by: SUBNET(C)

Used by: SYNTH(A,B,C)

A. Record 1: 16 bytes

1. IOPT; (I * 4)
2. NT; (I * 4)
3. PCT; (R * 4)
4. PD; (R * 4)

B. Record 1: 4 bytes

1. NOINS; (I * 4)

NOINS = the number of instructions generated by DECOMP for use in the synthesis process

Record 2; (27 × MAXINS) × 4 bytes

1. INSNO(MAXINS); (a one-dimension array of I * 4 integers)
2. ISUBNT(MAXINS); (a one-dimension array of I * 4 integers)
3. ISORP(MAXINS); (a one-dimension array of I * 4 integers)
4. JSUBNT(MAXINS, 24); (a two-dimensional array of I * 4 integers)

For I = 1, ..., NOINS,

INSNO(I) = the number of the I-th instruction

ISUBNT(I) = the number of the subnetwork resulting from the I-th instruction

ISORP(I) = indicates whether the subnetworks involved in the I-th instruction are in parallel or series

ISORP(I) = 0 implies the subnetworks are in series

ISORP(I) = 1 implies the subnetworks are in parallel

For J = 1, ..., 24,

JSUBNT(I,J) = the J-th subnetwork associated with the I-th instruction

C. One record for each subnetwork identified by DECOMP

For K = 1, ..., number of subnetworks,

Record K: (2 × JEDF + 2) × 4 bytes

1. ID; (I * 4)

   ID = the number of the K-th subnetwork

2. JEDF; (I * 4)

   JEDF = the number of subdivisions in the distribution function for the K-th subnetwork

3. XFD(JEDF,2); (a two-dimensional array of R * 4 reals)

   For J = 1, ..., JEDF,

   XFD(J,1) = the abscissa for the J-th tabled value of the cumulative distribution function for the K-th subnetwork

   XFD(J,2) = the ordinate for the J-th tabled value of the cumulative distribution function for the K-th subnetwork

7. FILE6

Written by: DPS(A)

Modified by: none

Used by: DSR(A)

A. One record for <u>each</u> segment of the entire project's time-cost curve

For K = 1, ..., NBREAK,

Record K: 4 × NACT bytes

1. XD(NACT,2); (a two-dimensional array of I * 2 integers)

For I = 1, ..., NACT

XD(I,1) = the direction (sign) of the change of the I-th activity's duration on the K-th function segment, i.e. +1, 0, -1

XD(I,2) = the maximum duration of the I-th activity on the K-th function segment

8. FILE7

Written by: SIMP(A)

Modified by: none

Used by: MODSIMP(A)

A. One record for each activity simplification operation identified by SIMP

For K = 1, ..., number of simplification operations,

Record K: 96 bytes

1. LIST(24); (a one-dimensional array of I * 4 integers)

For J = 1, ..., 24,

LIST(J) = a parameter which indentifies either the configuration or an activity involved in the K-th simplification operation

Table F.1

Contents of the Temporary Files at the Completion of Each Program

Program first iteration	FILE0	FILE1	FILE2	FILE3	FILE4	FILE5	FILE6	FILE7
MAIN	A,B	A	X	A	A	A	X	X
DPS	A,B,C	A	X	A	A	A	A	X
DSR	A,B,C	A	A	A	A	A	A	X
SIMP	A,B,C	A	A	A,B	A	A	A	A
MODSIMP	A,B,C	A	X	A,B	A,B	A	A	A
DECOMP	A,B,C	A	X	A,C	A,B	A,B	A	A
SUBNET	A,B,C	A	X	A,C	A	A,B,C	A	A
SYNTH	A,B,C	A	X	A,C	A	A,B	A	A
second & successive iterations								
DSR	A,B,C	A	A	A,C	A	A,B	A	A
MODSIMP	A,B,C	A	X	A,C	A,B	A,B	A	A
SUBNET	A,B,C	A	X	A,C	A	A,B,C	A	A
SYNTH	A,B,C	A	X	A,C	A	A,B	A	A

X means the file has no contents or that the information it does contain is never used in a subsequent step

Table F.2

Files Referenced by Each Program

Temporary Files*

PROGRAM	FILE0	FILE1	FILE2	FILE3	FILE4	FILE5	FILE6	FILE7
MAIN	X	X	X	X	X	X	X	X
DPS	X	X					X	
DSR	X	X	X		X		X	
SIMP	X			X				X
MODSIMP	X		X		X			X
DECOMP				X		X		
SUBNET				X	X	X		
SYNTH					X	X		
SAVEFIL	X	X	X	X	X	X	X	X
READFIL	X	X	X	X	X	X	X	X

* The temporary data sets referenced by each program in the computer system are indicated by an "X" in the appropriate column. Every data set referenced by a particular program should be defined for the job-step in which that program is executed.

Appendix C

Job Control Language for Texas A&M University Computer Facilities

1. Execution using FORTRAN source decks

A listing of the JCL used to generate the sample output of Appendix C is given below. During the first iteration each program in the sequence is compiled using the IBM FORTRAN G Compile and the load module which ultimately results is stored on the partitioned data set named USER.STAT.SIELKEN.PERT. Thus, on subsequent iterations the compile and link-edit steps are skipped.

```
//JOBLIB DD DSNAME=USER.STAT.SIELKEN.PERT,DISP=SHR
//MAIN EXEC FORTGCL,REGION=256K
//FORT.SYSIN DD *

        INSERT FORTRAN COURCE DECK FOR  MAIN  HERE

//LKED.SYSLMOD DD DSN=USER.STAT.SIELKEN.PERT(MAIN),DISP=SHR
//MAIN EXEC PGM=MAIN,REGION=320K,COND=(4,LT)
//FT06F001 DD SYSOUT=A
//FT08F001     DD UNIT=SYSDA,DSN=&&FILE0,DISP=(NEW,PASS),
//        SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT09F001     DD UNIT=SYSDA,DSN=&&FILE1,DISP=(NEW,PASS),
//        SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT10F001     DD UNIT=SYSDA,DSN=&&FILE2,DISP=(NEW,PASS),
//        SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT11F001     DD UNIT=SYSDA,DSN=&&FILE3,DISP=(NEW,PASS),
//        SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT12F001     DD UNIT=SYSDA,DSN=&&FILE4,DISP=(NEW,PASS),
//        SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT13F001     DD UNIT=SYSDA,DSN=&&FILE5,DISP=(NEW,PASS),
//        SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT14F001     DD UNIT=SYSDA,DSN=&&FILE6,DISP=(NEW,PASS),
//        SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT15F001     DD UNIT=SYSDA,DSN=&&FILE7,DISP=(NEW,PASS),
//        SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT05F001 DD *

        INSERT DATA CARDS FOR  MAIN  HERE

//DPS EXEC FORTGCL,REGION=384K
//FORT.SYSIN DD *

        INSERT FORTRAN COURCE DECK FOR  DPS  HERE
```

```
//LKED.SYSLMOD DD DSN=USER.STAT.SIELKEN.PERT(DPS),DISP=SHR
//DPS EXEC PGM=DPS,REGION=512K,COND=(4,LT)
//FT06F001 DD SYSOUT=A
//FT08F001     DD UNIT=SYSDA,DSN=&&FILE0,DISP=(OLD,PASS)
//FT09F001     DD UNIT=SYSDA,DSN=&&FILE1,DISP=(OLD,PASS)
//FT14F001     DD UNIT=SYSDA,DSN=&&FILE6,DISP=(OLD,PASS)
//DSR EXEC FORTGCL,REGION=256K
//FORT.SYSIN DD *

        INSERT FORTRAN COURCE DECK FOR  DSR  HERE

//LKED.SYSLMOD DD DSN=USER.STAT.SIELKEN.PERT(DSR),DISP=SHR
//DSR EXEC PGM=DSR,REGION=256K,COND=(4,LT)
//FT06F001 DD SYSOUT=A
//FT08F001     DD UNIT=SYSDA,DSN=&&FILE0,DISP=(OLD,PASS)
//FT09F001     DD UNIT=SYSDA,DSN=&&FILE1,DISP=(OLD,PASS)
//FT10F001     DD UNIT=SYSDA,DSN=&&FILE2,DISP=(OLD,PASS)
//FT12F001     DD UNIT=SYSDA,DSN=&&FILE4,DISP=(OLD,PASS)
//FT14F001     DD UNIT=SYSDA,DSN=&&FILE6,DISP=(OLD,PASS)
//SIMP EXEC FORTGCL,REGION=320K
//FORT.SYSIN DD *

        INSERT FORTRAN COURCE DECK FOR  SIMP  HERE

//LKED.SYSLMOD DD DSN=USER.STAT.SIELKEN.PERT(SIMP),DISP=SHR
//SIMP EXEC PGM=SIMP,REGION=256K,COND=(4,LT)
//FT06F001 DD SYSOUT=A
//FT08F001     DD UNIT=SYSDA,DSN=&&FILE0,DISP=(OLD,PASS)
//FT11F001     DD UNIT=SYSDA,DSN=&&FILE3,DISP=(OLD,PASS)
//FT15F001     DD UNIT=SYSDA,DSN=&&FILE7,DISP=(OLD,PASS)
//MODSIMP EXEC FORTGCL,REGION=256K
//FORT.SYSIN DD *

        INSERT FORTRAN COURCE DECK FOR  MODSIMP  HERE

//LKED.SYSLMOD DD DSN=USER.STAT.SIELKEN.PERT(MODSIMP),DISP=SHR
//MODSIMP EXEC PGM=MODSIMP,REGION=320K,COND=(4,LT)
//FT06F001 DD SYSOUT=A
//FT08F001     DD UNIT=SYSDA,DSN=&&FILE0,DISP=(OLD,PASS)
//FT10F001     DD UNIT=SYSDA,DSN=&&FILE2,DISP=(OLD,PASS)
//FT12F001     DD UNIT=SYSDA,DSN=&&FILE4,DISP=(OLD,PASS)
//FT15F001     DD UNIT=SYSDA,DSN=&&FILE7,DISP=(OLD,PASS)
//DECOMP EXEC FORTGCL,REGION=256K
//FORT.SYSIN DD *

        INSERT FORTRAN COURCE DECK FOR  DECOMP  HERE

//LKED.SYSLMOD DD DSN=USER.STAT.SIELKEN.PERT(DECOMP),DISP=SHR
//DECOMP EXEC PGM=DECOMP,REGION=320K,COND=(4,LT)
//FT06F001 DD SYSOUT=A
//FT11F001     DD UNIT=SYSDA,DSN=&&FILE3,DISP=(OLD,PASS)
//FT13F001     DD UNIT=SYSDA,DSN=&&FILE5,DISP=(OLD,PASS)
//SUBNET EXEC FORTGCL,REGION=256K
//FORT.SYSIN DD *

        INSERT FORTRAN COURCE DECK FOR  SUBNET  HERE

//LKED.SYSLMOD DD DSN=USER.STAT.SIELKEN.PERT(SUBNET),DISP=SHR
//SUBNET EXEC PGM=SUBNET,REGION=448K,COND=(4,LT)
//FT06F001 DD SYSOUT=A
```

```
//FT11F001     DD UNIT=SYSDA,DSN=&&FILE3,DISP=(OLD,PASS)
//FT12F001     DD UNIT=SYSDA,DSN=&&FILE4,DISP=(OLD,PASS)
//FT13F001     DD UNIT=SYSDA,DSN=&&FILE5,DISP=(OLD,PASS)
//SYNTH EXEC FORTGCL,REGION=256K
//FORT.SYSIN DD *

          INSERT FORTRAN COURCE DECK FOR  SYNTH  HERE

//LKED.SYSLMOD DD DSN=USER.STAT.SIELKEN.PERT(SYNTH),DISP=SHR
//SYNTH EXEC PGM=SYNTH,REGION=256K,COND=(4,LT)
//FT06F001 DD SYSOUT=A
//FT12F001     DD UNIT=SYSDA,DSN=&&FILE4,DISP=(OLD,PASS)
//FT13F001     DD UNIT=SYSDA,DSN=&&FILE5,DISP=(OLD,PASS)
//DSR EXEC PGM=DSR,REGION=256K,COND=(4,LT)
//FT06F001 DD SYSOUT=A
//FT08F001     DD UNIT=SYSDA,DSN=&&FILE0,DISP=(OLD,PASS)
//FT09F001     DD UNIT=SYSDA,DSN=&&FILE1,DISP=(OLD,PASS)
//FT10F001     DD UNIT=SYSDA,DSN=&&FILE2,DISP=(OLD,PASS)
//FT12F001     DD UNIT=SYSDA,DSN=&&FILE4,DISP=(OLD,PASS)
//FT14F001     DD UNIT=SYSDA,DSN=&&FILE6,DISP=(OLD,PASS)
//MODSIMP EXEC PGM=MODSIMP,REGION=320K,COND=(4,LT)
//FT06F001 DD SYSOUT=A
//FT08F001     DD UNIT=SYSDA,DSN=&&FILE0,DISP=(OLD,PASS)
//FT10F001     DD UNIT=SYSDA,DSN=&&FILE2,DISP=(OLD,PASS)
//FT12F001     DD UNIT=SYSDA,DSN=&&FILE4,DISP=(OLD,PASS)
//FT15F001     DD UNIT=SYSDA,DSN=&&FILE7,DISP=(OLD,PASS)
//SUBNET EXEC PGM=SUBNET,REGION=448K,COND=(4,LT)
//FT06F001 DD SYSOUT=A
//FT11F001     DD UNIT=SYSDA,DSN=&&FILE3,DISP=(OLD,PASS)
//FT12F001     DD UNIT=SYSDA,DSN=&&FILE4,DISP=(OLD,PASS)
//FT13F001     DD UNIT=SYSDA,DSN=&&FILE5,DISP=(OLD,PASS)
//SYNTH EXEC PGM=SYNTH,REGION=256K,COND=(4,LT)
//FT06F001 DD SYSOUT=A
//FT12F001     DD UNIT=SYSDA,DSN=&&FILE4,DISP=(OLD,PASS)
//FT13F001     DD UNIT=SYSDA,DSN=&&FILE5,DISP=(OLD,PASS)
/*END
```

2. Execution using load modules

If the load modules corresponding to the programs in the system have already been prepared and stored on the disk file USER.STAT.SIELKEN.PERT., the following JCL is appropriate.

```
//JOBLIB DD DSNAME=USER.STAT.SIELKEN.PERT,DISP=SHR
//MAIN EXEC PGM=MAIN,REGION=320K,COND=(4,LT)
//FT06F001 DD SYSOUT=A
//FT08F001     DD UNIT=SYSDA,DSN=&&FILE0,DISP=(NEW,PASS),
//        SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT09F001     DD UNIT=SYSDA,DSN=&&FILE1,DISP=(NEW,PASS),
//        SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT10F001     DD UNIT=SYSDA,DSN=&&FILE2,DISP=(NEW,PASS),
//        SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT11F001     DD UNIT=SYSDA,DSN=&&FILE3,DISP=(NEW,PASS),
//        SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT12F001     DD UNIT=SYSDA,DSN=&&FILE4,DISP=(NEW,PASS),
//        SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT13F001     DD UNIT=SYSDA,DSN=&&FILE5,DISP=(NEW,PASS),
//        SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT14F001     DD UNIT=SYSDA,DSN=&&FILE6,DISP=(NEW,PASS),
//        SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT15F001     DD UNIT=SYSDA,DSN=&&FILE7,DISP=(NEW,PASS),
//        SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT05F001 DD *

        INSERT DATA CARDS FOR  MAIN  HERE

//DPS EXEC PGM=DPS,REGION=512K,COND=(4,LT)
//FT06F001 DD SYSOUT=A
//FT08F001     DD UNIT=SYSDA,DSN=&&FILE0,DISP=(OLD,PASS)
//FT09F001     DD UNIT=SYSDA,DSN=&&FILE1,DISP=(OLD,PASS)
//FT14F001     DD UNIT=SYSDA,DSN=&&FILE6,DISP=(OLD,PASS)
//DSR EXEC PGM=DSR,REGION=256K,COND=(4,LT)
//FT06F001 DD SYSOUT=A
//FT08F001     DD UNIT=SYSDA,DSN=&&FILE0,DISP=(OLD,PASS)
//FT09F001     DD UNIT=SYSDA,DSN=&&FILE1,DISP=(OLD,PASS)
//FT10F001     DD UNIT=SYSDA,DSN=&&FILE2,DISP=(OLD,PASS)
//FT12F001     DD UNIT=SYSDA,DSN=&&FILE4,DISP=(OLD,PASS)
//FT14F001     DD UNIT=SYSDA,DSN=&&FILE6,DISP=(OLD,PASS)
//SIMP EXEC PGM=SIMP,REGION=256K,COND=(4,LT)
//FT06F001 DD SYSOUT=A
//FT08F001     DD UNIT=SYSDA,DSN=&&FILE0,DISP=(OLD,PASS)
//FT11F001     DD UNIT=SYSDA,DSN=&&FILE3,DISP=(OLD,PASS)
//FT15F001     DD UNIT=SYSDA,DSN=&&FILE7,DISP=(OLD,PASS)
//MODSIMP EXEC PGM=MODSIMP,REGION=320K,COND=(4,LT)
//FT06F001 DD SYSOUT=A
//FT08F001     DD UNIT=SYSDA,DSN=&&FILE0,DISP=(OLD,PASS)
//FT10F001     DD UNIT=SYSDA,DSN=&&FILE2,DISP=(OLD,PASS)
//FT12F001     DD UNIT=SYSDA,DSN=&&FILE4,DISP=(OLD,PASS)
//FT15F001     DD UNIT=SYSDA,DSN=&&FILE7,DISP=(OLD,PASS)
```

```
//DECOMP EXEC PGM=DECOMP,REGION=320K,COND=(4,LT)
//FT06F001 DD SYSOUT=A
//FT11F001    DD UNIT=SYSDA,DSN=&&FILE3,DISP=(OLD,PASS)
//FT13F001    DD UNIT=SYSDA,DSN=&&FILE5,DISP=(OLD,PASS)
//SUBNET EXEC PGM=SUBNET,REGION=448K,COND=(4,LT)
//FT06F001 DD SYSOUT=A
//FT11F001    DD UNIT=SYSDA,DSN=&&FILE3,DISP=(OLD,PASS)
//FT12F001    DD UNIT=SYSDA,DSN=&&FILE4,DISP=(OLD,PASS)
//FT13F001    DD UNIT=SYSDA,DSN=&&FILE5,DISP=(OLD,PASS)
//SYNTH EXEC PGM=SYNTH,REGION=256K,COND=(4,LT)
//FT06F001 DD SYSOUT=A
//FT12F001    DD UNIT=SYSDA,DSN=&&FILE4,DISP=(OLD,PASS)
//FT13F001    DD UNIT=SYSDA,DSN=&&FILE5,DISP=(OLD,PASS)
//DSR EXEC PGM=DSR,REGION=256K,COND=(4,LT)
//FT06F001 DD SYSOUT=A
//FT08F001    DD UNIT=SYSDA,DSN=&&FILE0,DISP=(OLD,PASS)
//FT09F001    DD UNIT=SYSDA,DSN=&&FILE1,DISP=(OLD,PASS)
//FT10F001    DD UNIT=SYSDA,DSN=&&FILE2,DISP=(OLD,PASS)
//FT12F001    DD UNIT=SYSDA,DSN=&&FILE4,DISP=(OLD,PASS)
//FT14F001    DD UNIT=SYSDA,DSN=&&FILE6,DISP=(OLD,PASS)
//MODSIMP EXEC PGM=MODSIMP,REGION=320K,COND=(4,LT)
//FT06F001 DD SYSOUT=A
//FT08F001    DD UNIT=SYSDA,DSN=&&FILE0,DISP=(OLD,PASS)
//FT10F001    DD UNIT=SYSDA,DSN=&&FILE2,DISP=(OLD,PASS)
//FT12F001    DD UNIT=SYSDA,DSN=&&FILE4,DISP=(OLD,PASS)
//FT15F001    DD UNIT=SYSDA,DSN=&&FILE7,DISP=(OLD,PASS)
//SUBNET EXEC PGM=SUBNET,REGION=448K,COND=(4,LT)
//FT06F001 DD SYSOUT=A
//FT11F001    DD UNIT=SYSDA,DSN=&&FILE3,DISP=(OLD,PASS)
//FT12F001    DD UNIT=SYSDA,DSN=&&FILE4,DISP=(OLD,PASS)
//FT13F001    DD UNIT=SYSDA,DSN=&&FILE5,DISP=(OLD,PASS)
//SYNTH EXEC PGM=SYNTH,REGION=256K,COND=(4,LT)
//FT06F001 DD SYSOUT=A
//FT12F001    DD UNIT=SYSDA,DSN=&&FILE4,DISP=(OLD,PASS)
//FT13F001    DD UNIT=SYSDA,DSN=&&FILE5,DISP=(OLD,PASS)
/*END
```

Appendix H

Interrupting the Computer System

The number of iterations necessary to identify an "acceptable" project schedule using the new project scheduling procedure depends on many factors and usually cannot be determined in advance. Also the number of job steps allowed in the JCL-program for a given run varies widely from one computer installation to the next. (Currently, at Texas A&M the limitation is 10 job steps.) Thus, some provision must be made for interrupting the computer system prior to the identification of an acceptable project schedule and restarting it from the point of interruption.

One method is to let the "temporary" data sets defined in Appendix F correspond to user-supplied (permanent) storage devices rather than system-supplied (temporary) ones. This requires, however, that the user supply (and pay for)eight different permanent storage devices (or locations).

A second method which requires that the user supply only one permanent storage device involves the use of the programs SAVEFIL and READFIL which allow the user the capability of 1) interrupting the computer system at any of the starred points (*) indicated on Figure H.1 and 2) continuing the procedure from the point of interruption at some future time. SAVEFIL and READFIL are documented in subsections 1 and 2, respectively, of this appendix. Subsection 3 gives a listing of their associated JCL.

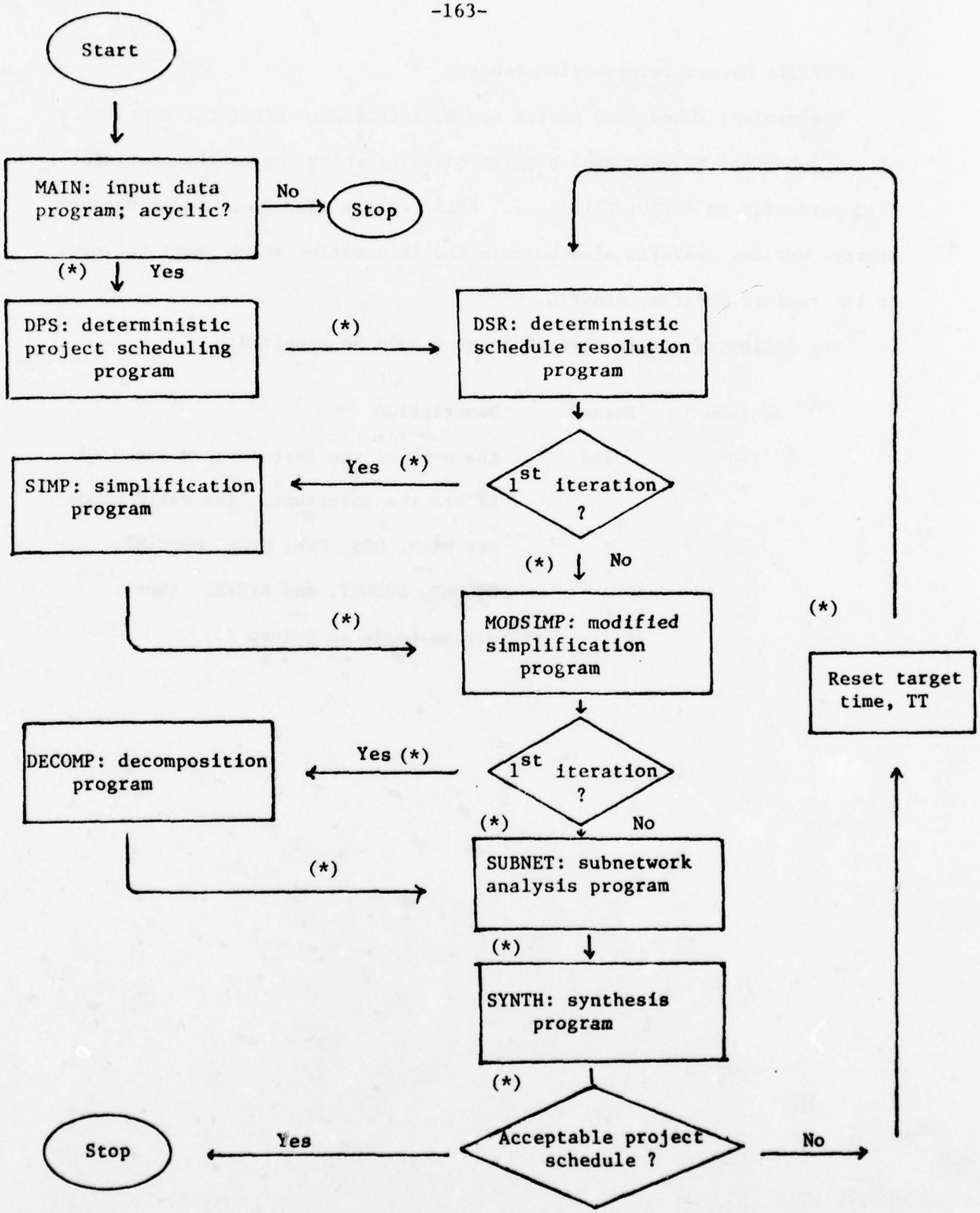


Figure H.1 Flowchart with (*) indicating where SAVEFIL may be used to interrupt the system.

1. SAVEFIL: System Interruption Program

The project scheduling system may be interrupted after the completion of any job step by executing program SAVEFIL, which copies the pertinent data currently on FILE0, FILE1, ..., FILE7 onto a user-supplied permanent storage device. SAVEFIL also outputs the information which must be input to the restart program, READFIL.

The following single card of input should be supplied:

Column	Format	Description
1-8	2A4	the name of the last program executed before the interrupt. The valid names are MAIN, DPS, DSR, SIMP, MODSIMP, DECOMP, SUBNET, and SYNTH. (Note: always begin in Column 1.)

2. READFIL: System Restart Program

When the project scheduling system has been interrupted by the use of SAVEFIL, READFIL must be used to restore the necessary information to the temporary data sets before the iterative procedure may resume. The following single card of input should be supplied:

Column	Format	Description
1-8	2A4	the name of the last program executed before the interrupt.
1-9	I2	0 if the procedure was interrupted before the first execution of SYNTH 1 otherwise

The remaining parameters should only be supplied if the procedure was interrupted immediately after the execution of SYNTH.

11-20	F10.5	The value of TT for the next iteration if different from the value specified by SYNTH.
21-25	I5	1 if the values of any of the following are to be different from their current values in the next iteration: IOPT, PCT, PD 0 otherwise
26-30	I5	the new value for IOPT
31-40	F10.5	the new value for PCT
41-50	F10.5	the new value for PD

See Appendix A for the definition of TT, IOPT, PCT, and PD.

3. JCL for SAVEFIL and READFIL

The following exemplifies the JCL required by program SAVEFIL for the Texas A&M computer facilities. For this example, the user-supplied permanent storage device is named USER.STAT.SIELKEN.DATA.

```
//JOBLIB DD DSNAME=USER.STAT.SIELKEN.PERT,DISP=SHR
//SAVEFIL EXEC FORTGCL,REGION=256K
//FORT.SYSIN DD *

        INSERT FORTRAN SOURCE DECK FOR  SAVEFIL  HERE

//LKED.SYSLMOD DD DSN=USER.STAT.SIELKEN.PERT(SAVEFIL),DISP=SHR
//SAVEFIL EXEC PGM=SAVEFIL,REGION=256K,COND=(4,LT)
//FT04F001     DD DSN=USER.STAT.SIELKEN.DATA,DISP=SHR
//FT06F001 DD SYSOUT=A
//FT08F001    DD UNIT=SYSDA,DSN=&&FILE0,DISP=(OLD,DELETE)
//FT09F001     DD UNIT=SYSDA,DSN=&&FILE1,DISP=(OLD,DELETE)
//FT10F001     DD UNIT=SYSDA,DSN=&&FILE2,DISP=(OLD,DELETE)
//FT11F001     DD UNIT=SYSDA,DSN=&&FILE3,DISP=(OLD,DELETE)
//FT12F001     DD UNIT=SYSDA,DSN=&&FILE4,DISP=(OLD,DELETE)
//FT13F001     DD UNIT=SYSDA,DSN=&&FILE5,DISP=(OLD,DELETE)
//FT14F001     DD UNIT=SYSDA,DSN=&&FILE6,DISP=(OLD,DELETE)
//FT15F001     DD UNIT=SYSDA,DSN=&&FILE7,DISP=(OLD,DELETE)
//FT05F001 DD *

        INSERT DATA CARD FOR  SAVEFIL  HERE

/*END
```

Similarly, the JCL for program READFIL is as follows:

```
//JOBLIB DD DSNAME=USER.STAT.SIELKEN.PERT,DISP=SHR
//READFIL EXEC FORTGCL,REGION=256K
//FORT.SYSIN DD *

          INSERT FORTRAN SOURCE DECK FOR  READFIL  HERE

//LKED.SYSLMOD DD DSN=USER.STAT.SIELKEN.PERT(READFIL),DISP=SHR
//READFIL EXEC PGM=READFIL,REGION=256K,COND=(4,LT)
//FT04F001     DD DSN=USER.STAT.SIELKEN.DATA,DISP=SHR
//FT06F001 DD SYSOUT=A
//FT08F001     DD UNIT=SYSDA,DSN=&&FILE0,DISP=(NEW,PASS),
//         SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT09F001     DD UNIT=SYSDA,DSN=&&FILE1,DISP=(NEW,PASS),
//         SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT10F001     DD UNIT=SYSDA,DSN=&&FILE2,DISP=(NEW,PASS),
//         SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT11F001     DD UNIT=SYSDA,DSN=&&FILE3,DISP=(NEW,PASS),
//         SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT12F001     DD UNIT=SYSDA,DSN=&&FILE4,DISP=(NEW,PASS),
//         SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT13F001     DD UNIT=SYSDA,DSN=&&FILE5,DISP=(NEW,PASS),
//         SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT14F001     DD UNIT=SYSDA,DSN=&&FILE6,DISP=(NEW,PASS),
//         SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT15F001     DD UNIT=SYSDA,DSN=&&FILE7,DISP=(NEW,PASS),
//         SPACE=(32000,(100,25)),DCB=(LRECL=3152,BLKSIZE=3156,RECFM=VBS)
//FT05F001 DD *

          INSERT DATA CARD FOR  READFIL  HERE

//-------CONTINUE WITH THE JCL FOR THE NEXT PROGRAM IN THE SEQUENCE
```

Appendix I

Program Listings

1. MAIN

```
C      MAIN PROGRAM                                                          1
C                                                                            2
C         THIS PROGRAM READS ALL USER SUPPLIED INPUT TO THE PROJECT          3
C         SCHEDULING SYSTEM AND STORES THE INFORMATION ON (SCRATCH)          4
C         DISK FILES DEFINED BY THE USER.  THE INFORMATION IS RETIEVED       5
C         AS NEEDED BY THE INDIVIDUAL PROGRAMS.                              6
C                                                                            7
C         SEE TECHNICAL REPORT #57 FOR SPECIFIC INPUT INSTRUCTIONS           8
C                                                                            9
C                                                                           10
C         FOR THE SAKE OF IDENTIFYING THE APPROPRIATE DIMENSIONS, LET       11
C         MMAX = THE MAXIMUM NUMBER OF ACTIVITIES IN THE ORIGINAL           12
C                PROJECT NETWORK                                            13
C         LNMAX = THE LARGEST NODE NUMBER ALLOWED IN THE PROJECT NETWORK     14
C         NCTMAX = THE MAXIMUM NUMBER OF COMPLETION TIMES AND COSTS FOR      15
C                EACH ACTIVITY                                              16
C         CURRENTLY, MMAX=1000; LNMAX=9999; NCTMAX=6                         17
C                                                                           18
C      INTEGER*4 FILE0(4*MMAX+5),NA(MMAX)                                   19
C      INTEGER*4 NODE0(MMAX),NODET(MMAX),NCT(MMAX),IDIST(MMAX,NCTMAX-1)     20
C      INTEGER*2 TIME(MMAX,NCTMAX),COST(MMAX,NCTMAX)                        21
C      REAL*4 PARM1(MMAX,NCTMAX-1),PARM2(MMAX,NCTMAX-1)                     22
C      REAL*4 PARM3(MMAX,NCTMAX-1),PARM4(MMAX,NCTMAX-1)                     23
C      EQUIVALENCE (FILE0(6),NA(1)),(FILE0(MMAX+6),NODE0(1))                24
C      EQUIVALENCE (FILE0(2*MMAX+6),NODET(1)),(FILE0(3*MMAX+6,NCT(1))       25
C         IN SUBROUTINE LOOP, THE ADDITIONAL ARRAY DIMENSIONS ARE:          26
C      DIMENSION A(LNMAX),B(LNMAX)                                          27
C                                                                           28
C         THE VALUES OF THE VARIABLES F0 - F7 SHOULD CORRESPOND TO UNIT     29
C         NUMBERS FOR THE INSTALLATION'S TEMPORARY STORAGE DEVISES(DISKS)   30
C         WHICH WILL BE REFERENCED BY THE PROJECT SCHEDULING SYSTEM.        31
C                                                                           32
      INTEGER F0/8/,F1/9/,        F3/11/,F4/12/,F5/13/                      33
      INTEGER*4 SAMSIZ,THELAM,FILE0(4005),NA(1000)                          34
      INTEGER*4 NODE0(1000),NODET(1000),NCT(1000),IDIST(1000,5)             35
      INTEGER*2 TIME(1000,6),COST(1000,6),TEST1,TEST2,TEST3,STIME           36
      REAL*4 PARM1(1000,5),PARM2(1000,5),PARM3(1000,5),PARM4(1000,5)        37
      REAL*4 THETA(3),LAMBDA(3)                                             38
      EQUIVALENCE (FILE0(1),NACT),(FILE0(2),NODES),(FILE0(3),NSRCE),        39
     *     (FILE0(4),NSINK),(FILE0(5),LNODEN),(FILE0(6),NA(1)),             40
     *     (FILE0(1006),NODE0(1)),(FILE0(2006),NODET(1)),                   41
     *     (FILE0(3006),NCT(1))                                             42
      COMMON /BLKA/NACT,NODES,NSRCE,NSINK,LNODEN,NA,NODE0,NODET,NCT         43
      INTEGER NORM(5)/'NORM','AL',3*' '/,BET(5)/'BETA',4*' '/               44
      INTEGER RECT(5)/'RECT','ANGU','LAR(','UNIF','ORM)'/                   45
      INTEGER FIXED(5)/'FIXE','D(CO','NSTA','NT)',' '/                      46
      WRITE(6,2000)                                                         47
2000  FORMAT(1H1,132('*')/1X,132('*')//////51X,'S T A T I S T I C A L      48
     *P E R T'/////)                                                        49
      WRITE(6,2001)                                                         50
2001  FORMAT(52X,'A PROJECT SCHEDULING PROCEDURE'///                        51
     *60X,'WRITTEN AT THE '//55X,'INSTITUTE OF STATISTICS'/                 52
     *57X,'TEXAS A&M UNIVERSITY'/53X,'COLLEGE STATION, TEXAS 77843'///     53
     *61X,'AUGUST 1978'/////33X,'INQUIRIES AND COMMENTS SHOULD BE ADDRES   54
     *SED TO: ROBERT L. SIELKEN JR.'//1X,132('*')/1X,132('*')///////)      55
C                                                                           56
C         GENERATE PARTS A AND B OF FILE0 AND FILE1                         57
C                                                                           58
      READ(5,110)NACT,NODES,NSRCE,NSINK,LNODEN                              59
      DO 1300 I=1,NACT                                                      60
```

```
      READ(5,110)NA(I),NODEO(I),NODET(I),NNCT                              61
      NCT(I)=NNCT                                                          62
      NNCT1=NNCT-1                                                         63
      READ(5,110)(TIME(I,J),J=1,NNCT)                                      64
      READ(5,110)(COST(I,J),J=1,NNCT)                                      65
      DO 1300 L=1,NNCT1                                                    66
      READ(5,150)J,IDIST(I,J),PARM1(I,J),PARM2(I,J),PARM3(I,J),PARM4(I,J   67
     *)                                                                    68
1300  CONTINUE                                                             69
      WRITE(F0)FILE0                                                       70
      READ(5,110)TEST1,TEST2,JMAT                                          71
      WRITE(F0)JMAT                                                        72
      ENDFILE F0                                                           73
      TEST3=0                                                              74
      STIME=0                                                              75
      WRITE(F1)TEST1,TEST2,TEST3,STIME                                     76
      DO 1400 I=1,NACT                                                     77
      NNCT=NCT(I)                                                          78
      NNCT1=NNCT-1                                                         79
      WRITE(F1)(TIME(I,J),J=1,NNCT),(COST(I,J),J=1,NNCT)                   80
      DO 1400 J=1,NNCT1                                                    81
      WRITE(F1)IDIST(I,J),PARM1(I,J),PARM2(I,J),PARM3(I,J),PARM4(I,J)      82
1400  CONTINUE                                                             83
      ENDFILE F1                                                           84
C                                                                          85
C        GENERATE PART A OF FILE3                                          86
C                                                                          87
      READ(5,110)IEDF,NMAX,IPOOL,SAMSIZ                                    88
      READ(5,112)THELAM,(THETA(I),I=1,3),(LAMBDA(I),I=1,3)                 89
      WRITE(F3)IEDF,NMAX,IPOOL,SAMSIZ,THELAM,(THETA(I),I=1,3),(LAMBDA(I)   90
     *,I=1,3)                                                              91
C                                                                          92
C        GENERATE PART A OF FILE4 AND PART A OF FILE5                      93
C                                                                          94
      READ(5,150)IOPT,NT,PCT,PD,TT                                         95
      NCYC=1                                                               96
      NFLAG=0                                                              97
      WRITE(F4)NCYC,TT,NFLAG                                               98
      WRITE(F5)IOPT,NT,PCT,PD                                              99
110   FORMAT(8I10)                                                        100
112   FORMAT(I10,6F10.5)                                                  101
150   FORMAT(2I10,4F10.5)                                                 102
C                                                                         103
C        OUTPUT THE PROBLEM AS IT WAS READ IN                             104
C                                                                         105
      WRITE(6,1000)                                                       106
1000  FORMAT('0THIS IS THE OUTPUT FROM THE MAIN PROGRAM: MAIN'////        107
     *'0THE PROJECT NETWORK AS IT WAS READ IN:')                          108
      WRITE(6,1001)NODES,NACT,NSRCE,NSINK,LNODEN                          109
1001  FORMAT(///'0THE NUMBER OF NODES IS ',I4,'.'/' THE NUMBER OF ACTIVI  110
     *TIES IS ',I4,'.'/' THE SOURCE NODE IS NUMBERED ',I5,' AND THE SINK  111
     * NODE IS NUMBERED ',I5,'.'/' THE LARGEST NODE NUMBER IS ',          112
     *I5,'.'//'1ACTIVITY ORIG TERM J TIME COST DISTRIBUTION',9X,'PARAMET  113
     *ERS'/)                                                              114
      DO 1100 I=1,NACT                                                    115
      NNCT=NCT(I)                                                         116
      NNCT1=NNCT-1                                                        117
      DO 1090 J=1,NNCT1                                                   118
      ID=IDIST(I,J)+2                                                     119
      GO TO (1020,1040,1060,1080),ID                                      120
1020  IF(PARM4(I,J).EQ.-1.)GO TO 1030                                     121
```

```
      WRITE(6,1021)I,NODEO(I),NODET(I),J,TIME(I,J),COST(I,J),
     *   BET,PARM1(I,J),PARM3(I,J),PARM2(I,J),PARM4(I,J)
 1021 FORMAT(3X,I4,2X,I5,    I5,1X,I1,I5,I5,1X,5A4,1X,
     *'MIN=',F13.5,', MAX=',F15.5,', ALPHA=',F12.5,', BETA=',F12.5)
      GO TO 1090
 1030 WRITE(6,1031)I,NODEO(I),NODET(I),J,TIME(I,J),COST(I,J),
     *BET,PARM1(I,J),PARM2(I,J),PARM3(I,J)
 1031 FORMAT(3X,I4,2X,I5,    I5,1X,I1,I5,I5,1X,5A4,1X,
     *'MIN=',F13.5,', MODE=',F14.5,', MAX=',F14.5)
      GO TO 1090
 1040 IF(PARM4(I,J).EQ.-1.)GO TO 1050
      WRITE(6,1041)I,NODEO(I),NODET(I),J,TIME(I,J),COST(I,J),
     *   NORM,PARM2(I,J),PARM3(I,J)
 1041 FORMAT(3X,I4,2X,I5,    I5,1X,I1,I5,I5,1X,5A4,1X,
     *'MEAN=',F12.5,', ST.DEV=',F12.5)
      GO TO 1090
 1050 WRITE(6,1051)I,NODEO(I),NODET(I),J,TIME(I,J),COST(I,J),
     *   NORM,PARM1(I,J),PARM2(I,J),PARM3(I,J)
 1051 FORMAT(3X,I4,2X,I5,    I5,1X,I1,I5,I5,1X,5A4,1X,
     *'MIN=',F13.5,', MEAN=',F14.5,', MAX=',F14.5)
      GO TO 1090
 1060 WRITE(6,1061)I,NODEO(I),NODET(I),J,TIME(I,J),COST(I,J),
     *   RECT,PARM1(I,J),PARM3(I,J)
 1061 FORMAT(3X,I4,2X,I5,    I5,1X,I1,I5,I5,1X,5A4,1X,
     *'MIN=',F13.5,', MAX=',F15.5)
      GO TO 1090
 1080 WRITE(6,1081)I,NODEO(I),NODET(I),J,TIME(I,J),COST(I,J),
     *   FIXED,PARM1(I,J)
 1081 FORMAT(3X,I4,2X,I5,    I5,1X,I1,I5,I5,1X,5A4,1X,'TIME=',F12.5)
 1090 CONTINUE
      J=NNCT
      WRITE(6,1095)I,NODEO(I),NODET(I),J,TIME(I,J),COST(I,J)
 1095 FORMAT(3X,I4,2X,I5,    I5,1X,I1,I5,I5)
 1100 CONTINUE
      IF(TEST1)1104,1104,1106
 1104 WRITE(6,1105)
 1105 FORMAT(////'0THE INFORMATION INPUT TO THE DETERMINISTIC PROJECT SC
     *HEDULING PROGRAM WILL BE PRINTED.')
      GO TO 1110
 1106 WRITE(6,1107)
 1107 FORMAT(////'0THE INFORMATION INPUT TO THE DETERMINISTIC PROJECT SC
     *HEDULING PROGRAM WILL NOT BE PRINTED.')
 1110 IF(TEST2)1114,1114,1116
 1114 WRITE(6,1115)
 1115 FORMAT(' THE INTERMEDIATE OUTPUT GENERATED BY THE DETERMINISTIC PR
     *OJECT SCHEDULING PROGRAM WILL BE PRINTED.')
      GO TO 1120
 1116 WRITE(6,1117)
 1117 FORMAT(' THE INTERMEDIATE OUTPUT GENERATED BY THE DETERMINISTIC PR
     *OJECT SCHEDULING PROGRAM WILL NOT BE PRINTED.')

 1120 IF(JMAT)1128,1128,1121
 1121 WRITE(6,1122)
 1122 FORMAT('0THE SIMPLIFICATION PROGRAM WILL PRINT ONLY THE MINIMUM CU
     *TPUT.')
      GO TO 1130
 1128 WRITE(6,1129)
 1129 FORMAT('0THE SIMPLIFICATION PROGRAM WILL PRINT THE COMPLETE OUTPUT
     *.')
 1130 CONTINUE
      WRITE(6,1131)IEDF,NMAX,SAMSIZ
```

```
1131  FORMAT('0DURING THE SUBNETWORK ANALYSIS PROGRAM,'/                    183
     *5X,'THE NUMBER OF SUBDIVISIONS IN THE ESTIMATED DURATION DISTRIBUT    184
     *ION FOR EACH SUBNETWORK WILL BE ',I4,';'/                            185
     *5X, 'THE LARGEST NUMBER OF ACTIVITIES IN A SUBNETWORK FOR WHICH TH   186
     *E SUBNETWORK',1H','S'/10X, 'EXACT DISCRETE DURATION DISTRIBUTION W   187
     *ILL BE ENUMERATED IS ',I4,';'/                                       188
     *5X,  'THE MAXIMUM NUMBER OF ACTIVITY DURATION CONFIGURATIONS EXPLI   189
     *CITLY CONSIDERED IN THE DETERMINATION OF THE'/10X,'UPPER AND LOWER   190
     * BOUNDS ON EACH SUBNETWORK',1H','S DISCRETE DURATION DISTRIBUTION    191
     *WILL BE ',I5,';')                                                    192
      IF(IPCOL)1132,1132,1134                                             193
1132  WRITE(6,1133)                                                       194
1133  FORMAT(5X,'THE MAXIMUM-CLUSTER PROCEDURE WILL BE USED IN THE DETER   195
     *MINATION OF THE BOUNDS;')                                           196
      GO TO 1136                                                          197
1134  WRITE(6,1135)                                                       198
1135  FORMAT(5X,'THE UNION-CLUSTER PROCEDURE WILL BE USED IN THE DETERMI   199
     *NATION OF THE BOUNDS;')                                             200
1136  WRITE(6,1137)                                                       201
1137  FORMAT(5X,'THE FOLLOWING (THETA,LAMBDA) PAIRS WILL BE USED IN THE    202
     *FORMATION OF THE CLUSTERS:')                                        203
      IF(THELAM)1138,1138,1140                                            204
1138  WRITE(6,1139)                                                       205
1139  FORMAT(10X,'(   1.00000,   1.00000),(   2.00000,   2.00000),(   3.   206
     *00000,   2.00000).')                                                207
      GO TO 1150                                                          208
1140  WRITE(6,1141)(THETA(I),LAMBDA(I),I=1,3)                             209
1141  FORMAT(10X,'(',F10.5,',',F10.5,'),(',F10.5,',',F10.5,'),(',F10.5,   210
     *',',F10.5,').')                                                     211
1150  WRITE(6,1151)NT                                                     212
1151  FORMAT('0THE NUMBER OF SUBDIVISIONS IN THE NETWORK',1H','S SYNTHES   213
     *IZED COMPLETION TIME DISTRIBUTION WILL BE 'I2,'.')                   214
      IF(IOPT)1160,1160,1165                                              215
1160  WRITE(6,1161)PCT,PD                                                 216
1161  FORMAT('0THE ',F7.2,'-TH PERCENTILE OF THE NETWORK',1H','S APPROXI    217
     *MATE COMPLETION TIME DISTRIBUTION WILL BE COMPARED TO THE'/          218
     *10X,'SPECIFIED PROJECT DEADLINE OF ',F10.5,'.')                      219
      GO TO 1170                                                          220
1165  WRITE(6,1166)PD                                                     221
1166  FORMAT('0THE MEAN OF THE NETWORK',1H','S APPROXIMATE COMPLETION TI    222
     *ME DISTRIBUTION WILL BE COMPARED TO THE'/                            223
     *10X,'SPECIFIED PROJECT DEADLINE OF ',F10.5,'.')                      224
1170  WRITE(6,1171)TT                                                     225
1171  FORMAT('0THE TARGET TIME FOR THE FIRST ITERATION IS ',F10.5,'.'//)  226
C                                                                         227
C        CHECK FOR LOOPS IN THE PROJECT NETWORK                           228
C                                                                         229
      CALL LOOP                                                           230
      STOP                                                                231
      END                                                                 232
      SUBROUTINE LOOP                                                     233
C                                                                         234
C        THIS PROGRAM DETERMINES WHETHER THE PROJECT NETWORK IS ACYCLIC    235
C        OR CONTAINS NODES WHICH ARE PART OF LOOPS (CYCLES)               236
C                                                                         237
      IMPLICIT INTEGER*4 (A-Z)                                            238
      DIMENSION AI(9999),BI(9999)                                         239
      DIMENSION NAT(1000),TAIL(1000),HEAD(1000),NCT(1000)                 240
      COMMON /BLK1/M,MRK,NSRCE,NSINK,LNODEN,NA,TAIL,HEAD,NCT              241
C                                                                         242
      DATA                                                                243
```

```
      WRITE(6,2004)                                                    244
2004  FORMAT('0THIS IS THE OUTPUT FROM SUBPROGRAM LOOP. THE PROGRAM SEAR  245
     *CHES THE GIVEN PROJECT NETWORK FOR LOOPS (CYCLES).'/' A VALID ',   246
     *'PROJECT NETWORK SHOULD CONTAIN NO LOOPS.'///)                     247
      DO 4 J=1,LNODEN                                                  248
      A(J)=0                                                           249
4     B(J)=0                                                           250
C     FORM THE 1ST HIERARCHY                                           251
      INODE = 0                                                        252
   80 HIER = 2                                                         253
82    INODE=INODE+1                                                    254
      IF(INODE.EQ.NSRCE.OR.INODE.EQ.NSINK)GO TO 81                     255
      DO 5 I=1,M                                                       256
      IF(INODE.EQ.TAIL(I))GO TO 81                                     257
5     CONTINUE                                                         258
      GO TO 82                                                         259
81    J=0                                                              260
      DO 1 I= 1,M                                                      261
      IF(HEAD(I).NE.INODE) GO TO 1                                     262
      J = J+1                                                          263
      A(J)=TAIL(I)                                                     264
      IF (TAIL(I).EQ.INODE) GO TO 998                                  265
    1 CONTINUE                                                         266
      IF (J.EQ.0) GO TO 997                                            267
      IA=J                                                             268
      J=0                                                              269
C     FORM THE SUBSEQUENT HIERARCHIES                                  270
  102 CONTINUE                                                         271
      DO 2 II=1,IA                                                     272
      DO 3 I=1,M                                                       273
      IF (HEAD(I).NE.A(II)) GO TO 3                                    274
      IF(TAIL(I).EQ.INODE) GO TO 998                                   275
      IF(J.EQ.0) GO TO 40                                              276
      DO 10 K=1,J                                                      277
      IF(TAIL(I).EQ.B(K)) GO TO 11                                     278
10    CONTINUE                                                         279
40    CONTINUE                                                         280
      J=J+1                                                            281
      B(J)=TAIL(I)                                                     282
11    CONTINUE                                                         283
    3 CONTINUE                                                         284
    2 CONTINUE                                                         285
      IF (J.EQ.0) GO TO 997                                            286
      HIER = HIER+1                                                    287
      IA=J                                                             288
      J = 0                                                            289
      DO 20 I=1,IA                                                     290
20    A(I)=B(I)                                                        291
      GO TO 102                                                        292
  997 CONTINUE                                                         293
      WRITE(6,2002) INODE                                              294
 2002 FORMAT(' NODE',I5,' IS NOT PART OF A LOOP')                      295
      IF (INODE.NE.LNODEN)GO TO 80                                     296
      IF(ILOOP.EQ.1) GO TO 50                                          297
      WRITE(6,1000)                                                    298
 1000 FORMAT (' THERE ARE NO LOOPS IN THIS NETWORK')                   299
      GO TO 999                                                        300
  50  WRITE(6,51)                                                      301
51    FORMAT(' THERE ARE NO OTHER LOOPS IN THIS NETWORK')              302
      A(1000000000) = 0.                                               303
      GO TO 999                                                        304
```

```
  998 WRITE (6,1001) INODE,HIER                                          305
1001  FORMAT(' NODE ',I5,'IS PART OF A LOOP.  THERE ARE ',I3,' ACTIVITIE  306
     *S IN THE LOOP.')                                                   307
      ILOOP=1                                                            308
      INOOO=INODE+1                                                      309
      IF(INOOO.NE.LNODEN)GO TO 80                                        310
999   RETURN                                                             311
      END                                                                312
```

2. DPS

```
C      DETERMINISTIC PROJECT SCHEDULING PROGRAM                                  1
C                                                                                2
C     DETERMINISTIC PROJECT SCHEDULER:  THIS PROGRAM COMPUTES THE                3
C     MINIMUM PROJECT COST AND ASSOCIATED OPTIMAL FIXED ACTIVITY TIMES           4
C     FOR EVERY FEASIBLE PROJECT DEADLINE.                                       5
C-                                                                               6
C         ALL VARIABLES ARE INTEGER.                                             7
C      (IF ANY VARIABLE IS NOT ALREADY IN INTEGER FORM,  THE VALUES MUST         8
C      BE RESCALED - THAT IS, MULTIPLIED BY AN APPROPRIATE POWER OF 10 -         9
C      UNTIL THE VALUES ARE INTEGER.)                                           10
C                                                                               11
C*******************************************************************************12
C                                                                               13
C      DEFINITION OF VARIABLES:                                                 14
C                                                                               15
C         ABAR(I,J) = TIME(I,NK(I)+1-J) + XNODE(ORIG(I))-XNODE(TERM(I))         16
C           C(I,J) = DECREASE IN I TH ACT'S COST PER UNIT FOR J TH TIME         17
C              CAP = MIN(FLOW REACHING ORIGIN NODE, EXCESS CAPACITY TO          18
C                        TERMINAL NODE)                                         19
C         COST(I,J) = COST OF COMPLETING ACTIVITY I AT TIME(I,J)                 20
C               DEL = MIN(DELTA1,DELTA2)                                       21
C            DELTA1 = MIN(-ABAR(I,J) WITH I LABELED AND J UNLABELED,           22
C                        ABAR(I,J)<0)                                           23
C            DELTA2 = MIN(ABAR(I,J) WITH I UNLABELED AND J LABELED,            24
C                        ABAR(I,J)>0)                                           25
C          DIREC(J) = DIRECTION OF FLOW REACHING NODE J                         26
C                        (0=FORWARD, 1=REVERSE)                                 27
C         FLOW(I,J) = FLOW IN J TH PIECE OF ACTIVITY I                          28
C               INF = ANY NUMBER GREATER THAN MAX(CAP)                          29
C                        (CURRENTLY SET AT (2*MAX +1))                          30
C             K1(I) = THE NUMBER OF THE TIME-COST PIECE USED IN                  31
C                        LABELING TERM(I) FROM ORIG(I)                          32
C            KOUNT = KEEPS TRACK OF ORDER IN WHICH NODES WERE LABELED           33
C          LABEL(I) = 0 IF NODE I UNLABELED                                     34
C                     1 IF NODE I LABELED                                       35
C            LINPUT = SPECIFIC VALUE OF LAMBDA IF TEST3=1 OR 2                  36
C                NA = TOTAL NUMBER OF ACTIVITIES                                37
C              NK(I) = NUMBER OF DIFFERENT TIMES AND COSTS FOR ACTIVITY I       38
C                NN = TOTAL NUMBER OF NODES                                     39
C           ORIG(I) = ORIGIN NODE FOR ACTIVITY I                                40
C          ORIG2(I) = WHERE THE FLOW IS FROM - USED IN LABELING ONLY            41
C             PCOST = PROJECT COST FUNCTION                                     42
C              SINK = NUMBER OF THE SINK NODE                                   43
C            SOURCE = NUMBER OF THE SOURCE NODE                                 44
C          TERM(I) = TERMINAL NODE FOR ACTIVITY I                               45
C            TEST1 = OPTION TO SUPPRESS PRINTING OF INPUT                       46
C                        (0=PRINT, 1=NO PRINT)                                  47
C            TEST2 = OPTION TO SUPPRESS INTERMEDIATE OUTPUT                     48
C                        (0=PRINT, 1=NO PRINT)                                  49
C            TEST3 = OPTION TO SPECIFY VALUE FOR LAMBDA                         50
C                        (0=NO, 1=YES AND SEE INTERMEDIATE OUTPUT,              51
C                        2=YES BUT NO INTERMEDIATE OUTPUT)                      52
C         TIME(I,J) = J TH BREAKPOINT (DURATION TIME) FOR ACTIVITY I            53
C           XACT(I) = ACTIVITY DURATION TIME                                    54
C          XNODE(I) = NODE TIME                                                 55
C          XDIFF(I) = XNODE(ORIG(I))-XNODE(TERM(I)),  AN UPPER BOUND ON         56
C                        THE ACTIVITY DURATION TIME                             57
C       I,J,K,M,N,P = INDICES                                                   58
C      INODE,ITERM,IACT,IORIG,IDIFF,ETC.                                        59
C                  = NON-INDEXED VERSIONS OF XNODE(I),TERM(I),XACT(I),          60
```

```
C                        ORIG(I),XDIFF(I),ETC.                                  61
C                                                                               62
C*******************************************************************************63
C                                                                               64
C        FOR THE SAKE OF IDENTIFYING THE APPROPRIATE DIMENSIONS, LET            65
C        MMAX = THE MAXIMUM NUMBER OF ACTIVITIES IN THE ORIGINAL                66
C                  PROJECT NETWORK                                              67
C        LNMAX = THE LARGEST NODE NUMBER ALLOWED IN THE PROJECT NETWORK         68
C        BKMAX = THE MAXIMUM NUMBER OF BREAK POINTS ALLOWED IN THE              69
C                  ENTIRE PROJECT'S TIME-COST CURVE                             70
C        NCTMAX = THE MAXIMUM NUMBER OF COMPLETION TIMES AND COSTS FOR          71
C                  EACH ACTIVITY                                                72
C        CURRENTLY, MMAX=1000; LNMAX=9999; BKMAX=3000; NCTMAX=6                 73
C                                                                               74
C     DIMENSION CAP(LNMAX),FLOW(MMAX,NCTMAX),C(MMAX,NCTMAX),ORIG(MMAX),        75
C    *          TERM(MMAX),TIME(MMAX,NCTMAX),COST(MMAX,NCTMAX),NK(MMAX),       76
C    *          ABAR(MMAX,NCTMAX),XDIFF(MMAX),XNODE(LNMAX),XACT(MMAX),         77
C    *          DIREC(LNMAX),LABEL(LNMAX),K1(LNMAX),ORIG2(LNMAX),              78
C    *          KOUNT(LNMAX),AORD(MMAX),ND(LNMAX),NDD(LNMAX),IP(MMAX),         79
C    *          CTIME(MMAX),FILE0(BKMAX),FILE6(2,MMAX),NUMS(MMAX)              80
C                                                                               81
C*******************************************************************************82
C                                                                               83
      IMPLICIT INTEGER*2(A-Z)                                                   84
      INTEGER*4 NA,NN,ORIG,TERM,NK,I,SOURCE,SINK,NUMS(1000),LNODEN              85
      REAL*4 CAP(9999),FLOW(1000, 6),C(1000, 6),PCOST,INF,PCOST1,TT,            86
     1KCOST,ACOST ,PNEW                                                         87
      COMMON ORIG,TERM,NK,NA,LNODEN,TIME,CTIME,XNODE,AORD,LMIN,LMAX,            88
     *       TEST1                                                              89
      DIMENSION ORIG(1000),TERM(1000),TIME(1000, 6),COST(1000, 6),              90
     *NK(1000),ABAR(1000, 6),XDIFF(1000),XNODE(9999),XACT(1000),                91
     *K1(9999),ORIG2(9999),KOUNT(9999),AORD(1000),CTIME(1000),                  92
     *DIREC(9999),LABEL(9999),ND(9999),NDD(9999),IP(1000)                       93
      INTEGER F0/8/,F1/9/,                                F6/14/                94
      DIMENSION FILE0(2000),FILE6(2,1000)                                       95
      WRITE(6,1500)                                                             96
1500  FORMAT(1H1,132('*')/1X,132('*')/'0THIS IS THE OUTPUT FROM THE DETE        97
     *RMINISTIC PROJECT SCHEDULER: DPS'//1X,132('*')/1X,132('*'))               98
C                                                                               99
C     INPUT DATA                                                               100
C                                                                              101
      REWIND F0                                                                102
      REWIND F1                                                                103
      KKKK=0                                                                   104
      READ(F0)NA,NN,SOURCE,SINK,LNODEN,NUMS,ORIG,TERM,NK                       105
      INF=0.                                                                   106
      PCOST=0.                                                                 107
      READ(F1)TEST1,TEST2,TEST3,LINPUT                                         108
      DO 12 I=1,NA                                                             109
      KN=NK(I)                                                                 110
      READ(F1)(TIME(I,J),J=1,KN),(COST(I,J),J=1,KN)                            111
      KN1=KN-1                                                                 112
      DO 12 J=1,KN1                                                            113
      READ(F1)                                                                 114
12    CONTINUE                                                                 115
      IF(TEST1.EQ.1) GO TO 401                                                 116
      WRITE(6,150) NN,NA,SOURCE,SINK                                           117
401   CALL ORDER                                                               118
C                                                                              119
C     SET UP INITIAL VALUES                                                    120
C                                                                              121
```

```
      IF(TEST1.EQ.1) GO TO 193                                          122
      K3=1                                                              123
  192 K2=K3+8                                                           124
      IF(K2.GT.LNODEN)K2=LNODEN                                         125
      WRITE(6,151) (K,K=K3,K2)                                          126
      WRITE(6,157) (XNODE(K),K=K3,K2)                                   127
      IF(K2.GE.LNODEN)GO TO 191                                         128
      K3=K2+1                                                           129
      GO TO 192                                                         130
  191 WRITE(6,152)                                                      131
  193 DO 10 I=1,NA                                                      132
      LABEL(I)=0                                                        133
      XDIFF(I)=XNODE(ORIG(I))-XNODE(TERM(I))                            134
      NKM1=NK(I)-1                                                      135
      KN=NK(I)                                                          136
      DO 9 J=1,NKM1                                                     137
      TT=TIME(I,J+1)-TIME(I,J)                                          138
      IF(TT)7,8,7                                                       139
    7 C(I,J)=(COST(I,J)-COST(I,J+1))/TT                                 140
      GO TO 6                                                           141
    8 C(I,J)=0.                                                         142
    6 IF(INF.LT.C(I,J))  INF=C(I,J)                                     143
      XACT(I)=XDIFF(I)                                                  144
      IF(XACT(I).LT.TIME(I,J+1)) XACT(I)=TIME(I,J+1)                    145
      JJ=NK(I)-J+1                                                      146
      ABAR(I,J)=TIME(I,JJ)+XDIFF(I)                                     147
      FLOW(I,J)=0                                                       148
    9 CONTINUE                                                          149
      FILE6(1,I)=0                                                      150
      FILE6(2,I)=XACT(I)                                                151
      ABAR(I,KN)=TIME(I,1)+XDIFF(I)                                     152
      FLOW(I,KN)=0                                                      153
      IF(TEST1.EQ.1) GO TO 10                                           154
      WRITE(6,153)NUMS(I),XACT(I),ORIG(I),TERM(I),(J,TIME(I,J),COST(I,J)155
     *,C(I,J),ABAR(I,J),J=1,NKM1)                                       156
      WRITE(6,156) KN,TIME(I,KN),COST(I,KN),ABAR(I,KN)                  157
   10 CONTINUE                                                          158
      WRITE(F6)((FILE6(I,J),I=1,2),J=1,NA)                              159
      INF=2.*INF+1.                                                     160
      DO 417 I=1,NA                                                     161
      C(I,NK(I))=0.                                                     162
      NKM1=NK(I)-1                                                      163
      PCOST1=0.                                                         164
      IKK=0                                                             165
      DO 418 K=1,NKM1                                                   166
      IF(K.NE.1) GO TO 40                                               167
      XIJ=XACT(I)                                                       168
      IF(XIJ.GT.TIME(I,2)) XIJ=TIME(I,2)                                169
      GO TO 41                                                          170
   40 XIJ=XACT(I)-TIME(I,K)                                             171
      IF(XIJ.LT.0) XIJ=0                                                172
      IF(XIJ.GT.(TIME(I,K+1)-TIME(I,K))) XIJ=TIME(I,K+1)-TIME(I,K)      173
      IF(IKK.EQ.1) GO TO 41                                             174
      IF(C(I,K).GT.C(I,K-1)) GO TO 50                                   175
   41 PCOST1=PCOST1+C(I,K)*XIJ                                          176
      GO TO 418                                                         177
   50 IKK=1                                                             178
      WRITE(6,237) I,I                                                  179
      PCOST1=PCOST1+C(I,K)*XIJ                                          180
  418 CONTINUE                                                          181
      PCOST=PCOST+COST(I,1)+C(I,1)*TIME(I,1)-PCOST1                     182
```

```
      PNEW=PCOST                                                        183
  417 CONTINUE                                                          184
      LAMBDA=LMAX                                                       185
      KKKK=KKKK+1                                                       186
      FILE0(KKKK)=LAMBDA                                                187
      IF (TEST3.GE.1) GO TO 700                                         188
      WRITE(6,154)                                                      189
      LINPUT=0                                                          190
      GO TO 96                                                          191
700   CONTINUE                                                          192
      IF(LINPUT.LT.LMIN) GO TO 705                                      193
      IF(LINPUT.GE.LMAX) GO TO 704                                      194
      IF (TEST3.EQ.2) GO TO 724                                         195
      WRITE(6,155) LINPUT                                               196
   96 WRITE(6,200) LAMBDA, PCOST                                        197
      IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 724                            198
      WRITE(6,235)                                                      199
  724 CAP(SOURCE)=INF                                                   200
      ITER=0                                                            201
   99 LABEL(SOURCE)=1                                                   202
      IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO  97                            203
      ITER=ITER+1                                                       204
      WRITE(6,225) ITER                                                 205
C                                                                       206
C     INITIAL LABELING ITERATION                                        207
C                                                                       208
   97 I=1                                                               209
      J=SOURCE                                                          210
      M=0                                                               211
C     IF ACTIVITY STARTS AT DESIGNATED ORIGIN, TRY TO LABEL.            212
C     OTHERWISE, CHANGE ORIGINS.                                        213
   14 IF (ORIG(I).NE.J) GO TO 13                                        214
      ITERM=TERM(I)                                                     215
C     CHECK IF NODE ALREADY LABELED AND                                 216
C     CHECK IF ABAR(I,NK(I))=0.                                         217
      IF (LABEL(ITERM).NE.0.OR.ABAR(I,NK(I)).NE.0) GO TO 13             218
C     IF NODE NOT ALREADY LABELED AND ABAR(I,NK(I))=0,                  219
C     PROCEDE WITH LABELING.                                            220
      LABEL(ITERM)=1                                                    221
      ORIG2(ITERM)=J                                                    222
      K1(ITERM)=NK(I)                                                   223
      DIREC(ITERM)=I                                                    224
      CAP(ITERM)=INF                                                    225
      IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 403                            226
      WRITE(6,201) ITERM,ORIG2(ITERM),K1(ITERM)                         227
C     IF CAN REACH SINK, TERMINATE (IMPLIES INFEASIBLE)                 228
  403 IF (ITERM.EQ.SINK) GO TO 15                                       229
      M=M+1                                                             230
      KOUNT(M)=ITERM                                                    231
C     IF EVERY PATH TESTED AND INFINITE FLOW NOT POSSIBLE,              232
C     GO ON TO LABELING PART(II).                                       233
   13 I=I+1                                                             234
      IF (I.GT.NA) GO TO 11                                             235
      GO TO 14                                                          236
C     CHANGE DESIGNATED ORIGINS.                                        237
   11 IF (J.EQ.SOURCE) P=1                                              238
      IF(P.GT.M) GO TO 16                                               239
C     IF ALL LABELED NODES HAVE BEEN SCANNED AND NO NEW NODES           240
C     HAVE BEEN LABELED, GO ON TO LABELING PART (II).                   241
      J=KOUNT(P)                                                        242
      P=P+1                                                             243
```

```
      I=1                                                                244
      GO TO 14                                                           245
   15 IF(TEST3.GE.1) GO TO 404                                           246
      WRITE(6,202) LAMBDA                                                247
  404 GO TO 999                                                          248
   16 IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 405                             249
      WRITE(6,203)                                                       250
C                                                                        251
C     NEXT LABELING PROCEDURE                                            252
C                                                                        253
  405 I=1                                                                254
      J=SOURCE                                                           255
C     AGAIN, CHECK ALL CONDITIONS FOR LABELING                           256
C     IE. CHECK IF NODE IS ALREADY LABELED, IF ABAR(I,J)=0, AND          257
C     IF THE FLOW(I,J) IS LESS THAN ITS UPPER BOUND.                     258
   20 IF (ORIG(I).NE.J) GO TO 24                                         259
      ITERM=TERM(I)                                                      260
      KN=NK(I)                                                           261
      DO 25 K=1,KN                                                       262
      IF (K.EQ.KN) GO TO 27                                              263
      IF(LABEL(ITERM).NE.0.OR.ABAR(I,K).NE.0.OR.FLOW(I,K).GE.            264
     1(C(I,NK(I)-K)-C(I,NK(I)-K+1)))GO TO 25                             265
      DIREC(ITERM)=I                                                     266
C     CAPACITY IS MIN OF PREVIOUS FLOW AND THE EXCESS CAPACITY           267
      CAP(ITERM)=C(I,NK(I)-K)-C(I,NK(I)-K+1) - FLOW(I,K)                 268
      GO TO 23                                                           269
   27 IF(LABEL(ITERM).NE.0.OR.ABAR(I,K).NE.0.OR.FLOW(I,K).GE.INF)        270
     1  GO TO 25                                                        271
C     IF THE NODE HAS NOT ALREADY BEEN LABELED, ABAR(I,J)=0, AND         272
C     THE FLOW IS LESS THAN ITS UPPER BOUND, PROCEDE WITH THE LABELING   273
C     OF THE NODE.                                                       274
      DIREC(ITERM)=I                                                     275
      CAP(ITERM)=INF                                                     276
   23 LABEL(ITERM)=1                                                     277
      ORIG2(ITERM)=J                                                     278
      K1(ITERM)=K                                                        279
      IF (CAP(ITERM).GT.CAP(ORIG(I)))  CAP(ITERM)=CAP(ORIG(I))           280
      IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 406                             261
      WRITE(6,204) ITERM,ORIG2(ITERM),K1(ITERM),DIREC(ITERM),CAP(ITERM)  282
C     IF SINK LABELED, GO TO UPDATE PROCEDURE                            283
  406 IF (ITERM.EQ.SINK) GO TO 21                                        284
      M=M+1                                                              285
      KOUNT(M)=ITERM                                                     286
C     CHECK IF ALL PATHS TRIED                                           287
   25 CONTINUE                                                           288
      GO TO 19                                                           289
   24 IF(TERM(I).NE.J) GO TO 19                                          290
      IORIG=ORIG(I)                                                      291
      KN=NK(I)                                                           292
      DO 26 K=1,KN                                                       293
      IF(LABEL(IORIG).NE.0.OR.ABAR(I,K).NE.0.OR.FLOW(I,K).LE.0)          294
     2 GO TO 26                                                          295
      DIREC(IORIG)=-I                                                    296
      CAP(IORIG)=FLOW(I,K)                                               297
      LABEL(IORIG)=1                                                     298
      ORIG2(IORIG)=J                                                     299
      K1(IORIG)=K                                                        300
      IF(CAP(IORIG).GT.CAP(TERM(I))) CAP(IORIG)=CAP(TERM(I))             301
      IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 402                             302
      WRITE(6,204) IORIG,ORIG2(IORIG),K1(IORIG),DIREC(IORIG),CAP(IORIG)  303
  402 M=M+1                                                              304
```

```
      KOUNT(M)=IORIG                                                    305
   26 CONTINUE                                                          306
   19 I=I+1                                                             307
      IF (I.GT.NA) GO TO 18                                             308
      GO TO 20                                                          309
   18 IF (J.EQ.SOURCE) P=1                                              310
      IF (P.GT.M) GO TO 22                                              311
      J=KOUNT(P)                                                        312
      P=P+1                                                             313
      I=1                                                               314
      GO TO 20                                                          315
C                                                                       316
C     NONBREAKTHROUGH HAS OCCURED. DELTAS ARE FOUND AND UPDATING        317
C     MADE IN THE XNODES AND XACTS.                                     318
C                                                                       319
   22 DELTA1=INF+1                                                      320
      DELTA2=INF+1                                                      321
      DO 4 I=1,NA                                                       322
      KN=NK(I)                                                          323
      IF (LABEL(ORIG(I)).EQ.1.AND.LABEL(TERM(I)).EQ.0) GO TO 1          324
C     A1 IS SET OF I LABELED AND J UNLABELED.                           325
C     A2 IS SET OF I UNLABELED AND J LABELED.                           326
      IF(LABEL(ORIG(I)).EQ.0.AND.LABEL(TERM(I)).EQ.1) GO TO 2           327
      GO TO 4                                                           328
C     FINDING DELTA1'S.                                                 329
    1 DO 3 J=1,KN                                                       330
      IF (ABAR(I,J).GE.0) GO TO 3                                       331
      IF (-ABAR(I,J).LT.DELTA1) DELTA1=-ABAR(I,J)                       332
    3 CONTINUE                                                          333
      GO TO 4                                                           334
C     FINDING DELTA2'S                                                  335
    2 DO 5 J=1,KN                                                       336
      IF(ABAR(I,J).LE.0) GO TO 4                                        337
      IF (ABAR(I,J).LT.DELTA2) DELTA2= ABAR(I,J)                        338
    5 CONTINUE                                                          339
    4 CONTINUE                                                          340
C     DEL=MIN(DELTA1,DELTA2)                                            341
      DEL=DELTA1                                                        342
      IF (DELTA2.LT.DEL) DEL=DELTA2                                     343
      LAMBDA=LAMBDA-DEL                                                 344
      KKKK=KKKK+1                                                       345
      FILE0(KKKK)=LAMBDA                                                346
C     UPDATING THE XNODES.                                              347
      IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 407                            348
      WRITE(6,206) LAMBDA                                               349
  407 IF (TEST3.EQ.2) GO TO 721                                         350
      DELTA= LAMBDA + DEL                                               351
      WRITE(6,209)          DEL,DELTA,LAMBDA,DELTA                      352
  721 IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 408                            353
      WRITE(6,207)                                                      354
  408 DO 80 I=1,LNODEN                                                  355
      INODE=XNODE(I)                                                    356
      IF(LABEL(I).EQ.0) GO TO 81                                        357
      IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 409                            358
      WRITE(6,210) I,INODE                                              359
  409 XNODE(I)=INODE                                                    360
      GO TO 80                                                          361
   81 IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 410                            362
      WRITE(6,211) I,INODE                                              363
  410 XNODE(I)=INODE-DEL                                                364
   80 CONTINUE                                                          365
```

```
C     UPDATING THE XACTS (3 PARTS)                                              36
      IF (TEST3.EQ.2) GO TO 722                                                 36
      WRITE(6,212) DELTA                                                        36
  722 PCOST=0.                                                                  36
      DO 82 I=1,NA                                                              37
      IP(I)=0                                                                   37
      PCOST1=0.                                                                 37
      NKM1=NK(I)-1                                                              37
      IACT=TIME(I,NK(I))                                                        37
      IORIG=ORIG(I)                                                             37
      ITERM=TERM(I)                                                             37
      IDIFF=XNODE(ITERM)-XNODE(IORIG)                                           37
      XDIFF(I)=-IDIFF                                                           37
      IF (IDIFF.GE.IACT) GO TO 86                                               37
      XACT(I)=IDIFF                                                             38
      DO 550 K=1,NKM1                                                           38
      IF(K.NE.1) GO TO 43                                                       38
      XIJ=XACT(I)                                                               38
      IF(XIJ.GT.TIME(I,2)) XIJ=TIME(I,2)                                        38
      FLAG1=0                                                                   38
      GO TO 42                                                                  38
   43 XIJ=XACT(I)-TIME(I,K)                                                     38
      IF(XIJ.LT.0) GO TO 552                                                    38
      IF(XIJ.GT.(TIME(I,K+1)-TIME(I,K))) XIJ=TIME(I,K+1)-TIME(I,K)              38
      FLAG1=0                                                                   39
      GO TO 42                                                                  39
  552 FLAG1=1                                                                   39
      FLAG2=K-1                                                                 39
      GO TO 553                                                                 39
   42 PCOST1=PCOST1+C(I,K)*XIJ                                                  39
  550 CONTINUE                                                                  39
  553 KCOST=COST(I,1)+C(I,1)*TIME(I,1)                                          39
      ACOST=KCOST-PCOST1                                                        39
      PCOST=PCOST+ACOST                                                         39
      IF (LABEL(IORIG)-LABEL(ITERM)) 83,84,85                                   40
   83 IDIFF=IDIFF-DEL                                                           40
      FILE6(1,I)=1                                                              40
      FILE6(2,I)=IDIFF                                                          40
      IF (TEST3.EQ.2) GO TO 82                                                  40
      IF(FLAG1.EQ.1) GO TO 59                                                   40
      ACOST=ACOST + C(I,NKM1)*DEL                                               40
      IP(I)=1                                                                   40
      WRITE(6,214)NUMS(I),IDIFF,ACOST,C(I,NKM1)                                 40
      GO TO 82                                                                  40
   59 ACOST=ACOST+C(I,FLAG2)*DEL                                                41
      IP(I)=1                                                                   41
      WRITE(6,214)NUMS(I),IDIFF,ACOST,C(I,FLAG2)                                41
      GO TO 82                                                                  41
   84 FILE6(1,I)=0                                                              41
      FILE6(2,I)=XACT(I)                                                        41
      IF(TEST3.EQ.2)GO TO 82                                                    41
      WRITE(6,216)NUMS(I),XACT(I),ACOST                                         41
      GO TO 82                                                                  41
   85 IDIFF=IDIFF+DEL                                                           41
      FILE6(1,I)=-1                                                             42
      FILE6(2,I)=IDIFF                                                          42
      IF(TEST3.EQ.2)GO TO 82                                                    42
      IF(FLAG1.EQ.1) GO TO 58                                                   42
      ACOST=ACOST - C(I,NKM1)*DEL                                               42
      IP(I)=2                                                                   42
      WRITE(6,213)NUMS(I),IDIFF,ACOST,C(I,NKM1)                                 42
```

```
      GO TO 82                                                          427
   58 ACOST=ACOST-C(I,FLAG2)*DEL                                        428
      IP(I)=2                                                           429
      WRITE(6,213)NUMS(I),IDIFF,ACOST,C(I,FLAG2)                        430
      GO TO 82                                                          431
   86 XACT(I)=IACT                                                      432
      DO 551 K=1,NKM1                                                   433
      IF(K.NE.1) GO TO 45                                               434
      XIJ=XACT(I)                                                       435
      IF(XIJ.GT.TIME(I,2)) XIJ=TIME(I,2)                                436
      GO TO 46                                                          437
   45 XIJ=XACT(I)-TIME(I,K)                                             438
      IF(XIJ.LT.0) XIJ=0                                                439
      IF(XIJ.GT.(TIME(I,K+1)-TIME(I,K))) XIJ=TIME(I,K+1)-TIME(I,K)      440
   46 PCOST1=PCOST1+C(I,K)*XIJ                                          441
  551 CONTINUE                                                          442
      KCOST=COST(I,1)+C(I,1)*TIME(I,1)                                  443
      ACOST=KCOST-PCOST1                                                444
      PCOST=PCOST+ACOST                                                 445
      FILE6(1,I)=0                                                      446
      FILE6(2,I)=XACT(I)                                                447
      IF (TEST3.EQ.2) GO TO 82                                          448
      WRITE(6,216)NUMS(I),XACT(I),ACOST                                 449
   82 CONTINUE                                                          450
      WRITE(F6)((FILE6(I,J),I=1,2),J=1,NA)                              451
      IF (TEST3.EQ.2) GO TO 723                                         452
      PCOST1=PNEW                                                       453
      PNEW=(PCOST-PNEW)/DEL                                             454
      WRITE(6,224) PCOST1,PNEW                                          455
  723 PNEW=PCOST                                                        456
      IF (TEST3.NE.0.AND.LAMBDA.LE.LINPUT) GO TO 703                    457
C     RESET LABELS TO 0 AND REFIGURE ABARS.                             458
C     THEN START OVER.                                                  459
      DO 87 I=1,LNODEN                                                  460
      LABEL(I)=0                                                        461
   87 CONTINUE                                                          462
      IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 420                            463
      WRITE(6,226) (J,J=1,11)                                           464
  420 DO 88 I=1,NA                                                      465
      NKM1=NK(I)-1                                                      466
      DO 500 K=1,NKM1                                                   467
      J=NKM1+2-K                                                        468
  500 ABAR(I,K)=TIME(I,J)+XDIFF(I)                                      469
      ABAR(I,NK(I))=TIME(I,     1)+XDIFF(I)                             470
      IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO  88                            471
      NK1=NK(I)                                                         472
      WRITE(6,227)NUMS(I),(ABAR(I,J),J=1,NK1)                           473
   88 CONTINUE                                                          474
      IF (LAMBDA.LT.LMIN) GO TO 998                                     475
      GO TO 99                                                          476
C                                                                       477
C     UPDATE THE FLOW AFTER BREAKTHROUGH.                               478
C                                                                       479
   21 IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO  34                            480
      WRITE(6,205)                                                      481
   34 FLOW(I,K1(ITERM))=FLOW(I,K1(ITERM))+CAP(SINK)                     482
C     IF DIREC =0 THEN CAP ADDED TO FLOW.                               483
C     IF DIREC =1 THEN CAP IS SUBTRACTED.                               484
   30 ITERM=CRIG2(ITERM)                                                485
C     CHECK IF BACK AT SOURCE.                                          486
      IF(ITERM.EQ.SOURCE) GO TO 33                                      487
```

```
      I=DIREC(ITERM)                                                    488
      I=IABS(I)                                                         489
      IF (DIREC(ITERM).GT.0) GO TO 34                                   490
      FLOW(I,K1(ITERM))=FLOW(I,K1(ITERM))-CAP(SINK)                     491
      GO TO 30                                                          492
C     RELABEL AND START OVER.                                           493
   33 IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 415                            494
      DO 560 I=1,NA                                                     495
      NK1=NK(I)                                                         496
      DO 560 K=1,NK1                                                    497
560   WRITE(6,220)NUMS(I),K,FLOW(I,K)                                   498
  415 DO 98 I=1,LNODEN                                                  499
      LABEL(I)=0                                                        500
   98 CONTINUE                                                          501
      GO TO 99                                                          502
C     PROGRAM TERMINATES WHEN EVENTUALLY AN INFINITE FLOW IS ACHIEVED   503
C     FROM THE SOURCE TO THE SINK.  OR WHEN THE VALUE OF LAMBDA DROPS   504
C     BELOW THE MINIMUM LENGTH OF THE NETWORK.                          505
  998 IF(TEST3.NE.0) GO TO 999                                          506
      WRITE(6,202) LAMBDA                                               507
      GO TO 999                                                         508
  705 WRITE(6,233) LINPUT,LMIN                                          509
      GO TO 999                                                         510
  704 WRITE(6,236) LINPUT,LMAX                                          511
      WRITE(6,238) LINPUT                                               512
      D=0                                                               513
      DO 60 I=1,NA                                                      514
   60 IP(I)=0                                                           515
      GO TO 707                                                         516
  703 WRITE(6,234) LINPUT                                               517
  706 WRITE(6,238) LINPUT                                               518
      D=LINPUT-LAMBDA                                                   519
  707 PCOST=0.                                                          520
      DO 57 I=1,NA                                                      521
      IF(IP(I).EQ.1.AND.D.GT.0) XACT(I)=XACT(I)-D                       522
      IF(IP(I).EQ.2.AND.D.GT.0) XACT(I)=XACT(I)+D                       523
      PCOST1=0.                                                         524
      NKM1=NK(I)-1                                                      525
      DO 51  K=1,NKM1                                                   526
      IF(K.NE.1) GO TO 52                                               527
      XIJ=XACT(I)                                                       528
      IF(XIJ.GT.TIME(I,2)) XIJ=TIME(I,2)                                529
      GO TO 53                                                          530
   52 XIJ=XACT(I)-TIME(I,K)                                             531
      IF(XIJ.LT.0) XIJ=0                                                532
      IF(XIJ.GT.(TIME(I,K+1)-TIME(I,K))) XIJ=TIME(I,K+1)-TIME(I,K)      533
   53 PCOST1=PCOST1+C(I,K)*XIJ                                          534
   51 CONTINUE                                                          535
      KCOST=COST(I,1)+C(I,1)*TIME(I,1)                                  536
      ACOST=KCOST-PCOST1                                                537
      WRITE(6,216)NUMS(I),XACT(I),ACOST                                 538
   57 PCOST=PCOST+ACOST                                                 539
      WRITE(6,239) PCOST                                                540
  999 WRITE(6,228)                                                      541
      READ(F0)                                                          542
      WRITE(F0)KKKK,(FILE0(I),I=1,KKKK)                                 543
      ENDFILE F6                                                        544
      ENDFILE F0                                                        545
      WRITE(6,228)                                                      546
      STOP                                                              547
  100 FORMAT(I4,1X,I4,1X,I1,1X,I1,1X,I4,1X,I4,1X,I1)                    548
```

```
150 FORMAT('-','THE NUMBER OF NODES IS ',I4,'.',/,1X,'THE NUMBER OF AC
   1TIVITIES IS ',I4,'.',/,1X,'THE SOURCE NODE IS NUMBERED ',I4,' AND
   2THE SINK NODE IS NUMBERED ',I4,'.',/,'-',' ** NODES: **')
151 FORMAT('0',16X,'K',7X,9(3X,I4,5X))
152 FORMAT('-',' ** ACTIVITIES: **',//,6X,'I',7X,'XACT',6X,'ORIG',3X,
   1'TERM',4X,'J',6X,'TIME',9X,'COST',14X,'C',13X,'ABAR')
153 FORMAT(' ',3X,I4,3X,I10,3X,I4,3X,I4,(T39,I2,3X,I10,3X,I10,3X,
   1E16.5,3X,I10))
154 FORMAT('-','THE ENTIRE PROJECT COST CURVE WILL BE DETERMINED.')
155 FORMAT('-','THE OPTIMAL ACTIVITY COMPLETION TIMES FOR A SPECIFIED
   1PROJECT DEADLINE TIME = ',I10,' ARE GOING TO BE DETERMINED.')
156 FORMAT(' ',T39,I2,3X,I10,3X,I10,22X,I10)
157 FORMAT('0',                                4X,'INITIAL XNODE(K)',3X,
   19(I10,2X))
200 FORMAT('0','LAMBDA = PROJECT COMPLETION TIME',//,
   1         1X,'THE STARTING VALUE OF LAMBDA IS ',I10,'.',//
   2          ,1X,'THE CORRESPONDING TOTAL PROJECT COST IS ',E16.5,'.')
201 FORMAT('0','THE NODE ',I4,' HAS THE LABEL (',I4,',',I4,',0,INF).')
202 FORMAT('0',////,30X,'* * * * * *',//////,1X,
   1           'THE SINK WAS REACHED WITH INFINITE CAPACITY IMPLYING A
   1N INFEASIBLE SOLUTION TO THE PRIMAL PROBLEM ',/,20X,'IF LAMBDA DRO
   2PS BELOW ITS CURRENT VALUE, ',I10,'.')
203 FORMAT('-','THE SINK HAS NOT BEEN REACHED WITH INFINITE CAPACITY -
   1 CONTINUE WITH THE LABELING PROCESS.',/  ,1X,'THE NODES THAT HAVE
   2BEEN LABELED WILL RETAIN THAT LABEL FOR THE REMAINDER OF THE ITERA
   3TION.')
204 FORMAT('0','THE NODE ',I4,' HAS THE LABEL (',I4,',',I4,',',I4,',',
   1E16.5,').')
205 FORMAT('-','BREAKTHROUGH:  UPDATE THE DUAL VARIABLES.'
   1,///,1X,'  ACTIVITY #: I',3X,'J',9X,'NEW FLOW: F(I,J)')
206 FORMAT('-','NONBREAKTHROUGH:  UPDATE THE PRIMAL VARIABLES;',/,1X,
   1'I.E.  DETERMINE OPTIMAL ACTIVITY TIMES FOR LAMBDA = ',I10,'.')
207 FORMAT(' ','  NODE #: K',5X,'NEW VALUE: XNODE(K)')
209 FORMAT('-','DELTA (REPRESENTED BY "D") RANGES FROM 0 TO'
   1 ,I4,'.',/,1X,'LAMBDA RANGES FROM',I10,' TO',I10,'.',
   2                  /,1X,'THE MINIMUM COST PROJECT SCHEDULE FOR PRO
   3JECT DEADLINE = ',I10,'-D:')
210 FORMAT(' ',7X,I4,12X,I10)
211 FORMAT(' ',7X,I4,12X,I10,'-D')
212 FORMAT('-','PROJECT COMPLETION TIME = ',I10,'-D.',//,1X,
   1          '  ACTIVITY #: I',3X,'NEW VALUE: XACT(I)',9X,'ACTIVITY CO
   2ST')
213 FORMAT(' ',5X,I4,12X,I10,'-D',9X,E16.5,' + (',E13.5,'*D)')
214 FORMAT(' ',5X,I4,12X,I10,'+D',9X,E16.5,' + (',E13.5,'*D)')
216 FORMAT(' ',5X,I4,12X,I10,11X,E16.5)
220 FORMAT(' ',12X,I4,2X,I2,7X,E16.5)
224 FORMAT('0','THE CURRENT VALUE OF THE PROJECT COST IS ',E16.5,
   1' + (',E13.5,'*D).')
225 FORMAT('-','***      ITERATION NUMBER',I6,'      ***')
226 FORMAT('-','NEW VALUES OF ABAR FOR J=1,2,...,NK(I)',//,6X,'I',3X,
   1'J:',11(5X,I2,3X))
227 FORMAT(' ',2X,I4,7X,11(I8,2X))
228 FORMAT(1H1)
230 FORMAT(I4,1X,I4,1X,I2)
231 FORMAT(8I10)
232 FORMAT(I10)
233 FORMAT('-','THE SPECIFIED VALUE OF LAMBDA,',I10,',IS LESS THAN THE
   1 MINIMUM VALUE,',I10,',IMPLYING AN INFEASIBLE SOLUTION.',//,1X,
   2'THE PROBLEM WILL NOT BE WORKED.')
234 FORMAT('1','THE SPECIFIED VALUE OF LAMBDA,',I10,',HAS BEEN REACHED
   1.')
```

```
  235 FORMAT('0',   'THE SOURCE HAS A VALUE OF ZERO AND IS ASSIGNED THE      610
     3LABEL (-,-,-,INF).',// )                                               611
  236 FORMAT('-','THE SPECIFIED VALUE OF LAMBDA, ',I10,',IS GREATER THAN     612
     1 OR EQUAL TO THE MAXIMUM VALUE, ',I10,';',//,                          613
     1                                    1X,'THEREFORE, THE ORIGINAL        614
     2XNODE(K)''S AND XACT(I)''S ARE OPTIMAL.')                              615
  237 FORMAT(' ','** WARNING:  ACTIVITY NUMBER ',I4,' HAS A NON-CONVEX C     616
     1OST FUNCTION;',/,12X,'IE. THE C(',I4,',M)''S ARE NOT NON-INCREASIN     617
     2G.')                                                                   618
  238 FORMAT('-','FOR PROJECT COMPLETION TIME = ',I10,', THE OPTIMAL SOL     619
     1UTION IS:',//,1X,                                                      620
     1          '  ACTIVITY #: I',3X,'NEW VALUE: XACT(I)',9X,'ACTIVITY CO    621
     2ST')                                                                   622
  239 FORMAT('-','THE CORRESPONDING PROJECT COST IS ',E16.5,'.')             623
      END                                                                    624
      SUBROUTINE ORDER                                                       625
C                                                                            626
C        THIS SUBROUTINE DETERMINES THE ORDER IN WHICH TO CONSIDER           627
C        THE ACTIVITIES FOR THE CALCULATION OF THE CRITICAL PATH TIME        628
C                                                                            629
      IMPLICIT INTEGER*2(A-Z)                                                630
      INTEGER*4 NA,NN,ORIG,TERM,NK,LNODEN                                    631
      COMMON ORIG,TERM,NK,NA,LNODEN,TIME,CTIME,XNODE,AORD,LMIN,LMAX,         632
     *       TEST1                                                           633
      DIMENSION ORIG(1000),TERM(1000),AORD(1000),CTIME(1000),                634
     *XNODE(9999),ND(9999),NDD(9999),TIME(1000, 6),NK(1000)                  635
      N=LNODEN                                                               636
      M=NA                                                                   637
      NDD(1)=1                                                               638
      DO 5 I=2,N                                                             639
    5 NDD(I)=0                                                               640
      DO 6 I=1,M                                                             641
    6 AORD(I)=0                                                              642
      K=0                                                                    643
      MP=M+1                                                                 644
      DO 1 II=1,MP                                                           645
      DO 20 I=1,N                                                            646
   20 ND(I)=NDD(I)                                                           647
      III=0                                                                  648
      IP=II+1                                                                649
      DO 2 J=1,M                                                             650
      IF(ND(ORIG(J)).NE.II) GO TO 2                                          651
      NDD(TERM(J))=IP                                                        652
      III=1                                                                  653
      IF(K.EQ.0) GO TO 14                                                    654
      DO 10 L=1,K                                                            655
      IF(AORD(L).EQ.J) GO TO 11                                              656
   10 CONTINUE                                                               657
   14 K=K+1                                                                  658
      GO TO 13                                                               659
   11 IF(L.EQ.K) GO TO 2                                                     660
      KM=K-1                                                                 661
      DO 12 LL=L,KM                                                          662
   12 AORD(LL)=AORD(LL+1)                                                    663
   13 AORD(K)=J                                                              664
    2 CONTINUE                                                               665
      IF(III.EQ.0) GO TO 3                                                   666
    1 CONTINUE                                                               667
    3 CONTINUE                                                               668
      DO 30 I=1,NA                                                           669
   30 CTIME(I)=TIME(I,1)                                                     670
```

AD-A060 568 TEXAS A AND M UNIV COLLEGE STATION INST OF STATISTICS F/G 9/2
A USER'S GUIDE TO THE COMPUTER IMPLEMENTATION OF THE NEW PROJEC--ETC(U)
AUG 78 T C BAKER, R L SIELKEN N00014-78-C-0426
UNCLASSIFIED THEMIS-TR-57 NL

3 OF 4
AD A060568

```
      LMIN=CPTIME(CPATHT)                                                671
      DO 31 I=1,NA                                                       672
      NK1=NK(I)                                                          673
31    CTIME(I)=TIME(I,NK1)                                               674
      LMAX=CPTIME(CPATHT)                                                675
      RETURN                                                             676
      END                                                                677
      FUNCTION CPTIME(CPATHT)                                            678
C                                                                        679
C        DETERMINE THE CRITICAL PATH TIME: CPTIME                        680
C        XNODE(I) = EARLIEST TIME THAT AN ACTIVITY BEGINNING AT NODE I   681
C                 CAN COMMENCE                                           682
C                                                                        683
      IMPLICIT INTEGER*2(A-Z)                                            684
      INTEGER*4 NA,NN,ORIG,TERM,NK,LNODEN                                685
      COMMON ORIG,TERM,NK,NA,LNODEN,TIME,CTIME,XNODE,ACRD,LMIN,LMAX,     686
     *       TEST1                                                       687
      DIMENSION ORIG(1000),TERM(1000),ACRD(1000),CTIME(1000),            688
     *XNODE(9999),ND(9999),NDD(9999),TIME(1000, 6),NK(1000)              689
      DO 1 I=1,LNODEN                                                    690
1     XNODE(I)=0                                                         691
      DO 2 II=1,NA                                                       692
      I=ACRD(II)                                                         693
2     IF(XNODE(ORIG(I))+CTIME(I).GT.XNODE(TERM(I)))                      694
     1   XNODE(TERM(I))=XNODE(ORIG(I))+CTIME(I)                          695
      CPTIME=XNODE(TERM(II))                                             696
      II=ACRD(NA)                                                        697
      RETURN                                                             698
      END                                                                699
```

3. DSR

```
C      DETERMINISTIC SCHEDULE RESOLUTION PROGRAM                                1
C                                                                               2
C         THIS PROGRAM ASCERTAINS THE DISTRIBUTIONAL PARAMETERS FOR EACH        3
C         ACTIVITY IN THE PROJECT NETWORK THAT CORRESPOND TO THE MINIMUM        4
C         COST PROJECT SCHEDULE WHICH YIELDS A DETERMINISTIC COMPLETION         5
C         TIME OF  PD.   THIS IS DONE BY REFERRING TO THE OUTPUT FROM THE       6
C         DETERMINISTIC PROJECT SCHEDULER WHICH HAS BEEN STORED ON DISK.        7
C                                                                               8
C         THE ACTIVITY DISTRIBUTIONS ARE USED BY SIMPLIFICATION AND             9
C         INDIRECTLY BY SUBNETWORK ANALYSIS.                                   10
C                                                                              11
C         FOR THE SAKE OF IDENTIFYING THE APPROPRIATE DIMENSIONS, LET          12
C         MMAX = THE MAXIMUM NUMBER OF ACTIVITIES IN THE ORIGINAL              13
C                  PROJECT NETWORK                                             14
C         BKMAX = THE MAXIMUM NUMBER OF BREAK POINTS ALLOWED IN THE            15
C                  ENTIRE PROJECT'S TIME-COST CURVE                            16
C         NCTMAX = THE MAXIMUM NUMBER OF COMPLETION TIMES AND COSTS FOR        17
C                  EACH ACTIVITY                                               18
C         CURRENTLY, MMAX=1000; BKMAX=3000; NCTMAX=6                           19
C                                                                              20
C      INTEGER*2 BREAK(BKMAX),FILE6(2*MMAX),TYME(NCTMAX)                       21
C      DIMENSION TIME(4),NCT(MMAX),FILE0(4*MMAX+5),NUMS(MMAX)                  22
C      EQUIVALENCE (FILE0(3*MMAX+6),NCT(1))                                    23
C                                                                              24
       INTEGER*2 NBREAK,BREAK(3000),FILE6(2000),TYME(6)                        25
       REAL*8 AA,XM,BB,ALPHA,BETA                                              26
       DIMENSION TIME(4),NCT(1000),FILE0(4005),NUMS(1000)                      27
       EQUIVALENCE (FILE0(1),NA),(FILE0(6),NUMS(1)),(FILE0(3006),NCT(1))       28
       INTEGER F0/8/,F1/9/,F2/10/,        F4/12/,        F6/14/                29
       INTEGER NORM(5)/'NORM','AL',3*' '/,BET(5)/'BETA',4*' '/                 30
       INTEGER RECT(5)/'RECT','ANGU','LAR(','UNIF','ORM)'/                     31
       INTEGER FIXED(5)/'FIXE','D(CO','NSTA','NT)',' '/                        32
C                                                                              33
C      FOR EACH ACTIVITY--                                                     34
C     USES ORIGINAL TIME VECTOR AND RESULTS OF THE DETERMINISTIC PROJECT       35
C     SCHEDULER TO CREATE A NEW TIME VECTOR.  THIS DETERMINES THE ACTIV-       36
C     ITY DISTRIBUTIONS TO BE USED BY SIMPLIFICATION AND INDIRECTLY BY         37
C     SUBNETWORK ANALYSIS.                                                     38
C                                                                              39
       WRITE(6,1500)                                                           40
1500   FORMAT(1H1,132('*')/1X,132('*')/'0THIS IS THE OUTPUT FROM THE DETE      41
      *RMINISTIC SCHEDULE RESOLUTION PROGRAM: DSR'                              42
      *//1X,132('*')/1X,132('*'))                                               43
       REWIND F0                                                               44
       REWIND F1                                                               45
       REWIND F2                                                               46
       REWIND F4                                                               47
       REWIND F6                                                               48
       READ(F0)FILE0                                                           49
       READ(F1)                                                                50
       READ(F4)NCYC,PD,NFLAG                                                   51
       WRITE(6,2000)PD                                                         52
2000   FORMAT(1H0,/'0THE MINIMUM COST PROJECT SCHEDULE FOR A TARGET TIME       53
      *OF ',F12.5,' IS AS FOLLOWS:'//'0ACTIVITY:',8X,                 'MEA     54
      *N',8X,'DISTRIBUTION',14X,'PARAMETERS'/)                                 55
       READ(F0)                                                                56
       READ(F0)NBREAK,(BREAK(I),I=1,NBREAK)                                    57
       REWIND F6                                                               58
       DO 1 I=1,NBREAK                                                         59
       J=I                                                                     60
```

```
      IF(PD.GE.BREAK(J))GO TO 2                                          61
1     CONTINUE                                                           62
      NBB=BREAK(NBREAK)                                                  63
      WRITE(6,100)NBB,PD                                                 64
100   FORMAT(1H0,'THE SMALLEST POSSIBLE PROJECT COMPLETION TIME BASED ON 65
     * NON-RANDOM INDIVIDUAL ACTIVITY COMPLETION TIMES IS ',I10,'.'/     66
     *'0THE OPTIMAL ACTIVITY COMPLETION TIMES FOR A TARGET TIM','        67
     *E OF ',E14.5,' HAVE BEEN REQUESTED.')                             68
      IF(NFLAG.EQ.1)GO TO 300                                            69
      PD=NBB                                                             70
      WRITE(6,101)PD                                                     71
101   FORMAT(' THE OPTIMAL ACTIVITY COMPLETION TIMES FOR A TARGET',      72
     *  ' TIME OF',E14.5,' WILL BE SUPPLIED.')                           73
      NFLAG=1                                                            74
      REWIND F4                                                          75
      WRITE(F4)NCYC,PD,NFLAG                                             76
      D=0.0                                                              77
      I=NBREAK-1                                                         78
      GO TO 200                                                          79
300   WRITE(6,102)                                                       80
102   FORMAT('0THIS IS THE SECOND CONSECUTIVE REQUEST FOR AN INFEASIBLE  81
     *PROJECT DEADLINE TIME.'/' A FATAL ERROR WILL BE COMMITTED SO THAT  82
     *ALL REMAINING JOB STEPS WILL BE SKIPPED.')                        83
      CALL ERRSET(251,1,-1,1)                                            84
      XM=DSQRT(-1.D0)                                                    85
      GO TO 400                                                          86
2     NFLAG=0                                                            87
      D=0.D0                                                             88
      I=J-1                                                              89
      IF(I.EQ.0)GO TO 4                                                  90
      D=BREAK(I)-PD                                                      91
200   DO 3 J=1,I                                                         92
3     READ(F6)                                                           93
4     N1=2*NA                                                            94
      READ(F6)(FILE6(I),I=1,N1)                                          95
      DO 700 JJ=1,NA                                                     96
      J=JJ*2                                                             97
      EMA=FILE6(J)+D*FILE6(J-1)                                          98
      NNCT=NCT(JJ)                                                       99
      NNCT1=NNCT-1                                                      100
      READ(F1)(TYME(J),J=1,NNCT)                                        101
      DO 709 J=1,NNCT                                                   102
      IF(EMA.GE.TYME(J))GO TO 709                                       103
      JK=J-1                                                            104
      GO TO 708                                                         105
709   CONTINUE                                                          106
      JK=NNCT1                                                          107
708   DO 707 J=1,NNCT1                                                  108
      IF(J-JK)706,705,706                                               109
705   READ(F1)IDIST,TIME                                                110
      GO TO 707                                                         111
706   READ(F1)                                                          112
707   CONTINUE                                                          113
      MFIX=IDIST+2                                                      114
      GO TO (710,711,712,715),MFIX                                      115
710   CONTINUE                                                          116
C                                                                       117
C     BETA DISTRIBUTION                                                 118
C                                                                       119
      IF(TIME(4)   .GT.-1.D0)GO TO 7100                                 120
C                                                                       121
```

```
C       CONVERT THE MOST LIKELY PARAMETER, TIME(2), TO ALPHA AND BETA
C
        AA = TIME(1)
        XM=TIME(2)
        BB = TIME(3)
        CALL ALBET(AA,XM,BB,ALPHA,BETA)
        TIME(2)     = ALPHA
        TIME(4)    = BETA
7100    CONTINUE
        A     = TIME(2)     + 1.
        B     = TIME(4)     + 1.
        XMIN = TIME(1)
        XMAX = TIME(3)
C
C       COMPUTE NEW ALPHA,BETA,MIN TIME, AND MAX TIME
C
        UA = (XMAX-XMIN)*(A/(A+B)) + XMIN
        C  = (UA-XMIN)/(XMAX-UA)
        V = A*B/((A+B)**2*(A+B+1.))
        TIME(1)     = XMIN*EMA/UA
        TIME(4)     = (C/(V*(C+1)*(C+1))-1.)/(C+1)-1.
        TIME(3)     = XMAX*EMA/UA
        TIME(2)     = C*(TIME(4)    +1.)-1.
        WRITE(6,2101)NUMS(JJ),EMA,BET, TIME(1),TIME(3),TIME(2),TIME(4)
2101    FORMAT(I6,6X,F12.5,6X,5A4,6X,'MIN=',F13.5,', MAX=',F15.5,', ALPHA=
       *',F12.5,', BETA=',F12.5)
        GO TO 713
711     CONTINUE
C
C       NORMAL DISTRIBUTION
C
        IF (TIME(4)    .EQ.1.) GO TO 714
        TIME(3)    =(TIME(3)    -TIME(1)    )/6.
        TIME(4)     = 1.
        TIME(1)    =0.
714     TIME(3)     = EMA/TIME(2)    *TIME(3)
        TIME(2)     = EMA
        WRITE(6,2201)NUMS(JJ),EMA,NORM,TIME(2),TIME(3)
2201    FORMAT(I6,6X,F12.5,6X,5A4,6X,'MEAN=',F12.5,', ST.DEV=',F12.5)
        GO TO 713
712     CONTINUE
C
C       UNIFORM DISTRIBUTION
C
        TDUM = 2.*EMA*TIME(1)    /(TIME(1)    +TIME(3)    )
        TIME(3)     = 2.*EMA*TIME(3)    /(TIME(1)    +TIME(3)    )
        TIME(1)     = TDUM
        WRITE(6,2301)NUMS(JJ),EMA,RECT,TIME(1),TIME(3)
2301    FORMAT(I6,6X,F12.5,6X,5A4,6X,'MIN=',F13.5,', MAX=',F15.5)
        GO TO 713
715     WRITE(6,2401)NUMS(JJ),EMA,FIXED,TIME(1)
2401    FORMAT(I6,6X,F12.5,6X,5A4,6X,'TIME=',F12.5)
713     CONTINUE
        WRITE(F2)IDIST,TIME
700     CONTINUE
400     STOP
        END
        SUBROUTINE ALBET (AA,XM,BB,ALPHA,BETA)
C SUBROUTINE TO CALCULATE THE ALPHA AND BETA VALUES OF THE BETA DENSITY.
C N#6.SEE PAPER BY A.W.WORTHAM AND R.E.COLE,INDUSTRIAL ENGINEERING DEPT.
C
```

```
C      GIVEN AA, BB, AND XM THE VALUES OF ALPHA AND BETA ARE DETERMINED      183
C      FROM A AND THE STANDARD APPROXIMATION B.                              184
C                                                                            185
C       CALLING ARGUMENTS.                                                   186
C                                                                            187
C          1. AA IS THE MINIMUM TIME ESTIMATE.                               188
C          2.  XM IS THE MOST LIKLEY TIME ESTIMATE.                          189
C          3.  BB IS THE MAXIMUM TIME ESTIMATE.                              190
C          4. ALPHA AND BETA ARE THE VALUES CALCULATED                       191
C TERPOL IS A ARITHMETIC STATEMENT FUNCTION TO DO LINEAR INTERPOLATION.      192
C IF X IS THE VALUE REQUIRED WHICH LIES BETWEEN THE VALUES  OF X1 AND X2     193
C IN A TABLE% X2 IS THE LARGER VALUE< Y1 AND Y2 THE CORRESPONDING VALUES     194
C IN THE TABLE,%Y2 GREATER THAN Y GREATER THAN Y1<.                          195
      IMPLICIT REAL*8(A-H,O-Z)                                               196
      TERPOL(X1,X2,Y,Y1,Y2)=X1+((X2-X1)*(Y-Y1)/(Y2-Y1))                      197
      KK=-1                                                                  198
      K=0                                                                    199
      KA=AA*10.D0 + 0.1D0                                                    200
      KM=XM*10.D0 + 0.1D0                                                    201
      KB=BB*10.D0 + 0.1D0                                                    202
      KB=KB-KM                                                               203
      KA=KM-KA                                                               204
      IF(XM.EQ.BB) GOTO 22                                                   205
      IF(XM.EQ.AA) GOTO 21                                                   206
      GOTO 23                                                                207
   21 ALPHA=0.D0                                                             208
      BETA=2.8724D0                                                          209
      GOTO 20                                                                210
   22 ALPHA=2.8724D0                                                         211
      BETA=0.D0                                                              212
      GOTO 20                                                                213
   23 IF(KA-KB) 5,7,5                                                        214
    5 S=(XM-AA)/(BB-XM)                                                      215
      IF(S-1.D0) 6,7,8                                                       216
    7 ALPHA=3.D0                                                             217
      BETA=3.D0                                                              218
      GO TO 20                                                               219
    6 S=1.D0/S                                                               220
      K=-1                                                                   221
    8 Y=6.D0                                                                 222
      DO 10 I=1,600                                                          223
      Y=Y-1.D-2                                                              224
      B=Y*36.D0                                                              225
      C=(Y**3)+(7.D0*(Y**2))-20.D0*Y -24.D0                                  226
      R=(B**2)-(144.D0*C)                                                    227
      ROOT=DSQRT(R)                                                          228
      ALF=(B+ROOT)/72.D0                                                     229
      BET=(B-ROOT)/72.D0                                                     230
      SS=ALF/BET                                                             231
      IF (SS- S) 10,11,14                                                    232
   14 IF (KK)15,16,11                                                        233
   15 KK=KK+1                                                                234
      S1=SS                                                                  235
      X1=ALF                                                                 236
      Z1=BET                                                                 237
      GOTO 10                                                                238
   10 CONTINUE                                                               239
      WRITE  (6,18)                                                          240
   18 FORMAT(1H1, 32H ERROR,NO VALUE OF ALPHA BETA    )                      241
      CALL EXIT                                                              242
   16 KK=KK+1                                                                243
```

```
      X2=ALF                                                          244
      Z2=BET                                                          245
      S2=SS                                                           246
      ALF=TERPOL(X1,X2,S,S1,S2)                                       247
      BET=TERPOL(Z1,Z2,S,S1,S2)                                       248
11    IF (K) 12,13,13                                                 249
12    BETA =ALF                                                       250
      ALPHA = BET                                                     251
      GO TO 20                                                        252
13    ALPHA = ALF                                                     253
      BETA = BET                                                      254
20    RETURN                                                          255
      END                                                             256
```

4. SIMP

```
C      SIMPLIFICATION PROGRAM                                                  1
C                                                                              2
C         THIS PROGRAM DETERMINES THE SIMPLIFIED PROJECT NETWORK BY            3
C         REPLACING CERTAIN CONFIGURATIONS OF ACTIVITIES WITH SINGLE           4
C         ACTIVITIES WHOSE DISTRIBUTIONS CAN BE DETERMINED TO MATCH THE        5
C         DISTRIBUTIONS OF THE CONFIGURATIONS OF ACTIVITIES REPLACED.          6
C                                                                              7
C         FOR THE SAKE OF IDENTIFYING THE APPROPRIATE DIMENSIONS, LET          8
C         MMAX = THE MAXIMUM NUMBER OF ACTIVITIES IN THE ORIGINAL              9
C                   PROJECT NETWORK                                           10
C         CURRENTLY, MMAX=1000                                                11
C                                                                             12
C      INTEGER FILE3(3*NA),S(NA,4)                                            13
C                                                                             14
       IMPLICIT REAL*8(A-H,O-R,T-Z)                                          15
       INTEGER F0/8/,              F3/11/,                        F7/15/     16
       INTEGER          FILE3(3000),S(1000,4),LIST(24),JP(20)                17
       EQUIVALENCE (FILE3(1),S(1))                                           18
       EQUIVALENCE (ISBR,LIST(1)),(I,LIST(2)),(JP(1),LIST(3))                19
       WRITE(6,9000)                                                         20
9000   FORMAT(1H1,132('*')/1X,132('*')/'0THIS IS THE OUTPUT FROM THE SIMP    21
      *LIFICATION PROGRAM: SIMP'/1H0,132('*')/1X,132('*'))                    22
       REWIND F0                                                             23
       NRED=0                                                                24
       READ(F0)NACT,NODES,NSRCE,NSINK,LNODEN,S                               25
       READ(F0)JMAT                                                          26
       NK=NACT                                                               27
       REWIND F3                                                             28
       READ(F3)                                                              29
       DO 99 I=1,NK                                                          30
99     S(I,4)=1                                                              31
C                                                                            32
C                                                                            33
C         BUILD FILE7 FOR THE MODIFIED SIMPLIFICATION PROGRAM                34
C                                                                            35
C      START OF SEARCH FOR REDUCIBLE NETWORKS                                36
C                                                                            37
  100  DO 190 I=1,NK                                                         38
       IF(S(I,4).LE.0)  GO TO 190                                            39
       KP=0                                                                  40
       DO 121 J=1,NK                                                         41
       IF(I.NE.J.AND.S(J,4).GT.0.AND.S(I,2).EQ.S(J,2).AND.S(I,3).EQ.S(J,3    42
      1)) GO TO 122                                                          43
       GO TO 121                                                             44
  122  KP=KP+1                                                               45
       JP(KP)=J                                                              46
       IF(KP.GE.20) GO TO 200                                                47
  121  CONTINUE                                                              48
       IF(KP)120,120,200                                                     49
  120  DO 130 J=1,NK                                                         50
       IF(I.NE.J.AND.S(J,4).GT.0.AND.S(I,3).EQ.S(J,2)) GO TO 125             51
  130  CONTINUE                                                              52
       GO TO 190                                                             53
  125  J1=J                                                                  54
       JK=J1+1                                                               55
       IF(JK-NK)127,127,141                                                  56
  127  DO 140 J=JK,NK                                                        57
       IF(I.NE.J.AND.S(J,4).GT.0.AND.S(J,2).EQ.S(J1,2)) GO TO 160            58
  140  CONTINUE                                                              59
  141  DO 150 J=1,NK                                                         60
```

```
      IF(I.NE.J.AND.J1.NE.J.AND.S(J,4).GT.0.AND.S(J,3).EQ.S(I,3)) GO TO
     1190
  150 CONTINUE
      GO TO 300
  160 J2=J
      K3=-1
      K4=-1
      K5=-1
  161 DO 165 J=1,NK
      IF(I.EQ.J.OR.J1.EQ.J.OR.J2.EQ.J.OR.S(J,4).LE.0) GO TO 165
      IF(S(J,3).EQ.S(I,3))GO TO 190
  165 CONTINUE
C
C     CHECK FOR FIRST ORDER CROSSED NETWORKS
C
   49 KK=0
   50 DO 60 J=1, NK
      IF(I.EQ.J.OR.J.EQ.J1.OR.J.EQ.J2.OR.S(J,4).LE.0) GO TO 60
      IF(S(J,2).EQ.S(I,2).AND.S(J,3).EQ.S(J2,3)) GO TO 51
      GO TO 60
   51 J3=J
   52 K3=K3+1
      IF(K3) 60,60,190
   60 CONTINUE
      IF (K3) 61,62,190
   61 IF(KK) 53,53,62
   53 L=J1
      J1=J2
      J2=L
      KK=1
      GO TO 50
   62 DO 170 J=1,NK
      IF(I.EQ.J.OR.J.EQ.J1.OR.J.EQ.J2.OR.S(J,4).LE.0) GO TO 170
      IF(S(J,2).EQ.S(J2,3).AND.S(J,3).EQ.S(J1,3)) GO TO 167
      IF(S(J,2).EQ.S(J1,3).AND.S(J,3).EQ.S(J2,3)) GO TO 168
      GO TO 170
  167 J4=J
      K4=K4+1
      IF(K4)170,170,190
  168 J5=J
      K5=K5+1
      IF(K5)170,170,190
  170 CONTINUE
      IF(K3.EQ.0.AND.K4.EQ.0.AND.K5.EQ.-1)GO TO 180
      IF(K3.EQ.0.AND.K4.EQ.-1.AND.K5.EQ.0) GO TO 171
      GO TO 190
  171 K6=-1
      K7=-1
      DO 175 J=1,NK
      IF(J.EQ.I.OR.J.EQ.J1.OR.J.EQ.J3.OR.J.EQ.J5.OR.S(J,4).LE.0.OR.J.EQ.
     1J2)  GO TO 175
      IF(S(J,2).EQ.S(J1,3)) GO TO 172
      IF(S(J,2).EQ.S(J3,3)) GO TO 173
      IF(S(J,3).EQ.S(J1,3).OR.S(J,3).EQ.S(J3,3).OR.S(J,2).EQ.S(I,3)) GO
     1TO 190
      GO TO 175
  172 J6=J
      K6=K6+1
      IF(K6)175,175,190
  173 J7=J
      K7=K7+1
```

```
      IF(K7)175,175,190                                                  122
  175 CONTINUE                                                           123
      IF(K6.EQ.0.AND.K7.EQ.0) GO TO 176                                  124
      GO TO 190                                                          125
  176 IF(S(J6,3).EQ.S(J7,3)) GO TO 600                                   126
      GO TO 190                                                          127
  180 K8=-1                                                              128
      K9=-1                                                              129
      K12=-1                                                             130
      K13=-1                                                             131
      DO 185 J=1,NK                                                      132
      IF(J.EQ.I.OR.J.EQ.J1.OR.J.EQ.J2.OR.J.EQ.J3.OR.J.EQ.J4.OR.S(J,4).LE 133
     1.0) GO TO 185                                                      134
      IF(S(J,2).EQ.S(J3,3)) GO TO 181                                    135
      IF(S(J,3).EQ.S(J3,3)) GO TO 182                                    136
      IF(S(J,2).EQ.S(J4,3)) GO TO 183                                    137
      IF(S(J,2).EQ.S(I,3)) GO TO 184                                     138
      GO TO 185                                                          139
  181 J8=J                                                               140
      K8=K8+1                                                            141
      IF(K8)185,185,190                                                  142
  182 IF(S(J,2).EQ.S(J2,2)) GO TO 190                                    143
      J9=J                                                               144
      K9=K9+1                                                            145
      IF(K9)185,185,190                                                  146
  183 J12=J                                                              147
      K12=K12+1                                                          148
      GO TO 185                                                          149
  184 IF(S(J,3).EQ.S(J1,3)) GO TO 190                                    150
      J13=J                                                              151
      K13=K13+1                                                          152
      IF(K13)185,185,190                                                 153
  185 CONTINUE                                                           154
      IF(K13.EQ.0.AND.K12.EQ.0.AND.K8.EQ.-1.AND.K9.EQ.-1) GO TO 186      155
      IF(K13.EQ.-1.AND.K12.EQ.0.AND.K8.EQ.0.AND.K9.EQ.-1) GO TO 187      156
      IF(K13.EQ.-1.AND.K9.EQ.0.AND.K8.EQ.-1) GO TO 188                   157
      IF(K13.EQ.-1.AND.K9.EQ.-1.AND.K8.EQ.0) GO TO 189                   158
      IF(K13.EQ.-1.AND.K9.EQ.-1.AND.K8.EQ.-1) GO TO 900                  159
      GO TO 190                                                          160
C                                                                        161
C     CRISS CROSS 1                                                      162
C                                                                        163
  186 IF(S(J13,3).NE.S(J12,3)) GO TO 190                                 164
      DO 192 J=1,NK                                                      165
      IF(J.EQ.I.OR.J.EQ.J1.OR.J.EQ.J2.OR.J.EQ.J3.OR.J.EQ.J4.OR.J.EQ.J12. 166
     1OR.J.EQ.J13.OR.S(J,4).LE.0) GO TO 192                              167
      IF(S(J,3).EQ.S(J4,3)) GO TO 190                                    168
  192 CONTINUE                                                           169
       GO TO 400                                                         170
C                                                                        171
C     CRISS CROSS 2                                                      172
C                                                                        173
  187 IF(S(J8,3).NE.S(J12,3)) GO TO 190                                  174
       DO 193 J=1,NK                                                     175
       IF(J.EQ.I.OR.J.EQ.J1.OR.J.EQ.J2.OR.J.EQ.J3.OR.J.EQ.J4.OR.J.EQ.J8. 176
     1OR.J.EQ.J9.OR.S(J,4).LE.0) GO TO 193                               177
      IF(S(J,3).EQ.S(J4,3)) GO TO 190                                    178
  193 CONTINUE                                                           179
      GO TO 500                                                          180
C                                                                        181
C     DOUBLE WHEATSTONE BRIDGE 1                                         182
```

```
C                                                                       183
  188 K10=-1                                                            184
      K11=-1                                                            185
      DO 196 J=1,NK                                                     186
      IF(J.EQ.I.OR.J.EQ.J1.OR.J.EQ.J2.OR.J.EQ.J3.OR.J.EQ.J4.OR.J.EQ.J9) 187
     1GO TO 196                                                         188
      IF(S(J,3).EQ.S(J9,2)) GO TO 194                                   189
      IF(S(J,2).EQ.S(J9,2)) GO TO 195                                   190
      GO TO 196                                                         191
  194 IF(S(J,2).NE.S(J3,2)) GO TO 190                                   192
      J10=J                                                             193
      K10=K10+1                                                         194
      IF(K10)196,196,190                                                195
  195 IF(S(J,3).NE.S(J4,3)) GO TO 190                                   196
      J11=J                                                             197
      K11=K11+1                                                         198
      IF(K11)196,196,190                                                199
  196 CONTINUE                                                          200
      IF(K10.EQ.0.AND.K11.EQ.0) GO TO 700                               201
      GO TO 190                                                         202
C                                                                       203
C     DOUBLE WHEATSTONE BRIDGE 2                                        204
C                                                                       205
  189 K10=-1                                                            206
      K11=-1                                                            207
      DO 199 J=1,NK                                                     208
      IF(J.EQ.I.OR.J.EQ.J1.OR.J.EQ.J2.OR.J.EQ.J3.OR.J.EQ.J4.OR.J.EQ.J8 . 209
     1OR.S(J,4).LE.0) GO TO 199                                         210
      IF(S(J,3).EQ.S(J8,3)) GO TO 197                                   211
      IF(S(J,2).EQ.S(J8,3)) GO TO 198                                   212
      GO TO 199                                                         213
  197 IF(S(J,2).NE.S(J3,2)) GO TO 190                                   214
      J10=J                                                             215
      K10=K10+1                                                         216
      IF(K10)199,199,190                                                217
  198 IF(S(J,3).NE.S(J4,3)) GO TO 190                                   218
      J11=J                                                             219
      K11=K11+1                                                         220
      IF(K11)199,199,190                                                221
  199 CONTINUE                                                          222
      IF(K10.EQ.0.AND.K11.EQ.0) GO TO 800                               223
  190 CONTINUE                                                          224
      IF(NRED)1103,1103,91                                              225
   91 NRED = 0                                                          226
      GO TO 100                                                         227
C                                                                       228
C  BEGINNING OF PARALLEL REDUCTION SECTION                              229
C                                                                       230
  200 NRED=NRED+1                                                       231
      J1=JP(1)                                                          232
      DO 201 J=2,KP                                                     233
      L=JP(J)                                                           234
201   S(L,4)=-1                                                         235
      S(I,4)=-1                                                         236
      ISBR=1                                                            237
      LIST(23)=KP                                                       238
      LIST(24)=J1                                                       239
      WRITE(F7) LIST                                                    240
      IF (JMAT) 1040,1040,120                                           241
 1040 LIM=KP-1                                                          242
      DO 202 K=1,KP                                                     243
```

```
      IP=JP(K)                                                          244
  202 JP(K)=S(IP,1)                                                     245
      WRITE(6,1024) S(I,1),(JP(K),K=1,LIM)                              246
      WRITE(6,1023) JP(KP)                                              247
 1024 FORMAT(1H0, 10X11H ACTIVITIES I4,(10(2H, I4)/12X))                248
 1023 FORMAT(1H ,10X4HAND I4,32H HAVE BEEN COMBINED IN PARALLEL.     )  249
      GO TO 1290                                                        250
C                                                                       251
C     SERIES REDUCTION                                                  252
C                                                                       253
  300 NRED=NRED+1                                                       254
      S(I,4)=-1                                                         255
      ISER=2                                                            256
      LIST(3)=J1                                                        257
      WRITE(F7) LIST                                                    258
      IF (JMAT) 1009,1009,120                                           259
 1009 WRITE(6,1025) S(I,1),S(J1,1)                                      260
 1025 FORMAT(1H0,42X11H ACTIVITIES I4,4H AND I4,20H COMBINED IN SERIES.)261
      GO TO 1290                                                        262
C                                                                       263
C     CRISS CROSS TYPE 1 REDUCTION                                      264
C                                                                       265
  400 NRED=NRED+1                                                       266
      S(I,4)=-1                                                         267
      S(J2,4)=-1                                                        268
      S(J3,4)=-1                                                        269
      S(J4,4)=-1                                                        270
      S(J12,4)=-1                                                       271
      S(J13,4)=-1                                                       272
      ISBR=3                                                            273
      LIST(3)=J1                                                        274
      LIST(4)=J2                                                        275
      LIST(5)=J3                                                        276
      LIST(6)=J4                                                        277
      LIST(7)=J12                                                       278
      LIST(8)=J13                                                       279
      WRITE(F7) LIST                                                    280
      IF(JMAT) 1230,1230,190                                            281
 1230 WRITE(6,1430)S(I,1),S(J1,1),S(J2,1),S(J3,1),S(J4,1),S(J12,1),S(J13282
     1,1)                                                               283
 1430 FORMAT(1H0,25X11HACTIVITIES I4,2H, I4,2H, I4,2H, I4,2H, I4,2H, I4,284
     1  6H, AND I4,44H HAVE BEEN COMBINED AS A CRISS CROSS TYPE 1.)    285
      GO TO 1290                                                        286
C                                                                       287
C     CRISS CROSS TYPE 2 REDUCTION                                      288
C                                                                       289
  500 NRED=NRED+1                                                       290
      S(I,4)=-1                                                         291
      S(J2,4)=-1                                                        292
      S(J3,4)=-1                                                        293
      S(J4,4)=-1                                                        294
      S(J8,4)=-1                                                        295
      S(J12,4)=-1                                                       296
      ISBR=4                                                            297
      LIST(3)=J1                                                        298
      LIST(4)=J2                                                        299
      LIST(5)=J3                                                        300
      LIST(6)=J4                                                        301
      LIST(7)=J8                                                        302
      LIST(8)=J12                                                       303
      WRITE(F7) LIST                                                    304
```

```
      IF(JMAT) 1240,1240,190                                              305
 1240 WRITE(6,1440)S(I,1),S(J1,1),S(J2,1),S(J3,1),S(J4,1),S(J8,1),S(J12,  306
     11)                                                                  307
 1440 FORMAT(1H0,25X 11HACTIVITIES I4,2H, I4,2H, I4,2H, I4,2H, I4,2H, I4  308
     1, 6H, AND I4,44H HAVE BEEN COMBINED AS A CRISS CROSS TYPE 2.)       309
      GO TO 1290                                                          310
C                                                                         311
C     CRISS CROSS TYPE 3 REDUCTION                                        312
C                                                                         313
  600 NRED=NRED+1                                                         314
      S(I,4)=-1                                                           315
      S(J2,4)=-1                                                          316
      S(J3,4)=-1                                                          317
      S(J5,4)=-1                                                          318
      S(J6,4)=-1                                                          319
      S(J7,4)=-1                                                          320
      ISBR=5                                                              321
      LIST(3)=J1                                                          322
      LIST(4)=J2                                                          323
      LIST(5)=J3                                                          324
      LIST(6)=J5                                                          325
      LIST(7)=J6                                                          326
      LIST(8)=J7                                                          327
      WRITE(F7) LIST                                                      328
      IF(JMAT) 1250,1250,190                                              329
 1250 WRITE(6,1450)S(I,1),S(J1,1),S(J2,1),S(J3,1),S(J5,1),S(J6,1),S(J7,1  330
     1)                                                                   331
 1450 FORMAT(1H0,25X 11HACTIVITIES I4,2H, I4,2H, I4,2H, I4,2H, I4,2H, I4  332
     1, 6H, AND I4,44H HAVE BEEN COMBINED AS A CRISS CROSS TYPE 3.)       333
      GO TO 1290                                                          334
C                                                                         335
C     DOUBLE WHEATSTONE BRIDGE TYPE 1 REDUCTION                           336
C                                                                         337
  700 NRED=NRED+1                                                         338
      S(I,4)=-1                                                           339
      S(J2,4)=-1                                                          340
      S(J3,4)=-1                                                          341
      S(J4,4)=-1                                                          342
      S(J9,4)=-1                                                          343
      S(J10,4)=-1                                                         344
      S(J11,4)=-1                                                         345
      ISBR=6                                                              346
      LIST(3)=J1                                                          347
      LIST(4)=J2                                                          348
      LIST(5)=J3                                                          349
      LIST(6)=J4                                                          350
      LIST(7)=J9                                                          351
      LIST(8)=J10                                                         352
      LIST(9)=J11                                                         353
      WRITE(F7) LIST                                                      354
      IF(JMAT) 1210,1210,190                                              355
 1210 WRITE(6,1410)S(I,1),S(J1,1),S(J2,1),S(J3,1),S(J4,1),S(J9,1),S(J10,  356
     11),S(J11,1)                                                         357
 1410 FORMAT(1H0,10X 11HACTIVITIES I4,2H, I4,2H, I4,2H, I4,2H, I4,2H, I4  358
     1,2H, I4,6H, AND I4, 57H HAVE BEEN COMBINED AS A DOUBLE WHEATSTONE   359
     2BRIDGE TYPE 1.)                                                     360
      GO TC 1290                                                          361
C                                                                         362
C      DOUBLE WHEATSTONE BRIDGE TYPE 2 REDUCTION                          363
C                                                                         364
  800 NRED=NRED+1                                                         365
```

```
      S(I,4)=-1                                                          366
      S(J2,4)=-1                                                         367
      S(J3,4)=-1                                                         368
      S(J4,4)=-1                                                         369
      S(J8,4)=-1                                                         370
      S(J10,4)=-1                                                        371
      S(J11,4)=-1                                                        372
      ISER=7                                                             373
      LIST(3)=J1                                                         374
      LIST(4)=J2                                                         375
      LIST(5)=J3                                                         376
      LIST(6)=J4                                                         377
      LIST(7)=J8                                                         378
      LIST(8)=J10                                                        379
      LIST(9)=J11                                                        380
      WRITE(F7) LIST                                                     381
      IF(JMAT) 1220,1220,190                                             382
 1220 WRITE(6,1420)S(I,1),S(J1,1),S(J2,1),S(J3,1),S(J4,1),S(J8,1),S(J10,  383
     11),S(J11,1)                                                        384
 1420 FORMAT(1H0,10X 11HACTIVITIES I4,2H, I4,2H, I4,2H, I4,2H, I4,2H, I4  385
     1,2H, I4,6H, AND I4, 57H HAVE BEEN COMBINED AS A DOUBLE WHEATSTONE   386
     2BRIDGE TYPE 2.)                                                    387
      GO TO 1290                                                         388
C                                                                        389
C     WHEATSTONE BRIDGE REDUCTION                                        390
C                                                                        391
  900 IF (NRED.EQ.0) NRED=NRED+1                                         392
      S(I,4)=-1                                                          393
      S(J2,4)=-1                                                         394
      S(J3,4)=-1                                                         395
      S(J4,4)=-1                                                         396
      ISER=8                                                             397
      LIST(3)=J1                                                         398
      LIST(4)=J2                                                         399
      LIST(5)=J3                                                         400
      LIST(6)=J4                                                         401
      WRITE(F7) LIST                                                     402
      IF(JMAT) 1201,1201,190                                             403
 1201 WRITE(6,1401) S(I,1),S(J1,1),S(J2,1),S(J3,1),S(J4,1)               404
 1401 FORMAT(1H0,20X 11HACITIVITES I4,2H, I4,2H, I4,2H, I4,6H, AND I4, 4  405
     13H HAVE BEEN COMBINED AS A WHEATSTONE BRIDGE.)                     406
C                                                                        407
 1290 WRITE(6,1490) S(J1,1),S(J1,2),S(J1,3)                              408
 1490 FORMAT(1H ,41X, 9HACTIVITY ,I4,18H NOW HAS TAIL NO. I4,14H AND HEA  409
     1D NO. ,I4,1H.)                                                     410
      GO TO 190                                                          411
1103  WRITE(6,1027)                                                      412
1027  FORMAT('1',10X,'THE FOLLOWING ACTIVITIES CANNOT BE FURTHER COMBINE 413
     *D:')                                                               414
      WRITE(6,2001)                                                      415
2001  FORMAT(///11X,'ACTIVITY     ORIGIN     TERMINAL'                   416
     *5X,'ACTIVITY NUMBER ASSIGNED BY DECOMP')                           417
      NACT=0                                                             418
      DO 1105 I=1,NK                                                     419
      IF(S(I,4).LT.0)GO TO 1105                                          420
      NACT=NACT+1                                                        421
1104  WRITE(6,1028)S(I,1),S(I,2),S(I,3),NACT                             422
1028  FORMAT(13X,I4,6X,I5,7X,I5,9X,I5)                                   423
1105  CONTINUE                                                           424
      ENDFILE F7                                                         425
C                                                                        426
```

```
C         BUILD FILE3 FOR DECOMPOSITION                                  427
C                                                                        428
      NACT = 0                                                           429
      LNODEN = 0                                                         430
C                                                                        431
C     COUNT ACTIVITIES AND DETERMINE LARGEST NODE NUMBER                 432
C     DETERMINE L U AND P FOR SUBNETWORK ANALYSIS                        433
C                                                                        434
      DO 1153 J=1,NK                                                     435
C                                                                        436
C     S(J,4) IS -1 IF J-TH ACTIVITY HAS BEEN ELIMINATED                  437
C                                                                        438
      IF(S(J,4).EQ.-1)GO TO 1153                                         439
      NACT=NACT+1                                                        440
      S(NACT,1)=NACT                                                     441
      S(NACT,2)=S(J,2)                                                   442
      S(NACT,3)=S(J,3)                                                   443
      IF (S(NACT,3).GT.LNODEN) LNODEN = S(NACT,3)                        444
1153  CONTINUE                                                           445
C                                                                        446
C     FIND SOURCE AND SINK                                               447
C                                                                        448
      NSRCE = 0                                                          449
      NSINK = 0                                                          450
      DO 1154  J=1,NACT                                                  451
      IFLG = 0                                                           452
      JFLG  = 0                                                          453
      IIRG = S(J,2)                                                      454
      ITER = S(J,3)                                                      455
      DO 1156  K=1,NACT                                                  456
      IF (IIRG.EQ.S(K,3)) IFLG=1                                         457
      IF (ITER.EQ.S(K,2)) JFLG=1                                         458
1156  CONTINUE                                                           459
      IF (IFLG.EQ.0) NSRCE = S(J,2)                                      460
      IF (JFLG.EQ.0) NSINK = S(J,3)                                      461
      IF (NSRCE.NE.0.AND.NSINK.NE.0) GO TO 1157                          462
1154  CONTINUE                                                           463
1157  CONTINUE                                                           464
      WRITE(F3)NACT,NODES,NSRCE,NSINK,LNODEN,FILE3                       465
      STOP                                                               466
      END                                                                467
```

5. MODSIMP

```
C      MODIFIED SIMPLIFICATION PROGRAM                                          1
C                                                                               2
C         THIS PROGRAM DETERMINES THE DISTRIBUTION OF EACH OF THE               3
C         ACTIVITIES IN THE SIMPLIFIED PROJECT NETWORK (DETERMINED BY THE       4
C         SIMPLIFICATION PROGRAM).                                              5
C                                                                               6
C         FOR THE SAKE OF IDENTIFYING THE APPROPRIATE DIMENSIONS, LET           7
C         MMAX = THE MAXIMUM NUMBER OF ACTIVITIES IN THE ORIGINAL               8
C                     PROJECT NETWORK                                           9
C         CURRENTLY, MMAX=1000                                                 10
C                                                                              11
C      DIMENSION S(MMAX,4),CDF(MMAX,12)                                        12
C                                                                              13
       IMPLICIT REAL*8(A-H,O-R,T-Z)                                            14
       INTEGER F0/8/,      F2/10/,       F4/12/,              F7/15/           15
       COMMON /BLKA/S,CDF,A,DENS,TI                                            16
       INTEGER S                                                               17
       DIMENSION S(1000,4),CDF(1000,12),A(12),DENS(400),TI(2)                  18
       COMMON /BLKB/TIME,Y                                                     19
       DIMENSION TIME(4),Y(12)                                                 20
       REAL*4 TYME(4)                                                          21
       DIMENSION              JP(20),LIST(24)                                  22
       EQUIVALENCE (ISBR,LIST(1)),(I,LIST(2)),(JP(1),LIST(3))                  23
       WRITE(6,9000)                                                           24
9000   FORMAT(1H1,132('*')/1X,132('*')/'0THIS IS THE OUTPUT FROM THE MODI      25
      *FIED SIMPLIFICATION PROGRAM: MODSIMP'/1H0,132('*')/1X,132('*'))         26
       REWIND F0                                                               27
       REWIND F2                                                               28
       REWIND F7                                                               29
       A(1)=0.0                                                                30
       A(12)=1.00                                                              31
       DO 82 J=2,11                                                            32
    82 A(J)= DFLOAT(J-2)*0.1D0 + 0.05D0                                        33
       I500=0                                                                  34
       J200=0                                                                  35
       READ(F0)NACT,NODES,NSRCE,NSINK,LNODEN,S                                 36
       IFLOP = 0                                                               37
     5 DO 79 I=1,NACT                                                          38
       READ(F2)MM,TYME                                                         39
       MM=MM+2                                                                 40
       TIME(1) = TYME(1)                                                       41
       TIME(2) = TYME(2)                                                       42
       TIME(3) = TYME(3)                                                       43
       TIME(4) = TYME(4)                                                       44
       CALL XDENS(MM,I,I500,J200)                                              45
       S(I,4)=MM                                                               46
       DO 42 J=1,12                                                            47
42     CDF(I,J)=Y(J)                                                           48
    79 CONTINUE                                                                49
       NK=NACT                                                                 50
100    READ(F7,END=99) LIST                                                    51
       GO TO (200,300,400,500,600,700,800,900),ISBR                            52
C                                                                              53
200    CONTINUE                                                                54
C      PARALLEL REDUCTION                                                      55
       KP = LIST(23)                                                           56
       J1 = LIST(24)                                                           57
       CALL PARRED(I,JP,KP,J1)                                                 58
       GO TO 100                                                               59
300    CONTINUE                                                                60
```

```
C                                                                       61
C       SERIES REDUCTION                                                62
C                                                                       63
        J1 = LIST(3)                                                    64
        S(I,4)=-1                                                       65
        CALL SERRED(I,J1)                                               66
        GO TO 100                                                       67
400     CONTINUE                                                        68
C                                                                       69
C       CRISS CROSS TYPE 1 REDUCTION                                    70
C                                                                       71
        J1 = LIST(3)                                                    72
        J2 = LIST(4)                                                    73
        J3 = LIST(5)                                                    74
        J4 = LIST(6)                                                    75
        J12= LIST(7)                                                    76
        J13= LIST(8)                                                    77
        S(I,4)=-1                                                       78
        S(J2,4)=-1                                                      79
        S(J3,4)=-1                                                      80
        S(J4,4)=-1                                                      81
        S(J12,4)=-1                                                     82
        S(J13,4)=-1                                                     83
        CALL CC1RED(I,J1,J2,J3,J4,J12,J13)                              84
        GO TO 100                                                       85
500     CONTINUE                                                        86
C                                                                       87
C       CRISS CROSS TYPE 2 REDUCTION                                    88
C                                                                       89
        J1 = LIST(3)                                                    90
        J2 = LIST(4)                                                    91
        J3 = LIST(5)                                                    92
        J4 = LIST(6)                                                    93
        J8 = LIST(7)                                                    94
        J12= LIST(8)                                                    95
        S(I,4)=-1                                                       96
        S(J2,4)=-1                                                      97
        S(J3,4)=-1                                                      98
        S(J4,4)=-1                                                      99
        S(J8,4)=-1                                                     100
        S(J12,4)=-1                                                    101
        CALL CC2RED(I,J1,J2,J3,J4,J8,J12)                              102
        GO TO 100                                                      103
600     CONTINUE                                                       104
C                                                                      105
C       CRISS CROSS TYPE 3 REDUCTION                                   106
C                                                                      107
        J1 = LIST(3)                                                   108
        J2 = LIST(4)                                                   109
        J3 = LIST(5)                                                   110
        J5 = LIST(6)                                                   111
        J6 = LIST(7)                                                   112
        J7 = LIST(8)                                                   113
        S(I,4)=-1                                                      114
        S(J2,4)=-1                                                     115
        S(J3,4)=-1                                                     116
        S(J5,4)=-1                                                     117
        S(J6,4)=-1                                                     118
        S(J7,4)=-1                                                     119
        CALL CC3RED(I,J1,J2,J3,J5,J6,J7)                               120
        GO TO 100                                                      121
```

```
700   CONTINUE                                                          122
C                                                                       123
C     DOUBLE WHEATSTONE BRIDGE TYPE 1 REDUCTION                         124
C                                                                       125
      J1 = LIST(3)                                                      126
      J2 = LIST(4)                                                      127
      J3 = LIST(5)                                                      128
      J4 = LIST(6)                                                      129
      J9 = LIST(7)                                                      130
      J10= LIST(8)                                                      131
      J11= LIST(9)                                                      132
      S(I,4)=-1                                                         133
      S(J2,4)=-1                                                        134
      S(J3,4)=-1                                                        135
      S(J4,4)=-1                                                        136
      S(J9,4)=-1                                                        137
      S(J10,4)=-1                                                       138
      S(J11,4)=-1                                                       139
      CALL DW1RED(I,J1,J2,J3,J4,J9,J10,J11)                             140
      GO TO 100                                                         141
800   CONTINUE                                                          142
C                                                                       143
C      DOUBLE WHEATSTONE BRIDGE TYPE 2 REDUCTION                        144
C                                                                       145
      J1 = LIST(3)                                                      146
      J2 = LIST(4)                                                      147
      J3 = LIST(5)                                                      148
      J4 = LIST(6)                                                      149
      J8 = LIST(7)                                                      150
      J10= LIST(8)                                                      151
      J11= LIST(9)                                                      152
      S(I,4)=-1                                                         153
      S(J2,4)=-1                                                        154
      S(J3,4)=-1                                                        155
      S(J4,4)=-1                                                        156
      S(J8,4)=-1                                                        157
      S(J10,4)=-1                                                       158
      S(J11,4)=-1                                                       159
      CALL DW2RED(I,J1,J2,J3,J4,J8,J10,J11)                             160
      GO TO 100                                                         161
900   CONTINUE                                                          162
C                                                                       163
C     WHEATSTONE BRIDGE REDUCTION                                       164
C                                                                       165
      J1 = LIST(3)                                                      166
      J2 = LIST(4)                                                      167
      J3 = LIST(5)                                                      168
      J4 = LIST(6)                                                      169
      S(I,4)=-1                                                         170
      S(J2,4)=-1                                                        171
      S(J3,4)=-1                                                        172
      S(J4,4)=-1                                                        173
      CALL WBRED(I,J1,J2,J3,J4)                                         174
99    CONTINUE                                                          175
C     PRINT RESULTING DISTRIBUTIONS                                     176
      DO 1100 I=1,NACT                                                  177
      IF (S(I,4).LT.0) GO TO 1100                                       178
      TIME(1)=0.D0                                                      179
      TIME(3)=0.D0                                                      180
      DO 270 K=2,11                                                     181
      TIME(1)=TIME(1)+CDF(I,K)                                          182
```

```
270   TIME(3)=TIME(3)+CDF(I,K)**2                                          183
      TIME(1)=TIME(1)*.100                                                 184
      TIME(3)=DSQRT(DMAX1(0.D0,TIME(3)*.100-TIME(1)**2))                   185
      IFLOP = IFLOP + 1                                                    186
      IF (MOD(IFLOP,3).EQ.0) WRITE(6,2001)                                 187
2001  FORMAT(1H1)                                                          188
      WRITE(6,1028)S(I,1),S(I,2),S(I,3),TIME(1),TIME(3)                    189
1028  FORMAT(///11X,'ACTIVITY      ORIGIN      TERMINAL      MEAN      STAND  190
     *ARD DEVIATION'/13X,I4,6X,I5,7X,I5,6X,F6.1, 8X,F7.2//32X,'F(T)        191
     * T')                                                                 192
      DO 1102 MP=1,12                                                      193
      WRITE(6,6000) A(MP),CDF(I,MP)                                        194
6000  FORMAT(31XF5.2,3H --  F10.4)                                         195
1102  CONTINUE                                                             196
1100  CONTINUE                                                             197
C                                                                          198
C     SAVE THE SIMPLIFIED ACTIVITY DURATION DISTRIBUTIONS FOR SUBNETWORK   199
C     ANALYSIS                                                             200
C                                                                          201
      REWIND F4                                                            202
      READ(F4)                                                             203
      NACT=0                                                               204
      DO 11002 J=1,NK                                                      205
      IF(S(J,4).GE.0)NACT=NACT+1                                           206
11002 CONTINUE                                                             207
      WRITE(F4) NACT                                                       208
      DO 1153 J=1,NK                                                       209
      IF(S(J,4).GT.0)WRITE(F4)(CDF(J,K),K=1,12)                            210
1153  CONTINUE                                                             211
      ENDFILE F4                                                           212
C     END OF MAIN PROGRAM                                                  213
C                                                                          214
      STOP                                                                 215
      END                                                                  216
      SUBROUTINE PARRED(I,JP,KP,J1)                                        217
      IMPLICIT REAL*8(A-H,O-R,T-Z)                                         218
      COMMON /BLKA/S,CDF,A,DENS,TI                                         219
      INTEGER S                                                            220
      DIMENSION S(1000,4),CDF(1000,12),A(12),DENS(400),TI(2)               221
      DIMENSION JP(20)                                                     222
      TI(1)=CDF(I,1)                                                       223
      TI(2)=CDF(I,12)                                                      224
      J1=JP(1)                                                             225
      DO 201 J=1,KP                                                        226
      L=JP(J)                                                              227
      IF(CDF(L,1).GT.TI(1)) TI(1)=CDF(L,1)                                 228
      IF(CDF(L,12).GT.TI(2)) TI(2)=CDF(L,12)                               229
      IF(L.EQ.J1) GO TO 201                                                230
      S(L,4)=-1                                                            231
  201 CONTINUE                                                             232
      S(I,4)=-1                                                            233
  213 L=I                                                                  234
      KM=10                                                                235
      K=1                                                                  236
      DELTAT=(TI(2)-TI(1))/10.0D0                                          237
      MP=0                                                                 238
      M=1                                                                  239
  217 DO 220 J=2,10                                                        240
      K=K+1                                                                241
      T=TI(1)+DFLOAT(J-1)*DELTAT                                           242
      IF(T.GE.CDF(L,12)) GO TO 221                                         243
```

```
  218 IF(T.GE.CDF(L,M).AND.T.LT.CDF(L,M+1)) GO TO 219                    244
      M=M+1                                                              245
      GO TO 218                                                          246
  219 IF(M.NE.1) GO TO260                                                247
      T1=CDF(L ,1)                                                       248
      T2=CDF(L ,2)                                                       249
      T3=CDF(L ,3)                                                       250
      Y1=0.D0                                                            251
      Y2=0.05D0                                                          252
      Y3=0.15D0                                                          253
      GO TO 261                                                          254
  260 T1=CDF(L,M)                                                        255
      T2=CDF(L,M+1)                                                      256
      T3=CDF(L,M-1)                                                      257
      Y1=A(M)                                                            258
      Y2=A(M+1)                                                          259
      Y3=A(M-1)                                                          260
  261 DENS(K)=TERPOL(Y1,Y2,Y3,T,T1,T2,T3)                                261
      IF(DENS(K).LT.0.D0) DENS(K)=0.D0                                   262
      IF(DENS(K).GE.1.D0) GO TO 221                                      263
  220 CONTINUE                                                           264
      GO TO 231                                                          265
  221 KO=K                                                               266
      DO 230 K=KO,KM                                                     267
  230 DENS(K)=1.D0                                                       268
  231 MP=MP+1                                                            269
      IF(MP.GT.KP) GO TO 240                                             270
      L=JP(MP)                                                           271
      K=KM                                                               272
      M=1                                                                273
      KM=KM+9                                                            274
      GO TO 217                                                          275
  240 CONTINUE                                                           276
      DO 251 M=2,10                                                      277
      DO 250 J=1,KP                                                      278
      L=M+J*9                                                            279
  250 DENS(M)=DENS(M)*DENS(L)                                            280
  251 CONTINUE                                                           281
      DENS(1)=0.0D0                                                      282
      DENS(11)=1.0D0                                                     283
      CALL INVSTR(J1,DELTAT)                                             284
      RETURN                                                             285
      END                                                                286
      SUBROUTINE SERRED(I,J1)                                            287
      IMPLICIT REAL*8(A-H,O-R,T-Z)                                       288
      COMMON /BLKA/S,CDF,A,DENS,TI                                       289
      INTEGER S                                                          290
      DIMENSION S(1000,4),CDF(1000,12),A(12),DENS(400),TI(2)             291
      TI(1)=CDF(I,1)+CDF(J1,1)                                           292
      TI(2)=CDF(I,12)+CDF(J1,12)                                         293
      IF(CDF(J1,12).EQ.CDF(J1,1)) GO TO 370                              294
      IF(CDF(I,12).EQ.CDF(I ,1)) GO TO 376                               295
      DENS(1)=0.D0                                                       296
      DENS(11)=1.0D0                                                     297
      TEMPO=TI(1)                                                        298
      DELTAT=(TI(2)-TI(1))/10.0D0                                        299
      DO 399 J=2,10                                                      300
      K1=1                                                               301
      T=TEMPO+DFLOAT(J-1)*DELTAT                                         302
      FOT=0.0D0                                                          303
      L1=11                                                              304
```

```
  311 K1=K1+1                                                          305
      IF(K1.GE.12) GO TO 390                                           306
      TDIF=T-CDF(I ,K1)                                                307
      IF(TDIF.GE.CDF(J1,12)) GO TO 350                                 308
  312 IF (TDIF.GT.CDF(J1,LI)) GO TO 340                                309
      LI=LI-1                                                          310
      IF(1-LI)312,312,390                                              311
  340 IF (LI .NE. 1) GO TO 360                                         312
      T1=CDF(J1,1)                                                     313
      T2=CDF(J1,2)                                                     314
      T3=CDF(J1,3)                                                     315
      Y1=0.D0                                                          316
      Y2=0.05D0                                                        317
      Y3=0.15D0                                                        318
      GO TO 361                                                        319
  360 T1=CDF(J1,LI)                                                    320
      T2=CDF(J1,LI+1)                                                  321
      T3=CDF(J1,LI-1)                                                  322
      Y1=A(LI)                                                         323
      Y2=A(LI+1)                                                       324
      Y3=A(LI-1)                                                       325
  361 FOT=FOT+TERPOL(Y1,Y2,Y3,TDIF,T1,T2,T3)                           326
      GO TO 311                                                        327
  350 FOT=FOT+1.D0                                                     328
      GO TO 311                                                        329
  390 DENS(J)=0.1D0*FOT                                                330
      IF(DENS(J).GE.1.D0) GO TO 380                                    331
      IF(DENS(J).GT.0.D0) GO TO 399                                    332
      DENS(J)=0.D0                                                     333
  395 CONTINUE                                                         334
      GO TO 382                                                        335
  380 DO 381 JJ=J,10                                                   336
      DENS(JJ)=1.D0                                                    337
  381 CONTINUE                                                         338
  382 S(J1,2)=S(I,2)                                                   339
  365 CONTINUE                                                         340
      CALL INVSTR(J1,DELTAT)                                           341
      GO TO 379                                                        342
  370 DO  372 JJ=1,12                                                  343
  372 CDF(J1,JJ)= CDF(I,JJ) + CDF(J1,12)                               344
      S(J1,2)=S(I,2)                                                   345
      GO TO 379                                                        346
  376 DO 378 JJ=1,12                                                   347
  378 CDF(J1,JJ)=CDF(J1,JJ) + CDF(I,12)                                348
      S(J1,2)= S(I,2)                                                  349
  379 RETURN                                                           350
      END                                                              351
      SUBROUTINE CC1RED(I,J1,J2,J3,J4,J12,J13)                         352
      IMPLICIT REAL*8(A-H,O-R,T-Z)                                     353
      COMMON /BLKA/S,CDF,A,DENS,TI                                     354
      INTEGER S                                                        355
      DIMENSION S(1000,4),CDF(1000,12),A(12),DENS(400),TI(2)           356
      DO 406 J=1,2                                                     357
      L=1+(J-1)*11                                                     358
      T1=CDF(J3,L)+CDF(J4,L)+CDF(J12,L)                                359
      T2=CDF(I,L)+CDF(J2,L)+CDF(J12,L)+CDF(J4,L)                       360
      T3=CDF(I,L)+CDF(J1,L)+CDF(J12,L)                                 361
      T4=CDF(I,L)+CDF(J13,L)                                           362
      IF(T1.GE.T2.AND.T1.GE.T3.AND.T1.GE.T4) GO TO 403                 363
      IF(T2.GE.T1.AND.T2.GE.T3.AND.T2.GE.T4) GO TO 404                 364
      IF(T3.GE.T1.AND.T3.GE.T2.AND.T3.GE.T4) GO TO 405                 365
```

```
      TI(J)=T4                                                          366
      GO TO 406                                                         367
  403 TI(J)=T1                                                          368
      GO TO 406                                                         369
  404 TI(J)=T2                                                          370
      GO TO 406                                                         371
  405 TI(J)=T3                                                          372
  406 CONTINUE                                                          373
      DELTAT=(TI(2)-TI(1))/10.0D0                                       374
      DENS(1)=0.D0                                                      375
      DENS(11)=1.000                                                    376
      DO 470 J=2,10                                                     377
      T=TI(1) + DFLOAT(J-1)*DELTAT                                      378
      KI=1                                                              379
      L13=11                                                            380
      FCT=0.0                                                           381
  410 KI=KI+1                                                           382
      IF(KI.GE.12) GO TO 460                                            383
      TDIF13=T-CDF(I,KI)                                                384
      IF(TDIF13.GE.CDF(J13,12)) GO TO 412                               385
  411 IF(TDIF13.GT.CDF(J13,L13)) GO TO 413                              386
      L13=L13-1                                                         387
      IF(1-L13)411,411,460                                              388
  412 F13=1.D0                                                          389
      GO TO 415                                                         390
  413 IF (L13.NE. 1) GO TO 491                                          391
      T1=CDF(J13,1)                                                     392
      T2=CDF(J13,2)                                                     393
      T3=CDF(J13,3)                                                     394
      Y1=0.D0                                                           395
      Y2=0.05D0                                                         396
      Y3=0.15D0                                                         397
      GO TO 414                                                         398
  491 T1=CDF(J13,L13)                                                   399
      T2=CDF(J13,L13+1)                                                 400
      T3=CDF(J13,L13-1)                                                 401
      Y1=A(L13)                                                         402
      Y2=A(L13+1)                                                       403
      Y3=A(L13-1)                                                       404
  414 F13=TERPOL(Y1,Y2,Y3,TDIF13,T1,T2,T3)                              405
      IF(F13.LT.0.D0)F13=0.D0                                           406
      IF(F13.GT.1.D0)F13=1.D0                                           407
  415 K12=1                                                             408
      L1=11                                                             409
      FCT2=0.D0                                                         410
  416 K12=K12+1                                                         411
      IF(K12.LT.12) GO TO 418                                           412
  417 FCT=FCT+FCT2*F13                                                  413
      GO TO 410                                                         414
  418 TDIF1=TDIF13-CDF(J12,K12)                                         415
      IF(TDIF1.GE.CDF(J1,12)) GO TO 420                                 416
  419 IF(TDIF1.GT.CDF(J1,L1)) GO TO 421                                 417
      L1=L1-1                                                           418
      IF(1-L1)419,419,417                                               419
  420 F1=1.D0                                                           420
      GO TO 422                                                         421
  421 IF (L1 .NE. 1) GO TO 482                                          422
      T1=CDF(J1,1)                                                      423
      T2=CDF(J1,2)                                                      424
      T3=CDF(J1,3)                                                      425
      Y1=0.D0                                                           426
```

```
      Y2=0.05D0                                                          427
      Y3=0.15D0                                                          428
      GO TO 427                                                          429
  482 T1=CDF(J1,L1)                                                      430
      T2=CDF(J1,L1+1)                                                    431
      T3=CDF(J1,L1-1)                                                    432
      Y1=A(L1)                                                           433
      Y2=A(L1+1)                                                         434
      Y3=A(L1-1)                                                         435
  427 F1=TERPOL(Y1, Y2, Y3, TDIF1, T1, T2, T3)                           436
      IF(F1.LT.0.D0) F1=0.D0                                             437
      IF(F1.GT.1.D0) F1=1.D0                                             438
  422 K4=1                                                               439
      L3=11                                                              440
      L2=11                                                              441
      FOT1=0.D0                                                          442
  423 K4=K4+1                                                            443
      IF(K4.LT.12) GO TO 425                                             444
  424 FOT2=FOT2+F1*FOT1                                                  445
      GO TO 416                                                          446
  425 TDIF3=T-CDF(J4,K4)-CDF(J12,K12)                                    447
      TDIF2=TDIF3-CDF(I,K1)                                              448
      IF(TDIF2.GE.CDF(J2,12)) GO TO 430                                  449
  426 IF(TDIF2.GT.CDF(J2,L2)) GO TO 431                                  450
      L2=L2-1                                                            451
      IF(1-L2)426,426,424                                                452
  430 F2=1.D0                                                            453
      GO TO 435                                                          454
  431 IF (L2 .NE. 1) GO TO 433                                           455
      T1=CDF(J2,1)                                                       456
      T2=CDF(J2,2)                                                       457
      T3=CDF(J2,3)                                                       458
      Y1=0.D0                                                            459
      Y2=0.05D0                                                          460
      Y3=0.15D0                                                          461
      GO TO 434                                                          462
  433 T1=CDF(J2,L2)                                                      463
      T2=CDF(J2,L2+1)                                                    464
      T3=CDF(J2,L2-1)                                                    465
      Y1=A(L2)                                                           466
      Y2=A(L2+1)                                                         467
      Y3=A(L2-1)                                                         468
  434 F2=TERPOL(Y1, Y2, Y3, TDIF2, T1, T2, T3)                           469
      IF(F2 .LT.0.D0) F2=0.D0                                            470
      IF(F2 .GT.1.D0) F2=1.D0                                            471
  435 IF(TDIF3.GE.CDF(J3,12)) GO TO 440                                  472
  436 IF(TDIF3.GT.CDF(J3,L3)) GO TO 445                                  473
      L3=L3-1                                                            474
      IF(1-L3)436,436,424                                                475
  440 F3=1.D0                                                            476
      GO TO 450                                                          477
  445 IF (L3 .NE. 1) GO TO 447                                           478
      T1=CDF(J3,1)                                                       479
      T2=CDF(J3,2)                                                       480
      T3=CDF(J3,3)                                                       481
      Y1=0.D0                                                            482
      Y2=0.05D0                                                          483
      Y3=0.15D0                                                          484
      GO TO 449                                                          485
  447 T1=CDF(J3,L3)                                                      486
      T2=CDF(J3,L3+1)                                                    487
```

```
      T3=CDF(J3,L3-1)                                                    488
      Y1=A(L3)                                                           489
      Y2=A(L3+1)                                                         490
      Y3=A(L3-1)                                                         491
  449 F3=TERPOL(Y1, Y2, Y3, TDIF3, T1, T2, T3)                           492
      IF(F3 .LT.0.D0) F3=0.D0                                            493
      IF(F3.GT.1.D0) F3=1.D0                                             494
  450 FCT1=FCT1+F2*F3                                                    495
      GC TC 423                                                          496
  460 DENS(J)=FCT*0.1D-2                                                 497
      IF(DENS(J).LT.0.D0) DENS(J)=0.D0                                   498
      IF(DENS(J).GE.1.D0) GO TO 480                                      499
  470 CONTINUE                                                           500
  480 DC 481 JJ=J,10                                                     501
      DENS(JJ)=1.D0                                                      502
  481 CONTINUE                                                           503
      S(J1,3)=S(J13,3)                                                   504
      S(J1,2)=S(I,2)                                                     505
      CALL INVSTR(J1,DELTAT)                                             506
      RETURN                                                             507
      END                                                                508
      SUBROUTINE CC2RED(I,J1,J2,J3,J4,J8,J12)                            509
      IMPLICIT REAL*8(A-H,O-R,T-Z)                                       510
      COMMON /BLKA/S,CDF,A,DENS,TI                                       511
      INTEGER S                                                          512
      DIMENSION S(1000,4),CDF(1000,12),A(12),DENS(400),TI(2)             513
      DO 510 J=1,2                                                       514
      L=1+(J-1)*11                                                       515
      T1=CDF(I,L)+CDF(J1,L)+CDF(J12,L)                                   516
      T2=CDF(I,L)+CDF(J2,L)+CDF(J4,L)+CDF(J12,L)                         517
      T3=CDF(I,L)+CDF(J2,L)+CDF(J8,L)                                    518
      T4=CDF(J3,L)+CDF(J4,L)+CDF(J12,L)                                  519
      T5=CDF(J3,L)+CDF(J8,L)                                             520
      IF(T1.GE.T2.AND.T1.GE.T3.AND.T1.GE.T4.AND.T1.GE.T5) GO TO 501      521
      IF(T2.GE.T1.AND.T2.GE.T3.AND.T2.GE.T4.AND.T2.GE.T5) GO TO 502      522
      IF(T3.GE.T1.AND.T3.GE.T2.AND.T3.GE.T4.AND.T3.GE.T5) GO TO 503      523
      IF(T4.GE.T1.AND.T4.GE.T2.AND.T4.GE.T3.AND.T4.GE.T5) GO TO 504      524
      TI(J)=T5                                                           525
      GO TO 510                                                          526
  501 TI(J)=T1                                                           527
      GC TO 510                                                          528
  502 TI(J)=T2                                                           529
      GO TO 510                                                          530
  503 TI(J)=T3                                                           531
      GO TO 510                                                          532
  504 TI(J)=T4                                                           533
  510 CONTINUE                                                           534
      DELTAT=(TI(2)-TI(1))/10.0D0                                        535
      DENS(1)=0.D0                                                       536
      DENS(11)=1.0D0                                                     537
      DO 570 J=2,10                                                      538
      T=TI(1) + DFLOAT(J-1)*DELTAT                                       539
      KI=1                                                               540
      FOT=0.D0                                                           541
  511 KI=KI+1                                                            542
      IF(KI.GE.12) GO TO 560                                             543
      K12=1                                                              544
      L1=11                                                              545
      FOT3=0.D0                                                          546
  512 K12=K12+1                                                          547
      IF(K12.LT.12) GO TO 514                                            548
```

```
513 FCT=FCT+FCT3                                                    549
    GO TO 511                                                       550
514 TDIF1=T-CDF(1,K1)-CDF(J12,K12)                                  551
    IF(TDIF1.GE.CDF(J1,12)) GO TO 517                               552
515 IF(TDIF1.GT.CDF(J1,L1)) GO TO 518                               553
    L1=L1-1                                                         554
    IF(1-L1)515,515,513                                             555
517 F1=1.D0                                                         556
    GO TO 519                                                       557
518 IF (L1 .NE. 1) GO TO 971                                        558
    T1=CDF(J1,1)                                                    559
    T2=CDF(J1,2)                                                    560
    T3=CDF(J1,3)                                                    561
    Y1=0.D0                                                         562
    Y2=0.05D0                                                       563
    Y3=0.15D0                                                       564
    GO TO 972                                                       565
971 T1=CDF(J1,L1)                                                   566
    T2=CDF(J1,L1+1)                                                 567
    T3=CDF(J1,L1-1)                                                 568
    Y1=A(L1)                                                        569
    Y2=A(L1+1)                                                      570
    Y3=A(L1-1)                                                      571
972 F1=TERPOL(Y1, Y2, Y3, TDIF1, T1, T2, T3)                        572
    IF(F1.LT.0.D0) F1=0.D0                                          573
    IF(F1.GT.1.D0) F1=1.D0                                          574
519 K3=1                                                            575
    FCT2=0.D0                                                       576
520 K3=K3+1                                                         577
    IF(K3.LT.12) GO TO 521                                          578
    FCT3=FCT3+F1*FCT2                                               579
    GO TO 512                                                       580
521 K2=1                                                            581
    FCT1=0.D0                                                       582
    L4=11                                                           583
    L8=11                                                           584
522 K2=K2+1                                                         585
    IF(K2.LT.12) GO TO 526                                          586
525 FCT2=FCT2+FCT1                                                  587
    GO TO 520                                                       588
526 T1=T-CDF(J3,K3)                                                 589
    T2=T-CDF(1,K1)-CDF(J2,K2)                                       590
    IF(T2-T1)528,527,527                                            591
528 TDIF8=T2                                                        592
    GO TO 530                                                       593
527 TDIF8=T1                                                        594
    TDIF4=TDIF8-CDF(J12,K12)                                        595
530 IF(TDIF4.GE.CDF(J4,12)) GO TO 535                               596
531 IF(TDIF4.GT.CDF(J4,L4)) GO TO 540                               597
    L4=L4-1                                                         598
535 F4=1.D0                                                         599
    GO TO 541                                                       600
540 IF (L4 .NE. 1) GO TO 973                                        601
    T1=CDF(J4,1)                                                    602
    T2=CDF(J4,2)                                                    603
    T3=CDF(J4,3)                                                    604
    Y1=0.D0                                                         605
    Y2=0.05D0                                                       606
    Y3=0.15D0                                                       607
    GO TO 974                                                       608
973 T1=CDF(J4,L4)                                                   609
```

```
      T2=CDF(J4,L4+1)                                                    610
      T3=CDF(J4,L4-1)                                                    611
      Y1=A(L4)                                                           612
      Y2=A(L4+1)                                                         613
      Y3=A(L4-1)                                                         614
  974 F4=TERPOL(Y1, Y2, Y3, TDIF4, T1, T2, T3)                           615
      IF(F4 .LT.0.D0) F4=0.D0                                            616
      IF(F4 .GT.1.D0) F4=1.D0                                            617
  541 IF(TDIF8.GE.CDF(J8,12)) GO TO 545                                  618
  542 IF(TDIF8.GT.CDF(J8,L8)) GO TO 550                                  619
      L8=L8-1                                                            620
      IF(1-L8) 542,542,525                                               621
  545 F8=1.D0                                                            622
      GO TO 551                                                          623
  550 IF (L8 .NE. 1) GO TO 975                                           624
      T1=CDF(J8,1)                                                       625
      T2=CDF(J8,2)                                                       626
      T3=CDF(J8,3)                                                       627
      Y1=0.D0                                                            628
      Y2=0.05D0                                                          629
      Y3=0.15D0                                                          630
      GO TO 976                                                          631
  975 T1=CDF(J8,L8)                                                      632
      T2=CDF(J8,L8+1)                                                    633
      T3=CDF(J8,L8-1)                                                    634
      Y1=A(L8)                                                           635
      Y2=A(L8+1)                                                         636
      Y3=A(L8-1)                                                         637
  976 F8=TERPOL(Y1, Y2, Y3, TDIF8, T1, T2, T3)                           638
      IF(F8 .LT.0.D0) F8=0.D0                                            639
      IF(F8. GT.1.D0) F8=1.D0                                            640
  551 FOT1=FOT1+F4*F8                                                    641
      GO TO 522                                                          642
  560 DENS(J)=FOT*0.1D-3                                                 643
      IF(DENS(J).LT.0.D0) DENS(J)=0.D0                                   644
      IF(DENS(J).GE.1.D0) GO TO 580                                      645
  570 CONTINUE                                                           646
  580 DO 581 JJ=J,10                                                     647
      DENS(JJ)=1.D0                                                      648
  581 CONTINUE                                                           649
      S(J1,2)=S(I,2)                                                     650
      S(J1,3)=S(J12,3)                                                   651
      CALL INVSTR(J1,DELTAT)                                             652
      RETURN                                                             653
      END                                                                654
      SUBROUTINE COJRED(I,J1,J2,J3,J5,J6,J7)                             655
      IMPLICIT REAL*8(A-H,O-R,T-Z)                                       656
      COMMON /BLKA/S,CDF,A,DENS,TI                                       657
      INTEGER S                                                          658
      DIMENSION S(1000,4),CDF(1000,12),A(12),DENS(400),TI(2)             659
      DO 610 J=1,2                                                       660
      L=1+(J-1)*11                                                       661
      T1=CDF(I,L)+CDF(J1,L)+CDF(J6,L)                                    662
      T2=CDF(I,L)+CDF(J2,L)+CDF(J7,L)                                    663
      T3=CDF(I,L)+CDF(J1,L)+CDF(J5,L)+CDF(J7,L)                          664
      T4=CDF(J3,L)+CDF(J7,L)                                             665
      IF(T1.GE.T2.AND.T1.GE.T3.AND.T1.GE.T4) GO TO 602                   666
      IF(T2.GE.T1.AND.T2.GE.T3.AND.T2.GE.T4) GO TO 603                   667
      IF(T3.GE.T1.AND.T3.GE.T2.AND.T3.GE.T4) GO TO 604                   668
      TI(J)=T4                                                           669
      GO TO 610                                                          670
```

```
602 TI(J)=T1                                                          671
    GO TO 610                                                         672
603 TI(J)=T2                                                          673
    GO TO 610                                                         674
604 TI(J)=T3                                                          675
    GO TO 610                                                         676
610 CONTINUE                                                          677
    DELTAT=(TI(2)-TI(1))/10.0D0                                       678
    DENS(1)=0.D0                                                      679
    DENS(11)=1.0D0                                                    680
    DO 670 J=2,10                                                     681
    T=TI(1) + DFLOAT(J-1)*DELTAT                                      682
    K7=1                                                              683
    L3=11                                                             684
    FCT=0.D0                                                          685
615 K7=K7+1                                                           686
    IF(K7.GE.12) GO TO 660                                            687
    TDIF3=T-CDF(J7,K7)                                                688
    IF(TDIF3.GE.CDF(J3,12)) GO TO 625                                 689
620 IF(TDIF3.GT.CDF(J3,L3)) GO TO 622                                 690
    L3=L3-1                                                           691
    IF(1-L3)620,620,660                                               692
622 IF (L3 .NE. 1) GO TO 671                                          693
    T1=CDF(J3,1)                                                      694
    T2=CDF(J3,2)                                                      695
    T3=CDF(J3,3)                                                      696
    Y1=0.D0                                                           697
    Y2=0.05D0                                                         698
    Y3=0.15D0                                                         699
    GO TO 672                                                         700
671 T1=CDF(J3,L3)                                                     701
    T2=CDF(J3,L3+1)                                                   702
    T3=CDF(J3,L3-1)                                                   703
    Y1=A(L3)                                                          704
    Y2=A(L3+1)                                                        705
    Y3=A(L3-1)                                                        706
672 F3=TERPOL(Y1, Y2, Y3, TDIF3, T1, T2, T3)                          707
    IF(F3 .LT.0.D0) F3=0.D0                                           708
    IF(F3.GT.1.D0) F3=1.D0                                            709
    GO TO 626                                                         710
625 F3=1.D0                                                           711
626 KI=1                                                              712
    L2=11                                                             713
    FOT2=0.D0                                                         714
627 KI=KI+1                                                           715
    IF(KI.LT.12) GO TO 630                                            716
628 FOT=FOT+F3*FOT2                                                   717
    GO TO 615                                                         718
630 TDIF2=TDIF3-CDF(I,KI)                                             719
    IF(TDIF2.GE.CDF(J2,12)) GO TO 635                                 720
631 IF(TDIF2.GT.CDF(J2,L2)) GO TO 636                                 721
    L2=L2-1                                                           722
    IF(1-L2)631,631,628                                               723
635 F2=1.D0                                                           724
    GO TO 637                                                         725
636 IF (L2 .NE. 1) GO TO 673                                          726
    T1=CDF(J2,1)                                                      727
    T2=CDF(J2,2)                                                      728
    T3=CDF(J2,3)                                                      729
    Y1=0.D0                                                           730
    Y2=0.05D0                                                         731
```

```
      Y3=0.15D0                                                          732
      GO TO 674                                                          733
  673 T1=CDF(J2,L2)                                                      734
      T2=CDF(J2,L2+1)                                                    735
      T3=CDF(J2,L2-1)                                                    736
      Y1=A(L2)                                                           737
      Y2=A(L2+1)                                                         738
      Y3=A(L2-1)                                                         739
  674 F2=TERPOL(Y1, Y2, Y3, TDIF2, T1, T2, T3)                           740
      IF(F2 .LT.0.D0) F2=0.D0                                            741
      IF(F2 .GT.1.D0) F2=1.D0                                            742
  637 K1=1                                                               743
      FOT1=0.D0                                                          744
      L5=11                                                              745
      L6=11                                                              746
  640 K1=K1+1                                                            747
      IF(K1.LT.12) GO TO 645                                             748
  641 FOT2=FOT2+F2*FOT1                                                  749
      GO TO 627                                                          750
  645 TDIF5=TDIF2-CDF(J1,K1)                                             751
      TDIF6=T-CDF(1,K1)-CDF(J1,K1)                                       752
      IF(TDIF5.GE.CDF(J5,12)) GO TO 650                                  753
  646 IF(TDIF5.GT.CDF(J5,L5)) GO TO 647                                  754
      L5=L5-1                                                            755
      IF(1-L5)646,646,641                                                756
  647 IF (L5 .NE. 1) GO TO 675                                           757
      T1=CDF(J5,1)                                                       758
      T2=CDF(J5,2)                                                       759
      T3=CDF(J5,3)                                                       760
      Y1=0.D0                                                            761
      Y2=0.05D0                                                          762
      Y3=0.15D0                                                          763
      GO TO 676                                                          764
  675 T1=CDF(J5,L5)                                                      765
      T2=CDF(J5,L5+1)                                                    766
      T3=CDF(J5,L5-1)                                                    767
      Y1=A(L5)                                                           768
      Y2=A(L5+1)                                                         769
      Y3=A(L5-1)                                                         770
  676 F5=TERPOL(Y1, Y2, Y3, TDIF5, T1, T2, T3)                           771
      IF(F5.LT.0.D0) F5=0.D0                                             772
      IF(F5.GT.1.D0) F5=1.D0                                             773
      GO TO 651                                                          774
  650 F5=1.D0                                                            775
  651 IF(TDIF6.GE.CDF(J6,12)) GO TO 655                                  776
  652 IF(TDIF6.GT.CDF(J6,L6)) GO TO 656                                  777
      L6=L6-1                                                            778
      IF(1-L6)652,652,641                                                779
  655 F6=1.D0                                                            780
      GO TO 659                                                          781
  656 IF (L6 .NE. 1) GO TO 677                                           782
      T1=CDF(J6,1)                                                       783
      T2=CDF(J6,2)                                                       784
      T3=CDF(J6,3)                                                       785
      Y1=0.D0                                                            786
      Y2=0.05D0                                                          787
      Y3=0.15D0                                                          788
      GO TO 678                                                          789
  677 T1=CDF(J6,L6)                                                      790
      T2=CDF(J6,L6+1)                                                    791
      T3=CDF(J6,L6-1)                                                    792
```

```
      Y1=A(L6)                                                          793
      Y2=A(L6+1)                                                        794
      Y3=A(L6-1)                                                        795
  678 F6=TERPCL(Y1, Y2, Y3, TDIF6, T1, T2, T3)                          796
      IF(F6.LT.0.D0)F6=0.D0                                             797
      IF(F6.GT.1.D0) F6=1.D0                                            798
  659 FOT1=FOT1+F5*F6                                                   799
      GO TO 640                                                         800
  660 DENS(J)= FOT*0.1D-2                                               801
      IF(DENS(J).LT.0.D0) DENS(J)=0.D0                                  802
      IF(DENS(J).GE.1.D0) GO TO 680                                     803
  670 CONTINUE                                                          804
  680 DO 681 JJ=J,10                                                    805
      DENS(JJ)=1.D0                                                     806
  681 CONTINUE                                                          807
      S(J1,2)=S(I,2)                                                    808
      S(J1,3)=S(J6,3)                                                   809
      CALL INVSTR(J1,DELTAT)                                            810
      RETURN                                                            811
      END                                                               812
      SUBROUTINE CWIRED(I,J1,J2,J3,J4,J9,J10,J11)                       813
      IMPLICIT REAL*8(A-H,O-R,T-Z)                                      814
      COMMON /BLKA/S,CDF,A,DENS,TI                                      815
      INTEGER S                                                         816
      DIMENSION S(1000,4),CDF(1000,12),A(12),DENS(400),TI(2)            817
      DO 710 J=1,2                                                      818
      L=1+(J-1)*11                                                      819
      T1=CDF(I,L)+CDF(J1,L)                                             820
      T2=CDF(I,L)+CDF(J2,L)+CDF(J4,L)                                   821
      T3=CDF(J10,L)+CDF(J11,L)                                          822
      T4=CDF(J9,L)+CDF(J10,L)+CDF(J4,L)                                 823
      T5=CDF(J3,L)+CDF(J4,L)                                            824
      IF(T1.GE.T2.AND.T1.GE.T3.AND.T1.GE.T4.AND.T1.GE.T5) GO TO 701     825
      IF(T2.GE.T1.AND.T2.GE.T3.AND.T2.GE.T4.AND.T2.GE.T5)GO TO 702      826
      IF(T3.GE.T1.AND.T3.GE.T2.AND.T3.GE.T4.AND.T3.GE.T5) GO TO 703     827
      IF(T4.GE.T1.AND.T4.GE.T2.AND.T4.GE.T3.AND.T4.GE.T5) GO TO 704     828
      TI(J)=T5                                                          829
      GO TO 710                                                         830
  701 TI(J)=T1                                                          831
      GO TO 710                                                         832
  702 TI(J)=T2                                                          833
      GO TO 710                                                         834
  703 TI(J)=T3                                                          835
      GO TO 710                                                         836
  704 TI(J)=T4                                                          837
  710 CONTINUE                                                          838
      DENS(1)=0.D0                                                      839
      DENS(11)=1.0D0                                                    840
      DELTAT=(TI(2)-TI(1))/10.0D0                                       841
      DO 790 J=2,10                                                     842
      T=TI(1) + DFLOAT(J-1)*DELTAT                                      843
      K4=1                                                              844
      FOT=0.D0                                                          845
      L3=11                                                             846
  715 K4=K4+1                                                           847
      IF(K4.GE.12) GO TO 785                                            848
      TDIF3=T-CDF(J4,K4)                                                849
      IF(TDIF3.GE.CDF(J3,12)) GO TO 720                                 850
  716 IF(TDIF3.GT.CDF(J3,L3)) GO TO 717                                 851
      L3=L3-1                                                           852
      IF(1-L3)716,716,785                                               853
```

```
717 IF (L3 .NE. 1) GO TO 791                                          854
    T1=CDF(J3,1)                                                      855
    T2=CDF(J3,2)                                                      856
    T3=CDF(J3,3)                                                      857
    Y1=0.D0                                                           858
    Y2=0.05D0                                                         859
    Y3=0.15D0                                                         860
    GO TO 792                                                         861
791 T1=CDF(J3,L3)                                                     862
    T2=CDF(J3,L3+1)                                                   863
    T3=CDF(J3,L3-1)                                                   864
    Y1=A(L3)                                                          865
    Y2=A(L3+1)                                                        866
    Y3=A(L3-1)                                                        867
792 F3=TERPOL(Y1, Y2, Y3, TDIF3, T1, T2, T3)                          868
    IF(F3 .LT.0.D0) F3=0.D0                                           869
    IF(F3.GT.1.D0) F3=1.D0                                            870
    GO TO 721                                                         871
720 F3=1.D0                                                           872
721 KI=1                                                              873
    FOT1=0.D0                                                         874
    L2=11                                                             875
    L1=11                                                             876
725 KI=KI+1                                                           877
    IF(KI.LT.12) GO TO 730                                            878
726 FOT=FOT+FOT1*F3                                                   879
    GO TO 715                                                         880
730 TDIF2=TDIF3-CDF(I,KI)                                             881
    TDIF1=T-CDF(I,KI)                                                 882
    IF(TDIF2.GE.CDF(J2,12)) GO TO 732                                 883
731 IF(TDIF2.GT.CDF(J2,L2)) GO TO 733                                 884
    L2=L2-1                                                           885
    IF(1-L2)731,731,726                                               886
732 F2=1.D0                                                           887
    GO TO 735                                                         888
733 IF (L2 .NE. 1) GO TO 793                                          889
    T1=CDF(J2,1)                                                      890
    T2=CDF(J2,2)                                                      891
    T3=CDF(J2,3)                                                      892
    Y1=0.D0                                                           893
    Y2=0.05D0                                                         894
    Y3=0.15D0                                                         895
    GO TO 974                                                         896
793 T1=CDF(J2,L2)                                                     897
    T2=CDF(J2,L2+1)                                                   898
    T3=CDF(J2,L2-1)                                                   899
    Y1=A(L2)                                                          900
    Y2=A(L2+1)                                                        901
    Y3=A(L2-1)                                                        902
974 F2=TERPOL(Y1, Y2, Y3, TDIF2, T1, T2, T3)                          903
    IF(F2 .LT.0.D0) F2=0.D0                                           904
    IF(F2 .GT.1.D0) F2=1.D0                                           905
735 IF(TDIF1.GE.CDF(J1,12)) GO TO 737                                 906
736 IF(TDIF1.GT.CDF(J1,L1)) GO TO 738                                 907
    L1=L1-1                                                           908
    IF(1-L1)736,736,726                                               909
737 F1=1.D0                                                           910
    GO TO 740                                                         911
738 IF (L1 .NE. 1) GO TO 795                                          912
    T1=CDF(J1,1)                                                      913
    T2=CDF(J1,2)                                                      914
```

```
      T3=CDF(J1,3)                                                      915
      Y1=0.D0                                                           916
      Y2=0.05D0                                                         917
      Y3=0.15D0                                                         918
      GO TO 796                                                         919
  795 T1=CDF(J1,L1)                                                     920
      T2=CDF(J1,L1+1)                                                   921
      T3=CDF(J1,L1-1)                                                   922
      Y1=A(L1)                                                          923
      Y2=A(L1+1)                                                        924
      Y3=A(L1-1)                                                        925
  796 F1=TERPOL(Y1, Y2, Y3, TDIF1, T1, T2, T3)                          926
      IF(F1.LT.0.D0) F1=0.D0                                            927
      IF(F1.GT.1.D0) F1=1.D0                                            928
  740 K10=1                                                             929
      FCT2=0.D0                                                         930
      L9=11                                                             931
      L11=11                                                            932
  741 K10=K10+1                                                         933
      IF(K10.LT.12) GO TO 743                                           934
  742 FCT1=FOT1+F2*F1*FOT2                                              935
      GO TO 725                                                         936
  743 TDIF9=TDIF3-CDF(J10,K10)                                          937
      TDIF11=T-CDF(J10,K10)                                             938
      IF(TDIF9.GE.CDF(J9,12)) GO TO 747                                 939
  744 IF(TDIF9.GT.CDF(J9,L9)) GO TO 748                                 940
      L9=L9-1                                                           941
      IF(1-L9)744,744,742                                               942
  747 F9=1.D0                                                           943
      GO TO 750                                                         944
  748 IF (L9 .NE. 1) GO TO 797                                          945
      T1=CDF(J9,1)                                                      946
      T2=CDF(J9,2)                                                      947
      T3=CDF(J9,3)                                                      948
      Y1=0.D0                                                           949
      Y2=0.05D0                                                         950
      Y3=0.15D0                                                         951
      GO TO 798                                                         952
  797 T1=CDF(J9,L9)                                                     953
      T2=CDF(J9,L9+1)                                                   954
      T3=CDF(J9,L9-1)                                                   955
      Y1=A(L9)                                                          956
      Y2=A(L9+1)                                                        957
      Y3=A(L9-1)                                                        958
  798 F9=TERPOL(Y1, Y2, Y3, TDIF9, T1, T2, T3)                          959
      IF(F9.LT.0.D0) F9=0.D0                                            960
      IF(F9.GT.1.D0) F9=1.D0                                            961
  750 IF(TDIF11.GE.CDF(J11,12)) GO TO 755                               962
  751 IF(TDIF11.GT.CDF(J11,L11)) GO TO 756                              963
      L11=L11-1                                                         964
      IF(1-L11)751,751,742                                              965
  755 F11=1.D0                                                          966
      GO TO 760                                                         967
  756 IF (L11.NE. 1) GO TO 799                                          968
      T1=CDF(J11,1)                                                     969
      T2=CDF(J11,2)                                                     970
      T3=CDF(J11,3)                                                     971
      Y1=0.D0                                                           972
      Y2=0.05D0                                                         973
      Y3=0.15D0                                                         974
      GO TO 800                                                         975
```

```
799 T1=CDF(J11,L11)                                                    976
    T2=CDF(J11,L11+1)                                                  977
    T3=CDF(J11,L11-1)                                                  978
    Y1=A(L11)                                                          979
    Y2=A(L11+1)                                                        980
    Y3=A(L11-1)                                                        981
800  F11=TERPOL(Y1, Y2, Y3, TDIF11, T1, T2, T3)                        982
    IF(F11.LT.0.D0) F11=0.D0                                           983
    IF(F11.GT.1.D0) F11=1.D0                                           984
760 FOT2=FOT2+F9*F11                                                   985
    GO TO 741                                                          986
785 DENS(J)= FOT*0.1D-2                                                987
    IF(DENS(J).LT.0.D0) DENS(J)=0.D0                                   988
    IF(DENS(J).GE.1.D0) GO TO 780                                      989
790 CONTINUE                                                           990
780 DO 781 JJ=J,10                                                     991
    DENS(JJ)=1.D0                                                      992
781 CONTINUE                                                           993
    S(J1,2)=S(I,2)                                                     994
    CALL INVSTR(J1,DELTAT)                                             995
    RETURN                                                             996
    END                                                                997
    SUBROUTINE D#2RED(I,J1,J2,J3,J4,J8,J10,J11)                        998
    IMPLICIT REAL*8(A-H,O-R,T-Z)                                       999
    COMMON /BLKA/S,CDF,A,DENS,TI                                      1000
    INTEGER S                                                         1001
    DIMENSION S(1000,4),CDF(1000,12),A(12),DENS(400),TI(2)            1002
    DO 810 J=1,2                                                      1003
    L=1+(J-1)*11                                                      1004
    T1=CDF(I,L)+CDF(J1,L)                                             1005
    T2=CDF(J2,L)+CDF(I,L)+CDF(J4,L)                                   1006
    T3=CDF(I,L)+CDF(J2,L)+CDF(J8,L)+CDF(J11,L)                        1007
    T4=CDF(J3,L)+CDF(J4,L)                                            1008
    T5=CDF(J3,L)+CDF(J8,L)+CDF(J11,L)                                 1009
    T6=CDF(J10,L)+CDF(J11,L)                                          1010
    IF(T1.GE.T2.AND.T1.GE.T3.AND.T1.GE.T4.AND.T1.GE.T5.AND.T1.GE.T6)  1011
   1GO TO 801                                                         1012
    IF(T2.GE.T1.AND.T2.GE.T3.AND.T2.GE.T4.AND.T2.GE.T5.AND.T2.GE.T6)  1013
   1GO TO 802                                                         1014
    IF(T3.GE.T1.AND.T3.GE.T2.AND.T3.GE.T4.AND.T3.GE.T5.AND.T3.GE.T6)  1015
   1GO TO 803                                                         1016
    IF(T4.GE.T1.AND.T4.GE.T2.AND.T4.GE.T3.AND.T4.GE.T5.AND.T4.GE.T6)  1017
   1GO TO 804                                                         1018
    IF(T5.GE.T1.AND.T5.GE.T2.AND.T5.GE.T3.AND.T5.GE.T4.AND.T5.GE.T6)  1019
   1GO TO 805                                                         1020
    TI(J)=T6                                                          1021
    GO TO 810                                                         1022
801 TI(J)=T1                                                          1023
    GO TO 810                                                         1024
802 TI(J)=T2                                                          1025
    GO TO 810                                                         1026
803 TI(J)=T3                                                          1027
    GO TO 810                                                         1028
804 TI(J)=T4                                                          1029
    GO TO 810                                                         1030
805 TI(J)=T5                                                          1031
810 CONTINUE                                                          1032
    DENS(1)=0.D0                                                      1033
    DENS(11)=1.0D0                                                    1034
    DELTAT=(TI(2)-TI(1))/10.0D0                                       1035
    DO 890 J=2,10                                                     1036
```

```
      T=TI(I) + DFLOAT(J-1)*DELTAT                                    1037
      K11=1                                                           1038
      L10=11                                                          1039
      FOT=0.D0                                                        1040
  815 K11=K11+1                                                       1041
      IF(K11.GE.12) GO TO 885                                         1042
      TDIF10=T-CDF(J11,K11)                                           1043
      IF(TDIF10.GE.CDF(J10,12)) GO TO 817                             1044
  816 IF(TDIF10.GT.CDF(J10,L10)) GO TO 818                            1045
      L10=L10-1                                                       1046
      IF(1-L10)816,816,885                                            1047
  817 F10=1.D0                                                        1048
      GO TO 820                                                       1049
  818 IF (L10.NE. 1) GO TO 791                                        1050
      T1=CDF(J10,1)                                                   1051
      T2=CDF(J10,2)                                                   1052
      T3=CDF(J10,3)                                                   1053
      Y1=0.D0                                                         1054
      Y2=0.05D0                                                       1055
      Y3=0.15D0                                                       1056
      GO TO 792                                                       1057
  791 T1=CDF(J10,L10)                                                 1058
      T2=CDF(J10,L10+1)                                               1059
      T3=CDF(J10,L10-1)                                               1060
      Y1=A(L10)                                                       1061
      Y2=A(L10+1)                                                     1062
      Y3=A(L10-1)                                                     1063
  792  F10=TERPOL(Y1, Y2, Y3, TDIF10, T1, T2, T3)                     1064
      IF(F10.LT.0.D0)F10=0.D0                                         1065
      IF(F10.GT.1.D0) F10=1.D0                                        1066
  820 KI=1                                                            1067
      FOT3=0.D0                                                       1068
      L1=11                                                           1069
  821 KI=KI+1                                                         1070
      IF(KI.LT.12) GO TO 823                                          1071
  822 FCT=FCT+F10*FOT3                                                1072
      GO TO 815                                                       1073
  823 TDIF1=T-CDF(I,KI)                                               1074
      IF(TDIF1.GE.CDF(J1,12)) GO TO 825                               1075
  824 IF(TDIF1.GT.CDF(J1,L1)) GO TO 826                               1076
      L1=L1-1                                                         1077
      IF(1-L1)824,824,822                                             1078
  825 F1=1.D0                                                         1079
      GO TO 830                                                       1080
  826 IF (L1 .NE. 1) GO TO 893                                        1081
      T1=CDF(J1,1)                                                    1082
      T2=CDF(J1,2)                                                    1083
      T3=CDF(J1,3)                                                    1084
      Y1=0.D0                                                         1085
      Y2=0.05D0                                                       1086
      Y3=0.15D0                                                       1087
      GO TO 894                                                       1088
  893 T1=CDF(J1,L1)                                                   1089
      T2=CDF(J1,L1+1)                                                 1090
      T3=CDF(J1,L1-1)                                                 1091
      Y1=A(L1)                                                        1092
      Y2=A(L1+1)                                                      1093
      Y3=A(L1-1)                                                      1094
  894 F1=TERPOL(Y1, Y2, Y3, TDIF1, T1, T2, T3)                        1095
      IF(F1.LT.0.D0) F1=0.D0                                          1096
      IF(F1.GT.1.D0) F1=1.D0                                          1097
```

```
830 K8=1                                                                1098
    FOT2=0.D0                                                           1099
831 K8=K8+1                                                             1100
    IF(K8.LT.12) GO TO 832                                              1101
    FOT3=FOT3+F1*FOT2                                                   1102
    GO TO 821                                                           1103
832 K4=1                                                                1104
    FOT1=0.D0                                                           1105
    L2=11                                                               1106
    L3=11                                                               1107
833 K4=K4+1                                                             1108
    IF(K4.GE.12) GO TO 840                                              1109
    T1=T-CDF(J4,K4)                                                     1110
    T2=TDIF10-CDF(J8,K8)                                                1111
    IF(T1-T2)834,835,835                                                1112
834 TDIF3=T1                                                            1113
    GO TO 836                                                           1114
835 TDIF3=T2                                                            1115
836 TDIF2=TDIF3-CDF(I,KI)                                               1116
    IF(TDIF2.GE.CDF(J2,12)) GO TO 838                                   1117
837 IF(TDIF2.GT.CDF(J2,L2)) GO TO 839                                   1118
    L2=L2-1                                                             1119
    IF(1-L2)837,837,840                                                 1120
840 FOT2=FOT2+FOT1                                                      1121
    GO TO 831                                                           1122
838 F2=1.D0                                                             1123
    GO TO 845                                                           1124
839 IF (L2 .NE. 1) GO TO 895                                            1125
    T1=CDF(J2,1)                                                        1126
    T2=CDF(J2,2)                                                        1127
    T3=CDF(J2,3)                                                        1128
    Y1=0.D0                                                             1129
    Y2=0.05D0                                                           1130
    Y3=0.15D0                                                           1131
    GO TO 896                                                           1132
895 T1=CDF(J2,L2)                                                       1133
    T2=CDF(J2,L2+1)                                                     1134
    T3=CDF(J2,L2-1)                                                     1135
    Y1=A(L2)                                                            1136
    Y2=A(L2+1)                                                          1137
    Y3=A(L2-1)                                                          1138
896 F2=TERPOL(Y1, Y2, Y3, TDIF2, T1, T2, T3)                            1139
    IF(F2 .LT.0.D0) F2=0.D0                                             1140
    IF(F2 .GT.1.D0) F2=1.D0                                             1141
845 IF(TDIF3.GE.CDF(J3,12)) GO TO 850                                   1142
846 IF(TDIF3.GT.CDF(J3,L3)) GO TO 851                                   1143
    L3=L3-1                                                             1144
    IF(1-L3)846,846,840                                                 1145
850 F3=1.D0                                                             1146
    GO TO 860                                                           1147
851 IF (L3 .NE. 1) GO TO 897                                            1148
    T1=CDF(J3,1)                                                        1149
    T2=CDF(J3,2)                                                        1150
    T3=CDF(J3,3)                                                        1151
    Y1=0.D0                                                             1152
    Y2=0.05D0                                                           1153
    Y3=0.15D0                                                           1154
    GO TO 898                                                           1155
897 T1=CDF(J3,L3)                                                       1156
    T2=CDF(J3,L3+1)                                                     1157
    T3=CDF(J3,L3-1)                                                     1158
```

```
      Y1=A(L3)                                                          1159
      Y2=A(L3+1)                                                        1160
      Y3=A(L3-1)                                                        1161
  898 F3=TERPOL(Y1, Y2, Y3, TDIF3, T1, T2, T3)                          1162
      IF(F3 .LT.0.D0) F3=0.D0                                           1163
      IF(F3.GT.1.D0) F3=1.D0                                            1164
  860 FOT1=FOT1+F3*F2                                                   1165
      GO TO 833                                                         1166
  885 DENS(J)= FOT*0.1D-3                                               1167
      IF(DENS(J).LT.0.D0) DENS(J)=0.D0                                  1168
      IF(DENS(J).GE.1.D0) GO TO 880                                     1169
  890 CONTINUE                                                          1170
  880 DO 881 JJ=J,10                                                    1171
      DENS(JJ)=1.D0                                                     1172
  881 CONTINUE                                                          1173
      S(J1,2)=S(I,2)                                                    1174
      CALL INVSTR(J1,DELTAT)                                            1175
      RETURN                                                            1176
      END                                                               1177
      SUBROUTINE WBRED(I,J1,J2,J3,J4)                                   1178
      IMPLICIT REAL*8(A-H,O-R,T-Z)                                      1179
      COMMON /BLKA/S,CDF,A,DENS,TI                                      1180
      INTEGER S                                                         1181
      DIMENSION S(1000,4),CDF(1000,12),A(12),DENS(400),TI(2)            1182
C     *     *     *     *     *     *     *     *     *     *     *     1183
C                                                                       1184
C                                                                       1185
      DO 910 J=1,2                                                      1186
      L=1+(J-1)*11                                                      1187
      T1=CDF(I,L)+CDF(J1,L)                                             1188
      T2=CDF(I,L)+CDF(J2,L)+CDF(J4,L)                                   1189
      T3=CDF(J3,L)+CDF(J4,L)                                            1190
      IF(T1-T2) 901,901,902                                             1191
  901 IF(T2-T3) 903,903,904                                             1192
  902 IF(T1-T3)903,903,905                                              1193
  903 TI(J)=T3                                                          1194
      GO TO 910                                                         1195
  904 TI(J)=T2                                                          1196
      GO TO 910                                                         1197
  905 TI(J)=T1                                                          1198
  910 CONTINUE                                                          1199
      DENS(1)=0.D0                                                      1200
      DENS(11)=1.0D0                                                    1201
      DELTAT=(TI(2)-TI(1))/10.0D0                                       1202
      DO 990 J=2,10                                                     1203
      T=TI(1)+DFLOAT(J-1)*DELTAT                                        1204
      K4=1                                                              1205
      FOT=0.D0                                                          1206
      L3=11                                                             1207
  915 K4=K4+1                                                           1208
      IF(K4.GE.12) GO TO 985                                            1209
      TDIF3=T-CDF(J4,K4)                                                1210
      IF(TDIF3.GE.CDF(J3,12)) GO TO 918                                 1211
  916 IF(TDIF3.GT.CDF(J3,L3)) GO TO 917                                 1212
      L3=L3-1                                                           1213
      IF(1-L3)916,916,985                                               1214
C                                                                       1215
C                                                                       1216
C                                                                       1217
C     *     *     *     *     *     *     *     *     *     *     *     1218
  917 IF (L3 .NE. 1) GO TO 951                                          1219
```

```
      T1=CDF(J3,1)                                                      1220
      T2=CDF(J3,2)                                                      1221
      T3=CDF(J3,3)                                                      1222
      Y1=0.D0                                                           1223
      Y2=0.05D0                                                         1224
      Y3=0.15D0                                                         1225
      GO TO 952                                                         1226
  951 T1=CDF(J3,L3)                                                     1227
      T2=CDF(J3,L3+1)                                                   1228
      T3=CDF(J3,L3-1)                                                   1229
      Y1=A(L3)                                                          1230
      Y2=A(L3+1)                                                        1231
      Y3=A(L3-1)                                                        1232
  952 F3=TERPOL(Y1, Y2, Y3, TDIF3, T1, T2, T3)                          1233
      IF(F3 .LT.0.D0) F3=0.D0                                           1234
      IF(F3.GT.1.D0) F3=1.D0                                            1235
C     *     *     *     *     *     *     *     *     *     *     *     1236
C                                                                       1237
C                                                                       1238
C                                                                       1239
      GO TO 920                                                         1240
  918 F3=1.D0                                                           1241
  920 KI=1                                                              1242
      FOT1=0.D0                                                         1243
      L2=11                                                             1244
      L1=11                                                             1245
  921 KI=KI+1                                                           1246
      IF(KI.LT.12) GO TO 925                                            1247
  922 FOT=FOT+FOT1*F3                                                   1248
      GO TO 915                                                         1249
  925 TDIF2=TDIF3-CDF(I,KI)                                             1250
      TDIF1=T-CDF(I,KI)                                                 1251
      IF(TDIF2.GE.CDF(J2,12)) GOTO 928                                  1252
  926 IF(TDIF2.GT.CDF(J2,L2)) GO TO 927                                 1253
      L2=L2-1                                                           1254
      IF(1-L2)926,926,922                                               1255
C                                                                       1256
C                                                                       1257
C                                                                       1258
C     *     *     *     *     *     *     *     *     *     *     *     1259
  927 IF (L2 .NE. 1) GO TO 961                                          1260
      T1=CDF(J2,1)                                                      1261
      T2=CDF(J2,2)                                                      1262
      T3=CDF(J2,3)                                                      1263
      Y1=0.D0                                                           1264
      Y2=0.05D0                                                         1265
      Y3=0.15D0                                                         1266
      GO TO 962                                                         1267
  961 T1=CDF(J2,L2)                                                     1268
      T2=CDF(J2,L2+1)                                                   1269
      T3=CDF(J2,L2-1)                                                   1270
      Y1=A(L2)                                                          1271
      Y2=A(L2+1)                                                        1272
      Y3=A(L2-1)                                                        1273
  962 F2=TERPOL(Y1, Y2, Y3, TDIF2, T1, T2, T3)                          1274
      IF(F2 .LT.0.D0) F2=0.D0                                           1275
      IF(F2 .GT.1.D0) F2=1.D0                                           1276
C     *     *     *     *     *     *     *     *     *     *     *     1277
C                                                                       1278
C                                                                       1279
C                                                                       1280
```

```
      GO TO 930                                                          1281
  928 F2=1.D0                                                            1282
  930 IF(TDIF1.GE.CDF(J1,12)) GO TO 933                                  1283
  931 IF(TDIF1.GT.CDF(J1,L1)) GO TO 932                                  1284
      L1=L1-1                                                            1285
      IF(1-L1)931,931,922                                                1286
C                                                                        1287
C                                                                        1288
C                                                                        1289
C     *     *     *     *     *     *     *     *     *     *     *      1290
  932 IF (L1 .NE. 1) GO TO 971                                           1291
      T1=CDF(J1,1)                                                       1292
      T2=CDF(J1,2)                                                       1293
      T3=CDF(J1,3)                                                       1294
      Y1=0.D0                                                            1295
      Y2=0.05D0                                                          1296
      Y3=0.15D0                                                          1297
      GO TO 972                                                          1298
  971 T1=CDF(J1,L1)                                                      1299
      T2=CDF(J1,L1+1)                                                    1300
      T3=CDF(J1,L1-1)                                                    1301
      Y1=A(L1)                                                           1302
      Y2=A(L1+1)                                                         1303
      Y3=A(L1-1)                                                         1304
  972 F1=TERPOL(Y1, Y2, Y3, TDIF1, T1, T2, T3)                           1305
      IF(F1.LT.0.D0) F1=0.D0                                             1306
      IF(F1.GT.1.D0) F1=1.D0                                             1307
C     *     *     *     *     *     *     *     *     *     *     *      1308
C                                                                        1309
C                                                                        1310
C                                                                        1311
      GO TO 940                                                          1312
  933 F1=1.D0                                                            1313
  940 FCT1=FCT1*F1*F2                                                    1314
      GO TO 921                                                          1315
  985 DENS(J)=FCT*0.1D-1                                                 1316
  990 CONTINUE                                                           1317
      S(J1,2)=S(I,2)                                                     1318
      CALL INVSTR(J1,DELTAT)                                             1319
      RETURN                                                             1320
      END                                                                1321
      FUNCTION TERPOL(X1,X2,X3,Y,Y1,Y2,Y3)                               1322
      IMPLICIT REAL*8(A-H,O-Z)                                           1323
      TERPOL=X1 + ((X2-X1)*(Y-Y1)/(Y2-Y1)) + ((X3-X2)/(Y3-Y2)-(X2-X1)/   1324
     *(Y2-Y1))*((Y-Y1)*(Y-Y2)/(Y3-Y1))                                   1325
      RETURN                                                             1326
      END                                                                1327
      SUBROUTINE INVSTR(I,DELTAT)                                        1328
C                                                                        1329
C     CDF INVERSE STORAGE ROUTINE                                        1330
C                                                                        1331
      IMPLICIT REAL*8(A-H,O-R,T-Z)                                       1332
      COMMON /BLKA/S,CDF,A,DENS,TI                                       1333
      INTEGER S                                                          1334
      DIMENSION S(1000,4),CDF(1000,12),A(12),DENS(400),TI(2)             1335
      TEMPO=TI(1)                                                        1336
      MIN=1                                                              1337
      DO 2263 IA=2,10                                                    1338
      IF (DENS(IA).GE.1.D0) GO TO 2273                                   1339
      IF (DENS(IA).GT.0.D0) GO TO 2263                                   1340
      MIN=IA                                                             1341
```

```
      TI(I)=TEMPO+DFLOAT(IA-1)*DELTAT                                    1342
 2263 CONTINUE                                                           1343
      MAX=11                                                             1344
      GO TO 251                                                          1345
 2273 MAX=IA                                                             1346
      TI(2)=TEMPO+DFLOAT(IA-1)*DELTAT                                    1347
  251 DENS(MIN)=0.D0                                                     1348
      DENS(MAX)=1.D0                                                     1349
      JA=0                                                               1350
      DO 2283 KA=MIN,MAX                                                 1351
      JA=JA+1                                                            1352
      DENS(JA)=DENS(KA)                                                  1353
 2283 CONTINUE                                                           1354
      DO 2293 LA=JA,11                                                   1355
 2293 DENS(LA)=1.D0                                                      1356
  252 K=1                                                                1357
      CDF(I,1)=TI(1)                                                     1358
      CDF(I,12)=TI(2)                                                    1359
      DO 260 M=2,6                                                       1360
  255 IF(A(M).GT.DENS(K).AND.A(M).LE.DENS(K+1)) GO TO 256                1361
      K=K+1                                                              1362
      GO TO 255                                                          1363
  256 Y=A(M)                                                             1364
  257 Y1=DENS(K)                                                         1365
      Y2=DENS(K+1)                                                       1366
      Y3=DENS(K+2)                                                       1367
      T1=TI(1) + DFLOAT(K-1)*DELTAT                                      1368
      D1=Y2-Y1                                                           1369
      D2=Y3-Y2                                                           1370
      D3=Y3-Y1                                                           1371
      DY1=Y-Y1                                                           1372
      DY2=Y-Y2                                                           1373
      CTEMP    =T1+(DY1*DELTAT/D1)+(DY1*DY2*DELTAT*(D1-D2)/(D1*D2*D3))   1374
      IF(CTEMP.LT.T1)CTEMP=T1                                            1375
      IF(CTEMP.GT.T1+2*DELTAT)CTEMP=T1+2*DELTAT                          1376
260   CDF(I,M)=CTEMP                                                     1377
      DO 270 M=7,11                                                      1378
  265 IF(A(M).GT.DENS(K).AND.A(M).LE.DENS(K+1)) GO TO 266                1379
      K=K+1                                                              1380
      GO TO 265                                                          1381
  266 Y=A(M)                                                             1382
      Y1=DENS(K+1)                                                       1383
      Y2=DENS(K)                                                         1384
      Y3=DENS(K-1)                                                       1385
      T1=TI(1) + DFLOAT(K)*DELTAT                                        1386
      D1=Y2-Y1                                                           1387
      D2=Y3-Y2                                                           1388
      D3=Y3-Y1                                                           1389
      DY1=Y-Y1                                                           1390
      DY2=Y-Y2                                                           1391
      CTEMP    =T1-(DY1*DELTAT/D1)-(DY1*DY2*DELTAT*(D1-D2)/(D1*D2*D3))   1392
      IF(CTEMP.LT.T1)CTEMP=T1                                            1393
      IF(CTEMP.GT.T1+2*DELTAT)CTEMP=T1+2*DELTAT                          1394
270   CDF(I,M)=CTEMP                                                     1395
      RETURN                                                             1396
      END                                                                1397
      SUBROUTINE XDENS(MM,I,I500,J200)                                   1398
      IMPLICIT REAL*8(A-H,O-R,T-Z)                                       1399
      COMMON /BLK8/TIME,..                                               1400
      DIMENSION TIME(4),Y(12)                                            1401
      GO TO (30,31,50,60),MM                                             1402
```

```
50      CALL XRECT                                                      1403
        GO TO 70                                                        1404
60      DO 61 M=1,12                                                    1405
61      Y(M)=TIME(1)                                                    1406
        TIME(3)=0.D0                                                    1407
        GO TO 70                                                        1408
   31 IF(TIME(4))32,32,33                                               1409
32      TIME(3)=(TIME(3)-TIME(1))/6.0                                   1410
   33 TIME(1)=TIME(2)                                                   1411
34      CALL NORMAL(J200)                                               1412
        GO TO 70                                                        1413
30      CALL XBETA(I500)                                                1414
70      CONTINUE                                                        1415
        RETURN                                                          1416
        END                                                             1417
        SUBRCUTINE XRECT                                                1418
C       SUBROUTINE TO CALCULATE THE INVERSE FUNCTION OF A RECTANGULAR DIST  1419
C       CALLING ARGUMENTS                                               1420
C          A IS THE MINIMUM VALUE OF THE VARIABLE                       1421
C          B IS THE MAXIMUM VALUE OF THE VARIABLE                       1422
C          Y IS THE ARRAY OF THE INVERSE FUNCTION OF THE DISTRIBUTION   1423
C       THE MEAN IS RETURNED IN A                                       1424
C       THE STANDARD DEVIATION IS RETURNED IN B                         1425
C                                                                       1426
        IMPLICIT REAL*8(A-H,O-Z)                                        1427
        COMMON /BLKB/TIME,Y                                             1428
        DIMENSION TIME(4),Y(12)                                         1429
        A=TIME(1)                                                       1430
        B=TIME(3)                                                       1431
        DELTA=(TIME(3)-TIME(1))/10.D0                                   1432
        Y(1)=A                                                          1433
        Y( 12)=B                                                        1434
        Y(2)=Y(1)+DELTA*0.5D0                                           1435
        DO 10 J=3,11                                                    1436
   10 Y(J)=Y(J-1)+DELTA                                                 1437
        XMU=(A+B)/2.D0                                                  1438
        SIGMA=(B-A)/3.4641D0                                            1439
        TIME(1)=XMU                                                     1440
        TIME(3)=SIGMA                                                   1441
         RETURN                                                         1442
        END                                                             1443
        SUBROUTINE XBETA(IG)                                            1444
        IMPLICIT REAL*8(A-H,O-Z)                                        1445
        COMMON /BLKB/TIME,Y                                             1446
        DIMENSION TIME(4),Y(12)                                         1447
        TERP  (X1,X2,Y,Y1,Y2)=X1+((X2-X1)*(Y-Y1)/(Y2-Y1))               1448
C TERP   IS A ARITHMETIC STATEMENT FUNCTION TO DO LINEAR INTERPOLATION. 1449
C IF X IS THE VALUE REQUIRED WHICH LIES BETWEEN THE VALUES  OF X1 AND X2 1450
C IN A TABLE AND Y1 AND Y2 ARE THECORRESPONDING VALUES WITH Y THE VALUE 1451
C CORRESPONDING TO THE VALUE OF X REQUIRED.                             1452
C SUB PROGRAM TO CALCULATE THE CUMULATIVE INVERSE FUNCTION OF THE BETA  1453
C                          DENSITY FUNCTION                             1454
C                                                                       1455
C                                                                       1456
C      CALLING ARGUMENTS.                                               1457
C        1. BETA # BETA OF THE DENSITY IF KNOWN. ENTER -1 IF IT IS      1458
C           DESIRED TO CALCULATE THESE POINTS BASED ON THE THREE PERT   1459
C           TIME ESTIMATES  XSEE SUBROUTINE ALBETX.                     1460
C        2. A # MINIMUM TIME ESTIMATE.                                  1461
C        3. XM # ALPHA OF THE DENSITY OR THE MOST LIKLEY TIME ESTIMATE. 1462
C        4. B # MAXIMUM TIME ESTIMATE.                                  1463
```

```
C         5. IG # SEE GAMMA SUBROUTINE.                                    1464
C         6. Y # ARRAY 101 X 1 IN WHICH THE VALUES ARE CALCULATED          1465
C         7.  SIGMA IS THE STANDARD DEVIATION OF THE DENSITY               1466
C      THE MEAN OF THE DENSITY IS RETURNED IN A.                           1467
C      THE STANDARD DEVIATION OF THE DENSITY IS RETURNED IN TIME%3<         1468
C                                                                          1469
       A=TIME(1)                                                           1470
       ALPHA=TIME(2)                                                       1471
       B=TIME(3)                                                           1472
       BETA=TIME(4)                                                        1473
       Y(1)=A                                                              1474
       Y(12)=B                                                             1475
    18 FALPHA=ALPHA+1.D0                                                   1476
       XM=ALPHA                                                            1477
       FBETA=BETA + 1.D0                                                   1478
       SUM=ALPHA + BETA + 1.D0                                             1479
       FSUM= SUM + 1.D0                                                    1480
       SIGMA=(FALPHA*FBETA*(B-A)**2)/(((FSUM)**2)*(FSUM+1.D0))             1481
       SIGMA=DSQRT(SIGMA)                                                  1482
       U =  ((A*FBETA)+(B*FALPHA))/FSUM                                    1483
       CALL GAMMA (FALPHA,IG)                                              1484
       CALL GAMMA (FBETA,IG)                                               1485
       CALL GAMMA  (FSUM,IG)                                               1486
       C=B-A                                                               1487
       BT=0.D0                                                             1488
       DELTAT=C/250D0                                                      1489
       T3=0.D0                                                             1490
       FT=0.D0                                                             1491
       CONST=DLOG(FSUM)-DLOG(FALPHA)-DLOG(FBETA)-(SUM*DLOG(C))             1492
       Z3=0.D0                                                             1493
       K=0                                                                 1494
       DO 50 J=2,20,2                                                      1495
       D=DFLOAT(J-1)/20.0                                                  1496
   49  T1=T3                                                               1497
       K=K+1                                                               1498
       X=K                                                                 1499
       XX=K*2                                                              1500
       AT=BT                                                               1501
       T3=X*DELTAT                                                         1502
       T2=(XX-1.D0)*T3/XX                                                  1503
       Z1=Z3                                                               1504
       Z2=CONST+DLOG(4.D0)+(ALPHA*DLOG(T2))+(BETA*DLOG(C-T2))              1505
       Z2=DEXP(Z2)                                                         1506
       Z3=CONST+(ALPHA*DLOG(T3))+(BETA*DLOG(C-T3))                         1507
       Z3=DEXP(Z3)                                                         1508
       FT=FT+Z1+Z2+Z3                                                      1509
       BT=FT*DELTAT/6.D0                                                   1510
       IF(BT.GT.1.D0) BT=1.D0                                              1511
       IF(D.GT.AT.AND.D.LE.BT)  GO TO 51                                   1512
       GO TO 49                                                            1513
   51  NJ=(J/2)+1                                                          1514
       Y(NJ)=TERP  (T1,T3,D,AT,BT) + Y(1)                                  1515
   50  CONTINUE                                                            1516
       TIME(1)=U                                                           1517
       TIME(2)=ALPHA                                                       1518
       TIME(3)=SIGMA                                                       1519
       TIME(4)=BETA                                                        1520
       RETURN                                                              1521
       END                                                                 1522
       SUBROUTINE GAMMA (X,M)                                              1523
C                                                                          1524
```

```
C                                                                       1525
C  SUBROUTINE TO COMPUTE THE GAMMA FUNCTION %SINGLE PRECISION<          1526
C                                                                       1527
C        CALLING ARGUMENTS                                              1528
C             X IS THE NUMBER OF WHICH THE GAMMA IS TO BE COMPUTED      1529
C             X IS A FLOATING NUMBER                                    1530
C             M IS AN INTEGER NUMBER                                    1531
C             M IS DEFINED AS FOLLOWS                                   1532
C                  M MUST BE ZERO THE FIRST TIME THE SUBROUTINE IS CALLE 1533
C                                                                       1534
      IMPLICIT REAL*8(A-H,O-R,T-Z)                                      1535
      DIMENSION Z(18)                                                   1536
      IF(M)  1, 1, 2                                                    1537
    1 B=1.D0                                                            1538
      D=2.D0                                                            1539
      M=1                                                               1540
      Z( 1)= 1.000000000D 00                                            1541
      Z( 2)=  .577215664D 00                                            1542
      Z( 3)=- .655878071D 00                                            1543
      Z( 4)=- .042002635D 00                                            1544
      Z( 5)=  .166538611D 00                                            1545
      Z( 6)=- .042197734D 00                                            1546
      Z( 7)=- .009621971D 00                                            1547
      Z( 8)=  .007218943D 00                                            1548
      Z( 9)=- .001165167D 00                                            1549
      Z(10)=- .000215241D 00                                            1550
      Z(11)=  .000128050D 00                                            1551
      Z(12)=- .000020134D 00                                            1552
      Z(13)=- .000001250D 00                                            1553
      Z(14)=  .000001133D 00                                            1554
      Z(15)=- .000000205D 00                                            1555
      Z(16)=  .000000006D 00                                            1556
      Z(17)=  .000000005D 00                                            1557
      Z(18)=- .000000001D 00                                            1558
    2 C=1.D0                                                            1559
    3 IF(X) 99,99, 4                                                    1560
    4 IF(X-B)  7, 6, 5                                                  1561
    5 IF(D-X) 8, 6, 9                                                   1562
    6 G=1.D0                                                            1563
      GO TO 11                                                          1564
    7 C=C*B/X                                                           1565
      X=X+B                                                             1566
      GO TO 3                                                           1567
    8 X=X-B                                                             1568
      C=C*X                                                             1569
      GO TO 3                                                           1570
    9 E=0.D0                                                            1571
      DO 10 I=1,18                                                      1572
      T=I                                                               1573
   10 E=E+(Z(I)*(X**T))                                                 1574
      G=B/E                                                             1575
   11 X=C*G                                                             1576
   99 RETURN                                                            1577
      END                                                               1578
      SUBROUTINE NORMAL(J2)                                             1579
C    SUBROUTINE TO CALCULATE THE INVERSE FUNCTION OF THE NORMAL DISTRIB. 1580
C  CALLING ARGUMENTS.                                                   1581
C    1.  Y IS A FLOATING POINT ARRAY FOR OUTPUT OF THE CUMULATIVE INVERS 1582
C        FUNCTION.                                                      1583
C    2.  U IS THE MEAN OF THE DENSITY.                                  1584
C    3.  SIGMA IS THE STANDARD DEVIATION OF THE DENSITY.                1585
```

```
C     4.  Z IS THE ARRAY IN WHICH THE VALUES OF THE FUNCTION ARE CALCULAT     1586
C     5.  J2 IS A DUMMY VARIABLE WHICH MUST BE 0 OR 1 THE FIRST TIME THE     1587
C         IS CALLED OR ANYTIME THE Z ARRAY IS USED AGAIN BY THE MAIN         1588
C         PROGRAM.  WHEN J2 IS      MINUS THE Z TABLE IS NOT RECALCULATED    1589
C               THE PROGRAM RETURNS VALUES OF Y FOR THE FUNCTION             1590
C     T # U &Z * SIGMA         FOR VALUES OF F%Y<#.01,.02, ....,99. ,1.0.    1591
C    Y%1< WILL CONTAIN U - 3 STANDARD DEVIATIONS.IF & OR.0 IF -. NEGATIVE     1592
C    VALUES OF Y ARE NOT CALCULATED.  Y% 22< WILL CONTAIN U & 3 STANDARD      1593
C    DEVIATIONS.                                                              1594
      IMPLICIT REAL*8(A-H,O-Z)                                               1595
      COMMON /BLK8/TIME,Y                                                    1596
      DIMENSION TIME(4),Y(12)                                                1597
      DIMENSION Z(10)                                                        1598
      U=TIME(2)                                                              1599
      SIGMA=TIME(3)                                                          1600
      IF (J2) 5,10,10                                                        1601
   10 J2=-1                                                                  1602
      Z(1)=-1.6449300                                                        1603
      Z(2)=-1.0364500                                                        1604
      Z(3)=-0.6744900                                                        1605
      Z(4)=-0.3853200                                                        1606
      Z(5)=-0.1256600                                                        1607
      Z( 6)=0.1256600                                                        1608
      Z( 7)=0.3853200                                                        1609
      Z( 8)=0.6744900                                                        1610
      Z( 9)=1.0364500                                                        1611
      Z(10)=1.6449300                                                        1612
    5 TZERO=U-(3.D0*SIGMA)                                                   1613
      IF (TZERO) 1,2,2                                                       1614
    1 Y(1)=0.D0                                                              1615
      GO TO 8                                                                1616
    2 Y(1)=TZERO                                                             1617
    8 DO 9 I=2,11                                                            1618
      Y(I)=Z(I-1)*SIGMA+U                                                    1619
      IF (Y(I)) 6,6,9                                                        1620
    6 Y(I)=0.D0                                                              1621
    9 CONTINUE                                                               1622
      Y(12)=U+3.D0*SIGMA                                                     1623
      RETURN                                                                 1624
      END                                                                    1625
      SUBROUTINE RD(J,A,N)                                                   1626
      DIMENSION A(N)                                                         1627
      READ(J) A                                                              1628
      RETURN                                                                 1629
      END                                                                    1630
```

6. DECOMP

```
C      DECOMPOSITION PROGRAM                                                    1
C                                                                               2
C         DECOMP  IDENTIFIES PARALLEL SUBNETWORKS AND SERIES SUBNETWORKS         3
C                                                                               4
C                                                                               5
C         THE FOLLOWING IS AN ALPHABETICAL LISTING OF THE VARIABLES AND         6
C         ARRAYS THAT ARE USED IN THIS MAIN PROGRAM AND ITS SUBROUTINES         7
C                                                                               8
C                                                                               9
C         ARCS = THE NUMBER OF ARCS IN THE SUBNETWORK                          10
C         BDNUM(I) = THE BUNDLE NUMBER TO WHICH NODE I IS ASSIGNED             11
C         CARC = AN ARRAY PUNCHED FOR USE IN PROGRAM: SYNTHESIS                12
C         CHECK = ARRAY USED TO STORE ARCS HAVING THE SOURCE AND THE           13
C                 SINK AS THEIR ONLY NODES                                     14
C         CTNSUB = THE TEMPORARY NUMBER OF SUBNETWORKS FOUND IN THE            15
C                  PREVIOUS STEP                                               16
C         HEAD(I)=TERMINATING NODE FOR ARC I IN THE RENUMBERED SUBNETWORK       17
C         INSTN = THE CURRENT INSTRUCTION NUMBER, PUNCHED FOR SYNTHESIS        18
C         LNODEN = THE LARGEST NODE NUMBER BEING READ IN                       19
C         MAXND = THE LARGEST NODE NUMBER THAT HAS ALREADY BEEN ASSIGNED       20
C              AT LEAST TEMPORARILY TO A BUNDLE                                21
C         NARCSS(I) = THE NUMBER OF ARCS IN SUBNETWORK I                       22
C         NODES=TEMPORARY ARRAY USED TO ORDER THE NODE NUMBERS IN A SUB        23
C         NSUB = THE TOTAL OF SUBNETWORKS THUS FAR                             24
C         NTARC = THE NUMBER OF ARCS IN THIS SUBNETWORK                        25
C         NUMBD = THE NUMBER OF BUNDLES CREATED                                26
C         NUMSUB=NUMBER OF MINIMUM SUBNETWORKS IDENTIFIED                      27
C         PPOST = ARRAY OF PAST POSTS USED IN CUT SUBROUTINE                   28
C         S(I) = THE STARTING NODE FOR ARC I                                   29
C         SINK   = THE NODE NUMBER CORRESPONDING TO THE SINK                   30
C         SINKS(I) = THE SINK IN SUBNETWORK I                                  31
C         SOURC(I) = SOURCE NODE IN SUBNETWORK I                               32
C         SOURCE = THE NODE NUMBER CORRESPONDING TO THE SOURCE                 33
C         STEP = STAGE NUMBER                                                  34
C         SUBNET(I,J) = THE ITH ARC IN THE JTH SUBNETWORK                      35
C         SUMARC = THE CURRENT NUMBER OF ARCS IN SUBNETWORK NSUB               36
C         T(I) = THE TERMINATING NODE FOR ARC I                                37
C         TAIL(I)=STARTING NODE FOR ARC I IN THE RENUMBERED SUBNETWORK         38
C         TARC = THE SUBNETWORK WITHOUT THE ARCS INVOLVING NODE K              39
C         TLNDEN = TEMPORARY LARGEST NODE NUMBER                               40
C         TNSUB = THE NUMBER OF SUBNETWORKS CREATED IN THE CURRENT STAGE       41
C         TSNSUB = THE TEMPORARY SUBNETWORK BEING USED IN STAGE                42
C                  SUBROUTINE                                                  43
C         TSUBN = THE NUMBER OF THE SUBNETWORK CURRENTLY BEING CONSIDERED      44
C         TYPESN = THE TYPE OF SUBNETWORK BEING CONSIDERED:                    45
C              1 = BUNDLE SUBNETWORK  2 = CUT SUBNETWORK                       46
C                                                                              47
C         A NON-DECOMPOSABLE SUBNETWORK IS ONE THAT IS NOT COMPOSED OF         48
C         SMALLER SUBNETWORKS WHICH ARE CONNECTED IN EITHER SERIES OF          49
C         PARALLEL                                                             50
C                                                                              51
C                                                                              52
C         FOR THE SAKE OF IDENTIFYING THE APPROPRIATE DIMENSIONS, LET          53
C         MMAX = THE MAXIMUM NUMBER OF ACTIVITIES IN THE ORIGINAL              54
C                PROJECT NETWORK                                               55
C         MSMAX = THE MAXIMUM NUMBER OF ACTIVITIES ALLOWED IN ANY ONE          56
C                SUBNETWORK                                                    57
C         NSUBMAX = THE MAXIMUM NUMBER OF SUBNETWORKS ALLOWED IN THE           58
C                DECOMPOSITION PROCESS                                         59
C         NCINSMAX = THE MAXIMUM NUMBER OF INSTRUCTIONS GENERATED BY THE       60
```

```
C                    DECOMPOSITION PROCESS                                  61
C        CURRENTLY, MMAX=1000; MSMAX=500; NSUBMAX=100; NOINSMAX=100         62
C                                                                           63
C      DIMENSION S(MMAX),T(MMAX),SUBNET(MMAX,NSUBMAX),SOURC(NSUBMAX),       64
C     *          NARCSS(NSUBMAX),CARD(NOINS,27)                             65
C                                                                           66
C        IN SUBROUTINE NETIN THE ARRAY DIMENSIONS ARE:                      67
C      DIMENSION FILE3(3*MMAX+5),IORG(MMAX),ITERM(MMAX),NUMS(MMAX)          68
C      EQUIVLAENCE (FILE3(MMAX+6),IORG(1)),(FILE3(2*MMAX+6),ITERM(1))       69
C                                                                           70
C        IN SUBROUTINE BUNDLE THE ARRAY DIMENSIONS ARE:                     71
C      DIMENSION BDNUM(MMAX),CHECK(MMAX)                                    72
C                                                                           73
C        IN SUBROUTINE CUT THE ARRAY DIMENSIONS ARE:                        74
C      DIMENSION ORIGIN(MMAX),POST(MMAX),TARC(MMAX),RCUT(NSUBMAX),          75
C     *          PPOST(MMAX)                                                76
C                                                                           77
C        IN SUBROUTINE PRESUB THE ARRAY DIMENSIONS ARE:                     78
C      DIMENSION HEAD(MMAX),TAIL(MMAX),NODE(MMAX),FILE3(3*MSMAX+3),         79
C     *          NET(3,MSMAX)                                               80
C                                                                           81
      IMPLICIT INTEGER*2 (A-Z)                                              82
      COMMON SUBNET,S,T,SOURC,SINKS,NARCSS,NSUB,TNSUB                       83
      COMMON /NEW/ CARD,INSTN                                               84
      COMMON/PREP/ NUMSUB                                                   85
      INTEGER                    F3/11/,      F5/13/                        86
      INTEGER CARD(100,27)                                                  87
      INTEGER*2 S(1000),T(1000),SUBNET(1000,100),SOURC(100),SINKS(100),     88
     *NARCSS(100)                                                           89
      WRITE(6,100)                                                          90
100   FORMAT(1H1,132('*')/1X,132('*')/'0THIS IS THE OUTPUT FROM THE DECO    91
     *MPOSITION PROGRAM: DECOMP'/1H0,132('*')/1X,132('*'))                  92
      REWIND F3                                                             93
      READ(F3)                                                              94
      REWIND F5                                                             95
      READ(F5)                                                              96
C                                                                           97
C        INITIALIZE INSTN                                                   98
C        INSTN = THE CURRENT INSTRUCTION NUMBER                             99
C                                                                          100
6000  INSTN=0                                                              101
C                                                                          102
C        INITIALIZE CARD                                                   103
C                                                                          104
      DO 22200 I=1,100                                                     105
      DO 22200 J=1,27                                                      106
22200 CARD(I,J)=0                                                          107
C                                                                          108
C        READ THE NETWORK IN                                               109
C                                                                          110
      CALL NETIN (LNODEN)                                                  111
      REWIND F3                                                            112
      READ(F3)                                                             113
      STEP=1                                                               114
      TYPESN=1                                                             115
      TSUBN=1                                                              116
      NUMSUB=0                                                             117
      CALL BUNDLE (LNODEN,TSUBN)                                           118
C                                                                          119
C        PRINT OUT THIS STAGE OF THE BREAKUP                               120
C                                                                          121
```

```
      CALL STAGE (TYPESN,TSUBN)                                          122
      CTNSUB=TNSUB                                                       123
      GO TO 70                                                           124
C                                                                        125
C        FIND THE NUMBER OF NEWLY CREATED SUBNETWORKS                    126
C                                                                        127
10    CTNSUB=0                                                           128
      TSUBN=NSUB-LOOP                                                    129
      TYPESN=1                                                           130
      DO 20 I=1,LOOP                                                     131
C                                                                        132
C        FIND THE NEXT SUBNETWORK TO BE FURTHER SUBDIVIDED               133
C                                                                        134
      TSUBN=TSUBN+1                                                      135
C                                                                        136
C        FIND THE LARGEST NODE NUMBER IN SUBNETWORK TSUBN                137
C                                                                        138
      CALL NODER (TLNDEN,TSUBN)                                          139
C                                                                        140
C        FIND THE BUNDLE SUBNETWORKS                                     141
C                                                                        142
      CALL BUNDLE (TLNDEN,TSUBN)                                         143
C                                                                        144
C        PRINT OUT THIS STAGE OF THE BREAKUP                             145
C                                                                        146
      CALL STAGE (TYPESN,TSUBN)                                          147
C                                                                        148
C        IF THERE IS ONLY ONE BUNDLE FOUND IN SUBNETWORK TSUBN, WE ARE   149
C        FINISHED.  PRINT OUT ITS COMPONENT ARCS                         150
C                                                                        151
      IF (TNSUB.EQ.1) CALL ENDSNT (TSUBN)                                152
C                                                                        153
C        COUNT THE NEW NUMBER OF SUBNETWORKS CREATED                     154
C                                                                        155
20    CTNSUB=TNSUB+CTNSUB                                                156
C                                                                        157
C        IF ALL SUBNETWORKS ARE IN THEIR SMALLEST FORM, WE ARE FINISHED  158
C                                                                        159
      IF (CTNSUB.EQ.0) GO TO 90                                          160
      GO TO 70                                                           161
80    LOOP=CTNSUB                                                        162
      CTNSUB=0                                                           163
      TSUBN=NSUB-LOOP                                                    164
C                                                                        165
C        FIND THE NUMBER OF NEWLY CREATED SUBNETWORKS                    166
C                                                                        167
      TYPESN=2                                                           168
      DO 30 I=1,LOOP                                                     169
C                                                                        170
C        FIND THE NEXT SUBNETWORK TO BE FURTHER SUBDIVIDED               171
C                                                                        172
      TSUBN=TSUBN+1                                                      173
      IF (TSUBN.EQ.1) GO TO 55                                           174
C                                                                        175
C        FIND THE LARGEST NODE NUMBER IN SUBNETWORK TSUBN                176
C                                                                        177
      CALL NODER (TLNDEN,TSUBN)                                          178
C                                                                        179
C        FIND THE CUT SUBNETWORKS                                        180
C                                                                        181
55    IF (TSUBN.EQ.1) TLNDEN=LNODEN                                      182
```

```
      CALL CUT (TSUBN,TLNDEN)                                          183
C                                                                      184
C        PRINT OUT THIS STAGE OF THE BREAKUP                           185
C                                                                      186
      CALL STAGE (TYPESN,TSUBN)                                        187
C                                                                      188
C        IF THERE ARE NO CUTS FOUND IN SUBNETWORK TSUBN, WE ARE FINISHED 189
C        PRINT OUT ITS COMPONENT ARCS                                  190
C                                                                      191
      IF (TNSUB.EQ.1) CALL ENDSNT (TSUBN)                              192
C                                                                      193
C        COUNT THE NEW NUMBER OF SUBNETWORKS CREATED                   194
C                                                                      195
30    CTNSUB=TNSUB+CTNSUB                                              196
C                                                                      197
C        IF ALL SUBNETWORKS ARE IN THEIR SMALLEST FORM, WE ARE FINISHED 198
C                                                                      199
      IF (CTNSUB.EQ.0) GO TO 90                                        200
      GO TO 70                                                         201
85    LOOP=CTNSUB                                                      202
      GO TO 10                                                         203
70    WRITE (6,900) STEP                                               204
900   FORMAT (1H1,5X,'STAGE',I3,' BREAKUP')                            205
      STEP=STEP+1                                                      206
C                                                                      207
C        LET'S GO BACK TO THE APPROPRIATE LOOP FOR THE NEXT STAGE      208
C                                                                      209
      GO TO (80,85),TYPESN                                             210
90    CONTINUE                                                         211
C                                                                      212
C        STORE INFORMATION FOR SYNTHESIS PROGRAM                       213
C                                                                      214
      CALL PUNSYN                                                      215
      WRITE(6,7001) NUMSUB                                             216
7001  FORMAT(1H0,'          THE NUMBER OF NON-DECOMPOSABLE SUBNETWORKS I 217
     *S ',I3)                                                          218
      WRITE (6,9000)                                                   219
9000  FORMAT (1H1)                                                     220
      STOP                                                             221
      END                                                              222
      SUBROUTINE NETIN (LNODEN)                                        223
      IMPLICIT INTEGER*2 (A-Z)                                         224
      COMMON SUBNET,S,T,SOURC,SINKS,NARCSS,NSUB,TNSUB                  225
      INTEGER*2 S(1000),T(1000),SUBNET(1000,100),SOURC(100),SINKS(100), 226
     *NARCSS(100)                                                      227
      INTEGER*4 FILE3(3005)                                            228
      INTEGER*4 IORG(1000),ITERM(1000),NUMS(1000)                      229
      EQUIVALENCE (FILE3(6),NUMS(1)),(FILE3(1006),IORG(1)),            230
     *(FILE3(2006),ITERM(1))                                           231
      INTEGER                        F3/11/                            232
      READ(F3) FILE3                                                   233
C                                                                      234
C        ZEROIZE SOURC ARRAY                                           235
C                                                                      236
      DO 20 I=1,50                                                     237
20    SOURC(I)=0                                                       238
C                                                                      239
C        READ IN THE INITIAL NETWORK LIMITS                            240
C                                                                      241
      ARCS = FILE3(1)                                                  242
      SOURCE = FILE3(3)                                                243
```

```
      SINK = FILE3(4)                                                    244
      LNODEN = FILE3(5)                                                  245
      WRITE (6,200)                                                      246
200   FORMAT (1H0,5X,'INPUT STAGE')                                      247
      WRITE (6,210) ARCS,SOURCE,SINK,LNODEN                              248
210   FORMAT(1H0,/6X,'THE SIMPLIFIED NETWORK HAS',12X,I4,' ARCS',//6X,   249
     *'THE SOURCE IS NODE NUMBER',11X,I3,//6X,'THE SINK IS NODE NUMBER',  250
     *13X,I3,//6X,'THE LARGEST NODE IS NODE NUMBER',5X,I3)               251
      WRITE (6,220)                                                      252
220   FORMAT(1H0,5X,'THE SIMPLIFIED NETWORK AS READ IN IS:',///6X,       253
     *'ARC NUMBER',5X,'ORIGIN NODE',5X,'TERMINAL NODE')                  254
C                                                                        255
C        READ IN EACH ARC AND ITS STARTING AND TERMINATING NODES         256
C        S = THE NODE NUMBER FOR THE START OF AN ARC                     257
C        T = THE TERMINAL NODE OF AN ARC                                 258
C                                                                        259
      DO 10 J=1,ARCS                                                     260
      I=NUMS(J)                                                          261
      S(I)=IORG(J)                                                       262
      T(I)=ITERM(J)                                                      263
      WRITE(6,240)I,S(I),T(I)                                            264
240   FORMAT (1H ,8X,I3,13X,I3,14X,I3)                                   265
C                                                                        266
C        CREATE THE FIRST SUBNETWORK                                     267
C                                                                        268
10    SUBNET(J,1)=I                                                      269
      SOURC(1)=SOURCE                                                    270
      SINKS(1)=SINK                                                      271
      NARCSS(1)=ARCS                                                     272
      NSUB=1                                                             273
      RETURN                                                             274
      END                                                                275
      SUBROUTINE NODER (TLNDEN,TSUBN)                                    276
C                                                                        277
C        FINDS LARGEST NODE NUMBER IN THE SUBNETWORK TSUBN               278
C                                                                        279
      IMPLICIT INTEGER*2 (A-Z)                                           280
      COMMON SUBNET,S,T,SOURC,SINKS,NARCSS,NSUB,TNSUB                    281
      INTEGER*2 S(1000),T(1000),SUBNET(1000,100),SOURC(100),SINKS(100),  282
     *NARCSS(100)                                                        283
      ARCS=NARCSS(TSUBN)                                                 284
      TLNDEN=0                                                           285
      DO 20 J=1,ARCS                                                     286
      A=SUBNET(J,TSUBN)                                                  287
      M=S(A)                                                             288
      N=T(A)                                                             289
      MAXND=N                                                            290
      IF (M.GT.N) MAXND=M                                                291
      IF (MAXND.GT.TLNDEN) TLNDEN=MAXND                                  292
20    CONTINUE                                                           293
      RETURN                                                             294
      END                                                                295
      SUBROUTINE ENDSNT (TSUBN)                                          296
C                                                                        297
C        PRINTS SMALLEST BREAKDOWN OF SUBNETWORK TSUBN                   298
C                                                                        299
      IMPLICIT INTEGER*2 (A-Z)                                           300
      COMMON SUBNET,S,T,SOURC,SINKS,NARCSS,NSUB,TNSUB                    301
      INTEGER*2 S(1000),T(1000),SUBNET(1000,100),SOURC(100),SINKS(100),  302
     *NARCSS(100)                                                        303
      SOURCE=SOURC(TSUBN)                                                304
```

```
      SINK=SINKS(TSUBN)                                                 305
      WRITE (6,100) TSUBN                                               306
100   FORMAT (1H0,//16X,'SUBNETWORK ',I3,' IS A NON-DECOMPOSABLE NETWORK 307
     *',/16X,'IT IS COMPOSED OF:')                                      308
      WRITE (6,200) SOURCE,SINK                                         309
200   FORMAT (1H0,19X,'SOURCE NODE = ',I3,//20X,'SINK NODE   = ',I3)    310
      M=NARCSS(TSUBN)                                                   311
      WRITE(6,500)                                                      312
500   FORMAT(1H0,15X,'THE SUBNETWORK AS IT IS READ IN:')                313
      WRITE (6,400)                                                     314
400   FORMAT (1H0,19X,'ARC',2X,'S(ARC)',2X,'T(ARC)')                    315
      DO 10 I=1,M                                                       316
      N=SUBNET(I,TSUBN)                                                 317
10    WRITE (6,300) N,S(N),T(N)                                         318
300   FORMAT (1H ,19X,I3,3X,I3,5X,I3)                                   319
C                                                                       320
C        SET TNSUB=0 SO THAT THE REMAINING NUMBER OF SUBNETWORKS DOESN'T 321
C        INCLUDE THIS MINIMUM SUBNETWORK                                322
C                                                                       323
      TNSUB=0                                                           324
C                                                                       325
C        PREPARE THE MINIMUM SUBNETWORK FOR SUBNETWORK ANALYSIS         326
C                                                                       327
      CALL PRESUB(TSUBN)                                                328
      RETURN                                                            329
      END                                                               330
      SUBROUTINE STAGE (TYPESN,TSUBN)                                   331
C                                                                       332
C        PRINTS OUT THE CURRENT STAGE OF BREAKUP                        333
C                                                                       334
      IMPLICIT INTEGER*2 (A-Z)                                          335
      COMMON SUBNET,S,T,SOURC,SINKS,NARCSS,NSUB,TNSUB                   336
      INTEGER*2 S(1000),T(1000),SUBNET(1000,100),SOURC(100),SINKS(100), 337
     *NARCSS(100)                                                       338
      COMMON /NEW/ CARD,INSTN                                           339
      INTEGER CARD(100,27)                                              340
C                                                                       341
C        TNSUB=THE NUMBER OF NEW SUBNETWORKS RESULTING FROM THE BREAKUP 342
C        IN THIS STAGE                                                  343
C                                                                       344
C        TNSUB=1 IMPLIES NO BREAKUP OCCURRED IN THIS STAGE              345
C                                                                       346
      IF (TNSUB.EQ.1) GO TO 600                                         347
      WRITE (6,100) TSUBN                                               348
100   FORMAT (1H0,/6X,'SUBNETWORK ',I3,' IS COMPOSED OF SUBNETWORKS: ') 349
      ISNSUB=NSUB-TNSUB                                                 350
      M=TSNSUB+1                                                        351
      WRITE (6,300)(I,I=M,NSUB)                                         352
300   FORMAT (1H0,5X,I3,20(',',I3))                                     353
C                                                                       354
C        PREPARE INFORMATION FOR SYNTHESIS PROGRAM                      355
C                                                                       356
      INSTN=INSTN+1                                                     357
      ISORP=0                                                           358
      IF(TYPESN.EQ.1) ISORP=1                                           359
      CARD(INSTN,1)=INSTN                                               360
      CARD(INSTN,2)=TSUBN                                               361
      CARD(INSTN,3)=ISORP                                               362
      DO 2250 I=M,NSUB                                                  363
      K=I-M+4                                                           364
2250  CARD(INSTN,K)=I                                                   365
```

```
      IF (TYPESN.EQ.1) GO TO 60                                          366
      WRITE (6,400)                                                      367
400   FORMAT (1H0,5X,'IN SERIES')                                        368
      GO TO 600                                                          369
60    WRITE (6,500)                                                      370
500   FORMAT (1H0,5X,'IN PARALLEL')                                      371
600   RETURN                                                             372
      END                                                                373
      SUBROUTINE BUNDLE (LNODEN,TSUBN)                                   374
C                                                                        375
C        BUNDLE IDENTIFIES PARALLEL SUBNETWORKS CONNECTING DESIGNATED    376
C        SOURCE AND SINK                                                 377
C                                                                        378
      IMPLICIT INTEGER*2 (A-Z)                                           379
      COMMON SUBNET,S,T,SOURC,SINKS,NARCSS,NSUB,TNSUB                    380
      INTEGER*2 S(1000),T(1000),SUBNET(1000,100),SOURC(100),SINKS(100),  381
     *NARCSS(100)                                                        382
      INTEGER*2 BDNUM(1000),CHECK(1000)                                  383
C                                                                        384
C        GROUP NODES INTO BUNDLES                                        385
C                                                                        386
       SOURCE=SOURC(TSUBN)                                               387
       SINK=SINKS(TSUBN)                                                 388
      NUMBD=1                                                            389
C                                                                        390
C        ZEROIZE THE BDNUM ARRAY                                         391
C                                                                        392
      DO 10 I=1,LNODEN                                                   393
   10 BDNUM(I)=0                                                         394
      K=SUBNET(1,TSUBN)                                                  395
      M=S(K)                                                             396
      N=T(K)                                                             397
      BDNUM(M)=1                                                         398
      BDNUM(N)=1                                                         399
      MAXND=N                                                            400
      IF (M.GT.N) MAXND=M                                                401
      ARCS=NARCSS(TSUBN)                                                 402
      IF (ARCS.EQ.1) GO TO 515                                           403
      DO 1 K=2,ARCS                                                      404
      I=SUBNET(K,TSUBN)                                                  405
      BDNUM(SOURCE)=0                                                    406
      BDNUM(SINK)=0                                                      407
      M=S(I)                                                             408
      N=T(I)                                                             409
      IF(M.GT.MAXND) MAXND=M                                             410
      IF(N.GT.MAXND) MAXND=N                                             411
C                                                                        412
C        IF AT LEAST 1 NODE ON THE ARC HAS NOT BEEN ASSIGNED TO A BUNDLE 413
C        GO TO 2                                                         414
C                                                                        415
      IF(BDNUM(M).EQ.0) GO TO 2                                          416
C                                                                        417
C        IF ONLY THE TERMINAL NODE ON THE ARC HAS NOT BEEN ASSIGNED TO A 418
C        BUNDLE, GO TO 3                                                 419
C                                                                        420
      IF(BDNUM(N).EQ.0) GO TO 3                                          421
C                                                                        422
C        IF BOTH NODES ON THE ARC HAVE BEEN ASSIGNED TO THE SAME BUNDLE  423
C        EVERYTHING IS OKAY, GO TRY ANOTHER ARC                          424
C                                                                        425
      IF(BDNUM(N).EQ.BDNUM(M)) GO TO 1                                   426
```

```
C                                                                       427
C       IF THE NODES ON THE ARC ARE ASSIGNED TO DIFFERENT BUNDLES,      428
C       THEN THESE TWO BUNDLES SHOULD BE POOLED                         429
C                                                                       430
      IF(BDNUM(N).LT.BDNUM(M)) GO TO 6                                  431
C                                                                       432
C       POOL BUNDLES                                                    433
C       THE BUNDLE WITH THE LARGER BUNDLE NUMBER IS POOLED INTO THE     434
C       BUNDLE WITH THE SMALLER BUNDLE NUMBER                           435
C       THE BUNDLE NUMBERS OF ALL BUNDLES ARE ALL ADJUSTED              436
C                                                                       437
      MAXBD=BDNUM(N)                                                    438
      MINBD=BDNUM(M)                                                    439
      GO TO 7                                                           440
    6 MAXBD=BDNUM(M)                                                    441
      MINBD=BDNUM(N)                                                    442
    7 DO 5 J=1,MAXND                                                    443
      B=BDNUM(J)                                                        444
      IF (B.EQ.MAXBD) BDNUM(J)=MINBD                                    445
      IF (B.GT.MAXBD) BDNUM(J)=BDNUM(J)-1                               446
    5 CONTINUE                                                          447
      NUMBD=NUMBD-1                                                     448
      GO TO 1                                                           449
C                                                                       450
C       IF BOTH NODES ON THE ARC ARE UNASSIGNED, GO TO 4 WHERE A NEW    451
C       BUNDLE IS CREATED                                               452
C                                                                       453
    2 IF(BDNUM(N).EQ.0) GO TO 4                                         454
C                                                                       455
C       ASSIGN THE ORIGIN NODE OF THE ARC TO THE BUNDLE CONTAINING THE  456
C       TERMINAL NODE                                                   457
C                                                                       458
      BDNUM(M)=BDNUM(N)                                                 459
      GO TO 1                                                           460
C                                                                       461
C       ASSIGN THE TERMINAL NODE OF THE ARC TO THE BUNDLE CONTAINING    462
C       THE ORIGIN NODE OF THE ARC                                      463
C                                                                       464
    3 BDNUM(N)=BDNUM(M)                                                 465
      GO TO 1                                                           466
C                                                                       467
C       CREATE A NEW BUNDLE                                             468
C                                                                       469
    4 NUMBD=NUMBD+1                                                     470
      BDNUM(M)=NUMBD                                                    471
      BDNUM(N)=NUMBD                                                    472
    1 CONTINUE                                                          473
  515 CONTINUE                                                          474
      BDNUM(SINK)=0                                                     475
C                                                                       476
C       IF WE ONLY HAVE 1 BUNDLE FROM THE SUBNETWORK, WE ARE FINISHED   477
C                                                                       478
      IF (NUMBD.EQ.1) GO TO 219                                         479
C                                                                       480
C       ZEROIZE CHECK ARRAY                                             481
C                                                                       482
      DO 290 I=1,ARCS                                                   483
  290 CHECK(I)=0                                                        484
      L=0                                                               485
C                                                                       486
C       THE NODES ARE IN BUNDLES.  PUT THE ASSOCIATED ARCS INTO         487
```

```
C          APPROPRIATE PARALLEL SUBNETWORKS                                488
C                                                                          489
      DO 33 I=1,NUMBD                                                      490
      SUMARC=0                                                             491
      NSUB=NSUB+1                                                          492
      DO 34 K=1,ARCS                                                       493
      M=SUBNET(K,ISUBN)                                                    494
      N=S(M)                                                               495
C                                                                          496
C          SOURCE AND SINK HAVE BUNDLE NUMBER 0                            497
C                                                                          498
      IF (N.EQ.SOURCE) N=T(M)                                              499
      IF (BDNUM(N).EQ.I) GO TO 239                                         500
      IF (N.EQ.SINK) GO TO 229                                             501
      GO TO 34                                                             502
C                                                                          503
C          SPECIAL CASE:  BUNDLE HAS ONLY 2 NODES: SOURCE, SINK.           504
C          PUT ALL ARCS THAT ARE PARALLEL SUBNETWORKS BY THEMSELVES INTO    505
C          THE CHECK ARRAY                                                 506
C                                                                          507
229   DO 291 J=1,K                                                         508
      W=CHECK(J)                                                           509
      IF (W.EQ.0) GO TO 292                                                510
      IF (M.EQ.W) GO TO 34                                                 511
291   CONTINUE                                                             512
292   CHECK(J)=M                                                           513
      GO TO 34                                                             514
C                                                                          515
C          THIS ARC IS IN THE BUNDLE I, HENCE IT IS IN THE ITH NEW         516
C          SUBNETWORK                                                      517
C                                                                          518
239   SUMARC=SUMARC+1                                                      519
      SUBNET(SUMARC,NSUB)=M                                                520
34    CONTINUE                                                             521
C                                                                          522
C          CREATE NEW SUBNETWORKS                                          523
C                                                                          524
C                                                                          525
C          IF THIS BUNDLE HAS NO NODES, PUT AN ARC SUBNETWORK INTO         526
C          SUBNET (1,NSUB)                                                 527
C                                                                          528
      IF (SUMARC.EQ.0) GO TO 333                                           529
343   NARCSS(NSUB)=SUMARC                                                  530
      SOURC(NSUB)=SOURCE                                                   531
      SINKS(NSUB)=SINK                                                     532
      GO TO 33                                                             533
C                                                                          534
C          STORE THE SUBNETWORKS THAT HAVE ONLY SOURCE AND SINK NODES      535
C                                                                          536
333   SUMARC=1                                                             537
      L=L+1                                                                538
      SUBNET(1,NSUB)=CHECK(L)                                              539
      GO TO 343                                                            540
33    CONTINUE                                                             541
219   INSUB=NUMBD                                                          542
      RETURN                                                               543
      END                                                                  544
      SUBROUTINE CUT (ISUBN,LNODEN)                                        545
C                                                                          546
C          CUT IDENTIFIES CUT NODES EXCLUDING THE DESIGNATED SOURCE AND    547
C          SINK                                                            548
```

```
C                                                                       549
C       CUT ALSO IDENTIFIES THE CUT GROUPS; THAT IS, THE SUBNETWORKS    550
C       WHICH ARE IN SERIES AND CONNECTED BY THE CUT NODES              551
C                                                                       552
      IMPLICIT INTEGER*2 (A-Z)                                          553
      COMMON SUBNET,S,T,SOURC,SINKS,NARCSS,NSUB,TNSUB                   554
      INTEGER*2 S(1000),T(1000),SUBNET(1000,100),SOURC(100),SINKS(100),555
     *NARCSS(100)                                                       556
      INTEGER*2 TARC(1000),ORIGIN(1000),POST(1000),RCUT(100),PPOST(1000)557
C                                                                       558
C       FIND THE CUT NODES                                              559
C       NCUT IS THE NUMBER OF CUT NODES FOUND THUS FAR                  560
C                                                                       561
      NCUT=0                                                            562
      ARCS=NARCSS(TSUBN)                                                563
      SOURCE=SOURC(TSUBN)                                               564
      SINK=SINKS(TSUBN)                                                 565
C                                                                       566
C       THE DO LOOP DOWN TO STATEMENT NUMBER 1 DETERMINES THE CUT NODES 567
C                                                                       568
      DO 1  K=1,LNODEN                                                  569
C                                                                       570
C       CHECK TO SEE IF NODE K IS ACTUALLY IN THE SUBNETWORK            571
C                                                                       572
      DO 20 J=1,ARCS                                                    573
      Z=SUBNET(J,TSUBN)                                                 574
      IF(S(Z).EQ.K) GO TO 21                                            575
      IF (T(Z).EQ.K) GO TO 21                                           576
20    CONTINUE                                                          577
C                                                                       578
C       NODE K IS NOT IN THIS SUBNETWORK                                579
C                                                                       580
      GO TO 1                                                           581
21    CONTINUE                                                          582
C                                                                       583
C       NODE K IS IN THIS SUBNETWORK                                    584
C                                                                       585
      IF(K.EQ.SOURCE) GO TO 1                                           586
      IF(K.EQ.SINK) GO TO 1                                             587
      NTARC=0                                                           588
      NPPOST=1                                                          589
      DO 2 J=1,ARCS                                                     590
      Z=SUBNET(J,TSUBN)                                                 591
      IF (S(Z).EQ.K) GO TO 2                                            592
      IF (T(Z).EQ.K) GO TO 2                                            593
      NTARC=NTARC+1                                                     594
      TARC(NTARC)=SUBNET(J,TSUBN)                                       595
2     CONTINUE                                                          596
C                                                                       597
C       TARC IS THE SUBNETWORK WITHOUT THE ARCS INVOLVING NODE K        598
C       IF TARC CONTAINS A PATH FROM THE SOURCE TO THE SINK, THEN NODE  599
C       K IS NOT A CUT NODE                                             600
C       OTHERWISE, K IS A CUT NODE                                      601
C                                                                       602
      ORIGIN(1)=SOURCE                                                  603
      PPOST(1)=SOURCE                                                   604
      NORIG=1                                                           605
11    CONTINUE                                                          606
      NPOST=0                                                           607
C                                                                       608
C       IF THERE ARE NO ARCS IN THE TARC ARRAY, K IS A CUT NODE         609
```

```
C                                                                       610
      IF (NTARC.EQ.0) GO TO 44                                          611
C                                                                       612
C        FIND ALL NODES WHICH COME AFTER AN ORIGIN:  PUT THEM IN POST    613
C        AND PPOST IF THEY ARE NOT PAST POSTS                           614
C                                                                       615
      DO 4 I=1,NORIG                                                    616
      Y=ORIGIN(I)                                                       617
      DO 5 J=1,NTARC                                                    618
      Z=TARC(J)                                                         619
      U=S(Z)                                                            620
      V=T(Z)                                                            621
      IF (U.NE.Y) GO TO 5                                               622
C                                                                       623
C        IF WE'VE REACHED THE SINK, NODE K IS NOT A CUT NODE            624
C                                                                       625
C                                                                       626
      IF (V.EQ.SINK) GO TO 1                                            627
C                                                                       628
C        IF V IS A PAST POST, LET'S IGNORE IT                           629
C                                                                       630
      DO 49 L=1,NPPOST                                                  631
      IF (V.EQ.PPOST(L)) GO TO 5                                        632
 49   CONTINUE                                                          633
      NPOST=NPOST+1                                                     634
      POST(NPOST)=V                                                     635
      NPPOST=NPPOST+1                                                   636
      PPOST(NPPOST)=V                                                   637
      GO TO 5                                                           638
5     CONTINUE                                                          639
4     CONTINUE                                                          640
C                                                                       641
C        IF THERE ARE NOW NO POSTS, NODE K IS A CUT NODE                642
C                                                                       643
44    IF(NPOST.NE.0) GO TO 13                                           644
      NCUT=NCUT+1                                                       645
      KCUT(NCUT)=K                                                      646
      GO TO 1                                                           647
13    NORIG=NPOST                                                       648
      DO 14 L=1,NORIG                                                   649
C                                                                       650
C        THESE ARE NOW OUR NEW ORIGINS                                  651
C                                                                       652
14    ORIGIN(L)=POST(L)                                                 653
C                                                                       654
C        CHECK THE NEW ORIGINS FOR THEIR POSTS                          655
C                                                                       656
      GO TO 11                                                          657
1     CONTINUE                                                          658
      IF (NCUT.EQ.0) GO TO 32                                           659
      NSUB=NSUB+1                                                       660
      SOURC(NSUB)=SOURCE                                                661
33    ORIGIN(1)=SOURCE                                                  662
C                                                                       663
C        NOW WE NEED TO FIND THE COMPONENTS OF THE SERIES SUBNETWORKS   664
C        THAT ARE SEPARATED BY THE CUT NODES                            665
C                                                                       666
39    SUMARC=0                                                          667
      NORIG=1                                                           668
      NPPOST=1                                                          669
23    NPOST=0                                                           670
```

```
      PPOST(1)=ORIGIN(1)                                                671
      DO 24 I=1,NORIG                                                   672
      Y=ORIGIN(I)                                                       673
      DO 25 J=1,ARCS                                                    674
      Z=SUBNET(J,TSUBN)                                                 675
C                                                                       676
C        ALL ARCS BEGINNING AT THIS ORIGIN GO INTO THE NEW SUBNETWORK   677
C                                                                       678
      IF (S(Z).NE.Y) GO TO 25                                           679
      SUMARC=SUMARC+1                                                   680
      SUBNET (SUMARC,NSUB)=Z                                            681
C                                                                       682
C        T(Z) WILL BE A POST IF IT ISN'T A REPEAT OF A CURRENT          683
C        ORIGIN OR A CUT NODE                                           684
C                                                                       685
C        CHECK TO SEE IF IT IS A REPEAT                                 686
C                                                                       687
      X=T(Z)                                                            688
      DO 30 K=1,NORIG                                                   689
C                                                                       690
C        IF T(Z) IS A REPEAT OF A CURRENT ORIGIN, LET'S IGNORE IT       691
C                                                                       692
      IF(X.EQ.ORIGIN(K)) GO TO 25                                       693
30    CONTINUE                                                          694
C                                                                       695
C        CHECK TO SEE IF T(Z) IS A CUT NODE.  IF IT IS, DON'T PUT IT    696
C        IN THE POST OR PPOST ARRAYS                                    697
C                                                                       698
      DO 52 W=1,NCUT                                                    699
      D=RCUT(W)                                                         700
      IF (X.NE.D) GO TO 52                                              701
C                                                                       702
C        THIS CUT NODE IS THE SINK OF THE SUBNETWORK UNDER              703
C        CONSIDERATION AND THE SOURCE OF THE NEXT SUBNETWORK TO BE      704
C        CONSIDERED.                                                    705
C                                                                       706
      SINKS(NSUB)=D                                                     707
      SOURC(NSUB+1)=D                                                   708
      GO TO 25                                                          709
52    CONTINUE                                                          710
C                                                                       711
C        IF T(Z) IS A REPEAT OF A PAST POST, LET'S IGNORE IT            712
C                                                                       713
      DO 7 K=1,NPPOST                                                   714
      IF (X.EQ.PPOST(K)) GO TO 25                                       715
7     CONTINUE                                                          716
      NPOST=NPOST+1                                                     717
      POST(NPOST)=X                                                     718
      NPPOST=NPPOST+1                                                   719
      PPOST(NPPOST)=X                                                   720
25    CONTINUE                                                          721
24    CONTINUE                                                          722
C                                                                       723
C        IF WE HAVE NO POSTS LEFT, WE HAVE FOUND ALL OF THIS SUBNETWORK 724
C                                                                       725
      IF (NPOST.EQ.0) GO TO 34                                          726
      NORIG=NPOST                                                       727
      DO 28 L=1,NORIG                                                   728
28    ORIGIN(L)=POST(L)                                                 729
      GO TO 23                                                          730
34    NARCSS(NSUB)=SUMARC                                               731
```

```
      NSUB=NSUB+1                                                          732
      X=SOURC(NSUB)                                                        733
C                                                                          734
C        IF THE SOURCE OF NSUB IS NOT A CUT NODE, WE NEED TO ADJUST NSUB   735
C        AND GO BACK TO THE MAIN PROGRAM FOR THE NEXT STAGE OF THE         736
C        BREAKUP                                                           737
C                                                                          738
      IF (X.EQ.0) GO TO 31                                                 739
      ORIGIN(1)=X                                                          740
      GO TO 39                                                             741
31    NSUB=NSUB-1                                                          742
      SINKS(NSUB)=SINK                                                     743
32    TNSUB=NCUT+1                                                         744
      RETURN                                                               745
      END                                                                  746
      SUBROUTINE PUNSYN                                                    747
C        THIS SUBROUTINE PUNCHES THE CARDS NEEDED FOR SYNTHESIS            748
C        CARD(I,1) = THE INSTRUCTION NUMBER,I                              749
C        CARD(I,2) = THE DECOMPOSABLE SUBNETWORK                           750
C        CARD(I,3) = ISORP;                                                751
C             ISORP=0 IMPLIES A SERIES DECOMPOSITION                       752
C             ISORP=1 IMPLIES A PARALLEL DECOMPOSITION                     753
C        CARD(I,4...) = THE SUBNETWORKS RESULTING FROM THE                 754
C                       DECOMPOSITION OF SUBNETWORK I                      755
      IMPLICIT INTEGER*2 (A-Z)                                             756
      COMMON /NEW/ CARD,INSTN                                              757
      INTEGER CARD(100,27),WHYNOT                                          758
      INTEGER                                F5/13/                        759
      WHYNOT = INSTN                                                       760
      WRITE(F5) WHYNOT                                                     761
      WRITE(F5) CARD                                                       762
      ENDFILE F5                                                           763
      RETURN                                                               764
      END                                                                  765
      SUBROUTINE PRESUB(TSUBN)                                             766
C                                                                          767
C        PREPARES THE SUBNETWORK FOR SUBNETWORK ANALYSIS                   768
C             INCLUDES RENUMBERING THE NODES 1,2,.... CONSECUTIVELY.       769
C                                                                          770
      IMPLICIT INTEGER*2 (A-Z)                                             771
      COMMON SUBNET,S,T,SOURC,SINKS,NARCSS,NSUB,TNSUB                      772
      INTEGER*2 S(1000),T(1000),SUBNET(1000,100),SOURC(100),SINKS(100),    773
     *NARCSS(100)                                                          774
      INTEGER*2 HEAD(1000),TAIL(1000),NODE(1000)                           775
      COMMON/PREP/ NUMSUB                                                  776
      INTEGER FILE3(1503),NET(3,500),N3                                    777
      INTEGER                        F3/11/                                778
      EQUIVALENCE (FILE3(4),NET(1))                                        779
      NUMSUB=NUMSUB+1                                                      780
C                                                                          781
C        TSUBN=THE NUMBER OF THE MINIMUM SUBNETWORK                        782
C                                                                          783
C                                                                          784
C        RENUMBER THE NODES IN THE SUBNETWORK                              785
C                                                                          786
      N=NARCSS(TSUBN)                                                      787
      DO 1 I=1,N                                                           788
      J=SUBNET(I,TSUBN)                                                    789
      NODE(2*I-1) = S(J)                                                   790
    1 NODE(2*I)   = T(J)                                                   791
C                                                                          792
```

```
C-----SORT THE NODES                                                    793
C                                                                       794
  100 LIMIT = 2*N                                                       795
      IPASS = 2*N-1                                                     796
      IFLG = 1                                                          797
      DO 2 J=1,IPASS                                                    798
      LIMIT = LIMIT-1                                                   799
      IF(IFLG.EQ.0) GO TO 3                                             800
      IFLG = 0                                                          801
      DO 4 K=1,LIMIT                                                    802
      IF (NODE(K).LE.NODE(K+1)) GO TO 4                                 803
      KEEP = NODE(K+1)                                                  804
      NODE(K+1) = NODE(K)                                               805
      NODE(K) = KEEP                                                    806
      IFLG = 1                                                          807
    4 CONTINUE                                                          808
    2 CONTINUE                                                          809
C                                                                       810
C-----DISCARD ANY DUPLICATIONS IN THE SORTED LIST                       811
C                                                                       812
    3 KOUNT=1                                                           813
      DO 5 I=1,IPASS                                                    814
      IF (NODE(I).EQ.NODE(I+1)) GO TO 5                                 815
      KOUNT = KOUNT+1                                                   816
      NODE(KOUNT) = NODE(I+1)                                           817
    5 CONTINUE                                                          818
C                                                                       819
C-----CHANGE INPUT-HEAD,TAIL                                            820
C                                                                       821
      DO 8 L=1,N                                                        822
      I=SUBNET(L,TSUBN)                                                 823
      DO 9 J=1,KOUNT                                                    824
      IF (   T(I).NE.NODE(J)) GO TO 9                                   825
      HEAD(L)=J                                                         826
      GO TO 8                                                           827
    9 CONTINUE                                                          828
    8 CONTINUE                                                          829
      DO 11 L=1,N                                                       830
      I=SUBNET(L,TSUBN)                                                 831
      DO 12 J=1,K                                                       832
      IF (   S(I).NE.NODE(J)) GO TO 12                                  833
      TAIL(L)=J                                                         834
      GO TO 11                                                          835
   12 CONTINUE                                                          836
   11 CONTINUE                                                          837
      FILE3(1) = N                                                      838
      FILE3(2) = TSUBN                                                  839
      FILE3(3) = KOUNT                                                  840
      DO 10 J=1,N                                                       841
      NET(1,J) = HEAD(J)                                                842
      NET(2,J) = TAIL(J)                                                843
10    NET(3,J) = SUBNET(J,TSUBN)                                        844
C                                                                       845
C        OUTPUT                                                         846
C                                                                       847
      WRITE(F3) FILE3(1)                                                848
      N3 = 2 + 3*N                                                      849
      CALL WRT(F3,FILE3(2),N3)                                          850
      WRITE(6,200)                                                      851
200   FORMAT(1H0,/16X,'THE FOLLOWING REPRESENTATION OF THE SUBNETWORK AB  852
     *OVE WILL BE USED BY THE SUBNETWORK ANALYSIS PROGRAM:')             853
```

```
      WRITE(6,30)                                                        854
30    FORMAT(1H0,19X,'ACTIVITY',4X,'TAIL',6X,'HEAD')                     855
      WRITE(6,31)                (I,TAIL(I),HEAD(I),I=1,N)               856
31    FORMAT(20X,I5,5X,I5,5X,I5)                                         857
      WRITE(6,7005)                                                      858
7005  FORMAT(1H1)                                                        859
      RETURN                                                             860
      END                                                                861
      SUBROUTINE WRT(J,A,N)                                              862
      DIMENSION A(N)                                                     863
      WRITE(J) A                                                         864
      RETURN                                                             865
      END                                                                866
      SUBROUTINE RD(J,A,N)                                               867
      DIMENSION A(N)                                                     868
      READ(J) A                                                          869
      RETURN                                                             870
      END                                                                871
```

7. SUBNET

```
C        SUBNETWORK ANALYSIS PROGRAM                                          1
C                                                                             2
C        THIS PROGRAM DETERMINES THE DURATION DISTRIBUTION FOR EACH           3
C        SUBNETWORK IDENTIFIED BY THE DECOMPOSITION PROGRAM.                  4
C                                                                             5
C        FOR THE SAKE OF  IDENTIFYING THE APPROPRIATE DIMENSIONS, LET         6
C        MMAX = THE MAXIMUM NUMBER OF ACTIVITIES IN THE ORIGINAL              7
C                  PROJECT NETWORK                                            8
C        LMAX = THE MAXIMUM NUMBER OF ACTIVITIES ON A CRITICAL PATH           9
C                  LMAX IS LESS THAN OR EQUAL TO MSMAX                       10
C        CMAX = THE MAXIMUM NUMBER OF ACTIVITIES IN A CLUSTER                 11
C                  CMAX IS LESS THAN OR EQUAL TO MSMAX                       12
C        MSMAX = THE MAXIMUM NUMBER OF ACTIVITIES ALLOWED IN ANY ONE          13
C                  SUBNETWORK                                                14
C        NSUBMAX = THE MAXIMUM NUMBER OF SUBNETWORKS ALLOWED IN THE           15
C                  DECOMPOSITION PROCESS                                     16
C        IEDFMAX = THE MAXIMUM NUMBER OF SUBDIVISIONS ALLOWED IN THE          17
C                  APPROXIMATE CDF FOR EACH SUBNETWORK                       18
C        SDIM = 12 * (IEDFMAX-1) + 2                                         19
C        CURRENTLY, MMAX=1000; LMAX=50; CMAX=100; MSMAX=500; NSUBMAX=100     20
C                  IEDFMAX=20; SDIM=230                                      21
C                                                                            22
C     DIMENSION CTIME(MSMAX),DTIME(MSMAX),AORD(MSMAX),TAIL(MSMAX),           23
C    *          HEAD(MSMAX),XNODE(MSMAX),IBB(LMAX),KB(LMAX),ICRITP(LMAX)    24
C    *          COT(MSMAX),SIGMA(MSMAX),NINCL(LMAX),INCLUS(LMAX,CMAX),      25
C    *          FD(IEDFMAX),ULEFD(IEDFMAX),LLEFD(IEDFMAX),FHI(MSMAX),       26
C    *          FLO(MSMAX),PP(MSMAX),PQ(MSMAX),FLHAT(IEDFMAX),              27
C    *          FUHAT(IEDFMAX),FDLHAT(3,IEDFMAX),FDUHAT(3,IEDFMAX),         28
C    *          FHAT(IEDFMAX),BRHS1(SDIM),INBASE(SDIM),XB1(SDIM),           29
C    *          Y1(SDIM),B1INV(SDIM,SDIM),FILE3(3*MSMAX+2),NET(3,MSMAX),    30
C    *          LUP(3,MMAX)                                                 31
C                                                                            32
C        IN SUBROUTINE ORDER THE ARRAYS ARE:                                 33
C     DIMENSION ND(MSMAX),NDD(MSMAX)                                         34
C                                                                            35
C        IN SUBROUTINE CLUSTR THE ARRAYS ARE:                                36
C     DIMENSION LEFT(MSMAX),LEFTO(MSMAX),NONCP(MSMAX),NINAG(MSMAX),          37
C    *          NCLINC(LMAX),ASSGRP(LMAX,LMAX),CLINCL(LMAX,LMAX),           38
C    *          EGRP(LMAX)                                                  39
C                                                                            40
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .    41
      IMPLICIT REAL*8 (A-H,O-Z)                                              42
      COMMON /BLKA/CTIME,DTIME,M                                             43
      DIMENSION CTIME(0500),DTIME(0500)                                      44
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .    45
      COMMON /BLAA/XNODE,TAIL,HEAD,AORD,NMM                                  46
      INTEGER AORD(500),TAIL(500),HEAD(500)                                  47
      DIMENSION XNODE(500)                                                   48
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .    49
C     COMMON /BLKB/IBB,KB,KKB,KKK                                            50
C     DIMENSION IBB(50),KB(50)                                               51
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .    52
      COMMON /BLKC/ICRITP,KCPB                                               53
      DIMENSION ICRITP(50)                                                   54
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .    55
      COMMON /BLKD/SIGMA,COT                                                 56
      DIMENSION COT(500),SIGMA(500)                                          57
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .    58
      COMMON /BLKE/INCLUS,NINCL,NCLUS                                        59
      DIMENSION NINCL(50),INCLUS(50,100)                                     60
```

```
C . . . . . .                                                          61
      COMMON /BLKG/FD,ULEFD,LLEFD                                      62
      REAL*8 FD(20),ULEFD(20),LLEFD(20)                                63
C . . . . . .                                                          64
      COMMON /BLKH/FHI,FLO,PP,PQ                                       65
      DIMENSION FHI(500),FLO(500),PP(500),PQ(500)                      66
C . . . . . .                                                          67
      COMMON /BLKI/IEDF,NSUB,SAMSIZ                                    68
      INTEGER SAMSIZ                                                   69
C . . . . . .                                                          70
      COMMON /BLKJ/THET,LAMBD                                          71
      REAL*8 LAMBD                                                     72
C . . . . . .                                                          73
      COMMON /BLKK/IR,IB                                               74
C . . . . . .                                                          75
      COMMON /BLKL/FLHAT,FUHAT                                         76
      DIMENSION FLHAT(20),FUHAT(20)                                    77
C . . . . . .                                                          78
      COMMON /BLKS/SUM,ICLGIB                                          79
C . . . . . .                                                          80
      COMMON /BLKY/FDUHAT,FDLHAT,FHAT                                  81
      DIMENSION FDLHAT(3,20),FDUHAT(3,20),FHAT(20)                     82
C . . . . . .                                                          83
      COMMON /BLKZ/THETA                                               84
      REAL*4 THETA(3)                                                  85
C . . . . . .                                                          86
C     COMMON /BLK1/BIINV,OMEGA,MM,KK,MMKK,MMKK2,N,NP1,JFIRST           87
C     COMMON /BLK2/Y1                                                  88
C     DIMENSION BRHS1(230),INBASE(230),XB1(230),Y1(230),BIINV(230,230) 89
C . . . . . .                                                          90
      INTEGER THELAM                                                   91
      INTEGER FILE3(1502)                                              92
      INTEGER                    F3/11/,F4/12/,F5/13/                  93
      REAL*8 LUP(3,1000)                                               94
      DIMENSION NET(3,500)                                             95
      DIMENSION NIMPT(3)                                               96
      DIMENSION CDF(12)                                                97
      REAL*4 LAMBDA(3)                                                 98
      REAL*4 RTHETA(3),RLAMBD(3)                                       99
      REAL*8 M1,M2,M3                                                 100
      EQUIVALENCE (FILE3(3),NET(1))                                   101
      EQUIVALENCE (FILE3(6),RTHETA(1)),(FILE3(9),RLAMBD(1))           102
C . . . . . .                                                         103
C                                                                     104
      WRITE(6,90001)                                                  105
90001 FORMAT(1H1,132('*')/1X,132('*')/'0THIS IS THE OUTPUT FROM THE SUBN  106
     *ETWORK ANALYSIS PROGRAM: SUBNET'/1H0,132('*')/1X,132('*')/////   107
     *' EACH SUBNETWORK',1H',''S APPROXIMATE DURATION DISTRIBUTION IS DET 108
     *ERMINED.')                                                      109
C**********************************************************************110
C                                                                     111
C        READ IN INFORMATION PERTAINING TO ALL SUBNETWORKS            112
C                                                                     113
C**********************************************************************114
      REWIND F3                                                       115
      REWIND F5                                                       116
      READ(F5)                                                        117
      READ(F5)                                                        118
      READ(F5)                                                        119
      REWIND F4                                                       120
      READ(F4)                                                        121
```

```
      READ(F4) NACT                                                          122
C*************************************************************************  123
C                                                                            124
C         DETERMINE L,U, AND P FOR EACH ACTIVITY IN THE SIMPLIFIED           125
C         PROJECT NETWORK                                                    126
C                                                                            127
C*************************************************************************  128
      DO 10 L=1,NACT                                                         129
      READ(F4)CDF                                                            130
      M1=0.D0                                                                131
      M2=0.D0                                                                132
      M3=0.D0                                                                133
      DO 20 K=2,11                                                           134
      M1=M1+CDF(K)                                                           135
      M2=M2+CDF(K)**2                                                        136
20    M3=M3+CDF(K)**3                                                        137
      M1=M1*.1D0                                                             138
      M2=M2*.1D0                                                             139
      M3=M3*.1D0                                                             140
      T3=M1*M1-M2                                                            141
      IF(DABS(T3).GE.1.D-10)GO TO 21                                         142
      U=CDF(2)                                                               143
      B=U                                                                    144
      P=1.D0                                                                 145
      GO TO 22                                                               146
21    T1=(M3-M1*M2)**2                                                       147
      T2=4.D0*T3*(M2*M2-M1*M3)                                               148
      T3=2.D0*T3                                                             149
      U=-M3+M1*M2-DSQRT(T1-T2)                                               150
      U=U/T3                                                                 151
      B=(M2-M1*U)/(M1-U)                                                     152
      P=(M1-U)/(B-U)                                                         153
22    LUP(1,L)=B                                                             154
      LUP(2,L)=U                                                             155
      LUP(3,L)=P                                                             156
10    CONTINUE                                                               157
      CALL RD(F3,FILE3,11)                                                   158
      IEOF=FILE3(1)                                                          159
      NMAX=FILE3(2)                                                          160
      IPOOL=FILE3(3)                                                         161
      SAMSIZ=FILE3(4)                                                        162
      THELAM=FILE3(5)                                                        163
      DO 2 I=1,J                                                             164
      THETA(I)=RTHETA(I)                                                     165
2     LAMBDA(I)=RLAMBD(I)                                                    166
1     CONTINUE                                                               167
C*************************************************************************  168
C                                                                            169
C         ANALYZE EACH SUBNETWORK SEPARATELY                                 170
C                                                                            171
C                                                                            172
C         READ IN THE DESCRIPTION OF THE SUBNETWORK                          173
C                                                                            174
C*************************************************************************  175
      READ(F3,END=999) M                                                     176
      N3 = 2 + 3*M                                                           177
      CALL RD(F3,FILE3(1),N3)                                                178
      NSUB = FILE3(1)                                                        179
      NAA  = FILE3(2)                                                        180
      WRITE(6,2600) NSUB                                                     181
2600  FORMAT(1H1, 5X,'SUBNETWORK ',I5)                                       182
```

```
C         M = THE NUMBER OF ACTIVITIES IN THE SUBNETWORK                    183
C         NMM = THE NUMBER OF NODES IN THE SUBNETWORK                       184
      MP1=M+1                                                               185
C                                                                           186
C         THE ACTIVITIES ARE DESCRIBED IN TERMS OF THEIR NODES              187
C         II=THE TAIL NODE, THE ORIGIN NODE                                 188
C         JJ=THE HEAD NODE, THE TERMINAL NODE                               189
C         FLO = THE LOWER POINT                                             190
C         FHI = THE UPPER POINT                                             191
C         SIGMA  = (FHI - FLO)*DSQRT( P*(1-P) ) = STD. DEVIATION            192
C         PP = THE PROBABILITY OF THE LOWER POINT                           193
C                                                                           194
      DO 610 I=1,M                                                          195
      HEAD(I) = NET(1,I)                                                    196
      TAIL(I) = NET(2,I)                                                    197
      K       = NET(3,I)                                                    198
      FLO(I)  = LUP(1,K)                                                    199
      FHI(I)  = LUP(2,K)                                                    200
      PP(I)   = LUP(3,K)                                                    201
      PQ(I)=1.D0-PP(I)                                                      202
      SIGMA(I)=(FHI(I)-FLO(I))*DSQRT( PP(I)*PQ(I))                          203
      COT(I)=PP(I)*FLO(I)+PQ(I)*FHI(I)                                      204
      CTIME(I) = COT(I)                                                     205
610   CONTINUE                                                              206
C         COT = THE ORIGINAL RIGHT-HAND SIDES, I.E. THE MEANS               207
C               OF FLO AND FHI                                              208
C         CTIME = THE CURRENT RIGHT-HAND SIDES                              209
      WRITE(6,2700)                                                         210
2700  FORMAT(1H0,10X,'INITIAL INPUT')                                       211
      WRITE(6,2701)                                                         212
2701  FORMAT(1H0,10X,'ACTIVITY  ORIGIN    TERMINAL  LOWER POINT  UPPER P    213
     *OINT      MEAN    STANDARD DEVIATION   PROB. LOWER PT.')             214
      DO 2704 I=1,M                                                         215
2704  WRITE(6,2702) I,TAIL(I),HEAD(I),FLO(I),FHI(I),CTIME(I),SIGMA(I),PP    216
     *(I)                                                                   217
2702  FORMAT(1H ,13X,I3,5X,I3,7X,I3,5X,F10.4,3X,F10.4,3X,F10.4,4X,F10.4,    218
     *14X,F7.5)                                                             219
C*************************************************************************220
C                                                                           221
C         IF THERE IS ONLY 1 ACTIVITY IN THE SUBNETWORK, LET  FHAT  BE      222
C         THAT ACTIVITY'S DURATION DISTRIBUTION                             223
C                                                                           224
C*************************************************************************225
      IF(M.GT.1)GO TO 4                                                     226
      REWIND F4                                                             227
      I=K+1                                                                 228
      DO 25 L=1,I                                                           229
      READ(F4)                                                              230
25    CONTINUE                                                              231
      READ(F4)(  FD(I),I=1,12)                                              232
      JEDF=12                                                               233
      FHAT(1)=0.D0                                                          234
      FHAT(12)=1.D0                                                         235
      M1=-.05D0                                                             236
      DO 30 L=2,11                                                          237
      M1=M1+.1D0                                                            238
30    FHAT(L)=M1                                                            239
      WRITE(6,700)NSUB                                                      240
700   FORMAT(1H1,5X,'THE DURATION DISTRIBUTION FOR SUBNETWORK ',I2)         241
      WRITE(6,108)                                                          242
      DO 701 I=1,JEDF                                                       243
```

```
701   WRITE(6,103)FD(I),FHAT(I)                                          244
      GO TO 3                                                            245
C*****************************************************************     246
C                                                                        247
C        INITIAL  DETERMINES THE CRITICAL PATH WHEN EACH                 248
C        ACTIVITY'S DURATION IS ITS MEAN, LOWER BOUND, AND UPPER BOUND   249
C                                                                        250
C*****************************************************************     251
4     CALL INITAL                                                        252
C*****************************************************************     253
C                                                                        254
C        IF M IS LESS THAN OR EQUAL TO NMAX, LET  FHAT  BE THAT          255
C        ACTIVITY'S DISCRETE DURATION DISTRIBUTION                       256
C                                                                        257
C*****************************************************************     258
      IF(M.GT.NMAX)GO TO 5                                               259
      CALL FINDF                                                         260
      JEDF=IEDF                                                          261
      GO TO 3                                                            262
C*****************************************************************     263
C                                                                        264
C        DETERMINE THETA AND LAMBDA                                      265
C                                                                        266
C        IF THELAM = 0 THE DEFAULT PAIRS (THETA,LAMBDA)=(1,1),(2,2),(3,2 267
C        WILL BE USED                                                    268
C                                                                        269
C        IF THELAM IS NOT 0 THE USER-SPECIFIED (THETA,LAMBDA) PAIRS      270
C        WILL BE USED                                                    271
C                                                                        272
C        NOTE: THE FOLLOWING RELATIONS MUST ALWAYS BE TRUE               273
C              THETA(1)<=THETA(2)<=THETA(3)                              274
C              LAMBDA(1)<=LAMBDA(2)<=LAMBDA(3)                           275
C                                                                        276
C*****************************************************************     277
5     DO 500 I=1,3                                                       278
      IF(THELAM.NE.0)GO TO 222                                           279
      LAMBD=2.D0                                                         280
      IF(I.EQ.1)LAMBD=1.D0                                               281
      THET=DFLOAT(I)                                                     282
      THETA(I)=THET                                                      283
      GO TO 201                                                          284
222   THET=THETA(I)                                                      285
      LAMBD=LAMBDA(I)                                                    286
C*****************************************************************     287
C                                                                        288
C        SUBROUTINE CLUSTR  DETERMINES THE CLUSTER FOR GIVEN THETA AND   289
C        LAMBDA                                                          290
C                                                                        291
C*****************************************************************     292
201   CALL CLUSTR                                                        293
C                                                                        294
C        THE FOLLOWING DETERMINES WHETHER OR NOT THESE CLUSTERS HAVE     295
C        BEEN CONSIDERED BEFORE                                          296
C                                                                        297
      ISUM=0                                                             298
      DO 200 II=1,NCLUS                                                  299
200   ISUM=ISUM+NINCL(II)                                                300
      NIMPT(I)=ISUM                                                      301
      IF(I.EQ.1)GO TO 202                                                302
      IF(ISUM.EQ.NIMPT(I-1))GO TO 203                                    303
202   CONTINUE                                                           304
```

```
C*************************************************************************      305
C                                                                               306
C       CALCULATE THE UPPER AND LOWER BOUNDS ON THE SUBNETWORK                  307
C       DURATION DISTRIBUTION                                                   308
C                                                                               309
C*************************************************************************      310
C                                                                               311
C       ELIMINATE FROM THE CLUSTERS ANY ACTIVITIES HAVING CONSTANT              312
C       DURATION.  IF THESE ACTIVITIES ARE LEFT IN, THE TOTAL NUMBER            313
C       OF POSSIBLE ACTIVITY DURATION CONFIGURATIONS FOR THE CLUSTERS           314
C       IS ARTIFICIALLY INFLATED.                                               315
C                                                                               316
      DO 250 II=1,KCPB                                                          317
      IL=NINCL(II)                                                              318
      IF(IL.LE.1)GO TO 250                                                      319
      III=1                                                                     320
251   IF(III.GT.IL)GO TO 254                                                    321
      L=INCLUS(II,III)                                                          322
      IF(SIGMA(L).NE.0.D0)GO TO 253                                             323
      IL=IL-1                                                                   324
      IF(III.GT.IL)GO TO 254                                                    325
      DO 252 IIII=III,IL                                                        326
252   INCLUS(II,IIII)=INCLUS(II,IIII+1)                                         327
253   III=III+1                                                                 328
      GO TO 251                                                                 329
254   NINCL(II)=IL                                                              330
      IF(IL.EQ.0)NCLUS=NCLUS-1                                                  331
250   CONTINUE                                                                  332
      IF(IPOOL.EQ.1) CALL UNION                                                 333
      IF(IPOOL.EQ.0) CALL SINGLE                                                334
C*************************************************************************      335
C                                                                               336
C       SAVE THE ESTIMATES CALCULATED BY  UNION  OR  SINGLE                     337
C       FOR THE  I-TH THETA-LAMBDA PAIR                                         338
C                                                                               339
C*************************************************************************      340
203   CONTINUE                                                                  341
      WRITE(6,100) NSUB,THET,LAMBD                                              342
100   FORMAT(1H1, 5X,'BOUNDS ON THE DURATION DISTRIBUTION FOR SUBNETWORK        343
     * ',      I4,' WHEN (THETA,LAMBDA) = ( ',E15.5,' , ',E15.5,' )')          344
      WRITE(6,101)                                                              345
101   FORMAT(1H0,14X,'TIME',16X,'LOWER BOUND',9X,'UPPER BOUND',/)               346
      IFLG=0                                                                    347
      DO 300 II=1,IEDF                                                          348
      FLTEMP=FLHAT(II)                                                          349
      FUTEMP=FUHAT(II)                                                          350
      WRITE(6,103)FD(II),FLTEMP,FUTEMP                                          351
103   FORMAT(1H ,10X,3(E15.5,5X))                                               352
      IF(FLTEMP.NE.FUTEMP)IFLG=1                                                353
      FHAT(II)=FLTEMP                                                           354
      FDLHAT(I,II)=FLTEMP                                                       355
300   FDUHAT(I,II)=FUTEMP                                                       356
      IF(IPOOL.EQ.1)WRITE(6,350)                                                357
      IF(IPOOL.EQ.0)WRITE(6,351)                                                358
350   FORMAT(1H0,/11X,'THE BOUNDS WERE DETERMINED USING THE UNION-CLUSTE        359
     *R PROCEDURE.')                                                            360
351   FORMAT(1H0,/11X,'THE BOUNDS WERE DETERMINED USING THE MAXIMUM-CLUS        361
     *TER PROCEDURE.')                                                          362
      IF(ICLGIB)352,353,353                                                     363
352   WRITE(6,6645)                                                             364
6645  FORMAT(1H0,/11X,'ALL ACTIVITY DURATION CONFIGURATIONS WERE CONSIDE        365
```

```
     *REC.')                                                                366
      GO TO 500                                                             367
353   IF(IPOOL)357,357,358                                                  368
358   WRITE(6,6046)SAMSIZ                                                   369
6046  FORMAT(1H0,/11X,'ONLY ',I8,' OF THE POSSIBLE ACTIVITY DURATION CON    370
     *FIGURATIONS WERE CONSIDERED.')                                        371
      IF(ICLGIB.EQ.0)WRITE(6,105)                                           372
      IF(ICLGIB.EQ.1)WRITE(6,106)                                           373
      GO TO 500                                                             374
357   WRITE(6,6047)SAMSIZ                                                   375
6047  FORMAT(1H0,/11X,   'ONLY ',I8,' OF THE POSSIBLE ACTIVITY DURATION     376
     *CONFIGURATIONS FOR EACH CLUSTER WERE CONSIDERED.')                     377
      IF(ICLGIB.EQ.0) WRITE(6,105)                                          378
      IF(ICLGIB.EQ.1) WRITE(6,106)                                          379
105   FORMAT(1H0,10X,'SYSTEMATIC SAMPLING WAS USED.')                       380
106   FORMAT(1H0,10X,'SIMPLE RANDOM SAMPLING WAS USED.')                    381
500   CONTINUE                                                              382
C                                                                           383
C        IF THE BEST UPPER AND LOWER BOUNDS COMPUTED ABOVE ARE EQUAL,       384
C        THERE IS NO NEED TO PERFORM THE EXTRAPOLATION                      385
C                                                                           386
C        IFLG = 0 IMPLIES THE BOUNDS ARE EQUAL                              387
C        IFLG = 1 IMPLIES THE BOUNDS ARE NOT EQUAL                          388
C                                                                           389
      IF(IFLG.EQ.1)GO TO 802                                                390
801   WRITE(6,803)NSUB                                                      391
803   FORMAT(1H1,5X,'THE UPPER AND LOWER BOUNDS ON THE DISCRETE SUBNETWO     392
     *RK DURATION DISTRIBUTION ARE EQUAL.'//6X,'THUS AN ESTIMATE OF THE     393
     *DISCRETE DURATION DISTRIBUTION FOR SUBNETWORK ',I4,' IS')             394
      JEDF=IEDF                                                             395
      GO TO 804                                                             396
C*************************************************************************397
C                                                                           398
C        SUBROUTINE SIMPLX  INTERPOLATES BETWEEN THE UPPER AND LOWER        399
C        BOUNDS DETERMINED ABOVE AND COMPUTES AN ESTIMATED COMPLETION       400
C        TIME DISTRIBUTION FOR THE SUBNETWORK BEING CONSIDERED              401
C                                                                           402
C*************************************************************************403
802   CALL SIMPLX                                                           404
9     JEDF=IEDF                                                             405
      WRITE(6,107)NSUB                                                      406
107   FORMAT(1H1,5X,'AN ESTIMATE OF THE DISCRETE DURATION DISTRIBUTION F     407
     *OR SUBNETWORK ',I2)                                                   408
804   WRITE(6,108)                                                          409
108   FORMAT(1H0,14X,'TIME',16X,'DISTRIBUTION'/)                            410
      DO 9471 I=1,JEDF                                                      411
9471  WRITE(6,103)FD(I),FHAT(I)                                             412
3     CONTINUE                                                              413
      WRITE(F5)NSUB,JEDF,(FD(I),FHAT(I),I=1,JEDF)                           414
      GO TO 1                                                               415
999   WRITE(6,850)                                                          416
850   FORMAT(1H1)                                                           417
      ENDFILE F5                                                            418
      STOP                                                                  419
      END                                                                   420
      SUBROUTINE RD(J,A,N)                                                  421
      DIMENSION A(N)                                                        422
      READ(J) A                                                             423
      RETURN                                                                424
      END                                                                   425
      SUBROUTINE INITAL                                                     426
```

```
C                                                                       427
C         INITIAL  DETERMINES THE CRITICAL PATH WHEN EACH               428
C         ACTIVITY'S DURATION IS ITS MEAN, LOWER BOUND, AND UPPER BOUND 429
C                                                                       430
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .   431
      IMPLICIT REAL*8 (A-H,O-Z)                                         432
      COMMON /BLKA/CTIME,DTIME,M                                        433
      DIMENSION CTIME(0500),DTIME(0500)                                 434
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .   435
      COMMON /BLKB/IBB,KB,KKB,KKK                                       436
      DIMENSION IBB(50),KB(50)                                          437
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .   438
      COMMON /BLKC/ICRITP,KCPB                                          439
      DIMENSION ICRITP(50)                                              440
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .   441
      COMMON /BLKG/FD,ULEFD,LLEFD                                       442
      REAL*8 FD(20),ULEFD(20),LLEFD(20)                                 443
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .   444
      COMMON /BLKH/FHI,FLO,PP,PQ                                        445
      DIMENSION FHI(500),FLO(500),PP(500),PQ(500)                       446
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .   447
      COMMON /BLKI/IEDF,NSUB,SAMSIZ                                     448
      INTEGER SAMSIZ                                                    449
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .   450
      CALL ORDER                                                        451
      TOTAL=CPTIME(CPATHT)                                              452
      CALL FPATH                                                        453
C                                                                       454
C         ICRITP(L)= THE L-TH ACTIVITY ON THE CRITICAL PATH             455
C          KCPB= THE NUMBER OF ACTIVITIES ON THE CRITICAL PATH          456
C                                                                       457
      KCPB=KKB                                                          458
      DO 2802 I=1,KCPB                                                  459
2802  ICRITP(I)=IBB(I)                                                  460
      X=TOTAL                                                           461
      WRITE(6,851) X                                                    462
851   FORMAT(1H0,5X,'THE CRITICAL PATH TIME WHEN EACH ACTIVITY''S COMPLE 463
     *TION TIME IS SET EQUAL TO ITS MEAN IS = ',D15.5)                  464
      WRITE(6,7606) KKK                                                 465
 7606 FORMAT(1H0,10X,'THE ',I3,' NODES ON THE CRITICAL PATH ARE AS FOLLO 466
     *WS BEGINNING WITH THE TERMINAL NODE:')                            467
      WRITE(6,7707) (KB(I),I=1,KKK)                                     468
 7707 FORMAT(15X,20(I3,','))                                            469
      WRITE(6,7710) KKB                                                 470
7710  FORMAT(1H0,10X,'THE ',I3,' CRITICAL ACTIVITIES ARE AS FOLLOWS BEGI 471
     *NNING WITH THE TERMINAL ACTIVITY:')                               472
      WRITE(6,7707) (IBB(I),I=1,KKB)                                    473
C                                                                       474
C         THE COMPLETION TIME FOR ALL ACTIVITIES IS SET TO THEIR LOWER  475
C         PERCENTILE.  THE RESULTING LONGEST PATH TIME IS A LOWER       476
C         BOUND ON THE SUBNETWORK DURATION TIME.                        477
C                                                                       478
      DO 6402 I=1,M                                                     479
 6402 CTIME(I) = FLO(I)                                                 480
      CPLB=CPTIME(CPATHT)                                               481
      CALL FPATH                                                        482
      WRITE(6,6405) CPLB                                                483
6405  FORMAT(1H0,5X,'A LOWER BOUND ON THE SUBNETWORK DURATION TIME IS =  484
     *',E15.5)                                                          485
      WRITE(6,8606) KKK                                                 486
8606  FORMAT(1H0,10X,'THE ',I3,' NODES ON THE LONGEST PATH ARE AS FOLLOW 487
```

```
     *S BEGINNING WITH THE TERMINAL NODE:')                                  488
      WRITE(6,7707) (KB(I),I=1,KKK)                                         489
      WRITE(6,8710) KKB                                                     490
8710  FORMAT(1H0,10X,'THE ',I3,' ACTIVITIES ON THE LONGEST PATH ARE AS F     491
     *OLLOWS BEGINNING WITH THE TERMINAL ACTIVITY:')                         492
      WRITE(6,7707) (IBB(I),I=1,KKB)                                        493
C                                                                           494
C        THE COMPLETION TIME FOR ALL ACTIVITIES IS SET TO THEIR UPPER       495
C        PERCENTILE.  THE RESULTING LONGEST PATH TIME IS AN UPPER BOUND     496
C        ON THE SUBNETWORK DURATION TIME.                                   497
C                                                                           498
      DO 6406 I=1,M                                                         499
 6406 CTIME(I) = FHI(I)                                                     500
      CPUB=CPTIME(CPATHT)                                                   501
      CALL FPATH                                                            502
      WRITE(6,6409) CPUB                                                    503
6409  FORMAT(1H0,5X,'A UPPER BOUND ON THE SUBNETWORK DURATION TIME IS =     504
     *',E15.5)                                                              505
      WRITE(6,8606) KKK                                                     506
      WRITE(6,7707) (KB(I),I=1,KKK)                                         507
      WRITE(6,8710) KKB                                                     508
      WRITE(6,7707) (IBB(I),I=1,KKB)                                        509
C                                                                           510
C        FD(I) = THE LOWER BOUND ON THE SUBNETWORK DURATION TIME            511
C                PLUS THE I-TH FRACTION OF THE DISTANCE TO THE UPPER        512
C                BOUND                                                      513
C                                                                           514
C        FD  IS USED TO BUILD AN 'EMPIRICAL' DISTRIBUTION OF THE            515
C        SUBNETWORK DURATION TIMES                                          516
C                                                                           517
      C=CPUB-CPLB                                                           518
      CC=.8D0*C                                                             519
      LEDF=.6D0*IEDF                                                        520
      CD=CC/LEDF                                                            521
      DO 6412 I=1,LEDF                                                      522
6412  FD(I)=CPLB+I*CD                                                       523
      CD=(C-CC)/(IEDF-LEDF)                                                 524
      II=LEDF+1                                                             525
      C=CPLB+CC                                                             526
      DO 6512 I=II,IEDF                                                     527
6512  FD(I)=C+(I-LEDF)*CD                                                   528
      RETURN                                                                529
      END                                                                   530
      SUBROUTINE ORDER                                                      531
C        THIS SUBROUTINE DETERMINES THE ORDER IN WHICH TO CONSIDER          532
C        THE ACTIVITIES FOR THE CALCULATION OF THE LONGEST  PATH TIME       533
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .   534
      IMPLICIT REAL*8 (A-H,O-Z)                                             535
      COMMON /BLKA/CTIME,DTIME,M                                            536
      DIMENSION CTIME(0500),DTIME(0500)                                     537
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .   538
      COMMON /BLAA/XNODE,TAIL,HEAD,ACRD,NMM                                 539
      INTEGER AORD(500),TAIL(500),HEAD(500)                                 540
      DIMENSION XNODE(500)                                                  541
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .   542
      DIMENSION ND(500),NDD(500)                                            543
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .   544
      N=NMM                                                                 545
      NDD(1)=1                                                              546
      DO 5 I=2,N                                                            547
5     NDD(I)=0                                                              548
```

```
      DO 6 I=1,M                                                          549
6     ACRD(I)=0                                                           550
      K=0                                                                 551
      MP=M+1                                                              552
      DO 1 II=1,MP                                                        553
      DO 20 I=1,N                                                         554
20    ND(I)=NDD(I)                                                        555
      III=0                                                               556
      IP=II+1                                                             557
      DO 2 J=1,M                                                          558
      IF(ND(TAIL(J)).NE.II) GO TO 2                                       559
      NDD(HEAD(J))=IP                                                     560
      III=1                                                               561
      IF(K.EQ.0) GO TO 14                                                 562
      DO 10 L=1,K                                                         563
      IF(ACRD(L).EQ.J) GO TO 11                                           564
10    CONTINUE                                                            565
14    K=K+1                                                               566
      GO TO 13                                                            567
   11 IF(L.EQ.K) GO TO 2                                                  568
      KM=K-1                                                              569
      DO 12 LL=L,KM                                                       570
12    ACRD(LL)=ACRD(LL+1)                                                 571
13    ACRD(K)=J                                                           572
2     CONTINUE                                                            573
      IF(III.EQ.0) GO TO 3                                                574
1     CONTINUE                                                            575
3     CONTINUE                                                            576
      WRITE(6,70)                                                         577
70    FORMAT(1H1,5X,'THE ORDER IN WHICH TO CONSIDER THE ACTIVITIES TO DE  578
     *TERMINE THE LONGEST PATH TIME:')                                    579
      WRITE(6,71)                                                         580
71    FORMAT(1H0,10X,'ORDER  ACTIVITY')                                   581
      DO 50 I=1,M                                                         582
50    WRITE(6,51) I,ACRD(I)                                               583
51    FORMAT(1H ,10X,I5,5X,I5)                                            584
      RETURN                                                              585
      END                                                                 586
      FUNCTION CPTIME(CPATHT)                                             587
C        DETERMINE THE CRITICAL PATH TIME: CPTIME                         588
C        XNODE(I) = EARLIEST TIME THAT AN ACTIVITY BEGINNING AT NODE I    589
C                 CAN COMMENCE                                            590
C                                                                         591
      IMPLICIT REAL*8 (A-H,O-Z)                                           592
C . . . . . .                                                             593
      COMMON /BLKA/CTIME,DTIME,M                                          594
      DIMENSION CTIME(0500),DTIME(0500)                                   595
C . . . . . .                                                             596
      COMMON /BLAA/XNODE,TAIL,HEAD,ACRD,NMM                               597
      INTEGER ACRD(500),TAIL(500),HEAD(500)                               598
      DIMENSION XNODE(500)                                                599
C . . . . . .                                                             600
      DO 1 I=1,NMM                                                        601
1     XNODE(I)=0.D0                                                       602
      DO 2 II=1,M                                                         603
      I=ACRD(II)                                                          604
2     XNODE(HEAD(I))=DMAX1(XNODE(TAIL(I))+CTIME(I),XNODE(HEAD(I)))        605
      I=HEAD(ACRD(M))                                                     606
      CPTIME=XNODE(I)                                                     607
      RETURN                                                              608
      END                                                                 609
```

```
      FUNCTION CLTIME(CPATHT)                                              610
C        DETERMINE THE CRITICAL PATH TIME: CLTIME                          611
C        XNODE(I) = EARLIEST TIME THAT AN ACTIVITY BEGINNING AT NODE I     612
C                CAN COMMENCE                                              613
C                                                                          614
      IMPLICIT REAL*8 (A-H,O-Z)                                            615
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  616
      COMMON /BLKA/CTIME,DTIME,M                                           617
      DIMENSION CTIME(0500),DTIME(0500)                                    618
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  619
      COMMON /BLAA/XNODE,TAIL,HEAD,AORD,NMM                                620
      INTEGER AORD(500),TAIL(500),HEAD(500)                                621
      DIMENSION XNODE(500)                                                 622
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  623
      DO 1 I=1,NMM                                                         624
1     XNODE(I)=0.D0                                                        625
      DO 2 II=1,M                                                          626
      I=AORD(II)                                                           627
2     XNODE(HEAD(I))=DMAX1(XNODE(TAIL(I))+DTIME(I),XNODE(HEAD(I)))         628
      I=HEAD(AORD(M))                                                      629
      CLTIME=XNODE(I)                                                      630
      RETURN                                                               631
      END                                                                  632
      SUBROUTINE FPATH                                                     633
C                                                                          634
C        SUBROUTINE FPATH  FINDS THE CRITICAL PATH                         635
C                                                                          636
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  637
      IMPLICIT REAL*8 (A-H,O-Z)                                            638
      COMMON /BLKA/CTIME,DTIME,M                                           639
      DIMENSION CTIME(0500),DTIME(0500)                                    640
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  641
      COMMON /BLAA/XNODE,TAIL,HEAD,AORD,NMM                                642
      INTEGER AORD(500),TAIL(500),HEAD(500)                                643
      DIMENSION XNODE(500)                                                 644
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  645
      COMMON /BLKB/IBB,KB,KKB,KKK                                          646
      DIMENSION IBB(50),KB(50)                                             647
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  648
C                                                                          649
C        KKK= THE NUMBER OF NODES ON THE CRITICAL PATH                     650
C        KB(L)= THE L-TH NODE IN THE CRITICAL PATH, COUNTING BACKWARDS     651
C                  FROM THE TERMINAL NODE                                  652
C        KKB= THE NUMBER OF ACTIVITIES ON THE CRITICAL PATH                653
C        IBB(L)= THE L-TH ACTIVITY ON THE CRITICAL PATH, COUNTING          654
C                  BACKWARDS FROM THE TERMINAL NODE                        655
C                                                                          656
C                                                                          657
C        THE FOLLOWING STATEMENTS DETERMINE THE NODES AND ACTIVITIES ON    658
C        THE CRITICAL PATH                                                 659
C                                                                          660
      KKK=1                                                                661
      KB(1)=NMM                                                            662
83001 IK=KB(KKK)                                                           663
      SMIN=999999.                                                         664
      ISMIN=0                                                              665
      DO 83000 I=1,M                                                       666
      IF(HEAD(I).NE.IK) GO TO 83000                                        667
      SLACK=XNODE(HEAD(I))-XNODE(TAIL(I))-CTIME(I)                         668
      IF(SLACK.GE.SMIN) GO TO 83000                                        669
      SMIN=SLACK                                                           670
```

```
      ISMIN=I                                                          671
83000 CONTINUE                                                         672
83003 IBB(KKK)=ISMIN                                                   673
      KKK=KKK+1                                                        674
      KB(KKK)=TAIL(ISMIN)                                              675
      IF(TAIL(ISMIN).GT.1) GO TO 83001                                 676
      KKB=KKK-1                                                        677
      RETURN                                                           678
      END                                                              679
      SUBROUTINE FINDF                                                 680
C                                                                      681
C        SUBROUTINE FINDF EVALUATES THE SUBNETWORK'S DISCRETE DURATION 682
C        DISTRIBUTION BY EXPLICITLY EVALUATING ALL OF THE SUBNETWORK'S 683
C        ACTIVITY DURATION CONFIGURATIONS                              684
C                                                                      685
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  686
      IMPLICIT REAL*8(A-H,O-Z)                                         687
      COMMON /BLKA/CTIME,DTIME,M                                       688
      DIMENSION CTIME(0500),DTIME(0500)                                689
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  690
      COMMON /BLKE/INCLUS,NINCL,NCLUS                                  691
      DIMENSION NINCL(50),INCLUS(50,100)                               692
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  693
      COMMON /BLKG/FD,ULEFD,LLEFD                                      694
      REAL*8 FD(20),ULEFD(20),LLEFD(20)                                695
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  696
      COMMON /BLKI/IEDF,NSUB,SAMSIZ                                    697
      INTEGER SAMSIZ                                                   698
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  699
      COMMON /BLKK/IR,IB                                               700
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  701
      COMMON /BLKS/SUM,ICLGIB                                          702
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  703
      COMMON /BLKY/FDUHAT,FDLHAT,FHAT                                  704
      DIMENSION FDLHAT(3,20),FDUHAT(3,20),FHAT(20)                     705
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  706
      IR=1                                                             707
C*********************************************************************** 708
C                                                                      709
C        FORM A CLUSTER CONTAINING ALL OF THE ACTIVITIES IN THE        710
C        SUBNETWORK                                                    711
C                                                                      712
C*********************************************************************** 713
      NINCL(1)=M                                                       714
      DO 1501 I=1,M                                                    715
1501  INCLUS(1,I)=I                                                    716
      DO 7101 I=1,IEDF                                                 717
7101  LLEFD(I)=0.D0                                                    718
      IB=2**M                                                          719
C*********************************************************************** 720
C                                                                      721
      CALL EXACTF                                                      722
C                                                                      723
C*********************************************************************** 724
      FHAT(1)=LLEFD(1)/SUM                                             725
      DO 6900 I=2,IEDF                                                 726
      II=I-1                                                           727
      LLEFD(I)=LLEFD(II)+LLEFD(I)                                      728
      FHAT(I)=LLEFD(I)/SUM                                             729
6900  CONTINUE                                                         730
      WRITE (6,6362)                                                   731
```

```
6362  FORMAT (1H1)                                                          732
      WRITE(6,100) NSUB                                                     733
100   FORMAT(1H1, 5X,'THE DISCRETE DURATION DISTRIBUTION FOR SUBNETWORK     734
     *',I2)                                                                 735
      WRITE(6,101)                                                          736
101   FORMAT(1H0,14X,'TIME',16X,'DISTRIBUTION'/)                            737
      DO 102 I=1,IEDF                                                       738
102   WRITE(6,103) FD(I),FHAT(I)                                            739
103   FORMAT(1H ,10X,2(E15.5,5X))                                           740
      RETURN                                                                741
      END                                                                   742
      SUBROUTINE EXACTF                                                     743
C                                                                           744
C        SUBROUTINE EXACTF DETERMINES THE EMPIRICAL DISTRIBUTION            745
C        FUNCTION FOR THE SUBNETWORK DURATIONS CORRESPONDING TO ALL         746
C        POSSIBLE ACTIVITY DURATION CONFIGURATIONS                          747
C                                                                           748
      IMPLICIT REAL*8(A-H,O-Z)                                              749
C . . . . . .                                                               750
      COMMON /BLKE/INCLUS,NINCL,NCLUS                                       751
      DIMENSION NINCL(50),INCLUS(50,100)                                    752
C . . . . . .                                                               753
      COMMON /BLKG/FD,ULEFD,LLEFD                                           754
      REAL*8 FD(20),ULEFD(20),LLEFD(20)                                     755
C . . . . . .                                                               756
      COMMON /BLKK/IR,IB                                                    757
C . . . . . .                                                               758
      COMMON /BLKS/SUM,ICLGIB                                               759
C . . . . . .                                                               760
      IJ=0                                                                  761
      NIB=0                                                                 762
      KJ=NINCL(IR)                                                          763
      SUM=0.D0                                                              764
2001  NIB=NIB+1                                                             765
      IF(NIB.GT.IB) GO TO 2000                                              766
      CALL CONVRT(NIB,IJ,KJ,SPROB)                                          767
      XL=CLTIME(CPATHT)                                                     768
      X=XL-1.D-10                                                           769
      I=0                                                                   770
6420  I=I+1                                                                 771
      IF(X.GT.FD(I)) GO TO 6420                                             772
      LLEFD(I) = LLEFD(I) + SPROB                                           773
      SUM=SUM+SPROB                                                         774
      GO TO 2001                                                            775
2000  CONTINUE                                                              776
      NIB=NIB-1                                                             777
      RETURN                                                                778
      END                                                                   779
      SUBROUTINE CLUSTR                                                     780
C                                                                           781
C        SUBROUTINE CLUSTER DETERMINES THE CLUSTER FOR GIVEN THETA AND      782
C        LAMBDA                                                             783
C                                                                           784
C . . . . . .                                                               785
      IMPLICIT REAL*8 (A-H,O-Z)                                             786
      COMMON /BLKA/CTIME,DTIME,M                                            787
      DIMENSION CTIME(0500),DTIME(0500)                                     788
C . . . . . .                                                               789
      COMMON /BLKB/IBB,KB,KKB,KKK                                           790
      DIMENSION IBB(50),KB(50)                                              791
C . . . . . .                                                               792
```

```
      COMMON /BLKC/ICRITP,KCPB                                          793
      DIMENSION ICRITP(50)                                              794
C . . . . . .                                                           795
      COMMON /BLKD/SIGMA,COT                                            796
      DIMENSION COT(500),SIGMA(500)                                     797
C . . . . . .                                                           798
      COMMON /BLKE/INCLUS,NINCL,NCLUS                                   799
      DIMENSION NINCL(50),INCLUS(50,100)                                800
C . . . . . .                                                           801
      COMMON /BLKJ/THETA,LAMBDA                                         802
      REAL*8 LAMBDA,THETA                                               803
C . . . . . .                                                           804
      DIMENSION LEFT(500),LEFTO(500),NONCP(500),NINAG(500)              805
      INTEGER ASSGRP(50,50),CLINCL(50,50),EGRP(50),NCLINC(50)           806
C . . . . . .                                                           807
C                                                                       808
C        THE ASSOCIATE GROUPS ARE NOW FORMED                            809
C                                                                       810
      WRITE(6,3165)LAMBDA                                               811
3165  FORMAT(1H1,5X,'THE ASSOCIATES ARE NOW IDENTIFIED FOR LAMBDA = ',  812
     *E15.5)                                                            813
      IIIII=1                                                           814
      DO 2825 I=1,M                                                     815
2825  CTIME(I)=COT(I)                                                   816
      IWWWQ=ICRITP(1)                                                   817
2801  CONTINUE                                                          818
22825 CTIME(IWWWQ)=COT(IWWWQ)                                           819
      IWWWQ=ICRITP(IIIII)                                               820
      TEX=COT(IWWWQ)-LAMBDA*SIGMA(IWWWQ)                                821
      IF(TEX.LT.0.D0)TEX=0.D0                                           822
      CTIME(IWWWQ)=TEX                                                  823
      DUMMY=CPTIME(CPATHT)                                              824
      CALL FPATH                                                        825
C                                                                       826
C        DETERMINE ASSOCIATE GROUP                                      827
C                                                                       828
2910  NINAG(IIIII)=0                                                    829
      DO 2911 K=1,KKB                                                   830
      KK=1                                                              831
2913  IF(IBB(K).EQ.ICRITP(KK)) GO TO 2911                               832
      IF(KK.GE.KCPB) GO TO 2912                                         833
      KK=KK+1                                                           834
      GO TO 2913                                                        835
2912  NINAG(IIIII)=NINAG(IIIII)+1                                       836
      ASSGRP(IIIII,NINAG(IIIII))=IBB(K)                                 837
2911  CONTINUE                                                          838
      WRITE(6,2915) IIIII,ICRITP(IIIII),NINAG(IIIII)                    839
2915  FORMAT(1H0,10X,'THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE ',I3,840
     *'-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY ',I3,', IS = ',I3)     841
      IDUCK=NINAG(IIIII)                                                842
      IF(IDUCK.EQ.0) GO TO 2810                                         843
      WRITE(6,2916) (ASSGRP(IIIII,I),I=1,IDUCK)                         844
2916  FORMAT(1H0,15X,'THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLO845
     *WS',/,15X,50(I3,','))                                             846
2810  IIIII=IIIII+1                                                     847
      IF(IIIII.LE.KCPB) GO TO 2801                                      848
C                                                                       849
C        DETERMINE THE CLUSTERS                                         850
C                                                                       851
C             THE CLUSTERS ARE POOLED TOWARD THE TERMINAL NODE          852
C             NCLUS = THE NUMBER OF NON-EMPTY CLUSTERS                  853
```

```
C             NINCL(I) = THE NUMBER OF ACTIVITIES IN THE I-TH CLUSTER          854
C             INCLUS(I,J) = THE J-TH ACTIVITY IN THE I-TH CLUSTER              855
C             NCLINC(I) = THE NUMBER OF CLUSTERS COMPRISING THE I-TH           856
C                  CLUSTER AFTER POOLING                                       857
C             CLINCL(I,J) = THE J-TH CLUSTER WHICH HAS BEEN POOLED INTO        858
C                  THE I-TH CLUSTER                                            859
C                                                                              860
C             NCLINC AND CLINCL HELP KEEP TRACK OF WHICH CLUSTER THE           861
C             CRITICAL PATH ACTIVITIES ARE IN                                  862
C                                                                              863
C                                                                              864
C        BELOW FORMS CLUSTERS BY PUTTING EACH CRITICAL PATH ACTIVITY IN        865
C        SEPARATE CLUSTER AND THEN ADDING EACH CRITICAL PATH ACTIVITY'S        866
C        ASSOCIATES TO ITS CLUSTER                                             867
C                                                                              868
      NCLUS=KCPB                                                               869
      DO 3020 I=1,KCPB                                                         870
      NCLINC(I)=1                                                              871
      CLINCL(I,1)=I                                                            872
      NINCL(I) =NINAG(I)+1                                                     873
      INCLUS(I,1)=ICRITP(I)                                                    874
      IF(NINAG(I).EQ.0) GO TO 3020                                             875
      IDUCK=NINCL(I)                                                           876
      DO 3021 J=2,IDUCK                                                        877
      JJ=J-1                                                                   878
3021  INCLUS(I,J)=ASSGRP(I,JJ)                                                 879
3020  CONTINUE                                                                 880
C                                                                              881
C        BELOW POOLS CLUSTERS FORMED FROM ASSOCIATES                           882
C                                                                              883
      IA=0                                                                     884
3031  IA=IA+1                                                                  885
      IF(IA.GE.KCPB) GO TO 3030                                                886
      IF(NCLUS.EQ.1) GO TO 3030                                                887
      IDIA=NINCL(IA)                                                           888
      IF(IDIA.EQ.0) GO TO 3031                                                 889
      IAA=IA+1                                                                 890
      DO 3023 II=IAA,KCPB                                                      891
      IDII=NINCL(II)                                                           892
      IF(IDII.EQ.0) GO TO 3023                                                 893
      DO 3025 I=1,IDIA                                                         894
      DO 3025 J=1,IDII                                                         895
      IF(INCLUS(II,J).EQ.INCLUS(IA,I)) GO TO 3027                              896
3025  CONTINUE                                                                 897
      GO TO 3023                                                               898
3027  NCLUS=NCLUS-1                                                            899
      DO 3028 J=1,IDII                                                         900
      DO 3029 I=1,IDIA                                                         901
      IF(INCLUS(II,J).EQ.INCLUS(IA,I)) GO TO 3028                              902
3029  CONTINUE                                                                 903
      NINCL(IA)=NINCL(IA)+1                                                    904
      INCLUS(IA,NINCL(IA) )=INCLUS(II,J)                                       905
3028  CONTINUE                                                                 906
      NINCL(II)=0                                                              907
      NCLINC(IA)=NCLINC(IA)+1                                                  908
      CLINCL(IA,NCLINC(IA)) = II                                               909
      NCLINC(II)=0                                                             910
3023  CONTINUE                                                                 911
      GO TO 3031                                                               912
3030  CONTINUE                                                                 913
C                                                                              914
```

```
C        BELOW DESCRIBES CLUSTERS AFTER POOLING BASED ON THE ASSOCIATES      915
C                                                                            916
      WRITE(6,3033) NCLUS                                                    917
3033  FORMAT(1HO,/6X,'THERE ARE ',I3,' NONEMPTY CLUSTERS AFTER POOLING O     918
     *N THE BASIS OF ASSOCIATES ONLY.')                                      919
      II=0                                                                   920
      DO 3034 I=1,KCPB                                                       921
      IF(NINCL(I).EQ.0) GO TO 3034                                           922
      II=II+1                                                                923
      IDUCJ=NINCL(I)                                                         924
      WRITE(6,3035)  I,(INCLUS(I,J),J=1,IDUCJ)                               925
3035  FORMAT(1HO,10X,'THE ACTIVITIES IN THE ',I3,'-TH CLUSTER ARE AS FOL     926
     *LOWS:',/,15X,50(I3,','))                                               927
3034  CONTINUE                                                               928
      DO 3101 I=1,M                                                          929
      LEFT(I)=0                                                              930
      DO 3102 J=1,KCPB                                                       931
      IF(NINCL(J).EQ.0) GO TO 3102                                           932
      IDUCK=NINCL(J)                                                         933
      DO 3110 K=1,IDUCK                                                      934
      IF(I.EQ.INCLUS(J,K)) GO TO 3107                                        935
3110  CONTINUE                                                               936
3102  CONTINUE                                                               937
      GO TO 3101                                                             938
3107  LEFT(I)=J                                                              939
3101  CONTINUE                                                               940
C                                                                            941
C        LEFTOVERS ARE ACTIVITIES NOT IN CLUSTERS AFTER ASSOCIATES HAVE      942
C        BEEN CONSIDERED BUT BEFORE ELIMINANTS HAVE BEEN CONSIDERED          943
C                                                                            944
C                                                                            945
C        DETERMINE THE NUMBER OF LEFTOVERS. NLEFT                            946
C             LEFTO(L) = J  IMPLIES THAT THE L-TH LEFTOVER IS THE J-TH       947
C                           ACTIVITY                                         948
C                                                                            949
      NLEFT=0                                                                950
      DO 3122 J=1,M                                                          951
      IF(LEFT(J).NE.0) GO TO 3122                                            952
      NLEFT=NLEFT+1                                                          953
      LEFTO(NLEFT)=J                                                         954
3122  CONTINUE                                                               955
      WRITE(6,3123) NLEFT                                                    956
3123  FORMAT (1HO,10X,'THERE ARE ',I3,' ACTIVITIES NOT IN ANY CLUSTER YE     957
     *T.')                                                                   958
      WRITE(6,3323)THETA                                                     959
3323  FORMAT(1H1,5X,'THE ELIMINANTS OF EACH NON-CRITICAL-PATH ACTIVITY A     960
     *RE NOW DETERMINED FOR THETA = ',E15.5)                                 961
C                                                                            962
C        ELIMINANTS FOR EACH NON-CRITICAL-PATH ACTIVITY ARE NOW              963
C        DETERMINED                                                          964
C             NNNCP = THE NUMBER OF ACTIVITIES NOT ON THE CRITICAL PATH      965
C             NONCP(LE) = THE LE-TH ACTIVITY NOT ON THE CRITICAL PATH        966
C                                                                            967
      NNNCP=M-KCPB                                                           968
      LE=0                                                                   969
      DO 5000 I=1,M                                                          970
      J=1                                                                    971
5001  IF(I.EQ.ICRITP(J)) GO TO 5000                                          972
      J=J+1                                                                  973
      IF(J.LE.KCPB) GO TO 5001                                               974
5002  LE=LE+1                                                                975
```

```
      NONCP(LE)=I                                                          976
5000  CONTINUE                                                             977
      WRITE(6,5005) NNNCP                                                  978
5005  FORMAT(1H0,10X,'THERE ARE ',I3,' ACTIVITIES NOT ON THE CRITICAL PA   979
     *TH.  THEY ARE AS FOLLOWS:')                                          980
      IF(NNNCP.EQ.0) GO TO 3124                                            981
      DO 5006 I=1,LE                                                       982
5006  WRITE(6,5007) I,NONCP(I)                                             983
5007  FORMAT(1H ,15X,I3,'.    ',I3)                                        984
      LE=0                                                                 985
3126  LE=LE+1                                                              986
      CTIME(IWWWQ) = COT(IWWWQ)                                            987
      CTIME(NONCP(LE)) = COT(NONCP(LE)) + THETA*SIGMA(NONCP(LE))           988
 7756 IWWWQ = NONCP(LE)                                                    989
      DUMMY=CPTIME(CPATHT)                                                 990
      CALL FPATH                                                           991
3121  CONTINUE                                                             992
C                                                                          993
C        DETERMINE THE ELIMINANTS OF THE LE-TH ACTIVITY NOT ON THE         994
C        CRITICAL PATH                                                     995
C             NE         = THE NUMBER OF ELIMINANTS FOR THE LE-TH         996
C                          ACTIVITY NOT ON THE CRITICAL PATH               997
C             EGRP(J)      = THE J-TH ELIMINANT FOR THE LE-TH ACTIVITY    998
C                            NOT ON THE CRITICAL PATH                      999
C                                                                         1000
      NE=0                                                                1001
      DO 3130 K=1,KCPB                                                    1002
      DO 3131 I=1,KKB                                                     1003
      IF(IBB(I).EQ.ICRITP(K)) GO TO 3130                                  1004
3131  CONTINUE                                                            1005
      NE=NE+1                                                             1006
      EGRP(NE)=ICRITP(K)                                                  1007
3130  CONTINUE                                                            1008
      WRITE(6,3133) NE,NONCP(LE)                                          1009
3133  FORMAT(1H0,10X,'THERE ARE ',I3,' ELIMINANTS CORRESPONDING TO ACTIV  1010
     *ITY ',I3)                                                           1011
      IF(NE.EQ.0) GO TO 3171                                              1012
      DO 3135 K=1,NE                                                      1013
3135  WRITE(6,3136) K,NONCP(LE),EGRP(K)                                   1014
3136  FORMAT(1H ,14X,'THE ',I3,'-TH ELIMINANT CORRESPONDING TO ACTIVITY   1015
     *',I3,' IS ACTIVITY ',I3)                                            1016
C                                                                         1017
C        DETERMINE WHETHER NONCP(LE) IS AN ASSOCIATE                      1018
C             JA = 1  IF NONCP(LE)  IS AN ASSOCIATE                       1019
C             JA = 2  IF NONCP(LE)  IS NOT AN ASSOCIATE                   1020
C                                                                         1021
      K=NONCP(LE)                                                         1022
      JA=1                                                                1023
      IF(LEFT(K).EQ.0) JA=2                                               1024
      IF(JA.EQ.2) GO TO 5010                                              1025
      IT=LEFT(K)                                                          1026
C                                                                         1027
C        THE IT-TH CLUSTER IS EXPANDED TO INCLUDE ELIMINANTS              1028
C                                                                         1029
      GO TO 5011                                                          1030
5010  CONTINUE                                                            1031
C                                                                         1032
C        ITTTT IS THE ACTIVITY NUMBER OF THE FIRST ELIMINANT              1033
C        IT IS THE CLUSTER TO WHICH THE FIRST ELIMINANT CURRENTLY BELONG   1034
C                                                                         1035
      ITTT=EGRP(1)                                                        1036
```

```
      IT=LEFT(ITIT)                                                     1037
      LEFT(NONCP(LE))=IT                                                1038
C                                                                       1039
C        THE IT-TH CLUSTER IS EXPANDED TO INCLUDE ELIMINANTS            1040
C                                                                       1041
      NINCL(IT)=NINCL(IT)+1                                             1042
      INCLUS(IT,NINCL(IT))=NONCP(LE)                                    1043
      IF(NE.EQ.1) GO TO 3171                                            1044
5011  DO 3172 J=JA,NE                                                   1045
C                                                                       1046
C        IU IS THE ACTIVITY NUMBER OF THE NEXT ELIMINANT                1047
C        IF IU IS IN CLUSTER K, THEN CLUSTER K IS POOLED INTO CLUSTER IT 1048
C                                                                       1049
      IU=EGRP(J)                                                        1050
      K=LEFT(IU)                                                        1051
3182  IF(IT.EQ.K) GO TO 3172                                            1052
      NCLUS=NCLUS-1                                                     1053
      IW=NCLINC(K)                                                      1054
      DO 3183 IA=1,IW                                                   1055
      LEFT(ICRITP(CLINCL(K,IA)))=IT                                     1056
      NCLINC(IT)=NCLINC(IT)+1                                           1057
3183  CLINCL(IT,NCLINC(IT))=CLINCL(K,IA)                                1058
      NCLINC(K)=0                                                       1059
      IW=NINCL(K)                                                       1060
      NINCL(K)=0                                                        1061
      DO 3184 IA=1,IW                                                   1062
      LEFT(INCLUS(K,IA))=IT                                             1063
      NINCL(IT)=NINCL(IT)+1                                             1064
3184  INCLUS(IT,NINCL(IT))=INCLUS(K,IA)                                 1065
3172  CONTINUE                                                          1066
3171  CONTINUE                                                          1067
      IF(LE.LT.NNNCP) GO TO 3126                                        1068
C                                                                       1069
C        END OF POOLING BASED ON ELIMINANTS EXCEPT FOR THE FOLLOWING    1070
C        DESCRIPTION                                                    1071
C                                                                       1072
      WRITE(6,3173) NCLUS                                               1073
3173  FORMAT(1H1,05X,'THERE ARE ',I3,' NONEMPTY CLUSTERS AFTER POOLING O 1074
     *N THE BASIS OF BOTH ASSOCIATES AND ELIMINANTS.')                  1075
      DO 3176 I=1,KCPB                                                  1076
      IF(NINCL(I).EQ.0) GO TO 3176                                      1077
      IDD=NINCL(I)                                                      1078
      WRITE(6,3174) NINCL(I),I,(INCLUS(I,J),J=1,IDD)                    1079
3174  FORMAT(1H0,10X,'THERE ARE ',I3,' ACTIVITIES IN THE ',I3,'-TH CLUST 1080
     *ER.  THEY ARE AS FOLLOWS:',/,20X,50(I3,','))                      1081
      IDUCK=NCLINC(I)                                                   1082
      WRITE(6,3175) NCLINC(I),(CLINCL(I,J),J=1,IDUCK)                   1083
3175  FORMAT(1H0,15X,I3,' CLUSTERS HAVE BEEN POOLED TO MAKE THIS CLUSTER 1084
     *. THEY WERE AS FOLLOWS:',/,20X,50(I3,','))                        1085
3176  CONTINUE                                                          1086
3124  CONTINUE                                                          1087
      RETURN                                                            1088
      END                                                               1089
      SUBROUTINE UNION                                                  1090
C                                                                       1091
C        UNION      DETERMINES UPPER AND LOWER BOUNDS ON               1092
C        THE DISTRIBUTION FUNCTION OF THE SUBNETWORK DURATION           1093
C        BASED ON THE UNION OF THE CLUSTERS                             1094
C                                                                       1095
      IMPLICIT REAL*8(A-H,O-Z)                                          1096
C . . . . . .                                                           1097
```

```
      COMMON /BLKA/CTIME,DTIME,M                                          1098
      DIMENSION CTIME(0500),DTIME(0500)                                   1099
C . . . . . .                                                             1100
      COMMON /BLKC/ICRITP,KCPB                                            1101
      DIMENSION ICRITP(50)                                                1102
C . . . . . .                                                             1103
      COMMON /BLKE/INCLUS,NINCL,NCLUS                                     1104
      DIMENSION NINCL(50),INCLUS(50,100)                                  1105
C . . . . . .                                                             1106
      COMMON /BLKG/FD,ULEFD,LLEFD                                         1107
      REAL*8 FD(20),ULEFD(20),LLEFD(20)                                   1108
C . . . . . .                                                             1109
      COMMON /BLKH/FHI,FLO,PP,PQ                                          1110
      DIMENSION FHI(500),FLO(500),PP(500),PQ(500)                         1111
C . . . . . .                                                             1112
      COMMON /BLKI/IEOF,NSUB,SAMSIZ                                       1113
      INTEGER SAMSIZ                                                      1114
C . . . . . .                                                             1115
      COMMON /BLKK/IR,IB                                                  1116
C . . . . . .                                                             1117
      COMMON /BLKL/FLHAT,FUHAT                                            1118
      DIMENSION FLHAT(20),FUHAT(20)                                       1119
C . . . . . .                                                             1120
      COMMON /BLKS/SUM,ICLGIB                                             1121
C . . . . . .                                                             1122
      IF(NCLUS.GT.1) GO TO 6000                                           1123
      I=0                                                                 1124
6001  I=I+1                                                               1125
      IF(NINCL(I).EQ.0) GO TO 6001                                        1126
      NCL=I                                                               1127
      GO TO 6002                                                          1128
C*************************************************************************1129
C                                                                         1130
C        POOL ALL OF THE PREVICUS CLUSTERS INTO ONE CLUSTER               1131
C        NCL =  THE INDEX OF THE RESULTANT POOLED CLUSTER                 1132
C        NNCL = THE NUMBER OF ACTIVITIES IN THIS POOLED CLUSTER           1133
C                                                                         1134
C*************************************************************************1135
6000  NMAX = 0                                                            1136
      DO 6003 I=1,KCPB                                                    1137
      IF(NINCL(I).LE.NMAX) GO TO 6003                                     1138
      NCL =I                                                              1139
      NMAX=NINCL(I)                                                       1140
6003  CONTINUE                                                            1141
      DO 6004 I=1,KCPB                                                    1142
      IF(NINCL(I).EQ.0) GO TO 6004                                        1143
      IF(I.EQ.NCL) GO TO 6004                                             1144
      K=NINCL(NCL)                                                        1145
      JJ=NINCL(I)                                                         1146
      NINCL(I)=0                                                          1147
      DO 6005 J=1,JJ                                                      1148
      K=K+1                                                               1149
6005  INCLUS(NCL,K)=INCLUS(I,J)                                           1150
      NINCL(NCL)=NINCL(NCL)+JJ                                            1151
6004  CONTINUE                                                            1152
6002  NNCL=NINCL(NCL)                                                     1153
C                                                                         1154
C        ICLGIB = -1  IF ALL CONFIGURATIONS ARE TO BE SAMPLED             1155
C        ICLGIB = 0  IF SAMSIZ CONFIGURATIONS ARE TO BE SAMPLED USING     1156
C                    STRATIFIED RANDOM SAMPLING                           1157
C        ICLGIB = 1  IF SAMSIZ CONFIGURATIONS ARE TO BE SAMPLED USING     1158
```

```
C                    SIMPLE RANDOM SAMPLING                                 1159
C                                                                           1160
      IF(NNCL.LT.31) GO TO 1                                                1161
      ICLGIB=1                                                              1162
      GO TO 2                                                               1163
1     IB=2**NNCL                                                            1164
      ICLGIB=-1                                                             1165
      IF(SAMSIZ.LT.IB) ICLGIB=0                                             1166
2     CONTINUE                                                              1167
      IR=NCL                                                                1168
C*************************************************************************  1169
C                                                                           1170
C        SET  ALL ACTIVITITY DURATIONS OUTSIDE THE IR-TH CLUSTER:           1171
C                  DISTRIBUTION                                             1172
C        CTIME( ) = UPPER POINT FOR LOWER BOUND ON SUBNETWORK DURATION      1173
C        DTIME( ) = LOWER POINT FOR UPPER BOUND ON SUBNETWORK DURATION      1174
C                  DISTRIBUTION                                             1175
C                                                                           1176
C*************************************************************************  1177
      DO 6006 I=1,M                                                         1178
      DTIME(I)=FLO(I)                                                       1179
6006  CTIME(I)=FHI(I)                                                       1180
      DO 7101 I=1,IEDF                                                      1181
      ULEFD(I)=0.D0                                                         1182
7101  LLEFD(I)=0.D0                                                         1183
C*************************************************************************  1184
C                                                                           1185
      CALL BOUND                                                            1186
C                                                                           1187
C        SUBROUTINE BOUND  DETERMINES                                       1188
C        EMPIRICAL DISTRIBUTION FUNCTION FOR THE SUBNETWORK DURATIONS       1189
C        CORRESPONDING TO THE ACTIVITY DURATION CONFIGURATIONS SPECIFIED     1190
C                                                                           1191
C*************************************************************************  1192
      FLHAT(1)=LLEFD(1)/SUM                                                 1193
      FUHAT(1)=ULEFD(1)/SUM                                                 1194
      DO 6900 I=2,IEDF                                                      1195
      II=I-1                                                                1196
      LLEFD(I)=LLEFD(II)+LLEFD(I)                                           1197
      FLHAT(I)=LLEFD(I)/SUM                                                 1198
      ULEFD(I)=ULEFD(II)+ULEFD(I)                                           1199
      FUHAT(I)=ULEFD(I)/SUM                                                 1200
6900  CONTINUE                                                              1201
      RETURN                                                                1202
      END                                                                   1203
      SUBROUTINE SINGLE                                                     1204
C                                                                           1205
C        SINGLE     DETERMINES UPPER AND LOWER BOUNDS ON                   1206
C        THE DISTRIBUTION FUNCTION OF THE SUBNETWORK DURATION               1207
C        BASED ON THE INDIVIDUAL CLUSTERS - SEE TECHNICAL REPORT 51         1208
C                                                                           1209
      IMPLICIT REAL*8(A-H,O-Z)                                              1210
C . . . . . .                                                               1211
      COMMON /BLKA/CTIME,DTIME,M                                            1212
      DIMENSION CTIME(0500),DTIME(0500)                                     1213
C . . . . . .                                                               1214
      COMMON /BLKC/ICRITP,KCPB                                              1215
      DIMENSION ICRITP(50)                                                  1216
C . . . . . .                                                               1217
      COMMON /BLKE/INCLUS,NINCL,NCLUS                                       1218
      DIMENSION NINCL(50),INCLUS(50,100)                                    1219
```

```
C . . . . . .                                                          1220
      COMMON /BLKG/FD,ULEFD,LLEFD                                      1221
      REAL*8 FD(20),ULEFD(20),LLEFD(20)                                1222
C . . . . . .                                                          1223
      COMMON /BLKH/FHI,FLO,PP,PQ                                       1224
      DIMENSION FHI(500),FLO(500),PP(500),PQ(500)                      1225
C . . . . . .                                                          1226
      COMMON /BLKI/IEDF,NSUB,SAMSIZ                                    1227
      INTEGER SAMSIZ                                                   1228
C . . . . . .                                                          1229
      COMMON /BLKK/IR,IB                                               1230
C . . . . . .                                                          1231
      COMMON /BLKL/FLHAT,FUHAT                                         1232
      DIMENSION FLHAT(20),FUHAT(20)                                    1233
C . . . . . .                                                          1234
      COMMON /BLKS/SUM,ICLGIB                                          1235
C . . . . . .                                                          1236
      DO 2020 I=1,IEDF                                                 1237
      FLHAT(I)=0.D0                                                    1238
2020  FUHAT(I)=0.D0                                                    1239
      IR=0                                                             1240
3200  IR=IR+1                                                          1241
      IF(IR.GT.KCPB) GO TO 3208                                        1242
      IF(NINCL(IR).EQ.0) GO TO 3200                                    1243
      NCL=IR                                                           1244
      NNCL=NINCL(IR)                                                   1245
C                                                                      1246
C        ICLGIB = -1  IF ALL CONFIGURATIONS ARE TO BE SAMPLED          1247
C        ICLGIB = 0  IF SAMSIZ CONFIGURATIONS ARE TO BE SAMPLED USING  1248
C                    STRATIFIED RANDOM SAMPLING                        1249
C        ICLGIB = 1  IF SAMSIZ CONFIGURATIONS ARE TO BE SAMPLED USING  1250
C                    SIMPLE RANDOM SAMPLING                            1251
C                                                                      1252
      IF(NNCL.LT.31) GO TO 1                                           1253
      ICLGIB=1                                                         1254
      GO TO 2                                                          1255
1     IB=2**NNCL                                                       1256
      ICLGIB=-1                                                        1257
      IF(SAMSIZ.LT.IB) ICLGIB=0                                        1258
2     CONTINUE                                                         1259
C**********************************************************************1260
C                                                                      1261
C        SET  ALL ACTIVITITY DURATIONS OUTSIDE THE IR-TH CLUSTER:      1262
C        CTIME( ) = UPPER POINT FOR LOWER BOUND ON SUBNETWORK DURATION 1263
C                   DISTRIBUTION                                       1264
C        CTIME( ) = LOWER POINT FOR UPPER BOUND ON SUBNETWORK DURATION 1265
C                   DISTRIBUTION                                       1266
C                                                                      1267
C**********************************************************************1268
      DO 6006 I=1,M                                                    1269
      CTIME(I)=FLO(I)                                                  1270
6006  CTIME(I)=FHI(I)                                                  1271
      DO 7101 I=1,IEDF                                                 1272
      ULEFD(I)=0.D0                                                    1273
7101  LLEFD(I)=0.D0                                                    1274
C**********************************************************************1275
C                                                                      1276
      CALL BOUND                                                       1277
C                                                                      1278
C        SUBROUTINE BOUND  DETERMINES                                  1279
C        EMPIRICAL DISTRIBUTION FUNCTION FOR THE SUBNETWORK DURATIONS  1280
```

```
C         CORRESPONDING TO THE ACTIVITY DURATION CONFIGURATIONS SPECIFIED     1281
C                                                                             1282
C*****************************************************************************1283
      XXXX=LLEFD(1)/SUM                                                       1284
      IF(XXXX.GT.FLHAT(1)) FLHAT(1)=XXXX                                      1285
      XXXX=ULEFD(1)/SUM                                                       1286
      IF(XXXX.LT.FUHAT(1)) FUHAT(1)=XXXX                                      1287
      DO 6900 I=2,IEDF                                                        1288
      II=I-1                                                                  1289
      LLEFD(I)=LLEFD(II)+LLEFD(I)                                             1290
      XXXX=LLEFD(I)/SUM                                                       1291
      IF(XXXX.GT.FLHAT(I)) FLHAT(I)=XXXX                                      1292
      ULEFD(I)=ULEFD(II)+ULEFD(I)                                             1293
      XXXX=ULEFD(I)/SUM                                                       1294
      IF(XXXX.LT.FUHAT(I)) FUHAT(I)=XXXX                                      1295
6900  CONTINUE                                                                1296
      GO TO 3200                                                              1297
3200  CONTINUE                                                                1298
      RETURN                                                                  1299
      END                                                                     1300
      SUBROUTINE BOUND                                                        1301
C                                                                             1302
C        SUBROUTINE BOUND  DETERMINES THE                                     1303
C        EMPIRICAL DISTRIBUTION FUNCTION FOR THE SUBNETWORK DURATIONS         1304
C        CORRESPONDING TO THE ACTIVITY DURATION CONFIGURATIONS SPECIFIED       1305
C                                                                             1306
C        ALL ACTIVITIES OUTSIDE THE CLUSTER BEING CONSIDERED HAVE             1307
C        FIXED DURATIONS                                                      1308
C        ACTIVITIES INSIDE THE CLUSTER BEING CONSIDERED HAVE BOTH             1309
C        UPPER AND LOWER DURATIONS CONSIDERED.                                1310
C        EITHER ALL COMBINATIONS OF UPPER AND LOWER POINTS ARE                1311
C        CONSIDERED OR A RANDOM SAMPLE THEREOF.                               1312
C                                                                             1313
      IMPLICIT REAL*8(A-H,O-Z)                                                1314
C . . . . . .                                                                 1315
      COMMON /BLKG/FD,ULEFD,LLEFD                                             1316
      REAL*8 FD(20),ULEFD(20),LLEFD(20)                                       1317
C . . . . . .                                                                 1318
      COMMON /BLKI/IEDF,NSUB,SAMSIZ                                           1319
      INTEGER SAMSIZ                                                          1320
C . . . . . .                                                                 1321
      COMMON /BLKK/IR,IB                                                      1322
C . . . . . .                                                                 1323
      COMMON /BLKS/SUM,ICLGIB                                                 1324
C . . . . . .                                                                 1325
      IY=13579753                                                             1326
      NIB=0                                                                   1327
      SUM=0.D0                                                                1328
      IF(ICLGIB) 3,6210,6211                                                  1329
3     CONTINUE                                                                1330
C*****************************************************************************1331
C                                                                             1332
C        ALL ACTIVITY DURATION CONFIGURATIONS ARE TO BE CONSIDERED            1333
C                                                                             1334
C*****************************************************************************1335
2001  NIB=NIB+1                                                               1336
      IF(NIB.GT.IB) GO TO 2000                                                1337
      CALL ALL(NIB,XL,XU,SPROB)                                               1338
      X=XL-1.D-10                                                             1339
      I=0                                                                     1340
6420  I=I+1                                                                   1341
```

```
      IF(X.GT.FD(I)) GO TO 6420                                          1342
      LLEFD(I) = LLEFD(I) + SPROB                                        1343
      SUM=SUM+SPROB                                                      1344
      X=XU-1.D-10                                                        1345
      I=0                                                                1346
9420  I=I+1                                                              1347
      IF(X.GT.FD(I)) GO TO 9420                                          1348
      ULEFD(I) = ULEFD(I) + SPROB                                        1349
      GO TO 2001                                                         1350
6210  CONTINUE                                                           1351
C*************************************************************************1352
C                                                                        1353
C        A STRATIFIED RANDOM SAMPLE OF THE ACTIVITY DURATION             1354
C        CONFIGURATIONS IS TO BE CONSIDERED                              1355
C                                                                        1356
C*************************************************************************1357
2011  NIB=NIB+1                                                          1358
      IF(NIB.GT.SAMSIZ) GO TO 2000                                       1359
      CALL STRAT(NIB,XL,XU,SPROB,IY)                                     1360
      X=XL-1.D-10                                                        1361
      I=0                                                                1362
7420  I=I+1                                                              1363
      IF(X.GT.FD(I)) GO TO 7420                                          1364
      LLEFD(I) = LLEFD(I) + SPROB                                        1365
      SUM=SUM+SPROB                                                      1366
      X=XU-1.D-10                                                        1367
      I=0                                                                1368
74201 I=I+1                                                              1369
      IF(X.GT.FD(I)) GO TO 74201                                         1370
      ULEFD(I) = ULEFD(I) + SPROB                                        1371
      GO TO 2011                                                         1372
6211  CONTINUE                                                           1373
C*************************************************************************1374
C                                                                        1375
C        A SIMPLE RANDOM SAMPLE OF THE ACTIVITY DURATION                 1376
C        CONFIGURATIONS IS TO BE CONSIDERED                              1377
                                                                         1378
C*************************************************************************1379
3011  NIB=NIB+1                                                          1380
      IF(NIB.GT.SAMSIZ) GO TO 2000                                       1381
      CALL SIMPLE(NIB,XL,XU,SPROB,IY)                                    1382
      X=XL-1.D-10                                                        1383
      I=0                                                                1384
8420  I=I+1                                                              1385
      IF(X.GT.FD(I)) GO TO 8420                                          1386
      LLEFD(I) = LLEFD(I) + SPROB                                        1387
      SUM=SUM+SPROB                                                      1388
      X=XU-1.D-10                                                        1389
      I=0                                                                1390
84201 I=I+1                                                              1391
      IF(X.GT.FD(I)) GO TO 84201                                         1392
      ULEFD(I) = ULEFD(I) + SPROB                                        1393
      GO TO 3011                                                         1394
2000  CONTINUE                                                           1395
      NIB=NIB-1                                                          1396
      RETURN                                                             1397
      END                                                                1398
      SUBROUTINE ALL(NIB,CPTIM,CLTIM,SPROB)                              1399
C                                                                        1400
C     ALL  RETURNS THE COMPLETION TIME AND PROBABILITY FOR THE           1401
C     NIB-TH CONFIGURATION OF THE 2**NINCL(L) POSSIBLE CONFIGURATIONS    1402
```

```
C                                                                       1403
      IMPLICIT REAL*8 (A-H,O-Z)                                         1404
C . . . . . .                                                           1405
      COMMON /BLKE/INCLUS,NINCL,NCLUS                                   1406
      DIMENSION NINCL(50),INCLUS(50,100)                                1407
C . . . . . .                                                           1408
      COMMON /BLKK/IR,IB                                                1409
C . . . . . .                                                           1410
      IJ=0                                                              1411
      KJ=NINCL(IR)                                                      1412
      CALL CONVRT(NIB,IJ,KJ,SPROB)                                      1413
      CPTIM=CPTIME(CPATHT)                                              1414
      CLTIM=CLTIME(CPATHT)                                              1415
      RETURN                                                            1416
      END                                                               1417
      SUBROUTINE CONVRT(JP,IJ,KJ,SPROB)                                 1418
      IMPLICIT REAL*8 (A-H,O-Z)                                         1419
C . . . . . .                                                           1420
      COMMON /BLKA/CTIME,DTIME,M                                        1421
      DIMENSION CTIME(0500),DTIME(0500)                                 1422
C . . . . . .                                                           1423
      COMMON /BLKE/INCLUS,NINCL,NCLUS                                   1424
      DIMENSION NINCL(50),INCLUS(50,100)                                1425
C . . . . . .                                                           1426
      COMMON /BLKH/FHI,FLO,PP,PQ                                        1427
      DIMENSION FHI(500),FLO(500),PP(500),PQ(500)                       1428
C . . . . . .                                                           1429
      COMMON /BLKK/IR,IB                                                1430
C . . . . . .                                                           1431
C                                                                       1432
C     CONVERT THE INTEGER  JP  TO A BINARY NUMBER TO DEFINE THIS        1433
C        SEGMENT OF THE ACTIVITY CONFIGURATION                          1434
C                                                                       1435
      SPROB=1.DO                                                        1436
      K=JP                                                              1437
      DO 1 I=1,KJ                                                       1438
      IH=K/2                                                            1439
      IZ=K-IH*2                                                         1440
      L=INCLUS(IR,I+IJ)                                                 1441
      CTIME(L)=IZ*FHI(L)+(1-IZ)*FLO(L)                                  1442
      DTIME(L)=CTIME(L)                                                 1443
      SPROB=SPROB*(IZ*PQ(L)+(1-IZ)*PP(L))                               1444
1     K=IH                                                              1445
      RETURN                                                            1446
      END                                                               1447
      SUBROUTINE STRAT(NIB,CPTIM,CLTIM,SPROB,IY)                        1448
C                                                                       1449
C        STRAT SAMPLES SYSTEMATICALLY FROM THE 2**NINCL(IR)             1450
C        POSSIBLE CONFIGURATIONS AND COMPUTES THE COMPLETION TIME       1451
C        AND PROBABILITY FOR EACH SAMPLED CONFIGURATION                 1452
C                                                                       1453
C     THE ROUTINE ASSUMES THAT A SAMPLE OF SIZE  SAMSIZ  WILL BE        1454
C        GENERATED AND RETURNS ONLY ONE SAMPLE POINT PER CALL           1455
C                                                                       1456
      IMPLICIT REAL*8(A-H,O-Z)                                          1457
C . . . . . .                                                           1458
      COMMON /BLKA/CTIME,DTIME,MM                                       1459
      DIMENSION CTIME(0500),DTIME(0500)                                 1460
C . . . . . .                                                           1461
      COMMON /BLKE/INCLUS,NINCL,NCLUS                                   1462
      DIMENSION NINCL(50),INCLUS(50,100)                                1463
```

```
C . . . . . .                                                          1464
      COMMON /BLKH/FHI,FLO,PP,PQ                                       1465
      DIMENSION FHI(500),FLO(500),PP(500),PQ(500)                      1466
C . . . . . .                                                          1467
      COMMON /BLKI/IEDF,NSUB,SAMSIZ                                    1468
      INTEGER SAMSIZ                                                   1469
C . . . . . .                                                          1470
      COMMON /BLKK/IR,IB                                               1471
C . . . . . .                                                          1472
      INTEGER*4 BSUM,BCOEF                                             1473
      INTEGER*2 DIGIT(31)                                              1474
C . . . . . .                                                          1475
      SPROB=1.D0                                                       1476
      IF(NIB.NE.1)GO TO 10                                             1477
C                                                                      1478
C     INITIALIZE AND FIND THE FIRST CONFIGURATION                      1479
C                                                                      1480
      M=SAMSIZ                                                         1481
      N=NINCL(IR)                                                      1482
      NM=1                                                             1483
      BCOEF=1                                                          1484
      BSUM=BCOEF                                                       1485
      NP1=N+1                                                          1486
      N1=N                                                             1487
      M1=M-1                                                           1488
      S=DFLOAT(2**N-2)/DFLOAT(M-2)                                     1489
      DO 5 II=1,N                                                      1490
      L=INCLUS(IR,II)                                                  1491
      SPROB=SPROB*PQ(L)                                                1492
      DTIME(L)=FHI(L)                                                  1493
5     CTIME(L)=FHI(L)                                                  1494
      CPTIM=CPTIME(CPATHT)                                             1495
      CLTIM=CLTIME(CPATHT)                                             1496
      CALL RANC(IY,X)                                                  1497
      RS=-S*X+2.D0                                                     1498
      GO TO 40                                                         1499
10    IF(NIB.GE.M)GO TO 30                                             1500
C                                                                      1501
C     FIND THE NEXT CONFIGURATION IN THE SYSTEMATIC SAMPLE             1502
C                                                                      1503
      RS=RS+S                                                          1504
      IS=IDINT(RS)                                                     1505
1     IF(BSUM.GE.IS)GO TO 2                                            1506
      BCOEF=BCOEF*N1/(NP1-N1)                                          1507
      N1=N1-1                                                          1508
      BSUM=BSUM+BCOEF                                                  1509
      GO TO 1                                                          1510
2     I=IS-BSUM+BCOEF                                                  1511
      CALL CONFIG(I,BCOEF,N1,N,DIGIT)                                  1512
C                                                                      1513
C        COMPUTE CPTIM,CLTIM,AND SPROB                                 1514
C                                                                      1515
      DO 20 II=1,N                                                     1516
      L=INCLUS(IR,II)                                                  1517
      IF(DIGIT(II))18,18,19                                            1518
18    CTIME(L)=FLO(L)                                                  1519
      DTIME(L)=FLO(L)                                                  1520
      SPROB=SPROB*PP(L)                                                1521
      GO TO 20                                                         1522
19    CTIME(L)=FHI(L)                                                  1523
      DTIME(L)=FHI(L)                                                  1524
```

```
      SPROB=SPROB*PQ(L)                                                 1525
20    CONTINUE                                                          1526
      CPTIM=CPTIME(CPATHT)                                              1527
      CLTIM=CLTIME(CPATHT)                                              1528
      GO TO 40                                                          1529
C                                                                       1530
C     FIND THE LAST CONFIGURATION                                       1531
C                                                                       1532
30    DO 6 II=1,N                                                       1533
      L=INCLUS(IR,II)                                                   1534
      SPROB=SPROB*PP(L)                                                 1535
      DTIME(L)=FLG(L)                                                   1536
6     CTIME(L)=FLG(L)                                                   1537
      CPTIM=CPTIME(CPATHT)                                              1538
      CLTIM=CLTIME(CPATHT)                                              1539
40    RETURN                                                            1540
      END                                                               1541
      SUBROUTINE CONFIG(I,BCOEF,N1,N,DIGIT)                             1542
C                                                                       1543
C        CONFIG  FINDS THE I-TH LARGEST N-DIGIT BINARY NUMBER           1544
C        HAVING N1-'ONES' AND (N - N1)-'ZEROES'.  BCOEF IS THE TOTAL    1545
C        NUMBER OF SUCH CONFIGURATIONS                                  1546
C                                                                       1547
      IMPLICIT INTEGER*4(A-Z)                                           1548
C . . . . . .                                                           1549
      INTEGER*2 DIGIT(31)                                               1550
C . . . . . .                                                           1551
      DO 1 II=1,N                                                       1552
1     DIGIT(II)=0                                                       1553
      N1P=N1                                                            1554
      NP=N                                                              1555
      R=I                                                               1556
      INDEX=N                                                           1557
      C=BCOEF                                                           1558
3     IF(N1P.LT.1)GO TO 6                                               1559
      C=C*N1P/NP                                                        1560
      N1P=N1P-1                                                         1561
      NP=NP-1                                                           1562
      RR=R-C                                                            1563
      IF(RR)4,4,5                                                       1564
4     DIGIT(INDEX)=1                                                    1565
      INDEX=INDEX-1                                                     1566
      GO TO 3                                                           1567
5     INDEX=INDEX-1                                                     1568
      R=RR                                                              1569
      C=C*(NP-N1P)/NP                                                   1570
      NP=NP-1                                                           1571
      RR=R-C                                                            1572
      IF(RR)4,4,5                                                       1573
6     RETURN                                                            1574
      END                                                               1575
      SUBROUTINE SIMPLE(NIB,CPTIM,CLTIM,SPROB,IY)                       1576
      IMPLICIT REAL*8(A-H,O-Z)                                          1577
C                                                                       1578
C        SIMPLE  GENERATES (AT RANDOM) ONE OF THE 2**NINCL(IR)          1579
C        POSSIBLE CONFIGURATIONS AND FINDS ITS COMPLETION TIME AND PROB 1580
C                                                                       1581
C     INITIALIZE                                                        1582
C                                                                       1583
C . . . . . .                                                           1584
      COMMON /BLKE/INCLUS,NINCL,NCLUS                                   1585
```

```
      DIMENSION NINCL(50),INCLUS(50,100)                                1586
C . . . . . .                                                           1587
      COMMON /BLK1/IEDF,NSUB,SAMSIZ                                     1588
      INTEGER SAMSIZ                                                    1589
C . . . . . .                                                           1590
      COMMON /BLKK/IR,IB                                                1591
C . . . . . .                                                           1592
      SPROB=1.D0                                                        1593
      IF(NIB.NE.1)GO TO 10                                              1594
      NCLUS=NINCL(IR)                                                   1595
      NNC=NCLUS/31                                                      1596
      NNCR=NCLUS-NNC*31                                                 1597
10    IF(NNC.EQ.0)GO TO 20                                              1598
      RNCL=DFLOAT(2**31-1)                                              1599
      KJ=31                                                             1600
      DO 1 I=1,NNC                                                      1601
      CALL RAND(IY,U)                                                   1602
      JP=IDINT(U*RNCL)+1                                                1603
      IF(NIB.EQ.1)JP=IDINT(RNCL+.5D0)                                   1604
      IF(NIB.EQ.SAMSIZ)JP=IDINT(RNCL+1.5D0)                             1605
      IJ=(I-1)*31                                                       1606
      CALL CONVRT(JP,IJ,KJ,SPROB)                                       1607
1     CONTINUE                                                          1608
20    IF(NNCR.EQ.0)GO TO 25                                             1609
      RNCL=DFLOAT(2**NNCR-1)                                            1610
      CALL RAND(IY,U)                                                   1611
      JP=IDINT(U*RNCL)+1                                                1612
      IF(NIB.EQ.1)JP=IDINT(RNCL+.5D0)                                   1613
      IF(NIB.EQ.SAMSIZ)JP=IDINT(RNCL+1.5D0)                             1614
      IJ=NNC*31                                                         1615
      CALL CONVRT(JP,IJ,NNCR,SPROB)                                     1616
25    CPTIM=CPTIME(CPATHT)                                              1617
      CLTIM=CLTIME(CPATHT)                                              1618
      RETURN                                                            1619
      END                                                               1620
      SUBROUTINE RAND(NSEED,U)                                          1621
C        ROUTINE FOR GENERATING UNIFORM RANDOM NUMBERS BETWEEN 0 AND    1622
C      1, BASED UPON A 32 BIT LENGTH INTEGER                            1623
C                                                                       1624
C                                                                       1625
C      NSEED IS THE STARTING INTEGER CHOSEN FROM A RANDOM NUMBER TABLE  1626
C      OF TEN DIGIT INTEGERS                                            1627
C                                                                       1628
C     U IS THE RETURNING UNIFORM RANDOM NUMBER                          1629
C                                                                       1630
C                                                                       1631
C     MD IS THE LARGEST SINGLE PRECISION INTEGER (MD = 2147483647)      1632
C      FMD = FLOAT ( MD )                                               1633
C                                                                       1634
C      SEE HOAGLIN FOR TEST OF RANDOMNESS                               1635
C                                                                       1636
      IMPLICIT REAL*8(A-H,O-Z)                                          1637
      DATA FMD/2147483647.D0/,MD/2147483647/,ML/764261123/              1638
      NSEED=NSEED*ML                                                    1639
      IF(NSEED.LT.0)NSEED=NSEED+MD+1                                    1640
      U=DFLOAT(NSEED)/FMD                                               1641
      RETURN                                                            1642
      END                                                               1643
      SUBROUTINE SIMPLX                                                 1644
C . . . . . .                                                           1645
      COMMON /BLK1/IEDF,NSUB,SAMSIZ                                     1646
```

```
      INTEGER SAMSIZ                                                    1647
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .   1648
      COMMON /BLKY/FDUHAT,FDLHAT,FHAT                                   1649
      DIMENSION FDLHAT(3,20),FDUHAT(3,20),FHAT(20)                      1650
      REAL*8 FDLHAT,FDUHAT,FHAT                                         1651
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .   1652
      COMMON /BLKZ/THETA                                                1653
      REAL*4 THETA(3)                                                   1654
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .   1655
      COMMON /BLK1/BIINV,OMEGA,MM,KK,MMKK,MMKK2,N,MP1,JFIRST            1656
      COMMON /BLK2/Y1                                                   1657
      DIMENSION BRHS1(230),INBASE(230),XB1(230),Y1(230),BIINV(230,230)  1658
      DIMENSION OMEGA(3)                                                1659
      INTEGER Z                                                         1660
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .   1661
      BT(OW)= (OW*2.-3.)*OW*OW+1.                                       1662
C                                                                       1663
C        THE FOLLOWING 'TOLERANCES' ARE USED IN THE ALGORITHM.          1664
C        THEY MUST BE NON-NEGATIVE AND WOULD BE ZERO EXCEPT FOR THE     1665
C        NUMERICAL INACCURACY OF THE COMPUTER                           1666
C             TOLR1 : IF THE MAX REDUCED COST IS LESS THAN OR EQUAL TO  1667
C                     TOLR1 THEN ALL REDUCED COSTS ARE CONSIDERED TO BE 1668
C                     NON-POSITIVE.                                     1669
C             TOLR2 : ANY COMPONENT Y(I,J) LESS THAN OR EQUAL TO TOLR2  1670
C                     IS CONSIDERED NON-POSITIVE.                       1671
C                                                                       1672
      TOLR1=1.E-4                                                       1673
      TOLR2=1.E-4                                                       1674
      CALL ERRSET(208,256,-1,1)                                         1675
C                                                                       1676
C        THE INPUT                                                      1677
C                                                                       1678
C        M = THE NUMBER OF CONSTRAINTS NOT INCLUDING THE OBJECTIVE      1679
C            FUNCTION                                                   1680
C        N = THE NUMBER OF VARIABLES                                    1681
C                                                                       1682
      JFIRST=0                                                          1683
      KK=3                                                              1684
      MM=IEDF-1                                                         1685
      M=4*MM*KK+1                                                       1686
      N=(6*KK+7)*MM+1                                                   1687
      MMKK=MM*KK                                                        1688
      MMKK2=MMKK*2                                                      1689
      MP1=M+1                                                           1690
      DO 100 L=1,KK                                                     1691
100   OMEGA(L)=1./(1.+THETA(L))                                         1692
C                                                                       1693
C        BRHS IS A COLUMN OF CONSTANTS.  THE I-TH ELEMENT               1694
C        OF BRHS IS THE CONSTANT ON THE RIGHT-HAND SIDE OF THE I-TH     1695
C        EQUALITY IN    AX = BRHS                                       1696
C                                                                       1697
      BRHS1(1)=0.                                                       1698
      I1=1                                                              1699
      DO 610 L=1,MM                                                     1700
      DO 610 I=1,KK                                                     1701
      I1=I1+1                                                           1702
      I2=MMKK+I1                                                        1703
      I3=MMKK+I2                                                        1704
      I4=MMKK+I3                                                        1705
      BRHS1(I1)=FDUHAT(I,L)                                             1706
      BRHS1(I2)=FDLHAT(I,L)                                             1707
```

```
      BRHS1(I3)=BRHS1(I1)                                                1708
610   BRHS1(I4)=BRHS1(I2)                                                1709
      BRHS1(MP1)=1.                                                      1710
C                                                                        1711
C        THE MP1-BY-N MATRIX A1 IS THE AUGMENTED MATRIX OF CONSTANTS IN  1712
C        THE CONSTRAINTS AX = BRHS.  IT WILL BE GENERATED COLUMN-BY-     1713
C        COLUMN AS NEEDED.                                               1714
C                                                                        1715
C                                                                        1716
C        INBASE IS A SET OF M INTEGER VARIABLES WHICH INDICATE THE       1717
C        COMPOSITION OF THE CURRENT BASIS B.  FOR EXAMPLE,               1718
C             INBASE(K) = 7  IMPLIES THAT THE K-TH COLUMN IN THE BASIS B 1719
C                            CORRESPONDS TO THE 7-TH VARIABLE            1720
C        INBASE(1) = 0  IMPLIES THAT THE FIRST COLUMN IN THE BASIS B2    1721
C             CORRESPONDS TO X0                                          1722
C                                                                        1723
      INBASE(1)=0                                                        1724
      IU=MMKK2+1                                                         1725
      DO 21 K=2,IU                                                       1726
21    INBASE(K)=K-1                                                      1727
      IU=IU+1                                                            1728
      IL=MMKK2+7*MM-1                                                    1729
      DO 52 K=IU,MP1                                                     1730
52    INBASE(K)=IL+K                                                     1731
C                                                                        1732
C        THE M+1 BY M+1 MATRIX B1INV IS THE INVERSE OF THE REVISED       1733
C        SIMPLEX BASIS MATRIX B1.                                        1734
C        THE FOLLOWING STATEMENTS CONSTRUCT THE INITIAL MATRIX B1INV     1735
C                                                                        1736
      DO 10 II=1,MP1                                                     1737
      DO 12 L=1,MP1                                                      1738
12    B1INV(L,II)=0.                                                     1739
10    B1INV(II,II)=1.                                                    1740
      IU=MMKK2+1                                                         1741
      DO 11 II=2,IU                                                      1742
      B1INV(MMKK2+II,II)=1.                                              1743
      II=II-1                                                            1744
      I2=II/KK                                                           1745
      L=II-I2*KK                                                         1746
      IF(L.EQ.0)L=KK                                                     1747
11    B1INV(1,II)=-WT(OMEGA(L))                                          1748
C                                                                        1749
C        CALCULATE THE INITIAL VALUES OF THE BASIC VARIABLES             1750
C                                                                        1751
      DO 753 I=1,MP1                                                     1752
      XB1(I)=0.                                                          1753
      DO 755 K=1,MP1                                                     1754
755   XB1(I)=XB1(I)+B1INV(I,K)*BRHS1(K)                                  1755
753   CONTINUE                                                           1756
350   CONTINUE                                                           1757
C                                                                        1758
C        START THE BASIC SIMPLEX ALGORITHM                               1759
C                                                                        1760
C        COMPUTE THE REDUCED COSTS:  C(J) -Z(J)                          1761
C        THE J-TH REDUCED COST IS DENOTED BY REDCOS(J)                   1762
C                                                                        1763
C        FIND THE MAXIMUM REDUCED COST                                   1764
C             RMAX = THE MAXIMUM REDUCED COST                            1765
C             IRMAX = THE SMALLEST INDEX J SUCH THAT REDCOS(J)=RMAX      1766
C                                                                        1767
      Z=1                                                                1768
```

```
      IRMAX=1                                                          1769
      RMAX=REDCST(Z,1)                                                 1770
      DO 24 J=2,N                                                      1771
      REDCOS=REDCST(Z,J)                                               1772
      IF(REDCOS.LE.RMAX)GO TO 24                                       1773
      RMAX=REDCOS                                                      1774
      IRMAX=J                                                          1775
24    CONTINUE                                                         1776
C                                                                      1777
C        IF RMAX IS NOT POSITIVE THEN ALL THE REDUCED COSTS ARE        1778
C        NON-POSITIVE AND THE CURRENT SOLUTION IS AN OPTIMAL SOLUTION  1779
      IF(RMAX .LE. TOLR1) GO TO 401                                    1780
C                                                                      1781
C        DETERMINE THE IRMAX-TH Y1 VECTOR                              1782
C                                                                      1783
      CALL YRMAX(IRMAX,MP1)                                            1784
C                                                                      1785
C        IF ALL THE COMPONENTS OF Y ARE NON-POSITIVE, THE PROBLEM HAS  1786
C        AN UNBOUNDED SOLUTION. FOR THE PROBLEM BEING SOLVED THERE IS NO 1787
C        NEED TO CHECK FOR UNBOUNDEDNESS SINCE CX IS BOUNDED ABOVE BY 0. 1788
C                                                                      1789
C        THE IRMAX-TH VARIABLE IS TO ENTER THE BASIS.                  1790
C                                                                      1791
C        THE VARIABLE TO LEAVE THE BASIS IS NOW DETERMINED.            1792
C             RMIN = THE MINIMUM RATIO OF XB(I)/Y(I) WITH Y(I) > 0.    1793
C             IRMIN = THE INDEX OF THE BASIS COLUMN TO BE REMOVED      1794
C                                                                      1795
      RMIN=.99E20                                                      1796
      IRMIN=0                                                          1797
      DO 32 II=2,MP1                                                   1798
      IF(Y1(II).LE. TOLR2)  GO TO 32                                   1799
      R=XB1(II)/Y1(II)                                                 1800
      RR=R-RMIN                                                        1801
      IF(RR.GE.0.)GO TO 32                                             1802
      RMIN=R                                                           1803
      IRMIN=II                                                         1804
32    CONTINUE                                                         1805
C                                                                      1806
C        REPLACE THE IRMIN-TH COLUMN OF THE BASIS BY THE IRMAX-TH      1807
C        COLUMN OF A                                                   1808
C                                                                      1809
C        UPDATE THE BASIS INVERSE: B1INV                               1810
C                                                                      1811
      DO 33 J=2,MP1                                                    1812
      WW=B1INV(IRMIN ,J)/Y1(IRMIN )                                    1813
      DO 37 L=1,MP1                                                    1814
37    B1INV(L,J)=B1INV(L,J)-WW*Y1(L)                                   1815
33    B1INV(IRMIN ,J)=WW                                               1816
C                                                                      1817
C        UPDATE THE BASIC VARIABLES: INBASE AND XB1                    1818
C                                                                      1819
      INBASE(IRMIN)=IRMAX                                              1820
      W=XB1(IRMIN )/Y1(IRMIN )                                         1821
      DO 38 I=1,MP1                                                    1822
38    XB1(I)=XB1(I)-Y1(I)*W                                            1823
      XB1(IRMIN )=W                                                    1824
      GO TO 350                                                        1825
401    CONTINUE                                                        1826
999   CONTINUE                                                         1827
      DO 1010 L=1,MM                                                   1828
1010  FHAT(L)=0.D0                                                     1829
```

```
      DO 1011 L=1,MP1                                                    1830
      KKK=INBASE(L)                                                      1831
      IF(KKK.GT.MMKK2.AND.KKK.LE.MMKK2+MM) FHAT(KKK-MMKK2)=XB1(L  )      1832
1011  CONTINUE                                                           1833
      DO 1013 L=2,MM                                                     1834
1013  FHAT(L)=FHAT(L)+FHAT(L-1)                                          1835
      FHAT(IEDF)=1.00                                                    1836
      RETURN                                                             1837
      END                                                                1838
      SUBROUTINE YRMAX(IRMAX,MP1)                                        1839
C . . . . . .                                                            1840
      COMMON /BLK2/Y1                                                    1841
      DIMENSION Y1(230)                                                  1842
      INTEGER Z                                                          1843
C . . . . . .                                                            1844
      DO 1 Z=1,MP1                                                       1845
1     Y1(Z)=-REDCST(Z,IRMAX)                                             1846
      RETURN                                                             1847
      END                                                                1848
      FUNCTION REDCST(Z,I)                                               1849
      IMPLICIT INTEGER*4 (C-Z)                                           1850
C . . . . . .                                                            1851
      COMMON /BLK1/B1INV,W,M,K,MK,MK2,NN,NP1,JFIRST                      1852
      REAL*4 B1INV(230,230)                                              1853
      REAL*4 W(3),WWT(3),WW(3,3),WT,OW,REDCST,DR                         1854
C . . . . . .                                                            1855
      WT(OW)= (OW*2.-3.)*OW*OW+1.                                        1856
      IF(JFIRST.NE.0)GO TO 101                                           1857
      JFIRST=1                                                           1858
      LL1=MK2                                                            1859
      LL2=LL1+M                                                          1860
      LL3=LL2+6*M                                                        1861
      LL4=NN                                                             1862
      LQ=MK2+1                                                           1863
      DO 100 J=1,K                                                       1864
      OW=W(J)                                                            1865
      WW(J,1)=OW                                                         1866
      WW(J,2)=OW*OW                                                      1867
      WW(J,3)=-OW*OW                                                     1868
100   WWT(J)=WT(OW)                                                      1869
101   IF(I.GT.LL1)GO TO 200                                              1870
      P=I+1                                                              1871
      REDCST=-B1INV(Z,P)+B1INV(Z,MK2+P)                                  1872
      IF(Z.NE.1)GO TO 500                                                1873
      P=I/K                                                              1874
      Q=I-K*P                                                            1875
      IF(Q.EQ.0)Q=K                                                      1876
      REDCST=REDCST-WWT(Q)                                               1877
      GO TO 500                                                          1878
200   IF(I.GT.LL2)GO TO 300                                              1879
      REDCST=0.                                                          1880
      Q=I-MK2                                                            1881
      IL=-MK+(Q-1)*K+2                                                   1882
      IU=1                                                               1883
      DO 202 P=1,4                                                       1884
      IL=IL+MK                                                           1885
      IU=IU+MK                                                           1886
      DO 202 L=IL,IU                                                     1887
202   REDCST=REDCST-B1INV(Z,L)                                           1888
      REDCST=REDCST-B1INV(Z,NP1)                                         1889
      GO TO 500                                                          1890
```

```
300   IF(I.GT.LL3)GO TO 400                                          1891
      REDCST=0.                                                      1892
      L=I-LL2                                                        1893
      P=L/3                                                          1894
      Q=L-3*P                                                        1895
      IF(Q.NE.0)GO TO 303                                            1896
      Q=3                                                            1897
      P=P-1                                                          1898
303   IL=P*K+1                                                       1899
      IU=MK2+IL                                                      1900
      DO 302 R=1,K                                                   1901
      DR=WW(R,Q)                                                     1902
      IF(P.GE.M.AND.Q.EQ.1)DR=-DR                                    1903
302   REDCST=REDCST-DR*(B1INV(Z,IL+R)+B1INV(Z,IU+R))                 1904
      GO TO 500                                                      1905
400   IF(I.GT.LL4)GO TO 500                                          1906
      Q=I-LL3+1                                                      1907
      IF(Q-LQ)401,401,402                                            1908
401   REDCST=B1INV(Z,Q)                                              1909
      GO TO 500                                                      1910
402   REDCST=-B1INV(Z,Q)                                             1911
500   RETURN                                                         1912
      END                                                            1913
```

8. SYNTH

```
C     SYNTHESIS PROGRAM                                                    1
C                                                                          2
C  SYNTHESIS COMBINES PARALLEL AND SERIES SUBNETWORKS IDENTIFIED BY        3
C  THE DECOMPOSITION PROGRAM INCORPORATING SUBNETWORK DURATION             4
C  DISTRIBUTION INFORMATION PROVIDED BY THE SUBNETWORK ANALYSIS PROGRAM    5
C                                                                          6
C  THE FOLLOWING IS AN ALPHABETICAL LIST OF THE VARIABLES AND ARRAYS       7
C  USED BY BOTH THE MAIN PROGRAM AND ITS SUBROUTINES                       8
C                                                                          9
C  FX(I,J)=JTH C.D.F. VALUE FOR THE ITH SUBNETWORK                        10
C  FXSAVE(I)=ITH C.D.F. VALUE RESULTING FROM THE CURRENTLY SYNTHESIZED    11
C  SUBNETWORKS                                                            12
C  K1=NUMBER OF THE 1ST SUBNETWORK IN PAIR CURRENTLY BEING SYNTHESIZED    13
C  K2=NUMBER OF THE 2ND SUBNETWORK IN PAIR CURRENTLY BEING SYNTHESIZED    14
C  NIEDF(I)=NUMBER OF POINTS IN THE ITH SUBNETWORK DISTRIBUTION           15
C  NT=NUMBER OF C.D.F. SUBDIVISIONS USED THROUGHOUT SYNTHESIS (MAX=99)    16
C  NUMSUB = NUMBER OF SUBNETWORKS                                         17
C  X(I,J)=JTH X VALUE FOR THE ITH SUBNETWORK                              18
C  XSAVE(I)=ITH X VALUE RESULTING FROM THE CURRENTLY SYNTHESIZED SUBNETW  19
C                                                                         20
C  THE FOLLOWING IS AN ALPHABETICAL LIST OF THE VARIABLES AND ARRAYS USE  21
C  ONLY IN THE MAIN PROGRAM                                               22
C                                                                         23
C  ID=NUMBER OF THE SUBNETWORK                                            24
C  IEDF=NUMBER OF POINTS IN THE C.D.F.                                    25
C  INSNO(I)=ITH INSTRUCTION NUMBER                                        26
C  IOPT=0 INDICATES S WILL BE CHOSEN AS A PERCENTILE OF THE SYNTHESIZED   27
C         DISTRIBUTION                                                    28
C      =1 INDICATES S WILL BE THE MEAN OF THE SYNTHESIZED DISTRIBUTION     29
C  ISORP(I)=0 INDICATES THE CURRENT SUBNETWORKS ARE IN SERIES IN THE      30
C             ITH INSTRUCTION                                             31
C           =1 INDICATES THE CURRENT SUBNETWORKS ARE IN PARALLEL IN THE   32
C             ITH INSTRUCTION                                             33
C  ISUBNT(I)=SUBNETWORK RESULTING FROM THE ITH INSTRUCTION                34
C  JSUBNT(I,J)=JTH SUBNETWORK CONTAINED IN THE ITH INSTRUCTION            35
C              =0 INDICATES THE PREVIOUS NUMBER WAS THE LAST SUBNETWORK F  36
C                 THIS INSTRUCTION                                        37
C  NOINS=NUMBER OF INSTRUCTIONS TO BE SYNTHESIZED                         38
C  PCT=PERCENTILE VALUE TO BE USED IN DETERMINING S                       39
C  PD=DESIRED PROJECT DEADLINE                                            40
C  S=MEAN OR PERCENTILE OF FINAL DISTRIBUTION                             41
C  TT=TARGET TIME USED FOR THE TIME COMPRESSION PROGRAM                   42
C                                                                         43
C       FOR THE SAKE OF IDENTIFYING THE APPROPRIATE DIMENSIONS, LET       44
C       NTMAX = THE MAXIMUM NUMBER OF SUBDIVIDIONS ALLOWED IN THE CDF     45
C       NTMAX = THE MAXIMUM NUMBER OF SUBDIVISIONS ALLOWED IN THE CDF     46
C                  FOR THE ENTIRE PROJECT                                 47
C       NSUBMAX = THE MAXIMUM NUMBER OF SUBNETOWRKS ALLOWED IN THE        48
C                  DECOMPOSITION PROCESS                                  49
C       IEDFMAX = THE MAXIMUM NUMBER OF SUBDIVISIONS ALLOWED IN THE       50
C                  APPROXIMATE CDF FOR EACH SUBNETWORK                    51
C       NOINSMAX = THE MAXIMUM NUMBER OF INSTRUCTIONS GENERATED BY THE    52
C                  DECOMPOSITION PROCESS                                  53
C       MAX = MAXIMUM OF (NTMAX,IEDFMAX)                                  54
C       CURRENTLY, NTMAX=50; NSUBMAX=100; IEDFMAX=20; NOINSMAX=100        55
C                                                                         56
C     DIMENSION X(NSUBMAX,MAX),FX(NSUBMAX,MAX),FXSAVE(NTMAX),             57
C    *          NIEDF(NSUBMAX),INSNO(NOINSMAX),ISORP(NOINSMAX),          58
C    *          ISUBNT(NOINSMAX),JSUBNT(NOINSMAX,24),R(NTMAX),S(NTMAX),  59
C    *          FR(NTMAX),FS(NTMAX),FILE5(NOINSMAX*27),                  60
```

```
C      EQUIVLANECE (FILES(NOINSMAX+1),ISUBNT(1)),                        61
C     *            (FILES(2*NOINSMAX+1),ISORP(1)),                      62
C     *            FILES(3*NOINSMAX+1),JSUBNT(1))                       63
C                                                                        64
       IMPLICIT REAL*8 (A-H,O-Z)                                         65
       COMMON /BLKA/X(100,50),FX(100,50),XSAVE(50),FXSAVE(50),           66
      *NIEDF(100),NT                                                     67
C      COMMON /BLKB/R( 50),S( 50),FR( 50),FS( 50)                        68
       REAL*4 TTT,DPD,DCT                                                69
       DIMENSION INSNO(100),ISORP(100),ISUBNT(100),JSUBNT(100,24)        70
       INTEGER*4 FILES(2700)                                             71
       INTEGER                                F4/12/,F5/13/              72
       EQUIVALENCE (FILES(1),INSNO(1)),(FILES(101),ISUBNT(1)),           73
      *            (FILES(201),ISORP(1)),(FILES(301),JSUBNT(1))         74
C                                                                        75
C  READ DATA IN AND PRINT INPUT INFORMATION                              76
C                                                                        77
       WRITE(6,200)                                                      78
200    FORMAT(1H1,132('*')/1X,132('*')/                                  79
      *'0THIS IS THE OUTPUT FROM THE SYNTHESIS PROGRAM: SYNTH'/           80
      *1H0,132('*')/1X,132('*')///////' THE CURRENT PROJECT SCHEDULE IS TH 81
      *E ONE MOST RECENTLY LISTED BY THE DETERMINISTIC SCHEDULE RESOLUTIO 82
      *N PROGRAM.'/'0THE PROJECT',1H',S CORRESPONDING APPROXIMATE COMPLE  83
      *TION TIME DISTRIBUTION IS DETERMINED BELOW:')                      84
       REWIND F4                                                         85
       REWIND F5                                                         86
       READ(F4)NCYC,TTT,NFLAG                                            87
       READ(F5) IOPT,NT,DCT,DPD                                          88
       PCT = DCT                                                         89
       TT = TTT                                                          90
       PD = DPD                                                          91
       NTP1=NT+1                                                         92
       WRITE(6,11)NT,TT                                                  93
11     FORMAT(1H0//' FOR THIS PROBLEM'/'0THE NUMBER OF C.D.F. SUBDIVISION 94
      *S USED THROUGHOUT SYNTHESIS IS',I5,'.'/' THE TARGET TIME USED BY T 95
      *HE DETERMINISTIC SCHEDULER WAS ',F10.5)                            96
       READ(F5) NOINS                                                    97
       IF(IOPT.EQ.0) GO TO 17                                            98
       WRITE(6,16)PD                                                     99
16     FORMAT(' THE MEAN OF THE PROJECT',1H',S APPROXIMATE COMPLETION TI 100
      *ME DISTRIBUTION WILL BE COMPARED TO THE'/' SPECIFIED PROJECT DEADL 101
      *INE TIME OF ',F12.5)                                              102
       GO TO 19                                                          103
   17  P=PCT*100.                                                        104
       WRITE(6,18) P,PD                                                  105
18     FORMAT(' THE ',F6.2,'-TH PERCENTILE OF THE PROJECT',1H',S APPROXI 106
      *MATE COMPLETION TIME DISTRIBUTION WILL BE COMPARED TO THE'/        107
      *' SPECIFIED PROJECT DEADLINE TIME OF ',F12.5)                     108
   19 WRITE(6,600)                                                       109
  600 FORMAT(//,    10X,'THE INSTRUCTIONS TO BE PERFORMED ARE',//)       110
       WRITE(6,605)                                                      111
  605 FORMAT(T40,'SERIES=0')                                             112
       WRITE(6,610)                                                      113
  610 FORMAT(  5X,'INSTRUCTION NO.',T25,'SUBNETWORK',T40,'PARALLEL=1',   114
     $T55,'SUBNETWORKS TO BE SYNTHESIZED')                               115
       READ(F5) FILES                                                    116
       DO 20 I=1,NOINS                                                   117
       NSP=0                                                             118
       DO 210 J=1,24                                                     119
       IF(JSUBNT(I,J).LE.0)GO TO 211                                     120
210    NSP=NSP+1                                                         121
```

```
211    CONTINUE                                                          122
       WRITE(6,26) INSNU(I),ISUBNT(I),ISORP(I),(JSUBNT(I,J),J=1,NSP)     123
    26 FORMAT(/,T11,I3,T28,I3,T43,I2,T55,24I3)                           124
    20 CONTINUE                                                          125
       WRITE(6,31)                                                       126
    31 FORMAT(1H1,//,10X,'SUBNETWORK',T30,'APPROXIMATE DISTRIBUTION DETER 127
      $MINED USING SUBNETWORK ANALYSIS')                                 128
35     READ(F5,END=999)ID,IEDF,(X(ID,J),FX(ID,J),J=1,IEDF)               129
       NIEDF(ID)=IEDF                                                    130
       WRITE(6,51) ID                                                    131
    51 FORMAT(/,T14,I3,T34,'X',T48,'F(X)')                               132
       WRITE(6,52) (X(ID,J),FX(ID,J),J=1,IEDF)                           133
    52 FORMAT(  T25,F15.5,T39,F15.5)                                     134
       GO TO 35                                                          135
C                                                                        136
C  CONSIDER INSTRUCTIONS IN REVERSE ORDER AND SYNTHESIZE ALL SUBNETWORKS 137
C  ACCORDING TO SERIES OR PARALLEL                                       138
C                                                                        139
  999 DO 100 I=1,NOINS                                                   140
       II=NCINS-I+1                                                      141
       DO 80 J=1,24                                                      142
       K1=JSUBNT(II,J)                                                   143
       JP1=J+1                                                           144
       K2=JSUBNT(II,JP1)                                                 145
       IF(K2.EQ.0) GO TO 75                                              146
       IF(ISORP(II).EQ.1) GO TO 55                                       147
       CALL SERIES(K1,K2)                                                148
       GO TO 60                                                          149
    55 CALL PARAL(K1,K2)                                                 150
C                                                                        151
C  STORE CURRENT DISTRIBUTION IN SUBNETWORK K2                           152
C                                                                        153
    60 DO 65 K=1,NTP1                                                    154
       X(K2,K)=XSAVE(K)                                                  155
       FX(K2,K)=FXSAVE(K)                                                156
    65 CONTINUE                                                          157
       NIEDF(K2)=NTP1                                                    158
       GO TO 80                                                          159
C                                                                        160
C  STORE DISTRIBUTION RESULTING FROM CURRENT INSTRUCTION IN SUBNETWORK    161
C  GIVEN BY INSTRUCTION                                                  162
C                                                                        163
    75 DO 78 K=1,NTP1                                                    164
       X(ISUBNT(II),K)=XSAVE(K)                                          165
       FX(ISUBNT(II),K)=FXSAVE(K)                                        166
    78 CONTINUE                                                          167
       NIEDF(ISUBNT(II))=NTP1                                            168
       GO TO 100                                                         169
    80 CONTINUE                                                          170
   100 CONTINUE                                                          171
C                                                                        172
C  OUTPUT FINAL DISTRIBUTION                                             173
C                                                                        174
       WRITE(6,105)                                                      175
   105 FORMAT(1H1,//,10X,'THE DISTRIBUTION FOR THE SYNTHESIZED NETWORK IS 176
      $',//,T15,'T',T30,'F(T)')                                          177
       DO 110 K=1,NTP1                                                   178
       IF(PCT.GT.FX(1,K)) KEEP=K                                         179
   110 WRITE(6,120) X(1,K),FX(1,K)                                       180
   120 FORMAT(    5X,2F15.5)                                            181
C                                                                        182
```

```
C  COMPUTE S AND PRODUCE INFORMATION NEEDED FOR TIME COMPRESSION                183
C                                                                              184
      IF(ICPT.EQ.0) GO TO 140                                                  185
      S=X(1,1)*FX(1,1)                                                         186
      DO 130 K=2,NTP1                                                          187
      KM1=K-1                                                                  188
  130 S=S+X(1,K)*(FX(1,K)-FX(1,KM1))                                           189
      GO TO 150                                                                190
  140 KEEP1=KEEP+1                                                             191
      DIFF1=X(1,KEEP1)-X(1,KEEP)                                               192
      DIFF2=FX(1,KEEP1)-FX(1,KEEP)                                             193
      DIFF3=PCT-FX(1,KEEP)                                                     194
      S=X(1,KEEP)+DIFF1*DIFF3/DIFF2                                            195
  150 TT=PD*TT/S                                                               196
      ERRPCT=DABS(S-PD)*100./PD                                                197
      IF(IOPT)1,1,2                                                            198
1     P=PCT*100.                                                               199
      WRITE(6,4)P,S                                                            200
4     FORMAT(//,10X,'THE ',F6.2,'-TH PERCENTILE OF THE SYNTHESIZED DISTR       201
     *IBUTION IS ',F12.5,/)                                                    202
      GO TO 3                                                                  203
2     WRITE(6,5)S                                                              204
5     FORMAT(//10X'THE MEAN OF THE SYNTHESIZED DISTRIBUTION IS ',F12.5/)       205
3     WRITE(6,180)S,ERRPCT,TT                                                  206
180   FORMAT(   10X,'THE DIFFERENCE BETWEEN 'F12.5,' AND THE PROJECT DEA       207
     *DLINE IS ',F9.2,' PERCENT OF THE PROJECT DEADLINE.'/                     208
     *10X,'HENCE, THE NEW TARGET TIME IS ',F12.5)                              209
      WRITE(6,300)NCYC                                                         210
300   FORMAT(//1H0,132('*')/'0THIS COMPLETES ITERATION',I7/1H0,132('*'))       211
      NCYC=NCYC+1                                                              212
      TTT=TT                                                                   213
      REWIND F4                                                                214
      WRITE(F4)NCYC,TTT,NFLAG                                                  215
  190 STOP                                                                     216
      END                                                                      217
C                                                                              218
C  THE FOLLOWING IS AN ALPHABETICAL LIST OF THE VARIABLES AND ARRAYS USE       219
C  IN BOTH SUBROUTINES                                                         220
C                                                                              221
C  BT=BEGINNING TIME FOR RESULTING DISTRIBUTION                                222
C  DELT=INCREMENT FOR THE X VALUES ALLOWING FOR NT EQUAL DIVISIONS             223
C  FR(I)=ITH C.D.F. VALUE FOR THE 1ST SUBNETWORK OF PAIR TO BE SYNTHESIZ       224
C  FS(I)=ITH C.D.F. VALUE FOR THE 2ND SUBNETWORK OF PAIR TO BE SYNTHESIZ       225
C  NR=NUMBER OF POINTS IN 1ST SUBNETWORK                                       226
C  NS=NUMBER OF POINTS IN 2ND SUBNETWORK                                       227
C  R(I)=ITH X VALUE FOR THE 1ST SUBNETWORK OF PAIR TO BE SYNTHESIZED           228
C  S(I)=ITH X VALUE FOR THE 2ND SUBNETWORK OF PAIR TO BE SYNTHESIZED           229
C  T=X VALUE FOR THE RESULTING SYNTHESIZED SUBNETWORK                          230
C  XMAX=MAXIMUM POSSIBLE X VALUE FOR THE RESULTING SYNTHESIZED SUBNETWOR       231
C  XMIN=MINIMUM POSSIBLE X VALUE FOR THE RESULTING SYNTHESIZED SUBNETWOR       232
C                                                                              233
      SUBROUTINE PARAL(K1,K2)                                                  234
C                                                                              235
C  SYNTHESIZES 2 PARALLEL SUBNETWORKS                                          236
C                                                                              237
      IMPLICIT REAL*8 (A-H,O-Z)                                                238
      COMMON /BLKA/X(100,50),FX(100,50),XSAVE(50),FXSAVE(50),                  239
     *NIEDF(100),NT                                                           240
      COMMON /BLKB/R( 50),S( 50),FR( 50),FS( 50)                               241
      NR=NIEDF(K1)                                                             242
      NS=NIEDF(K2)                                                             243
```

```
      DO 10 I=1,NR                                                    244
      R(I)=X(K1,I)                                                    245
   10 FR(I)=FX(K1,I)                                                  246
      DO 15 I=1,NS                                                    247
      S(I)=X(K2,I)                                                    248
   15 FS(I)=FX(K2,I)                                                  249
      XMIN=DMIN1(R(1),S(1))                                           250
      XMAX=DMAX1(R(NR),S(NS))                                         251
      BT=XMIN                                                         252
      DELT=(XMAX-XMIN)/NT                                             253
      T=BT-DELT                                                       254
      DO 50 I=1,NT                                                    255
      F1=0.0D0                                                        256
      F2=0.0D0                                                        257
      T=T+DELT                                                        258
      DO 40 J=1,NR                                                    259
      IF(R(J).GT.T) GO TO 41                                          260
      F1=FR(J)                                                        261
   40 CONTINUE                                                        262
   41 DO 45 J=1,NS                                                    263
      IF(S(J).GT.T) GO TO 46                                          264
      F2=FS(J)                                                        265
   45 CONTINUE                                                        266
   46 FXSAVE(I)=F1*F2                                                 267
      XSAVE(I)=T                                                      268
   50 CONTINUE                                                        269
      NTP1=NT+1                                                       270
      FXSAVE(NTP1)=1.0D0                                              271
      XSAVE(NTP1)=XMAX                                                272
      RETURN                                                          273
      END                                                             274
      SUBROUTINE SERIES(K1,K2)                                        275
C                                                                     276
C  SYNTHESIZES 2 SUBNETWORKS IN SERIES                                277
C                                                                     278
      IMPLICIT REAL*8 (A-H,O-Z)                                       279
      COMMON /BLKA/X(100,50),FX(100,50),XSAVE(50),FXSAVE(50),         280
     *NIEDF(100),NT                                                   281
      COMMON /BLKB/R( 50),S( 50),FR( 50),FS( 50)                      282
      NR=NIEDF(K1)                                                    283
      NS=NIEDF(K2)                                                    284
      DO 10 I=1,NR                                                    285
      R(I)=X(K1,I)                                                    286
   10 FR(I)=FX(K1,I)                                                  287
      DO 15 I=1,NS                                                    288
      S(I)=X(K2,I)                                                    289
   15 FS(I)=FX(K2,I)                                                  290
      XMIN=R(1)+S(1)                                                  291
      XMAX=R(NR)+S(NS)                                                292
      BT=XMIN                                                         293
      DELT=(XMAX-XMIN)/NT                                             294
      T=BT-DELT                                                       295
      DO 55 I=1,NT                                                    296
      FT=0.0D0                                                        297
      T=T+DELT                                                        298
      DO 50 J=1,NS                                                    299
      IF(S(J).GT.T) GO TO 54                                          300
      IF(J.GT.1) GO TO 30                                             301
      FF=FS(1)                                                        302
      GO TO 35                                                        303
   30 JJ=J-1                                                          304
```

```
      FF=FS(J)-FS(JJ)                                                   305
   35 FTMS=0.0D0                                                        306
      TMS=T-S(J)                                                        307
      DO 40 K=1,NR                                                      308
      IF(R(K).GT.TMS) GO TO 45                                          309
      FTMS=FR(K)                                                        310
   40 CONTINUE                                                          311
   45 FT=FT+FF*FTMS                                                     312
   50 CONTINUE                                                          313
   54 XSAVE(I)=T                                                        314
      FXSAVE(I)=FT                                                      315
   55 CONTINUE                                                          316
      NTP1=NT+1                                                         317
      FXSAVE(NTP1)=1.0D0                                                318
      XSAVE(NTP1)=XMAX                                                  319
      RETURN                                                            320
      END                                                               321
```

9. SAVEFIL

```
C      PROGRAM SAVEFIL                                                    1
C                                                                         2
C         THIS PROGRAM TRANSFERS THE OPERATIONAL INFORMATION GENERATED    3
C         BY THE STATISTICAL PERT PROGRAMS FROM THE TEMPORARY FILES TO    4
C         THE PERMANENT DATA SET DEFINED AS I/O-UNIT PS                   5
C                                                                         6
C                                                                         7
C         INPUT INSTRUCTIONS                                              8
C                                                                         9
C         A SINGLE CARD SHOULD BE SUPPLIED AS INPUT TO THIS PROGRAM.     10
C         THE CARD SHOULD BEAR IN COLUMNS 1 - 8 THE NAME OF THE LAST     11
C         PROGRAM EXECUTED BEFORE THIS INTERRUPT.                        12
C         (FORMAT 2A4, LEFT JUSTIFIED)                                   13
C                                                                        14
C         FOR THE SAKE OF IDENTIFYING THE APPROPRIATE DIMENSIONS, LET    15
C         MMAX = THE MAXIMUM NUMBER OF ACTIVITIES IN THE ORIGINAL        16
C                   PROJECT NETWORK                                      17
C         BKMAX = THE MAXIMUM NUMBER OF BREAK POINTS ALLOWED IN THE      18
C                   ENTIRE PROJECT'S TIME-COST CURVE                     19
C         NCTMAX = THE MAXIMUM NUMBER OF COMPLETION TIMES AND COSTS FOR  20
C                   EACH ACTIVITY                                        21
C         IEDFMAX = THE MAXIMUM NUMBER OF SUBDIVISIONS ALLOWED IN THE    22
C                   APPROXIMATE CDF FOR EACH SUBNETWORK                  23
C         NOINSMAX = THE MAXIMUM NUMBER OF INSTRUCTIONS GENERATED BY THE 24
C                   DECOMPOSITION PROCESS                                25
C                                                                        26
C      INTEGER*4 FILE0(4*MMAX+5),NCT(MMAX),FILE3B(3*MMAX+5),             27
C     *          FILE5(NOINSMAX*27)                                     28
C      INTEGER*2 BREAK(BKMAX),TIMCST(2*NCTMAX),FILE6(2*MMAX)            29
C      REAL*8 FD(IEDFMAX),X(IEDFMAX)                                    30
C      EQUIVALENCE (FILE0(3*MMAX+6),NCT(1))                             31
C                                                                       32
C                                                                       33
       INTEGER F0/8/,F1/9/,F2/10/,F3/11/,F4/12/,F5/13/,F6/14/,F7/15/     34
       INTEGER PS/4/                                                    35
       INTEGER FILE0(4005),NCT(1000),FILE3A(11),FILE3B(3005),FILE5(2700), 36
      *        LIST(24)                                                  37
       REAL FILE1(5),FILE2(5)                                           38
       REAL*8 CDF(12),FD(20),X(20)                                      39
       INTEGER*2 NBREAK,BREAK(3000),TIMCST(12),TEST1,TEST2,TEST3,STIME, 40
      *          FILE6(2000)                                           41
       EQUIVALENCE (FILE0(1),NACT),(FILE0(3006),NCT(1))                 42
       EQUIVALENCE (FILE1(1),FILE2(1),FILE3A(1),FILE3B(1),FILE5(1),     43
      *            FILE6(1),BREAK(1),TIMCST(1),LIST(1),CDF(1),FD(1))   44
       INTEGER IPROG(2)                                                 45
       INTEGER PROG(13)/'MAIN','LOOP','OPS ','DSR ','SIMP','MODS','DECO', 46
      *                 'SUBN','SYNT','DSR ','MODS','SUBN','SYNT'/     47
C                                                                       48
       WRITE(6,112)                                                     49
       REWIND F4                                                        50
       READ(F4)NCYC                                                     51
       READ(5,110)IPROG                                                 52
       K1=1                                                             53
       IF(NCYC.GE.2)K1=10                                               54
       DO 1 I=K1,13                                                     55
       IF(IPROG(1).EQ.PROG(I))GO TO 2                                   56
1      CONTINUE                                                         57
       WRITE(6,111)                                                     58
       GO TO 109                                                        59
2      K=I                                                              60
```

```
      REWIND F0                                                          61
      REWIND F1                                                          62
      REWIND F2                                                          63
      REWIND F3                                                          64
      REWIND F4                                                          65
      REWIND F5                                                          66
      REWIND F6                                                          67
      REWIND F7                                                          68
      REWIND PS                                                          69
C                                                                        70
C        READ AND STORE FILE0                                            71
C                                                                        72
      READ(F0) FILE0                                                     73
      WRITE(PS)FILE0                                                     74
      READ(F0) JMAT                                                      75
      WRITE(PS)JMAT                                                      76
      IF(K.LT.3)GO TO 3                                                  77
      READ(F0) NBREAK,(BREAK(I),I=1,NBREAK)                              78
      WRITE(PS)NBREAK,(BREAK(I),I=1,NBREAK)                              79
3     CONTINUE                                                           80
C                                                                        81
C        READ AND STORE FILE1                                            82
C                                                                        83
      READ(F1) TEST1,TEST2,TEST3,STIME                                   84
      WRITE(PS)TEST1,TEST2,TEST3,STIME                                   85
      DO 4 I=1,NACT                                                      86
      NNCT=NCT(I)                                                        87
      NNCT1=NNCT-1                                                       88
      NNCT2=2*NNCT                                                       89
      READ(F1) (TIMCST(J),J=1,NNCT2)                                     90
      WRITE(PS)(TIMCST(J),J=1,NNCT2)                                     91
      DO 4 J=1,NNCT1                                                     92
      READ(F1) FILE1                                                     93
      WRITE(PS)FILE1                                                     94
4     CONTINUE                                                           95
C                                                                        96
C        READ AND STORE FILE2                                            97
C                                                                        98
      IF(K.NE.4.AND.K.NE.5.AND.K.NE.10)GO TO 6                           99
      DO 5 I=1,NACT                                                     100
      READ(F2) FILE2                                                    101
      WRITE(PS)FILE2                                                    102
5     CONTINUE                                                          103
6     CONTINUE                                                          104
C                                                                       105
C        READ AND STORE FILE3                                           106
C                                                                       107
      READ(F3) FILE3A                                                   108
      WRITE(PS)FILE3A                                                   109
      IF(K.NE.5.AND.K.NE.6)GO TO 7                                      110
      READ(F3) FILE3B                                                   111
      WRITE(PS)FILE3B                                                   112
7     CONTINUE                                                          113
      IF(K.LT.7)GO TO 10                                                114
8     READ(F3,END=9)M                                                   115
      WRITE(PS)     M                                                   116
      MM=2+3*M                                                          117
      READ(F3) (FILE3B(I),I=1,MM)                                       118
      WRITE(PS)(FILE3B(I),I=1,MM)                                       119
      GO TO 8                                                           120
9     M=0                                                               121
```

AD-A060 568 TEXAS A AND M UNIV COLLEGE STATION INST OF STATISTICS F/G 9/2
A USER'S GUIDE TO THE COMPUTER IMPLEMENTATION OF THE NEW PROJEC--ETC(U)
AUG 78 T C BAKER, R L SIELKEN N00014-78-C-0426
UNCLASSIFIED THEMIS-TR-57 NL

4 OF 4
AD
A060568

END
DATE
FILMED
01-79
DDC

```
      WRITE(PS)M                                                        122
10    CONTINUE                                                          123
C                                                                       124
C        READ AND STORE FILE4                                           125
C                                                                       126
      READ(F4) NCYC,TT,NFLAG                                            127
      WRITE(PS)NCYC,TT,NFLAG                                            128
      IF(K.NE.6.AND.K.NE.7.AND.K.NE.11)GO TO 12                         129
      READ(F4) JNACT                                                    130
      WRITE(PS)JNACT                                                    131
      DO 11 I=1,JNACT                                                   132
      READ(F4) CDF                                                      133
      WRITE(PS)CDF                                                      134
11    CONTINUE                                                          135
12    CONTINUE                                                          136
C                                                                       137
C        READ AND STORE FILE5                                           138
C                                                                       139
      READ(F5) IOPT,NT,PCT,PD                                           140
      WRITE(PS)IOPT,NT,PCT,PD                                           141
      IF(K.LT.7)GO TO 15                                                142
      READ(F5) NOINS                                                    143
      WRITE(PS)NOINS                                                    144
      READ(F5) FILE5                                                    145
      WRITE(PS)FILE5                                                    146
      IF(K.NE.8.AND.K.NE.12)GO TO 15                                    147
13    READ(F5,END=14)ID,IEDF,(FD(I),X(I),I=1,IEDF)                      148
      WRITE(PS)      ID,IEDF,(FD(I),X(I),I=1,IEDF)                      149
      GO TO 13                                                          150
14    ID=-1                                                             151
      WRITE(PS)      ID,IEDF,(FD(I),X(I),I=1,IEDF)                      152
15    CONTINUE                                                          153
C                                                                       154
C     READ AND STORE FILE6                                              155
C                                                                       156
      IF(K.LT.3)GO TO 17                                                157
      NACT2=2*NACT                                                      158
      DO 16 I=1,NBREAK                                                  159
      READ(F6) (FILE6(J),J=1,NACT2)                                     160
      WRITE(PS)(FILE6(J),J=1,NACT2)                                     161
16    CONTINUE                                                          162
17    CONTINUE                                                          163
C                                                                       164
C        READ AND STORE FILE7                                           165
C                                                                       166
      IF(K.LT.5)GO TO 20                                                167
18    READ(F7,END=19)LIST                                               168
      WRITE(PS)      LIST                                               169
      GO TO 18                                                          170
19    LIST(1)=-1                                                        171
      WRITE(PS)LIST                                                     172
20    CONTINUE                                                          173
C                                                                       174
C     WRITE TERMINATION INFORMATION                                     175
C                                                                       176
      WRITE(6,113)PS,NCYC,IPROG                                         177
109   STOP                                                              178
110   FORMAT(2A4)                                                       179
111   FORMAT(1H0,5X,'THE NAME SPECIFIED FOR THE LAST PROGRAM EXECUTED IS 180
     * INVALID.')                                                       181
112   FORMAT(1H1,132('*')/1X,132('*')/'0THIS IS THE OUTPUT FOR THE SYSTE 182
```

```
     *M INTERRUPTION PROGRAM: SAVEFIL'/1H0,132('*')/1X,132('*'))                183
113   FORMAT(1H0,5X,'THE OPERATIONAL INFORMATION GENERATED BY THE STATIS        184
     *TICAL PERT PROGRAMS HAS BEEN SUCCESSFULLY TRANSFERRED'/6X,'TO THE        185
     *DATA SET DEFINED AS I/O-UNIT ',I2//6X,'THE PROCEDURE WAS INTERRUPT       186
     *ED DURING THE ',I5,'-TH ITERATION.'/6X,2A4,' WAS THE LAST PROGRAM        187
     *EXECUTED.')                                                              188
      END                                                                      189
```

10. READFIL

```
C      PROGRAM READFIL                                                    1
C                                                                         2
C         THIS PROGRAM RETRIEVES THE OPERATIONAL INFORMATION FOR THE      3
C         STATISTICAL PERT PROGRAMS FROM THE PERMANENT FILE REFERENCED BY 4
C         I/O-UNIT PS AND RESTORES IT TO THE APPROPRIATE TEMPORARY FILES  5
C                                                                         6
C                                                                         7
C         INPUT INSTRUCTIONS                                              8
C                                                                         9
C         A SINGLE CARD SHOULD BE SUPPLIED AS INPUT TO THIS PROGRAM.     10
C                                                                        11
C         THE CARD SHOULD BEAR THE FOLLOWING:                            12
C              COLS  1- 8:  THE NAME OF THE LAST PROGRAM EXECUTED BEFORE 13
C                           THE INTERRUPT (FORMAT 2A4, LEFT JUSTIFIED)   14
C              COL      10:  0  IF THE PROCEDURE WAS INTERRPUTED BEFORE   15
C                               THE FIRST EXECUTION OF SYNTH             16
C                           1  OTHERWISE                                 17
C              THE REMAINING PARAMETERS SHOULD ONLY BE SUPPLIED IF THE    18
C              PROCEDURE WAS INTERRUPTED IMMEDIATELY AFTER THE EXECUTION  19
C              OF SYNTH                                                   20
C                                                                         21
C              COLS 11-20:  THE VALUE OF  TT  FOR THE NEXT ITERATION IF   22
C                           DIFFERENT FROM THE VALUE SPECIFIED BY SYNTH    23
C              COLS 21-25:  1  IF THE VALUES OF ANY OF THE FOLLOWING      24
C                              PARAMETERS ARE TO BE DIFFERENT FROM THEIR  25
C                              CURRENT VALUES IN THE NEXT ITERATION:      26
C                                  IOPT,PCT,PD                            27
C                           0  OTHERWISE                                  28
C              COLS 26-30:  THE VALUE OF  IOPT  FOR THE NEXT ITERATION    29
C              COLS 31-40:  THE VALUE OF  PCT  FOR THE NEXT ITERATION     30
C              COLS 41-50:  THE VALUE OF  PD  FOR THE NEXT ITERATION      31
C                                                                        32
C         FOR THE SAKE OF IDENTIFYING THE APPROPRIATE DIMENSIONS, LET    33
C         MMAX = THE MAXIMUM NUMBER OF ACTIVITIES IN THE ORIGINAL        34
C                PROJECT NETWORK                                         35
C         BKMAX = THE MAXIMUM NUMBER OF BREAK POINTS ALLOWED IN THE      36
C                 ENTIRE PROJECT'S TIME-COST CURVE                       37
C         NCTMAX = THE MAXIMUM NUMBER OF COMPLETION TIMES AND COSTS FOR  38
C                  EACH ACTIVITY                                         39
C         IEDFMAX = THE MAXIMUM NUMBER OF SUBDIVISIONS ALLOWED IN THE    40
C                   APPROXIMATE CDF FOR EACH SUBNETWORK                  41
C         NOINSMAX = THE MAXIMUM NUMBER OF INSTRUCTIONS GENERATED BY THE 42
C                    DECOMPOSITION PROCESS                               43
C         CURRENTLY, MMAX=1000; BKMAX=3000; NCTMAX=6; IEDFMAX=20;        44
C                    NOINSMAX=100                                        45
C                                                                        46
C      INTEGER*4 FILE0(4*MMAX+5),NCT(MMAX),FILE3B(3*MMAX+5),             47
C     *          FILE5(NOINSMAX*27)                                      48
C      INTEGER*2 BREAK(BKMAX),TIMCST(2*NCTMAX),FILE6(2*MMAX)             49
C      REAL*8 FD(IEDFMAX),X(IEDFMAX)                                     50
C      EQUIVALENCE (FILE0(3*MMAX+6),NCT(1))                              51
C                                                                        52
       INTEGER F0/8/,F1/9/,F2/10/,F3/11/,F4/12/,F5/13/,F6/14/,F7/15/      53
       INTEGER PS/4/                                                     54
       INTEGER FILE0(4005),NCT(1000),FILE3A(11),FILE3B(3005),FILE5(2700), 55
      *        LIST(24)                                                   56
       REAL FILE1(5),FILE2(5)                                            57
       REAL*8 CDF(12),FD(20),X(20)                                       58
       INTEGER*2 NBREAK,BREAK(3000),TIMCST(12),TEST1,TEST2,TEST3,STIME,  59
      *          FILE6(2000)                                             60
```

```
      EQUIVALENCE (FILE0(1),NACT),(FILE0(3006),NCT(1))                  61
      EQUIVALENCE (FILE1(1),FILE2(1),FILE3A(1),FILE3B(1),FILE5(1),      62
     *            FILE6(1),BREAK(1),TIMCST(1),LIST(1),CDF(1),FD(1))     63
      INTEGER IPROG(2)                                                  64
      INTEGER PROG(13)/'MAIN','LOOP','DPS ','DSR ','SIMP','MODS','DECO',65
     *                 'SUBN','SYNT','DSR ','MODS','SUBN','SYNT'/       66
C                                                                       67
      WRITE(6,112)                                                      68
      READ(5,110)IPROG,NCYC,IFLG,TTT,IIOPT,PPCT,PPD                     69
      K1=10                                                             70
      IF(NCYC.EQ.0)K1=1                                                 71
      DO 1 I=K1,13                                                      72
      IF(IPROG(1).EQ.PROG(I))GO TO 2                                    73
1     CONTINUE                                                          74
      WRITE(6,111)                                                      75
      GO TO 109                                                         76
2     K=I                                                               77
      REWIND F0                                                         78
      REWIND F1                                                         79
      REWIND F2                                                         80
      REWIND F3                                                         81
      REWIND F4                                                         82
      REWIND F5                                                         83
      REWIND F6                                                         84
      REWIND F7                                                         85
      REWIND PS                                                         86
C                                                                       87
C        RESTORE FILE0                                                  88
C                                                                       89
      READ(PS) FILE0                                                    90
      WRITE(F0)FILE0                                                    91
      READ(PS) JMAT                                                     92
      WRITE(F0)JMAT                                                     93
      IF(K.LT.3)GO TO 3                                                 94
      READ(PS) NBREAK,(BREAK(I),I=1,NBREAK)                             95
      WRITE(F0)NBREAK,(BREAK(I),I=1,NBREAK)                             96
      ENDFILE F0                                                        97
3     CONTINUE                                                          98
C                                                                       99
C        RESTORE FILE1                                                 100
C                                                                      101
      READ(PS) TEST1,TEST2,TEST3,STIME                                 102
      WRITE(F1)TEST1,TEST2,TEST3,STIME                                 103
      DO 4 I=1,NACT                                                    104
      NACT=NCT(I)                                                      105
      NACT1=NACT-1                                                     106
      NNCT2=2*NACT                                                     107
      READ(PS) (TIMCST(J),J=1,NNCT2)                                   108
      WRITE(F1)(TIMCST(J),J=1,NNCT2)                                   109
      DO 4 J=1,NNCT1                                                   110
      READ(PS) FILE1                                                   111
      WRITE(F1)FILE1                                                   112
4     CONTINUE                                                         113
      ENDFILE F1                                                       114
C                                                                      115
C        RESTORE FILE2                                                 116
C                                                                      117
      IF(K.NE.4.AND.K.NE.5.AND.K.NE.10)GO TO 6                         118
      DO 5 I=1,NACT                                                    119
      READ(PS) FILE2                                                   120
      WRITE(F2)FILE2                                                   121
```

```
5     CONTINUE                                                          122
      ENDFILE F2                                                        123
6     CONTINUE                                                          124
C                                                                       125
C        RESTORE FILE3                                                  126
C                                                                       127
      READ(PS) FILE3A                                                   128
      WRITE(F3)FILE3A                                                   129
      IF(K.NE.5.AND.K.NE.6)GO TO 7                                      130
      READ(PS) FILE3B                                                   131
      WRITE(F3)FILE3B                                                   132
7     CONTINUE                                                          133
      IF(K.LT.7)GO TO 10                                                134
8     READ(PS)M                                                         135
      IF(M.EQ.0)GO TO 9                                                 136
      WRITE(F3)M                                                        137
      MM=2+3*M                                                          138
      READ(PS) (FILE3B(I),I=1,MM)                                       139
      WRITE(F3)(FILE3B(I),I=1,MM)                                       140
      GO TO 8                                                           141
9     CONTINUE                                                          142
      ENDFILE F3                                                        143
10    CONTINUE                                                          144
C                                                                       145
C        RESTORE FILE4                                                  146
C                                                                       147
      READ(PS) NCYC,TT,NFLAG                                            148
      IF(TTT.NE.0.0)TT=TTT                                              149
      WRITE(F4)NCYC,TT,NFLAG                                            150
      IF(K.NE.6.AND.K.NE.7.AND.K.NE.11)GO TO 12                         151
      READ(PS) JNACT                                                    152
      WRITE(F4)JNACT                                                    153
      DO 11 I=1,JNACT                                                   154
      READ(PS) CDF                                                      155
      WRITE(F4)CDF                                                      156
11    CONTINUE                                                          157
12    CONTINUE                                                          158
C                                                                       159
C        RESTORE FILE5                                                  160
C                                                                       161
      READ(PS) IOPT,NT,PCT,PD                                           162
      IF(IFLG.EQ.0)GO TO 21                                             163
      IOPT=IIOPT                                                        164
      PCT=PPCT                                                          165
      PD=PPD                                                            166
21    WRITE(F5)IOPT,NT,PCT,PD                                           167
      IF(K.LT.7)GO TO 15                                                168
      READ(PS) NOINS                                                    169
      WRITE(F5)NOINS                                                    170
      READ(PS) FILE5                                                    171
      WRITE(F5)FILE5                                                    172
      IF(K.NE.8.AND.K.NE.12)GO TO 15                                    173
13    READ(PS)       ID,IEOF,(FD(I),X(I),I=1,IEOF)                      174
      IF(ID.EQ.-1)GO TO 14                                              175
      WRITE(F5)      ID,IEOF,(FD(I),X(I),I=1,IEOF)                      176
      GO TO 13                                                          177
14    CONTINUE                                                          178
      ENDFILE F5                                                        179
15    CONTINUE                                                          180
C                                                                       181
C        RESTORE FILE6                                                  182
```

```
C                                                                       183
      IF(K.LT.3)GO TO 17                                                184
      NACT2=2*NACT                                                      185
      DO 16 I=1,NBREAK                                                  186
      READ(PS) (FILE6(J),J=1,NACT2)                                     187
      WRITE(F6)(FILE6(J),J=1,NACT2)                                     188
16    CONTINUE                                                          189
      ENDFILE F6                                                        190
17    CONTINUE                                                          191
C                                                                       192
C        RESTORE FILE7                                                  193
C                                                                       194
      IF(K.LT.5)GO TO 20                                                195
18    READ(PS)       LIST                                               196
      IF(LIST(1).EQ.-1)GO TO 19                                         197
      WRITE(F7)      LIST                                               198
      GO TO 18                                                          199
19    CONTINUE                                                          200
      ENDFILE F7                                                        201
20    CONTINUE                                                          202
C                                                                       203
C     WRITE TERMINATION INFORMATION                                     204
C                                                                       205
      WRITE(6,113)   NCYC,IPROG                                         206
109   STOP                                                              207
110   FORMAT(2A4,I2,F10.5,2I5,2F10.5)                                   208
111   FORMAT(1H0,5X,'THE NAME SPECIFIED FOR THE LAST PROGRAM EXECUTED IS  209
     * INVALID.')                                                       210
112   FORMAT(1H1,132('*')/1X,132('*')/'0THIS IS THE OUTPUT FOR THE SYSTE  211
     *M RESTART PROGRAM: READFIL'/1H0,132('*')/1X,132('*'))             212
113   FORMAT(1H0,5X,'THE OPERATIONAL INFORMATION GENERATED BY THE STATIS  213
     *TICAL PERT PROGRAMS HAS BEEN SUCCESSFULLY RESTORED'/6X,   'TO THE  214
     *TEMPORARY DATA SETS.',            //6X,'THE PROCEDURE WAS INTERRUPT  215
     *ED DURING THE ',I5,'-TH ITERATION.'/6X,2A4,' WAS THE LAST PROGRAM  216
     *EXECUTED.')                                                       217
      END                                                               218
```

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

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

1. ORIGINATING ACTIVITY (Corporate author)	2a. REPORT SECURITY CLASSIFICATION
Texas A&M University	Unclassified
	2b. GROUP: Unclassified

3. REPORT TITLE

A USER'S GUIDE TO THE COMPUTER IMPLEMENTATION OF THE NEW PROJECT SCHEDULING PROCEDURE: STATISTICAL PERT

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

Technical Report

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

Thomas C. Baker, Jr. and Robert L. Sielken Jr.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
August 1978	301	12

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
N00014-78-C-0426	Report No. 57
b. PROJECT NO. NR047-179	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT

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

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

This report documents a new project scheduling procedure developed at the which Institute of Statistics, Texas A&M University. The project scheduling algorithm is a five step iterative procedure capable of determining a minimum cost project schedule when the activities making up the project have durations which are random variables. The cost of an activity is assumed to be a convex piecewise linear function of the activity's mean duration. The problem is to determine the activity mean durations which both minimize the total project cost and insure that the mean (or some specified percentile) of the corresponding project completion time distribution is less than or equal to a specified project deadline. The entire distribution of the project's completion time under the minimum cost schedule is a valuable by-product. Information on the trade-off between the project's minimum cost and its specified deadline is also provided.

This report is intended as a user's guide to the new project scheduling procedure and its computer implementation. The report includes a description of the project scheduling problem, a general overview of the scheduling procedure including references to technical reports documenting the development of the procedure, and an example of the procedure's performance. The documentation of the computer implementation includes specific input instructions; sample input and output; flowcharts; individual program descriptions; technical details concerning temporary data sets, job control language, and program interruption and restart procedures; and program listings.

ATTACHMENT III

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6801

Unclassified

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Statistical PERT						
Project Scheduling						
Minimum Cost Scheduling						
Random Activity Durations						
Project Completion Time Distribution						
Network Duration Distribution						
User's Guide						
Computer Programs for PERT						

ATTACHMENT III (Continued)

BASIC DISTRIBUTION LIST FOR
UNCLASSIFIED TECHNICAL REPORTS

Operations Research	Copies
Office of Naval Research (Code 434) Arlington, Virginia 22217	3
Director, Naval Research Laboratory Attn: Library, Code 2029 (ONRL) Washington, D.C. 20390	2
Defense Documentation Center Cameron Station Alexandria, Virginia 22314	12
Defense Logistics Studies Information Exchange Army Logistics Management Center Fort Lee, Virginia 23801	1
Technical Information Center Naval Research Laboratory Washington, D.C. 20390	6
Office of Naval Research New York Area Office 207 West 24th Street New York, New York 10011 Attn: Dr. J. Laderman	1
Director, Office of Naval Research Branch Office 1030 East Green Street Pasadena, California 91101 Attn: Dr. A.R. Laufer	1
Yale University Department of Statistics Box 2179 - Yale Station New Haven, CT 06520 Attn: Prof. I.R. Savage	1
Mr. William Dejka Code 360 Naval Electronics Laboratory Center San Diego, California 92132	1
Professor Lucien A. Schmit, Jr. Room 6731, Boelter Hall School of Engrng. & Applied Sci. University of California Los Angeles, California 90024	1

Director	Copies
Office of Naval Research Branch Office 495 Summer Street Boston, Massachusetts 02210 Attn: Dr. A.L. Powell	1
Prof. Emanuel Parzen Institute of Statistics Texas A&M University College Station, TX 77843	1
Office of Naval Research San Francisco Area Office 760 Market St. - Room 447 San Francisco, California 94103	1
Technical Library Naval Ordnance Station Indian Head, Maryland 20640	1
Bureau of Naval Personnel Navy Department Technical Library Washington, D.C. 20370 STOP 82	2
Library, Code 0212 Naval Postgraduate School Monterey, California 93940	1
Naval Ordnance Station Louisville, Kentucky 40214	1
Library Naval Electronics Laboratory Center San Diego, California 92152	1
Naval Ship Engineering Center Philadelphia Division Technical Library Philadelphia, Pennsylvania 19112	1
Dr. A.L. Slafkosky Scientific Advisor Commandant Marine Corps (Code AX) Washington, D.C. 20380	1

	Copies
Purdue University Graduate School of Ind. Admin. Lafayette, Indiana 47907 Attn: Prof. Andrew Whinston	1
Department of Economics Tisch Hall - 5th Floor Washington Square, NY Univ. New York City, NY 10003 Attn: Prof. O. Morgenstern	1
Dept. of Statistics Syntex Research 3401 Hillview Palo Alto, CA 94304 Attn: Prof. Stuart Bessler	1
University of California Department of Engineering Los Angeles, California 90024 Attn: Prof. R.R. O'Neill	1
Maritime Transportation Research Board National Academy of Sciences 2101 Constitution Avenue Washington, D.C. 20418 Attn: RADM J.B. Oren USCG (RET)	1
Stanford University Department of Operations Research Stanford, California 94305 Attn: Prof. A.F. Veinott, Jr.	1
Texas A&M Foundation College Station, Texas 77843 Attn: Prof. H.O. Hartley	1
Case Western Reserve University Cleveland, Ohio 44106 Attn: Prof. B.V. Dean	1
University of California Center for Research in Management Science Berkeley, California 94720 Attn: Prof. W.I. Zangwill	1
U.S. Naval Postgraduate School Department of Operations Research and Economics Monterey, California 93940 Attn: Prof. C.R. Jones	1

	Copies
Dr. Del Cilmer Naval Weapons Center (Code 607) China Lake, California 93555	1
Dr. Paul Murrill Project Themis Department of Chemical Engineering Louisiana State University Baton Rouge, Louisiana 70803	1
Director Army Materials & Mechanics Research Center Attn: Mr. J. Bluhm Watertown, Massachusetts 02172	1
Commanding Officer U.S. Army Ballistic Research Laboratories Attn: Dr. John H. Giese Aberdeen Proving Ground, Maryland 21005	1
University of Iowa Department of Mechanics and Hydraulics Iowa City, Iowa 52240 Attn: Prof. W.F. Ames	1
Office of Naval Research Resident Representative 1105 Guadalupe Lowich Building Austin, Texas 78701 Attn: Mr. Francis M. Lucas	1
Dr. Jerome Bracken Institute of Defense Analyses 400 Army-Navy Drive Arlington, Virginia 22202	1
Professor Richard L. Fox School of Engineering Case Western Reserve University Cleveland, Ohio 44106	1
	1

	Copies
	1
University of Chicago Statistical Research Center Chicago, Illinois 60637 Attn: Prof. W. Kruskal	1
Stanford University Department of Statistics Stanford, California 94305 Attn: Prof. G.J. Lieberman	1
	1
Princeton University Department of Mathematics Princeton, New Jersey 08540 Attn: Prof. J.W. Tukey	1
Stanford University Department of Statistics Stanford, California 94305 Attn: Prof. T.W. Anderson	1
University of California Department of Statistics Berkeley, California 94720 Attn: Prof. P.J. Bickel	1
University of Washington Department of Mathematics Seattle, Washington 98105 Attn: Prof. Z.W. Birnbaum	1
Cornell University Department of Mathematics White Hall Ithaca, New York 14850 Attn: Prof. J. Kiefer	1

	Copies
Harvard University Department of Statistics Cambridge, Massachusetts 02139 Attn: Prof. W.G. Cochran	1
Columbia University Department of Industrial Engineering New York, New York 10027 Attn: Prof. C. Derman	1
New York University Institute of Mathematical Sciences New York, New York 10453 Attn: Prof. W.M. Hirsch	1
University of North Carolina Statistics Department Chapel Hill, North Carolina 27515 Attn: Prof. W.L. Smith and Prof. M.R. Leadbetter	1
Purdue University Division of Mathematical Sciences Lafayette, Indiana 47079 Attn: Prof. H. Rubin	1
University of California, San Diego Department of Mathematics P.O. Box 109 La Jolla, California 92038 Attn: Prof. M. Rosenblatt	1
Florida State University Department of Statistics Tallahassee, Florida 32306 Attn: Prof. R.A. Bradley	1
New York University Department of Industrial Engineering & Operations Research Bronx, New York 10453 Attn: Prof. J.H.K. Kao	1
University of Wisconsin Department of Statistics Madison, Wisconsin 53706 Attn: Prof. G.E.P. Box	1

	Copies
Logistics Research Project The George Washington University 707 - 22nd Street, N.W. Washington, D. C. 20037 Attn: Dr. W. H. Marlow	1
International Business Machines Corporation P.O. Box 218, Lamb Estate Yorktown Heights, New York 10598 Attn: Dr. Alan Hoffman	1
University of California Management Sciences Research Project Los Angeles, California 90024 Attn: Dr. J.R. Jackson	1
Harvard University Department of Economics Cambridge, Massachusetts 02138 Attn: Prof. K.J. Arrow	1
Cowles Commission for Research in Economics Yale University New Haven, Connecticut 06520 Attn: Prof. Martin Shubik	1
Carnegie-Mellon University Graduate School of Industrial Administration Pittsburgh, Pennsylvania 15213 Attn: Prof. G. Thompson	1
University of California Department of Economics Berkeley, California 94720 Attn: Prof. R. Radner	1
University of California Operations Research Center Institute of Engineering Research Berkeley, California 94720 Attn: Prof. D. Gale	1
University of California Graduate School of Business Administration Los Angeles, California 90024 Attn: Prof. J. Marschak	1
The University of Michigan Department of Mathematics, W.E. Ann Arbor, Michigan 48104 Attn: Prof. R.M. Thrall	1
Princeton University Department of Mathematics Princeton, New Jersey 08540 Attn: Prof. A.W. Tucker	1
Case Western Reserve University Systems Research Center Cleveland, Ohio 44106 Attn: Prof. M. Mesarovic	1
University of Texas Department of Mathematics Austin, Texas 78712 Attn: Dr. A. Charnes	1
Stanford University Department of Operations Research Stanford, California 94305 Attn: Dr. D.L. Iglehart	1
University of Delaware Department of Mathematics Newark, Delaware 19711 Attn: Prof. R. Remage, Jr.	1
Stanford University Department of Operations Research Stanford, California 94305 Attn: Prof. F.S. Hillier	1
Dr. Claude-Alain Burdet Asst. Prof. Industrial Admin. Carnegie-Mellon University Pittsburgh, Pennsylvania 15213	1
Stanford University Department of Operations Research Stanford, California 94305 Attn: Prof. G. B. Dantzig	1
Chief of Naval Research (Code 436) Department of the Navy Arlington, Va. 22217	1
Science Librarian Kresge Library Oakland University Rochester, Michigan 48063	1

	Copies
University of Connecticut Department of Statistics Storrs, Connecticut 06268 Attn: Prof. H.O. Posten	1
	1
ARCON Corporation Lakeside Office Park North Avenue at Route 128 Wakefield, Massachusetts 01880 Attn: Dr. A. Albert	1
Stanford University Department of Statistics Stanford, California 94305 Attn: Prof. H. Chernoff	1
Yale University Department of Statistics New Haven, Connecticut 06520	1
Rutgers-The State University Statistics Center New Brunswick, New Jersey 08903 Attn: Prof. H.F. Dodge	1
Yale University Department of Statistics New Haven, Connecticut 06520 Attn: Prof. F. J. Anscombe	1
Purdue University Division of Mathematical Sciences Lafayette, Indiana 47907 Attn: Prof. S.S. Gupta	1
Cornell University Department of Industrial Engineering Ithaca, New York 19850 Attn: Prof. R.E. Bechhofer	1
Mrs. Barbara Eaudi Univ. Program Coordinator, B.E. NASA Johnson Space Center Houston, TX 77058	1

	Copies
Professor Geoffrey S. Watson, Chairman Department of Statistics Princeton University Fine Hall Princeton, N. J. 08540	1
Stanford University Department of Mathematics Stanford, California 94305 Attn: Prof. S. Karlin	1
University of Sheffield Department of Probability and Statistics Sheffield 10, ENGLAND Attn: Prof. J. Gani	1
University of California Operations Research Center Institute of Engineering Research Berkeley, California 94720 Attn: Prof. R.E. Barlow	1
Stanford University Department of Statistics Stanford, California 94305 Attn: Prof. H. Solomon	1
Applied Mathematics Laboratory Naval Ships Research Development Center Washington, D.C. 20007	1
Systems Analysis Division Room BE760, Pentagon Washington, D.C. 20350 Attn: Mr. A.S. Rhodes, Op-964	1
Department of Statistics University of North Carolina Chapel Hill, North Carolina 27515 Attn: Prof. M.R Leadbetter	1
Southern Methodist University Department of Statistics Dallas, Texas 75222 Attn: Prof. D.B. Owen	1
Israel Institute of Technology Technion Haifa, ISRAEL Attn: Prof. P. Naor	

Office of Gifts & Exchanges	Copies
Library Texas A&M University College Station, Texas 77843	2
Archives Texas A&M University College Station, TX 77843	1