TEXAS A&M UNIVERSITY

COLLEGE STATION, TEXAS 77843

THEMIS

TEXAS A&M UNIVERSITY
PROJECT THEMIS

Technical Report No. 51

STATISTICAL PERT: AN IMPROVED SUBNETWORK ANALYSIS PROCEDURE

by

R. L. Sielken Jr., H. O. Hartley, R. K. Spoeri

Texas A&M Research Foundation
Office of Naval Research
Contract N00014-68-A-0140
Project NR047-700

ATTACHMENT I

STATISTICAL PERT: AN IMPROVED SUBNETWORK ANALYSIS PROCEDURE

by

R. L. Sielken Jr., H. O. Hartley, R. K. Spoeri

THEMIS OPTIMIZATION RESEARCH PROGRAM,
Technical Report No. 51

INSTITUTE OF STATISTICS
Texas A&M University
November 1975

Research conducted through the
Texas A&M Research Foundation
and sponsored by the
Office of Naval Research
Contract N00014-68-A-0140
Project NR047-700

ATTACHMENT II

TABLE OF CONTENTS

ABSTRACT

R. L. Sielken Jr., H. O. Hartley, R. K. Spoeri

Statistical PERT is a new procedure for obtaining information about the distribution of a project's completion time when the project is comprised of a large number of activities and the time required to complete an individual activity once it can be begun is a random variable. The project is represented as an acyclic network whose arcs correspond to the project activities. This network is simplified by replacing various activity configurations by single equivalent activities and then decomposed into several subnetworks. The distribution and moments of each subnetwork's completion time are bounded and approximated on the basis of two points from each activity's completion time distribution by using some mathematical programming techniques and a new result in the theory of networks. The project's completion time distribution is then approximated by combining the approximate subnetwork distributions.

This report describes several refinements in the subnetwork analysis procedure. One major refinement greatly reduces the computational effort in obtaining bounds on the project completion time moments and distribution. A second major refinement allows the two-point approximation of an activity's completion time distribution to better represent skewed distributions. The computer programs required to implement the new subnetwork analysis procedure are listed and documented.

Statistical PERT: An Improved Subnetwork Analysis Procedure

R. L. Sielken Jr., H. O. Hartley, R. K. Spoeri

The well-known Program Evaluation and Review Technique (PERT) is concerned with a 'project' comprised of a large number of 'activities' which are arranged as the arcs in a complex acyclic network (see e.g. Figure 1). The activities at any network node 'commence' as soon as <u>all</u> activities 'terminating' at that node are completed. The time required to complete an activity once it can be begun is a random variable, and hence the time needed to complete the entire project is also a random variable.

In Technical Report No. 48 "Statistical Critical Path Analysis in Acyclic Networks: Statistical PERT" a comprehensive new procedure for obtaining information on the project completion time and its distribution was described and illustrated. That procedure involved the following five general steps:

Step 1: Identification

Represent the project and its component activities in terms of an acyclic network with one source and one sink. Identify each activity's completion time distribution or at least two points on each activity's completion time distribution.

Step 2: Simplification

Replace various activity configurations and their associated completion time distributions by a single equivalent activity and completion time distribution.

Step 3: Decomposition

Decompose the simplified network into several subnetworks by separating parallel subnetworks and then separating the resulting subnetworks at each cut vertex. A cut vertex is any node such that every path from the source to the sink passes through it.

Step 4: Analysis

Each subnetwork arising from Step 3 is analyzed on the basis of two points from each component activity's completion time distribution. The result of this analysis is an approximation of each subnetwork's completion time distribution and the moments of this distribution.

Step 5: Synthesis

Combine the approximate subnetwork completion time distributions determined in Step 4. The result is an approximate completion time distribution for the entire project.

The purpose of this report is to document several refinements in the subnetwork analysis step, Step 4.

1. Analysis of a Subnetwork

The analytical procedure described in this section yields the following information on each subnetwork when each component activity's completion time distribution is replaced by a discrete two-point distribution:

(a) Upper and lower bounds on the mean subnetwork completion time as well as the other moments of the subnetwork completion time.

(b) Upper and lower bounds on the distribution function of the subnetwork completion time.

(c) An approximate distribution function of the subnetwork completion time.

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

The analysis of each subnetwork involves essentially two parts:

1. The formation of "clusters" of activities whose effect on the subnetwork completion time seems to be interrelated.
2. The approximation of the subnetwork completion time moments and distribution on the basis of the clusters.

1.1 Formation of Clusters

The actual completion time distribution of each individual activity, A, in the subnetwork is replaced by a discrete distribution with probability P at the lower point, ℓ_A, and Q = 1-P at the upper point, u_A.

Let n be the number of activities in the subnetwork. Then for each of the 2^n combinations of the ℓ_A's and u_A's there will be a subnetwork completion time (a critical path time). The r-th moment of these 2^n times will be denoted by T_r, and the distribution function of these times will be denoted by F. The approximation of the T_r's (especially T_1, the mean) and F is the goal of the subnetwork analysis. Since n will usually be fairly large, the complete enumeration of the 2^n critical path times will usually be unreasonable. Hence the activities which are most likely to be on the critical path through the subnetwork are identified and their joint behavior investigated.

The mean of the completion time distribution for activity A is defined to be $m_A = P\ell_A + Qu_A$. The standard deviation of the completion time distribution for activity A is defined to be $s_A = \sqrt{P\ell_A^2 + Qu_A^2 - m_A^2}$

and is assumed to be positive. (The assumption that $s_A > 0$ is not really a practical restriction since the difference between a fixed activity completion time and one with a very small dispersion is negligible from a practical viewpoint.) The subnetwork's critical path when each activity's completion time is set equal to its mean will be referred to as the "original" critical path. The activities on this critical path will be referred to as "critical activities" with K equalling the number of such activities. Some non-critical activities might become critical if some of the completion times for the original critical activities were decreased. These activities are identified as follows. The completion time for one critical activity, say A, is set equal to $\max\{m_A - \lambda s_A, 0\}$ where λ is a non-negative algorithm parameter which the user specifies. All other completion times are set equal to their means. Then the longest path through the resulting network is determined. Any activities on this path which were not on the original critical path are now referred to as the"associates" of A since the effect of these "associates" on the networks' completion time is related to A's completion time. This procedure is repeated for each original critical activity.

Each critical activity and its associates make up one "cluster". These K initial clusters are now "pooled" by combining any two clusters with at least one activity in common. In general there will still be more than one cluster, and many of the n - K non-critical activities will not occur in any cluster.

The associates correspond to the activities which become critical when the completion times of the original critical activities are lowered. However, some of the originally non-critical activities may also become critical if their completion times exceed their means and the completion times of the original critical activities are at their means. These activities are identified next. Each originally non-critical activity is investigated separately. If activity A is being investigated, then the completion time for A is set equal to $m_A + \theta s_A$ where θ is a non-negative algorithm parameter which the user specifies. The completion times for all other activities are set equal to their means, and the corresponding critical path determined. This critical path will either be the original critical path or a new path which includes A. In the latter case, the activities on the original critical path which are not on a new critical path containing A are called the "eliminants" of A. Thus, the effect of A's eliminants on the networks completion time is related to the completion time for A. Hence, A is added to any cluster which contains at least one of A's eliminants. After this procedure has been repeated for each originally non-critical activity, the resultant clusters are "pooled" again by combining any two clusters with at least one activity in common.

Although the number of clusters is reduced when the pooling on the basis of the associates occurs and then further reduced when the pooling on the basis of the eliminants occurs, there will generally remain more than one cluster and several activities not in any cluster.

In general the larger the values of λ and θ the greater the number of activities in the clusters and the smaller the number of clusters. In particular the procedure for forming the clusters has the following 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 originally non-critical activity A there exists θ_A such that A will have some eliminants for any $\theta \geq \theta_A$.

Property 4: For any fixed value of λ, the set of activities in the union of the clusters is monotically non-decreasing as $\theta \to \infty$.

Property 5: There exists a finite value θ^* such that if $\theta \geq \theta^*$, then every activity would be in some cluster.

Property 6: The number of clusters, originally K, is non-increasing as $\theta \to \infty$.

Property 7: There exists a finite value θ^* such that if $\theta \geq \theta^*$, then there would only be one cluster.

Most of the properties of the cluster formation procedure are fairly straightforward; however, Property 7 requires some special justification. This justification is based on the following definition and theorem which is proven in Appendix A.

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

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

Property 3 implies that all activities will belong to some cluster if $\theta \geq \theta^*$ and

$$\theta^* = \max\{\theta_A\text{: A originally non-critical}\}.$$

Now consider any two consecutive activities A_1 and A_2 on the original critical path. Theorem 1 implies that there is a bridge over A_1 and A_2, say A_3. Since the original critical path passes through A_1 and A_2, A_3 cannot be on the original 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 original critical path activities belong to the same cluster when $\theta \geq \theta^*$, there is only one cluster when $\theta \geq \theta^*$ and Property 7 is established.

1.2 Approximate Subnetwork Completion Time Moments and Distribution

1.2.1 A Lower Bound on T_r and an Upper Bound on F

For each cluster C let n_c denote the number of activities in C, and let $v = 1, \ldots, 2^{n_c}$ index the 2^{n_c} configurations of activity completion times when

(a) the completion time for each activity A not in C is equal to its lower point, ℓ_A, and

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

Let

$$t_v = \text{critical path time for the v-th configuration}$$

and

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

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

where

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

$= 0$ if the time for the i-th activity in C is ℓ_i in the v-th configuration.

Then

$$\hat{T}_r^-(C) \equiv \sum_{v=1}^{2^{n_c}} p_v t_v^r$$

and

$$\hat{F}^+(t;\, C) \equiv \sum_{v=1}^{2^{n_c}} p_v\, I_t(t_v)$$

where

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

are the r-th moment of the 2^{n_c} critical path times and their distribution function respectively. Let

$$T_r^-(\theta,\, \lambda) = \max_C \hat{T}_r^-(C)$$

and

$$F^+(t;\, \theta,\, \lambda) = \min_C \hat{F}^+(t;\, C)$$

which depend on θ and λ since the composition and number of clusters depend on θ and λ.

The first step in showing that $T_r^-(\theta,\, \lambda)$ is a lower bound for T_r is proving the following theorem:

<u>Theorem 2</u>: For any cluster C, any positive integer r, and any activity A not in C,

$$\hat{T}_r^-(C \cup \{A\}) \geq \hat{T}_r^-(C).$$

Proof: Consider any particular critical path for a particular one of the 2^{n_c} combinations of upper and lower points involved in $\hat{T}_r^-(C)$. Consider the following two cases:

(i) activity A with its completion time equal to ℓ_A is on the critical path, and

(ii) activity A with its completion time equal to ℓ_A is not on the critical path.

Let $\delta_A = (u_A - \ell_A)$ and the particular critical path time be t. Then in case (i)

(a) if the completion time for A is set equal to u_A, this will increase the r-th moment of the critical path time by $Q[(t + \delta_A)^r - t^r]$; and

(b) if the completion time for A is set equal to ℓ_A, this will not alter the r-th moment of the critical path time.

In case (ii)

(a) if the completion time for A is set equal to u_A, this may increase the r-th moment of the critical path time by $Q[(t + \delta_A)^r - t^r]$ or less, and

(b) if the completion time for A is set equal to ℓ_A, this will not alter the r-th moment of the critical path time.

Therefore in either case the contribution to $\hat{T}_r^-(C \cup \{A\}) - \hat{T}_r^-(C)$ for this particular critical path will be between 0 and $Q[(t + \delta_A)^r - t^r]$. Since the contribution is non-negative for each particular combination,

$$\hat{T}_r^-(C \cup \{A\}) \geq \hat{T}_r^-(C). \qquad QED$$

A straightforward application of Theorem 2 yields the following theorem:

<u>Theorem 3</u>: For any two clusters C_1 and C_2 and any positive integer r,

$$\hat{T}_r^-(C_1 \cup C_2) \geq \max\{\hat{T}_r^-(C_1), \hat{T}_r^-(C_2)\}.$$

Property 2 of the cluster formation procedure implies that if θ is increased the clusters expand or are pooled. Thus, Theorems 2 and 3 imply that, for fixed λ, $T_r^-(\theta, \lambda)$ is non-decreasing as θ increases. Furthermore, Properties 5 and 7 together imply that for θ sufficiently large there is only one cluster and all of subnetwork activities are in that cluster. Hence, for θ sufficiently large $T_r^-(\theta, \lambda) = T_r$, and the following theorem is true:

<u>Theorem 4</u>:

(a) $T_r^-(\theta, \lambda)$ is a non-decreasing function of θ for any fixed values of λ and r;

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

$$T_r^-(\theta, \lambda) = T_r$$

for any λ, and r; and

(c) for any θ, λ, and r

$$T_r^-(\theta, \lambda) \leq T_r.$$

Similarly, the first step in showing that $F^+(t; \theta, \lambda)$ is an upper bound on F(t) is the following theorem:

<u>Theorem 5</u>: For any cluster C, any value of t, and any activity A not in C,

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

Proof: Consider any particular configuration of activity times used to determine $F^+(t; C)$ before C is augmented by A. When A is added to C, this particular configuration will appear once with A at its upper percentile and once with A at its lower percentile. When A is at its lower percentile, the configuration's critical path time is unchanged. However, when A is at its upper percentile, the configurations' critical path time is either unchanged or possibly increased if A were on the configuration's critical path. Thus, the addition of A leaves the cumulative probability associated with critical path times less than or equal to t either unchanged or decreased. *QED*

A straightforward extension of Theorem 5 is the following theorem:

Theorem 6: For any two clusters C_1 and C_2 and any t,

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

Since Property 2 of the cluster formation procedure implies that the clusters expand or are pooled if θ is increased, Theorem 5 and Theorem 6 together imply that, for all t and any fixed λ, $F^+(t; \theta, \lambda)$ is non-increasing function of θ. Furthermore, Properties 5 and 7 together imply that for θ sufficiently large there is only one cluster and all of the subnetwork activities are in that cluster. Hence, for θ sufficiently large $F^+(t; \theta, \lambda) = F(t)$ and the following theorem is true:

Theorem 7:

(a) $F^+(t; \theta, \lambda)$ is a non-increasing function of θ for every t and any λ;

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

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

for every t and λ; and

(c) for any θ, λ, and t

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

1.2.2 An Upper Bound on T_r and a Lower Bound on F

For each cluster C let n_c denote the number of activities in C, and let $v = 1, \ldots, 2^{n_c}$ index the 2^{n_c} configurations of activity completion times when

(a) the completion time for each activity not in the cluster is equal to its upper point, and

(b) the completion times for the activities in the cluster are at each of the 2^{n_c} possible combinations of their upper and lower percentiles.

Let t_v, p_v, and $I_t(t_v)$ be as before and define

$$\hat{T}_r^+(C) = \sum_{v=1}^{2^{n_c}} p_v\, t_v^r,$$

$$T_r^+(\theta,\ \lambda) = \min_C \hat{T}_r^+(C)$$

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

and

$$F^-(t;\ \theta,\ \lambda) = \max_C \hat{F}^-(t;\ C).$$

Then an argument completely analogous to that used to prove Theorem 4 and Theorem 7 leads to the following theorems:

Theorem 8:

(a) $T_r^+(\theta, \lambda)$ is a non-increasing function of θ for any fixed values of λ and r;

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

$$T_r^+(\theta, \lambda) = T_r$$

for any λ and r; and

(c) for any θ, λ, and r

$$T_r \leq T_r^+(\theta, \lambda).$$

Theorem 9:

(a) There exists a finite value θ^* such that $\theta \geq \theta^*$ implies

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

for every t and any λ; and

(b) for any θ, λ, and t

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

1.2.3 Summary

Theorems 2-9 together imply that for any value of θ and λ chosen by the algorithm user:

(a) $T_r^-(\theta, \lambda) \leq T_r \leq T_r^+(\theta, \lambda)$, for any positive integer r; and

(b) $F^-(t; \theta, \lambda) \leq F(t) \leq F^+(t; \theta, \lambda)$ for any t.

They also imply that, for any r, λ, and t,

$$T_r^+(\theta, \lambda) - T_r^-(\theta, \lambda)$$

and

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

would decrease monotonically to zero if θ were increased. In fact, Theorems 2-9 imply that there exists a value θ* which doesn't depend on r, λ, or t such that $\theta \geq \theta^*$ implies that

$$T_r^-(\theta, \lambda) = T_r = T_r^+(\theta, \lambda)$$

and

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

A reasonable approximation for F(t) is

$$\hat{F}(t; \theta, \lambda) = \tfrac{1}{2}[F^-(t; \theta, \lambda) + F^+(t; \theta, \lambda)].$$

2. Comparison with the Original Subnetwork Analysis Procedure

The original subnetwork analysis procedure documented in Technical Report No. 48 formed the clusters in essentially the same way as the new subnetwork analysis procedure described in this report except that in the original procedure P always equalled ½ and θ and λ multiplied the point difference, $u_A - \ell_A$, instead of the standard deviation, $\sqrt{P\ell_A^2 + Qu_A^2 - m_A^2}$.

In the original procedure $n_U = \sum_c n_c$ denotes the number of activities in the union of the clusters, and $T_r^+(\theta, \lambda)$ is defined to be the average of the r-th power of the 2^{n_U} critical path times when

(a) the completion time for each activity not in the union of the clusters is equal to its upper point, and

(b) the completion times for the activities in the union of the clusters are at each of the 2^{n_U} possible combinations of their upper and lower points.

Correspondingly, $F^-(t;\ \theta,\ \lambda)$ was defined to be the proportion of these 2^{n_U} critical path times that were less than or equal to t. Analogously, $F^+(t;\ \theta,\ \lambda)$ was the proportion of the 2^{n_U} critical path times less than or equal to t when

(a) the completion time for each activity not in the union of the clusters is equal to its lower point, and

(b) the completion times for the activities in the union of the clusters are at each of the 2^{n_U} possible combinations of their upper and lower points.

The only problem with this procedure is that 2^{n_U} may be quite large even for relatively small values of $(\theta,\ \lambda)$. For example, if the original critical path contains 10 activities and each critical activity has one associate, then $2^{n_U} = 2^{20} = 1{,}048{,}576$, and the determination of $T_r^+(\theta,\ \lambda)$, $F^-(t;\ \theta,\ \lambda)$, and $F^+(t;\ \theta,\ \lambda)$ requires the evaluation of over 2 million critical path times. On the other hand, in the new procedure the determination of $T_r^-(\theta,\ \lambda)$, $T_r^+(\theta,\ \lambda)$, $F^-(t;\ \theta,\ \lambda)$, $F^+(t;\ \theta,\ \lambda)$ requires the evaluation of only $2\sum_c 2^{n_c}$ critical path times, 80 in the example. Thus the new procedure greatly reduces the computational effort required to bound the project completion time moments and distribution.

A practical alternative to evaluating all 2^{n_U} critical path times called for in the original procedure is to randomly sample the 2^{n_U} critical path times when n_U is large and base the bounds on the sample critical path times. A computer program for the original procedure with a sampling option is documented in Appendix C. (This program supercedes the subnetwork analysis program given in Technical Report No. 48.) It should be noted that the original subnetwork analysis procedure with the sampling option in effect is still superior to a simple Monte Carlo simulation of the subnetwork since the subnetwork analysis procedure

(i) provides the information in terms of associates and eliminants about which activities play an important role in determining the subnetwork's completion time and the interactions among activities, and

(ii) samples only those activities which most influence the subnetwork completion time,

Furthermore, the loss in accuracy due to sampling in the subnetwork analysis procedure seems to be quite minimal even for relatively small sample sizes - see for example Table 2 and Table 3.

All specific computational results documented in this report are for the project network given in Figure 1. (This network is the simplified network from a large naval PERT problem.) A listing of each activity's two-point approximation except for its P and Q is given in Table 1.

Sampling can also be used in the new subnetwork analysis procedure in the somewhat unlikely event that θ and λ are chosen so large that 2^{n_c} for some cluster C is too large. In this case the probability that the v-th configuration of activity completion times $v = 1, \ldots, 2^{n_c}$ is selected on a sampling trial is

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

where

Figure 1. A Project Network

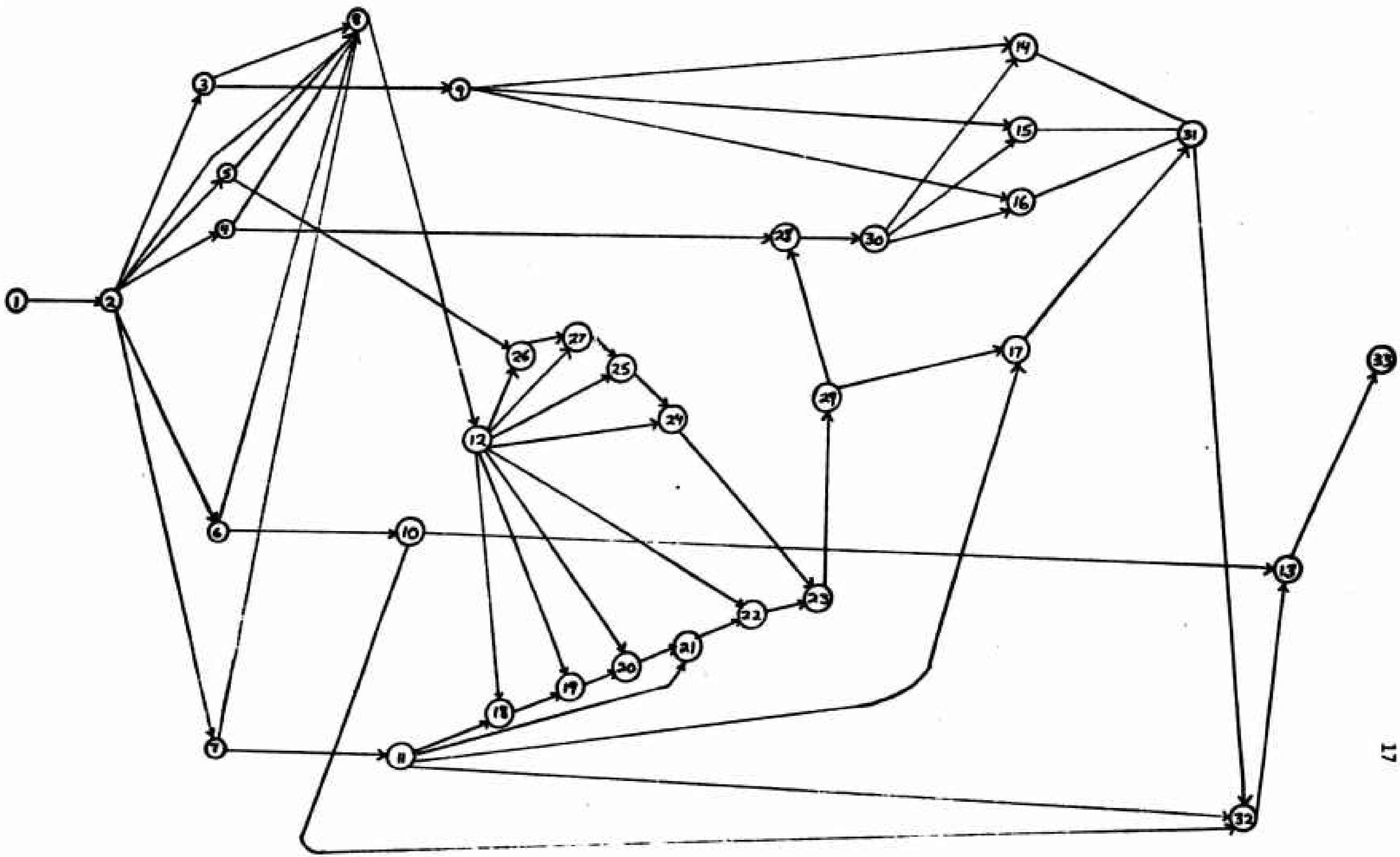


Table 1. The Upper and Lower Points in the Two-Point Approximations to the Activity Completion Time Distributions for the Project Network in Figure 1

Activity Number	Origin Node	Terminal Node	Lower Point	Upper Point
1	1	2	0.0	0.0
2	8	12	336.27	429.47
3	2	3	57.47	89.96
4	2	4	57.47	89.96
5	2	5	57.47	89.96
6	2	6	57.47	89.96
7	2	7	57.47	89.96
8	3	8	68.96	107.95
9	4	8	68.96	107.95
10	5	8	68.96	107.95
11	6	8	68.96	107.95
12	7	8	68.96	107.95
13	2	8	150.36	193.49
14	6	10	333.96	403.85
15	3	9	333.96	403.85
16	7	11	355.10	409.85
17	11	18	141.75	221.90
18	10	13	672.36	783.11
19	9	14	560.89	660.00
20	9	15	560.89	660.00
21	9	16	560.89	660.00
22	11	17	542.80	638.71
23	12	18	111.10	173.92
24	18	19	256.03	346.98
25	12	19	302.80	400.67
26	12	20	311.95	410.71
27	11	21	423.58	530.74
28	12	22	315.35	415.71
29	19	20	7.66	11.99
30	20	21	16.77	22.55
31	21	22	11.49	17.99
32	22	23	39.54	48.87
33	12	26	301.91	400.86
34	5	26	767.09	892.74
35	12	27	350.31	460.06
36	12	24	382.32	464.98
37	12	25	385.28	461.54
38	25	24	11.49	17.99
39	24	23	16.28	23.00
40	26	27	7.66	11.99
41	27	25	20.86	28.29
42	4	28	810.17	976.10
43	23	29	15.32	23.99

Activity Number	Origin Node	Terminal Node	Lower Point	Upper Point
44	28	30	15.32	23.99
45	14	31	57.47	89.96
46	16	31	49.81	77.96
47	15	31	53.64	83.96
48	17	31	88.12	137.94
49	31	32	3.83	6.00
50	32	13	49.81	77.96
51	13	33	109.01	152.63
52	11	32	745.50	811.32
53	10	32	714.74	799.83
54	30	14	0.0	0.0
55	30	16	0.0	0.0
56	30	15	0.0	0.0
57	29	17	0.0	0.0
58	29	28	0.0	0.0

Table 2. The Effect of Sampling in the Original Subnetwork Analysis Procedure: Percentiles of the Project Completion Time Distribution

Percentiles	Sample Sizes						
	$(\theta,\lambda) = (0,0)$			$(\theta,\lambda) = (.25,0)$			
	$2^{n_U} = 2^6$	2^5	2^4	$2^{n_U} = 2^{11}$	1000	500	200	100
.05	1360	1363	1377	1381	1381	1381	1384	1384
.10	1383	1390	1413	1409	1406	1411	1406	1414
.15	1403	1413	1423	1424	1421	1429	1424	1423
.20	1416	1423	1436	1441	1436	1443	1441	1446
.25	1430	1430	1446	1451	1448	1451	1451	1453
.30	1446	1453	1466	1458	1456	1461	1458	1463
.35	1459	1479	1472	1468	1466	1471	1468	1471
.40	1476	1486	1476	1478	1473	1478	1476	1478
.45	1499	1499	1512	1483	1483	1483	1483	1483
.50	1542	1542	1516	1488	1486	1491	1491	1488
.55	1578	1578	1575	1496	1496	1498	1498	1498
.60	1578	1578	1575	1501	1501	1505	1501	1503
.65	1602	1602	1579	1513	1510	1513	1513	1513
.70	1605	1605	1602	1515	1515	1515	1523	1515
.75	1605	1618	1605	1525	1525	1525	1528	1525
.80	1621	1618	1605	1533	1533	1530	1538	1530
.85	1622	1622	1618	1543	1543	1543	1545	1543
.90	1645	1622	1622	1558	1558	1558	1570	1577
.95	1648	1645	1648	1585	1585	1582	1587	1597
.975	1648	1648	1648	1607	1607	1605	1605	1622
.99	1648	1648	1648	1625	1625	1625	1622	1647
1.00	1648	1648	1648	1649	1649	1649	1649	1649

Table 3. The Effect of Skewed Activity Completion Time Distributions on the Subnetwork Analysis Procedures.

Percentile	Monte Carlo	Original Subnetwork Analysis Procedure (Sample Size = 1000/cluster)		New Subnetwork Analysis Procedure (Sample Size = 1000/cluster)	
		Estimated Percentiles of the Project Completion Time Distribution			
		1st Run	2nd Run	1st Run	2nd Run
.05	1405	1402	1396	1370	1374
.10	1423	1416	1416	1390	1398
.15	1435	1430	1433	1402	1410
.20	1447	1438	1443	1425	1437
.25	1457	1448	1449	1437	1449
.30	1467	1453	1456	1450	1453
.35	1477	1462	1466	1461	1461
.40	1486	1469	1473	1484	1500
.45	1496	1476	1479	1504	1524
.50	1507	1481	1483	1528	1528
.55	1516	1487	1489	1528	1528
.60	1527	1495	1496	1528	1528
.65	1538	1500	1499	1563	1560
.70	1549	1510	1512	1567	1567
.75	1560	1517	1516	1579	1579
.80	1580	1526	1529	1587	1587
.85	1603	1541	1542	1598	1602
.90	1627	1550	1552	1610	1614
.95	1675	1575	1582	1622	1673
1.00	2019	1646	1648	1720	1720

*The "1st run" and "2nd run" correspond to two different samples with different initializations of the random number generator.

$$\delta_{v,i} = 1 \quad \text{if the time for the i-th activity in C is } u_i \text{ in the v-th configuration}$$
$$= 0 \quad \text{if the time for the i-th activity in C is } \ell_i \text{ in the v-th configuration.}$$

Then with $I_t(t_v)$ as before the estimated bounds from C are

$$\hat{T}_r^-(C) = \frac{2^{n_c}}{N} \sum_{v=1}^{2^{n_c}} w_v p_v t_v^r$$

and

$$\hat{F}^+(t; C) = \frac{2^{n_c}}{N} \sum_{v=1}^{2^{n_c}} w_v p_v I_t(t_v)$$

where w_v is the number of times the v-th configuration appears in the sample, N is the sample size, and t_v is the critical path time corresponding to the v-th configuration of activity completion times with the completion time for each activity A not in C equal to ℓ_A. Similar modifications are made for $T_r^+(C)$ and $F^-(t; C)$. The corresponding estimators $T_r^-(\theta, \lambda)$, $T_r^+(\theta, \lambda)$, $F^-(t; \theta, \lambda)$, and $F^+(t; \theta, \lambda)$ are all unbiased.

A computer implementation of the new subnetwork analysis procedure including the sampling option is documented in Appendix B. In addition to reducing the computational effort, the new subnetwork analysis procedure allows the user to specify any probabilities (P, Q) for (ℓ, u) instead of requiring (1/2, 1/2). This would not be of great significance if all activity completion time distributions were symmetric. However, since many completion time distributions are skewed, the ability to specify (P, Q) can be a real advantage. To exemplify this advantage, the completion time for each activity in Figure 1 was taken to be a linearly transformed chi-square random variable, $(\chi_3^2 - c_1)/c_2$, where c_1 and c_2 were determined so that the points in Table 1 corresponded to the 15-th and 85-th percentiles respectively. This made the activity's completion time distribution highly skewed. Then the corresponding project completion time distribution was approximated using

(i) a Monte Carlo simulation of size 1000,

(ii) the new subnetwork analysis procedure with (P, Q) = (1/2, 1/2) for all activities, and

(iii) the new procedure with each activity's (P, Q) chosen so that the mode and first two moments of its two-point approximation equalled the mode and first two moments of its transformed chi-square distribution.

The results are given in Table 3. If the Monte Carlo approximation is used as a basis for comparison, the value of being able to specify (P, Q) is obvious.

The Monte Carlo PERT simulation program used in the above experiment is documented in Appendix D and supercedes the Monte Carlo PERT program given in Technical Report No. 48.

In the special case where every activity's (P, Q) is (1/2, 1/2), the $\hat{T}_r^-(C)$ and $\hat{F}^+(t; C)$ can be computed on the basis of the 2^{n_c} critical path times corresponding to

(a) the completion time for each activity A not in C being equal to the mean, m_A, and

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

This is a change from the usual computation in that the activities not in C are at their means here instead of their lower points. This change will tend to improve the estimators $T_r^-(\theta, \lambda)$ and $F^+(t; \theta, \lambda)$. However, this change is only guaranteed not to invalidate Theorems 4 and 7 when all (P, Q) are (1/2, 1/2). The computer implementation of the new subnetwork analysis procedure includes the option to make this change.

3. Conclusion

The new subnetwork analysis procedure

(i) forms associates, eliminants, and clusters in essentially the same way as the original procedure,

(ii) bounds the project's completion time moments and distribution primarily on the basis of the individual clusters instead of on a pooled cluster, and

(iii) approximates an activity's completion time distribution by a two-point distribution with possibly unequal probabilities for the two points instead of always equal probabilities.

The advantage of (ii) is that much much fewer critical path times need to be evaluated in determining the bounds on the project completion time. The advantage to (iii) is the ability to better approximate skewed activity completion time distributions.

Computer implementations of the new subnetwork analysis procedure, the original subnetwork analysis procedure, and a Monte Carlo simulation algorithm are documented in Appendices B, C, and D respectively. Both subnetwork analysis procedures include options to use sampling for large clusters.

References

Sielken, R. L. Jr., L. J. Ringer, H. O. Hartley, and E. Arseven, "Statistical Critical Path Analysis in Acyclic Stochastic Networks: Statistical PERT," Institute of Statistics, Texas A&M University Project Themis Technical Report No. 48, November 1974.

APPENDIX A

Proof of Theorem 1

The principle objective of this appendix is to prove the following theorem:

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

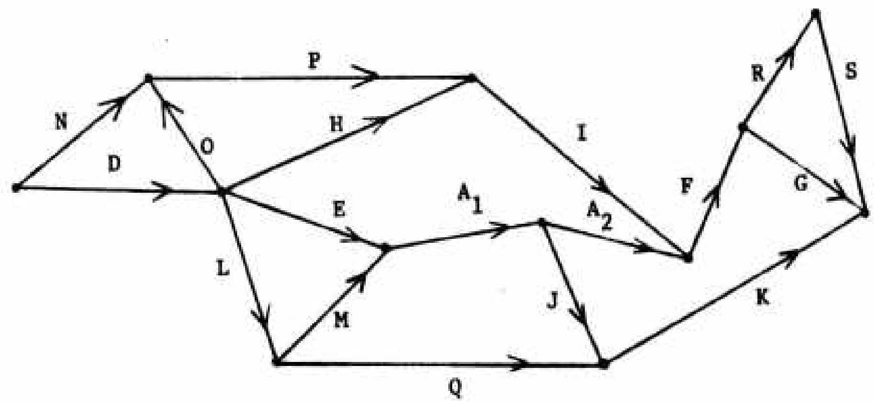
The networks considered in this appendix are assumed to be acyclic, have no cut vertices, and have one source and one sink. Also the two arcs A_1 and A_2 are any two adjacent (consecutive) arcs with A_1 preceding A_2.

Definition 1: A bridge over A_1 and A_2 is any arc, say 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.

Definition 2: An origin violator of A_1 and A_2 is an arc, say A_3, such that there exists a path from the terminal node of A_1 to the sink which passes through A_3.

Definition 3. A terminal violator of A_1 and A_2 is an arc, say A_3, such that there exists a path from the source to the terminal node of A_1 which passes through A_3.

An intuitive feeling for these definitions can be obtained by considering any path P* from the source to the sink which passes through A_1 and A_2. If an arc A_3 is an origin violator, then there is a path from the source to the sink which follows along P* through the terminal node of A_1 and then goes through A_3. This path "originates" from P* too late for A_3 to be a bridge over A_1 and A_2. Similarly, if A_3 is a terminal violator, then there is a path from the source to the sink which passes through A_3 and then joins into P* before P* passes through the terminal node of A_1. This path "terminates" into P* too early for A_3 to be a bridge.



BRIDGES over A_1 and A_2: H, I, N, O, P, and Q

ORIGIN VIOLATORS of A_1 and A_2: A_2, F, G, J, K, R, and S

TERMINAL VIOLATORS of A_1 and A_2: A_1, D, E, L, and M

The following three lemmas are straightforward consequences of the definitions of a bridge, an origin violator, and a terminal violator.

Lemma 1: Every branch in the network is either a bridge over A_1 and A_2, an origin violator of A_1 and A_2, or a terminal violator of A_1 and A_2.

Proof of Lemma 1: Suppose that A_3 is not a bridge. Then there exists a path P from the source to the sink which contains A_3 and either A_1 or A_2.

Suppose that P contains A_1. If $A_3 = A_1$ or A_3 precedes A_1 on P, then P contains a path from the source to the terminal node of A_1 which passes through A_3, and A_3 would be a terminal violator. On the other hand, if A_3 follows A_1 on P, then P contains a path from the terminal node of A_1 to the sink which passes through A_3, and A_3 would be an origin violator.

Suppose that P contains A_2. If $A_3 = A_2$ or A_3 comes after A_2 on P, then P contains a path from the terminal node of A_1 (the origin node of A_2) to the sink which passes through A_3, and A_3 would be an origin violator. If A_3 comes before A_2 on P, then P contains a path from the source to the terminal node of A_1 (the origin node of A_2) which passes through A_3, and A_3 would be a terminal violator. This completes the proof of Lemma 1.

Lemma 2: A_1 is a terminal violator of A_1 and A_2, and A_2 is an origin violator of A_1 and A_2.

Lemma 3: Any arc A_3 cannot be both an origin violator of A_1 and A_2 and a terminal violator of A_1 and A_2.

Proof of Lemma 3: Suppose that an arc A_3 is both an origin violator and a terminal violator. Since A_3 is an origin violator, there exists a path from the terminal node of A_1 to the origin node of A_3. Since A_3 is a terminal violator, there exists a path from the terminal node of A_3 to the terminal node of A_1. The existence of these two paths, however, implies the existence of a circuit which contradicts the given acyclic structure of the network. This completes the proof of Lemma 3.

Proof of Theorem 1: Since the terminal node of A_1 cannot be a cut vertex, there exists a path P from the source to the sink which does not pass through the terminal node of A_1. Denote the arcs on P by $C_1, C_2, \ldots, C_p$ with C_{i-1} preceding C_i on P.

Suppose that none of $C_1, C_2, \ldots, C_p$ are bridges over A_1 and A_2. Then Lemma 1 and Lemma 3 together imply that each of $C_1, C_2, \ldots, C_p$ is either an origin violator of A_1 and A_2 or a terminal violator of A_1 and A_2 but not both. Since the origin node of C_1 is the source, C_1 cannot be an origin violator and must be a terminal violator. Similarly, since the terminal node of C_p is the sink, C_p cannot be a terminal violator and must be an origin violator. Hence, there exists $j \geq 1$ such that $C_1, C_2, \ldots, C_j$ are all terminal violators and C_{j+1} is an origin violator.

Since C_j is a terminal violator, there exists a path from the terminal node of C_j to the terminal node of A_1. Furthermore, since C_{j+1} is an origin violator, there is a path from the terminal node of A_1 to the origin node of C_{j+1} (the terminal node of C_j). These two paths imply the existence of a circuit from the terminal node of

C_j to the terminal node of A_1 and then back to the terminal node of C_j. (The definition of P implies that the terminal node of C_j is not the terminal node of A_1.) This contradicts the given acyclic structure of the network and completes the proof of Theorem 1.

APPENDIX B

New Subnetwork Analysis Program

The New Subnetwork Analysis Program is an implementation of the analytical procedure described in Section 1 of this report. The basic required input is

(a) an acyclic network with one source and one sink,

(b) two points from each component activity's completion time distribution

(c) probability P to be associated with the lower point, and

(d) specified values for the algorithm parameters θ and λ.

The output is mainly

(a) upper and lower bounds on the moments of the network completion time, $T_r^+(\theta, \lambda)$ and $T_r^-(\theta, \lambda)$ $r = 1, 2, \ldots, 10$;

(b) upper and lower bounds on the distribution function of the network completion time, $F^+(\cdot; \theta, \lambda)$ and $F^-(\cdot; \theta, \lambda)$; and

(c) an approximate network completion time distribution, $F(\cdot; \theta, \lambda) = \frac{1}{2}[F^+(\cdot; \theta, \lambda) + F^-(\cdot; \theta, \lambda)]$.

The basic computational technique for determining critical path times is the Simplex Algorithm. This algorithm is applied to the dual problem. The Simplex Algorithm is used instead of the standard network analysis techniques because the Simplex Algorithm is ideally suited for the type of parametric programming required to evaluate several critical path times when only the activity times vary from one problem to the next.

A listing of the Subnetwork Analysis Program and a program flowchart are given at the end of this appendix.

Specific Input Instructions:

Card 1. Col. 1-3: The number of activities in the network, Format (I3).

Col. 4-6: The number of nodes in the network, Format (I3).

For each activity one card with:

Col. 11-15: The origin node of the activity, Format (I5).

Col. 21-25: The terminal node of the activity, Format (I5).

Col. 31-40: The lower point on the activity's completion time distribution, Format (F10.0).

Col. 41-50: The upper point on the activity's completion time distribution, Format (F10.0).

Col. 51-60: The probability P to be associated with the lower point (1-P will be associated with the upper point), Format (F10.5).

Next Card. Col. 1: OPTON1. OPTON1=1 implies that the program will terminate after the clusters have been formed on the basis of associates and eliminants. OPTON≠1 implies that the program will follow the normal procedure.

Col. 2: OPTON2. OPTON2=1 implies that the lower bounds on the moments of the project completion time and the upper bound on its distribution will be determined by using all activity times outside the cluster at their means instead of their lower points. This is only guaranteed to be a valid procedure when all (P, Q) = (1/2, 1/2). OPTON2≠1 implies that the program will follow the normal procedure.

Next Card. Col. 1-3: IEDF. The program computes an absolute upper and lower bound for the network completion time. This range is subdivided into IEDF equal parts and the approximate distribution function (F^+, F^-, $\hat{F}$) values are printed at each of these dividing points. IEDF would usually be between 10 and 100. IEDF, Format (I3).

Next Card. Col. 1-5: θ, Format (F5.2).

Col.6-10: λ, Format (F5.2).

Next Card. Col.1-10: SAMSIZ. The number of activity time configurations to be randomly selected for explicit consideration in each cluster analysis. If SAMSIZ < 0 or SAMSIZ > 2^{n_c}, all activity time configurations will be explicitly considered - no random sampling will be done. Format (I10).

The nodes should be numbered 1, 2, ..., n with the source being number 1, the sink being number n, and the other node numbers being arbitrary. The activities should be numbered 1, 2, ... in any order desired.

<u>Current Dimension Restrictions</u>:

Currently the program is dimensioned for a maximum of

60 Activities

40 Nodes

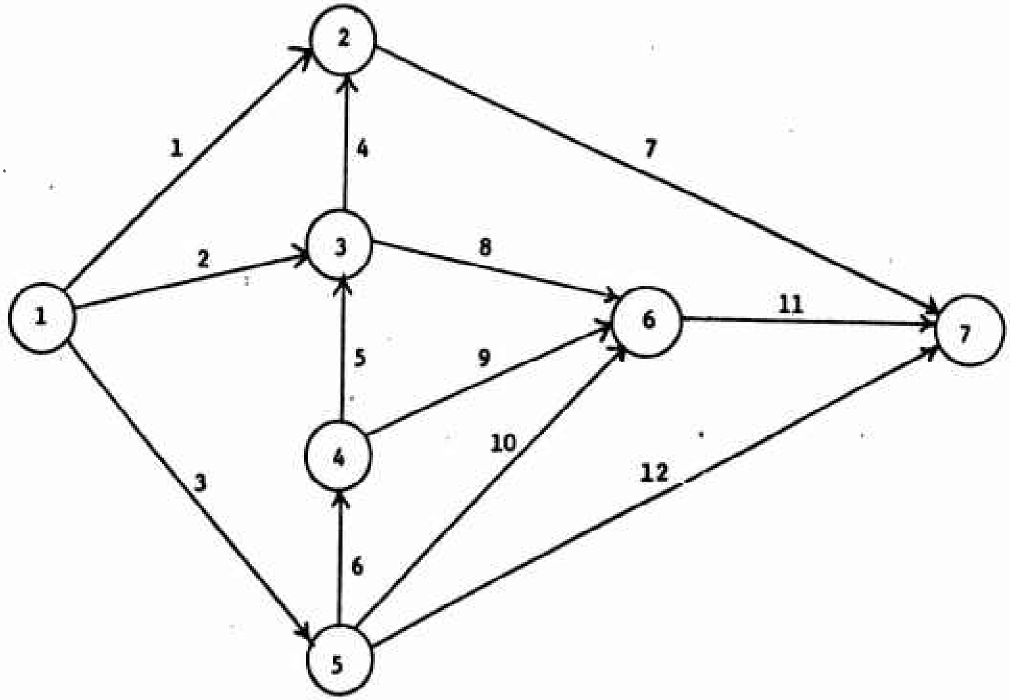
25 Clusters

25 Activities/Cluster and IEDF $\leq$ 100.

<u>Example</u>:

The Program's input and output are illustrated in terms of the network in Figure B-1.

Figure B-1: New Subnetwork Analysis Program Example Network



SAMPLE INPUT

```
012C07
                    1        2       17.26      19.44      .5
                    1        3       19.26      21.44      .9
                    1        5       12.76      15.91      .3
                    3        2        3.51       4.01      .8
                    4        3        3.01       5.43      .75
                    5        4        3.52       4.25      .5
                    2        7       13.75      14.48      .65
                    3        6        5.05       8.43      .1
                    4        6        5.36       6.51      .7
                    5        6        8.78      11.44      .55
                    6        7       15.76      17.21      .8
                    5        7       14.32      18.35      .9
00
020
1.    1.
         -1
```

SAMPLE OUTPUT

INITIAL INPUT

ACTIVITY	ORIGIN	TERMINAL	LOWER POINT	UPPER POINT	MEAN	STANDARD DEVIATION	PROB. LOWER PT.
1	1	2	17.2600	19.4400	18.3500	1.0900	0.5000
2	1	3	19.2600	21.4400	19.4780	0.6540	0.9000
3	1	5	12.7600	15.9100	14.9650	1.4435	0.3000
4	3	2	3.5100	4.0100	3.6100	0.2000	0.8000
5	4	3	3.0100	5.4300	3.6150	1.0479	0.7500
6	5	4	3.5200	4.2500	3.8850	0.3650	0.5000
7	2	7	13.7500	14.4900	14.0055	0.3482	0.6500
8	3	6	5.6500	9.4300	9.0920	1.0140	0.1000
9	4	6	5.3600	6.5100	5.7050	0.5270	0.7000
10	5	6	8.7800	11.4400	9.9770	1.3233	0.5500
11	6	7	15.7600	17.2100	16.0500	0.5800	0.8000
12	5	7	14.3200	18.3500	14.7230	1.2090	0.9000

THE CRITICAL PATH TIME WHEN EACH ACTIVITY'S COMPLETION TIME IS SET EQUAL TO ITS MEAN IS = 0.466070 02

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

THE 5 CRITICAL ACTIVITIES ARE AS FOLLOWS BEGINNING WITH THE TERMINAL ACTIVITY:
11. 8. 5. 6. 3.

THETA = 0.10000D 01 LAMBDA = 0.10000D 01

A LOWER BOUND ON THE EXPECTED CRITICAL PATH TIME IS = 0.40100D 02

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

THE 5 CRITICAL ACTIVITIES ARE AS FOLLOWS BEGINNING WITH THE TERMINAL ACTIVITY:
11. 8. 5. 6. 3.

A UPPER BOUND ON THE EXPECTED CRITICAL PATH TIME IS = 0.51230D 02

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

THE 5 CRITICAL ACTIVITIES ARE AS FOLLOWS BEGINNING WITH THE TERMINAL ACTIVITY:
11. 8. 5. 6. 3.

```
THE ASSOCIATES ARE NOW IDENTIFIED:

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   1-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY  11. IS =   0

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   2-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY   8. IS =   0

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   3-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY   5. IS =   0

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   4-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY   6. IS =   0

     THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   5-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY   3. IS =   0
```

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

THE ACTIVITIES IN THE    1-TH CLUSTER ARE AS FOLLOWS:
      11.

THE ACTIVITIES IN THE    2-TH CLUSTER ARE AS FOLLOWS:
       8.

THE ACTIVITIES IN THE    3-TH CLUSTER ARE AS FOLLOWS:
       5.

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

THE ACTIVITIES IN THE    5-TH CLUSTER ARE AS FOLLOWS:
       3.

THE CLUSTER TO WHICH EACH ACTIVITY BELONGS:
    (ZERO IMPLIES THAT THE ACTIVITY IS NOT IN ANY CLUSTER)
     THE    1-TH ACTIVITY IS IN THE    0-TH CLUSTER
     THE    2-TH ACTIVITY IS IN THE    0-TH CLUSTER
     THE    3-TH ACTIVITY IS IN THE    5-TH CLUSTER
     THE    4-TH ACTIVITY IS IN THE    0-TH CLUSTER
     THE    5-TH ACTIVITY IS IN THE    3-TH CLUSTER
     THE    6-TH ACTIVITY IS IN THE    4-TH CLUSTER
     THE    7-TH ACTIVITY IS IN THE    0-TH CLUSTER
     THE    8-TH ACTIVITY IS IN THE    2-TH CLUSTER
     THE    9-TH ACTIVITY IS IN THE    0-TH CLUSTER
     THE   10-TH ACTIVITY IS IN THE    0-TH CLUSTER
     THE   11-TH ACTIVITY IS IN THE    1-TH CLUSTER
     THE   12-TH ACTIVITY IS IN THE    0-TH CLUSTER

THERE ARE    7 ACTIVITIES NOT IN ANY CLUSTER YET.
```

```
THE ELIMINANTS OF EACH NON-CRITICAL-PATH ACTIVITY ARE NOW DETERMINED:

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


THE COMPLETION TIME FOR THE    1-TH ACTIVITY HAS BEEN CHANGED TO       0.19440D 02

      THERE ARE    0 ELIMINANTS CORRESPONDING TO ACTIVITY    1


THE COMPLETION TIME FOR THE    2-TH ACTIVITY HAS BEEN CHANGED TO       0.20132D 02

      THERE ARE    0 ELIMINANTS CORRESPONDING TO ACTIVITY    2


THE COMPLETION TIME FOR THE    4-TH ACTIVITY HAS BEEN CHANGED TO       0.38100D 01

      THERE ARE    0 ELIMINANTS CORRESPONDING TO ACTIVITY    4


THE COMPLETION TIME FOR THE    7-TH ACTIVITY HAS BEEN CHANGED TO       0.14354D 02

      THERE ARE    0 ELIMINANTS CORRESPONDING TO ACTIVITY    7


THE COMPLETION TIME FOR THE    9-TH ACTIVITY HAS BEEN CHANGED TO       0.62320D 01

      THERE ARE    0 ELIMINANTS CORRESPONDING TO ACTIVITY    9


THE COMPLETION TIME FOR THE   10-TH ACTIVITY HAS BEEN CHANGED TO       0.11300D 02

      THERE ARE    0 ELIMINANTS CORRESPONDING TO ACTIVITY   10


THE COMPLETION TIME FOR THE   12-TH ACTIVITY HAS BEEN CHANGED TO       0.15932D 02

      THERE ARE    0 ELIMINANTS CORRESPONDING TO ACTIVITY   12
```

```
THERE ARE   5 CLUSTLRS.
     THERF ARE    1 ACTIVITIFS IN THF    1-TH CLUSTER.   THEY ARE AS FOLLOWS:
                11.
            1 CLUSTFRS HAVE RFEN POOLED TO MAKF THIS CLUSTFF. THFY WERE AS FOLLOWS:
                1.
     THERF ARF    1 ACTIVITIFS IN THE    2-TH CLUSTER.   THFY ARF AS FOLLCWS:
                8.
            1 CLUSTERS HAVF RFEN POOLED TO MAKE THIS CLUSTFF. THFY WFRE AS FOLLOWS:
                2.
     THERE ARE    1 ACTIVITIES IN THF    3-TH CLUSTEP.   THFY ARF AS FOLLCWS:
                5.
            1 CLUSTERS HAVE BEEN POOLED TO MAKE THIS CLUSTFR. THEY WERE AS FOLLOWS:
                3.
     THERE ARE    1 ACTIVITIES IN THE    4-TH CLUSTEP.   THEY ARE AS FOLLCWS:
                6.
            1 CLUSTERS HAVE RFFN POOLED TO MAKF THIS CLUSTEP. THFY WERF AS FOLLOWS:
                4.
     THERE ARF    1 ACTIVITIES IN THF    5-TH CLUSTFP.   THEY ARE AS FOLLCWS:
                3.
            1 CLUSTERS HAVF BEEN POOLEC TO MAKE THIS CLUSTEP. THFY WERE AS FOLLOWS:
                5.
THE FOLLCWING TAPLES WFRE DETERMINED CCNSIDERING ALL  ACTIVITY CCNFIGURATIONS.
THE INITIALIZATION PARAMETER FOP ANY SAMPLING IS IY =          77
```

A LOWER BOUND, T-(1;THETA.LAMBDA), ON THE 1-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.43142D 02

A LOWER BOUND, T-(2;THETA.LAMBDA), ON THE 2-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.18623D 04

A LOWER BOUND, T-(3;THETA.LAMBDA), ON THE 3-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.80428D 05

A LOWER BOUND, T-(4;THETA.LAMBDA), ON THE 4-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.34752D 07

A LOWER BOUND, T-(5;THETA.LAMBDA), ON THE 5-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.15023D 09

A LOWER BOUND, T-(6;THETA.LAMBDA), ON THE 6-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.64968D 10

A LOWER BOUND, T-(7;THETA.LAMBDA), ON THE 7-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.28108D 12

A LOWER BOUND, T-(8;THETA.LAMBDA), ON THE 8-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.12165D 14

A LOWER BOUND, T-(9;THETA.LAMBDA), ON THE 9-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.52667D 15

A LOWER BOUND, T-(10;THETA.LAMBDA), ON THE 10-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.22809D 17

AN UPPER BOUND ON THE NETWORK COMPLETION TIME DISTRIBUTION: F+(.;THETA;LAMBDA)

F+(	0.40657D 02;	0.10000D 01;	0.10000D 01)	=	0.10000D 00
F+(	0.41213D 02;	0.10000D 01;	0.10000D 01)	=	0.10000D 00
F+(	0.41770D 02;	0.10000D 01;	0.10000D 01)	=	0.10000D 00
F+(	0.42326D 02;	0.10000D 01;	0.10000D 01)	=	0.10000D 00
F+(	0.42883D 02;	0.10000D 01;	0.10000D 01)	=	0.10000D 00
F+(	0.43439D 02;	0.10000D 01;	0.10000D 01)	=	0.10000D 00
F+(	0.43996D 02;	0.10000D 01;	0.10000D 01)	=	0.10000D 01
F+(	0.44552D 02;	0.10000D 01;	0.10000D 01)	=	0.10000D 01
F+(	0.45109D 02;	0.10000D 01;	0.10000D 01)	=	0.10000D 01
F+(	0.45665D 02;	0.10000D 01;	0.10000D 01)	=	0.10000D 01
F+(	0.46221D 02;	0.10000D 01;	0.10000D 01)	=	0.10000D 01
F+(	0.46778D 02;	0.10000D 01;	0.10000D 01)	=	0.10000D 01
F+(	0.47334D 02;	0.10000D 01;	0.10000D 01)	=	0.10000D 01
F+(	0.47891D 02;	0.10000D 01;	0.10000D 01)	=	0.10000D 01
F+(	0.48447D 02;	0.10000D 01;	0.10000D 01)	=	0.10000D 01
F+(	0.49004D 02;	0.10000D 01;	0.10000D 01)	=	0.10000D 01
F+(	0.49560D 02;	0.10000D 01;	0.10000D 01)	=	0.10000D 01
F+(	0.50117D 02;	0.10000D 01;	0.10000D 01)	=	0.10000D 01
F+(	0.50673D 02;	0.10000D 01;	0.10000D 01)	=	0.10000D 01
F+(	0.51230D 02;	0.10000D 01;	0.10000D 01)	=	0.10000D 01

```
AN UPPER BOUND, T+( 1;THETA,LAMBDA), ON THE  1-TH MOMENT OF THE NETWORK COMPLETION TIME =     0.494150 02
AN UPPER BOUND, T+( 2;THETA,LAMBDA), ON THE  2-TH MOMENT OF THE NETWORK COMPLETION TIME =     0.244290 04
AN UPPER BOUND, T+( 3;THETA,LAMBDA), ON THE  3-TH MOMENT OF THE NETWORK COMPLETION TIME =     0.120830 06
AN UPPER BOUND, T+( 4;THETA,LAMBDA), ON THE  4-TH MOMENT OF THE NETWORK COMPLETION TIME =     0.597890 07
AN UPPER BOUND, T+( 5;THETA,LAMBDA), ON THE  5-TH MOMENT OF THE NETWORK COMPLETION TIME =     0.296000 09
AN UPPER BOUND, T+( 6;THETA,LAMBDA), ON THE  6-TH MOMENT OF THE NETWORK COMPLETION TIME =     0.146610 11
AN UPPER BOUND, T+( 7;THETA,LAMBDA), ON THE  7-TH MOMENT OF THE NETWORK COMPLETION TIME =     0.726550 12
AN UPPER BOUND, T+( 8;THETA,LAMBDA), ON THE  8-TH MOMENT OF THE NETWORK COMPLETION TIME =     0.360230 14
AN UPPER BOUND, T+( 9;THETA,LAMBDA), ON THE  9-TH MOMENT OF THE NETWORK COMPLETION TIME =     0.178700 16
AN UPPER BOUND, T+(10;THETA,LAMBDA), ON THE 10-TH MOMENT OF THE NETWORK COMPLETION TIME =     0.886940 17
```

A LOWER BOUND ON THE NETWORK COMPLETION TIME DISTRIBUTION: F-(.;THETA;LAMBDA)

F-(	0.406657D 02;	0.100000D 01;	0.100000D 01) =	0.000000D 00
F-(	0.412130D 02;	0.100000D 01;	0.100000D 01) =	0.000000D 00
F-(	0.417700D 02;	0.100000D 01;	0.100000D 01) =	0.000000D 00
F-(	0.423260D 02;	0.100000D 01;	0.100000D 01) =	0.000000D 00
F-(	0.428830D 02;	0.100000D 01;	0.100000D 01) =	0.000000D 00
F-(	0.434390D 02;	0.100000D 01;	0.100000D 01) =	0.000000D 00
F-(	0.439960D 02;	0.100000D 01;	0.100000D 01) =	0.000000D 00
F-(	0.445520D 02;	0.100000D 01;	0.100000D 01) =	0.000000D 00
F-(	0.451090D 02;	0.100000D 01;	0.100000D 01) =	0.000000D 00
F-(	0.456650D 02;	0.100000D 01;	0.100000D 01) =	0.000000D 00
F-(	0.462210D 02;	0.100000D 01;	0.100000D 01) =	0.000000D 00
F-(	0.467780D 02;	0.100000D 01;	0.100000D 01) =	0.000000D 00
F-(	0.473340D 02;	0.100000D 01;	0.100000D 01) =	0.000000D 00
F-(	0.478910D 02;	0.100000D 01;	0.100000D 01) =	0.100000D 00
F-(	0.484470D 02;	0.100000D 01;	0.100000D 01) =	0.300000D 00
F-(	0.490040D 02;	0.100000D 01;	0.100000D 01) =	0.750000D 00
F-(	0.495600D 02;	0.100000D 01;	0.100000D 01) =	0.750000D 00
F-(	0.501170D 02;	0.100000D 01;	0.100000D 01) =	0.800000D 00
F-(	0.506730D 02;	0.100000D 01;	0.100000D 01) =	0.800000D 00
F-(	0.512300D 02;	0.100000D 01;	0.100000D 01) =	0.100000D 01

AN APPROXIMATE NETWORK COMPLETION TIME DISTRIBUTION:

F(.;THETA,LAMBDA) = .5 * (F+(.;THETA,LAMBDA) + F-(.;THETA,LAMBDA))

F(	0.40657D 02;	0.100000D 01;	0.100000D 01) =	0.500000D-01
F(	0.41213D 02;	0.100000D 01;	0.100000D 01) =	0.500000D-01
F(	0.41770D 02;	0.100000D 01;	0.100000D 01) =	0.500000D-01
F(	0.42326D 02;	0.100000D 01;	0.100000D 01) =	0.500000D-01
F(	0.42883D 02;	0.100000D 01;	0.100000D 01) =	0.500000D-01
F(	0.43439D 02;	0.100000D 01;	0.100000D 01) =	0.500000D-01
F(	0.43996D 02;	0.100000D 01;	0.100000D 01) =	0.500000D 00
F(	0.44552D 02;	0.100000D 01;	0.100000D 01) =	0.500000D 00
F(	0.45109D 02;	0.100000D 01;	0.100000D 01) =	0.500000D 00
F(	0.45665D 02;	0.100000D 01;	0.100000D 01) =	0.500000D 00
F(	0.46221D 02;	0.100000D 01;	0.100000D 01) =	0.500000D 00
F(	0.46778D 02;	0.100000D 01;	0.100000D 01) =	0.500000D 00
F(	0.47334D 02;	0.100000D 01;	0.100000D 01) =	0.500000D 00
F(	0.47891D 02;	0.100000D 01;	0.100000D 01) =	0.550000D 00
F(	0.48447D 02;	0.100000D 01;	0.100000D 01) =	0.650000D 00
F(	0.49004D 02;	0.100000D 01;	0.100000D 01) =	0.875000D 00
F(	0.49560D 02;	0.100000D 01;	0.100000D 01) =	0.875000D 00
F(	0.50117D 02;	0.100000D 01;	0.100000D 01) =	0.900000D 00
F(	0.50673D 02;	0.100000D 01;	0.100000D 01) =	0.900000D 00
F(	0.51230D 02;	0.100000D 01;	0.100000D 01) =	0.100000D 01

```
C
C          NEW SUBNETWORK ANALYSIS PROGRAM
C
      IMPLICIT REAL*8 (A-H,O-Z)
C        FOR THE SAKE OF  IDENTIFYING THE APPPROPRIATE DIMENSIONS, LET
C              M = THE NUMBER OF ACTIVITIES IN THE NETWORK
C              NMM = NUMBER OF NODES IN THE NETWORK
C              NMMP1 = NMM + 1
C              N = M + NMM
C              L = THE LENGTH OF THE CRITICAL PATH
C              C = THE MAXIMUM NUMBER OF BRANCHES IN A CLUSTER
C              IEDF = THE NUMBER OF DIVISIONS IN THE EMPIRICAL
C                     DISTRIBUTION FUNCTION
C
C
C     INTEGER TAIL( M),HEAD( M),ASSGRP( L,L ),CLINCL(L,L),EGRP(L)
C     DIMENSION NINCL( C),INCLUS( L, C),NCLINC( L)
C     DIMENSION FD(IEDF),NLEFD(L,IEDF),NSAVE(IEDF),NIB(L)
C     DIMENSION           AVG( L),THAT(L)
C     DIMENSION LEFT(M ),LEFTO(M ),NONCP(M)
C     DIMENSION INBASE(NMM),XNODE(NMM)
C     DIMENSION XB1(NMMP1),Y1(NMMP1),REDCOS(N),ISTAT(N)
C     DIMENSION ICRITP(L),NINAG(M),ICRITN(L+1),CTIME(N),COT(N)
C     DIMENSION KB(L),IBB(L),FLO(M),FHI(M),SIGMA(M),B1INV(NMMP1,NMMP1)
C     REAL          MOMENT(L,10)
C     DIMENSION PP(M),PQ(M)
C
C        OF COURSE THESE DIMENSIONS ARE MERELY UPPER BOUNDS
C
C
      COMMON B1INV,REDCOS,CTIME,XB1,INBASE,HEAD,TAIL,NMMP1,NMM,N,ISTAT
      COMMON M,MP1
      INTEGER TAIL(60),HEAD(60),ASSGRP(25,25),CLINCL(25,25),EGRP(25)
      INTEGER  SAMSIZ,RANSAM
      DIMENSION NINCL(25),INCLUS(25,25),NCLINC(25)
      DIMENSION FD(100),NIB(25)
      REAL*8 NLEFD(25,100),NSAVE(100)
      DIMENSION AVG(25),THAT(25)
      DIMENSION INBASE(40)
      DIMENSION XNODE(40)
      DIMENSION XB1(41),Y1(41),REDCOS(100),ISTAT(100)
      DIMENSION B1INV(41,41),KB(25),IBB(25),FLO(60),FHI(60),SIGMA(60)
      DIMENSION ICRITP(25),NINAG(50),ICRITN(26),COT(100),CTIME(100)
      DIMENSION LEFT(60),LEFTO(60),NONCP(60)
      REAL*8 LAMBDA,MOMENT(25,10)
      DIMENSION PP(60),PQ(60)
      INTEGER OPTON1,OPTON2
C        M = THE NUMBER OF ACTIVITIES IN THE NETWORK
C        NMM = THE NUMBER OF NODES IN THE PERT NETWORK
      READ(5,100)  M,NMM
100   FORMAT(2I3)
      N=NMM+M
      MP1=M+1
      NMMP1=NMM+1
C
C        THE ACTIVITIES ARE DESCRIBED IN TERMS OF THEIR NODES
C        II=THE TAIL NODE, THE ORIGIN NODE
C        JJ=THE HEAD NODE, THE TERMINAL NODE
C        FLO = THE LOWER POINT
```

```
C         FHI = THE UPPER POINT
C         SIGMA  = (FHI - FLO)*DSQRT( PP*(1-PP)) = STD. DEVIATION
C         PP = THE PROBABILITY OF THE LOWER POINT
C
      DO 610 I=1,M
      READ(5,2501) II,JJ,FLO(I ),FHI(I ),PP(I)
      PQ(I)=1.D0-PP(I)
      SIGMA(I)=(FHI(I)-FLO(I))*DSQRT( PP(I)*PQ(I))
 2501 FORMAT(10X,I5,5X,I5, 5X,F10.0,F10.0,F10.5)
      COT(I)=PP(I)*FLO(I)+PQ(I)*FHI(I)
      CTIME(I) = COT(I)
      TAIL(I) = II
  610 HEAD(I)=JJ
C         COT = THE ORIGINAL RIGHT-HAND SIDES, I.E. THE MEANS
C               OF FLO AND FHI
C         CTIME = THE CURRENT RIGHT-HAND SIDES
      DO 55610 I=MP1,N
55610 CTIME(I) = 0.
C
C                OPTON1 =1 IMPLIES THAT THE PROGRAM WILL TERMINATE AFTER
C                          THE CLUSTERS HAVE BEEN FORMED.  NO BOUNDS ON
C                          THE PROJECT COMPLETION TIME MOMENTS OR
C                          DISTRIBUTION WILL BE DETERMINED.
C                OPTON1  NOT= 1  IMPLIES THAT THE NORMAL PROCEDURE WILL BE
C                          FOLLOWED.
C                OPTON2 =1  IMPLIES THAT THE LOWER BOUNDS ON THE MOMENTS
C                           AND THE UPPER BOUND ON THE DISTRIBUTION WILL
C                           BE DETERMINED USING ALL ACTIVITY TIMES OUTSIDE
C                           THE CLUSTER AT THEIR MEAN.  THIS PROCEDURE IS
C                           ONLY GUARANTEED TO BE VALID WHEN ALL
C                           (P,Q) = (.5,.5) .
C                OPTON2  NOT= 1  IMPLIES THAT THE NORMAL PROCEDURE WILL BE
C                          FOLLOWED.
C
      READ(5,77551) OPTON1,OPTON2
77551 FORMAT(10I1)
      I=OPTON1+OPTON2
      IF(I.GE.1) WRITE(6,77553)
77553 FORMAT(1H1)
      IF(OPTON1 .EQ.1) WRITE(6,77552)
77552 FORMAT(1H0,10X,'OPTION1=1 AND THE PROGRAM WILL TERMINATE AFTER THE
     * CLUSTERS HAVE BEEN FORMED.',/,11X,'NO BOUNDS ON THE PROJECT COMPL
     *ETION TIME MOMENTS OR DISTRIBUTION WILL BE DETERMINED.')
      IF(OPTON2 .EQ.1) WRITE(6,77554)
77554 FORMAT(1H0,10X,'OPTION2=1 AND THE LOWER BOUNDS ON THE PROJECT COMP
     *ETION TIME MOMENTS AND THE UPPER BOUND ON THE PROJECT COMPLETION T
     *IME DISTRIBUTION',/,11X,'WILL BE DETERMINED USING ALL ACTIVITY TIM
     *ES OUTSIDE THE CLUSTER AT THEIR MEAN.',/,11X,'THIS PROCEDURE IS ON
     *LY GUARANTEED TO BE VALID WHEN ALL (P,Q) = (.5,.5) .')
C              IEDF = THE NUMBER OF DIVISIONS IN THE EMPIRICAL
C                        DISTRIBUTION FUNCTION
      READ(5,100) IEDF
      WRITE(6,2700)
 2700 FORMAT(1H1,15X,'INITIAL INPUT')
      WRITE(6,2701)
 2701 FORMAT(1H0,10X,'ACTIVITY  ORIGIN    TERMINAL  LOWER POINT  UPPER P
     *OINT      MEAN    STANDARD DEVIATION   PROB. LOWER PT.')
      DO 2704 I=1,M
 2704 WRITE(6,2702) I,TAIL(I),HEAD(I),FLO(I),FHI(I),CTIME(I),SIGMA(I),PP
     *(I)
```

```
2702  FORMAT(1H ,13X,I3,5X,I3,7X,I3,5X,F10.4,3X,F10.4,3X,F10.4,4X,F10.4,
     *14X,F6.4)
C
C
C        THE FOLLOWING INDICATORS ARE USED:
C
C
C        IPARM = 1   IMPLIES THE CRITICAL PATH TIME WHEN ALL ACTIVITY
C                    COMPLETION TIMES ARE SET EQUAL TO THEIR MEANS IS
C                    BEING DETERMINED
C        IPARM = 2   IMPLIES THAT THE LOWER BOUND ON THE COMPLETION TIME
C                     FOR THE SUBNETWORK IS BEING DETERMINED
C        IPARM = 3   IMPLIES THAT THE UPPER BOUND ON THE COMPLETION TIME
C                     FOR THE SUBNETWORK IS BEING DETERMINED
C        IPARM > 3   WHEN INDEXL=0 IMPLIES THAT THE ASSOCIATES ARE
C                    BEING DETERMINED
C
C
C        INDEXL=0  IMPLIES THAT INITIAL CLUSTERS ARE STILL BEING FORMED
C        INDEXL=1  IMPLIES THAT THE LEFTOVERS, THEIR ELIMINANTS, AND
C                        POOLED CLUSTERS ARE BEING DETERMINED
C        INDEXL=2  IMPLIES THAT THE 2**NINCL( ) RUNS FOR EACH CLUSTER
C                  ARE BEING MADE
C
C
C        ICBCP = 0 IMPLIES THAT THE PROCEDURE FOR DETERMINING UPPER
C                  BOUNDS ON THE MOMENTS OF THE NETWORK COMPLETION TIME
C                  AND LOWER BOUNDS ON THE DISTRIBUTION OF COMPLETION
C                  TIMES HAS NOT BEEN BEGUN
C        ICBCP = 1 IMPLIES THAT THE PROCEDURE FOR DETERMINING UPPER
C                  BOUNDS ON THE MOMENTS OF THE NETWORK COMPLETION TIME
C                  AND LOWER BOUNDS ON THE DISTRIBUTION OF COMPLETION
C                  TIMES IS BEING INITIALIZED
      CONTINUE
C        ICBCP = 2 IMPLIES THAT THE PROCEDURE FOR DETERMINING UPPER
C                  BOUNDS ON THE MOMENTS OF THE NETWORK COMPLETION TIME
C                  AND LOWER BOUNDS ON THE DISTRIBUTION OF COMPLETION
C                  TIMES IS BEING CARRIED OUT
C
C
C
      IPARM=1
      INDEXL=0
      ICBCP=0
6010  CONTINUE
      DO 104  I=1,NMM
  104 INPASE(I)=M+I
      DO 2001  J=1,M
2001  ISTAT(J)=0.
      DO 2002  J=MP1,N
2002  ISTAT(J)=1
      DO 10  II=1,NMMP1
      DO 12  L=1,NMMP1
   12 B1INV(L,II) = 0.
   10 B1INV(II,II) = 1.
      DO 30  I=1,NMM
   30 XB1(I) = 0.
      XB1(NMMP1) = 1.
      TOLR1=1.0D-10
C
```

```
C          START THE SIMPLEX ALGORITHM
C          SOLVE THE DUAL PROBLEM
C           THE NUMBER OF VARIABLES IS M REAL + NMM SLACKS
C           FOR A TOTAL OF N VARIABLES
C
 350    CONTINUE
  2800  DO 23 J=1,N
        RATS = 0.
        IF (ISTAT(J).EQ.1) GO TO 52800
        IF (J.GT.M) GO TO 22
        RATS =-B1INV(1,HEAD(J)+1)+B1INV(1,TAIL(J)+1) + CTIME(J)
        GO TO 52800
    22  RATS =-B1INV(1,J-M+1)
 52800  REDCOS(J)= RATS
    23  CONTINUE
 22800  CONTINUE
        IRMAX=1
        RMAX=REDCOS(1)
        DO 24 J=2,N
        IF(REDCOS(J) .LE. RMAX) GO TO 24
        RMAX=REDCOS(J)
        IRMAX=J
 24     CONTINUE
        IF(RMAX .LE. TOLR1) GO TO 401
 22824  CONTINUE
        DO 26 L=1,NMMP1
        IF (IRMAX.GT.M) GO TO 50026
        Y1(L) =-B1INV(L,TAIL(IRMAX)+1)+B1INV(L,HEAD(IRMAX)+1)
        GO TO 26
 50026  Y1(L) = B1INV(L,IRMAX-M+1)
    26  CONTINUE
        Y1(1) = Y1(1) - CTIME(IRMAX)
        NUMBER=0
        DO 27 L=2,NMMP1
 27     IF(Y1(L) .LE. TOLR1)   NUMBER=NUMBER+1
        IF(NUMBER .EQ. NMM) GO TO 403
        RMIN=.99D 20
        IRMIN=0.
        DO 32 II=2,NMMP1
        IF(Y1(II).LE. TOLR1)  GO TO 32
        RATS =XB1(II)/Y1(II)
        RR=RATS-RMIN
        IF(RR .GE. 0.D0) GO TO 32
        RMIN=RATS
        IRMIN=II
 32     CONTINUE
        DO 33 J=2,NMMP1
        WW=B1INV(IRMIN ,J)/Y1(IRMIN )
        DO 37 L=1,NMMP1
 37     B1INV(L ,J)=B1INV(L,J)-WW*Y1(L)
 33     B1INV(IRMIN ,J)=WW
C
C         UPDATE THE BASIC VARIABLES: INBASE AND XB1
C
        ISTAT(INBASE(IRMIN-1))=0
        ISTAT(IRMAX)=1
        INBASE(IRMIN-1)=IRMAX
        W=XB1(IRMIN )/Y1(IRMIN )
        DO 38 I=1,NMMP1
 38     XB1(I)=XB1(I)-Y1(I)*W
```

```
      XBI(IRMIN )=W
      GO TO 350
 403  WRITE(6,530)
 530  FORMAT(1H0,5X,'NO FEASIBLE SOLUTION EXISTS.  CHECK YOUR INPUT DATA
     *.')
      WRITE(6,850)
 850  FORMAT(1H1)
      GO TO 999
C
C        END OF THE SIMPLEX ALGORITHM
C
 401  CONTINUE
      IF(ICBCP.EQ.1) GO TO 6008
      IF(INDEXL.EQ.2) GO TO 3204
C
C        KKK= THE NUMBER OF NODES ON THE CRITICAL PATH
C        KB(L)= THE L-TH NODE IN THE CRITICAL PATH. COUNTING BACKWARDS
C                     FROM THE TERMINAL NODE
C        KKB= THE NUMBER OF ACTIVITIES ON THE CRITICAL PATH
C        IBB(L)= THE L-TH ACTIVITY ON THE CRITICAL PATH, COUNTING
C                     BACKWARDS FROM THE TERMINAL NODE
C
      CONTINUE
C
C        INBASE IS A SET OF M INTEGER VARIABLES WHICH INDICATE THE
C        COMPOSITION OF THE CURRENT BASIS.  FOR EXAMPLE,
C             INBASE(K) = 7  IMPLIES THAT THE K-TH COLUMN IN THE BASIS B
C                            CORRESPONDS TO THE 7-TH VARIABLE
C
C
C        ISTAT INDICATES THE BASIC STATUS OF EACH VARIABLE
C             ISTAT(K) = 1  IMPLIES THAT THE K-TH VARIABLE IS IN THE
C                           DUAL BASIS
C             ISTAT(K) = 0  IMPLIES THAT THE K-TH VARIABLE IS NOT IN THE
C                           DUAL BASIS
C
C
C        THE FOLLOWING STATEMENTS DETERMINE THE NODES AND ACTIVITIES ON
C        THE CRITICAL PATH
C
C
C        THE DUAL SOLUTION IMPLIES THE FOLLOWING OPTIMAL SOLUTION TO THE
C        PRIMAL PERT PROBLEM. HOWEVER SOME OF THE NODE TIMES(OTHER THAN
C        THE LAST ONE) MAY BE HIGHER THAN NECESSARY.  THUS IN
C        DETERMINING THE CRITICAL PATH AN ALTERNATIVE OPTIMAL SOLUTION
C        MAY HAVE TO BE IDENTIFIED.
C        BIINV IS NOT CHANGED.
C
      DO 83002 I=1,NMM
83002 XNODE(I)=BIINV(1,I+1)
      KKK=1
      KB(1)=NMM
83001 IK=KB(KKK)
C
C        DETERMINE WHETHER THE TIME TO REACH NODE   IK   IS NECESSARILY
C        AS LARGE AS INDICATED FROM THE DUAL SOLUTION
C
      SMIN=999999.
      ISMIN=0
      DO 83000 I=1,M
```

```
      IF(HEAD(I).NE.IK) GO TO 83000
      SLACK=XNODE(HEAD(I))-XNODE(TAIL(I))-CTIME(I)
      IF(SLACK.GE.SMIN) GO TO 83000
      SMIN=SLACK
      ISMIN=I
83000 CONTINUE
      IF(SMIN.LT.0.0001) GO TO 83003
C
C        THE TIME FOR NODE IK WAS UNNECCESSARILY LARGE
C
      XNODE(IK)=XNODE(IK)-SMIN
      KKK=KKK-1
      GO TO 83001
83003 IBB(KKK)=ISMIN
      KKK=KKK+1
      KB(KKK)=TAIL(ISMIN)
      IF(TAIL(ISMIN).GT.1) GO TO 83001
      KKB=KKK-1
      IF(INDEXL.EQ.1) GO TO 3121
      IPARM=IPARM+1
      IF(IPARM.GT.4) GO TO 2910
      IF(IPARM.EQ.3) GO TO 6400
      IF(IPARM.EQ.4) GO TO 6401
C
C        ICRITP(L)= THE L-TH ACTIVITY ON THE ORIGINAL CRITICAL PATH
C         KCPB= THE NUMBER OF ACTIVITIES ON THE ORIGINAL CRITICAL PATH
C
      TOTAL = B1INV(1,NMMP1)
      KCPB=KKB
      ICRITN(1) = NMM
      DO 2802 I=1,KCPB
      ICRITN(I+1) = KB(I+1)
2802  ICRITP(I)=IBB(I)
      X=TOTAL
      WRITE(6,851) X
851   FORMAT(1H0,5X,'THE CRITICAL PATH TIME WHEN EACH ACTIVITY''S COMPLE
     *TION TIME IS SET EQUAL TO ITS MEAN IS = ',D15.5)
      WRITE(6,7606) KKK
 7606 FORMAT(1H0,10X,'THE ',I3,' NODES ON THE CRITICAL PATH ARE AS FOLLO
     *WS BEGINNING WITH THE TERMINAL NODE:')
      WRITE(6,7707) (KB(I),I=1,KKK)
 7707 FORMAT(15X,20(I3,','))
      WRITE(6,7710) KKB
7710  FORMAT(1H0,10X,'THE ',I3,' CRITICAL ACTIVITIES ARE AS FOLLOWS BEGI
     *NNING WITH THE TERMINAL ACTIVITY:')
      WRITE(6,7707) (IBB(I),I=1,KKB)
      READ(5,2920) THETA,LAMBDA
2920   FORMAT(2F5.2)
      WRITE(6,3071) THETA,LAMBDA
3071  FORMAT(1H0,10X,'THETA   = ',E15.5,'     LAMBDA = ',E15.5)
C
C       SAMSIZ = THE NUMBER OF ACTIVITY TIME CONFIGURATIONS TO BE
C                RANDOMLY SELECTED FOR CONSIDERATION IN EACH CLUSTER
C
C     NOTE:  SINCE THIS IS A RANDOM SAMPLE . SOME PERCENTILE
C            COMBINATIONS MAY BE CONSIDERED MORE THAN ONCE.
C
      READ (5,3209) SAMSIZ
 3209 FORMAT (I10)
C
```

```
C         THE COMPLETION TIME FOR ALL ACTIVITIES IS SET TO THEIR LOWER
C         PERCENTILE.  THE RESULTING CRITICAL PATH TIME IS A LOWER
C         BOUND ON THE EXPECTED CRITICAL PATH TIME.
C
      DO 6402 I=1,M
 6402 CTIME(I) = FLO(I)
      CALL BINVA(&2800)
 6400 CPLB= BIINV(1,NMMP1)
      WRITE(6,6405) CPLB
 6405 FORMAT(1H0,5X,'A LOWER BOUND ON THE EXPECTED CRITICAL PATH TIME IS
     * = ',F15.5)
      WRITE(6,7606) KKK
      WRITE(6,7707) (KB(I),I=1,KKK)
      WRITE(6,7710) KKB
      WRITE(6,7707) (IBB(I),I=1,KKB)
C
C         THE COMPLETION TIME FOR ALL ACTIVITIES IS SET TO THEIR UPPER
C         PERCENTILE.  THE RESULTING CRITICAL PATH TIME IS A UPPER
C         BOUND ON THE EXPECTED CRITICAL PATH TIME.
C
      DO 6406 I=1,M
 6406 CTIME(I) = FHI(I)
      CALL BINVA(&2800)
 6401 CPUB= BIINV(1,NMMP1)
      WRITE(6,6409) CPUB
 6409 FORMAT(1H0,5X,'A UPPER BOUND ON THE EXPECTED CRITICAL PATH TIME IS
     * = ',E15.5)
      WRITE(6,7606) KKK
      WRITE(6,7707) (KB(I),I=1,KKK)
      WRITE(6,7710) KKB
      WRITE(6,7707) (IBB(I),I=1,KKB)
C
C         FD(I) = THE LOWER BOUND ON THE EXPECTED CRITICAL PATH TIME
C                   PLUS 1/IEDF OF THE DISTANCE TO THE UPPER BOUND
C         NLEFD(IR,I) = THE SUM OF (THE CRITICAL PATH TIME FOR A
C                        CONFIGURATION * THE PROBABILITY OF THE
C                        CONFIGURATION  --- WHEN THE CRITICAL PATH TIME IS
C                        <= FD(I) ) * (THE NUMBER OF POSSIBLE
C                        CONFIGURATIONS) / (THE SAMPLE SIZE)
C                        FOR THE IR-TH CLUSTER
C
C         FD AND NLEFD ARE USED TO BUILD AN 'EMPIRICAL' DISTRIBUTION OF
C         THE CRITICAL PATH TIMES
C
      C=(CPUB-CPLB)/IEDF
      DO 6412 K=1,KCPB
      DO 6412 I=1,IEDF
      FD(I)=CPLB+I*C
 6412 NLEFD( K,I)=0.D0
C
C         THE ASSOCIATE GROUPS ARE NOW FORMED
C
      WRITE(6,3165)
 3165 FORMAT(1H1,5X,'THE ASSOCIATES ARE NOW IDENTIFIED:')
      IIIII=1
      DO 2825 I=1,M
 2825 CTIME(I)=COT(I)
      IWWWO=ICRITP(1)
      CHANG= LAMBDA*SIGMA(IWWWO)
      TFX=COT(IWWWO)-CHANG
```

```
      IF(TEX.LT.0.0) CHANG=COT(IWWWQ)
      CTIME(IWWWQ)=COT(IWWWQ)-CHANG
      CALL BINVA(&22800)
 2801 CONTINUE
      IF(ISTAT(IWWWQ).EQ.1) CALL BINV1(&22825,COT(IWWWQ),CTIME(IWWWQ),
     *IWWWQ)
      REDCOS(IWWWQ) = REDCOS(IWWWQ)+COT(IWWWQ)-CTIME(IWWWQ)
22825 CTIME(IWWWQ)=COT(IWWWQ)
      IWWWQ=ICRITP(IIIII)
      CHANG= LAMBDA*SIGMA(IWWWQ)
      TEX=COT(IWWWQ)-CHANG
      IF(TEX.LT.0.0) CHANG=COT(IWWWQ)
      CTIME(IWWWQ)=COT(IWWWQ)-CHANG
      IF (ISTAT(IWWWQ).EQ.1) CALL BINV1(&22800,CTIME(IWWWQ),COT(IWWWQ),
     *IWWWQ)
      REDCOS(IWWWQ)=REDCOS(IWWWQ)-COT(IWWWQ)+CTIME(IWWWQ)
      GO TO 22800
C
C        DETERMINE ASSOCIATE GROUP
C
 2910 NINAG(IIIII)=0
      DO 2911 K=1,KKB
      KK=1
 2913 IF(IBB(K).EQ.ICRITP(KK)) GO TO 2911
      IF(KK.GE.KCPB) GO TO 2912
      KK=KK+1
      GO TO 2913
 2912 NINAG(IIIII)=NINAG(IIIII)+1
      ASSGRP(IIIII,NINAG(IIIII))=IBB(K)
 2911 CONTINUE
      WRITE(6,2915) IIIII,ICRITP(IIIII),NINAG(IIIII)
 2915 FORMAT(1H0,10X,'THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE ',I3,
     *'-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY ',I3,', IS = ',I3)
      IDUCK=NINAG(IIIII)
      IF(IDUCK.EQ.0) GO TO 2810
      WRITE(6,2916) (ASSGRP(IIIII,I),I=1,IDUCK)
 2916 FORMAT(1H0,15X,'THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLO
     *WS',/,15X,50(I3,','))
 2810 IIIII=IIIII+1
      IF(IIIII.LE.KCPB) GO TO 2801
C
C        DETERMINE THE CLUSTERS
C
C              THE CLUSTERS ARE POOLED TOWARD THE TERMINAL NODE
C              NCLUS = THE NUMBER OF NON-EMPTY CLUSTERS
C              NINCL(I) = THE NUMBER OF ACTIVITIES IN THE I-TH CLUSTER
C              INCLUS(I,J) = THE J-TH ACTIVITY IN THE I-TH CLUSTER
C              NCLINC(I) = THE NUMBER OF CLUSTERS COMPRISING THE I-TH
C                   CLUSTER AFTER POOLING
C              CLINCL(I,J) = THE J-TH CLUSTER WHICH HAS BEEN POOLED INTO
C                   THE I-TH CLUSTER
C
C              NCLINC AND CLINCL HELP KEEP TRACK OF WHICH CLUSTER THE
C              CRITICAL PATH ACTIVITIES ARE IN
C
C
C        BELOW FORMS CLUSTERS BY PUTTING EACH CRITICAL PATH ACTIVITY IN
C        SEPARATE CLUSTER AND THEN ADDING EACH CRITICAL PATH ACTIVITY'S
C        ASSOCIATES TO ITS CLUSTER
C
```

```
      NCLUS=KCPB
      DO 3020 I=1,KCPB
      NCLINC(I)=1
      CLINCL(I,1)=I
      NINCL(I) =NINAG(I)+1
      INCLUS(I,1)=ICRITP(I)
      IF(NINAG(I).EQ.0) GO TO 3020
      IDUCK=NINCL(I)
      DO 3021 J=2,IDUCK
      JJ=J-1
 3021 INCLUS(I,J)=ASSGRP(I,JJ)
 3020 CONTINUE
C
C         BELOW POOLS CLUSTERS FORMED FROM ASSOCIATES
C
      IA=0
 3031 IA=IA+1
      IF(IA.GE.KCPB) GO TO 3030
      IF(NCLUS.EQ.1) GO TO 3030
      IDIA=NINCL(IA)
      IF(IDIA.EQ.0) GO TO 3031
      IAA=IA+1
      DO 3023 II=IAA,KCPB
      IDII=NINCL(II)
      IF(IDII.EQ.0) GO TO 3023
      DO 3025 I=1,IDIA
      DO 3025 J=1,IDII
      IF(INCLUS(II,J).EQ.INCLUS(IA,I)) GO TO 3027
 3025 CONTINUE
      GO TO 3023
 3027 NCLUS=NCLUS-1
      DO 3028 J=1,IDII
      DO 3029 I=1,IDIA
      IF(INCLUS(II,J).EQ.INCLUS(IA,I)) GO TO 3028
 3029 CONTINUE
      NINCL(IA)=NINCL(IA)+1
      INCLUS(IA,NINCL(IA) )=INCLUS(II,J)
 3028 CONTINUE
      NINCL(II)=0
      NCLINC(IA)=NCLINC(IA)+1
      CLINCL(IA,NCLINC(IA)) = II
      NCLINC(II)=0
 3023 CONTINUE
      GO TO 3031
 3030 CONTINUE
C
C         BELOW DESCRIBES CLUSTERS AFTER POOLING BASED ON THE ASSOCIATES
C
      WRITE(6,3033) NCLUS
 3033 FORMAT(1H1,10X,'THERE ARE ',I3,' NONEMPTY CLUSTERS AFTER POOLING O
     *N THE BASIS OF ASSOCIATES ONLY.')
      II=0
      DO 3034 I=1,KCPB
      IF(NINCL(I).EQ.0) GO TO 3034
      II=II+1
      IDUCJ=NINCL(I)
      WRITE(6,3035)  I,(INCLUS(I,J),J=1,IDUCJ)
 3035 FORMAT(1H0,10X,'THE ACTIVITIES IN THE ',I3,'-TH CLUSTER ARE AS FOL
     *LOWS:',/,15X,50(I3,','))
 3034 CONTINUE
```

```
C
C        DESCRIBES WHERE EACH ACTIVITY IS BEFORE ELIMINANTS ARE
C        CONSIDERED
C
C
C        EXAMINE EACH ACTIVITY AND DETERMINE WHICH CLUSTER,IF ANY, IT IS
C        IN.
C             LEFT(I) = 0  IMPLIES THAT THE I-TH ACTIVITY IS NOT IN ANY
C                          CLUSTER
C             LEFT(I) = J  IMPLIES THE I-TH ACTIVITY IS IN THE J-TH
C                          CLUSTER
C
      WRITE(6,3104)
 3104 FORMAT(1H0,10X,'THE CLUSTER TO WHICH EACH ACTIVITY BELONGS:',/,15X
     *,'(ZERO IMPLIES THAT THE ACTIVITY IS NOT IN ANY CLUSTER)')
      DO 3101 I=1,M
      LEFT(I)=0
      DO 3102 J=1,KCPB
      IF(NINCL(J).EQ.0) GO TO 3102
      IDUCK=NINCL(J)
      DO 3110 K=1,IDUCK
      IF(I.EQ.INCLUS(J,K)) GO TO 3107
 3110 CONTINUE
 3102 CONTINUE
      GO TO 3101
 3107 LEFT(I)=J
 3101 WRITE(6,3103) I,LEFT(I)
 3103 FORMAT(1H ,15X,'THE ',I3,'-TH ACTIVITY IS IN THE ',I3,'-TH CLUSTER
     *')
      INDEXL=1
C
C        LEFTOVERS ARE ACTIVITIES NOT IN CLUSTERS AFTER ASSOCIATES HAVE
C        BEEN CONSIDERED BUT BEFORE ELIMINANTS HAVE BEEN CONSIDERED
C
C
C        DETERMINE THE NUMBER OF LEFTOVERS, NLEFT
C             LEFTO(L) = J  IMPLIES THAT THE L-TH LEFTOVER IS THE J-TH
C                           ACTIVITY
C
      NLEFT=0
      DO 3122 J=1,M
      IF(LEFT(J).NE.0) GO TO 3122
      NLEFT=NLEFT+1
      LEFTO(NLEFT)=J
 3122 CONTINUE
      WRITE(6,3123) NLEFT
 3123 FORMAT (1H0,10X,'THERE ARE ',I3,' ACTIVITIES NOT IN ANY CLUSTER YE
     *T.')
      WRITE(6,3323)
 3323 FORMAT(1H1,5X,'THE ELIMINANTS OF EACH NON-CRITICAL-PATH ACTIVITY A
     *RE NOW DETERMINED:')
C
C        ELIMINANTS FOR EACH NON-CRITICAL-PATH ACTIVITY ARE NOW
C        DETERMINED
C             NNNCP = THE NUMBER OF ACTIVITIES NOT ON THE CRITICAL PATH
C             NONCP(LE) = THE LE-TH ACTIVITY NOT ON THE CRITICAL PATH
C
      NNNCP=M-KCPB
      LE=0
      DO 5000 I=1,M
```

```
      J=1
5001  IF(I.EQ.ICRITP(J)) GO TO 5000
      J=J+1
      IF(J.LE.KCPB) GO TO 5001
5002  LE=LE+1
      NONCP(LE)=I
5000  CONTINUE
      WRITE(6,5005) NNNCP
5005  FORMAT(1H0,05X,'THERE ARE ',I3,' ACTIVITIES NOT ON THE CRITICAL PA
     *TH.  THEY ARE AS FOLLOWS:')
      IF(NNNCP.EQ.0) GO TO 3124
      DO 5006 I=1,LE
5006  WRITE(6,5007) I,NONCP(I)
5007  FORMAT(1H ,15X,I3,'.     ',I3)
      IF(NNNCP.EQ.0) GO TO 3124
      LE=0
3126  LE=LE+1
      IF (ISTAT(IWWWQ).EQ.1) CALL BINV1(&23127,COT(IWWWQ),CTIME(IWWWQ),
     *IWWWQ)
      REDCOS (IWWWQ) = REDCOS(IWWWQ)-CTIME(IWWWQ)+COT(IWWWQ)
23127 CONTINUE
      CTIME(IWWWQ) = COT(IWWWQ)
      CTIME(NONCP(LE)) = COT(NONCP(LE)) + THETA*SIGMA(NONCP(LE))
      IF (ISTAT(NONCP(LE)).EQ.1) CALL BINV1(&7756,CTIME(NONCP(LE)),
     *    COT(NONCP(LE)),NONCP(LE))
      REDCOS(NONCP(LE))=REDCOS(NONCP(LE))-COT(NONCP(LE))+CTIME(NONCP(LE)
     *)
 7756 IWWWQ = NONCP(LE)
      WRITE(6,3152) NONCP(LE),CTIME(NONCP(LE))
3152  FORMAT(1H0,///, 5X,'THE COMPLETION TIME FOR THE ',I3,'-TH ACTIVITY
     * HAS BEEN CHANGED TO ',E15.5)
      GO TO 22800
3121  CONTINUE
C
C         DETERMINE THE ELIMINANTS OF THE LE-TH ACTIVITY NOT ON THE
C         CRITICAL PATH
C                NE          = THE NUMBER OF ELIMINANTS FOR THE LE-TH
C                                ACTIVITY NOT ON THE CRITICAL PATH
C                EGRP(J)       = THE J-TH ELIMINANT FOR THE LE-TH ACTIVITY
C                                  NOT ON THE CRITICAL PATH
C
      NE=0
      DO 3130 K=1,KCPB
      DO 3131 I=1,KKB
      IF(IBB(I).EQ.ICRITP(K)) GO TO 3130
3131  CONTINUE
      NE=NE+1
      EGRP(NE)=ICRITP(K)
3130  CONTINUE
      WRITE(6,3133) NE,NONCP(LE)
3133  FORMAT(1H0,10X,'THERE ARE ',I3,' ELIMINANTS CORRESPONDING TO ACTIV
     *ITY ',I3)
      IF(NE.EQ.0) GO TO 3171
      DO 3135 K=1,NE
3135  WRITE(6,3136) K,NONCP(LE),EGRP(K)
3136  FORMAT(1H ,14X,'THE ',I3,'-TH ELIMINANT CORRESPONDING TO ACTIVITY
     *',I3,' IS ACTIVITY ',I3)
C
C         DETERMINE WHETHER NONCP(LE) IS AN ASSOCIATE
C                JA = 1   IF NONCP(LE)   IS AN ASSOCIATE
```

```
C              JA = 2  IF NONCP(LE)  IS NOT AN ASSOCIATE
C
      K=NONCP(LE)
      JA=1
      IF(LEFT(K).EQ.0) JA=2
      IF(JA.EQ.2) GO TO 5010
      IT=LEFT(K)
C
C         THE IT-TH CLUSTER IS EXPANDED TO INCLUDE ELIMINANTS
C
      GO TO 5011
 5010 CONTINUE
C
C         ITTTT IS THE ACTIVITY NUMBER OF THE FIRST ELIMINANT
C         IT IS THE CLUSTER TO WHICH THE FIRST ELIMINANT CURRENTLY BELONG
C
      ITTT=EGRP(1)
      IT=LEFT(ITTT)
      LEFT(NONCP(LE))=IT
C
C         THE IT-TH CLUSTER IS EXPANDED TO INCLUDE ELIMINANTS
C
      NINCL(IT)=NINCL(IT)+1
      INCLUS(IT,NINCL(IT))=NONCP(LE)
      IF(NE.EQ.1) GO TO 3171
 5011 DO 3172 J=JA,NE
C
C         IU IS THE ACTIVITY NUMBER OF THE NEXT ELIMINANT
C         IF IU IS IN CLUSTER K, THEN CLUSTER K IS POOLED INTO CLUSTER IT
C
      IU=EGRP(J)
      K=LEFT(IU)
 3182 IF(IT.EQ.K) GO TO 3172
      NCLUS=NCLUS-1
      IW=NCLINC(K)
      DO 3183 IA=1,IW
      LEFT(ICRITP(CLINCL(K,IA)))=IT
      NCLINC(IT)=NCLINC(IT)+1
 3183 CLINCL(IT,NCLINC(IT))=CLINCL(K,IA)
      NCLINC(K)=0
      IW=NINCL(K)
      NINCL(K)=0
      DO 3184 IA=1,IW
      LEFT(INCLUS(K,IA))=IT
      NINCL(IT)=NINCL(IT)+1
 3184 INCLUS(IT,NINCL(IT))=INCLUS(K,IA)
 3172 CONTINUE
 3171 CONTINUE
      IF(LE.LT.NNNCP) GO TO 3126
C
C         END OF POOLING BASED ON ELIMINANTS EXCEPT FOR THE FOLLOWING
C         DESCRIPTION
C
      WRITE(6,3173) NCLUS
 3173 FORMAT(1H1,05X,'THERE ARE ',I3,' CLUSTERS.')
      DO 3176 I=1,KCP9
      IF(NINCL(I).EQ.0) GO TO 3176
      IDD=NINCL(I)
      WRITE(6,3174) NINCL(I),I,(INCLUS(I,J),J=1,IDD)
 3174 FORMAT(1H0,10X,'THERE ARE ',I3,' ACTIVITIES IN THE ',I3,'-TH CLUST
```

```
     *ER.  THEY ARE AS FOLLOWS:',/,20X,50(I3,','))
      IDUCK=NCLINC(I)
      WRITE(6,3175) NCLINC(I),(CLINCL(I,J),J=1,IDUCK)
3175  FORMAT(1H0,15X,I3,' CLUSTERS HAVE BEEN POOLED TO MAKE THIS CLUSTER
     *. THEY WERE AS FOLLOWS:',/,20X,50(I3,','))
3176  CONTINUE
      IF (SAMSIZ.LE.0) WRITE(6,6045)
      IF (SAMSIZ.GT.0) WRITE(6,6056) SAMSIZ,SAMSIZ
 6056 FORMAT( 5X,' THE FOLLOWING TABLES WERE DETERMINED CONSIDERING AT M
     *OST ',I8,' ACTIVITY CONFIGURATIONS PER CLUSTER.',/,11X,'IF THERE A
     *RE NO MORE THAN ',I8,' ACTIVITY CONFIGURATIONS IN A CLUSTER.',/,11
     *X,'THEN ALL ACTIVITY CONFIGURATIONS ARE EXPLICITLY CONSIDERED, AND
     * NO SAMPLING IS DONE.')
 6045 FORMAT (5X,' THE FOLLOWING TABLES WERE DETERMINED CONSIDERING ALL
     * ACTIVITY CONFIGURATIONS.')
C
C          WARNING: THE SYSTEM SUBROUTINE CLOCK MAY NOT BE A PART OF
C          ALL  SYSTEMS.  XRAN NEEDS TO BE A RANDOM SEED.
C
      CALL CLOCK(XRAN)
      IYUTS = XRAN
      WRITE (6,3238) IYUTS
 3238 FORMAT (1H0,5X,'THE INITIALIZATION PARAMETER FOR ANY SAMPLING IS I
     *Y = ',I10)
C
C          STATEMENT NUMBER 3124 MARKS THE END OF POOLING CLUSTERS BASED
C                ON LEFTOVERS AND ELIMINANTS
C
3124  CONTINUE
      IF(OPTON1 .EQ.1) GO TO 999
C
C
C          THE FINAL CLUSTERS HAVE NOW BEEN DETERMINED
C          THE 2**NINCL(I) RUNS ARE NOW MADE FOR ALL I WITH NINCL(I)>0.
C
C
6008  IF(ICBCP.EQ.1) ICBCP=2
      INDEXL=2
      IF(OPTON2 .EQ.1) GO TO 87701
      IF(ICBCP.EQ.2) GO TO 87701
      DO 17701 I=1,M
17701 COT(I)=FLO(I)
87701 CONTINUE
      IR=0
3200  IR=IR+1
      IF(IR.GT.KCPB) GO TO 3208
      IF(NINCL(IR).EQ.0) GO TO 3200
      DO 6031 I=1,10
6031  MOMENT(IR,I) = 0.D0
      DO 9101 I=1,IEDF
9101  NLEFD(IR,I)=0.D0
3310  IP = 0
      NIB(IP)=0
C
C     NIB   = NUMBER OF ACTIVITY CONFIGURATIONS IN THE SAMPLE
C     NSAVE = VECTOR CONTAINING THE UPPER BOUNDS ON THE NETWORK
C             COMPLETION TIME DISTRIBUTION TO BE AVERAGED WITH THE
C             LOWER BOUNDS ON THE NETWORK TO YIELD THE AVERAGE NETWORK
C             COMPLETION TIME DISTRIBUTION.
C
```

```
C
      IC=NINCL(IR)
      IB=2**NINCL(IR)
C     IDOALL = 1 MEANS ALL ACTIVITY CONFIGURATIONS ARE EXPLICITLY
C                 CONSIDERED.
C     IDOALL = 0 MEANS TO SAMPLE.
C
C
      IDOALL = 0
      IF (SAMSIZ.LE.0.OR.SAMSIZ.GE.IB) IDOALL=1
      DO 3222 I=1,M
 3222 CTIME(I) = COT(I)
      GO TO 20000
C
C        STATEMENT 3204  IS THE RE-ENTRY POINT FROM THE SIMPLEX
C        ALGORITHM WHEN CLUSTER BASED BOUNDS ARE BEING COMPUTED
C
3204  CONTINUE
      DO 6032 I=1,10
6032  MOMENT(IR,I) = MOMENT(IR,I) +(B1INV(1,NMMP1)**I)*SPROB
9900  X= B1INV(1,NMMP1)
      X=X-1.0D-10
      I=0
6420  I=I+1
      IF(X.GT.FD(I)) GO TO 6420
      NLEFD(IR,I)=NLEFD(IR,I)+SPROB
10001 CONTINUE
20000 IP=IP+1
      NIB(IP)=NIB(IR)+1
C
C        GENERATE NEXT ACTIVITY CONFIGURATION TO BE EXPLICITLY
C        CONSIDERED.
C
      IF (IDOALL.EQ.0) GO TO 20500
      IF (IP.GT.IB) GO TO 3207
      RANSAM = IP
      GO TO 20501
20500 IF(NIB(IR).GT.SAMSIZ) GO TO 3207
      IYUTS = IYUTS*65539
      IF (IYUTS) 6210,6211,6211
 6210 IYUTS = IYUTS+2147483647+1
 6211 XRAN = IYUTS
      XRAN = XRAN*.4656613E-9
      RANSAM = XRAN*DFLOAT(IB-1) + 1
20501 CONTINUE
C
C     CONVERT THE RANDOM NUMBER, RANSAM, TO A BINARY NUMBER TO DEFINE AN
C     ACTIVITY CONFIGURATION.
C
      KRAN = RANSAM
      SPROB=1.D0
      DO 8505 I=1,IC
      IHALF = KRAN/2
      IZ = KRAN - IHALF*2
      L = INCLUS(IR,I)
      CTIME(L) = IZ*FHI(L)-IZ*FLO(L) + FLO(L)
      SPROB=SPROB*(IZ*PQ(L)+(1-IZ)*PP(L))*2.D0
C
C        SPROB= 2**NINCL(IR) * THE PROBABILITY OF THIS CONFIGURATION
C
```

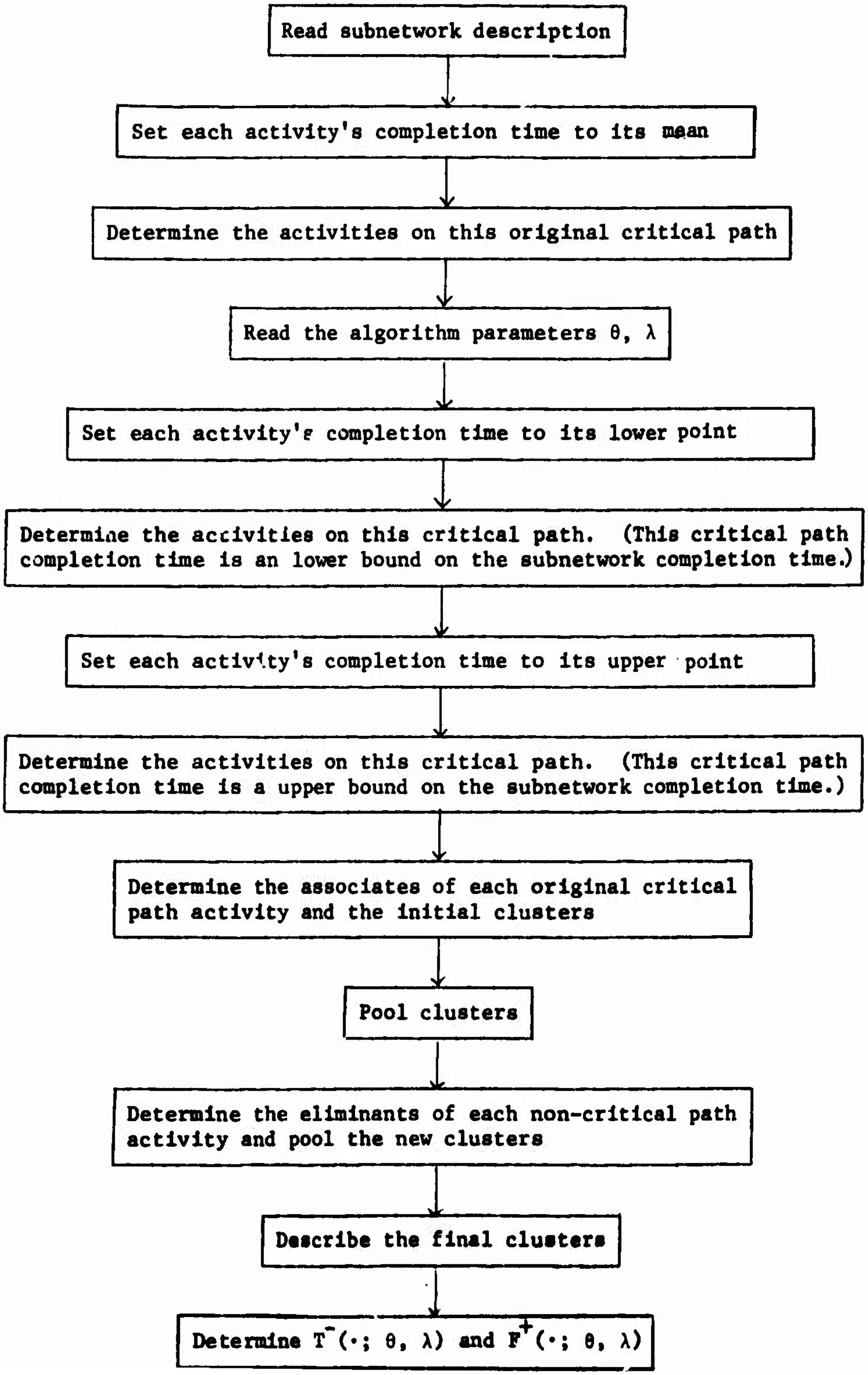
```
8505  KRAN = IHALF
      CALL BINVA(62800)
3207  NIB(IR)=NIB(IR)-1
      DO 6030 I=1,10
6030  MOMENT(IR,I) = MOMENT(IR,I)/NIB(IR)
      GO TO 3200
3208  WRITE (6,6362)
6362  FORMAT (1H1)
      IF (ICBCP.EQ.2) GO TO 6011
      DO 10000 J=1,10
      ITMAX=0
      TMAX=0.
      DO 5021 I=1,KCPB
      IF(NINCL(I).EQ.0) GO TO 5021
      IF (MOMENT(I,J).LE.TMAX) GO TO 5021
      TMAX = MOMENT(I,J)
      ITMAX=I
5021  CONTINUE
      WRITE (6,5023) J,J,MOMENT(ITMAX,J)
5023  FORMAT (1H0,5X,'A LOWER BOUND, T-(',I2,';THETA,LAMBDA), ON THE '
     *,        I2 ,'-TH MOMENT OF THE NETWORK COMPLETION TIME = ',E15.5)
10000  CONTINUE
C
C
C        BEGIN THE PROCEDURE FOR DETERMINING UPPER BOUNDS ON THE
C        NETWORK COMPLETION TIME DISTRIBUTION
C
9011  CONTINUE
      DO 9007 IR=1,KCPB
      IF(NINCL(IR).EQ.0) GO TO 9007
      DO 9990 I=2,IEDF
      II=I-1
9990  NLEFD(IR,I)=NLEFD(IR,I)+NLEFD(IR,II)
      WNNIB=DFLOAT(NIB(IR))
      DO 39990 I=1,IEDF
39990 NLEFD(IR,I)=NLEFD(IR,I)/WNNIB
9007  CONTINUE
      DO 10111 IR=1,KCPB
      IF(NINCL(IR).GT.0) GO TO 10112
10111 CONTINUE
10112 IRR=IR
C
C        IRR = NON-EMPTY CLUSTER WITH THE SMALLEST INDEX
C
      DO 10119 IR=1,KCPB
      IF(NINCL(IR).EQ.0) GO TO 10119
      DO 10110 I=1,IEDF
      IF(NLEFD(IR,I).LT.NLEFD(IRR,I)) NLEFD(IRR,I)=NLEFD(IR,I)
10110 CONTINUE
10119 CONTINUE
      WRITE (6,6264)
      WRITE(6,9423)
9423  FORMAT(1H0,5X,'AN UPPER BOUND ON THE NETWORK COMPLETION TIME DISTR
     *IBUTION: F+(.;THETA;LAMBDA)')
      DO 9421 I=1,IEDF
      NSAVE(I)=NLEFD(IRR,I)
      X=NLEFD(IRR,I)
9421  WRITE(6,9422) FD(I),THETA,LAMBDA,X
9422  FORMAT(17X,'F+(',F15.5,';',E15.5,';',F15.5,') = ',E15.5)
C
```

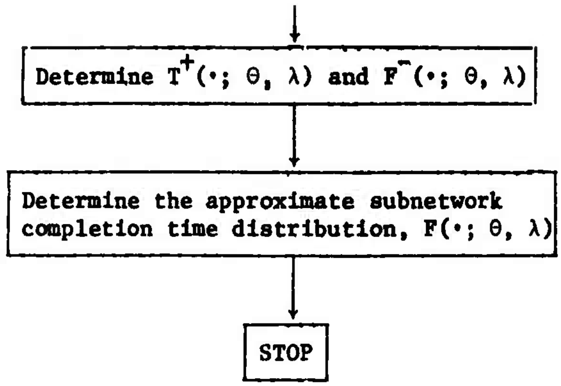
```
C         BEGIN THE PROCEDURE FOR DETERMINING UPPER BOUNDS ON THE MOMENTS
C         OF THE NETWORK COMPLETION TIME AND LOWER BOUNDS ON THE
C         DISTRIBUTION OF THE COMPLETION TIMES.
C
C
C         THE UPPER BOUNDS, T+(R,THETA,LAMBDA), ARE NOW DETERMINED.
C
C
C
C
C         FOR THE SAKE OF NUMERICAL ACCURACY THE PERT PROBLEM
C         WITH NEW ACTIVITY TIMES IS INITIALLY SOLVED FROM SCRATCH
C         INSTEAD OF UPDATING AN OLD SOLUTION.  AFTER THIS
C         REINITIALIZATION, THE REMAINING CRITICAL PATH TIMES ARE
C         DETERMINED BY UPDATING THIS SOLUTION.
C
C
      ICBCP=1
      DO 6001 I=1,M
      CTIME(I)=FHI(I)
6001  COT(I)=FHI(I)
      GO TO 6010
6011  WRITE (6,6264)
6264  FORMAT(////)
      DO 10009 J=1,10
      ITMIN=IRR
      TMIN=MOMENT(IRR,J)
      DO 7101 I=1,KCPB
      IF(NINCL(I ).EQ.0) GO TO 7101
      IF(MOMENT(I,J).GT.TMIN) GO TO 7101
      ITMIN=I
      TMIN=MOMENT(I,J)
7101  CONTINUE
10009 WRITE (6,6012) J,J,MOMENT(ITMIN,J)
6012  FORMAT (1H0,5X,'AN UPPER BOUND, T+(',I2,';THETA,LAMBDA), ON THE '
     *,I2,'-TH MOMENT OF THE NETWORK COMPLETION TIME = ',E15.5)
      DO 66900 IR=1,KCPB
      IF(NINCL(IR).EQ.0) GO TO 66900
      DO 6900 I=2,IEDF
      II=I-1
6900  NLEFD(IR,I)=NLEFD(IR,I)+NLEFD(IR,II)
      WNNIB=DFLOAT(NIB(IR))
      DO 49990 I=1,IEDF
49990 NLEFD(IR,I)=NLEFD(IR,I)/WNNIB
66900 CONTINUE
      DO 20115 IR=1,KCPB
      IF(NINCL(IR).EQ.0) GO TO 20115
      DO 20110 I=1,IEDF
      IF(NLEFD(IR,I).GT.NLEFD(IRR,I)) NLEFD(IRR,I)=NLEFD(IR,I)
20110 CONTINUE
20115 CONTINUE
      WRITE (6,6362)
      WRITE(6,6423)
6423  FORMAT(1H0,5X, 'A LOWER BOUND ON THE NETWORK COMPLETION TIME DISTR
     *IBUTION: F-(.;THETA;LAMBDA)')
      DO 6421 I=1,IEDF
      X=NLEFD(IRR,I)
6421  WRITE(6,6422) FD(I),THETA,LAMBDA,X
6422  FORMAT(17X,'F-(',F15.5,';',E15.5,';',E15.5,') = ',E15.5)
C
```

```
C         THE APPROXIMATE NETWORK COMPLETION TIME DISTRIBUTION
C
      WRITE (6,6362)
      WRITE(6,9472)
9472  FORMAT(1H0,5X,'AN APPROXIMATE NETWORK COMPLETION TIME DISTRIBUTION
     *:',//,15X,'F(.;THETA,LAMBDA) = .5 * ( F+(.;THETA,LAMBDA) + F-(.;TH
     *ETA,LAMBDA) )',//)
      DO 9471 I=1,IEDF
      X=.5D0*NSAVE(I)       +.5D0*NLEFD(IRR,I)
9471  WRITE(6,9473) FD(I),THETA,LAMBDA,X
9473  FORMAT(17X,' F(',E15.5,';',E15.5,';',E15.5,') = ',F15.5)
999   WRITE(6,850)
      STOP
      END
      SUBROUTINE BINVA(*)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON B1INV,REDCOS,CTIME,XB1,INBASE,IHEAD,ITAIL,NMMP1,NMM,N,ISTAT
      COMMON M,MP1
      DIMENSION  ISTAT(100),IHEAD(60),ITAIL(60),XB1(41)
      DIMENSION B1INV(41,41),INBASE(40),CTIME(100),REDCOS(100)
C
C        UPDATE THE FIRST ROW OF B1INV AFTER CHANGING CTIME
C
      DO 1 I=2,NMMP1
      B1INV(1,I) = 0.0
      DO 1 J=2,NMMP1
    1 B1INV(1,I) = B1INV(1,I) + B1INV(J,I)*CTIME(INBASE(J-1))
C
C        UPDATE VALUE OF THE OBJECTIVE FUNCTION
C
      XB1(1) = B1INV(1,NMMP1)
      RETURN1
      END
      SUBROUTINE BINV1 (*,TMNEW,TMOLD,ID)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON B1INV,REDCOS,CTIME,XB1,INBASE,IHEAD,ITAIL,NMMP1,NMM,N,ISTAT
      COMMON M,MP1
      DIMENSION  ISTAT(100),IHEAD(60),ITAIL(60),XB1(41)
      DIMENSION B1INV(41,41),INBASE(40),CTIME(100),REDCOS(100)
C
C        COMPUTE THE REDCOS CORRESPONDING TO ONE CHANGE IN CTIME
C
      DO 2 I=1,NMM
    2 IF(INBASE(I).EQ.ID) II=I+1
      DIFF = TMNEW-TMOLD
C        TMNEW IS THE NEW TIME AND TMOLD IS THE OLD TIME CORRESPONDING
C        TO THE SINGLE CHANGE IN CTIME
C
      DO 1 K=1,M
      IF(ISTAT(K).EQ.1) GO TO 1
      REDCOS(K) = REDCOS(K)-DIFF*(B1INV(II,IHEAD( K)+1) -
     *   B1INV(II,ITAIL( K)+1))
    1 CONTINUE
      DO 3 K=MP1,N
      IF (ISTAT(K).EQ.1) GO TO 3
      REDCOS(K) = REDCOS(K) - DIFF*B1INV(II,K-M+1)
    3 CONTINUE
C
C        UPDATE THE FIRST ROW OF B1INV AFTER CHANGING CTIME
C
```

```
      DO 10 I=2,NMMP1
      B1INV(1,I)=B1INV(1,I)+DIFF*B1INV(II,I)
   10 CONTINUE
C
C        UPDATE VALUE OF THE OBJECTIVE FUNCTION
C
      XB1(1) = B1INV(1,NMMP1)
      RETURN1
      END
```

New Subnetwork Analysis Program: Flowchart




Determine $T^+(\cdot;\ \theta,\ \lambda)$ and $F^-(\cdot;\ \theta,\ \lambda)$
Determine the approximate subnetwork completion time distribution, $F(\cdot;\ \theta,\ \lambda)$
STOP

APPENDIX C

Original Subnetwork Analysis Program

The Original Subnetwork Analysis Program is an implementation and extension of the analytical procedure described in Section 3 of Technical Report No. 48. The basic required input is

(a) an acyclic network with one source and one sink,

(b) two points from each component activity's completion time distribution, and

(c) specified values for the algorithm parameters θ and λ.

The output is mainly

(a) upper and lower bounds on the moments of the network completion time, $T_r^+(\theta, \lambda)$ and $T_r^-(\theta, \lambda)$ $r = 1, 2, \ldots, 10$;

(b) upper and lower bounds on the distribution function of the network completion time, $F^+(\cdot; \theta, \lambda)$ and $F^-(\cdot; \theta, \lambda)$; and

(c) an approximate network completion time distribution, $F(\cdot; \theta, \lambda) = 1/2[F^+(\cdot; \theta, \lambda) + F^-(\cdot; \theta, \lambda)]$.

The main extension of this program is the inclusion of an option to consider only a random sample of the 2^{n_c} activity time configurations for a cluster C instead of explicitly evaluating the critical path time for all of the 2^{n_c} activity time configurations.

The basic computational technique for determining critical path times is the Simplex Algorithm. This algorithm is applied to the dual problem. The Simplex Algorithm is used instead of the standard network analysis techniques because the Simplex Algorithm is ideally suited for the type of parametric programming required to evaluate several critical path times when only the activity times vary from one problem to the next.

A listing of the Original Subnetwork Analysis Program and a program flowchart are given at the end of this appendix.

<u>Specific Input Instructions</u>:

Card 1. Col. 1-3 : The number of activities in the network, Format (I3).

Col. 4-6 : The number of nodes in the network, Format (I3).

For each activity one card with:

Col. 11-15: The origin node of the activity, Format (I5).

Col. 21-25: The terminal node of the activity, Format (I5).

Col. 31-40: The lower point on the activity's completion time distribution, Format (F10.0)

Col. 41-50: The upper point on the activity's completion time distribution, Format (F10.0)

Next Card. Col. 1: OPTON1. OPTON1=1 implies that the program will terminate after the clusters have been formed on the basis of associates and eliminants. OPTON≠1 implies that the program will follow the normal procedure.

Next Card. Col. 1-3: IEDF. The program computes an absolute upper and lower bound for the network completion time. This range is subdivided into IEDF equal parts and the approximate distribution function (F^+, F^-, $\hat{F}$) values are printed at each of these dividing points. IEDF would usually be between 10 and 100. IEDF, Format (I3).

Next Card. Col. 1-5 : θ, Format (F5.2).

Col. 6-10: λ, Format (F5.2)

Next Card. Col. 1-10: SAMSIZ. The number of activity time configurations to be randomly selected for explicit consideration in each cluster analysis. If SAMSIZ < 0 or SAMSIZ > 2^{n_c}, all activity time configurations will be explicitly considered - no random sampling will be done. Format (I10).

The nodes should be numbered 1, 2, ..., n with the source being number 1, the sink being number n, and the other node numbers being arbitrary. The activities should be numbered 1, 2, ... in any order desired.

<u>Current Dimension Restrictions</u>:

Currently the program is dimensioned for a maximum of

60 Activities

40 Nodes

25 Clusters

25 Activities/Cluster and IEDF $\leq$ 500.

<u>Example</u>:

The program's input and output are illustrated in terms of the network in Figure C-1.

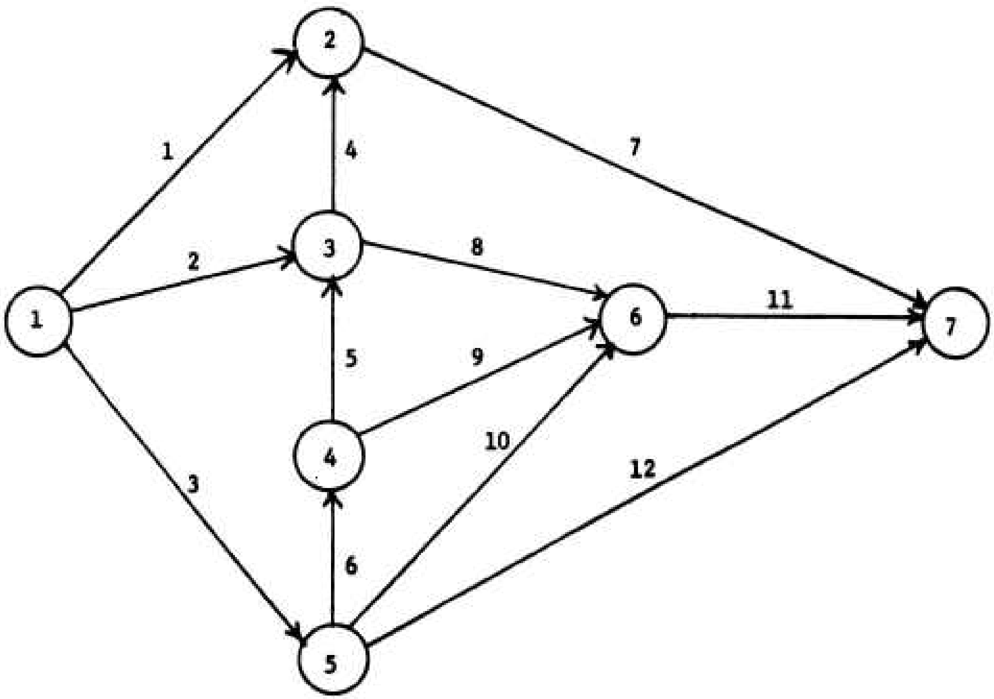


Figure C-1. Original Subnetwork Analysis Program Example Network

SAMPLE INPUT

```
012007
                  1          2       17.26      19.44
                  1          3       19.26      21.44
                  1          5       12.76      15.91
                  3          2        3.51       4.01
                  4          3        3.01       5.43
                  5          4        3.52       4.25
                  2          7       13.75      14.48
                  3          6        5.05       8.43
                  4          6        5.36       6.51
                  5          6        8.78      11.44
                  6          7       15.76      17.21
                  5          7       14.32      18.35
0
020
1.   1.
        -1
```

SAMPLE OUTPUT

INITIAL INPUT

ACTIVITY	ORIGIN	TERMINAL	LOWER PERC.	UPPER PERC.	AVERAGE	PERCENTILE DIFFERENCE
1	1	2	17.2600	19.4400	18.3500	2.1800
2	1	3	19.2600	21.4400	20.3500	2.1800
3	1	5	12.7600	15.9100	14.3350	3.1500
4	3	2	3.5100	4.0100	3.7600	0.5000
5	4	3	3.0100	5.4300	4.2200	2.4200
6	5	4	3.5200	4.2500	3.8850	0.7300
7	2	7	13.7500	14.4800	14.1150	0.7300
8	3	6	5.0500	8.4300	6.7400	3.3800
9	4	6	5.3600	6.5100	5.9350	1.1500
10	5	6	8.7800	11.4400	10.1100	2.6600
11	6	7	15.7600	17.2100	16.4850	1.4500
12	5	7	14.3200	18.3500	16.3350	4.0300

THE CRITICAL PATH TIME WHEN EACH ACTIVITY'S COMPLETION TIME IS SET EQUAL TO ITS AVERAGE IS = 0.45665D 02

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

THE 5 CRITICAL ACTIVITIES ARE AS FOLLOWS BEGINNING WITH THE TERMINAL ACTIVITY:
11. 8. 5. 6. 3.

THETA = 0.10000D 01 LAMBDA = 0.10000D 01

A LOWER BOUND ON THE EXPECTED CRITICAL PATH TIME IS = 0.40100D 02

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

THE 5 CRITICAL ACTIVITIES ARE AS FOLLOWS BEGINNING WITH THE TERMINAL ACTIVITY:
11. 8. 5. 6. 3.

A UPPER BOUND ON THE EXPECTED CRITICAL PATH TIME IS = 0.51230D 02

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

THE 5 CRITICAL ACTIVITIES ARE AS FOLLOWS BEGINNING WITH THE TERMINAL ACTIVITY:
11. 8. 5. 6. 3.

```
THE ASSOCIATES ARE NOW IDENTIFIED:

      THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   1-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY  11. IS =   0

      THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   2-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY   8. IS =   0

      THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   3-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY   5. IS =   1

         THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS
          2.

      THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   4-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY   6. IS =   0

      THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE   5-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY   3. IS =   1

         THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLOWS
          2.
```

```
THERE ARE    4 NONEMPTY CLUSTERS AFTER POOLING ON THE BASIS OF ASSOCIATES ONLY.

THE ACTIVITIES IN THE    1-TH CLUSTER ARE AS FOLLOWS:
     11.

THE ACTIVITIES IN THE    2-TH CLUSTER ARE AS FOLLOWS:
      8.

THE ACTIVITIES IN THE    3-TH CLUSTER ARE AS FOLLOWS:
      5.  2.  3.

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

THE CLUSTER TO WHICH EACH ACTIVITY BELONGS:
    (ZERO IMPLIES THAT THE ACTIVITY IS NOT IN ANY CLUSTER)
     THE    1-TH ACTIVITY IS IN THE    0-TH CLUSTER
     THE    2-TH ACTIVITY IS IN THE    3-TH CLUSTER
     THE    3-TH ACTIVITY IS IN THE    3-TH CLUSTER
     THE    4-TH ACTIVITY IS IN THE    0-TH CLUSTER
     THE    5-TH ACTIVITY IS IN THE    3-TH CLUSTER
     THE    6-TH ACTIVITY IS IN THE    4-TH CLUSTER
     THE    7-TH ACTIVITY IS IN THE    0-TH CLUSTER
     THE    8-TH ACTIVITY IS IN THE    2-TH CLUSTER
     THE    9-TH ACTIVITY IS IN THE    0-TH CLUSTER
     THE   10-TH ACTIVITY IS IN THE    0-TH CLUSTER
     THE   11-TH ACTIVITY IS IN THE    1-TH CLUSTER
     THE   12-TH ACTIVITY IS IN THE    0-TH CLUSTER

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

```
THE ELIMINANTS OF EACH NON-CRITICAL-PATH ACTIVITY ARE NOW DETERMINED:

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


THE COMPLETION TIME FOR THE    1-TH ACTIVITY HAS BEEN CHANGED TO      0.205300 02

      THERE ARE    0 ELIMINANTS CORRESPONDING TO ACTIVITY    1


THE COMPLETION TIME FOR THE    2-TH ACTIVITY HAS BEEN CHANGED TO      0.225300 02

      THERE ARE    3 ELIMINANTS CORRESPONDING TO ACTIVITY    2
          THE   1-TH ELIMINANT CORRESPONDING TO ACTIVITY    2 IS ACTIVITY    5
          THE   2-TH ELIMINANT CORRESPONDING TO ACTIVITY    2 IS ACTIVITY    6
          THE   3-TH ELIMINANT CORRESPONDING TO ACTIVITY    2 IS ACTIVITY    3


THE COMPLETION TIME FOR THE    4-TH ACTIVITY HAS BEEN CHANGED TO      0.426000 01

      THERE ARE    0 ELIMINANTS CORRESPONDING TO ACTIVITY    4


THE COMPLETION TIME FOR THE    7-TH ACTIVITY HAS BEEN CHANGED TO      0.149450 02

      THERE ARE    0 ELIMINANTS CORRESPONDING TO ACTIVITY    7


THE COMPLETION TIME FOR THE    9-TH ACTIVITY HAS BEEN CHANGED TO      0.708500 01

      THERE ARE    0 ELIMINANTS CORRESPONDING TO ACTIVITY    9


THE COMPLETION TIME FOR THE   10-TH ACTIVITY HAS BEEN CHANGED TO      0.127700 02

      THERE ARE    0 ELIMINANTS CORRESPONDING TO ACTIVITY   10


THE COMPLETION TIME FOR THE   12-TH ACTIVITY HAS BEEN CHANGED TO      0.203650 02
```

THERE ARE 0 ELIMINANTS CORRESPONDING TO ACTIVITY 12

```
THERE ARE    3 CLUSTERS.

     THERE ARE    1 ACTIVITIES IN THE    1-TH CLUSTER.   THEY ARE AS FOLLOWS:
          11.

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

     THERE ARE    1 ACTIVITIES IN THE    2-TH CLUSTER.   THEY ARE AS FOLLOWS:
          8.

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

     THERE ARE    4 ACTIVITIES IN THE    3-TH CLUSTER.   THEY ARE AS FOLLOWS:
          5.  2.  3.  6.

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


THE NUMBER OF PERCENTILE COMBINATIONS EXPLICITLY CONSIDERED IN DETERMINING THE UPPER BOUNDS AND LOWER BOUNDS
ON THE NETWORK COMPLETION TIME DISTRIBUTION AND THE UPPER BOUNDS ON ITS MOMENTS IS EQUAL TO      -1

THE INITIALIZATION PARAMETER FOR THE SAMPLING IS IY =          77
```

THE FOLLOWING TABLE WAS COMPUTED CONSIDERING ALL PERCENTILE COMBINATIONS.

A LOWER BOUND, T-(1;THETA,LAMBDA), ON THE 1-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.45888D 02

A LOWER BOUND, T-(2;THETA,LAMBDA), ON THE 2-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.21099D 04

A LOWER BOUND, T-(3;THETA,LAMBDA), ON THE 3-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.97064D 05

A LOWER BOUND, T-(4;THETA,LAMBDA), ON THE 4-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.44742D 07

A LOWER BOUND, T-(5;THETA,LAMBDA), ON THE 5-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.20655D 09

A LOWER BOUND, T-(6;THETA,LAMBDA), ON THE 6-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.95494D 10

A LOWER BOUND, T-(7;THETA,LAMBDA), ON THE 7-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.44215D 12

A LOWER BOUND, T-(8;THETA,LAMBDA), ON THE 8-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.20503D 14

A LOWER BOUND, T-(9;THETA,LAMBDA), ON THE 9-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.95211D 15

A LOWER BOUND, T-(10;THETA,LAMBDA), ON THE 10-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.44279D 17

THE FOLLOWING TABLE WAS COMPUTED CONSIDERING ALL PERCENTILE COMBINATIONS.

AN UPPER BOUND, T+(1;THETA,LAMBDA), ON THE 1-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.45888D 02

AN UPPER BOUND, T+(2;THETA,LAMBDA), ON THE 2-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.21123D 04

AN UPPER BOUND, T+(3;THETA,LAMBDA), ON THE 3-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.97530D 05

AN UPPER BOUND, T+(4;THETA,LAMBDA), ON THE 4-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.45170D 07

AN UPPER BOUND, T+(5;THETA,LAMBDA), ON THE 5-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.20983D 09

AN UPPER BOUND, T+(6;THETA,LAMBDA), ON THE 6-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.97769D 10

AN UPPER BOUND, T+(7;THETA,LAMBDA), ON THE 7-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.45688D 12

AN UPPER BOUND, T+(8;THETA,LAMBDA), ON THE 8-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.21412D 14

AN UPPER BOUND, T+(9;THETA,LAMBDA), ON THE 9-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.10064D 16

AN UPPER BOUND, T+(10;THETA,LAMBDA), ON THE 10-TH MOMENT OF THE NETWORK COMPLETION TIME = 0.47434D 17

THE FOLLOWING TABLE WAS COMPUTED CONSIDERING ALL PERCENTILE COMBINATIONS.

A LOWER BOUND ON THE NETWORK COMPLETION TIME DISTRIBUTION: F-(.;THETA;LAMBDA)

F-(	0.40657D 02;	0.10000D 01;	0.10000D 01)	=	0.15625D-01
F-(	0.41213D 02;	0.10000D 01;	0.10000D 01)	=	0.31250D-01
F-(	0.41770D 02;	0.10000D 01;	0.10000D 01)	=	0.46875D-01
F-(	0.42326D 02;	0.10000D 01;	0.10000D 01)	=	0.93750D-01
F-(	0.42883D 02;	0.10000D 01;	0.10000D 01)	=	0.12500D 00
F-(	0.43439D 02;	0.10000D 01;	0.10000D 01)	=	0.18750D 00
F-(	0.43996D 02;	0.10000D 01;	0.10000D 01)	=	0.29688D 00
F-(	0.44552D 02;	0.10000D 01;	0.10000D 01)	=	0.31250D 00
F-(	0.45109D 02;	0.10000D 01;	0.10000D 01)	=	0.39063D 00
F-(	0.45665D 02;	0.10000D 01;	0.10000D 01)	=	0.46875D 00
F-(	0.46221D 02;	0.10000D 01;	0.10000D 01)	=	0.53125D 00
F-(	0.46778D 02;	0.10000D 01;	0.10000D 01)	=	0.62500D 00
F-(	0.47334D 02;	0.10000D 01;	0.10000D 01)	=	0.68750D 00
F-(	0.47891D 02;	0.10000D 01;	0.10000D 01)	=	0.78125D 00
F-(	0.48447D 02;	0.10000D 01;	0.10000D 01)	=	0.84375D 00
F-(	0.49004D 02;	0.10000D 01;	0.10000D 01)	=	0.87500D 00
F-(	0.49560D 02;	0.10000D 01;	0.10000D 01)	=	0.90625D 00
F-(	0.50117D 02;	0.10000D 01;	0.10000D 01)	=	0.93750D 00
F-(	0.50673D 02;	0.10000D 01;	0.10000D 01)	=	0.96875D 00
F-(	0.51230D 02;	0.10000D 01;	0.10000D 01)	=	0.10000D 01

THE FOLLOWING TABLE WAS COMPUTED CONSIDERING ALL PERCENTILE COMBINATIONS.

AN UPPER BOUND ON THE NETWORK COMPLETION TIME DISTRIBUTION: F+(.;THETA;LAMBDA)

F+(	0.40657D 02;	0.10000D 01;	0.10000D 01)	=	0.15625D-01
F+(	0.41213D 02;	0.10000D 01;	0.10000D 01)	=	0.31250D-01
F+(	0.41770D 02;	0.10000D 01;	0.10000D 01)	=	0.46875D-01
F+(	0.42326D 02;	0.10000D 01;	0.10000D 01)	=	0.93750D-01
F+(	0.42883D 02;	0.10000D 01;	0.10000D 01)	=	0.12500D 00
F+(	0.43439D 02;	0.10000D 01;	0.10000D 01)	=	0.18750D 00
F+(	0.43996D 02;	0.10000D 01;	0.10000D 01)	=	0.29688D 00
F+(	0.44552D 02;	0.10000D 01;	0.10000D 01)	=	0.31250D 00
F+(	0.45109D 02;	0.10000D 01;	0.10000D 01)	=	0.39063D 00
F+(	0.45665D 02;	0.10000D 01;	0.10000D 01)	=	0.46875D 00
F+(	0.46221D 02;	0.10000D 01;	0.10000D 01)	=	0.53125D 00
F+(	0.46778D 02;	0.10000D 01;	0.10000D 01)	=	0.62500D 00
F+(	0.47334D 02;	0.10000D 01;	0.10000D 01)	=	0.68750D 00
F+(	0.47891D 02;	0.10000D 01;	0.10000D 01)	=	0.78125D 00
F+(	0.48447D 02;	0.10000D 01;	0.10000D 01)	=	0.84375D 00
F+(	0.49004D 02;	0.10000D 01;	0.10000D 01)	=	0.87500D 00
F+(	0.49560D 02;	0.10000D 01;	0.10000D 01)	=	0.90625D 00
F+(	0.50117D 02;	0.10000D 01;	0.10000D 01)	=	0.93750D 00
F+(	0.50673D 02;	0.10000D 01;	0.10000D 01)	=	0.96875D 00
F+(	0.51230D 02;	0.10000D 01;	0.10000D 01)	=	0.10000D 01

AN APPROXIMATE NETWORK COMPLETION TIME DISTRIBUTION:

F(.;THETA,LAMBDA) = .5 * (F+(.;THETA,LAMBDA) + F-(.;THETA,LAMBDA))

F(	0.40657D 02;	0.10000D 01;	0.10000D 01) =	0.15625D-01
F(	0.41213D 02;	0.10000D 01;	0.10000D 01) =	0.31250D-01
F(	0.41770D 02;	0.10000D 01;	0.10000D 01) =	0.46875D-01
F(	0.42326D 02;	0.10000D 01;	0.10000D 01) =	0.93750D-01
F(	0.42883D 02;	0.10000D 01;	0.10000D 01) =	0.12500D 00
F(	0.43439D 02;	0.10000D 01;	0.10000D 01) =	0.18750D 00
F(	0.43996D 02;	0.10000D 01;	0.10000D 01) =	0.29688D 00
F(	0.44552D 02;	0.10000D 01;	0.10000D 01) =	0.31250D 00
F(	0.45109D 02;	0.10000D 01;	0.10000D 01) =	0.39063D 00
F(	0.45665D 02;	0.10000D 01;	0.10000D 01) =	0.46875D 00
F(	0.46221D 02;	0.10000D 01;	0.10000D 01) =	0.53125D 00
F(	0.46778D 02;	0.10000D 01;	0.10000D 01) =	0.62500D 00
F(	0.47334D 02;	0.10000D 01;	0.10000D 01) =	0.68750D 00
F(	0.47891D 02;	0.10000D 01;	0.10000D 01) =	0.78125D 00
F(	0.48447D 02;	0.10000D 01;	0.10000D 01) =	0.84375D 00
F(	0.49004D 02;	0.10000D 01;	0.10000D 01) =	0.87500D 00
F(	0.49560D 02;	0.10000D 01;	0.10000D 01) =	0.90625D 00
F(	0.50117D 02;	0.10000D 01;	0.10000D 01) =	0.93750D 00
F(	0.50673D 02;	0.10000D 01;	0.10000D 01) =	0.96875D 00
F(	0.51230D 02;	0.10000D 01;	0.10000D 01) =	0.10000D 01

PROGRAM LISTING

```
C
C       ORIGINAL SUBNETWORK ANALYSIS PROGRAM
C
      IMPLICIT REAL*8 (A-H,O-Z)
C        FOR THE SAKE OF  IDENTIFYING THE APPROPRIATE DIMENSIONS, LET
C              M = THE NUMBER OF ACTIVITIES IN THE NETWORK
C              NMM = NUMBER OF NODES IN THE NETWORK
C              NMMP1  = NMM + 1
C              N = M + NMM
C              L = THE LENGTH OF THE CRITICAL PATH
C              C = THE MAXIMUM NUMBER OF BRANCHES IN A CLUSTER
C              IEDF = THE NUMBER OF DIVISIONS IN THE EMPIRICAL
C                      DISTRIBUTION FUNCTION
C
C
C     INTEGER TAIL( M),HEAD( M),ASSGRP( L,L ),CLINCL(L,L),EGRP(L)
C     DIMENSION NINCL( C),INCLUS( L, C),NCLINC( L)
C     DIMENSION NLEFD(IEDF),FD( IEDF),NSAVE( IEDF)
C     DIMENSION IZZ( C),AVG( L),THAT(L)
C     DIMENSION INBASE(NMM),XNODE(NMM)
C     DIMENSION LEFT(M ),LEFTO(M ),NONCP(M)
C     DIMENSION XB1(NMMP1),Y1(NMMP1),REDCOS(N),ISTAT(N)
C     DIMENSION ICRITP(L),NINAG(M),ICRITN(L+1),CTIME(N),COT(N)
C     DIMENSION KB(L),IBB(L),F25(M),F75(M),SIGMA(M),B1INV(NMMP1,NMMP1)
C     REAL          MOMENT(L,10)
C
C        OF COURSE THESE DIMENSIONS ARE MERELY UPPER BOUNDS
C
C
      COMMON B1INV,REDCOS,CTIME,XB1,INBASE,HEAD,TAIL,NMMP1,NMM,N,ISTAT
      COMMON M,MP1
      INTEGER TAIL(60),HEAD(60),ASSGRP(25,25),CLINCL( 25,25),EGRP(25)
      INTEGER  SAMSIZ,RANSAM
      DIMENSION NINCL(25),INCLUS(25,25),NCLINC( 25)
      DIMENSION FD(500),NLEFD(500),NSAVE(500)
      DIMENSION AVG(25),THAT( 25)
      DIMENSION INBASE(40)
      DIMENSION XNODE(40)
      DIMENSION XB1(41),Y1(41),REDCOS(100),ISTAT(100)
      DIMENSION B1INV(41,41),KB(25),IBB(25),FLO(60),FHI(60),SIGMA(60)
      DIMENSION ICRITP(25),NINAG(50),ICRITN(26),COT(100),CTIME(100)
      DIMENSION LEFT(60),LEFTO(60),NONCP(60)
      REAL*8 LAMBDA,MOMENT(25,10)
C        M = THE NUMBER OF ACTIVITIES IN THE NETWORK
C        NMM = THE NUMBER OF NODES IN THE PERT NETWORK
      READ(5,100)  M,NMM
100   FORMAT(2I3)
      N=NMM+M
      MP1=M+1
      NMMP1=NMM+1
C
C        THE ACTIVITIES ARE DESCRIBED IN TERMS OF THEIR NODES
C        II=THE TAIL NODE, THE ORIGIN NODE
C        JJ=THE HEAD NODE, THE TERMINAL NODE
C        FLO  =  THE LOWER PERCENTILE
C        FHI  =  THE UPPER PERCENTILE
C        SIGMA  =  FLO - FHI
C
      DO 610  I=1,M
```

```
      READ(5,2501) II,JJ,FLO(I ),FHI(I )
      SIGMA(I) = FHI(I)-FLO(I)
 2501 FORMAT(10X,I5,5X,I5, 5X,F10.0,F10.0)
      COT(I) = (FLO(I)+FHI(I))/2.
      CTIME(I) = COT(I)
      TAIL(I) = II
  610 HEAD(I)=JJ
C         COT(M)      = THE ORIGINAL RIGHT HAND SIDES, I.E. THE AVERAGE
C         OF FLO AND FHI
      DO 55610 I=MP1,N
55610 CTIME(I) = 0.
C
C                OPTON1 =1 IMPLIES THAT THE PROGRAM WILL TERMINATE AFTER
C                          THE CLUSTERS HAVE BEEN FORMED.  NO BOUNDS ON
C                          THE PROJECT COMPLETION TIME MOMENTS OR
C                          DISTRIBUTION WILL BE DETERMINED.
C                OPTON1  NOT= 1  IMPLIES THAT THE NORMAL PROCEDURE WILL BE
C                          FOLLOWED.
C
      READ(5,77551) OPTON1
77551 FORMAT(10I1)
      IF(OPTON1 .EQ.1) WRITE(6,77552)
77552 FORMAT(1H1,10X,'OPTION1=1 AND THE PROGRAM WILL TERMINATE AFTER THE
     * CLUSTERS HAVE BEEN FORMED.',/,11X,'NO BOUNDS ON THE PROJECT COMPL
     *ETION TIME MOMENTS OR DISTRIBUTION WILL BE DETERMINED.')
      READ(5,100) IEDF
      WRITE(6,2700)
 2700 FORMAT(1H1,15X,'INITIAL INPUT')
      WRITE(6,2701)
 2701 FORMAT(1H0,10X,'ACTIVITY  ORIGIN    TERMINAL  LOWER PERC.  UPPER P
     *ERC.    AVERAGE  PERCENTILE DIFFERENCE')
      DO 2704 I=1,M
 2704 WRITE(6,2702) I,TAIL(I),HEAD(I),FLO(I),FHI(I),CTIME(I),SIGMA(I)
 2702 FORMAT(1H ,13X,I3,5X,I3,7X,I3,5X,F10.4,3X,F10.4,3X,F10.4,4X,F10.4)
C
C         THE FOLLOWING INDICATORS ARE USED:
C
C
C         IPARM = 1  IMPLIES THE CRITICAL PATH TIME WHEN ALL ACTIVITY
C                    COMPLETION TIMES ARE SET EQUAL TO THEIR AVERAGES IS
C                    BEING DETERMINED
C         IPARM = 2  IMPLIES THAT THE LOWER BOUND ON THE COMPLETION TIME
C                     FOR THE SUBNETWORK IS BEING DETERMINED
C         IPARM = 3  IMPLIES THAT THE UPPER BOUND ON THE COMPLETION TIME
C                     FOR THE SUBNETWORK IS BEING DETERMINED
C         IPARM > 3  WHEN INDEXL=0 IMPLIES THAT THE ASSOCIATES ARE
C                    BEING DETERMINED
C
C
C         INDEXL=0  IMPLIES THAT INITIAL CLUSTERS ARE STILL BEING FORMED
C         INDEXL=1  IMPLIES THAT THE LEFTOVERS, THEIR ELIMINANTS, AND
C                         POOLED CLUSTERS ARE BEING DETERMINED
C         INDEXL=2  IMPLIES THAT THE 2**NINCL( ) RUNS FOR EACH CLUSTER
C                   ARE BEING MADE AND AVERAGED
C
C
C         ICBCP = 0 IMPLIES THAT THE PROCEDURE FOR DETERMINING UPPER
C                   BOUNDS ON THE MOMENTS OF THE NETWORK COMPLETION TIME
C                   AND LOWER BOUNDS ON THE DISTRIBUTION OF COMPLETION
C                   TIMES HAS NOT BEEN BEGUN
```

```
C          ICBCP = 1 IMPLIES THAT THE PROCEDURE FOR DETERMINING UPPER
C                    BOUNDS ON THE MOMENTS OF THE NETWORK COMPLETION TIME
C                    AND LOWER BOUNDS ON THE DISTRIBUTION OF COMPLETION
C                    TIMES IS BEING INITIALIZED
C          ICBCP = 2 IMPLIES THAT THE PROCEDURE FOR DETERMINING UPPER
C                    BOUNDS ON THE MOMENTS OF THE NETWORK COMPLETION TIME
C                    AND LOWER BOUNDS ON THE DISTRIBUTION OF COMPLETION
C                    TIMES IS BEING CARRIED OUT
C
C
C          IDLB = 0   IMPLIES THAT THE PROCEDURE FOR DETERMINING A UPPER
C               BOUND ON THE NETWORK COMPLETION TIME DISTRIBUTION HAS NOT
C               BEGUN
C          IDLB = 1   IMPLIES THAT THE UPPER BOUND ON THE NETWORK
C               COMPLETION TIME DISTRIBUTION IS BEING DETERMINED
C
      IPARM=1
      INDEXL=0
      ICBCP=0
      IDLB = 0
 6010 CONTINUE
      DO 104 I=1,NMM
  104 INBASE(I)=M+1
      DO 2001 J=1,M
 2001 ISTAT(J)=0.
      DO 2002 J=MP1,N
 2002 ISTAT(J)=1
      DO 10 II=1,NMMP1
      DO 12 L=1,NMMP1
   12 B1INV(L,II) = 0.
   10 B1INV(II,II) = 1.
      DO 30 I=1,NMM
   30 XB1(I) = 0.
      XB1(NMMP1) = 1.
      TOLR1=1.0D-10
C
C         START THE SIMPLEX ALGORITHM
C         SOLVE THE DUAL PROBLEM
C          THE NUMBER OF VARIABLES IS M REAL + NMM SLACKS
C          FOR A TOTAL OF N VARIABLES
C
  350 CONTINUE
 2800 DO 23 J=1,N
      RATS = 0.
      IF (ISTAT(J).EQ.1) GO TO 52800
      IF (J.GT.M) GO TO 22
      RATS =-B1INV(1,HEAD(J)+1)+B1INV(1,TAIL(J)+1) + CTIME(J)
      GO TO 52800
   22 RATS =-B1INV(1,J-M+1)
52800 REDCOS(J)= RATS
   23 CONTINUE
22800 CONTINUE
      IRMAX=1
      RMAX=REDCOS(1)
      DO 24 J=2,N
      IF(REDCOS(J) .LE. RMAX) GO TO 24
      RMAX=REDCOS(J)
      IRMAX=J
   24 CONTINUE
      IF(RMAX .LE. TOLR1) GO TO 401
```

```
22824 CONTINUE
      DO 26 L=1,NMMP1
      IF (IRMAX.GT.M) GO TO 50026
      Y1(L) =-B1INV(L,TAIL(IRMAX)+1)+B1INV(L,HEAD(IRMAX)+1)
      GO TO 26
50026 Y1(L) = B1INV(L,IRMAX-M+1)
   26 CONTINUE
      Y1(1) = Y1(1) - CTIME(IRMAX)
      NUMBER=0
      DO 27 L=2,NMMP1
27    IF(Y1(L) .LE. TOLR1)    NUMBER=NUMBER+1
      IF(NUMBER .EQ. NMM) GO TO 403
      RMIN=.99D 20
      IRMIN=0.
      DO 32 II=2,NMMP1
      IF(Y1(II).LE. TOLR1)  GO TO 32
      RATS =XB1(II)/Y1(II)
      RR=RATS-RMIN
      IF(RR .GE. 0.00) GO TO 32
      RMIN=RATS
      IRMIN=II
32    CONTINUE
      DO 33 J=2,NMMP1
      WW=B1INV(IRMIN ,J)/Y1(IRMIN )
      DO 37 L=1,NMMP1
37    B1INV(L,J)=B1INV(L,J)-WW*Y1(L)
33    B1INV(IRMIN ,J)=WW
C
C        UPDATE THE BASIC VARIABLES: INBASE AND XB1
C
      ISTAT(INBASE(IRMIN-1))=0
      ISTAT(IRMAX)=1
      INBASE(IRMIN-1)=IRMAX
      W=XB1(IRMIN )/Y1(IRMIN )
      DO 38 I=1,NMMP1
38    XB1(I)=XB1(I)-Y1(I)*W
      XB1(IRMIN )=W
      GO TO 350
403   WRITE(6,530)
530   FORMAT(1H0,5X,'NO FEASIBLE SOLUTION EXISTS.  CHECK YOUR INPUT DATA
     *.')
      WRITE(6,850)
850   FORMAT(1H1)
      GO TO 999
C
C        END OF THE SIMPLEX ALGORITHM
C
401   CONTINUE
C
C        KKK= THE NUMBER OF NODES ON THE CRITICAL PATH
C        KB(L)= THE L-TH NODE IN THE CRITICAL PATH, COUNTING BACKWARDS
C                   FROM THE TERMINAL NODE
C        KKB= THE NUMBER OF ACTIVITIES ON THE CRITICAL PATH
C        IBB(L)= THE L-TH ACTIVITY ON THE CRITICAL PATH, COUNTING
C                   BACKWARDS FROM THE TERMINAL NODE
C
      IF(ICBCP.EQ.1) GO TO 6008
      IF(INDEXL.EQ.2) GO TO 3204
C
C        INBASE IS A SET OF M INTEGER VARIABLES WHICH INDICATE THE
```

```
C        COMPOSITION OF THE CURRENT BASIS.  FOR EXAMPLE,
C             INBASE(K) = 7  IMPLIES THAT THE K-TH COLUMN IN THE BASIS B
C                            CORRESPONDS TO THE 7-TH VARIABLE
C
C
C        ISTAT INDICATES THE BASIC STATUS OF EACH VARIABLE
C             ISTAT(K) = 1  IMPLIES THAT THE K-TH VARIABLE IS IN THE
C                           DUAL BASIS
C             ISTAT(K) = 0  IMPLIES THAT THE K-TH VARIABLE IS NOT IN THE
C                           DUAL BASIS
C
C
C        THE FOLLOWING STATEMENTS DETERMINE THE NODES AND ACTIVITIES ON
C        THE CRITICAL PATH
C
C
C        THE DUAL SOLUTION IMPLIES THE FOLLOWING OPTIMAL SOLUTION TO THE
C        PRIMAL PERT PROBLEM. HOWEVER SOME OF THE NODE TIMES(OTHER THAN
C        THE LAST ONE) MAY BE HIGHER THAN NECESSARY.  THUS IN
C        DETERMINING THE CRITICAL PATH AN ALTERNATIVE OPTIMAL SOLUTION
C        MAY HAVE TO BE IDENTIFIED.
C        B1INV IS NOT CHANGED.
C
      DO 83002 I=1,NMM
83002 XNODE(I)=B1INV(1,I+1)
      KKK=1
      KB(1)=NMM
83001 IK=KB(KKK)
C
C        DETERMINE WHETHER THE TIME TO REACH NODE   IK   IS NECESSARILY
C        AS LARGE AS INDICATED FROM THE DUAL SOLUTION
C
      SMIN=999999.
      ISMIN=0
      DO 83000 I=1,M
      IF(HEAD(I).NE.IK) GO TO 83000
      SLACK=XNODE(HEAD(I))-XNODE(TAIL(I))-CTIME(I)
      IF(SLACK.GE.SMIN) GO TO 83000
      SMIN=SLACK
      ISMIN=I
83000 CONTINUE
      IF(SMIN.LT.0.0001) GO TO 83003
C
C        THE TIME FOR NODE IK WAS UNNECCESSARILY LARGE
C
      XNODE(IK)=XNODE(IK)-SMIN
      KKK=KKK-1
      GO TO 83001
83003 IBB(KKK)=ISMIN
      KKK=KKK+1
      KB(KKK)=TAIL(ISMIN)
      IF(TAIL(ISMIN).GT.1) GO TO 83001
      KKB=KKK-1
      IF(INDEXL.EQ.1) GO TO 3121
      IPARM=IPARM+1
      IF(IPARM.GT.4) GO TO 2910
      IF(IPARM.EQ.3) GO TO 6400
      IF(IPARM.EQ.4) GO TO 6401
C
C        ICRITP(L)= THE L-TH ACTIVITY ON THE ORIGINAL CRITICAL PATH
```

```
C         KCPB= THE NUMBER OF ACTIVITIES ON THE ORIGINAL CRITICAL PATH
C
      TOTAL = B1INV(1,NMMP1)
      KCPB=KKB
      ICRITN(1) = NMM
      DO 2802 I=1,KCPB
      ICRITN(I+1) = KB(I+1)
2802  ICRITP(I)=IBB(I)
      X=TOTAL
      WRITE(6,851) X
851   FORMAT(1H0,5X,'THE CRITICAL PATH TIME WHEN EACH ACTIVITY''S COMPLE
     *TION TIME IS SET EQUAL TO ITS AVERAGE IS = ',E15.5)
      WRITE(6,7606) KKK
 7606 FORMAT(1H0,10X,'THE ',I3,' NODES ON THE CRITICAL PATH ARE AS FOLLO
     *WS BEGINNING WITH THE TERMINAL NODE:')
      WRITE(6,7707) (KB(I),I=1,KKK)
 7707 FORMAT(15X,20(I3,','))
      WRITE(6,7710) KKB
7710  FORMAT(1H0,10X,'THE ',I3,' CRITICAL ACTIVITIES ARE AS FOLLOWS BEGI
     *NNING WITH THE TERMINAL ACTIVITY:')
      WRITE(6,7707) (IBB(I),I=1,KKB)
      READ(5,2920) THETA,LAMBDA
2920   FORMAT(2F5.2)
      WRITE(6,3071) THETA,LAMBDA
3071  FORMAT(1H0,10X,'THETA   = ',E15.5,'    LAMBDA = ',E15.5)
C
C        SAMSIZ = THE NUMBER OF ACTIVITY TIME CONFIGURATIONS TO BE
C                 RANDOMLY SELECTED FOR CONSIDERATION IN EACH CLUSTER
C
C     NOTE:  SINCE THIS IS A RANDOM SAMPLE , SOME PERCENTILE
C            COMBINATIONS MAY BE CONSIDERED MORE THAN ONCE.
C
      READ (5,3209) SAMSIZ
 3209 FORMAT (I10)
C
C        THE COMPLETION TIME FOR ALL ACTIVITIES IS SET TO THEIR LOWER
C        PERCENTILE.  THE RESULTING CRITICAL PATH TIME IS A LOWER
C        BOUND ON THE EXPECTED CRITICAL PATH TIME.
C
      DO 6402 I=1,M
 6402 CTIME(I) = FLO(I)
      CALL BINVA(62800)
 6400 CPLB= B1INV(1,NMMP1)
      WRITE(6,6405) CPLB
6405  FORMAT(1H0,5X,'A LOWER BOUND ON THE EXPECTED CRITICAL PATH TIME IS
     * = ',F15.5)
      WRITE(6,7606) KKK
      WRITE(6,7707) (KB(I),I=1,KKK)
      WRITE(6,7710) KKB
      WRITE(6,7707) (IBB(I),I=1,KKB)
C
C        THE COMPLETION TIME FOR ALL ACTIVITIES IS SET TO THEIR UPPER
C        PERCENTILE.  THE RESULTING CRITICAL PATH TIME IS A UPPER
C        BOUND ON THE EXPECTED CRITICAL PATH TIME.
C
      DO 6406 I=1,M
 6406 CTIME(I) = FHI(I)
      CALL BINVA(62800)
 6401 CPUB= B1INV(1,NMMP1)
      WRITE(6,6409) CPUB
```

```
 6409  FORMAT(1H0,5X,'A UPPER BOUND ON THE EXPECTED CRITICAL PATH TIME IS
      * = ',F15.5)
       WRITE(6,7606) KKK
       WRITE(6,7707) (KB(I),I=1,KKK)
       WRITE(6,7710) KKB
       WRITE(6,7707) (IBB(I),I=1,KKB)
 C
 C        FD(I) = THE LOWER BOUND ON THE EXPECTED CRITICAL PATH TIME
 C                PLUS 1/IEDF OF THE DISTANCE TO THE UPPER BOUND
 C        NLEFD(I) = THE OBSERVED NUMBER OF CRITICAL PATH TIMES
 C                   THAT ARE < OR= FD(I)
 C        FD AND NLEFD ARE USED TO BUILD AN 'EMPIRICAL' DISTRIBUTION OF
 C        THE CRITICAL PATH TIMES
 C
       C=(CPUB-CPLB)/IEDF
       DO 6412 K=1,KCPB
       DO 6412 I=1,IEDF
       FD(I)=CPLB+I*C
 6412  NLEFD(I) = 0
 C
 C        THE ASSOCIATE GROUPS ARE NOW FORMED
 C
       WRITE(6,3165)
 3165  FORMAT(1H1,5X,'THE ASSOCIATES ARE NOW IDENTIFIED:')
       IIIII=1
       DO 2825 I=1,M
 2825  CTIME(I)=COT(I)
       IWWWQ=ICRITP(1)
       CHANG= LAMBDA*SIGMA(IWWWQ)
       TEX=COT(IWWWQ)-CHANG
       IF(TEX.LT.0.0) CHANG=COT(IWWWQ)
       CTIME(IWWWQ)=COT(IWWWQ)-CHANG
       CALL BINVA(&2800)
 2801  CONTINUE
       IF(ISTAT(IWWWQ).EQ.1) CALL BINV1(&22825,COT(IWWWQ),CTIME(IWWWQ),
      *IWWWQ)
       REDCOS(IWWWQ) = REDCOS(IWWWQ)+COT(IWWWQ)-CTIME(IWWWQ)
 22825 CTIME(IWWWQ)=COT(IWWWQ)
       IWWWQ=ICRITP(IIIII)
       CHANG= LAMBDA*SIGMA(IWWWQ)
       TEX=COT(IWWWQ)-CHANG
       IF(TEX.LT.0.0) CHANG=COT(IWWWQ)
       CTIME(IWWWQ)=COT(IWWWQ)-CHANG
       IF (ISTAT(IWWWQ).EQ.1) CALL BINV1(&22800,CTIME(IWWWQ),COT(IWWWQ),
      *IWWWQ)
       REDCOS(IWWWQ)=REDCOS(IWWWQ)-COT(IWWWQ)+CTIME(IWWWQ)
       GO TO 22800
 C
 C        DETERMINE ASSOCIATE GROUP
 C
 2910  NINAG(IIIII)=0
       DO 2911 K=1,KKB
       KK=1
 2913  IF(IBB(K).EQ.ICRITP(KK)) GO TO 2911
       IF(KK.GE.KCPB) GO TO 2912
       KK=KK+1
       GO TO 2913
 2912  NINAG(IIIII)=NINAG(IIIII)+1
       ASSGRP(IIIII,NINAG(IIIII))=IBB(K)
 2911  CONTINUE
```

```
      WRITE(6,2915) IIIII,ICRITP(IIIII),NINAG(IIIII)
 2915 FORMAT(1H0,10X,'THE NUMBER OF ASSOCIATES ASSOCIATED WITH THE ',I3,
     *'-TH CRITICAL PATH ACTIVITY, I.E. ACTIVITY ',I3,', IS = ',I3)
      IDUCK=NINAG(IIIII)
      IF(IDUCK.EQ.0) GO TO 2810
      WRITE(6,2916) (ASSGRP(IIIII,I),I=1,IDUCK)
 2916 FORMAT(1H0,15X,'THE ACTIVITIES IN THE ASSOCIATE GROUP ARE AS FOLLO
     *WS',/,15X,50(I3,','))
 2810 IIIII=IIIII+1
      IF(IIIII.LE.KCPB) GO TO 2801
C
C        DETERMINE THE CLUSTERS
C
C                THE CLUSTERS ARE POOLED TOWARD THE TERMINAL NODE
C                NCLUS = THE NUMBER OF NON-EMPTY CLUSTERS
C                NINCL(I) = THE NUMBER OF ACTIVITIES IN THE I-TH CLUSTER
C                INCLUS(I,J) = THE J-TH ACTIVITY IN THE I-TH CLUSTER
C                NCLINC(I) = THE NUMBER OF CLUSTERS COMPRISING THE I-TH
C                    CLUSTER AFTER POOLING
C                CLINCL(I,J) = THE J-TH CLUSTER WHICH HAS BEEN POOLED INTO
C                    THE I-TH CLUSTER
C
C                NCLINC AND CLINCL HELP KEEP TRACK OF WHICH CLUSTER THE
C                CRITICAL PATH ACTIVITIES ARE IN
C
C
C        BELOW FORMS CLUSTERS BY PUTTING EACH CRITICAL PATH ACTIVITY IN
C        SEPARATE CLUSTER AND THEN ADDING EACH CRITICAL PATH ACTIVITY'S
C        ASSOCIATES TO ITS CLUSTER
C
      NCLUS=KCPB
      DO 3020 I=1,KCPB
      NCLINC(I)=1
      CLINCL(I,1)=I
      NINCL(I)=NINAG(I)+1
      INCLUS(I,1)=ICRITP(I)
      IF(NINAG(I).EQ.0) GO TO 3020
      IDUCK=NINCL(I)
      DO 3021 J=2,IDUCK
      JJ=J-1
 3021 INCLUS(I,J)=ASSGRP(I,JJ)
 3020 CONTINUE
C
C        BELOW POOLS CLUSTERS FORMED FROM ASSOCIATES
C
      IA=0
 3031 IA=IA+1
      IF(IA.GE.KCPB) GO TO 3030
      IF(NCLUS.EQ.1) GO TO 3030
      IDIA=NINCL(IA)
      IF(IDIA.EQ.0) GO TO 3031
      IAA=IA+1
      DO 3023 II=IAA,KCPB
      IDII=NINCL(II)
      IF(IDII.EQ.0) GO TO 3023
      DO 3025 I=1,IDIA
      DO 3025 J=1,IDII
      IF(INCLUS(II,J).EQ.INCLUS(IA,I)) GO TO 3027
 3025 CONTINUE
      GO TO 3023
```

```
3027  NCLUS=NCLUS-1
      DO 3028 J=1,IDII
      DO 3029 I=1,IDIA
      IF(INCLUS(II,J).EQ.INCLUS(IA,I)) GO TO 3028
3029  CONTINUE
      NINCL(IA)=NINCL(IA)+1
      INCLUS(IA,NINCL(IA) )=INCLUS(II,J)
3028  CONTINUE
      NINCL(II)=0
      NCLINC(IA)=NCLINC(IA)+1
      CLINCL(IA,NCLINC(IA)) = II
      NCLINC(II)=0
3023  CONTINUE
      GO TO 3031
3030  CONTINUE
C
C        BELOW DESCRIBES CLUSTERS AFTER POOLING BASED ON THE ASSOCIATES
C
      WRITE(6,3033) NCLUS
3033  FORMAT(1H1,10X,'THERE ARE ',I3,' NONEMPTY CLUSTERS AFTER POOLING O
     *N THE BASIS OF ASSOCIATES ONLY.')
      II=0
      DO 3034 I=1,KCPB
      IF(NINCL(I).EQ.0) GO TO 3034
      II=II+1
      IDUCJ=NINCL(I)
      WRITE(6,3035)  I,(INCLUS(I,J),J=1,IDUCJ)
3035  FORMAT(1H0,10X,'THE ACTIVITIES IN THE ',I3,'-TH CLUSTER ARE AS FOL
     *LOWS:',/,15X,50(I3,','))
3034  CONTINUE
C
C        DESCRIBES WHERE EACH ACTIVITY IS BEFORE ELIMINANTS ARE
C        CONSIDERED
C
C
C        EXAMINE EACH ACTIVITY AND DETERMINE WHICH CLUSTER,IF ANY, IT IS
C        IN.
C              LEFT(I) = 0   IMPLIES THAT THE I-TH ACTIVITY IS NOT IN ANY
C                            CLUSTER
C              LEFT(I) = J   IMPLIES THE I-TH ACTIVITY IS IN THE J-TH
C                            CLUSTER
C
      WRITE(6,3104)
3104  FORMAT(1H0,10X,'THE CLUSTER TO WHICH EACH ACTIVITY BELONGS:',/,15X
     *'(ZERO IMPLIES THAT THE ACTIVITY IS NOT IN ANY CLUSTER)')
      DO 3101 I=1,M
      LEFT(I)=0
      DO 3102 J=1,KCPB
      IF(NINCL(J).EQ.0) GO TO 3102
      IDUCK=NINCL(J)
      DO 3110 K=1,IDUCK
      IF(I.EQ.INCLUS(J,K)) GO TO 3107
3110  CONTINUE
3102  CONTINUE
      GO TO 3101
3107  LEFT(I)=J
3101  WRITE(6,3103) I,LEFT(I)
3103  FORMAT(1H ,15X,'THE ',I3,'-TH ACTIVITY IS IN THE ',I3,'-TH CLUSTER
     *')
      INDEXL=1
```

```
C
C        LEFTOVERS ARE ACTIVITIES NOT IN CLUSTERS AFTER ASSOCIATES HAVE
C        BEEN CONSIDERED BUT BEFORE ELIMINANTS HAVE BEEN CONSIDERED
C
C
C        DETERMINE THE NUMBER OF LEFTOVERS. NLEFT
C              LEFTO(L) = J  IMPLIES THAT THE L-TH LEFTOVER IS THE J-TH
C                                ACTIVITY
C
      NLEFT=0
      DO 3122 J=1,M
      IF(LEFT(J).NE.0) GO TO 3122
      NLEFT=NLEFT+1
      LEFTO(NLEFT)=J
 3122 CONTINUE
      WRITE(6,3123) NLEFT
 3123 FORMAT (1H0,10X,'THERE ARE ',I3,' ACTIVITIES NOT IN ANY CLUSTER YE
     *T.')
      WRITE(6,3323)
 3323 FORMAT(1H1,5X,'THE ELIMINANTS OF EACH NON-CRITICAL-PATH ACTIVITY A
     *RE NOW DETERMINED:')
C
C        ELIMINANTS FOR EACH NON-CRITICAL-PATH ACTIVITY ARE NOW
C        DETERMINED
C              NNNCP = THE NUMBER OF ACTIVITIES NOT ON THE CRITICAL PATH
C              NONCP(LE) - THE LE-TH ACTIVITY NOT ON THE CRITICAL PATH
C
      NNNCP=M-KCPB
      LE=0
      DO 5000 I=1,M
      J=1
 5001 IF(I.EQ.ICRITP(J)) GO TO 5000
      J=J+1
      IF(J.LE.KCPB) GO TO 5001
 5002 LE=LE+1
      NONCP(LE)=I
 5000 CONTINUE
      WRITE(6,5005) NNNCP
 5005 FORMAT(1H0,05X,'THERE ARE ',I3,' ACTIVITIES NOT ON THE CRITICAL PA
     *TH.  THEY ARE AS FOLLOWS:')
      IF(NNNCP.EQ.0) GO TO 3124
      DO 5006 I=1,LE
 5006 WRITE(6,5007) I,NONCP(I)
 5007 FORMAT(1H ,15X,I3,'.     ',I3)
      IF(NNNCP.EQ.0) GO TO 3124
      LE=0
 3126 LE=LE+1
      IF (ISTAT(IWWWO).EQ.1) CALL BINV1(&23127,COT(IWWWO),CTIME(IWWWO),
     *IWWWO)
      REDCOS (IWWWO) = REDCOS(IWWWO)-CTIME(IWWWO)+COT(IWWWO)
23127 CONTINUE
      CTIME(IWWWO) = COT(IWWWO)
      CTIME(NONCP(LE)) = COT(NONCP(LE)) + THETA*SIGMA(NONCP(LE))
      IF (ISTAT(NONCP(LE)).EQ.1) CALL BINV1(&7756,CTIME(NONCP(LE)),
     *   COT(NONCP(LE)),NONCP(LE))
      REDCOS(NONCP(LE))=REDCOS(NONCP(LE))-COT(NONCP(LE))+CTIME(NONCP(LE)
     *)
 7756 IWWWO = NONCP(LE)
      WRITE(6,3152) NONCP(LE),CTIME(NONCP(LE))
 3152 FORMAT(1H0,///, 5X,'THE COMPLETION TIME FOR THE ',I3,'-TH ACTIVITY
```

```
     * HAS BEEN CHANGED TO ',E15.5)
      GO TO 22800
 3121 CONTINUE
C
C        DETERMINE THE ELIMINANTS OF THE LE-TH ACTIVITY NOT ON THE
C        CRITICAL PATH
C              NE         = THE NUMBER OF ELIMINANTS FOR THE LE-TH
C                             ACTIVITY NOT ON THE CRITICAL PATH
C              EGRP(J)      = THE J-TH ELIMINANT FOR THE LE-TH ACTIVITY
C                               NOT ON THE CRITICAL PATH
C
      NE=0
      DO 3130 K=1,KCPB
      DO 3131 I=1,KKB
      IF(IBB(I).EQ.ICRITP(K)) GO TO 3130
 3131 CONTINUE
      NE=NE+1
      EGRP(NE)=ICRITP(K)
 3130 CONTINUE
      WRITE(6,3133) NE,NONCP(LE)
 3133 FORMAT(1H0,10X,'THERE ARE ',I3,' ELIMINANTS CORRESPONDING TO ACTIV
     *ITY ',I3)
      IF(NE.EQ.0) GO TO 3171
      DO 3135 K=1,NE
 3135 WRITE(6,3136) K,NONCP(LE),EGRP(K)
 3136 FORMAT(1H ,14X,'THE ',I3,'-TH ELIMINANT CORRESPONDING TO ACTIVITY
     *',I3,' IS ACTIVITY ',I3)
C
C        DETERMINE WHETHER NONCP(LE) IS AN ASSOCIATE
C              JA = 1   IF NONCP(LE)  IS AN ASSOCIATE
C              JA = 2   IF NONCP(LE)  IS NOT AN ASSOCIATE
C
      K=NONCP(LE)
      JA=1
      IF(LEFT(K).EQ.0) JA=2
      IF(JA.EQ.2) GO TO 5010
      IT=LEFT(K)
C
C        THE IT-TH CLUSTER IS EXPANDED TO INCLUDE ELIMINANTS
C
      GO TO 5011
 5010 CONTINUE
C
C        ITTTT IS THE ACTIVITY NUMBER OF THE FIRST ELIMINANT
C        IT IS THE CLUSTER TO WHICH THE FIRST ELIMINANT CURRENTLY BELONG
C
      ITTT=EGRP(1)
      IT=LEFT(ITTT)
      LEFT(NONCP(LE))=IT
C
C        THE IT-TH CLUSTER IS EXPANDED TO INCLUDE ELIMINANTS
C
      NINCL(IT)=NINCL(IT)+1
      INCLUS(IT,NINCL(IT))=NONCP(LE)
      IF(NE.EQ.1) GO TO 3171
 5011 DO 3172 J=JA,NE
C
C        IU IS THE ACTIVITY NUMBER OF THE NEXT ELIMINANT
C        IF IU IS IN CLUSTER K, THEN CLUSTER K IS POOLED INTO CLUSTER IT
C
```

```
      IU=EGRP(J)
      K=LEFT(IU)
 3182 IF(IT.EQ.K) GO TO 3172
      NCLUS=NCLUS-1
      IW=NCLINC(K)
      DO 3183 IA=1,IW
      LEFT(ICRITP(CLINCL(K,IA)))=IT
      NCLINC(IT)=NCLINC(IT)+1
 3183 CLINCL(IT,NCLINC(IT))=CLINCL(K,IA)
      NCLINC(K)=0
      IW=NINCL(K)
      NINCL(K)=0
      DO 3184 IA=1,IW
      LEFT(INCLUS(K,IA))=IT
      NINCL(IT)=NINCL(IT)+1
 3184 INCLUS(IT,NINCL(IT))=INCLUS(K,IA)
 3172 CONTINUE
 3171 CONTINUE
      IF(LE.LT.NNNCP) GO TO 3126
C
C        END OF POOLING BASED ON ELIMINANTS EXCEPT FOR THE FOLLOWING
C        DESCRIPTION
C
      WRITE(6,3173) NCLUS
 3173 FORMAT(1H1,05X,'THERE ARE ',I3,' CLUSTERS.')
      DO 3176 I=1,KCPB
      IF(NINCL(I).EQ.0) GO TO 3176
      IDD=NINCL(I)
      WRITE(6,3174) NINCL(I),I,(INCLUS(I,J),J=1,IDD)
 3174 FORMAT(1H0,10X,'THERE ARE ',I3,' ACTIVITIES IN THE ',I3,'-TH CLUST
     *ER.  THEY ARE AS FOLLOWS:',/,20X,50(I3,','))
      IDUCK=NCLINC(I)
      WRITE(6,3175) NCLINC(I),(CLINCL(I,J),J=1,IDUCK)
 3175 FORMAT(1H0,15X,I3,' CLUSTERS HAVE BEEN POOLED TO MAKE THIS CLUSTER
     *. THEY WERE AS FOLLIWS:',/,20X,50(I3,','))
 3176 CONTINUE
      WRITE (6,2776) SAMSIZ
 2776 FORMAT (///5X,          ' THE NUMBER OF PERCENTILE COMBINATIONS EXP
     *LICITLY CONSIDERED IN DETERMINING THE UPPER BOUNDS AND LOWER BOUND
     *S'/,6X,'ON THE NETWORK COMPLETION TIME DISTRIBUTION AND THE UPPER
     *BOUNDS ON ITS MOMENTS IS EQUAL TO  ',I5)
      CALL CLOCK(XRAN)
      IYUTS = XRAN
      WRITE (6,3238) IYUTS
 3238 FORMAT (1H0,5X,'THE INITIALIZATION PARAMETER FOR THE SAMPLING IS I
     *Y = ',I10)
C
C        STATEMENT NUMBER 3124 MARKS THE END OF POOLING CLUSTERS BASED
C              ON LEFTOVERS AND ELIMINANTS
C
 3124 CONTINUE
C
C
C        THE FINAL CLUSTERS HAVE NOW BEEN DETERMINED
C        THE 2**NINCL(I) RUNS ARE NOW AVERAGED FOR ALL I WITH NINCL(I)>0
C        THE IZZ( ) ARE USED TO REPRESENT ALL OF THE 2**NINCL
C        POSSIBILITIES.
C        THIS IS WHERE THE BINARY REPRESENTATION IS CONSTRUCTED.
C
 6008 IF(ICBCP.EQ.1) ICBCP=2
```

```
      INDEXL=2
      IR=0
 3200 IR=IR+1
      DO 6031 I=1,10
 6031 MOMENT(IR,I) = 0.0
      IF(IR.GT.KCPB) GO TO 3208
      IF(NINCL(IR).EQ.0) GO TO 3200
      IDUCK=NCLINC(IR)
      DO 3410 I=1,IDUCK
      K=CLINCL(IR,I)
 3410 CONTINUE
      L=ICRITP(K)
 3310 IP = 0
      NIB= 0
C
C     NIB   = NUMBER OF PERCENTILE COMBINATIONS IN THE SAMPLE
C     PNIB  = PERCENTAGE OF THE TOTAL NUMBER OF PERCENTILE COMBINATIONS
C             EXPLICITLY CONSIDERED
C     NSAVE = VECTOR CONTAINING THE LOWER BOUNDS ON THE NETWORK
C             COMPLETION TIME DISTRIBUTION TO BE AVERAGED WITH THE
C             UPPER BOUNDS ON THE NETWORK TO YIELD THE AVERAGE NETWORK
C             COMPLETION TIME DISTRIBUTION.
C     KNIB  = NIB ASSOCIATED WITH NSAVE.
C
C
      IC=NINCL(IR)
      IB=2**NINCL(IR)
C     IDOALL = 1 MEANS ALL PERCENTILE COMBINATIONS ARE EXPLICITLY
C               CONSIDERED.
C     IDOALL = 0 MEANS TO SAMPLE.
C
C
      IDOALL = 0
      IF (SAMSIZ.LE.0.OR.SAMSIZ.GE.IB) IDOALL=1
      DO 3222 I=1,M
 3222 CTIME(I) = COT(I)
      GO TO 20000
C
C        STATEMENT 3204  IS THE RE-ENTRY POINT FROM THE DUAL SIMPLEX
C        ALGORITHM WHEN A CLUSTER AVERAGE IS BEING COMPUTED
C
 3204 CONTINUE
      IF (IDLB.EQ.1) GO TO 9900
      DO 6032 I=1,10
 6032 MOMENT(IR,I) = MOMENT(IR,I) + B1INV(1,NMMP1)**I
      IF (ICBCP.NE.2) GO TO 10001
 9900 X= B1INV(1,NMMP1)
      X=X-1.0D-10
      I=0
 6420 I=I+1
      IF(X.GT.FD(I)) GO TO 6420
      NLEFD(I) = NLEFD(I) + 1
10001 CONTINUE
20000 IP=IP+1
      NIB = NIB+1
C
C        GENERATE NEXT PERCENTILE COMBINATIONS TO BE EXPLICITLY
C        CONSIDERED.
C
      IF (IDOALL.EQ.0) GO TO 20500
```

```
      IF (IP.GT.IB) GO TO 3207
      RANSAM = IP
      GO TO 20501
20500 IF (NIB.GT.SAMSIZ) GO TO 3207
      IYUTS = IYUTS*65539
      IF (IYUTS) 6210,6211,6211
 6210 IYUTS = IYUTS+2147483647+1
 6211 XRAN = IYUTS
      XRAN = XRAN*.4656613E-9
      RANSAM = XRAN*DFLOAT(IB-1) + 1
20501 CONTINUE
C
C     CONVERT THE RANDOM NUMBER, RANSAM, TO A BINARY NUMBER TO DEFINE A
C     PERCENTILE COMBINATION.
C
      KRAN = RANSAM
      DO 8505 I=1,IC
      IHALF = KRAN/2
      IZ = KRAN - IHALF*2
      L = INCLUS(IP,I)
      CTIME(L) = IZ*FHI(L)-IZ*FLO(L) + FLO(L)
 8505 KRAN = IHALF
      CALL BINVA(&2800)
 3207 NIB = NIB-1
      IF (IDLB.EQ.1) GO TO 9011
      DO 6030 I=1,10
 6030 MOMENT(IR,I) = MOMENT(IR,I)/NIB
      IF (ICBCP.EQ.2) GO TO 6011
      GO TO 3200
 3208 WRITE (6,6362)
 6362 FORMAT (1H1)
      PNIB = DFLOAT(NIB*100)/DFLOAT(IB)
      IF (SAMSIZ.LE.0) WRITE(6,6045)
      IF (SAMSIZ.GT.0) WRITE(6,6056) SAMSIZ
 6056 FORMAT( 5X,' THE FOLLOWING TABLE WAS COMPUTED CONSIDERING AT MOST
     *',I8,' PERCENTILE COMBINATIONS .')
 6046 FORMAT( 5X,' THE FOLLOWING TABLE WAS COMPUTED CONSIDERING ONLY ',
     *  I8,' PERCENTILE COMBINATIONS OR  ',F6.2,' PERCENT OF ALL COMBINA
     *TIONS.')
 6045 FORMAT (5X,' THE FOLLOWING TABLE WAS COMPUTED CONSIDERING ALL PERC
     *ENTILE COMBINATIONS.')
      DO 10000 J=1,10
      ITMAX=0
      TMAX=0.
      DO 5021 I=1,KCPB
      IF(NINCL(I).EQ.0) GO TO 5021
      IF (MOMENT(I,J).LE.TMAX) GO TO 5021
      TMAX = MOMENT(I,J)
      ITMAX=I
 5021 CONTINUE
      WRITE (6,5023) J,J,MOMENT(ITMAX,J)
 5023 FORMAT (1H0,5X,'A LOWER BOUND, T-(',I2,';THETA,LAMBDA), ON THE '
     *,         I2 ,'-TH MOMENT OF THE NETWORK COMPLETION TIME = ',F15.5)
10000  CONTINUE
C
C         BEGIN THE PROCEDURE FOR DETERMINING UPPER BOUNDS ON THE MOMENTS
C         OF THE NETWORK COMPLETION TIME AND LOWER BOUNDS ON THE
C         DISTRIBUTION OF THE COMPLETION TIMES.
C
C
```

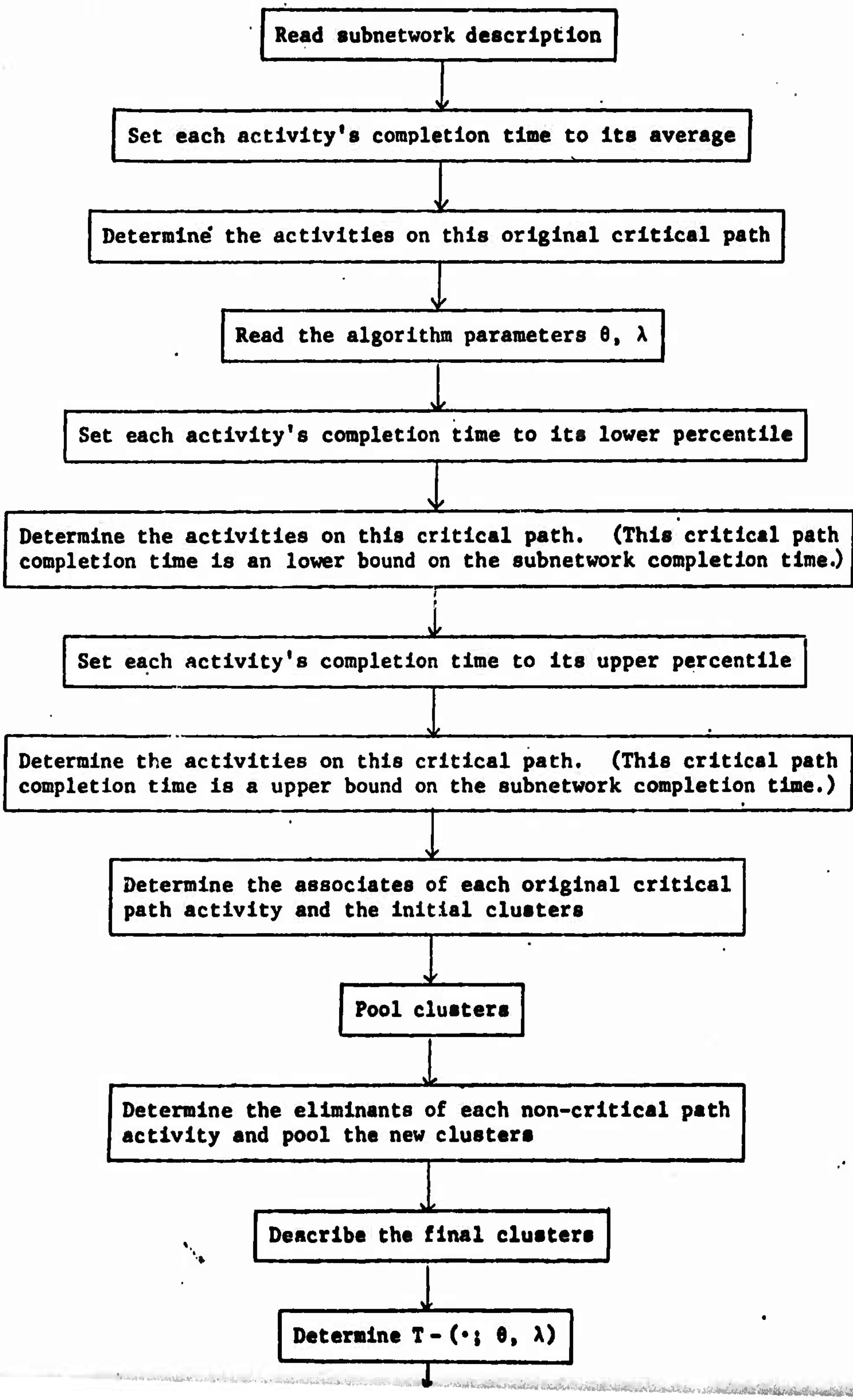
```
C         POOL ALL OF THE PREVIOUS CLUSTERS INTO ONE CLUSTER
C         NCL  =  THE INDEX OF THE RESULTANT POOLED CLUSTER
C         NNCL = THE NUMBER OF ACTIVITIES IN THIS POOLED CLUSTER
C
      IF(NCLUS.GT.1) GO TO 6000
      I=0
 6001 I=I+1
      IF(NINCL(I).EQ.0) GO TO 6001
      NCL=I
      GO TO 6002
 6000 NMAX = 0
      DO 6003 I=1,KCPB
      IF(NINCL(I).LE.NMAX) GO TO 6003
      NCL =I
      NMAX=NINCL(I)
 6003 CONTINUE
      DO 6004 I=1,KCPB
      IF(NINCL(I).EQ.0) GO TO 6004
      IF(I.EQ.NCL) GO TO 6004
      K=NINCL(NCL)
      JJ=NINCL(I)
      NINCL(I)=0
      DO 6005 J=1,JJ
      K=K+1
 6005 INCLUS(NCL,K)=INCLUS(I,J)
      NINCL(NCL)=NINCL(NCL)+JJ
 6004 CONTINUE
 6002 NNCL=NINCL(NCL)
C
C
C
C         THE UPPER BOUNDS, T+(R,THETA,LAMBDA), ARE NOW DETERMINED.
C
C
C         FOR THE SAKE OF NUMERICAL ACCURACY THE PERT PROBLEM
C         WITH NEW ACTIVITY TIMES IS INITIALLY SOLVED FROM SCRATCH
C         INSTEAD OF UPDATING AN OLD SOLUTION.  AFTER THIS
C         REINITIALIZATION, THE REMAINING CRITICAL PATH TIMES ARE
C         DETERMINED BY UPDATING THIS SOLUTION.
C
      DO 6006 I=1,M
      CTIME(I)=FHI(I)
 6006 COT(I)=FHI(I)
      DO 6007 J=1,NNCL
      I =INCLUS(NCL,J)
 6007 CTIME(I)=.5*(FLO(I )+FHI(I ))
      DO 7101 I=1,IEDF
 7101 NLEFD(I) = 0
      ICBCP=1
      GO TO 6010
 6011 WRITE (6,6264)
 6264 FORMAT(////)
      PNIB = DFLOAT(NIB*100)/DFLOAT(IB)
      IF (IDOALL.EQ.0) WRITE(6,6046) NIB,PNIB
      IF (IDOALL.EQ.1) WRITE(6,6045)
      DO 10009 J=1,10
10009 WRITE (6,6012) J,J,MOMENT(NCL,J)
 6012 FORMAT (1H0,5X,'AN UPPER BOUND, T+(',I2,';THETA,LAMBDA), ON THE '
     *,I2,'-TH MOMENT OF THE NETWORK COMPLETION TIME = ',E15.5)
      DO 6900 I=2,IEDF
```

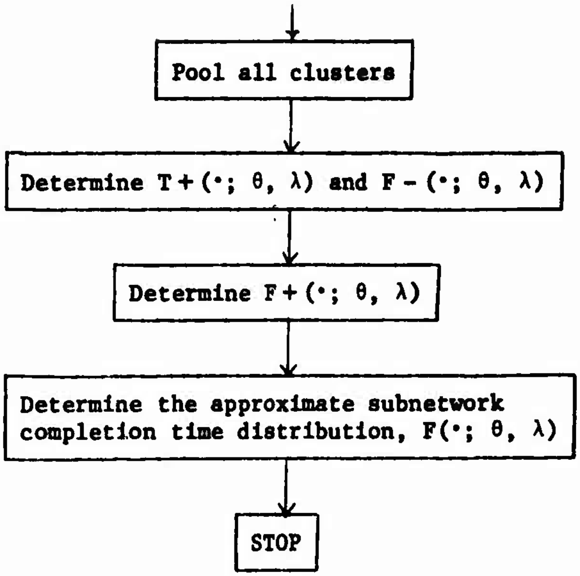
```
      II=I-1
 6900 NLEFD(I) = NLEFD(I) + NLEFD(II)
      WRITE (6,6362)
      PNIB = DFLOAT(NIB*100)/DFLOAT(IB)
      IF (IDOALL.EQ.0) WRITE(6,6046) NIB,PNIB
      IF (IDOALL.EQ.1) WRITE(6,6045)
      WRITE(6,6423)
 6423 FORMAT(1H0,5X, 'A LOWER BOUND ON THE NETWORK COMPLETION TIME DISTR
     *IBUTION: F-(.;THETA;LAMBDA)')
      KNIB = NIB
      DO 6421 I=1,IEDF
      NSAVE(I)=NLEFD(I)
      X=NLEFD(I)
      X=X/NIB
 6421 WRITE(6,6422) FD(I),THETA,LAMBDA,X
 6422 FORMAT(17X,'F-(',E15.5,';',E15.5,';',E15.5,') = ',E15.5)
C
C
C         BEGIN THE PROCEDURE FOR DETERMINING UPPER BOUNDS ON THE
C         NETWORK COMPLETION TIME DISTRIBUTION
C
C         FOR THE SAKE OF NUMERICAL ACCURACY THE PERT PROBLEM
C         WITH NEW ACTIVITY TIMES IS INITIALLY SOLVED FROM SCRATCH
C         INSTEAD OF UPDATING AN OLD SOLUTION.  AFTER THIS
C         REINITIALIZATION, THE REMAINING CRITICAL PATH TIMES ARE
C         DETERMINED BY UPDATING THIS SOLUTION.
      DO 9006 I=1,M
      CTIME(I)=FLO(I)
 9006 COT(I)=FLO(I)
      DO 9007 J=1,NNCL
      I =INCLUS(NCL,J)
 9007 CTIME(I)=.5*(FLO(I )+FHI(I ))
      DO 9101 I=1,IEDF
 9101 NLEFD(I) = 0
      ICBCP = 1
      IDLB = 1
      GO TO 6010
 9011 DO 9990 I=2,IEDF
      II=I-1
 9990 NLEFD(I) = NLEFD(I) + NLEFD(II)
      WRITE (6,6264)
      PNIB = DFLOAT(NIB*100)/DFLOAT(IB)
      IF (IDOALL.EQ.0) WRITE(6,6046) NIB,PNIB
      IF (IDOALL.EQ.1) WRITE(6,6045)
      WRITE(6,9423)
 9423 FORMAT(1H0,5X,'AN UPPER BOUND ON THE NETWORK COMPLETION TIME DISTR
     *IBUTION: F+(.;THETA;LAMBDA)')
      DO 9421 I=1,IEDF
      X=NLEFD(I)
      X=X/NIB
 9421 WRITE(6,9422) FD(I),THETA,LAMBDA,X
 9422 FORMAT(17X,'F+(',E15.5,';',E15.5,';',E15.5,') = ',E15.5)
C
C         THE APPROXIMATE NETWORK COMPLETION TIME DISTRIBUTION
      WRITE (6,6362)
C
      WRITE(6,9472)
 9472 FORMAT(1H0,5X,'AN APPROXIMATE NETWORK COMPLETION TIME DISTRIBUTION
     *:',//,15X,'F(.;THETA,LAMBDA) = .5 * ( F+(.;THETA,LAMBDA) + F-(.;TH
     *ETA,LAMBDA) )',//)
```

```
      DO 9471 I=1,IEDF
      X = .5*DFLOAT(NSAVE(I))/KNIB+.5*DFLOAT(NLEFD(I))/NIB
 9471 WRITE(6,9473) FD(I),THETA,LAMBDA,X
 9473 FORMAT(17X,' F(',E15.5,';',E15.5,';',E15.5,') = ',E15.5)
 999  WRITE(6,850)
      STOP
      END
      SUBROUTINE BINVA(*)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON B1INV,REDCOS,CTIME,XB1,INBASE,IHEAD,ITAIL,NMMP1,NMM,N,ISTAT
      COMMON M,MP1
      DIMENSION  ISTAT(100),IHEAD(60),ITAIL(60),XB1(41)
      DIMENSION B1INV(41,41),INBASE(40),CTIME(100),REDCOS(100)
C
C        UPDATE THE FIRST ROW OF B1INV AFTER CHANGING CTIME
C
      DO 1 I=2,NMMP1
      B1INV(1,I) = 0.0
      DO 1 J=2,NMMP1
    1 B1INV(1,I) = B1INV(1,I) + B1INV(J,I)*CTIME(INBASE(J-1))
C
C        UPDATE VALUE OF THE OBJECTIVE FUNCTION
C
      XB1(1) = B1INV(1,NMMP1)
      RETURN1
      END
      SUBROUTINE BINV1 (*,TMNEW,TMOLD,ID)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON B1INV,REDCOS,CTIME,XB1,INBASE,IHEAD,ITAIL,NMMP1,NMM,N,ISTAT
      COMMON M,MP1
      DIMENSION  ISTAT(100),IHEAD(60),ITAIL(60),XB1(41)
      DIMENSION B1INV(41,41),INBASE(40),CTIME(100),REDCOS(100)
C
C        COMPUTE THE REDCOS CORRESPONDING TO ONE CHANGE IN CTIME
C
      DO 2 I=1,NMM
    2 IF(INBASE(I).EQ.ID) II=I+1
      DIFF =  TMNEW-TMOLD
C        TMNEW IS THE NEW TIME AND TMOLD IS THE OLD TIME CORRESPONDING
C        TO THE SINGLE CHANGE IN CTIME
C
      DO 1 K=1,M
      IF(ISTAT(K).EQ.1) GO TO 1
      REDCOS(K) = REDCOS(K)-DIFF*(B1INV(II,IHEAD( K)+1) -
     *   B1INV(II,ITAIL( K)+1))
    1 CONTINUE
      DO 3 K=MP1,N
      IF (ISTAT(K).EQ.1) GO TO 3
      REDCOS(K) = REDCOS(K) - DIFF*B1INV(II,K-M+1)
    3 CONTINUE
C
C        UPDATE THE FIRST ROW OF B1INV AFTER CHANGING CTIME
C
      DO 10 I=2,NMMP1
      B1INV(1,I)=B1INV(1,I)+DIFF*B1INV(II,I)
   10 CONTINUE
C
C        UPDATE VALUE OF THE OBJECTIVE FUNCTION
C
      XB1(1) = B1INV(1,NMMP1)
```

```
      RETURN1
      END
```

Original Subnetwork Analysis Program: Flowchart




Pool all clusters
Determine T+(·; θ, λ) and F−(·; θ, λ)
Determine F+(·; θ, λ)
Determine the approximate subnetwork completion time distribution, F(·; θ, λ)
STOP

APPENDIX D

Monte Carlo PERT Simulation Program

The Monte Carlo PERT Simulation Program will generate a random sample of network completion times. The required input is

(a) an acyclic network with one sink,

(b) parameters describing each activity's completion time distribution, and

(c) the size of the random sample to be generated.

Currently, the program generates each activity's random sample of completion times from a chi-square distribution with 3 degrees of freedom that has been linearly transformed to have specified 15-th and 85-th percentiles. The activity time distribution can be easily changed. The basic output is the ordered sample of random network completion times and the corresponding empirical distribution function. The critical paths associated with the sample of network completion times are not determined.

The generation of the random sample of network completion times involves only one network but varying values of the individual activity completion times. It is computationally faster to find the network completion time for a new set of activity completion times by "updating" the network completion time for a previous set of activity completion times than it is to start all over each time. Since the Simplex Algorithm applied to the dual of the PERT problem is ideally suited for this type of "updating", the basic computational technique for determining the network completion times is the Simplex Algorithm (see e.g., G. Hadley, Linear Programming).

A listing of the Monte Carlo Simulation Program is given at the end of this appendix.

Specific Input Instructions:

Card 1.	Col. 1-3 :	The number of activities in the network, Format (I3).
	Col. 4-6 :	The number of nodes in the network, Format (I3).
Card 2.	Col. 11-15:	The number of random network completion times to be generated, Format (I5).
	Col. 21-25:	The number of parameters needed for generating the random activity completion times, Format (I5).

For each activity one card with:

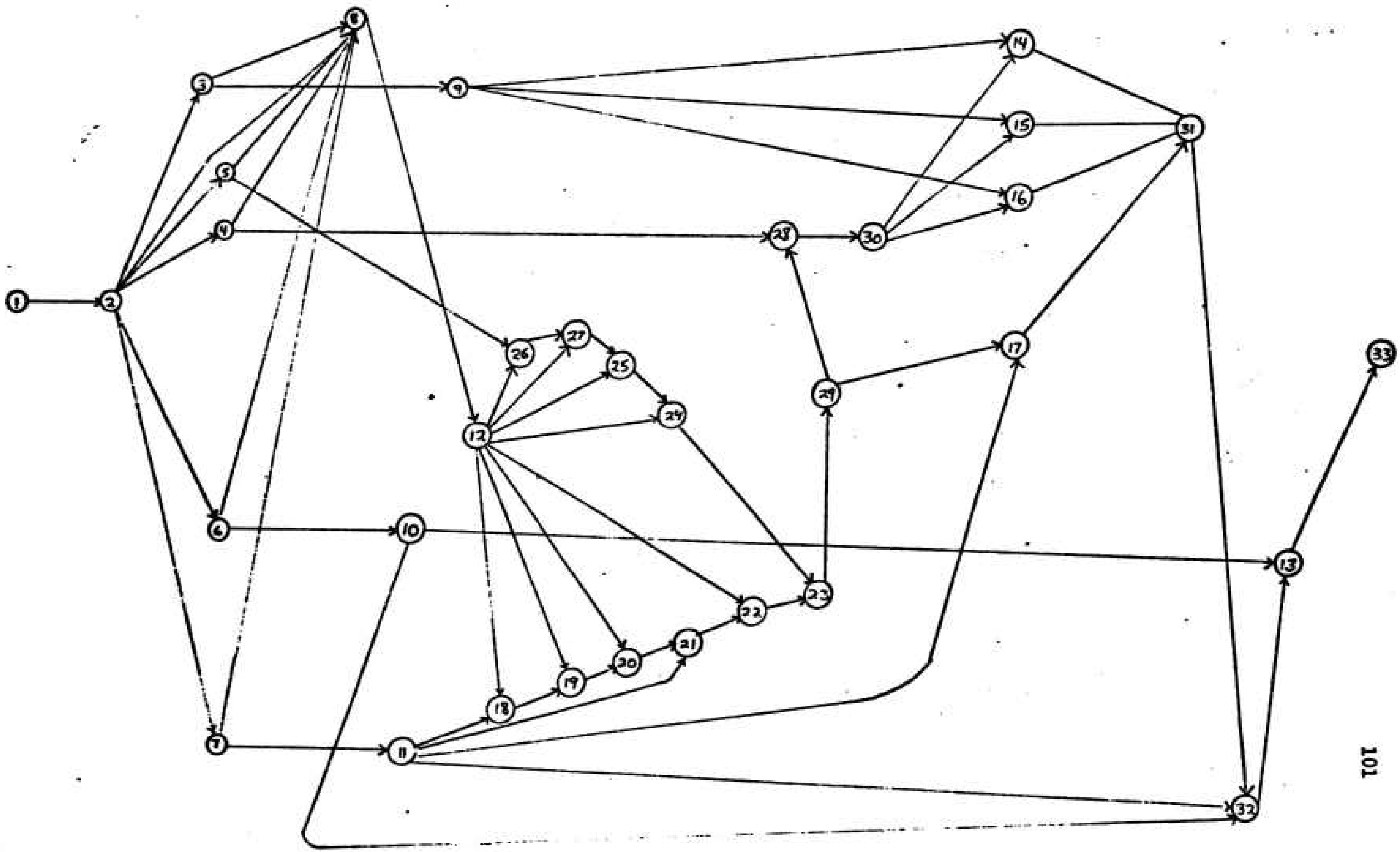
Col. 11-15:	The origin node of the activity, Format (I5)
Col. 21-25:	The terminal node of the activity, Format (I5)
Col. 31-40:	Parameter 1. The 15-th percentile of the activity's completion time distribution, Format (F10.4).
Col. 41-50:	Parameter 2. The 85-th percentile of the activity's completion time distribution, Format (F10.4)

The nodes should be numbered 1, 2, ..., n with the sink being number n.

Example:

The program's input and output are illustrated in terms of the network in Figure D-1.

Figure D-1. Monte Carlo PERT Simulation Program Example Network



SAMPLE INPUT

58 33

	5	2		
01	1	2		
02	8	12	336.27	429.47
03	2	3	57.47	89.96
04	2	4	57.47	89.96
05	2	5	57.47	89.96
06	2	6	57.47	89.96
07	2	7	57.47	89.96
08	3	8	68.[illegible]	107.95
09	4	8	68.[illegible]6	107.95
10	5	8	68.[illegible]	107.95
11	6	8	68.96	107.95
12	7	8	68.96	107.95
13	2	8	150.36	193.49
14	6	10	333.96	403.85
15	3	9	333.96	403.85
16	7	11	355.10	409.85
17	11	18	141.75	221.90
18	10	13	672.36	783.11
19	9	14	560.89	660.00
20	9	15	560.89	660.00
21	9	16	560.89	660.00
22	11	17	542.80	638.71
23	12	18	111.10	173.92
24	18	19	256.03	346.98
25	12	19	302.80	400.67
26	12	20	311.95	410.71
27	11	21	423.58	530.74
28	12	22	315.35	415.71
29	19	20	7.66	11.99
30	20	21	16.77	22.55
31	21	22	11.49	17.99
32	22	23	39.54	48.87
33	12	26	301.91	400.86
34	5	26	767.09	892.74
35	12	27	350.31	460.06
36	12	24	382.32	464.98
37	12	25	385.28	461.54
38	25	24	11.49	17.99
39	24	23	16.28	23.00
40	26	27	7.66	11.99
41	27	25	20.86	28.29
42	4	28	810.17	976.10
43	23	29	15.32	23.99
44	28	30	15.32	23.99
45	14	31	57.47	89.96
46	16	31	49.81	77.96
47	15	31	53.64	83.96
48	17	31	88.12	137.94
49	31	32	3.83	6.00
50	32	13	49.81	77.96
51	13	33	109.01	152.63
52	11	32	745.50	811.32
53	10	32	714.74	799.83
54	30	14		
55	30	16		
56	30	15		
57	29	17		

58 29 28 103

SAMPLE OUTPUT

INITIAL INPUT

ACTIVITY	ORIGIN	TERMINAL	PARAMETER 1	PARAMETER 2	PARAMETER 3 (NONE SPECIFIED)	PARAMETER 4 (NONE SPECIFIED)	PARAMETER 5 (NONE SPECIFIED)
1	1	2	0.0	0.0			
2	8	12	336.2700	429.4700			
3	2	3	57.4700	89.9600			
4	2	4	57.4700	89.9600			
5	2	5	57.4700	89.9600			
6	2	6	57.4700	89.9600			
7	2	7	57.4700	89.9600			
8	3	8	68.9600	107.9500			
9	4	8	68.9600	107.9500			
10	5	8	68.9600	107.9500			
11	6	8	68.9600	107.9500			
12	7	8	68.9600	107.9500			
13	2	8	150.3600	193.4900			
14	6	10	333.9600	403.8500			
15	3	9	333.9600	403.8500			
16	7	11	355.1000	409.8500			
17	11	18	141.7500	221.9000			
18	10	13	672.3600	783.1100			
19	9	14	560.8900	660.0000			
20	9	15	560.8900	660.0000			
21	9	16	560.8900	660.0000			
22	11	17	542.8000	638.7100			
23	12	18	111.1000	173.9200			
24	18	19	256.0300	346.9800			
25	12	19	302.8000	400.6700			
26	12	20	311.9500	410.7100			
27	11	21	423.5800	530.7400			
28	12	22	315.3500	415.7100			
29	19	20	7.6600	11.9900			
30	20	21	16.7700	22.5500			
31	21	22	11.4900	17.9900			
32	22	23	39.5400	48.8700			
33	12	26	301.9100	400.8600			
34	5	26	767.0900	892.7400			
35	12	27	350.3100	460.0600			
36	12	24	382.3200	464.9800			
37	12	25	385.2800	461.5400			
38	25	24	11.4900	17.9900			
39	24	23	16.2800	23.0000			
40	26	27	7.6600	11.9900			
41	27	25	20.8600	28.2900			
42	4	28	810.1700	976.1000			
43	23	29	15.3200	23.9900			
44	28	30	15.3200	23.9900			
45	14	31	57.4700	89.9600			
46	16	31	49.8100	77.9600			
47	15	31	53.6400	83.9600			
48	17	31	88.1200	137.9400			
49	31	32	3.8300	6.0000			
50	32	13	49.8100	77.9600			
51	13	33	109.0100	152.6300			
52	11	32	745.5000	811.3200			
53	10	32	714.7400	799.8300			
54	30	14	0.0	0.0			
55	30	16	0.0	0.0			
56	30	15	0.0	0.0			
57	29	17	0.0	0.0			

58	29	28	0.0	0.0

THE CRITICAL PATH TIMES FOR 5 TRIALS ARE AS FOLLOWS: (TIME/OBSERVED PERCENTILE)

0.14837D+04	0.15212D+04	0.15322D+04	0.15327D+04	0.15587D+04
0.20000D+00	0.40000D+00	0.60000D+00	0.80000D+00	0.10000D+01

```
C
C        MONTE CARLO PERT
C
      IMPLICIT REAL*8 (A-H,O-Z)
C        FOR THE SAKE OF  IDENTIFYING THE APPROPRIATE DIMENSIONS, LET
C              M = THE NUMBER OF ACTIVITIES IN THE NETWORK
C              NMM = NUMBER OF NODES IN THE NETWORK
C              NMMP1  = NMM + 1
C              N  = M + NMM
C              NTRIAL = NUMBER OF SIMULATED SETS OF ACTIVITY
C              COMPLETION TIMES
C              NPARM = NUMBER OF PARAMETERS NEEDED FOR GENERATING RANDOM
C              ACTIVITY COMPLETION TIMES FROM A PARTICULAR DISTRIBUTION
C
C
C     INTEGER TAIL( M),HEAD( M),PARM(M,NPARM),TIMES(NTRIAL)
C     DIMENSION B1INV(NMMP1,NMMP1),CTIME(N),XB1(NMMP1),Y1(NMMP1)
C     DIMENSION REDCOS(N),ISTAT(N),INBASE(NMM),CDF(5),NSPEC(20)
C
C
C        CURRENTLY THE DIMENSIONS ARE SET FOR
C              M=60
C              NMM=40
C              NTRIAL=2000
C
      COMMON B1INV,REDCOS,CTIME,XB1,INBASE,HEAD,TAIL,NMMP1,NMM,N,ISTAT
      COMMON M,MP1
      INTEGER TAIL(60),HEAD(60),TRIAL,NSPEC(20)
      DIMENSION INBASE(40),PARM(60,5),CDF(5),TIMES(2000)
      DIMENSION B1INV(41,41),CTIME(100)
      DIMENSION XB1(41),Y1(41),REDCOS(100),ISTAT(100)
      DATA NSPEC(1),NSPEC(5),NSPEC(9),NSPEC(13),NSPEC(17)/5*'(NON'/
      DATA NSPEC(2),NSPEC(6),NSPEC(10),NSPEC(14),NSPEC(18)/5*'E SP'/
      DATA NSPEC(3),NSPEC(7),NSPEC(11),NSPEC(15),NSPEC(19)/5*'ECIF'/
      DATA NSPEC(4),NSPEC(8),NSPEC(12),NSPEC(16),NSPEC(20)/5*'IED)'/
      DATA BLANKS/'    '/
C        M = THE NUMBER OF ACTIVITIES IN THE NETWORK
C        NMM = THE NUMBER OF NODES IN THE PERT NETWORK
      READ(5,100)  M,NMM
100   FORMAT(2I3)
      N=NMM+M
      MP1=M+1
      NMMP1=NMM+1
      TRIAL = 0
C
C        SIMULATION VARIABLES
C
C              NTRIAL = NUMBER OF SIMULATED SETS OF ACTIVITY
C              COMPLETION TIMES
C              NPARM = NUMBER OF PARAMETERS NEEDED FOR GENERATING RANDOM
C              ACTIVITY COMPLETION TIMES FROM A PARTICULAR DISTRIBUTION
C        TIMES = VECTOR CONTAINING THE OPTIMUM VALUE FOR EACH TRIAL
C        PARM(I,1) = THE LOWER PERCENTILE POINT FOR THE I-TH ACTIVITY
C        PARM(I,2) = THE UPPER PERCENTILE POINT FOR THE I-TH ACTIVITY
C
C
      READ (5,2501) NTRIAL,NPARM
C
C        THE ACTIVITIES ARE DESCRIBED IN TERMS OF THEIR NODES
```

```
C         II=THE TAIL NODE, THE ORIGIN NODE
C         JJ=THE HEAD NODE, THE TERMINAL NODE
C
      DO 610 I=1,M
      READ(5,2501) II,JJ,(PARM(I,J),J=1,NPARM)
 2501 FORMAT(10X,I5,5X,I5, 5X,5F10.4)
      TAIL(I) = II
  610 HEAD(I)=JJ
      NNPARM = 4*NPARM
      DO 222 I=1,NNPARM
  222 NSPEC(I) = BLANKS
      WRITE(6,2700)
 2700 FORMAT(1H1,15X,'INITIAL INPUT')
      WRITE(6,2701)
 2701 FORMAT(1H0,10X,'ACTIVITY   ORIGIN    TERMINAL    PARAMETER 1       P
     *ARAMETER 2       PARAMETER 3       PARAMETER 4       PARAMETER 5')
      WRITE (6,2509) (NSPEC(I),I=1,20)
 2509 FORMAT (41X,4A4,4(1X,4A4))
      DO 2704 I=1,M
 2704 WRITE(6,2702) I,TAIL(I),HEAD(I),(PARM(I,J),J=1,NPARM)
 2702 FORMAT(1H ,13X,I3,5X,I3,7X,I3,1X,5(7X,F10.4))
C
      DO 104 I=1,NMM
  104 INBASE(I)=M+I
      DO 2001 J=1,M
 2001 ISTAT(J)=0.
      DO 2002 J=MP1,N
 2002 ISTAT(J)=1
      DO 10 II=1,NMMP1
      DO 12 L=1,NMMP1
   12 B1INV(L,II) = 0.
   10 B1INV(II,II) = 1.
      DO 30 I=1,NMM
   30 XB1(I) = 0.
      XB1(NMMP1) = 1.
      TOLR1=1.0D-10
C
      DO 55610 I=MP1,N
55610 CTIME(I) = 0.
 350  CONTINUE
C
C     GENERATE A SET OF ACTIVITY COMPLETION TIMES
C
      TRIAL = TRIAL + 1
      CALL RANTIM (CTIME,PARM,M)
      CALL BINVA (&2800)
C         START THE SIMPLEX ALGORITHM
C         SOLVE THE DUAL PROBLEM
C          THE NUMBER OF VARIABLES IS M REAL + NMM SLACKS
C          FOR A TOTAL OF N VARIABLES
C
 2800 DO 23 J=1,N
      RATS = 0.
      IF (ISTAT(J).EQ.1) GO TO 52800
      IF (J.GT.M) GO TO 22
      RATS =-B1INV(1,HEAD(J)+1)+B1INV(1,TAIL(J)+1) + CTIME(J)
      GO TO 52800
   22 RATS =-B1INV(1,J-M+1)
52800 REDCOS(J)= RATS
   23 CONTINUE
```

```
22800 CONTINUE
      IRMAX=1
      RMAX=REDCOS(1)
      DO 24 J=2,N
      IF(REDCOS(J) .LE. RMAX) GO TO 24
      RMAX=REDCOS(J)
      IRMAX=J
24    CONTINUE
      IF(RMAX .LE. TOLR1) GO TO 401
22824 CONTINUE
      DO 26 L=1,NMMP1
      IF (IRMAX.GT.M) GO TO 50026
      Y1(L) =-B1INV(L,TAIL(IRMAX)+1)+B1INV(L,HEAD(IRMAX)+1)
      GO TO 26
50026 Y1(L) = B1INV(L,IRMAX-M+1)
   26 CONTINUE
      Y1(1) = Y1(1) - CTIME(IRMAX)
      NUMBER=0
      DO 27 L=2,NMMP1
27    IF(Y1(L) .LE. TOLR1)   NUMBER=NUMBER+1
      IF(NUMBER .EQ. NMM) GO TO 403
      RMIN=.99D 20
      IRMIN=0.
      DO 32 II=2,NMMP1
      IF(Y1(II).LE. TOLR1)  GO TO 32
      RATS =XB1(II)/Y1(II)
      RR=RATS-RMIN
      IF(RR .GE. 0.D0) GO TO 32
      RMIN=RATS
      IRMIN=II
32    CONTINUE
      DO 33 J=2,NMMP1
      WW=B1INV(IRMIN ,J)/Y1(IRMIN )
      DO 37 L=1,NMMP1
37    B1INV(L,J)=B1INV(L,J)-WW*Y1(L)
33    B1INV(IRMIN ,J)=WW
C
C        UPDATE THE BASIC VARIABLES: INBASE AND XB1
C
      ISTAT(INBASE(IRMIN-1))=0
      ISTAT(IRMAX)=1
      INBASE(IRMIN-1)=IRMAX
      W=XB1(IRMIN )/Y1(IRMIN )
      DO 38 I=1,NMMP1
38    XB1(I)=XB1(I)-Y1(I)*W
      XB1(IRMIN )=W
      GO TO 2800
403   WRITE(6,530)
530   FORMAT(1H0,5X,'NO FEASIBLE SOLUTION EXISTS.  CHECK YOUR INPUT DATA
     *.')
      WRITE(6,850)
850   FORMAT(1H1)
      GO TO 999
C
C        END OF THE SIMPLEX ALGORITHM
C
401   TIMES(TRIAL) = B1INV(1,NMMP1)
      IF (TRIAL.LT.NTRIAL) GO TO 350
C
C        INBASE IS A SET OF NMM INTEGER VARIABLES WHICH INDICATE THE
```

```
C         COMPOSITION OF THE CURRENT BASIS.  FOR EXAMPLE,
C              INBASE(K) = 7  IMPLIES THAT THE K-TH COLUMN IN THE BASIS B
C                             CORRESPONDS TO THE 7-TH VARIABLE
C
C
C         ISTAT INDICATES THE BASIC STATUS OF EACH VARIABLE
C              ISTAT(K) = 1  IMPLIES THAT THE K-TH VARIABLE IS IN THE
C                            DUAL BASIS
C              ISTAT(K) = 0  IMPLIES THAT THE K-TH VARIABLE IS NOT IN THE
C                            DUAL BASIS
C
C
C
C
C         ORDER THE RANDOMLY GENERATED NETWORK COMPLETION TIMES
C
 3000 LIMIT = NTRIAL
      IPASS = NTRIAL-1
      ICHNG = 1
      DO 4001 J=1,IPASS
      LIMIT = LIMIT-1
      IF (ICHNG.EQ.0) GO TO 3060
      ICHNG = 0
      DO 4002 I=1,LIMIT
      IF (TIMES(I).LE.TIMES(I+1)) GO TO 4002
      TEMP = TIMES(I+1)
      TIMES(I+1) = TIMES(I)
      TIMES(I) = TEMP
      ICHNG = 1
 4002 CONTINUE
 4001 CONTINUE
C
C         DESCRIBE THE ORDERED NETWORK COMPLETION TIMES
C
 3060 WRITE(6,9662)
 9662 FORMAT(1H1)
      WRITE(6,3012) NTRIAL
 3012 FORMAT (1H0,10X,'THE CRITICAL PATH TIMES FOR ',I5,'  TRIALS ARE AS
     * FOLLOWS: (TIME/OBSERVED PERCENTILE)')
      LINE = (NTRIAL+4)/5
      DO 3050 J=1,LINE
      I2 = (J-1)*5+1
      I3 = J*5
      ICNT = 0
      IF (NTRIAL-I2+1.LT.5) I3= NTRIAL
      DO 3051 K=I2,I3
      ICNT = ICNT+1
 3051 CDF(ICNT) = DFLOAT(K)/DFLOAT(NTRIAL)
      WRITE (6,3011)  (TIMES(I),I= I2,I3)
 3011 FORMAT(1H0,5X,5(6X,F15.5))
      WRITE (6,3013) (CDF(K),K=1,ICNT)
 3013 FORMAT (6X,5(6X,F15.5))
 3050 CONTINUE
  999 STOP
      END
      SUBROUTINE BINVA(*)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON BIINV,REDCOS,CTIME,XBI,INBASE,IHEAD,ITAIL,NMMP1,NMM,N,ISTAT
      COMMON M,MP1
      DIMENSION  ISTAT(100),IHEAD(60),ITAIL(60),XBI(41)
```

```
      DIMENSICN B1INV(41,41),INBASE(40),CTIME(100),REDCOS(100)
C
C        UPDATE THE FIRST ROW OF B1INV AFTER CHANGING CTIME
C
      DO 1 I=2,NMMP1
      B1INV(1,I) = 0.0
      DO 1 J=2,NMMP1
    1 B1INV(1,I) = B1INV(1,I) + B1INV(J,I)*CTIME(INBASE(J-1))
C
C        UPDATE VALUE OF THE OBJECTIVE FUNCTION
C
      XB1(1) = B1INV(1,NMMP1)
      RETURN1
      END
      SUBROUTINE RANTIM (CTIMF,PARM,M)
C
C     SUBROUTINE RANTIM GENERATES M RANDOM TIMES FROM A SPECIFIED
C     DISTRIBUTION WITH PARAMETERS CONTAINED  IN PARM(60,5) AND RETURNS
C     WITH THE RESULTS IN CTIMF(99).
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DATA J/0/,IY/19447/,TPI/6.2831853/
      DIMENSICN PARM(60,5),CTIME(99),SAVTIM(99)
      IF (J.NE.0) GO TO 30
C
C     THE FOLLOWING GENERATES A CHI SQUARE RANDOM DEVIATE WITH 3 DF'S
C          TRANSFORMED TO MAKE THE LOWER POINT CORRESPOND TO THE 15-TH
C          PERCENTILE AND THE UPPER POINT CORRESPOND TO THE 85-TH PERC.
C
C     PARM(3,I) = THE  PERCENTILE DIFFERENCE
C
      DO 5 I=1,M
      PARM(I,3) = PARM(I,2)-PARM(I,1)
5     IF (PARM(I,3).EQ.0) CTIME(I) = PARM(I,1)
30    J = J+1
      IF (MOD(J,2).EQ.1) GO TO 20
C
      DO 15 I=1,M
      IF (PARM(I,3).EQ.0 ) GO TO 15
      CTIME(I) = SAVTIM(I)
15    CONTINUE
      RETURN
C
C
C        GENERATE A PAIR OF COMPLETION TIMES FOR EACH ACTIVITY.
C        THE  BOX-MULLER METHOD IS USED TO GENERATE A PAIR OF
C        NORMAL DEVIATES
C        U1 AND U2 = UNIFORM RANDOM NUMBERS
C        W*DSIN(AN) = STANDARD  NORMAL RANDOM VARIABLE
C        W*DCOS(AN) = STANDARD  NORMAL RANDOM VARIABLE
C        CHI3 = A CHI-SQUARE RANDOM VARIABLF  -- GENERATED USING THE
C                 METHOD OF WILSON AND HILFERTY - PROC. NAT. ACADEMY OF
C                 SCIENCE, 1931
C        C1 AND C2 TRANSFORM THE CHI-SQUARE WITH 3 D.F.
C
20    DO 3002 I=1,M
      IF (PARM(I,3).EQ.0) GO TO 3002
      IY = IY*65539
      IF (IY) 3015,3016,3016
3015  IY = IY + 2147483647 + 1
```

```
3016  YFL  =IY
      U1 = YFL*.465613E-9
      IY = IY*65539
      IF (IY) 3025,3026,3026
3025  IY = IY + 2147483647 + 1
3026  YFL= IY
      U2 = YFL*.465613E-9
      W   =-2.*DLOG(U2)
      W = DSQRT(W)
      AN = TPI*U1
      C2 = 4.51927/PARM(I,3)
      C1 = .79777-PARM(I,1)*C2
      CTIME(I) = ((.392530614*W*DSIN(AN)+1.335416269)**3-C1)/C2
      SAVTIM(I) = ((.392530614*W*DCOS(AN)+1.335416269)**3-C1)/C2
3002  CONTINUE
      RETURN
      END
```

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) Texas A&M University	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP Unclassified

3. REPORT TITLE

(6) STATISTICAL PERT: AN IMPROVED SUBNETWORK ANALYSIS PROCEDURE,

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

Progress Report

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

R. L. Sielken Jr., H. O. Hartley, R. K. Spoerl (10) (12) 111p.

6. REPORT DATE (11) Nov[redacted]75	7a. TOTAL NO. OF PAGES 111	7b. NO. OF REFS 1

8a. CONTRACT OR GRANT NO. (15) N00014-68-A-0140 b. PROJECT NO. NRO 47-700 (16) NR-047-700 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) Number 51 (14) THEMIS-TR-51 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

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

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Office of Naval Research

13. ABSTRACT

See page following Table of Contents

(9) THEMIS optimization research program,

347 380 ATTACHMENT III

Unclassified

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Statistical PERT						
Subnetwork Analysis						
Project Scheduling						
Two Point Approximation						
Stochastic Networks						

ATTACHMENT III (continued)

BASIC DISTRIBUTION LIST FOR UNCLASSIFIED TECHNICAL REPORTS

	Copies
Operations Research 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
Lt. Dennis Johnston - ED3 Manned Spacecraft Center NASA Houston, Texas 77058	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 Office of Naval Research Branch Office 495 Summer Street Boston, Massachusetts 02210 Attn: Dr. A.L. Powell	1
Director Office of Naval Research Branch Office 219 South Dearborn Street Chicago, Illinois 60604 Attn: Dr. A.R. Dawe	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

Address	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	

Address	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
Director, National Security Agency Ft. George G. Meade, Maryland 20755 Attn: Dr. Joseph Blum, R44	1

	Copies
Naval Applied Science Laboratory Technical Library, Code 222 Flushing & Washington Avenues Brooklyn, New York 11251	1
Naval Undersea Warfare Center 3202 E. Foothill Boulevard Pasadena, California 91107 Technical Library	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
Florida State University Department of Statistics Tallahassee, Florida 32306 Attn: Prof. I.R. Savage	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
Prof. Emanuel Parzen Statistics Department State Univ. New York at Buffalo 4230 Ridge Lea Road Amherst, New York 14226	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 Attn: Prof. L.A. Savage	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	

	Copies
Office of Gifts & Exchanges Library Texas A&M University College Station, Texas 77843	2
Archives Texas A&M University College Station, TX 77843	1