

AD-A071 498

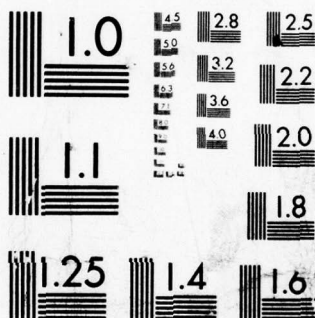
TEXAS A AND M UNIV COLLEGE STATION INST OF STATISTICS F/G 15/5
STATISTICAL PERT: AN IMPROVED PROJECT SCHEDULING ALGORITHM.(U)
FEB 77 C S DUNN, R L SIELKEN N00014-76-C-0038
THEMIS-TR-55 NL

UNCLASSIFIED

1 OF 2

AD
A071-98





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

8/31/79

LEVEL

ADP

①

ADA 071 498



TEXAS A&M UNIVERSITY

COLLEGE STATION, TEXAS 77843

DDC FILE COPY

DDC
JUL 19 1979
RESOLVED
A

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

THEMIS

79 07 18 053

TEXAS A&M UNIVERSITY
PROJECT THEMIS

Technical Report No. 55

STATISTICAL PERT: AN IMPROVED
PROJECT SCHEDULING ALGORITHM

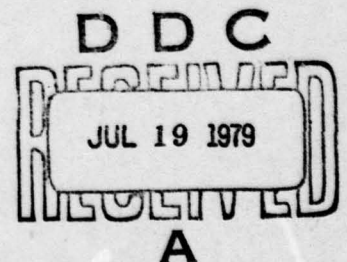
by

Cynthia S. Dunn and Robert L. Sielken, Jr.

Texas A&M University
Office of Naval Research
Contract N00014-76-C-0038
Project NRO47-700

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

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



ATTACHMENT I

79 07 18 053

9) Technical rept,

6) STATISTICAL PERT: AN IMPROVED
PROJECT SCHEDULING ALGORITHM.

by

10) Cynthia S. Dunn and Robert L. Sielken, Jr.

THEMIS OPTIMIZATION RESEARCH PROGRAM

Technical Report No. 55

11) 3 February 1977

12) 105 p.

INSTITUTE OF STATISTICS
Texas A&M University

Research conducted through the
Texas A&M Research Foundation
and sponsored by the
Office of Naval Research

Contract N00014-76-C-0038

15) Project NR047-700

14) THEMIS-TR-55

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

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

ATTACHMENT II

347 380 gmr

Abstract

A project scheduling algorithm is developed and illustrated. For each feasible project deadline time the minimum project cost and corresponding optimal deterministic activity durations are derived. The cost of an activity is assumed to be a convex piecewise linear function of its duration. The algorithm is based upon network-flow techniques including the use of a labeling procedure which preserves complementary slackness.

A computer implementation of the algorithm is documented.

Accession For	
NTIS GRA&I	☒
DDC TAB	☐
Unannounced	☐
Justification	☐
By _____	
Distribution/	
Availability Codes	
Dist.	Avail and/or special
A	23

TABLE OF CONTENTS

	Page
1. INTRODUCTION	1
2. PROBLEM FORMULATION AND SOLUTION PROCEDURE	3
2.1. Problem Formulation	3
2.2. The Dual Problem	11
2.3. The Network-Flow Algorithm	18
2.3.1. The Sketch	20
2.3.2. The Details	22
2.4. Flowchart of the Algorithm	26
3. VERIFICATION OF CLAIMS	30
3.1. Summary	30
3.2. Proofs	31
4. A COMPUTER IMPLEMENTATION	59
4.1. Specific Input Instructions	60
4.2. An Example	62
References	81
Appendix: Program Listing	82

1. INTRODUCTION

This paper describes a scheduling algorithm for a project composed of "jobs" or "activities." These activities are represented by arcs in a directed network. The network nodes represent events in time. The activities at any node can "commence" as soon as all activities "terminating" at that node are completed. Associated with each activity is an interval of possible completion times and an associated piecewise linear cost function. Given that the project must be completed by a specified deadline time, the algorithm determines the individual activity completion times which minimize the total project cost. Repeating the process for all feasible deadline times yields the entire project cost curve and associated optimal activity completion times.

For example, suppose the project consists of activities A, B, C, D, E and the order relations:

A precedes C and D,

B precedes D,

C and D both precede E

and those implied by transitivity. The corresponding network representation is shown in Figure 1 where the arcs represent activities and nodes are events. Notice that arc F does not correspond to any "real" activity but merely represents the order relation that A must precede D. We shall assume that such dummy activities have zero completion times and zero costs.

Using this network representation of the project, the problem of computing the cost curve can be formulated as a network-flow problem. We shall make the following assumptions about the network: there are no directed cycles, and each arc is contained in some directed path from the beginning

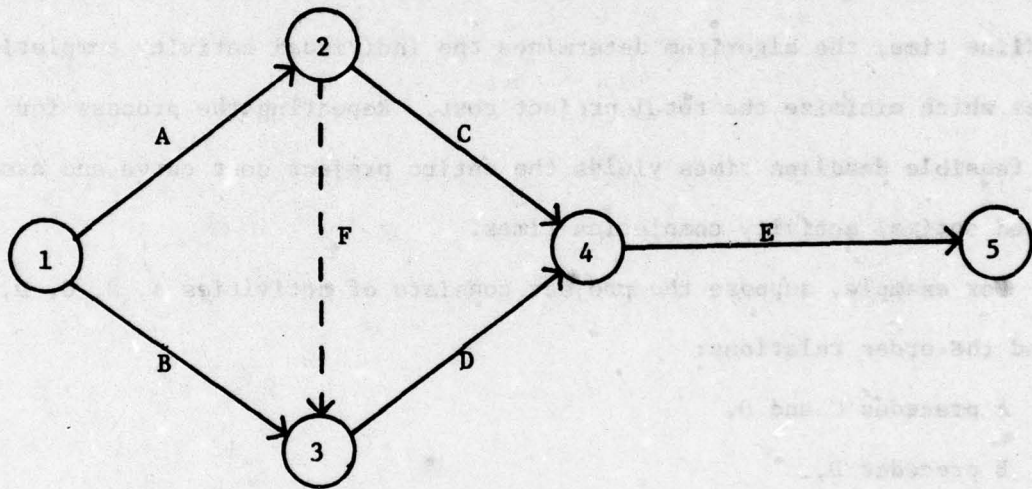


FIGURE 1

node (called the "source") to the terminal node (called the "sink").

This problem can also be formulated as a linear programming problem; however, due to the large number of variables and constraints, it would be impractical storage-wise to solve it using linear programming methods.

D. R. Fulkerson (1961) has formulated a very efficient network-flow algorithm for solving the problem with a linear activity cost function. In this paper, Fulkerson's algorithm has been extended to accept a convex piecewise linear cost function for each individual activity.

2. PROBLEM FORMULATION AND SOLUTION PROCEDURE

2.1. Problem Formulation

The cost of completing an activity is assumed to be a convex piecewise linear function. The cost curve for activity I is depicted in Figure 2. Note that the allowable completion times for activity I have been divided into $NK(I)-1$ intervals: $[TIME(I,1), TIME(I,2)]$, $[TIME(I,2), TIME(I,3)]$, ..., $\{TIME[I, NK(I)-2], TIME[I, NK(I)-1]\}$, $\{TIME[I, NK(I)-1], TIME[I, NK(I)]\}$ with

$$TIME(I, 1) \leq TIME(I, 2) \leq \dots \leq TIME[I, NK(I)]. \quad (2.1)$$

Here we interpret $TIME(I,1)$ as the shortest possible completion time and $TIME[I, NK(I)]$ as the cheapest completion time. Even though the duration of activity I could be greater than $TIME[I, NK(I)]$, such durations would be needlessly expensive and hence $TIME[I, NK(I)]$ is the practical upper bound on the duration of activity I. The intermediate times, $TIME(I,2)$, ..., $TIME[I, NK(I)-1]$, will be called "breakpoints."

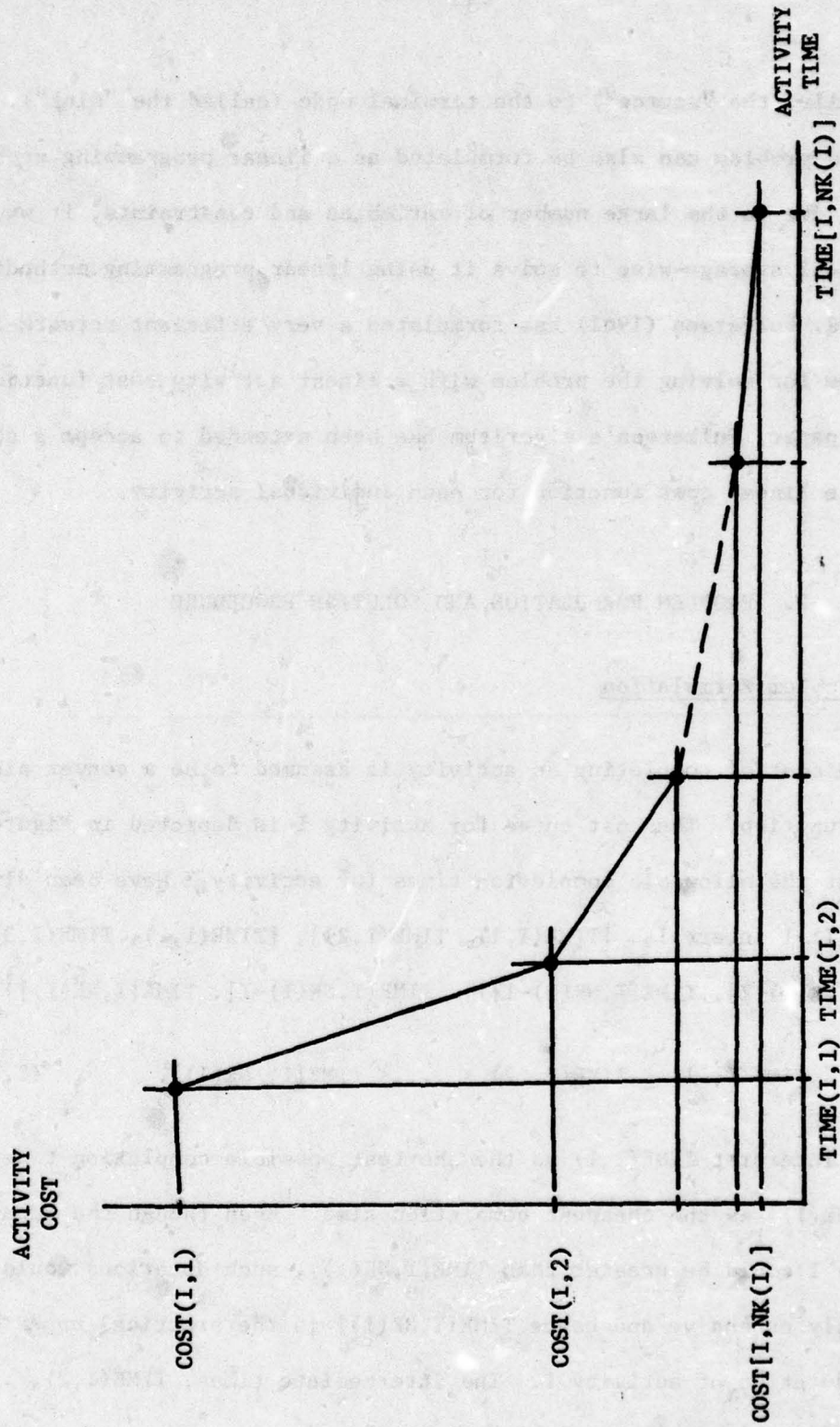


FIGURE 2

Breakpoints arise when there are alternative methods of performing an activity. These methods do not differ in the end result, but they do differ in the amount of time they take and their cost. For example, suppose that snow plows rent for a fixed \$200/day and cost a varying amount per hour to operate depending upon the speed at which they are operated. A corresponding activity cost curve might be as in Figure 3 where the "breakpoints" correspond to the use of different numbers of plows.

The cost for completing activity I in time $\text{TIME}(I,M)$ is $\text{COST}(I,M)$ which satisfies

$$\text{COST}(I,1) \geq \text{COST}(I,2) \geq \dots \geq \text{COST}[I,\text{NK}(I)]. \quad (2.2)$$

Furthermore, letting $C(I,M)$ represent the rate of decrease in the cost of activity I on the M^{th} interval implies

$$C(I,1) = \frac{\text{COST}(I,1) - \text{COST}(I,2)}{\text{TIME}(I,2) - \text{TIME}(I,1)}, \quad (2.3)$$

$$C[I,\text{NK}(I)-1] = \frac{\text{COST}[I,\text{NK}(I)-1] - \text{COST}[I,\text{NK}(I)]}{\text{TIME}[I,\text{NK}(I)] - \text{TIME}[I,\text{NK}(I)-1]}.$$

The convexity of the piecewise linear cost function implies that

$$C[I,\text{NK}(I)-1] \leq C[I,\text{NK}(I)-2] \leq \dots \leq C(I,1). \quad (2.3a)$$

Let $\text{XACT}(I)$ represent the duration time for activity I. This duration time, $\text{XACT}(I)$, can be decomposed as

$$\text{XACT}(I) = \sum_{M=1}^{\text{NK}(I)-1} \text{XACT}(I,M) \quad (2.4)$$

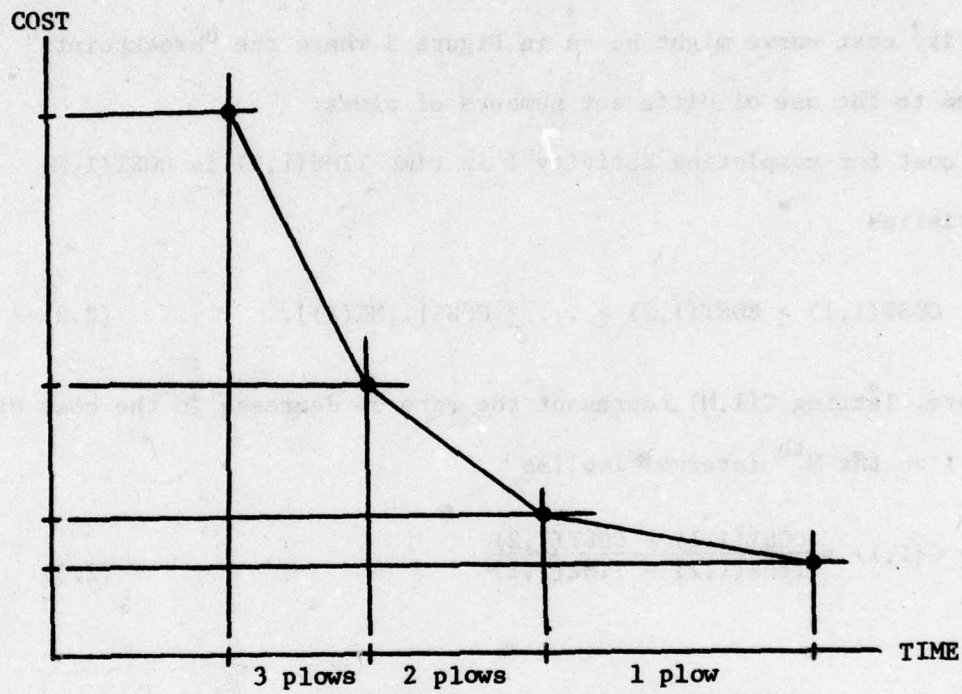


FIGURE 3

where

$$XACT(I,1) = \min[TIME(I,2), XACT(I)] \quad (2.5)$$

and for $M = 2, \dots, NK(I) - 1$

$$XACT(I,M) = \min\{TIME(I,M+1) - TIME(I,M), \max[0, XACT(I) - TIME(I,M)]\}. \quad (2.6)$$

For example, suppose that in Figure 4 $XACT(I) = 25$, then

$$XACT(I,1) = \min[TIME(I,2), XACT(I)]$$

$$= \min[10, 25]$$

$$= 10,$$

$$XACT(I,2) = \min\{TIME(I,3) - TIME(I,2), \max[0, XACT(I) - TIME(I,2)]\}$$

$$= \min\{20 - 10, \max[0, 25 - 10]\}$$

$$= 10,$$

$$XACT(I,3) = \min\{TIME(I,4) - TIME(I,3), \max[0, XACT(I) - TIME(I,3)]\}$$

$$= \min\{30 - 20, \max[0, 25 - 20]\}$$

$$= 5,$$

and

$$XACT(I) = \sum_{M=1}^3 XACT(I,M)$$

$$= 10 + 10 + 5$$

$$= 25.$$

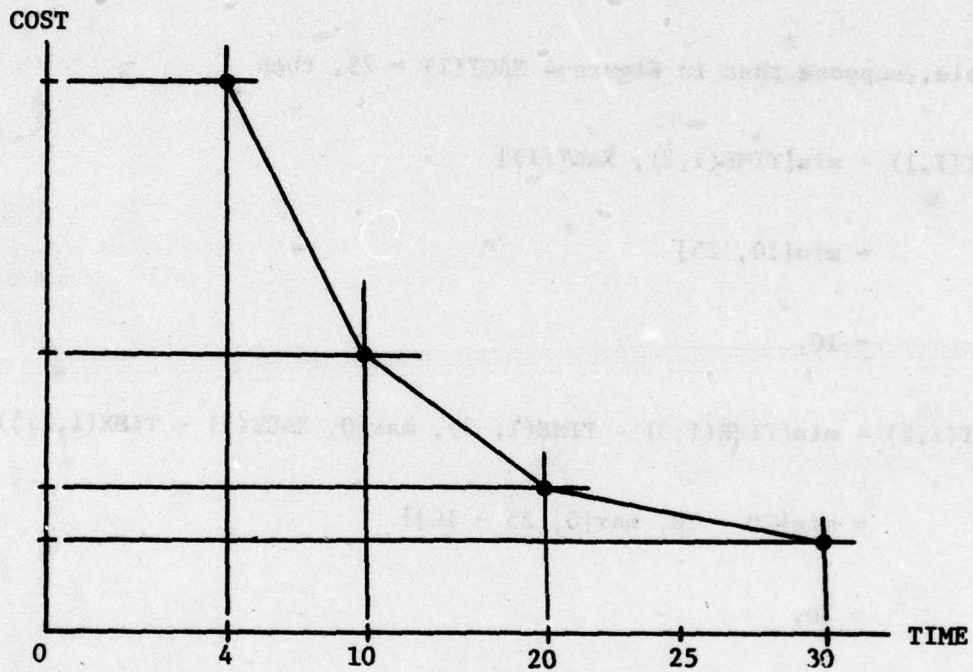


FIGURE 4

The total cost associated with duration time $XACT(I)$ for activity I is

$$KK(I) = \sum_{M=1}^{NK(I)-1} C(I,M)XACT(I,M) \quad (2.7)$$

where

$$KK(I) = COST(I,1) + C(I,1)TIME(I,1). \quad (2.8)$$

The total project cost is

$$\sum_I [KK(I) - \sum_{M=1}^{NK(I)-1} C(I,M)XACT(I,M)]. \quad (2.9)$$

Let the node time $XNODE(K)$ be the "length" of the longest path from the source node to node K when the "length" of an arc (activity) is its completion time. Thus, for example, in Figure 1

$$XNODE(1) = 0,$$

$$XNODE(2) = A,$$

$$XNODE(3) = \max(B, A + F),$$

$$XNODE(4) = \max(A + C, B + D, A + F + D), \text{ and}$$

$$XNODE(5) = \max(A + C + E, B + D + E, A + F + D + E).$$

If activity I originates at node O_I , terminates at node T_I , and takes $XACT(I)$ units of time, feasibility requires that

$$XNODE(O_I) + XACT(I) \leq XNODE(T_I). \quad (2.10)$$

Note that the time to complete the entire project is

$$XNODE(SINK) - XNODE(SOURCE).$$

In what follows

$$XNODE(SOURCE) \equiv 0 \quad (2.11)$$

without loss of generality.

The problem is to minimize the total project cost (2.9) subject to the condition that the project is completed by a specified time LAMBDA. This problem can now be formulated as

$$\min\{PCOST(LAMBDA) \equiv \sum_I [KK(I) - \sum_{M=1}^{NK(I)-1} C(I,M)XACT(I,M)]\} \quad (2.12)$$

subject to the constraints

$$XNODE(O_I) + \sum_{M=1}^{NK(I)-1} XACT(I,M) - XNODE(T_I) \leq 0, \text{ all } I, \quad (2.13)$$

$$XNODE(SINK) \leq LAMBDA, \quad (2.14)$$

$$XACT(I,M) \leq U(I,M), \quad \text{all } I \text{ and } M, \quad (2.15)$$

$$XACT(I,M) \geq L(I,M), \quad \text{all } I \text{ and } M, \quad (2.16)$$

where

$$U(I,M) = \begin{cases} \text{TIME}(I,2) & M = 1, \\ \text{TIME}(I,M+1) - \text{TIME}(I,M) & M = 2, \dots, NK(I)-1, \end{cases} \quad (2.17)$$

$$L(I,M) = \begin{cases} \text{TIME}(I,1) & M = 1, \\ 0 & M = 2, \dots, NK(I)-1, \end{cases} \quad (2.18)$$

O_I = the origin node of activity I,

T_I = the terminal node of activity I.

Since the addition or subtraction of a constant in the objective function does not change the problem, we can represent the objective function as

$$\max_I \sum_{M=1}^{NK(I)-1} C(I,M) XACT(I,M). \quad (2.19)$$

We shall solve this problem for all feasible values of LAMBDA. The minimum feasible value of LAMBDA, LMIN, is the length of the longest path from the source to the sink when the XACT(I)'s are at their lower bounds, XACT(I) = TIME(I,1) for all I. The maximum value of interest for LAMBDA, LMAX, is the length of the longest path from the source to the sink when the XACT(I)'s represent the cheapest practical times, XACT(I) = TIME[I,NK(I)] for all I. Thus, for a given LAMBDA such that

$$LMIN \leq LAMBDA \leq LMAX,$$

the constraints (2.13) - (2.16) are feasible. The proof for this and all other underlying theorems presented in the problem formulation and algorithm are found in Chapter 3, Section 2. We shall refer to the problem given in (2.13) - (2.19) as the Primal Problem.

In the Primal Problem, dummy activities may be assumed to have times and costs equal to zero.

2.2. The Dual Problem

The standard duality theory for linear programming implies that, if the primal problem has the form

$$\begin{aligned} &\max \quad c^T x \\ &\text{subject to the constraints} \\ &Ax \leq b, \end{aligned} \quad (2.20)$$

then the corresponding dual problem is

$$\begin{aligned} \min & b^T w \\ \text{subject to the constraints} \\ & A^T w = c \\ & w \geq 0, \end{aligned} \quad (2.21)$$

see for example Hadley (1962). Writing our Primal Problem in the form (2.20) implies that our dual problem can be written as

$$\min [\text{LAMBDA} \cdot V + \sum_{I,M} U(I,M) \cdot G(I,M) - \sum_{I,M} L(I,M) \cdot H(I,M)] \quad (2.22)$$

subject to the constraints

$$F(I) + G(I,M) - H(I,M) = C(I,M) \quad \text{all } I, M \quad (2.23)$$

$$\sum_{I \in J, I=K} F(I) - \sum_{I \in T, I=K} F(I) = \begin{cases} 0 & K = \text{node} \neq \text{SOURCE, SINK} \\ -V & K = \text{SINK,} \end{cases} \quad (2.24)$$

$$F(I), V, G(I,M), H(I,M) \geq 0. \quad (2.25)$$

Note that the coefficients in (2.21) of the s^{th} dual variable are the coefficients in the s^{th} primal constraint, so that, there is a natural one-to-one correspondence between primal constraints and dual variables.

The dual problem (2.22) - (2.25) can be interpreted as a flow problem for the project network. The dual variable, $F(I)$, associated with constraint (2.13) is the flow for the I^{th} activity. The constraint (2.24) implies that except for the source and the sink the flow going into a node equals the flow coming out of that node. Thus at all nodes other than the source and the sink we have conservation of flow. The total flow of the network is

$$V = \sum_{I \in T_I = \text{SINK}} F(I) - \sum_{I \in O_I = \text{SINK}} F(I) = \sum_{I \in T_I = \text{SINK}} F(I), \quad (2.26)$$

and V is the dual variable associated with constraint (2.14) for a fixed LAMBDA .

The dual variables $G(I, M)$ and $H(I, M)$ are associated with the upper and lower bounds for $XACT(I, M)$ respectively.

Rearranging (2.23), we have an equation of the form

$$g - h = c - f.$$

For a fixed value of f , we have $c - f = r$, say, and

$$g = h + r. \quad (2.27)$$

In (2.22) we want to minimize an expression of the form

$$Ug - Lh,$$

or equivalently using (2.27)

$$Ug - Lh = U(h + r) - Lh = Ur + h(U - L).$$

Since $g = h + r$ and both $g \geq 0$ and $h \geq 0$, making h as small as possible implies

$$h = \max(0, -r).$$

Correspondingly

$$g = h + r = \max(r, 0).$$

Thus

$$g = \max(0, c - f),$$

$$h = \max(0, f - c),$$

and correspondingly

$$G(I,M) = \max[0, C(I,M) - F(I)], \quad (2.28)$$

$$H(I,M) = \max[0, F(I) - C(I,M)]. \quad (2.29)$$

Using (2.28) and (2.29) the dual becomes

$$\begin{aligned} \min \{ & \text{LAMBDA} \cdot V + \sum_{I,M} U(I,M) \cdot \max[0, C(I,M) - F(I)] \\ & - \sum_{I,M} L(I,M) \cdot \max[0, F(I) - C(I,M)] \} \end{aligned} \quad (2.30)$$

subject to the constraints

$$\sum_{I \in O} F(I) - \sum_{I \in T} F(I) = \begin{cases} 0 & K = \text{node} \neq \text{SOURCE, SINK} \\ -V & K = \text{SINK,} \end{cases}$$

$$F(I), V \geq 0.$$

A key observation at this point is that for all (I,M)

$$U(I,M) \max[0, C(I,M) - F(I)] - L(I,M) \max[0, F(I) - C(I,M)] \quad (2.31)$$

is a convex piecewise linear function of F(I) as sketched in Figure 5.

The convexity of (2.31) follows from $U(I,M) \geq L(I,M)$. Furthermore, since the sum of convex piecewise linear functions is also a convex piecewise linear function, it follows that

$$\sum_M U(I,M) \max[0, C(I,M) - F(I)] - \sum_M L(I,M) \max[0, F(I) - C(I,M)] \quad (2.32)$$

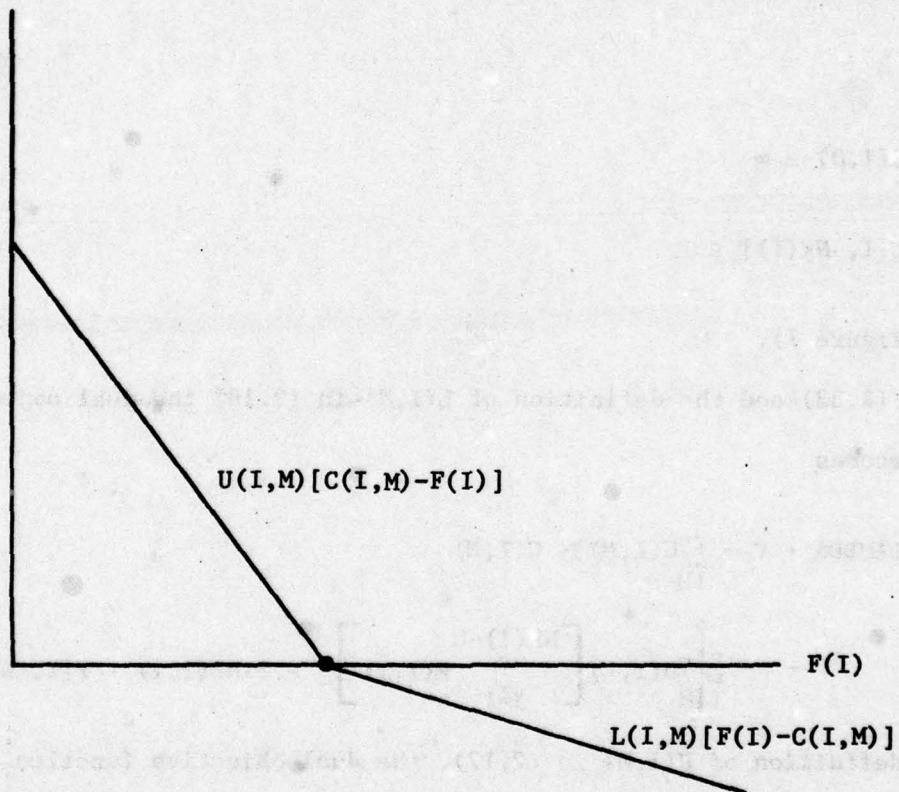


FIGURE 5

is a convex piecewise linear function of $F(I)$ as sketched in Figure 6.

The piecewise linear behavior of (2.32) suggests the following decomposition of $F(I)$:

$$F(I) = \sum_{J=1}^{NK(I)} F(I, J) \quad (2.33)$$

where

$$0 \leq F(I, J) \leq C[I, NK(I) - J] - C[I, NK(I) - J + 1] \quad (2.34)$$

and

$$C(I, 0) \equiv \infty$$

$$C[I, NK(I)] \equiv 0.$$

(also see Figure 7).

Using (2.33) and the definition of $L(I, M)$ in (2.18) the dual objective function becomes

$$\begin{aligned} & \text{LAMBDA} \cdot V + \sum_{I, M} U(I, M) \cdot C(I, M) \\ & - \sum_I \left\{ \sum_M U(I, M) \left[\sum_{J=1}^{NK(I)-M} F(I, J) \right] - \text{TIME}(I, 1) \cdot F[I, NK(I)] \right\}. \end{aligned}$$

Using the definition of $U(I, M)$ in (2.17), the dual objective function can be further simplified to

$$\text{LAMBDA} \cdot V + \sum_{I, M} U(I, M) \cdot C(I, M) - \sum_{I, J} \text{TIME}[I, NK(I)+1-J] F(I, J). \quad (2.35)$$

Since $\sum_{I, M} U(I, M) C(I, M)$ is a constant, (2.35) is equivalent to

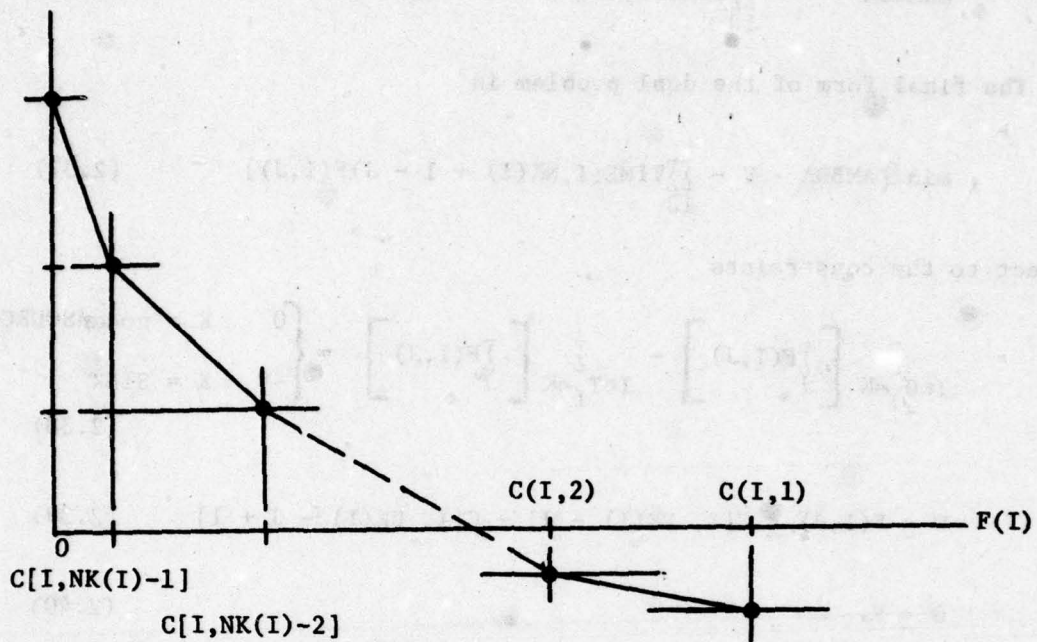


FIGURE 6

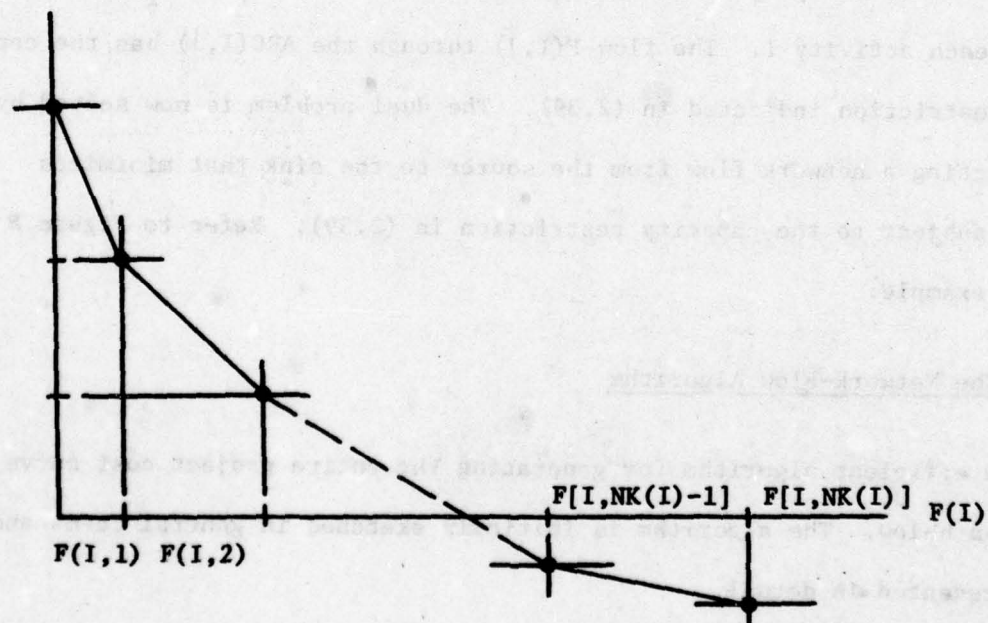


FIGURE 7

$$\text{LAMBDA} \cdot V - \sum_{I,J} \text{TIME}(I, \text{NK}(I) + 1 - J) F(I, J). \quad (2.36)$$

The final form of the dual problem is

$$\min \{ \text{LAMBDA} \cdot V - \sum_{I,J} \text{TIME}(I, \text{NK}(I) + 1 - J) F(I, J) \} \quad (2.37)$$

subject to the constraints

$$\sum_{I \in 0_I = K} \left[\sum_J F(I, J) \right] - \sum_{I \in T_I = K} \left[\sum_J F(I, J) \right] = \begin{cases} 0 & K = \text{node} \neq \text{SOURCE}, \text{SINK} \\ -V & K = \text{SINK} \end{cases} \quad (2.38)$$

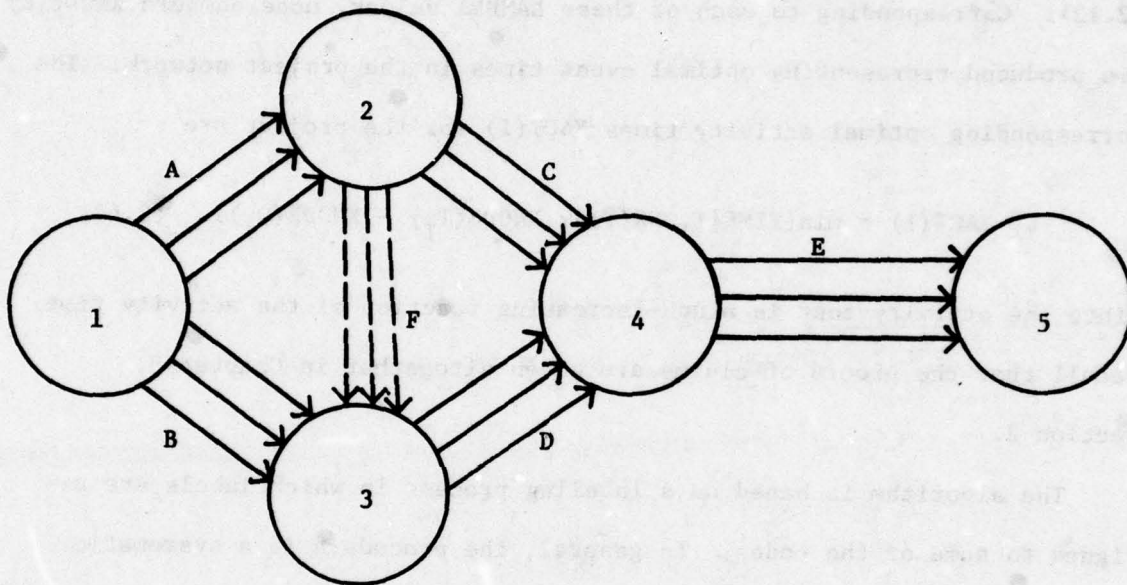
$$0 \leq F(I, J) \leq C[I, \text{NK}(I) - J] - C[I, \text{NK}(I) - J + 1] \quad (2.39)$$

$$0 \leq V. \quad (2.40)$$

This dual problem can be solved as a network flow problem. The original project network is enlarged by adding one arc, say $\text{ARC}(I, J)$, for each $F(I, J)$ so that the project network has $\text{NK}(I)$ arcs from 0_I to T_I corresponding to each activity I . The flow $F(I, J)$ through the $\text{ARC}(I, J)$ has the capacity restriction indicated in (2.39). The dual problem is now solved by constructing a network flow from the source to the sink that minimizes (2.36) subject to the capacity restriction in (2.39). Refer to Figure 8 for an example.

2.3. The Network-Flow Algorithm

An efficient algorithm for generating the entire project cost curve is given below. The algorithm is initially sketched in general terms and then presented in detail.



Example: $NK(I) = 3$, all I .

FIGURE 8

2.3.1. The Sketch

The algorithm starts with the largest LAMBDA of interest, LMAX, and sequentially determines the LAMBDA corresponding to each breakpoint of the convex piecewise linear project cost function PCOST(LAMBDA) defined in (2.12). Corresponding to each of these LAMBDA values, node numbers XNODE(K) are produced representing optimal event times in the project network. The corresponding optimal activity times XACT(I) for the project are

$$XACT(I) = \min\{TIME[I, NK(I)], XNODE(T_I) - XNODE(0_I)\} \quad (2.41)$$

since the activity cost is a non-increasing function of the activity time. Recall that the proofs of claims are given altogether in Chapter 3, Section 2.

The algorithm is based on a labeling process in which labels are assigned to some of the nodes. In general, the procedure is a systematic search for a path from the source to the sink having certain desired properties. Flow along this path may travel through arcs either in the same direction as their orientation or in the opposite direction. Such flows will be called forward and reverse flows respectively. Roughly speaking, a reverse flow is really only a reversal or re-routing of earlier flow in the forward direction. No net flow in the reverse direction is allowed.

The labeling process is started with a feasible and optimal solution to the primal and dual problems for LAMBDA = LMAX. The initial node times are found by setting the activity times equal to their upper bounds. These initial XNODE(K)'s and the initial flow - $F(I,J) = 0$ for all (I,J) - satisfy the following properties:

$$ABAR(I,J) < 0 \Rightarrow F(I,J) = 0, \text{ and} \quad (2.42)$$

$$ABAR(I,J) > 0 \Rightarrow F(I,J) = C[I, NK(I) - J] - C[I, NK(I) - J + 1] \quad (2.43)$$

where

$$ABAR(I,J) \equiv TIME[I, NK(I) + 1 - J] + XNODE(0_I) - XNODE(T_I). \quad (2.44)$$

Note that no restrictions are placed on $F(I,J)$ when $ABAR(I,J) = 0$. Henceforth, the properties (2.42) and (2.43) will be referred to as the "optimality properties" for $LAMBDA = XNODE(SINK)$. These optimality properties imply that complementary slackness holds and that the flow $F(I,J)$ minimizes (2.36).

The labeling process has been divided into two parts called the first and second labelings, respectively. In both of these procedures we have freedom to label with respect to complementary slackness since we work exclusively with arcs having $ABAR(I,J) = 0$. The first labeling seeks a path from the source to the sink composed of infinite capacity arcs, i.e. those corresponding to $J = NK(I)$. If such a path is found, the algorithm terminates since the Primal Problem will be infeasible if the current value of $LAMBDA$ is decreased. If no such path is found, we go on to the second labeling in which we search for a path from the source to the sink having the following desired properties: for all forward arcs of the path $ABAR(I,J) = 0$ and $F(I,J)$ is less than its upper bound in (2.39); for all reverse arcs of the path $ABAR(I,J) = 0$ and $F(I,J) > 0$. If at the end of the second labeling the sink has been labeled, we say "breakthrough" has occurred.

If breakthrough occurs, then the minimum arc capacity along the path is determined, say $CAP(SINK)$. The old flow $F(I,J)$ is changed by adding $CAP(SINK)$ to the amount of all forward flows on the path and by subtracting $CAP(SINK)$ from the amount of all reverse flows on the path. This new flow still satisfies the optimality properties and is interpreted as an alternate optimal dual solution for the current $LAMBDA = XNODE(SINK)$. On the other hand, if the sink has not been labeled at the end of the second labeling, we say "nonbreakthrough" has occurred. When this happens, the old dual variables are optimal for the old primal problem and no new alternate dual solution can be found. In this case the node numbers $XNODE(K)$'s are changed by subtracting a positive quantity DEL from all $XNODE(K)$ corresponding to unlabeled nodes K . This does not change $XNODE(SOURCE) = 0$ but reduces $XNODE(SINK) = LAMBDA$ by DEL . Through (2.41), these new node times imply a set of optimal activity times for the new $LAMBDA$ where

$$\text{new } LAMBDA = \text{old } LAMBDA - DEL.$$

The definition of DEL guarantees that the new $XNODE(K)$'s and the old $F(I,J)$'s still satisfy the optimality properties. Hence, when nonbreakthrough occurs, we have identified the point on the project cost curve corresponding to the new $LAMBDA$.

The second labeling can terminate only in breakthrough or nonbreakthrough. After either of these, the entire labeling process is repeated.

2.3.2. The Details

Initially, the algorithm sets each activity time to its smallest (most expensive) feasible value and determines the corresponding

minimum feasible project completion time (deadline time LMIN). Then, the algorithm sets each activity completion time to its largest (cheapest) feasible value and determines the corresponding minimum project cost and maximum completion time of interest (deadline time LMAX).

The iterative procedure is begun with the node times XNODE(K) corresponding to all activity completion times at their largest (cheapest) values and all flows F(I,J) equal to zero. These node times and flows satisfy the optimality properties.

A. Labeling Process. During this routine, a node is considered to be in one of three states: unlabeled, labeled and unscanned, or labeled and scanned. Initially all nodes are unlabeled.

In general, a node label has four parts [A, B, C, D] when the node is being labeled because it is at "the other end" of an arc associated with some F(I,J). If "the other end" is the terminal node T_I , then the label contains

$A = O_I$, $B = J$, $D =$ maximum allowable flow, and
 $C = 0$ [denoting that flow will be in the forward direction
 $(O_I \rightarrow T_I)$].

If "the other end" is the origin O_I , then the label contains $A = T_I$, $B = J$, $D =$ maximum allowable flow, and $C = 1$ [denoting that flow will be in the reverse direction $(T_I \rightarrow O_I)$].

1. First Labeling. Assign the source node the label [-, -, -, CAP(SOURCE) = ∞]. In general, select any labeled, unscanned node, say node n, and search for all unlabeled nodes T_I such that $n = O_I$ and ARC[I, NK(I)] is an arc with

$$ABAR[I, NK(I)] = 0. \quad (2.45)$$

Label such nodes T_I with $[O_I, NK(I), 0, CAP(T_I) = \infty]$. Such T_I 's are now labeled and unscanned, and node n is labeled and scanned. Repeat this step until either the sink node is labeled and unscanned, or no more nodes can be labeled and the sink node is unlabeled. In the former case, terminate the algorithm. In the latter case, go on to the Second Labeling.

2. Second Labeling. Nodes that were labeled from the First Labeling retain their labels. However, all nodes revert back to an unscanned state. The general step is to select any labeled, unscanned node, say n .

(i) Scan n for all unlabeled nodes T_I such that $n = O_I$. For each such node T_I find the J (if one exists) such that both

$$ABAR(I, J) = 0 \quad (2.46)$$

and

$$F(I, J) < C[I, NK(I) - J] - C[I, NK(I) - J + 1]; \quad (2.47)$$

then assign node T_I the label $[O_I, J, 0, CAP(T_I)]$ where

$$CAP(T_I) = \min\{CAP(O_I), C[I, NK(I) - J] - C[I, NK(I) - J + 1] - F(I, J)\} \quad (2.48)$$

so that T_I is now labeled and unscanned. If no such J exists, the node T_I is not labeled.

(ii) Scan n for all unlabeled nodes O_I such that $n = T_I$. For each such node O_I find the J (if one exists) such that both

$$ABAR(I, J) = 0 \quad (2.49)$$

and

$$F(I,J) > 0; \quad (2.50)$$

then assign node O_I the label $[T_I, J, 1, CAP(O_I)]$ where

$$CAP(O_I) = \min[CAP(T_I), F(I,J)] \quad (2.51)$$

so that O_I is now labeled and unscanned. If no such J exists, the node O_I is not labeled.

Repeat the general step until either the sink node is labeled and unscanned (breakthrough), or no more nodes can be labeled and the sink node is unlabeled (nonbreakthrough). If breakthrough occurs, go on to routine B; if nonbreakthrough occurs, go to routine C.

B. Flow Change. The labeling process has resulted in breakthrough. The sink node will have a label of the form $[O_I, J, 0, CAP(SINK)]$. The total network flow will now be increased by $CAP(SINK)$. The flows are updated as follows. Add $CAP(SINK)$ to $F(I,J)$; then go on to node $n = O_I$ and its label. The general step for node n depends on its label and is:

1. Label = $[O_I, J, 0, CAP(T_I)]$. Add $CAP(SINK)$ to $F(I,J)$ since this additional flow along $ARC(I,J)$ will be forward flow from O_I to $n = T_I$. The next node to consider is $n = O_I$.
2. Label = $[T_I, J, 1, CAP(O_I)]$. Subtract $CAP(SINK)$ from $F(I,J)$ since this additional flow along $ARC(I,J)$ will be a reversal of previous flow from $n = O_I$ to T_I . The next node to consider is $n = T_I$.

This iterative procedure is continued until $n = SOURCE$. At this point a path from the source to the sink has been retraced working backwards

from the sink. The arcs on this path that are traversed in the forward direction ($0_I \rightarrow T_I$) as we go from the source to the sink have their flows increased by $CAP(SINK)$ while the arcs on this path that are traversed in the reverse direction ($T_I \rightarrow 0_I$) have their flows decreased by $CAP(SINK)$.

All labels are now discarded and the labeling process (A) is started over.

C. Node Number Change. The labeling process has resulted in non-breakthrough. The following subsets of arcs are determined:

$$A_1 = \{(I,J) | 0_I \text{ labeled, } T_I \text{ unlabeled, } ABAR(I,J) < 0\}, \quad (2.52)$$

$$A_2 = \{(I,J) | 0_I \text{ unlabeled, } T_I \text{ labeled, } ABAR(I,J) > 0\}. \quad (2.53)$$

We now define

$$DELTA1 = \min_{A_1} [-ABAR(I,J)], \quad (2.54)$$

$$DELTA2 = \min_{A_2} [ABAR(I,J)], \quad (2.55)$$

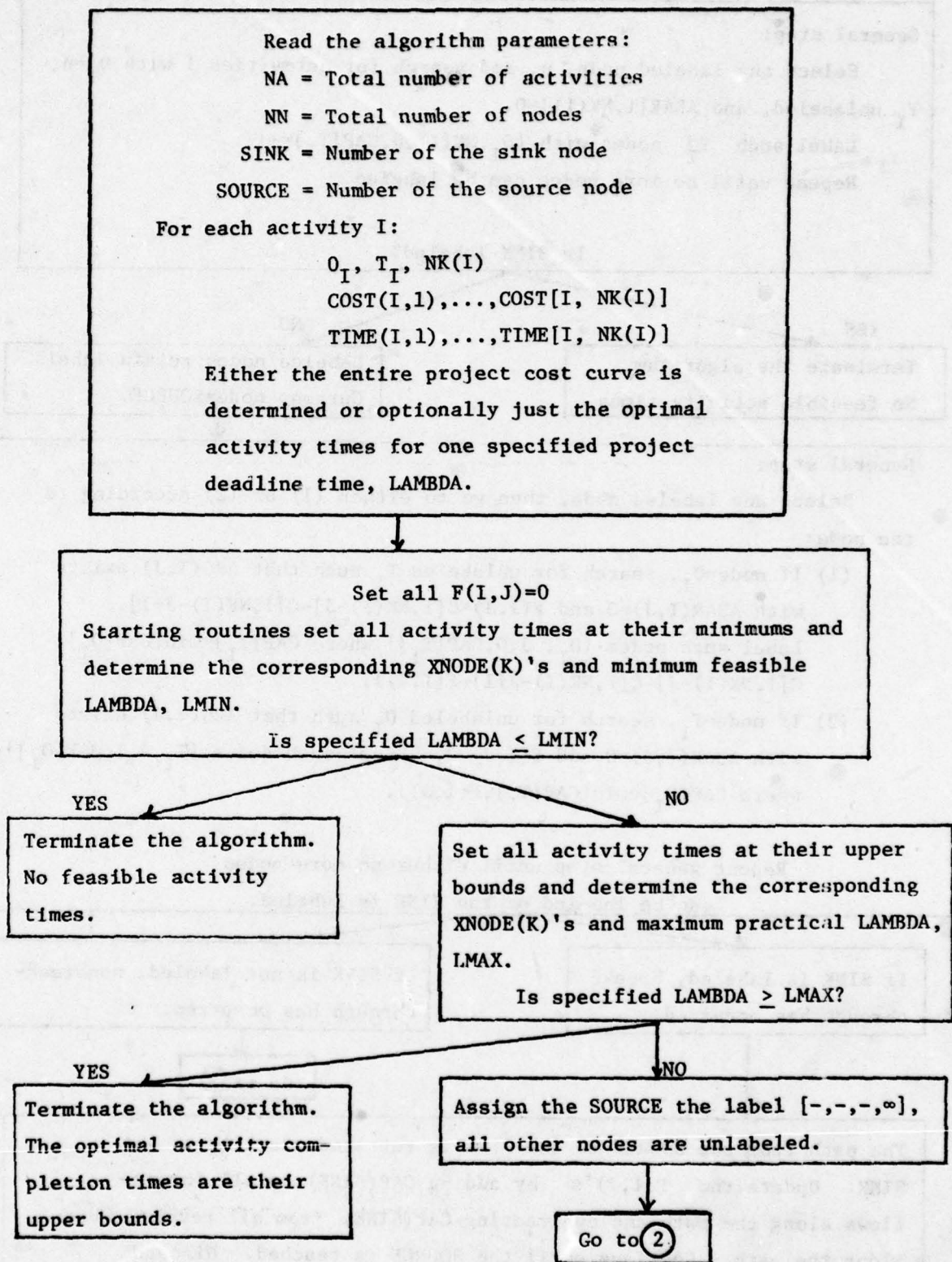
$$DEL = \min(DELTA1, DELTA2). \quad (2.56)$$

The node numbers $XNODE(K)$ are changed by subtracting DEL from all $XNODE(K)$ corresponding to unlabeled K . All labels are discarded and the labeling process (A) is started over.

2.4. Flowchart of the Algorithm

See Figure 9.

FIGURE 9: Flowchart of the Algorithm



2

General step:

Select any labeled node, n , and search for activities I with $O_I = n$, T_I unlabeled, and $ABAR[I, NK(I)] = 0$.

Label such T_I nodes with $\{O_I, NK(I), 0, CAP[T_I] = \infty\}$

Repeat until no more nodes can be labeled.

Is SINK labeled?

YES

Terminate the algorithm.

No feasible activity times.

NO

Labeled nodes retain labels.

Current node = SOURCE.

General step:

Select any labeled node, then go to either (1) or (2) according to the node:

(1) If node = O_I , search for unlabeled T_I such that $ARC(I, J)$ exists with $ABAR(I, J) = 0$ and $F(I, J) < C[I, NK(I) - J] - C[I, NK(I) - J + 1]$.

Label such nodes $\{O_I, J, 0, CAP[T_I]\}$ where $CAP[T_I] = \min\{CAP[O_I], C[I, NK(I) - J] - C[I, NK(I) - J + 1] - F(I, J)\}$.

(2) If node = T_I , search for unlabeled O_I such that $ARC(I, J)$ exists with $ABAR(I, J) = 0$ and $F(I, J) > 0$. Label such nodes $\{T_I, J, 1, CAP[O_I]\}$ where $CAP[O_I] = \min\{CAP[T_I], F(I, J)\}$.

Repeat general step until either no more nodes can be labeled or the SINK is labeled.

If SINK is labeled, break-through has occurred.

If SINK is not labeled, nonbreak-through has occurred.

Go to ③

The path from the SOURCE to the SINK is retraced starting at the SINK. Update the $F(I, J)$'s by adding $CAP(SINK)$ to all forward flows along the path and subtracting $CAP(SINK)$ from all reverse flows along the path. Continue until the SOURCE is reached. Discard labels and return to ①.

3

Find the following subsets:

A_1 : $\{(I,J) \text{ such that } O_I \text{ is labeled, } T_I \text{ is unlabeled, and } ABAR(I,J) < 0\}$

A_2 : $\{(I,J) \text{ such that } O_I \text{ is unlabeled, } T_I \text{ is labeled, and } ABAR(I,J) > 0\}$

$DELTA1 = \min_{A_1} [-ABAR(I,J)]$

$DELTA2 = \min_{A_2} [ABAR(I,J)]$

$DEL = \min[DELTA1, DELTA2].$

Subtract DEL from all unlabeled nodes $XNODE(K)$. Then the

$XACT(I) = \min\{TIME[I, NK(I)], XNODE(T_I) - XNODE(O_I)\}$

are an alternative optimal solution for the current LAMBDA and also an optimal solution for new LAMBDA = current LAMBDA - DEL.

A new point on the project cost curve has been determined.

Is new LAMBDA $\leq$ specified LAMBDA?

YES

Terminate the algorithm.
Desired solution found.

NO

Discard labels and return to

1.

3. VERIFICATION OF CLAIMS

The algorithm described in the previous chapter is based on many claims. The lemmas and theorems given below prove these claims and, at the same time, show that the given algorithm does indeed yield for each project deadline time LAMBDA the individual activity completion times which minimize the total project cost.

3.1. Summary

The initial primal and dual variables $XACT(I)$, $XNODE(K)$ and $F(I,J)$ provide a feasible and optimal solution for the largest LAMBDA of interest, LMAX (Lemmas 1, 5 and 6). The changes applied to these variables arise from either breakthrough or nonbreakthrough and force the variables to remain feasible and satisfy the optimality properties throughout the algorithm (Lemmas 2, 3, 4, 5, 6 and Theorem 1). The optimality properties imply that complementary slackness holds which, combined with feasibility, implies that the solution is optimal for a given LAMBDA (Lemma 7 and Theorem 2).

The algorithm itself terminates after a finite number of applications of the labeling procedure (Theorem 3). At the conclusion of the computations a path from the source to the sink has been identified in the First Labeling step such that along this path

$$TIME(I,1) + XNODE(O_I) = XNODE(T_I).$$

Since $(TIME(I,1))$ is the minimum feasible completion time for activity I, this means that the minimum possible time to complete the sequence of activities along this path is $XNODE(SINK) = LAMBDA$. Hence any further

decrease in LAMBDA would make the Primal Problem infeasible; i.e., the project cannot be completed in any shorter time.

The project cost function PCOST(LAMBDA) is convex and is linear between the successively determined values of LAMBDA generated in the computations (Lemmas 8 and 9). Given two successively determined values of LAMBDA, say L_1 and $L_2 = L_1 - \text{DEL}$, the optimal node times and activity completion times for any project deadline time L between L_1 and L_2 are

$$\begin{aligned} \text{XNODE}_L(K) &= \begin{cases} \text{XNODE}_{L_1}(K) & \text{if } K \text{ labeled when } \text{LAMBDA}=L_1, \\ \text{XNODE}_{L_1}(K) - (L_1 - L) & \text{if } K \text{ unlabeled when } \text{LAMBDA}=L_1, \end{cases} \\ \text{XACT}_L(I) &= \min\{\text{TIME}[I, \text{NK}(I)], \text{XNODE}_L[T_I] - \text{XNODE}_L[0_I]\} \end{aligned}$$

where the subscript L_1 implies $\text{LAMBDA} = L_1$ (Theorem 4).

One additional feature of the algorithm is that, if the problem is specified in terms of integers, then the breakpoints of the project cost curve PCOST(LAMBDA) and the corresponding optimal activity times will all be integers.

3.2. Proofs

Lemma 1: The original set of node integers $\text{XNODE}(K)$ and the zero flow $F(I, J)$ satisfy the optimality properties. Furthermore, this $F(I, J)$ minimizes (2.36) implying an optimal solution for $\text{LAMBDA} = \text{LMAX}$.

Proof: In a starting routine the activity times $\text{XACT}(I)$ are set to their largest feasible (cheapest) values. Then the node times $\text{XNODE}(K)$ are set to their corresponding smallest feasible values. This implies that

$$\text{TIME}[I, \text{NK}(I)] \leq \text{XNODE}(T_I) - \text{XNODE}(O_I)$$

or equivalently

$$\text{TIME}[I, \text{NK}(I)] + \text{XNODE}(O_I) - \text{XNODE}(T_I) \leq 0.$$

Thus all $\text{ABAR}(I, J) \leq 0$. Finally, since all $\text{ABAR}(I, J) \leq 0$ and $F(I, J) = 0$, the optimality properties are satisfied.

The dual objective function is

$$\begin{aligned} \text{LAMBDA} \cdot V &= \sum_{I, J} \text{TIME}(I, \text{NK}(I) + 1 - J) \cdot F(I, J) \\ &= -\left[\sum_{I, J} \text{TIME}(I, \text{NK}(I) + 1 - J) \cdot F(I, J) - \text{LAMBDA} \cdot V \right] \\ &= -\left[\sum_{I, J} \text{TIME}(I, \text{NK}(I) + 1 - J) \cdot F(I, J) + [\text{XNODE}(\text{SOURCE}) \right. \\ &\quad \left. - \text{XNODE}(\text{SINK})] \cdot V \right] \\ &= -\left[\sum_{I, J} \text{TIME}(I, \text{NK}(I) + 1 - J) \cdot F(I, J) + \sum_{I, J} [\text{XNODE}(O_I) \right. \\ &\quad \left. - \text{XNODE}(T_I)] \cdot F(I, J) \right] \\ &= -\left[\sum_{I, J} \text{ABAR}(I, J) \cdot F(I, J) \right]. \end{aligned}$$

Thus, since all $\text{ABAR}(I, J) \leq 0$, $F(I, J) = 0$ is optimal. QED.

Lemma 2: If breakthrough occurs, the old node numbers and the new flow satisfy the optimality properties.

Proof: The node numbers $\text{XNODE}(K)$ do not change. The new flows are obtained by adding the positive number $\text{CAP}(\text{SINK})$ to all $F(I, J)$ corresponding to forward arcs of the path from the source to the sink, and subtracting $\text{CAP}(\text{SINK})$ from all $F(I, J)$ corresponding to reverse arcs of the path.

Flow changes occur only in arcs for which $ABAR(I,J) = 0$. No restriction is imposed on the $F(I,J)$'s in the optimality properties when $ABAR(I,J) = 0$. Thus, the old $XNODE(K)$'s and the new $F(I,J)$'s still satisfy the optimality properties. QED.

Lemma 3: If nonbreakthrough occurs, the node number change, DEL, is a well-defined positive number.

Proof: For DEL to be well-defined, at least one of the sets of arcs A_1, A_2 (as defined in equations (2.52) and (2.53)) is non-empty.

Suppose A_1 were empty. Since there is a path from the source to the sink in the project network, and since the source is labeled and the sink is unlabeled, there must be a set of arcs $\{ARC(I,J), J = 1, \dots, NK(I)\}$ in the enlarged network with O_I labeled and T_I unlabeled. The definition of A_1 implies that if A_1 is empty, then $ABAR(I,J) \geq 0$ for this set of arcs. From labeling rules (2.46) and (2.47), if $ABAR(I,J) = 0$ then $F(I,J)$ cannot be less than $\{C[I, NK(I) - J] - C[I, NK(I) - J + 1]\}$, otherwise T_I would have been labeled from O_I . From (2.43), if $ABAR(I,J) > 0$, this implies that $F(I,J) = C[I, NK(I) - J] - C[I, NK(I) - J + 1]$. Hence we have $F[I, NK(I)] = \infty$. But this $F[I, NK(I)]$ is part of the actual flow through the network and, if it equals infinity, the first labeling process would have terminated the algorithm. Since this has not happened, there are no infinite flows and A_1 is non-empty.

By definition, DEL is always positive. QED.

Lemma 4: If nonbreakthrough occurs, for any DEL' satisfying $0 \leq DEL' \leq DEL$, the new node numbers

$$XNODE'(K) = \begin{cases} XNODE(K) & \text{for } K \text{ labeled,} \\ XNODE(K) - DEL' & \text{for } K \text{ unlabeled,} \end{cases}$$

and the old flow $F(I,J)$ still satisfy the optimality properties.

Proof: The new $ABAR'(I,J) = TIME(I, NK(I) + 1 - J) + XNODE'(0_I) - XNODE'(T_I)$.

(i) Suppose $ABAR'(I,J) < 0$. Then $F(I,J) = 0$ because of the following:

(a) If $ABAR(I,J) < 0$, then $F(I,J) = 0$ by (2.42).

(b) If $ABAR(I,J) = 0$, then

$$TIME(I, NK(I) + 1 - J) + XNODE(0_I) - XNODE(T_I) = 0,$$

or equivalently

$$TIME(I, NK(I) + 1 - J) = -XNODE(0_I) + XNODE(T_I);$$

so that

$$ABAR'(I,J) = TIME(I, NK(I) + 1 - J) + XNODE'(0_I) - XNODE'(T_I) < 0,$$

implies

$$-XNODE(0_I) + XNODE(T_I) + XNODE'(0_I) - XNODE'(T_I) < 0,$$

and finally

$$XNODE'(0_I) - XNODE(0_I) < XNODE'(T_I) - XNODE(T_I);$$

but this can happen only when 0_I is unlabeled and T_I is labeled. Hence,

if $ABAR(I,J) = 0$, then by labeling rules (2.49) and (2.50), $F(I,J) = 0$,

otherwise 0_I would be labeled from T_I .

(c) If $ABAR(I,J) > 0$, then

$$TIME(I, NK(I) + 1 - J) + XNODE(0_I) - XNODE(T_I) > 0,$$

or equivalently

$$TIME(I, NK(I) + 1 - J) > -XNODE(0_I) + XNODE(T_I);$$

so that

$$ABAR'(I,J) = TIME(I,NK(I) + 1 - J) + XNODE'(0_I) - XNODE'(T_I) < 0,$$

implies

$$TIME(I,NK(I) + 1 - J) < XNODE'(T_I) - XNODE'(0_I),$$

and

$$XNODE'(T_I) - XNODE'(0_I) > -XNODE(0_I) + XNODE(T_I),$$

and finally

$$XNODE'(T_I) - XNODE(T_I) > XNODE'(0_I) - XNODE(0_I).$$

Again, this can happen only when 0_I is unlabeled and T_I is labeled. But then the arc $ARC(I,J)$ is in A_2 and $DEL \leq ABAR(I,J)$. This would imply that

$$\begin{aligned} ABAR'(I,J) &= TIME(I,NK(I) + 1 - J) + XNODE(0_I) - DEL - XNODE(T_I) \\ &= ABAR(I,J) - DEL \\ &\geq 0. \end{aligned}$$

which contradicts the assumption $ABAR'(I,J) < 0$. Hence this case cannot occur.

(ii) Suppose $ABAR'(I,J) = 0$. There are no restrictions on $F(I,J)$ so the optimality properties still hold.

(iii) Suppose $ABAR'(I,J) > 0$. Then

$$F(I,J) = C[I, NK(I) - J] - C[I, NK(I) - J + 1]$$

because of the following:

(a) If $ABAR(I,J) > 0$, $F(I,J) = C[I, NK(I) - J] - C[I, NK(I) - J + 1]$ by (2.43).

(b) If $ABAR(I,J) = 0$, then

$$ABAR(I,J) < ABAR'(I,J)$$

or equivalently

$$\text{TIME}(I, \text{NK}(I) + 1 - J) + \text{XNODE}(O_I) - \text{XNODE}(T_I)$$

$$< \text{TIME}(I, \text{NK}(I) + 1 - J) + \text{XNODE}'(O_I) - \text{XNODE}'(T_I);$$

so that

$$\text{XNODE}(O_I) - \text{XNODE}'(O_I) < \text{XNODE}(T_I) - \text{XNODE}'(T_I).$$

This can happen only if O_I is labeled and T_I is unlabeled. Hence, by labeling rules (2.46) and (2.47)

$$F(I, J) = C[I, \text{NK}(I) - J] - C[I, \text{NK}(I) - J + L],$$

otherwise T_I would be labeled from O_I .

(c) If $\text{ABAR}(I, J) < 0$, then

$$\text{ABAR}(I, J) < \text{ABAR}'(I, J),$$

and again

$$\text{XNODE}(O_I) - \text{XNODE}'(O_I) < \text{XNODE}(T_I) - \text{XNODE}'(T_I).$$

This can happen only if O_I is labeled and T_I is unlabeled. But then the arc $\text{ARC}(I, J)$ is in A_I and $\text{DEL} \leq -\text{ABAR}(I, J)$ which would imply that

$$\begin{aligned} \text{ABAR}'(I, J) &= \text{TIME}(I, \text{NK}(I) + 1 - J) + \text{XNODE}'(O_I) - \text{XNODE}'(T_I) \\ &= \text{TIME}(I, \text{NK}(I) + 1 - J) + \text{XNODE}(O_I) - \text{XNODE}(T_I) + \text{DEL} \\ &= \text{ABAR}(I, J) + \text{DEL} \\ &\leq 0. \end{aligned}$$

This contradicts the assumption $\text{ABAR}'(I, J) > 0$. Hence this case cannot occur.

Cases (i) - (iii) together imply that the new node numbers and the old flow still satisfy the optimality properties. QED.

Theorem 1: The optimality properties (2.42) and (2.43) are maintained throughout the algorithm.

Proof: From Lemma 1, we see that the initial node numbers $XNODE(K)$'s and the zero flow provide an optimal solution for $LAMBDA = LMAX$. If breakthrough occurs, we see that the new $F(I,J)$'s are still optimal (Lemma 2). If nonbreakthrough occurs, we have a well-defined positive number DEL with which to update the $XNODE(K)$'s (Lemma 3) and, from Lemma 4, these updated values satisfy the optimality properties. QED.

Lemma 5: The starting values of the $XNODE(K)$'s and $XACT(I)$'s are feasible and remain feasible throughout the algorithm.

Proof: The starting values are found by an algorithm that sets the $XACT(I)$'s to their largest feasible times, $TIME[I, NK(I)]$. Correspondingly the $XACT(I,M)$'s are set equal to their upper bounds and hence (2.15) and (2.16) are satisfied. Then the algorithm sets $XNODE(K)$ equal to the length of the longest path from the source to node K , which implies that (2.13) is satisfied. We also define $XNODE(SOURCE) \equiv 0$ and $LAMBDA = XNODE(SINK)$; hence (2.14) is satisfied and the initial values are feasible.

If breakthrough occurs, the $XNODE(K)$'s and $XACT(I)$'s are not changed and hence remain feasible.

If nonbreakthrough occurs, the labeled $XNODE(K)$'s are unchanged, and the unlabeled $XNODE(K)$'s are updated by subtracting DEL , determined by (2.56). Then

$$\text{new } LAMBDA \equiv XNODE(SINK) - DEL$$

so (2.14) is satisfied.

(1) Suppose both O_I and T_I are labeled for activity I . Then neither these nodes nor $XACT(I)$ are updated and hence $XACT(I)$ remains

feasible.

(ii) Suppose both O_I and T_I are unlabeled for activity I. Then

$$\text{new XNODE}(O_I) = \text{old XNODE}(O_I) - \text{DEL},$$

$$\text{new XNODE}(T_I) = \text{old XNODE}(T_I) - \text{DEL}, \text{ and}$$

$$\text{new XACT}(I) = \min\{\text{TIME}[I, \text{NK}(I)], \text{old XNODE}(T_I) - \text{DEL}$$

$$- \text{old XNODE}(O_I) + \text{DEL}\}$$

$$= \text{old XACT}(I)$$

$$\leq \text{old XNODE}(T_I) - \text{old XNODE}(O_I)$$

$$= \text{new XNODE}(T_I) - \text{new XNODE}(O_I),$$

or equivalently

$$\text{new XACT}(I) + \text{new XNODE}(O_I) - \text{new XNODE}(T_I) \leq 0;$$

so that (2.13) is satisfied. Since $\text{XACT}(I)$ has not changed, (2.15) and (2.16) are still satisfied. Therefore, in this case, feasibility is maintained.

(iii) Suppose O_I is labeled and T_I is unlabeled for activity I.

Then

$$\text{new XNODE}(T_I) = \text{old XNODE}(T_I) - \text{DEL},$$

$$\text{new XNODE}(O_I) = \text{old XNODE}(O_I), \text{ and}$$

$$\text{new XACT}(I) = \min\{\text{TIME}[I, \text{NK}(I)], \text{old XNODE}(T_I) - \text{DEL}$$

$$- \text{old XNODE}(O_I)\};$$

so that (2.13) and (2.15) are satisfied. The lower bound constraint, (2.16), is also satisfied because of the following:

(a) Suppose $ABAR[I, NK(I)] < 0$. Then since O_I is labeled and T_I is unlabeled, the definition of DEL implies that

$$XNODE(T_I) - XNODE(O_I) - TIME(I,1) \geq DEL$$

and hence

$$XNODE(T_I) - XNODE(O_I) - DEL \geq TIME(I,1)$$

which implies that $XACT(I) \geq TIME(I,1)$.

(b) Now $ABAR[I, NK(I)] = 0$ cannot occur, since T_I would have been labeled from O_I .

(c) Also $ABAR[I, NK(I)] > 0$ cannot happen since this would imply that

$$\text{old } XNODE(O_I) + TIME(I,1) \geq \text{old } XNODE(T_I)$$

which contradicts the feasibility of the previous node times.

(iv) Suppose O_I is unlabeled and T_I is labeled for activity I. Then

$$\text{new } XNODE(O_I) = \text{old } XNODE(O_I) - DEL,$$

$$\text{new } XNODE(T_I) = \text{old } XNODE(T_I), \text{ and}$$

$$\text{new } XACT(I) = \min\{TIME[I, NK(I)], \text{old } XNODE(T_I)$$

$$- \text{old } XNODE(O_I) + DEL\};$$

so that (2.13) and (2.15) are satisfied. Since

$$TIME(I,1) \leq \text{old } XACT(I) \leq \text{new } XACT(I),$$

the lower bound constraint, (2.16), is trivially satisfied. QED.

Lemma 6: The starting values of the $F(I,J)$'s and V are feasible and remain feasible throughout the algorithm.

Proof: Initially, the values of the $F(I,J)$'s and V are set to zero. Conservation of flow, (2.24), is trivially satisfied since the flow going into each node is equal to zero which is also equal to the flow going out of each node, i.e.

$$\sum_{I \in O_I = K} [\sum_J F(I,J)] = 0 = \sum_{I \in T_I = K} [\sum_J F(I,J)].$$

Since all $F(I,J)$ are set equal to zero, they satisfy their upper and lower bounds. Hence, the starting values are feasible.

If nonbreakthrough occurs, the values for $F(I,J)$ do not change, hence remain feasible.

If breakthrough occurs, the $F(I,J)$ along the path from the source to the sink are updated by a positive number $CAP(SINK)$ determined by (2.48) or (2.51); all other flows remain unchanged. Suppose activity I is an arc along the path from the source to the sink. Then either T_I is labeled from O_I or O_I is labeled from T_I .

(1) In the former case, $F(I,J) < C[I, NK(I) - J] - C[I, NK(I) - J + 1]$ by labeling rules (2.46) and (2.47), and $CAP(I)$ is given by (2.48). This $CAP(I)$ is the minimum of the previous CAP and $C[I, NK(I) - J] - C[I, NK(I) - J + 1] - F(I, J) > 0$. Now $CAP(SINK) \leq CAP(I)$ and new $F(I,J) = \text{old } F(I,J) + CAP(SINK)$. Conservation of flow is satisfied since the same value is added to or subtracted from all activities along this path and V , the total flow, is increased by $CAP(SINK)$. Also, (2.34) is satisfied since

$$0 \leq \text{old } F(I,J) + \text{CAP}(\text{SINK}) \leq \text{old } F(I,J) + \text{CAP}(I) \\ < C[I, \text{NK}(I) - J] - C[I, \text{NK}(I) - J + 1].$$

Hence the new $F(I,J)$'s are feasible.

(ii) In the latter case, $F(I,J) > 0$ by labeling rules (2.49) and (2.50), and $\text{CAP}(I)$ is given by (2.51); i.e., the minimum of the previous $\text{CAP}(K)$ and $F(I,J)$. Conservation of flow is again satisfied. The following also shows that (2.34) is satisfied: Now

$$\text{old } F(I,J) \leq C[I, \text{NK}(I) - J] - C[I, \text{NK}(I) - J + 1], \text{ and}$$

$$\text{new } F(I,J) = \text{old } F(I,J) - \text{CAP}(\text{SINK}).$$

Since $\text{CAP}(\text{SINK}) \leq \text{CAP}(I) \leq \text{old } F(I,J)$, this implies that

$$0 \leq \text{new } F(I,J) \leq C[I, \text{NK}(I) - J] - C[I, \text{NK}(I) - J + 1].$$

Hence, $F(I,J)$ remains feasible for this case as well and, therefore, remains feasible throughout the algorithm. QED.

Lemma 7: The optimality properties (2.42) and (2.43) imply that complementary slackness holds between the primal and the dual problems.

Proof: We will use the original pair of primal and dual problems ((2.13) - (2.19) and (2.22) - (2.25) respectively) along with the definitions of $G(I,M)$, $H(I,M)$ and $F(I,J)$ to show that the complementary slackness conditions are satisfied; i.e.,

$$(i) \quad \text{XACT}(I) + \text{XNODE}(O_I) - \text{XNODE}(T_I) < 0$$

implies that $F(I) = 0$;

$$(ii) \quad \text{XACT}(I,M) < U(I,M) \text{ implies that } G(I,M) = 0; \text{ and}$$

$$(iii) \quad \text{XACT}(I,M) > L(I,M) \text{ implies that } H(I,M) = 0.$$

$$(1) \quad \text{If } XACT(I) + XNODE(O_I) - XNODE(T_I) < 0, \text{ then} \\ XACT(I) < XNODE(T_I) - XNODE(O_I).$$

Since

$$XACT(I) = \min\{TIME[I, NK(I)], XNODE(T_I) - XNODE(O_I)\},$$

this implies that

$$XACT(I) = TIME[I, NK(I)].$$

Hence,

$$TIME[I, NK(I)] + XNODE(O_I) - XNODE(T_I) < 0.$$

Since

$$TIME(I, 1) \leq TIME(I, 2) \leq \dots \leq TIME[I, NK(I)],$$

it follows that

$$TIME(I, NK(I) + 1 - J) + XNODE(O_I) - XNODE(T_I) < 0$$

for $J = 1, 2, \dots, NK(I)$. From optimality property (2.42), $F(I, J) = 0$ for $J = 1, 2, \dots, NK(I)$, and finally $F(I) = \sum_J F(I, J) = 0$.

(Remark: Since

$$ABAR(I, J) = TIME[I, NK(I) + 1 - J] + XNODE(O_I) - XNODE(T_I)$$

and

$$TIME(I, 1) \leq TIME(I, 2) \leq \dots \leq TIME[I, NK(I)],$$

it follows that

$$ABAR(I,1) \geq ABAR(I,2) \geq \dots \geq ABAR[I, NK(I)].$$

Now the $TIME(I,J)$'s will be strictly increasing and the $ABAR(I,J)$'s strictly decreasing unless there is only one possible value for $XACT(I)$ in which case the upper and lower bounds for $F(I)$ and the $F(I,J)$'s are 0.

Therefore in the Second Labeling part (i), page 24, there can only be one J such that

$$ABAR(I,J) = 0$$

and

$$F(I,J) < C[I, NK(I) - J] - C[I, NK(I) - J + 1].$$

For this J

$$0 > ABAR(I, J + 1) > \dots > ABAR[I, NK(I)],$$

so that by optimality property (2.42)

$$F(I, J + 1) = \dots = F[I, NK(I)] = 0.$$

Also, for this J

$$ABAR(I, 1) > \dots > ABAR(I, J - 1) > 0,$$

so that by optimality property (2.43) $F(I, 1), \dots, F(I, J - 1)$ are all at their upper bounds. Thus, when $F(I)$ is increased, it is the $F(I,J)$ with the smallest index J such that $F(I,J)$ is less than its upper bound which is increased.

Similarly the Second Labeling part (ii) and the optimality properties imply that when $F(I)$ is decreased it is the $F(I, J)$ with the largest index J such that $F(I, J) > 0$ which is decreased. Therefore, if $F(I, J)$ is positive, then $F(I, 1), \dots, F(I, J - 1)$ are all at their upper bounds; and, if $F(I, J) = 0$, then $F(I, J + 1), \dots, F(I, NK(I))$ also equal 0. These natural properties of the $F(I, J)$'s are used in parts (ii) and (iii) below.)

(ii) Show that $XACT(I, M) < U(I, M)$ implies $G(I, M) = 0$ where, as in (2.15) and (2.28),

$$U(I, M) = \begin{cases} TIME(I, 2) & M = 1 \\ TIME(I, M + 1) - TIME(I, M) & M = 2, \dots, NK(I) - 1, \end{cases}$$

$$G(I, M) = \max\{0, C(I, M) - F(I)\},$$

and

$$XACT(I, M) = \begin{cases} \min[U(I, M), XACT(I)] & M = 1 \\ \min[U(I, M), \max\{0, XACT(I) - TIME(I, M)\}] & M = 2, \dots, NK(I) - 1. \end{cases}$$

If $XACT(I, M) < U(I, M)$, then

$$XACT(I, M) = \begin{cases} XACT(I) & M = 1 \\ \max\{0, XACT(I) - TIME(I, M)\} & M = 2, \dots, NK(I) - 1. \end{cases}$$

Case I: $M = 1$.

Since

$$TIME(I, 2) = U(I, 1) > XACT(I, 1) = XACT(I),$$

it follows that

$$\text{TIME}(I, 2) > \text{XACT}(I) = \min\{\text{TIME}[I, \text{NK}(I)], \text{XNODE}(T_I) - \text{XNODE}(O_I)\}.$$

Since

$$\text{TIME}(I, 2) \leq \text{TIME}[I, \text{NK}(I)],$$

this implies that

$$\text{TIME}(I, 2) > \text{XNODE}(T_I) - \text{XNODE}(O_I)$$

and

$$\text{ABAR}[I, \text{NK}(I) - 1] = \text{TIME}(I, 2) + \text{XNODE}(O_I) - \text{XNODE}(T_I) > 0.$$

By (2.43),

$$F[I, \text{NK}(I) - 1] = C(I, 1) - C(I, 2).$$

$$\text{Therefore } F(I, J) = C[\text{NK}(I) - J] - C[\text{NK}(I) - J + 1] \quad J = 1, \dots, \text{NK}(I) - 1.$$

Hence

$$\begin{aligned} F(I) &= \sum_J F(I, J) = F[I, \text{NK}(I)] + \sum_{J=1}^{\text{NK}(I)-1} F(I, J) \\ &= F[I, \text{NK}(I)] + C[I, \text{NK}(I) - 1] - C[I, \text{NK}(I)] \\ &\quad + C[I, \text{NK}(I) - 2] - C[I, \text{NK}(I) - 1] \\ &\quad + \dots \\ &\quad + C(I, 1) - C(I, 2) \\ &= F[I, \text{NK}(I)] + C(I, 1) - C[I, \text{NK}(I)]. \end{aligned}$$

Since $C[I, \text{NK}(I)] \equiv 0$, $F(I) \geq C(I, 1)$. Therefore,

$$\begin{aligned} G(I, 1) &= \max[0, C(I, 1) - F(I)] \\ &= 0. \end{aligned}$$

Case II: $M = 2, \dots, NK(I) - 1$.

Now $U(I, M) > XACT(I, M) = \max[0, XACT(I) - TIME(I, M)]$ implies

$$U(I, M) > XACT(I) - TIME(I, M)$$

and

$$U(I, M) > \min\{TIME[I, NK(I)], XNODE(T_I) - XNODE(O_I)\} - TIME(I, M);$$

so that

$$U(I, M) + TIME(I, M) > \min\{TIME[I, NK(I)], XNODE(T_I) - XNODE(O_I)\}.$$

Since $U(I, M) = TIME(I, M + 1) - TIME(I, M)$,

$$U(I, M) + TIME(I, M) = TIME(I, M + 1)$$

and

$$TIME(I, M + 1) > \min\{TIME[I, NK(I)], XNODE(T_I) - XNODE(O_I)\}.$$

Since $TIME(I, M + 1) \leq TIME[I, NK(I)]$, this implies

$$TIME(I, M + 1) > XNODE(T_I) - XNODE(O_I)$$

or

$$TIME(I, M + 1) + XNODE(O_I) - XNODE(T_I) > 0.$$

By (2.43),

$$F[I, NK(I) - M] = C(I, M) - C(I, M + 1).$$

Therefore $F(I, 1), \dots, F[I, NK(I) - M - 1]$ are also at their upper bounds.

Of course $\sum_{J=NK(I)-M+1}^{NK(I)} F(I,J) \geq 0$.

Thus

$$\begin{aligned} F(I) &= \sum_J F(I,J) \geq \sum_J^{NK(I)-M} F(I,J) \\ &= C[I, NK(I) - 1] - C[I, NK(I)] \\ &\quad + C[I, NK(I) - 2] - C[I, NK(I) - 1] \\ &\quad + \dots \\ &\quad + C(I, M) - C(I, M + 1) \\ &= C(I, M) - C[I, NK(I)]. \end{aligned}$$

Since $C[I, NK(I)] \equiv 0$, $F(I) \geq C(I, M)$.

Therefore,

$$\begin{aligned} G(I, M) &= \max[0, C(I, M) - F(I)] \\ &= 0 \end{aligned}$$

for $M = 2, \dots, NK(I) - 1$.

(iii) Show that $XACT(I, M) > L(I, M)$ implies that $H(I, M) = 0$ where, as in (2.16) and (2.29),

$$L(I, M) = \begin{cases} TIME(I, 1) & M = 1 \\ 0 & M = 2, \dots, NK(I) - 1, \end{cases}$$

$$H(I, M) = \max[0, F(I) - C(I, M)],$$

and

$$XACT(I,M) = \begin{cases} \min[U(I,M), XACT(I)] & M = 1 \\ \min\{U(I,M), \max[0, XACT(I) - TIME(I,M)]\} & M = 2, \dots, NK(I)-1. \end{cases}$$

Case I: $M = 1$.

If $XACT(I,1) > L(I,1)$, then

$$TIME(I,1) = L(I,1) < XACT(I,1) = \min[U(I,1), XACT(I)].$$

Since $U(I,1) = TIME(I,2)$,

$$TIME(I,1) < \min[TIME(I,2), XACT(I)]$$

and

$$TIME(I,1) < XACT(I).$$

Thus

$$TIME(I,1) < XACT(I) = \min\{TIME[I, NK(I)], XNODE(T_I) - XNODE(0_I)\}$$

and

$$TIME(I,1) < XNODE(T_I) - XNODE(0_I).$$

Therefore

$$ABAR[I, NK(I)] = TIME(I,1) + XNODE(0_I) - XNODE(T_I) < 0.$$

Then by (2.42)

$$F[I, NK(I)] = 0,$$

and

$$\begin{aligned}
 F(I) &= \sum_{J=1}^{NK(I)} F(I, J) = \sum_{J=1}^{NK(I)-1} F(I, J) \\
 &\leq \sum_{J=1}^{NK(I)-1} \{C[I, NK(I) - J] - C[I, NK(I) - J + 1]\} \\
 &= C[I, NK(I) - 1] - C[I, NK(I)] \\
 &\quad + C[I, NK(I) - 2] - C[I, NK(I) - 1] \\
 &\quad + \dots \\
 &\quad + C[I, 1] - C[I, 2] \\
 &= C[I, 1] - C[I, NK(I)].
 \end{aligned}$$

Since $C[I, NK(I)] \equiv 0$, $F(I) \leq C(I, 1)$. Therefore

$$\begin{aligned}
 H(I, 1) &= \max[0, F(I) - C(I, 1)] \\
 &= 0.
 \end{aligned}$$

Case II: $M = 2, \dots, NK(I) - 1$.

If

$$0 = L(I, M) < XACT(I, M) = \min\{U(I, M), \max[0, XACT(I) - TIME(I, M)]\},$$

then

$$0 < \min\{U(I, M), \max[0, XACT(I) - TIME(I, M)]\},$$

$$0 < \max[0, XACT(I) - TIME(I, M)], \text{ and}$$

$$0 < XACT(I) - TIME(I, M).$$

This implies that

$$TIME(I, M) < XACT(I) = \min\{TIME[I, NK(I)], XNODE(T_I) - XNODE(0_I)\};$$

so that

$$\text{TIME}(I,M) < \text{XNODE}(T_I) - \text{XNODE}(O_I),$$

and

$$\text{ABAR}[I, \text{NK}(I) - M + 1] = \text{TIME}(I,M) + \text{XNODE}(O_I) - \text{XNODE}(T_I) < 0.$$

By (2.42),

$$F[I, \text{NK}(I) - M + 1] = 0.$$

Therefore $F[I, \text{NK}(I) - M + 2], \dots, F[I, \text{NK}(I)]$ are all equal to 0. Hence,

$$\begin{aligned} F(I) &= \sum_J F(I,J) \leq \sum_{J=1}^{\text{NK}(I)-M} \{C[I, \text{NK}(I) - J] - C[I, \text{NK}(I) - J + 1]\} \\ &= C[I, \text{NK}(I) - 1] - C[I, \text{NK}(I)] \\ &\quad + C[I, \text{NK}(I) - 2] - C[I, \text{NK}(I) - 1] \\ &\quad + \dots \\ &\quad + C(I,M) - C(I,M + 1) \\ &= C(I,M) - C[I, \text{NK}(I)]. \end{aligned}$$

Since $C[I, \text{NK}(I)] \equiv 0$, $F(I) \leq C(I,M)$. Therefore,

$$H(I,M) = \max[0, F(I) - C(I,M)]$$

$$= 0$$

for $M = 2, \dots, \text{NK}(I) - 1$. QED.

Theorem 2: Since the $\text{XNODE}(K)$'s, $\text{XACT}(I,M)$'s, V , AND $F(I)$'s are feasible and complementary slackness holds, they are optimal.

Proof: The primal problem (2.13) - (2.19) is in the form

$$\max c^T x$$

subject to

$$Ax \leq b,$$

where the x vector contains the $XNODE(K)$'s and $XACT(I,M)$'s. The dual problem (2.22) - (2.25) is in the form

$$\min b^T w$$

subject to

$$A^T w = c$$

$$w \geq 0$$

where the w vector contains V and the $F(I)$'s.

For any feasible x

$$Ax \leq b;$$

so that for any feasible w

$$w^T Ax \leq w^T b.$$

Since $w^T A = c^T$ for any feasible w ,

$$c^T x \leq b^T w$$

holds for any feasible x and w . When $Ax \leq b$ is rewritten in the form

$$Ax + x_s = b$$

where x_s is a vector of slack variables, complementary slackness means

$$w^T x_s = 0.$$

Therefore, since for any feasible x and w

$$w^T A x + w^T x_s = w^T b$$

or

$$c^T x + w^T x_s = b^T w,$$

complementary slackness implies

$$c^T x = b^T w$$

and hence that both x and w are optimal. QED.

Theorem 3: The algorithm terminates after finitely many applications of the labeling procedure.

Proof: In order that the algorithm fail to terminate, an infinite sequence of breakthroughs and nonbreakthroughs would have to occur.

Since the flow change following a breakthrough has a positive minimum, an infinite number of breakthroughs would produce flows having arbitrarily large values V . However, when a sufficiently large value V is reached, there will be a path from the source to the sink with $F[I, NK(I)] > 0$ all along this path. Since $ABAR[I, NK(I)] \leq 0$ throughout the computations, we would have $ABAR[I, NK(I)] = 0$ for arcs on this path. But then the first labeling procedure would terminate. Therefore, there can only be a finite number of breakthroughs.

Following a nonbreakthrough, all nodes previously labeled can again be labeled. (This follows from the fact that for labeled O_I and T_I , the new $ABAR(I, J)$ is equal to the old $ABAR(I, J)$). In addition, at least one more node can be

labeled (the node(s) corresponding to the arc(s) in A_1 and A_2 that determine DEL). Eventually, the number of nodes that can be labeled will reach the total number of nodes implying that the sink can be labeled and the occurrence of a breakthrough. Therefore, infinitely many successive nonbreakthroughs cannot occur.

Hence, there can only be a finite number of applications of the labeling procedure. QED.

Definition: A function $P(X)$ is said to be convex over some interval in X , if for any two points X_1, X_2 in the interval and for all $\alpha, 0 \leq \alpha \leq 1$,

$$P[\alpha \cdot X_2 + (1 - \alpha)X_1] \leq \alpha \cdot P(X_2) + (1 - \alpha) \cdot P(X_1).$$

Lemma 8: PCOST(LAMBDA) is convex for $L_{MAX} \geq LAMBDA \geq L_{MIN}$, where

$L_{MAX} \equiv$ the longest (cheapest) time to complete the project

and

$L_{MIN} \equiv$ the shortest time to complete the project.

Proof: Let $L_1 > L_2$ both be in the interval $[L_{MIN}, L_{MAX}]$. Let

$$L = \alpha L_2 + (1 - \alpha)L_1$$

for some α in $[0, 1]$. Also let $XACT_1(I)$, $XNODE_1(0_I)$, $XNODE_1(T_I)$, $XACT_2(I)$, $XNODE_2(0_I)$, and $XNODE_2(T_I)$ represent optimal solutions to the problems corresponding to $LAMBDA = L_1$ and $LAMBDA = L_2$ respectively. We first want to show that $[\alpha XNODE_2(K) + (1 - \alpha) XNODE_1(K)]$ and $[\alpha \cdot XACT_2(I, M) + (1 - \alpha) XACT_1(I, M)]$ are feasible when $LAMBDA = L$. This result follows easily since the constraints (2.13), (2.14), (2.15), and (2.16) are linear:

$$(i) \quad \text{Since } XNODE_1(0_I) + \sum_M XACT_1(I,M) - XNODE_1(T_I) \leq 0$$

and

$$XNODE_2(0_I) + \sum_M XACT_2(I,M) - XNODE_2(T_I) \leq 0,$$

it follows that

$$\begin{aligned} & [\alpha XNODE_2(0_I) + (1 - \alpha) XNODE_1(0_I)] + \sum_M [\alpha XACT_2(I,M) + (1 - \alpha) XACT_1(I,M)] \\ & - [\alpha XNODE_2(T_I) + (1 - \alpha) XNODE_1(T_I)] \\ & \leq 0 \end{aligned}$$

and the constraints (2.13) are satisfied.

$$(ii) \quad \text{Now } XNODE_1(SINK) \leq L1 \text{ and } XNODE_2(SINK) \leq L2; \text{ so that}$$

$$(1 - \alpha)XNODE_1(SINK) + \alpha XNODE_2(SINK) \leq (1 - \alpha)L1 + \alpha L2 = L$$

and constraint (2.14) is satisfied.

$$(iii) \quad \text{Also, } L(I,M) \leq XACT_1(I,M) \leq U(I,M) \text{ and } L(I,M) \leq XACT_2(I,M) \leq U(I,M)$$

implies

$$\begin{aligned} L(I,M) & \leq (1 - \alpha)XACT_1(I,M) + \alpha XACT_2(I,M) \\ & \leq U(I,M) \end{aligned}$$

and hence constraints (2.15) and (2.16) are satisfied.

Recall that

$$PCOST(LAMBDA) = KK - \sum_{I,M} [C(I,M)XACT(I,M)]$$

where

$$KK = \sum_I [COST(I,1) + C(I,1)TIME(I,1)].$$

Hence,

$$\begin{aligned} & \alpha PCOST(L2) + (1 - \alpha) PCOST(L1) \\ &= \alpha \{ KK - \sum_{I,M} [C(I,M) XACT_2(I,M)] \} + (1 - \alpha) \{ KK - \sum_{I,M} [C(I,M) XACT_1(I,M)] \} \\ &= \alpha KK + (1 - \alpha) KK - \alpha \sum_{I,M} C(I,M) XACT_2(I,M) - (1 - \alpha) \sum_{I,M} C(I,M) XACT_1(I,M) \\ &= KK - \sum_{I,M} C(I,M) [\alpha XACT_2(I,M)] - \sum_{I,M} C(I,M) [(1 - \alpha) XACT_1(I,M)] \\ &= KK - \sum_{I,M} C(I,M) [\alpha XACT_2(I,M) + (1 - \alpha) XACT_1(I,M)]. \end{aligned}$$

Furthermore $\alpha PCOST(L2) + (1 - \alpha) PCOST(L1)$ is the objective function value corresponding to $[\alpha XNODE_2(K) + (1 - \alpha) XNODE_1(K)]$ and $[\alpha XACT_2(I,M) + (1 - \alpha) XACT_1(I,M)]$ which we have just shown are feasible. Therefore, since we are minimizing $PCOST(LAMBDA)$,

$$\begin{aligned} PCOST(L) &= PCOST[\alpha L2 + (1 - \alpha) L1] \\ &\leq \alpha PCOST(L2) + (1 - \alpha) PCOST(L1), \end{aligned}$$

and $PCOST(LAMBDA)$ is convex.

Lemma 9: The project cost function, $PCOST(LAMBDA)$, is piecewise linear.

Proof: Let $L1 > L2 = L1 - DEL$ be two successively determined $LAMBDA$'s where DEL is determined by (2.56). (Of course, $L1$ could be the initial value of $LAMBDA$.) Suppose $L1 \geq LAMBDA \geq L2$ and that $F(I,J)$'s and V were the flows when $LAMBDA$ was changed from $L2$ to $L1$. Recall that

$$PCOST(LAMBDA) = PCOST[XNODE(SINK)] = \sum_I [KK(I) - \sum_M C(I,M) XACT(I,M)]$$

which is the primal objective function. Since the primal and dual objective functions are equal under optimality, we have, for all LAMBDA with $L1 \geq LAMBDA \geq L2$,

$$PCOST(LAMBDA) = Z - LAMBDA \cdot V + \sum_{IJ} F(I,J)TIME[I, NK(I) - J + 1]$$

where Z is a constant. Therefore

$$\begin{aligned} PCOST(LAMBDA) - PCOST(L1) &= Z - LAMBDA \cdot V + \sum_{IJ} F(I,J)TIME[I, NK(I) - J + 1] \\ &\quad - Z + L1 \cdot V - \sum_{IJ} F(I,J)TIME[I, NK(I) - J + 1] \\ &= - LAMBDA \cdot V + L1 \cdot V \\ &= (L1 - LAMBDA) \cdot V \end{aligned}$$

for $L1 \geq LAMBDA \geq L2$, so that PCOST(LAMBDA) is linear on the given interval. QED.

Theorem 4: If $L1 > L2 = L1 - DEL$ are two successively determined values of LAMBDA where DEL is determined by (2.56), then for any value of L such that $L1 > L \geq L2$ the optimal values of the XNODE(K)'s and XACT(I)'s for that L are given by

$$XNODE_L(K) = \begin{cases} XNODE_{L1}(K) & \text{if K is labeled when} \\ & LAMBDA = L1, \\ XNODE_{L1}(K) - (L1 - L) & \text{if K is unlabeled when} \\ & LAMBDA = L1, \end{cases}$$

$$XACT_L(I) = \min\{TIME[I, NK(I)], XNODE_L(T_I) - XNODE_L(0_I)\}$$

where the subscripts L and L1 imply $LAMBDA = L$ and $L1$ respectively.

Proof: Since Lemma 1 states that we begin with an optimal solution when $LAMBDA = LMAX$, we can without loss of generality assume that we have found optimal solutions for all $LAMBDA$ values produced by the nonbreakthrough procedure up to $LAMBDA = L1$. We will now show that the above $XNODE(K)$'s and $XACT(I)$'s are optimal for all $LAMBDA$ between $L1$ and $L2$ including $L2$. The terms "labeled" and "unlabeled" below refer to "labeled when $LAMBDA = L1$ " and "unlabeled when $LAMBDA = L1$ " respectively.

We first want to show that for $L1 > L \geq L2$ the $XNODE_L(K)$'s and $XACT_L(I)$'s are feasible. Since the definition of $XACT_L(I)$ implies that

$$XACT_L(I) + XNODE_L(0_I) - XNODE_L(T_I) \leq 0$$

and

$$XACT_L(I) \leq TIME[I, NK(I)],$$

(2.13) and (2.15) are satisfied. Therefore, the only aspect of feasibility left to show is (2.16), i.e.

$$TIME(I,1) \leq XACT_L(I)$$

or equivalently

$$TIME(I,1) \leq XNODE_L(T_I) - XNODE_L(0_I).$$

(1) Suppose 0_I and T_I are both labeled for a specific activity I. Then

$$XNODE_L(T_I) - XNODE_L(0_I) = XNODE_{L1}(T_I) - XNODE_{L1}(0_I) \geq TIME(I,1)$$

since the solution at $L1$ is feasible.

(ii) Suppose 0_I is labeled and T_I is unlabeled. Then

$$XNODE_L(T_I) - XNODE_L(0_I) = XNODE_{L1}(T_I) - (L1 - L) - XNODE_{L1}(0_I).$$

The definition of DEL implies that if $ABAR[I, NK(I)] < 0$, then

$$XNODE_{L1}(T_I) - XNODE_{L1}(0_I) - TIME(I, 1) \geq DEL \geq L1 - L.$$

Hence,

$$XNODE_{L1}(T_I) - XNODE_{L1}(0_I) - (L1 - L) \geq TIME(I, 1)$$

and

$$XNODE_L(T_I) - XNODE_L(0_I) \geq TIME(I, 1).$$

If $ABAR[I, NK(I)] = 0$, then T_I would have been labeled from 0_I . Since feasibility is satisfied at $LAMBDA = L1$, it follows that

$$TIME(I, 1) + XNODE_{L1}(0_I) - XNODE_{L1}(T_I) \leq 0,$$

and consequently, since

$$ABAR[I, NK(I)] = TIME(I, 1) + XNODE_{L1}(0_I) - XNODE_{L1}(T_I),$$

$ABAR[I, NK(I)]$ cannot be positive.

(iii) Suppose 0_I and T_I are both unlabeled. Then

$$\begin{aligned} XNODE_L(T_I) - XNODE_L(0_I) &= XNODE_{L1}(T_I) - (L1 - L) - [XNODE_{L1}(0_I) - (L1 - L)] \\ &= XNODE_{L1}(T_I) - XNODE_{L1}(0_I) \\ &\geq TIME(I, 1) \end{aligned}$$

since the solution at $L1$ is feasible.

(iv) Suppose 0_I is unlabeled and T_I is labeled. Then

$$\begin{aligned} XNODE_L(T_I) - XNODE_L(0_I) &= XNODE_{L1}(T_I) - [XNODE_{L1}(0_I) - (L1 - L)] \\ &= XNODE_{L1}(T_I) - XNODE_{L1}(0_I) + (L1 - L). \end{aligned}$$

Since feasibility is satisfied at $LAMBDA = L1$,

$$XNODE_{L1}(T_I) - XNODE_{L1}(0_I) \geq TIME(I,1)$$

and trivially

$$XNODE_{L1}(T_I) - XNODE_{L1}(0_I) + (L1 - L) \geq TIME(I,1).$$

Hence,

$$XNODE_L(T_I) - XNODE_L(0_I) \geq TIME(I, 1).$$

Now, we have just shown that the $XNODE_L(K)$'s and $XACT_L(I)$'s are feasible. Lemma 6 implies that the $F(I,J)$'s are always kept feasible. Lemma 4 implies that the optimality properties (2.42) and (2.43) are satisfied for these $XNODE_L(K)$'s and $F(I,J)$'s; so that by Lemma 7 these $XNODE_L(K)$'s and $F(I,J)$'s also satisfy complementary slackness. Since we have shown that feasibility and complementary slackness are satisfied, Theorem 3 implies that the $XNODE_L(K)$'s and $XACT_L(I)$'s for $L1 > L \geq L2$ are optimal. QED.

4. A COMPUTER IMPLEMENTATION

A computer program implementing the improved project scheduling algorithm described in Chapter 2 is available. The basic input to the program is

- (a) an acyclic project network with one source and one sink, and
- (b) a collection of activity completion times and their associated costs.

The program's output for each feasible project deadline time consists mainly of

- (a) the optimal activity completion times and costs, and
- (b) the total project cost.

Optional output may include node labels, optimal node times for each project deadline time, and dual variables (flows).

Incorporated in this program is the option to have the minimum project cost and corresponding optimal activity completion times determined for only one specific project deadline time.

A listing of the computer program is given in the appendix. The flowchart for this program is given in Chapter 2, Section 4, pages 27-29.

4.1. Specific Input Instructions

Card 1. Col. 1 - 4: The number of nodes in the network,

Format (I4).

Col. 6 - 9: The number of activities in the network,

Format (I4).

Col. 11: TEST1 = 0 print the input data,

= 1 do not print the input data.

Col. 13: TEST2 = 0 print the intermediate output,

= 1 do not print the intermediate output.

Col. 15-18: The number of the source node,

Format (I4).

Col. 20-23: The number of the sink node,

Format (I4).

Col. 25: TEST3 = 0 do not wish to specify a single value for

LAMBDA,

= 1 do wish to specify a single value for

LAMBDA and print the intermediate output,

= 2 do wish to specify a single value for

LAMBDA but do not print the intermediate output.

For each activity I one set of 3 - 5 cards:

Card 1. Col. 1 - 4: O_I = the number of the origin node,

Format (I4).

Col. 6 - 9: T_I = the number of the terminal node,

Format (I4).

Col. 11-12: $NK(I)$ = the number of activity completion times and costs that are read in (≤ 11),

Format (I2).

Card(s) 2 - 3. Format (8I10): $TIME(I,1), \dots, TIME[I, NK(I)]$ = the activity completion times in increasing order (8 on Card 2, 3 on Card 3 if needed).

Card(s) 4 - 5. Format (8I10): $COST(I,1), \dots, COST[I, NK(I)]$ = the cost associated with each activity completion time (8 on Card 4, 3 on Card 5 if needed).

The next card is present only if $TEST3 = 1$ or 2 .

Last Card. Col. 1 -10: Specified project deadline time,

LAMBDA, Format (I10).

The nodes and activities may be numbered in any order. The current dimensions will allow 3000 nodes, 3000 activities, and at most 11 different completion times and costs.

4.2. An Example

The program's input and output are illustrated in terms of the example network in Figure 10. The input data are found in Table 1. As an example, the activity cost curve for activity 7 is illustrated in Figure 11.

A listing of the computer input is given in Figure 12. The optimal project cost curve determined by the algorithm is plotted in Figure 13. The optimal activity durations for two values of the project deadline time, LAMBDA , are given in Table 2. The actual computer output is given in Figure 14.

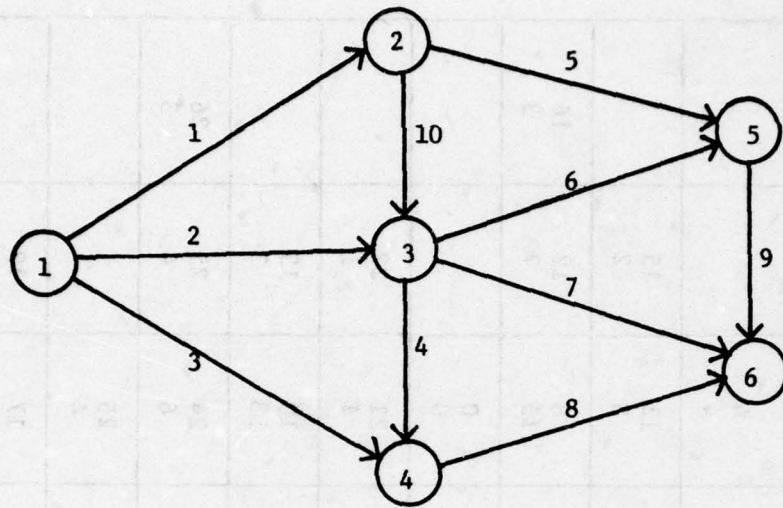


FIGURE 10

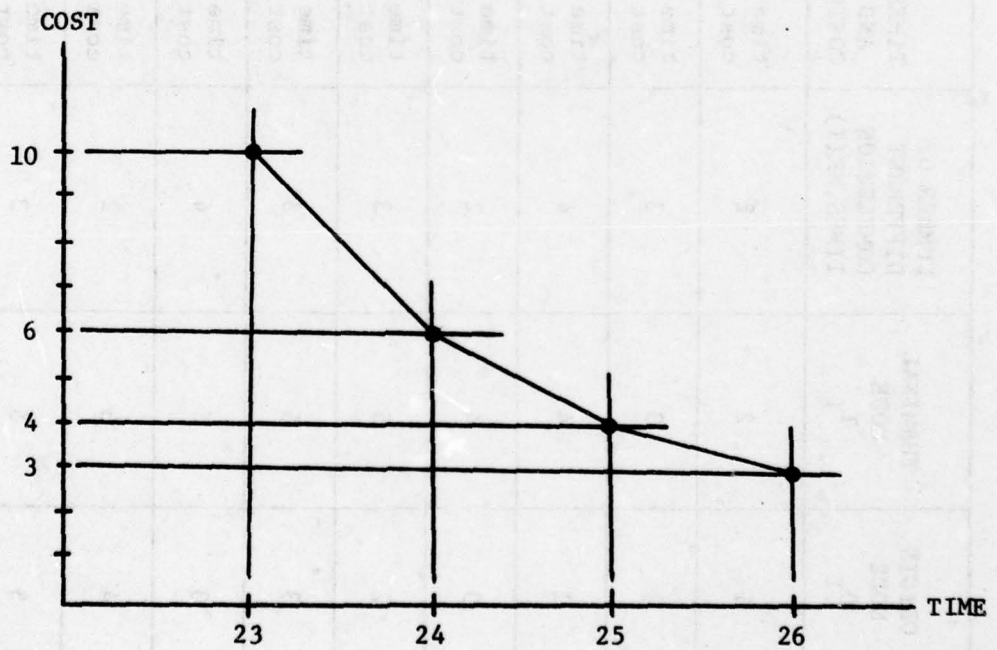


FIGURE 11

TABLE 1: EXAMPLE DATA

ACTIVITY NUMBER I	ORIGIN NODE O _I	TERMINAL NODE T _I	NUMBER OF DIFFERENT COMPLETION TIMES, NK(I)	TIMES AND COSTS	J = 1	J = 2	J = 3	J = 4
1	1	2	2	time cost	2 8	4 4		
2	1	3	3	time cost	7 23	12 8	15 2	
3	1	4	4	time cost	4 27	8 15	12 7	16 3
4	3	4	2	time cost	0 0	0 0		
5	2	5	3	time cost	20 8	21 4	22 1	
6	3	5	3	time cost	5 28	10 13	15 3	
7	3	6	4	time cost	23 10	24 6	25 4	26 3
8	4	6	2	time cost	23 8	25 4		
9	5	6	3	time cost	16 12	17 7	19 3	
10	2	3	2	time cost	6 4	6 4		

Figure 11: Computer Input for Example

6	10	0	0	1	6	0
1	2	2				
	2			4		
	8			4		
1	3	3				
	7			12	15	
	23			8	2	
1	4	4				
	4			8	12	16
	27			15	7	3
3	4	2				
	0			0		
	0			0		
2	5	3				
	20			21	22	
	8			4	1	
3	5	3				
	5			10	15	
	28			13	3	
3	6	4				
	23			24	25	26
	10			6	4	3
4	6	2				
	23			25		
	8			4		
5	6	3				
	16			17	19	
	12			7	3	
2	3	2				
	6			6		
	4			4		

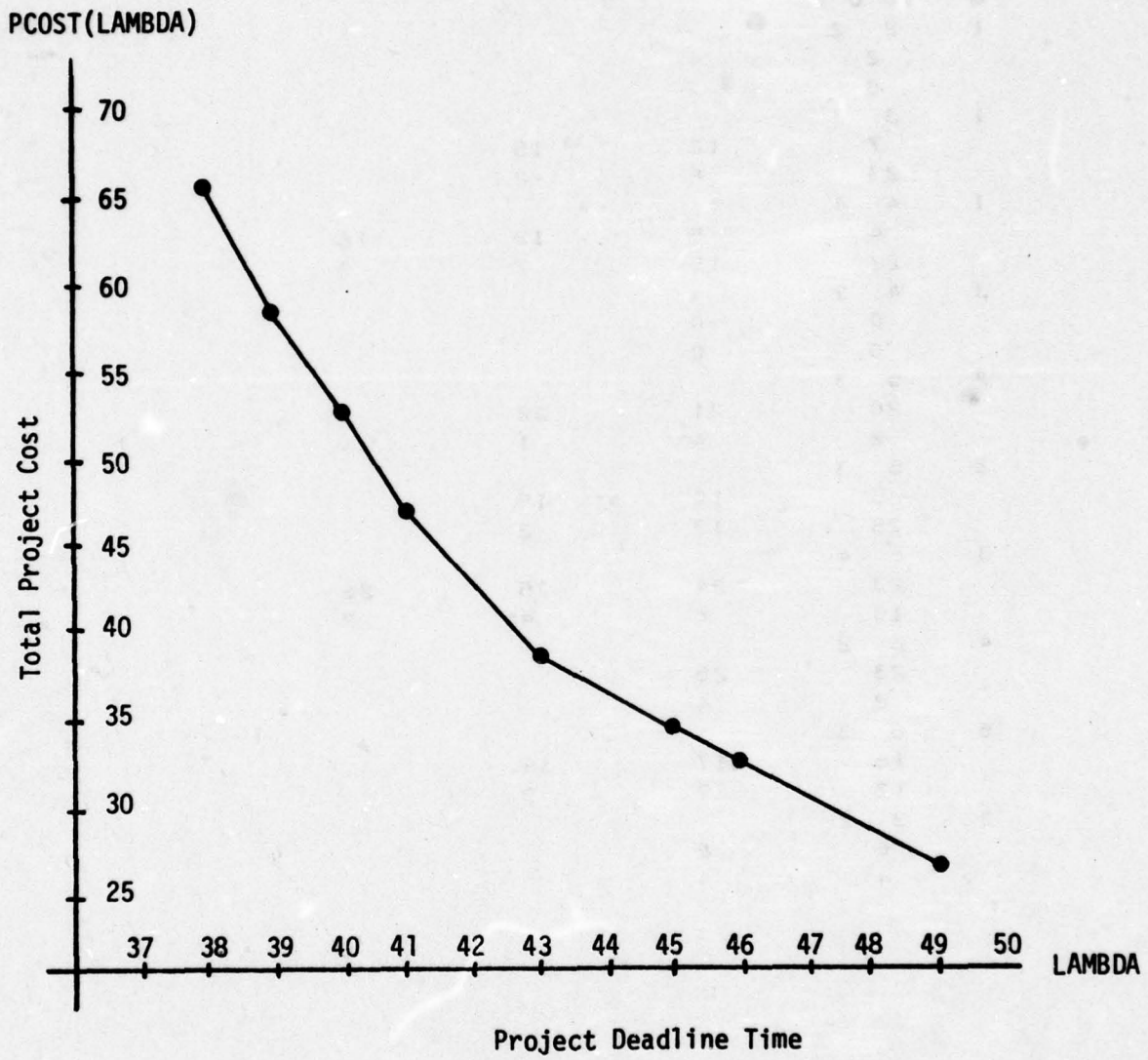


Figure 13.

Table 2: OPTIMAL PROJECT SCHEDULES
FOR TWO SPECIFIED DEADLINE TIMES

Activity # (I)	Project Deadline Time LAMBDA = 40		Project Deadline Time LAMBDA = 44	
	Activity Duration Time XACT(I)	Activity Cost	Activity Duration Time XACT(I)	Activity Cost
1	2	8	4	4
2	12	8	12	8
3	15	4	16	3
4	0	0	0	0
5	21	4	22	1
6	11	11	14	5
7	26	3	26	3
8	25	4	25	4
9	17	7	18	5
10	6	4	6	4

Figure 14: Computer Output for Example

THE NUMBER OF NODES IS 6.
 THE NUMBER OF ACTIVITIES IS 10.
 THE SOURCE NODE IS NUMBERED 1 AND THE SINK NODE IS NUMBERED 6.

** NODES: **

K	1	2	3	4	5	6
INITIAL XNODE(K)	0	4	15	16	30	49

** ACTIVITIES: **

I	XACT	ORIG	TERM	J	TIME	COST	C	ABAR
1	4	1	2	1	2	8	0.20000E 01	0
				2	4	4		-2
2	15	1	3	1	7	23	0.30000E 01	0
				2	12	8	0.20000E 01	-3
				3	15	2		-8
3	16	1	4	1	4	27	0.30000E 01	0
				2	8	15	0.20000E 01	-4
				3	12	7	0.10000E 01	-8
				4	16	3		-12
4	0	3	4	1	0	0	0.0	-1
				2	0	0		-1
5	22	2	5	1	20	8	0.40000E 01	-4
				2	21	4	0.30000E 01	-5
				3	22	1		-6
6	15	3	5	1	5	28	0.30000E 01	0
				2	10	13	0.20000E 01	-5
				3	15	3		-10
7	26	3	6	1	23	10	0.40000E 01	-8
				2	24	6	0.20000E 01	-9
				3	25	4	0.10000E 01	-10
				4	26	3		-11
8	25	4	6	1	23	8	0.20000E 01	-8
				2	25	4		-10
9	19	5	6	1	16	12	0.50000E 01	0
				2	17	7	0.20000E 01	-2
				3	19	3		-3
10	6	2	3	1	6	4	0.0	-5
				2	6	4		-5

THE ENTIRE PROJECT COST CURVE IS GOING TO BE DETERMINED.

LAMBDA = PROJECT COMPLETION TIME

THE STARTING VALUE OF LAMBDA IS 49.

THE CORRESPONDING TOTAL PROJECT COST IS 0.27000E 02.

THE SOURCE HAS A VALUE OF ZERO AND IS ASSIGNED THE LABEL (-,-,-,INF).

*** ITERATION NUMBER 1 ***

THE SINK HAS NOT BEEN REACHED WITH INFINITE CAPACITY - CONTINUE WITH THE LABELING PROCESS.
THE NODES THAT HAVE BEEN LABELED WILL RETAIN THAT LABEL FOR THE REMAINDER OF THE ITERATION.

THE NODE 2 HAS THE LABEL (1, 1, 0, 0.20000E 01).
THE NODE 3 HAS THE LABEL (1, 1, 0, 0.20000E 01).
THE NODE 4 HAS THE LABEL (1, 1, 0, 0.10000E 01).
THE NODE 5 HAS THE LABEL (3, 1, 0, 0.20000E 01).
THE NODE 6 HAS THE LABEL (5, 1, 0, 0.20000E 01).

BREAKTHROUGH: UPDATE THE DUAL VARIABLES.

ACTIVITY #:	I	J	NEW FLOW: F(I,J)
	1	1	0.0
	1	2	0.0
	2	1	0.20000E 01
	2	2	0.0
	2	3	0.0
	3	1	0.0
	3	2	0.0
	3	3	0.0
	3	4	0.0
	4	1	0.0
	4	2	0.0
	5	1	0.0
	5	2	0.0
	5	3	0.0
	6	1	0.20000E 01
	6	2	0.0
	6	3	0.0
	7	1	0.0
	7	2	0.0
	7	3	0.0
	7	4	0.0
	8	1	0.0
	8	2	0.0
	9	1	0.20000E 01
	9	2	0.0
	9	3	0.0
	10	1	0.0
	10	2	0.0

*** ITERATION NUMBER 2 ***

THE SINK HAS NOT BEEN REACHED WITH INFINITE CAPACITY - CONTINUE WITH THE LABELING PROCESS.
THE NODES THAT HAVE BEEN LABELED WILL RETAIN THAT LABEL FOR THE REMAINDER OF THE ITERATION.

THE NODE 2 HAS THE LABEL (1, 1, 0, 0.20000E 01).
THE NODE 4 HAS THE LABEL (1, 1, 0, 0.10000E 01).

NONBREAKTHROUGH: UPDATE THE PRIMAL VARIABLES;
I.E., DETERMINE OPTIMAL ACTIVITY TIMES FOR LAMBDA = 46.

DELTA (REPRESENTED BY "D") RANGES FROM 0 TO 3.
LAMBDA RANGES FROM 49 TO 46.
THE MINIMUM COST PROJECT SCHEDULE FOR PROJECT DEADLINE = 49-D:

NODE #: K	NEW VALUE: XNODE(K)
1	0
2	4
3	15-D
4	16
5	30-D
6	49-D

PROJECT COMPLETION TIME = 49-D.

ACTIVITY #: I	NEW VALUE: XACT(I)	ACTIVITY COST
1	4	0.40000E 01
2	15-D	0.20000E 01 + (0.20000E 01*D)
3	16	0.30000E 01
4	0	0.0
5	22	0.10000E 01
6	15	0.30000E 01
7	26	0.30000E 01
8	25	0.40000E 01
9	19	0.30000E 01
10	6	0.40000E 01

THE CURRENT VALUE OF THE PROJECT COST IS 0.27000E 02 + (0.20000E 01*D).

NEW VALUES OF ABAR FOR J=1,2,...,NK(I)

I	J:	1	2	3	4	5	6	7	8	9
1		0	-2							
2		3	0	-5						
3		0	-4	-8	-12					
4		-4	-4							
5		-1	-2	-3						
6		0	-5	-10						
7		-8	-9	-10	-11					
8		-5	-7							
9		0	-2	-3						
10		-2	-2							

*** ITERATION NUMBER 3 ***

THE SINK HAS NOT BEEN REACHED WITH INFINITE CAPACITY - CONTINUE WITH THE LABELING PROCESS.
THE NODES THAT HAVE BEEN LABELED WILL RETAIN THAT LABEL FOR THE REMAINDER OF THE ITERATION.

THE NODE 2 HAS THE LABEL (1, 1, 0, 0.20000E 01).

THE NODE 3 HAS THE LABEL (1, 2, 0, 0.10000E 01).

THE NODE 4 HAS THE LABEL (1, 1, 0, 0.10000E 01).

NONBREAKTHROUGH: UPDATE THE PRIMAL VARIABLES;

I.E.: DETERMINE OPTIMAL ACTIVITY TIMES FOR LANUCA = 45.

DELTA (REPRESENTED BY "D") RANGES FROM 0 TO 1.

LAMBDA RANGES FROM 46 TO 45.

THE MINIMUM CCST PROJECT SCHEDULE FOR PROJECT DEADLINE = 46-D:

NODE #: K	NEW VALUE: XNODE(K)
1	0
2	4
3	12
4	16
5	27-D
6	46-D

PROJECT COMPLETION TIME = 46-D.

ACTIVITY #: I	NEW VALUE: XACT(I)	ACTIVITY COST
1	4	0.40000E 01
2	12	0.80000E 01
3	16	0.30000E 01
4	0	0.0
5	22	0.10000E 01
6	15-D	0.30000E 01 + (0.20000E 01*D)
7	26	0.30000E 01
8	25	0.40000E 01
9	19	0.30000E 01
10	6	0.40000E 01

THE CURRENT VALUE OF THE PROJECT COST IS 0.33000E 02 + (0.20000E 01*D).

NEW VALUES OF ABAR FOR J=1,2,...,NK(I)

I	J:	1	2	3	4	5	6	7	8	9
1		0	-2							
2		3	0	-5						
3		0	-4	-8	-12					
4		-4	-4							
5		0	-1	-2						
6		1	-4	-9						
7		-7	-8	-9	-10					
8		-4	-6							
9		0	-2	-3						
10		-2	-2							

*** ITERATION NUMBER 4 ***

THE SINK HAS NOT BEEN REACHED WITH INFINITE CAPACITY - CONTINUE WITH THE LABELING PROCESS.
THE NODES THAT HAVE BEEN LABELED WILL RETAIN THAT LABEL FOR THE REMAINDER OF THE ITERATION.

THE NODE 2 HAS THE LABEL (1. 1. 0. 0.20000E 01).

THE NODE 3 HAS THE LABEL (1. 2. 0. 0.10000E 01).

THE NODE 4 HAS THE LABEL (1. 1. 0. 0.10000E 01).

THE NODE 5 HAS THE LABEL (2. 1. 0. 0.20000E 01).

NONBREAKTHROUGH: UPDATE THE PRIMAL VARIABLES;
I.E. : DETERMINE OPTIMAL ACTIVITY TIMES FOR $\lambda = 43$.

DELTA (REPRESENTED BY "D") RANGES FROM 0 TO 2.
LAMBDA RANGES FROM 45 TO 43.
THE MINIMUM COST PROJECT SCHEDULE FOR PROJECT DEADLINE = 45-D:
NODE #: K NEW VALUE: XNODE(K)

1	0
2	4
3	12
4	16
5	26
6	45-D

PROJECT COMPLETION TIME = 45-D.

ACTIVITY #: I	NEW VALUE: XACT(I)	ACTIVITY COST
1	4	0.40000E 01
2	12	0.80000E 01
3	16	0.30000E 01
4	0	0.0
5	22	0.10000E 01
6	14	0.50000E 01
7	26	0.30000E 01
8	25	0.40000E 01
9	19-D	0.30000E 01 + (0.20000E 01*D)
10	6	0.40000E 01

THE CURRENT VALUE OF THE PROJECT COST IS 0.35000E 02 + (0.20000E 01*D).

NEW VALUES OF ABAR FOR $J=1,2,\dots,NK(I)$

I	J:	1	2	3	4	5	6	7	8	9
1		0	-2							
2		3	0	-5						
3		0	-4	-8	-12					
4		-4	-4							
5		0	-1	-2						
6		1	-4	-9						
7		-5	-6	-7	-8					
8		-2	-4							
9		2	0	-1						
10		-2	-2							

*** ITERATION NUMBER 5 ***

THE SINK HAS NOT BEEN REACHED WITH INFINITE CAPACITY - CONTINUE WITH THE LABELING PROCESS.
THE NODES THAT HAVE BEEN LABELED WILL RETAIN THAT LABEL FOR THE REMAINDER OF THE ITERATION.

THE NODE 2 HAS THE LABEL (1, 1, 0, 0.20000E 01).
THE NODE 3 HAS THE LABEL (1, 2, 0, 0.10000E 01).
THE NODE 4 HAS THE LABEL (1, 1, 0, 0.10000E 01).
THE NODE 5 HAS THE LABEL (2, 1, 0, 0.20000E 01).

THE NODE 6 HAS THE LABEL (5, 2, 0, 0.20000E 01).

BREAKTHROUGH: UPDATE THE DUAL VARIABLES.

ACTIVITY #:	I	J	NEW FLOW: F(I,J)
	1	1	0.20000E 01
	1	2	0.0
	2	1	0.20000E 01
	2	2	0.0
	2	3	0.0
	3	1	0.0
	3	2	0.0
	3	3	0.0
	3	4	0.0
	4	1	0.0
	4	2	0.0
	5	1	0.20000E 01
	5	2	0.0
	5	3	0.0
	6	1	0.20000E 01
	6	2	0.0
	6	3	0.0
	7	1	0.0
	7	2	0.0
	7	3	0.0
	7	4	0.0
	8	1	0.0
	8	2	0.0
	9	1	0.20000E 01
	9	2	0.20000E 01
	9	3	0.0
	10	1	0.0
	10	2	0.0

*** ITERATION NUMBER 6 ***

THE SINK HAS NOT BEEN REACHED WITH INFINITE CAPACITY - CONTINUE WITH THE LABELING PROCESS.
THE NODES THAT HAVE BEEN LABELED WILL RETAIN THAT LABEL FOR THE REMAINDER OF THE ITERATION.

THE NODE 3 HAS THE LABEL (1, 2, 0, 0.10000E 01).

THE NODE 4 HAS THE LABEL (1, 1, 0, 0.10000E 01).

NONBREAKTHROUGH: UPDATE THE PRIMAL VARIABLES;
I.E. DETERMINE OPTIMAL ACTIVITY TIMES FOR LAMBDA = 41.

DELTA (REPRESENTED BY "D") RANGES FROM 0 TO 2.
LAMBDA RANGES FROM 43 TO 41.
THE MINIMUM COST PROJECT SCHEDULE FOR PROJECT DEADLINE = 43-D:

NODE #:	K	NEW VALUE: XNCDE(K)
	1	0
	2	4-D
	3	12
	4	16
	5	26-D

6

43-D

PROJECT COMPLETION TIME = 43-D.

ACTIVITY #:	I	NEW VALUE: XACT(I)	ACTIVITY COST
1		4-D	0.40000E 01 + (0.20000E 01*D)
2		12	0.80000E 01
3		16	0.30000E 01
4		0	0.0
5		22	0.10000E 01
6		14-D	0.50000E 01 + (0.20000E 01*D)
7		26	0.30000E 01
8		25	0.40000E 01
9		17	0.70000E 01
10		6	0.40000E 01

THE CURRENT VALUE OF THE PROJECT COST IS 0.39000E 02 + (0.40000E 01*D).

NEW VALUES OF ABAR FOR J=1,2,...,NK(I)

I	J:	1	2	3	4	5	6	7	8	9
1		2	0							
2		3	0	-5						
3		0	-4	-8	-12					
4		-4	-4							
5		0	-1	-2						
6		3	-2	-7						
7		-3	-4	-5	-6					
8		0	-2							
9		2	0	-1						
10		-4	-4							

*** ITERATION NUMBER 7 ***

THE NODE 2 HAS THE LABEL (1, 2.0, INF).

THE SINK HAS NOT BEEN REACHED WITH INFINITE CAPACITY - CONTINUE WITH THE LABELING PROCESS.
THE NODES THAT HAVE BEEN LABELED WILL RETAIN THAT LABEL FOR THE REMAINDER OF THE ITERATION.

THE NODE 3 HAS THE LABEL (1, 2, 0, 0.10000E 01).

THE NODE 4 HAS THE LABEL (1, 1, 0, 0.10000E 01).

THE NODE 5 HAS THE LABEL (2, 1, 0, 0.10000E 01).

THE NODE 6 HAS THE LABEL (4, 1, 0, 0.10000E 01).

BREAKTHROUGH: UPDATE THE DUAL VARIABLES.

ACTIVITY #:	I	J	NEW FLOW: F(I,J)
	1	1	0.20000E 01
	1	2	0.0
	2	1	0.20000E 01
	2	2	0.0
	2	J	0.0

3	1	0.10000E 01
3	2	0.0
3	3	0.0
3	4	0.0
4	1	0.0
4	2	0.0
5	1	0.20000E 01
5	2	0.0
5	3	0.0
6	1	0.20000E 01
6	2	0.0
6	3	0.0
7	1	0.0
7	2	0.0
7	3	0.0
7	4	0.0
8	1	0.10000E 01
8	2	0.0
9	1	0.20000E 01
9	2	0.20000E 01
9	3	0.0
10	1	0.0
10	2	0.0

*** ITERATION NUMBER 8 ***

THE NODE 2 HAS THE LABEL (1, 2.0, INF).

THE SINK HAS NOT BEEN REACHED WITH INFINITE CAPACITY - CONTINUE WITH THE LABELING PROCESS.
THE NODES THAT HAVE BEEN LABELED WILL RETAIN THAT LABEL FOR THE REMAINDER OF THE ITERATION.

THE NODE 3 HAS THE LABEL (1, 2, 0, 0.10000E 01).

THE NODE 5 HAS THE LABEL (2, 1, 0, 0.10000E 01).

THE NODE 6 HAS THE LABEL (5, 2, 0, 0.10000E 01).

BREAKTHROUGH: UPDATE THE DUAL VARIABLES.

ACTIVITY #:	I	J	NEW FLOW: F(I,J)
	1	1	0.20000E 01
	1	2	0.0
	2	1	0.20000E 01
	2	2	0.0
	2	3	0.0
	3	1	0.10000E 01
	3	2	0.0
	3	3	0.0
	3	4	0.0
	4	1	0.0
	4	2	0.0
	5	1	0.30000E 01
	5	2	0.0
	5	3	0.0
	6	1	0.20000E 01
	6	2	0.0
	6	3	0.0

7	1	0.0
7	2	0.0
7	3	0.0
7	4	0.0
8	1	0.10000E 01
8	2	0.0
9	1	0.20000E 01
9	2	0.30000E 01
9	3	0.0
10	1	0.0
10	2	0.0

*** ITERATION NUMBER 9 ***

THE NODE 2 HAS THE LABEL (1, 2.0, INF).

THE SINK HAS NOT BEEN REACHED WITH INFINITE CAPACITY - CONTINUE WITH THE LABELING PROCESS.
THE NODES THAT HAVE BEEN LABELED WILL RETAIN THAT LABEL FOR THE REMAINDER OF THE ITERATION.

THE NODE 3 HAS THE LABEL (1, 2, 0, 0.10000E 01).

NONBREAKTHROUGH: UPDATE THE PRIMAL VARIABLES;
I.E. DETERMINE OPTIMAL ACTIVITY TIMES FOR LAMBDA = 40.

DELTA (REPRESENTED BY "D") RANGES FROM 0 TO 1.
LAMBDA RANGES FROM 41 TO 40.
THE MINIMUM COST PROJECT SCHEDULE FOR PROJECT DEADLINE = 41-D:

NODE #: K	NEW VALUE: XNODE(K)
1	0
2	2
3	12
4	16-D
5	24-D
6	41-D

PROJECT COMPLETION TIME = 41-D.

ACTIVITY #: I	NEW VALUE: XACT(I)	ACTIVITY COST
1	2	0.80000E 01
2	12	0.80000E 01
3	16-D	0.30000E 01 + (0.10000E 01*D)
4	0	0.0
5	22-D	0.10000E 01 + (0.30000E 01*D)
6	12-D	0.90000E 01 + (0.20000E 01*D)
7	26	0.30000E 01
8	25	0.40000E 01
9	17	0.70000E 01
10	6	0.40000E 01

THE CURRENT VALUE OF THE PROJECT COST IS 0.47000E 02 + (0.60000E 01*D).

NEW VALUES OF AEAR FOR J=1,2,...,NK(I)

1	J:	1	2	3	4	5	6	7	8	9
1		2	0							

2	3	0	-5	
3	1	-3	-7	-11
4	-3	-3		
5	1	0	-1	
6	4	-1	-6	
7	-2	-3	-4	-5
8	0	-2		
9	2	0	-1	
10	-4	-4		

*** ITERATION NUMBER 10 ***

THE NODE 2 HAS THE LABEL (1, 2.0, INF).

THE SINK HAS NOT BEEN REACHED WITH INFINITE CAPACITY - CONTINUE WITH THE LABELING PROCESS.
THE NODES THAT HAVE BEEN LABELED WILL RETAIN THAT LABEL FOR THE REMAINDER OF THE ITERATION.

THE NODE 3 HAS THE LABEL (1, 2, 0, 0.10000E 01).

THE NODE 5 HAS THE LABEL (2, 2, 0, 0.10000E 01).

NONBREAKTHROUGH: UPDATE THE PRIMAL VARIABLES;
I.E. DETERMINE OPTIMAL ACTIVITY TIMES FOR LAMBDA = 39.

DELTA (REPRESENTED BY "D") RANGES FROM 0 TO 1.
LAMBDA RANGES FROM 40 TO 39.
THE MINIMUM COST PROJECT SCHEDULE FOR PROJECT DEADLINE = 40-D:
 NODE #: K NEW VALUE: XNODE(K)
 1 0
 2 2
 3 12
 4 15-D
 5 23
 6 40-D

PROJECT COMPLETION TIME = 40-D.

ACTIVITY #:	NEW VALUE: XACT(I)	ACTIVITY COST
1	2	0.80000E 01
2	12	0.80000E 01
3	15-D	0.40000E 01 + (0.10000E 01*D)
4	0	0.0
5	21	0.40000E 01
6	11	0.11000E 02
7	26	0.40000E 01
8	25	0.40000E 01
9	17-D	0.70000E 01 + (0.50000E 01*D)
10	6	0.40000E 01

THE CURRENT VALUE OF THE PROJECT COST IS 0.53000E 02 + (0.60000E 01*D).

NEW VALUES OF ADJN FOR J=1,2,...,NK(I)

I	J:	1	2	3	4	5	6	7	8	9
1		2	0							

2	3	0	-5	
3	2	-2	-6	-10
4	-2	-2		
5	1	0	-1	
6	4	-1	-6	
7	-1	-2	-3	-4
8	0	-2		
9	3	1	0	
10	-4	-4		

*** ITERATION NUMBER 11 ***

THE NODE 2 HAS THE LABEL (1, 2.0, INF).

THE SINK HAS NOT BEEN REACHED WITH INFINITE CAPACITY - CONTINUE WITH THE LABELING PROCESS.
THE NODES THAT HAVE BEEN LABELED WILL RETAIN THAT LABEL FOR THE REMAINDER OF THE ITERATION.

THE NODE 3 HAS THE LABEL (1, 2, 0, 0.10000E 01).

THE NODE 5 HAS THE LABEL (2, 2, 0, 0.10000E 01).

THE NODE 6 HAS THE LABEL (5, 3, 0, 0.10000E 01).

BREAKTHROUGH: UPDATE THE DUAL VARIABLES.

ACTIVITY #:	I	J	NEW FLOW: F(I,J)
	1	1	0.20000E 01
	1	2	0.0
	2	1	0.20000E 01
	2	2	0.0
	2	3	0.0
	3	1	0.10000E 01
	3	2	0.0
	3	3	0.0
	3	4	0.0
	4	1	0.0
	4	2	0.0
	5	1	0.30000E 01
	5	2	0.10000E 01
	5	3	0.0
	6	1	0.20000E 01
	6	2	0.0
	6	3	0.0
	7	1	0.0
	7	2	0.0
	7	3	0.0
	7	4	0.0
	8	1	0.10000E 01
	8	2	0.0
	9	1	0.20000E 01
	9	2	0.30000E 01
	9	3	0.10000E 01
	10	1	0.0
	10	2	0.0

*** ITERATION NUMBER 12 ***

THE NODE 2 HAS THE LABEL (1. 2.0.INF).

THE SINK HAS NOT BEEN REACHED WITH INFINITE CAPACITY - CONTINUE WITH THE LABELING PROCESS.
THE NODES THAT HAVE BEEN LABELED WILL RETAIN THAT LABEL FOR THE REMAINDER OF THE ITERATION.

THE NODE 3 HAS THE LABEL (1. 2. 0. 0.10000E 01).

NONBREAKTHROUGH: UPDATE THE PRIMAL VARIABLES;
I.E. DETERMINE OPTIMAL ACTIVITY TIMES FOR LAMBDA = 38.

DELTA (REPRESENTED BY "D") RANGES FROM 0 TO 1.
LAMBDA RANGES FROM 39 TO 38.
THE MINIMUM COST PROJECT SCHEDULE FOR PROJECT DEADLINE = 39-D:

NODE #: K	NEW VALUE: XNODE(K)
1	0
2	2
3	12
4	14-D
5	23-D
6	39-D

PROJECT COMPLETION TIME = 39-D.

ACTIVITY #: I	NEW VALUE: XACT(I)	ACTIVITY COST
1	2	0.80000E 01
2	12	0.80000E 01
3	14-D	0.50000E 01 + (0.10000E 01*D)
4	0	0.0
5	21-D	0.40000E 01 + (0.40000E 01*D)
6	11-D	0.11000E 02 + (0.20000E 01*D)
7	26	0.30000E 01
8	25	0.40000E 01
9	16	0.12000E 02
10	6	0.40000E 01

THE CURRENT VALUE OF THE PROJECT COST IS 0.59000E 02 + (0.70000E 01*D).

NEW VALUES OF ABAR FOR J=1,2,...,NK(I)

I	J:	1	2	3	4	5	6	7	8	9
1		2	0							
2		3	0	-5						
3		3	-1	-5	-9					
4		-1	-1							
5		2	1	0						
6		5	0	-5						
7		0	-1	-2	-3					
8		0	-2							
9		3	1	0						
10		-4	-4							

*** ITERATION NUMBER 13 ***

THE NODE 2 HAS THE LABEL (1. 2.0.INF).

THE NODE 5 HAS THE LABEL (2, 3.0,INF).

THE NODE 6 HAS THE LABEL (5, 3.0,INF).

* * * * *

THE SINK WAS REACHED WITH INFINITE CAPACITY IMPLYING AN INFEASIBLE SOLUTION TO THE PRIMAL PROBLEM
IF LAMUDA DROPS BELOW ITS CURRENT VALUE. 38.

REFERENCES

1. Fulkerson, D. R. (1961). "A Network Flow Computation for Project Cost Curves," Management Science, 7, 167-178.
2. Hadley, G., Linear Programming. Addison-Wesley Publishing Co., Inc., Reading, Massachusetts, 1963, 221-272.

Appendix: Program Listing

COST001

C ***** COST001
 C COST002
 C THIS PROGRAM IS DESIGNED TO FIND THE MINIMUM PROJECT COST AS A COST002
 C FUNCTION OF PROJECT DEADLINE TIME. CURRENT DIMENSIONS WILL COST003
 C ALLOW A PROJECT NETWORK WITH UP TO 3000 NODES, 3000 ACTIVITIES, COST003
 C AND 11 LEVELS OF COSTS AND TIMES. ALL VARIABLES ARE INTEGER*2. COST004
 C (IF ANY VARIABLE IS NOT ALREADY IN INTEGER FORM, THE VALUES MUST COST004
 C BE RESCALED - THAT IS, MULTIPLIED BY AN APPROPRIATE POWER OF 10 - COST005
 C UNTIL THE VALUES ARE INTEGER.) COST005
 C COST006
 C ***** COST006

C THE INPUT IS AS FOLLOWS (ALL RIGHT=JUSTIFIED): COST007
 C COST007

	COLUMN	DESCRIPTION	
CARD1:	1-4	NUMBER OF NODES	COST008
	6-9	NUMBER OF ACTIVITIES	COST009
	11	OPTION TO SUPPRESS PRINTING OF INPUT = TEST1	COST009
		(0=PRINT, 1=NO PRINT)	COST010
	13	OPTION TO SUPPRESS INTERMEDIATE OUTPUT=TEST3	COST011
		(0=PRINT, 1=NO PRINT)	COST011
	15-18	SOURCE NODE	COST012
	20-23	SINK NODE	COST012
	25	OPTION TO SPECIFY VALUE FOR LAMBDA = TEST3	COST013
		(0=NO, 1=YES AND SEE INTERMEDIATE	COST013
		OUTPUT, 2=YES BUT NO INTERMEDIATE OUTPUT)	COST014

C THE FOLLOWING CARDS ARE IN SETS OF 3-5 CARDS COST014
 C (ONE SET FOR EACH ACTIVITY). COST015
 C COST015

	COLUMN	DESCRIPTION	
CARD1:	1-4	ORIGIN NODE	COST016
	6-9	TERMINAL NODE	COST017
	11-12	NUMBER OF ACTIVITY COMPLETION TIMES	COST017
		AND COSTS THAT ARE READ IN (<=11)	COST018
CARD(S)2-3:	FORMAT 8I10	COMPLETION TIMES (8 ON CARD 2,	COST019
		3 ON CARD 3 IF NEEDED)	COST019
CARD(S)4-5:	FORMAT 8I10	COST ASSOCIATED W/EACH COMPLETION	COST020
		TIME (8 ON CARD 4, 3 ON CARD 5)	COST020

C LAST CARD (USE ONLY IF TEST3 = 1 OR 2): COST021
 C COST021

COLUMN	DESCRIPTION	
1-10	SPECIFIC VALUE OF LAMBDA	COST022

C ***** COST023
 C COST024
 C DEFINITION OF VARIABLES: COST024
 C COST025

ABAR(I,J) = TIME(I,NK(I)+1-J) + XNODE(ORIG(I))-XNODE(TERM(I)) COST025
 C(I,J) = DECREASE IN I TH ACT'S COST PER UNIT FOR J TH TIME COST026
 CAP = MIN(FLOW REACHING ORIGIN NODE, EXCESS CAPACITY TO COST026
 C TERMINAL NODE) COST027
 COST(I,J) = COST OF COMPLETING ACTIVITY I AT TIME(I,J) COST027
 C DEL = MIN(DELTA1,DELTA2) COST028
 C DELTA1 = MIN(-ABAR(I,J) WITH I LABELED AND J UNLABELED, COST028
 C ABAR(I,J)<0) COST029
 C DELTA2 = MIN(ABAR(I,J) WITH I UNLABELED AND J LABELED, COST029
 C ABAR(I,J)>0) COST030
 C DIPEC(J) = DIRECTION OF FLOW REACHING NODE J COST030
 C (0=FORWARD, 1=REVERSE) COST031

```

C      FLOW(I,J) = FLOW IN J TH PIECE OF ACTIVITY I          COST031
C      INF = ANY NUMBER GREATER THAN MAX(CAP)                COST032
C      (CURRENTLY SET AT (2*MAX +1))                          COST032
C      K1(I) = THE NUMBER OF THE TIME-COST PIECE USED IN      COST033
C      LABELING TERM(I) FROM ORIG(I)                         COST033
C      KOUNT = KEEPS TRACK OF ORDER IN WHICH NODES WERE LABELED COST034
C      LABEL(I) = 0 IF NODE I UNLABELED                      COST034
C      1 IF NODE I LABELED                                   COST035
C      LINPUT = SPECIFIC VALUE OF LAMBDA IF TEST3=1 OR 2      COST035
C      NA = TOTAL NUMBER OF ACTIVITIES                       COST036
C      NK(I) = NUMBER OF DIFFERENT TIMES AND COSTS FOR ACTIVITY I COST036
C      NN = TOTAL NUMBER OF NODES                           COST037
C      ORIG(I) = ORIGIN NODE FOR ACTIVITY I                  COST037
C      ORIG2(I) = WHERE THE FLOW IS FROM - USED IN LABELING ONLY COST038
C      PCOST = PROJECT COST FUNCTION                        COST038
C      SINK = NUMBER OF THE SINK NODE                        COST039
C      SOURCE = NUMBER OF THE SOURCE NODE                   COST039
C      TERM(I) = TERMINAL NODE FOR ACTIVITY I                COST040
C      TEST1 = OPTION TO SUPPRESS PRINTING OF INPUT          COST040
C      (0=PRINT, 1=NO PRINT)                                COST041
C      TEST2 = OPTION TO SUPPRESS INTERMEDIATE OUTPUT        COST041
C      (0=PRINT, 1=NO PRINT)                                COST042
C      TEST3 = OPTION TO SPECIFY VALUE FOR LAMBDA            COST042
C      (0=NO, 1=YES AND SEE INTERMEDIATE OUTPUT,            COST043
C      2=YES BUT NO INTERMEDIATE OUTPUT)                    COST043
C      TIME(I,J) = J TH BREAKPOINT (DURATION TIME) FOR ACTIVITY I COST044
C      XACT(I) = ACTIVITY DURATION TIME                     COST044
C      XNODE(I) = NODE TIME                                  COST045
C      XDIFF(I) = XNODE(ORIG(I))-XNODE(TERM(I)), AN UPPER BOUND ON COST045
C      THE ACTIVITY DURATION TIME                          COST046
C      I,J,K,M,N,P = INDICES                                COST046
C      INODE,ITERM,IORIG,IDIFF,ETC.                        COST047
C      = NON-INDEXED VERSIONS OF XNODE(I),TERM(I),XACT(I), COST047
C      ORIG(I),XDIFF(I),ETC.                                COST048
C      COST048
C*****COST049
C      COSTC49
C      DIMENSIONS:                                          COST050
C      NN = TOTAL NUMBER OF NODES                          COST050
C      NA = TOTAL NUMBER OF ACTIVITIES                     COST051
C      MAX = MAX(NK(I))                                     COST051
C      CAP(NN),FLOW( NA,MAX),C(NA,MAX),ORIG(NA),TERM(NA),TIME(NA,MAX),COST052
C      COST(NA,MAX),NK(MAX),ABAR(NA,MAX),XDIFF(NN),XNODE(NN),XACT(NA),COST052
C      DIREC(NN),LABEL(NN),K1(NN),ORIG2(NN),KOUNT(NN),AORD(NA),COST053
C      ND(NN),NDD(NN),IP(NA),CTIME(NA)                    COST053
C      COST054
C*****COST054
C      COST055
C      IMPLICIT INTEGER*2(A-Z)                             COST055
C      REAL*4 CAP(3000),FLOW(3000,11),C(3000,11),PCOST,INF,PCOST1,COST056
C      1K1COST,ACOST ,PNEW                                  COST056
C      COMMON TIME,CTIME,XNODE,ORIG,TERM,AORD,NK,NN,NA,LMIN,LMAX,TEST1,COST057
C      DIMENSION CRIG(3000),TERM(3000),TIME(3000,11),COST(3000,11),COST057
C      1NK(3000), ABAR(3000,11),XDIFF(3000),XNODE(3000),COST058
C      2XACT(3000), DIREC(3000), LABEL(3000),COST058
C      3K1(3000),ORIG2(3000),KCOUNT(3000),AORD(3000),CTIME(3000),COST059
C      4ND(3000),NDD(3000),IP(3000)                       COST059
C      COST060
C      INPUT DATA                                          COST060
C      COST060
C      READ(5,100) NN,NA,TEST1,TEST2,SOURCE,SINK,TEST3    COST061
C      COST061

```



```

INF=0.
PCOST=0.
WRITE(6,228)
IF(TEST1.EQ.1) GO TO 401
WRITE(6,150) NN,NA,SOURCE,SINK
401 DO 12 I=1,NA
  READ(5,230) ORIG(I),TERM(I),NK(I)
  KN=NK(I)
  READ(5,231) (TIME(I,J),J=1,KN)
  READ(5,231) (COST(I,J),J=1,KN)
12 CONTINUE
  CALL ORDER

C
C  SET UP INITIAL VALUES
C

  IF(TEST1.EQ.1) GO TO 193
  K3=1
192 K2=K3+8
  IF(K2.GT.NN) K2=NN
  WRITE(6,151) (K,K=K3,K2)
  WRITE(6,157) (XNODE(K),K=K3,K2)
  IF(K2.GE.NN) GO TO 191
  K3=K2+1
  GO TO 192
191 WRITE(6,152)
193 DO 10 I=1,NA
  LABEL(I)=0
  XDIFF(I)=XNODE(ORIG(I))-XNODE(TERM(I))
  NKM1=NK(I)-1
  KN=NK(I)
  DO 9 J=1,NKM1
    IF(TIME(I,J+1)=TIME(I,J)) 7,8,7
  7 C(I,J)=(COST(I,J)-CCOST(I,J+1))/(TIME(I,J+1)-TIME(I,J))
    GO TO 6
  8 C(I,J)=0.
  6 IF(INF.LT.C(I,J)) INF=C(I,J)
  XACT(I)=XDIFF(I)
  IF(XACT(I).LT.TIME(I,J+1)) XACT(I)=TIME(I,J+1)
  JJ=NK(I)-J+1
  ABAR(I,J)=TIME(I,JJ)+XDIFF(I)
  FLOW(I,J)=0
  9 CONTINUE
  ABAR(I,KN)=TIME(I,1)+XDIFF(I)
  FLOW(I,KN)=0
  IF(TEST1.EQ.1) GO TO 10
  WRITE(6,153) I,XACT(I),ORIG(I),TERM(I),(J,TIME(I,J),COST(I,J),
1 C(I,J),ABAR(I,J),J=1,NKM1)
  WRITE(6,156) KN,TIME(I,KN),COST(I,KN),ABAR(I,KN)
10 CONTINUE
  INF=2.*INF+1.
  DO 417 I=1,NA
  C(I,NK(I))=0.
  NKM1=NK(I)-1
  PCOST1=0.
  IKK=0
  DO 418 K=1,NKM1
  IF(K.NE.1) GO TO 40
  XIJ=XACT(I)
  IF(XIJ.GT.TIME(I,2)) XIJ=TIME(I,2)
  GO TO 41
40 XIJ=XACT(I)-TIME(I,K)

```

[illegible]


```

IF(XIJ.LT.0) XIJ=0
IF(XIJ.GT.(TIME(I,K+1)-TIME(I,K))) XIJ=TIME(I,K+1)-TIME(I,K)
IF(IIK.EQ.1) GO TO 41
IF(C(I,K).GT.C(I,K-1)) GO TO 50
41 PCOST1=PCOST1+C(I,K)*XIJ
GO TO 418
50 IIK=1
WRITE(6,237) I,I
PCOST1=PCOST1+C(I,K)*XIJ
418 CONTINUE
PCOST=PCOST+COST(I,1)+C(I,1)*TIME(I,1)-PCOST1
PNEW=PCOST
417 CONTINUE
LAMBDA=LMAX
IF (TEST3.GE.1) GO TO 700
WRITE(6,154)
LINPUT=0
GO TO 96
700 READ(5,232) LINPUT
IF(LINPUT.LT.LMIN) GO TO 705
IF(LINPUT.GE.LMAX) GO TO 704
IF (TEST3.EQ.2) GO TO 724
WRITE(6,155) LINPUT
96 WRITE(6,200) LAMBDA, PCOST
IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 724
WRITE(6,235)
724 CAP(SOURCE)=INF
ITER=0
99 LABEL(SOURCE)=1
IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 97
ITER=ITER+1
WRITE(6,225) ITER
C
C INITIAL LABELING ITERATION
C
97 I=1
J=SOURCE
M=0
C IF ACTIVITY STARTS AT DESIGNATED ORIGIN, TRY TO LABEL,
C OTHERWISE, CHANGE ORIGINS.
14 IF (ORIG(I).NE.J) GO TO 13
ITERM=TERM(I)
C CHECK IF NCDE ALREADY LABELED AND
C CHECK IF ABAR(I,NK(I))=0.
IF (LABEL(ITERM).NE.0.OR.ABAR(I,NK(I)).NE.0) GO TO 13
C IF NODE NOT ALREADY LABELED AND ABAR(I,NK(I))=0,
C PROCEED WITH LABELING.
LABEL(ITERM)=1
ORIG2(ITERM)=J
K1(ITERM)=NK(I)
DIREC(ITERM)=0
CAP(ITERM)=INF
IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 403
WRITE(6,201) ITERM,ORIG2(ITERM),K1(ITERM)
C IF CAN REACH SINK, TERMINATE (IMPLIES INFEASIBLE)
403 IF (ITERM.EQ.SINK) GO TO 15
M=M+1
KOUNT(M)=ITERM
C IF EVERY PATH TESTED AND INFINITE FLOW NOT POSSIBLE,
C GO ON TO LABELING PART(II).
13 I=I+1

```

COST092
COST093
COST093
COST094
COST094
COST095
COST095
COST096
COST096
COST097
COST097
COST098
COST098
COST099
COST099
COST100
COST100
COST101
COST101
COST102
COST102
COST103
COST103
COST104
COST104
COST105
COST105
COST106
COST106
COST107
COST107
COST108
COST108
COST109
COST109
COST110
COST110
COST111
COST111
COST112
COST112
COST113
COST113
COST114
COST114
COST115
COST115
COST116
COST116
COST117
COST117
COST118
COST118
COST119
COST119
COST120
COST120
COST121
COST121
COST122
COST122

IF (I.GT.NA) GO TO 11	COST123
GO TO 14	COST123
C CHANGE DESIGNATED ORIGINS.	COST124
11 IF (J.EQ.SOURCE) P=1	COST124
IF(P.GT.M) GO TO 16	COST125
C IF ALL LABELED NODES HAVE BEEN SCANNED AND NO NEW NODES	COST125
C HAVE BEEN LABELED, GO ON TO LABELING PART (II).	COST126
J=KOUNT(P)	COST126
P=P+1	COST127
I=1	COST127
GO TO 14	COST128
15 IF(TEST3.GE.1) GO TO 404	COST128
WRITE(6,202) LAMBDA	COST129
404 GO TO 999	COST129
16 IF(TEST2.EG.1.OR.TEST3.GE.1) GO TO 405	COST130
WRITE(6,203)	COST130
C	COST131
C NEXT LABELING PROCEDURE	COST131
C	COST132
405 I=1	COST132
J=SOURCE	COST133
C AGAIN, CHECK ALL CONDITIONS FOR LABELING	COST133
C IE. CHECK IF NODE IS ALREADY LABELED, IF ABAR(I,J)=0, AND	COST134
C IF THE FLOW(I,J) IS LESS THAN ITS UPPER BOUND.	COST134
20 IF (ORIG(I).NE.J) GO TO 24	COST135
ITERM=TERM(I)	COST135
KN=NK(I)	COST136
DO 25 K=1,KN	COST136
IF (K.EQ.KN) GO TO 27	COST137
IF(LABEL(ITERM).NE.0.OR.ABAR(I,K).NE.0.OR.FLOW(I,K).GE.	COST137
1(C(I,NK(I)=K)=C(I,NK(I)=K+1)))GO TO 25	COST138
DIREC(ITERM)=0	COST138
C CAPACITY IS MIN OF PREVIOUS FLOW AND THE EXCESS CAPACITY	COST139
CAP(ITERM)=C(I,NK(I)=K)-C(I,NK(I)=K+1) - FLOW(I,K)	COST139
GO TO 23	COST140
27 IF(LABEL(ITERM).NE.0.OR.ABAR(I,K).NE.0.OR.FLOW(I,K).GE.INF)	COST140
1 GO TO 25	COST141
C IF THE NODE HAS NOT ALREADY BEEN LABELED, ABAR(I,J)=0, AND	COST141
C THE FLOW IS LESS THAN ITS UPPER BOUND, PROCEED WITH THE LABELING	COST142
C OF THE NODE.	COST142
DIREC(ITERM)=0	COST143
CAP(ITERM)=INF	COST143
23 LABEL(ITERM)=1	COST144
ORIG2(ITERM)=J	COST144
K1(ITERM)=K	COST145
IF (CAP(ITERM).GT.CAP(ORIG(I))) CAP(ITERM)=CAP(ORIG(I))	COST145
IF(TEST2.EG.1.OR.TEST3.GE.1) GO TO 406	COST146
WRITE(6,204) ITERM,ORIG2(ITERM),K1(ITERM),DIREC(ITERM),CAP(ITERM)	COST146
C IF SINK LABELED, GO TO UPDATE PROCEDURE	COST147
406 IF (ITERM.EQ.SINK) GO TO 21	COST147
M=M+1	COST148
KOUNT(M)=ITERM	COST148
C CHECK IF ALL PATHS TRIED	COST149
25 CONTINUE	COST149
GO TO 19	COST150
24 IF(TERM(I).NE.J) GO TO 19	COST150
IORIG=ORIG(I)	COST151
KN=NK(I)	COST151
DO 26 K=1,KN	COST152
IF(LABEL(IORIG).NE.0.OR.ABAR(I,K).NE.0.OR.FLOW(I,K).LE.0)	COST152
2 GO TO 26	COST153


```

DIREC(IORIG)=1
CAP(IORIG)=FLCW(I,K)
LABEL(IORIG)=1
ORIG2(IORIG)=J
K1(IORIG)=K
IF(CAP(IORIG).GT.CAP(TERM(I))) CAP(IORIG)=CAP(TERM(I))
IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 402
WRITE(6,204) IORIG,ORIG2(ICRIG),K1(IORIG),DIREC(IORIG),CAP(IORIG)
402 M=M+1
KOUNT(M)=ICRIG
26 CONTINUE
19 I=I+1
IF (I.GT.NA) GO TO 18
GO TO 20
18 IF (J.EQ.SOURCE) P=1
IF (P.GT.M) GO TO 22
J=KOUNT(P)
P=P+1
I=1
GO TO 20

C
C NONBREAKTHROUGH HAS OCCURED. DELTAS ARE FOUND AND UPDATING
C MADE IN THE XNODES AND XACTS.
C

22 DELTA1=INF+1
DELTA2=INF+1
DO 4 I=1,NA
KN=NK(I)
IF (LABEL(ORIG(I)).EQ.1.AND.LABEL(TERM(I)).EQ.0) GO TO 1
C A1 IS SET CF I LAELED AND J UNLAELED,
C A2 IS SET OF I UNLAELED AND J LAELED.
IF(LABEL(ORIG(I)).EQ.0.AND.LABEL(TERM(I)).EQ.1) GO TO 2
GO TO 4
C FINDING DELTA1'S.
1 DO 3 J=1,KN
IF (ABAR(I,J).GE.0) GO TO 3
IF (-ABAR(I,J).LT.DELTA1) DELTA1=-ABAR(I,J)
3 CONTINUE
GO TO 4
C FINDING DELTA2'S
2 DO 5 J=1,KN
IF(ABAR(I,J).LE.0) GO TO 4
IF (ABAR(I,J).LT.DELTA2) DELTA2= ABAR(I,J)
5 CONTINUE
4 CONTINUE
C DEL=MIN(DELTA1,DELTA2)
DEL=DELTA1
IF (DELTA2.LT.DEL) DEL=DELTA2
LAMBDA=LAMBDA+DEL
C UPDATING THE XNODES.
IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 407
WRITE(6,206) LAMBDA
407 IF (TEST3.EQ.2) GO TO 721
DELTA= LAMBDA + DEL
WRITE(6,209) DEL,DELTA,LAMBDA,DELTA
721 IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 408
WRITE(6,207)
408 DO 80 I=1,NA
INODE=XNODE(I)
IF(LABEL(I).EQ.0) GO TO 81
IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 409

```

COST153
COST154
COST154
COST155
COST155
COST156
COST156
COST157
COST157
COST158
COST158
COST159
COST159
COST160
COST160
COST161
COST161
COST162
COST162
COST163
COST163
COST164
COST164
COST165
COST165
COST166
COST166
COST167
COST167
COST168
COST168
COST169
COST169
COST170
COST170
COST171
COST171
COST172
COST172
COST173
COST173
COST174
COST174
COST175
COST175
COST176
COST176
COST177
COST177
COST178
COST178
COST179
COST179
COST180
COST180
COST181
COST181
COST182
COST182
COST183
COST183


```

WRITE(6,210) I,INODE
409 XNODE(I)=INODE
GO TO 80
81 IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 410
WRITE(6,211) I,INODE
410 XNODE(I)=INODE=DEL
80 CONTINUE
IF (TEST3.EQ.2) GO TO 722
WRITE(6,212) DELTA
722 PCOST=0.
DO 82 I=1,NA
IP(I)=0
PCOST1=0.
NKM1=NK(I)-1
IACT=TIME(I,NK(I))
IORIG=ORIG(I)
ITERM=TERM(I)
IDIFF=XNODE(ITERM)-XNODE(ICRIG)
XDIFF(I)=IDIFF
IF (IDIFF.GE.IACT) GO TO 86
XACT(I)=IDIFF
DO 550 K=1,NKM1
IF(K.NE.1) GO TO 43
XIJ=XACT(I)
IF(XIJ.GT.TIME(I,2)) XIJ=TIME(I,2)
FLAG1=0
GO TO 42
43 XIJ=XACT(I)-TIME(I,K)
IF(XIJ.LT.0) GO TO 552
IF(XIJ.GT.(TIME(I,K+1)-TIME(I,K))) XIJ=TIME(I,K+1)-TIME(I,K)
FLAG1=0
GO TO 42
552 FLAG1=1
FLAG2=K-1
GO TO 553
42 PCOST1=PCOST1+C(I,K)*XIJ
550 CONTINUE
553 KCOST=COST(I,1)+C(I,1)*TIME(I,1)
ACOST=K COST-PCOST1
PCOST=PCOST+ACOST
IF (TEST3.EQ.2) GO TO 82
IF (LABEL(IORIG)=LABEL(ITERM)) 83,84,85
83 IDIFF=IDIFF=DEL
IF(FLAG1.EQ.1) GO TO 59
ACOST=ACOST + C(I,NKM1)*DEL
IP(I)=1
WRITE(6,214) I,IDIFF,ACOST,C(I,NKM1)
GO TO 82
59 ACOST=ACOST+C(I,FLAG2)*DEL
IP(I)=1
WRITE(6,214) I,IDIFF,ACOST,C(I,FLAG2)
GO TO 82
84 WRITE(6,216) I,XACT(I),ACOST
GO TO 82
85 IDIFF=IDIFF+DEL
IF(FLAG1.EQ.1) GO TO 58
ACOST=ACOST - C(I,NKM1)*DEL
IP(I)=2
WRITE(6,213) I,IDIFF,ACOST,C(I,NKM1)
GO TO 82
58 ACOST=ACOST-C(I,FLAG2)*DEL

```

COST1840
COST1845
COST1850
COST1855
COST1860
COST1865
COST1870
COST1880
COST1885
COST1890
COST1895
COST1900
COST1905
COST1910
COST1915
COST1920
COST1925
COST1930
COST1935
COST1940
COST1945
COST1950
COST1955
COST1960
COST1965
COST1970
COST1975
COST1980
COST1985
COST1990
COST1995
COST2000
COST2005
COST2010
COST2015
COST2020
COST2025
COST2030
COST2035
COST2040
COST2045
COST2050
COST2055
COST2060
COST2065
COST2070
COST2075
COST2080
COST2085
COST2090
COST2095
COST2100
COST2105
COST2110
COST2115
COST2120
COST2125
COST2130
COST2135
COST2140
COST2145

COST219
COST219
COST219
COST219
COST219
COST219
COST219
COST219
COST219
COST219
COST219
COST220
COST221
COST221
COST222
COST222
COST223
COST223
COST224
COST224
COST225
COST225
COST226
COST226
COST227
COST227
COST228
COST228
COST229
COST229
COST230
COST230
COST231
COST231
COST232
COST232
COST233
COST233
COST234
COST234
COST235
COST235
COST236
COST236
COST237
COST237
COST238
COST238
COST239
COST239
COST240
COST240
COST241
COST241
COST242
COST242
COST243
COST243
COST244
COST244
COST244

C	CHECK IF DIRECTION OF FLOW IS POSITIVE OR NEGATIVE,	COST245
C	ALSO CHECK IF CAPACITY IS INFINITE.	COST246
31	IF(CAP(ITERM).EQ.INF) GO TO 33	COST246
	IF (DIREC(ITERM).EQ.0) GO TO 34	COST247
	FLOW(I,K1(ITERM))=FLOW(I,K1(ITERM))-CAP(ITERM)	COST247
	GO TO 30	COST248
C	RELABEL AND START OVER.	COST248
33	IF(TEST2.EQ.1.OR.TEST3.GE.1) GO TO 415	COST249
	DO 560 I=1,NA	COST249
	NK1=NK(I)	COST250
	DO 560 K=1,NK1	COST250
560	WRITE(6,220) I,K,FLOW(I,K)	COST251
415	DO 98 I=1,NN	COST251
	LABEL(I)=0	COST252
98	CONTINUE	COST252
	GO TO 99	COST253
C	PROGRAM TERMINATES WHEN EVENTUALLY AN INFINITE FLOW IS ACHIEVED	COST253
C	FROM THE SOURCE TO THE SINK, OR WHEN THE VALUE OF LAMBDA DROPS	COST254
C	BELOW THE MINIMUM LENGTH OF THE NETWORK.	COST254
998	IF(TEST3.NE.0) GO TO 999	COST255
	WRITE(6,202) LAMBDA	COST255
	GO TO 999	COST256
705	WRITE(6,233) LINPUT,LMIN	COST256
	GO TO 999	COST257
704	WRITE(6,236) LINPUT,LMAX	COST257
	WRITE(6,238) LINPUT	COST258
	D=0	COST258
	DO 60 I=1,NA	COST259
60	IP(I)=0	COST259
	GO TO 707	COST260
703	WRITE(6,234) LINPUT	COST260
706	WRITE(6,238) LINPUT	COST261
	D=LINPUT=LAMBDA	COST261
707	PCOST=0.	COST262
	DO 57 I=1,NA	COST262
	IF(IP(I).EQ.1.AND.D.GT.0) XACT(I)=XACT(I)-D	COST263
	IF(IP(I).EQ.2.AND.D.GT.0) XACT(I)=XACT(I)+D	COST263
	PCOST1=0.	COST264
	NKM1=NK(I)-1	COST264
	DO 51 K=1,NKM1	COST265
	IF(K.NE.1) GO TO 52	COST265
	XIJ=XACT(I)	COST266
	IF(XIJ.GT.TIME(I,2)) XIJ=TIME(I,2)	COST266
	GO TO 53	COST267
52	XIJ=XACT(I)-TIME(I,K)	COST267
	IF(XIJ.LT.0) XIJ=0	COST268
	IF(XIJ.GT.(TIME(I,K+1)-TIME(I,K))) XIJ=TIME(I,K+1)-TIME(I,K)	COST268
53	PCOST1=PCOST1+C(I,K)*XIJ	COST269
51	CONTINUE	COST269
	KCOST=COST(I,1)+C(I,1)*TIME(I,1)	COST270
	ACOST=KCOST-PCOST1	COST270
	WRITE(6,216) I,XACT(I),ACOST	COST271
57	PCOST=PCOST+ACOST	COST271
	WRITE(6,239) PCOST	COST272
999	WRITE(6,228)	COST272
	STOP	COST273
100	FORMAT(I4,1X,I4,1X,I1,1X,I1,1X,I4,1X,I4,1X,I1)	COST273
150	FORMAT('=', 'THE NUMBER OF NODES IS ', I4, ' ', '/', I1, 'THE NUMBER OF ACCOST2740	
	1TIVITIES IS ', I4, ' ', '/', I1, 'THE SOURCE NODE IS NUMBERED ', I4, ' AND COST2745	
	2THE SINK NODE IS NUMBERED ', I4, ' ', '/', ' ', ' ** NCDES: **')	COST2750
151	FORMAT('0', 16X, 'K', 7X, 9(3X, I4, 5X))	COST2755


```

152 FORMAT('=', ' ** ACTIVITIES: **', //, 6X, 'I', 7X, 'XACT', 6X, 'ORIG', 3X, COST27
1   'TERM', 4X, 'J', 6X, 'TIME', 5X, 'COST', 14X, 'C', 13X, 'ABAR') COST27
153 FORMAT(' ', 3X, I4, 3X, I10, 3X, I4, 3X, I4, (T39, I2, 3X, I10, 3X, I10, 3X, COST27
1   IE16.5, 3X, I10)) COST27
154 FORMAT('=', 'THE ENTIRE PROJECT COST CURVE IS GOING TO BE DETERMINECOST27
1   ID.') COST27
155 FORMAT('=', 'THE OPTIMAL ACTIVITY COMPLETION TIMES FOR A SPECIFIED COST27
1   PROJECT DEADLINE TIME = ', I10, ' ARE GOING TO BE DETERMINED.') COST27
156 FORMAT(' ', T39, I2, 3X, I10, 3X, I10, 22X, I10) COST280
157 FORMAT('0', 4X, 'INITIAL XNODE(K)', 3X, COST280
1   I9(I10, 2X)) COST281
200 FORMAT('0', 'LAMBDA = PROJECT COMPLETION TIME', //, COST281
1   1 1X, 'THE STARTING VALUE OF LAMEDA IS ', I10, ' ', // COST282
2   1X, 'THE CORRESPONDING TOTAL PROJECT COST IS ', E16.5, ' ') COST282
201 FORMAT('0', 'THE NODE ', I4, ' HAS THE LABEL (', I4, ' ', I4, ' ', 0, INF), ') COST283
202 FORMAT('0', //, //, 30X, ' * * * * ', //, //, 1X, COST283
1   'THE SINK WAS REACHED WITH INFINITE CAPACITY IMPLYING ACOST284
1   IN INFEASIBLE SOLUTION TO THE PRIMAL PROBLEM ', //, 20X, 'IF LAMBDA DROCOST284
2   PS BELOW ITS CURRENT VALUE, ', I10, ' ') COST285
203 FORMAT('=', 'THE SINK HAS NOT BEEN REACHED WITH INFINITE CAPACITY -COST285
1   CONTINUE WITH THE LABELING PROCESS.', / 1X, 'THE NODES THAT HAVE COST286
2   BEEN LABELED WILL RETAIN THAT LABEL FOR THE REMAINDER OF THE ITERACOST286
3   TION.') COST287
204 FORMAT('0', 'THE NODE ', I4, ' HAS THE LABEL (', I4, ' ', I4, ' ', I4, ' ', COST287
1   IE16.5, ' ') COST288
205 FORMAT('=', 'BREAKTHROUGH: UPDATE THE DUAL VARIABLES,' COST288
1   1, //, 1X, ' ACTIVITY #: I', 3X, 'J', 9X, 'NEW FLOW: F(I, J)') COST289
206 FORMAT('=', 'NONBREAKTHROUGH: UPDATE THE PRIMAL VARIABLES:', //, 1X, COST289
1   'I.E. DETERMINE OPTIMAL ACTIVITY TIMES FOR LAMBDA = ', I10, ' ') COST290
207 FORMAT(' ', ' NODE #: K', 5X, 'NEW VALUE: XNODE(K)') COST290
209 FORMAT('=', 'DELTA (REPRESENTED BY "D") RANGES FROM 0 TO' COST291
1   1, I4, ' ', //, 1X, 'LAMBDA RANGES FROM', I10, ' TO', I10, ' ', COST291
2   /, 1X, 'THE MINIMUM COST PROJECT SCHEDULE FOR PROCCOST292
3   JECT DEADLINE = ', I10, '=D:') COST292
210 FORMAT(' ', 7X, I4, 12X, I10) COST293
211 FORMAT(' ', 7X, I4, 12X, I10, '=D') COST293
212 FORMAT('=', 'PROJECT COMPLETION TIME = ', I10, '=D', //, 1X, COST294
1   ' ACTIVITY #: I', 3X, 'NEW VALUE: XACT(I)', 9X, 'ACTIVITY COCOST294
2   ST') COST295
213 FORMAT(' ', 5X, I4, 12X, I10, '=D', 9X, E16.5, ' + (', E13.5, '*D)') COST295
214 FORMAT(' ', 5X, I4, 12X, I10, '=D', 9X, E16.5, ' + (', E13.5, '*D)') COST296
216 FORMAT(' ', 5X, I4, 12X, I10, 11X, E16.5) COST296
220 FORMAT(' ', 12X, I4, 2X, I2, 7X, E16.5) COST297
224 FORMAT('0', 'THE CURRENT VALUE OF THE PROJECT COST IS ', E16.5, COST297
1   ' + (', E13.5, '*D), ') COST298
225 FORMAT('=', '*** ITERATION NUMBER', I6, ' ***') COST298
226 FORMAT('=', 'NEW VALUES OF ABAR FOR J=1,2,...,NK(I)', //, 6X, 'I', 3X, COST299
1   'J:', 11(5X, I2, 3X)) COST299
227 FORMAT(' ', 2X, I4, 7X, 11(I8, 2X)) COST300
228 FORMAT(1H1) COST300
230 FORMAT(I4, 1X, I4, 1X, I2) COST301
231 FORMAT(8I10) COST301
232 FORMAT(I10) COST302
233 FORMAT('=', 'THE SPECIFIED VALUE OF LAMBDA, ', I10, ' IS LESS THAN THECOST302
1   MINIMUM VALUE, ', I10, ' IMPLYING AN INFEASIBLE SOLUTION.', //, 1X, COST303
2   'THE PROBLEM WILL NOT BE WORKED.') COST303
234 FORMAT('1', 'THE SPECIFIED VALUE OF LAMBDA, ', I10, ' HAS BEEN REACHEDCOST304
1   ') COST304
235 FORMAT('0', 'THE SOURCE HAS A VALUE OF ZERO AND IS ASSIGNED THE COST305
3   LABEL (=, -, -, INF), // ) COST305
236 FORMAT('=', 'THE SPECIFIED VALUE OF LAMBDA, ', I10, ' IS GREATER THANCOST306

```

AD-A071 498

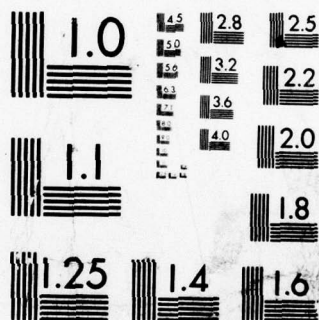
TEXAS A AND M UNIV COLLEGE STATION INST OF STATISTICS F/G 15/5
STATISTICAL PERT: AN IMPROVED PROJECT SCHEDULING ALGORITHM.(U)
FEB 77 C S DUNN, R L SIELKEN N00014-76-C-0038
THEMIS-TR-55 NL

UNCLASSIFIED

2 OF 2
AD
A071-98



END
DATE
FILMED
8-79
DOC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

	1 OR EQUAL TO THE MAXIMUM VALUE. ',I10.',',/',	COST3065
	1 IX,'THEREFORE, THE ORIGINAL COST3070	
	2XNODE(K)''S AND XACT(I)''S ARE OPTIMAL.') COST3075	
237	FORMAT(' ','** WARNING: ACTIVITY NUMBER ',I4,' HAS A NON-CONVEX CCOST3080	
	1OST FUNCTION;',/,12X,'IE. THE C(',I4,',M)''S ARE NOT NON-INCREASINCOST3085	
	2G.'') COST3090	
238	FORMAT('=','FOR PROJECT COMPLETION TIME = ',I10.', THE OPTIMAL SOLCOST3095	
	1UTION IS:',/,1X, COST3100	
	1 ' ACTIVITY #: I',3X,'NEW VALUE: XACT(I)',9X,'ACTIVITY COCOST3105	
	2ST') COST3110	
239	FORMAT('=','THE CORRESPONDING PROJECT COST IS ',E16.5,',') COST3115	
	END COST3120	
	SUBROUTINE CRDER COST3125	
C		COST3130
C	THIS SUBRCUTINE DETERMINES THE ORDER IN WHICH TO CONSIDER COST3135	
C	THE ACTIVITIES FOR THE CALCULATION OF THE CRITICAL PATH TIME COST3140	
C	DIMENSIONS:	COST3145
C	NA=M= THE NUMBER OF ACTIVITIES IN THE NETWORK COST3150	
C	NN=N= THE NUMBER OF NODES IN THE NETWORK COST3155	
C	ORIG(NA),TERM(NA),AORD(NA),CTIME(NA),XNODE(NN),ND(NN),NDD(NN), COST3160	
C	TIME(NA,MAX),NK(MAX) COST3165	
C		COST3170
	IMPLICIT INTEGER*2(A-Z) COST3175	
	COMMON TIME,CTIME,XNODE,ORIG,TERM,AORD,NK,NN,NA,LMIN,LMAX,TEST1 COST3180	
	DIMENSION ORIG(3000),TERM(3000),AORD(3000),CTIME(3000). COST3185	
	1XNODE(3000),ND(3000),NDD(3000),TIME(3000,11),NK(3000) COST3190	
	N=NN COST3195	
	M=NA COST3200	
	NDD(1)=1 COST3205	
	DO 5 I=2,N COST3210	
5	NDD(I)=0 COST3215	
	DO 6 I=1,M COST3220	
6	AORD(I)=0 COST3225	
	K=0 COST3230	
	MP=M+1 COST3235	
	DO 1 II=1,MP COST3240	
	DO 20 I=1,N COST3245	
20	ND(I)=NDD(I) COST3250	
	III=0 COST3255	
	IP=II+1 COST3260	
	DO 2 J=1,M COST3265	
	IF(ND(ORIG(J)).NE.II) GO TO 2 COST3270	
	NDD(TERM(J))=IP COST3275	
	III=1 COST3280	
	IF(K.EQ.0) GO TO 14 COST3285	
	DO 10 L=1,K COST3290	
	IF(AORD(L).EQ.J) GO TO 11 COST3295	
10	CONTINUE COST3300	
14	K=K+1 COST3305	
	GO TO 13 COST3310	
11	IF(L.EC.K) GC TO 2 COST3315	
	KM=K+1 COST3320	
	DO 12 LL=L,KM COST3325	
12	AORD(LL)=ACRD(LL+1) COST3330	
13	AORD(K)=J COST3335	
2	CONTINUE COST3340	
	IF(III.EQ.0) GO TC 3 COST3345	
1	CONTINUE COST3350	
3	CONTINUE COST3355	
	DO 30 I=1,NA COST3360	
30	CTIME(I)=TIME(I,1) COST3365	

```

LMIN=CPTIME(CPATHT)
DO 31 I=1,NA
  NK1=NK(I)
31 CTIME(I)=TIME(I,NK1)
  LMAX=CPTIME(CPATHT)
  RETURN
  END
  FUNCTION CPTIME(CPATHT)

C
C      DETERMINE THE CRITICAL PATH TIME: CPTIME
C      XNODE(I) = EARLIEST TIME THAT AN ACTIVITY BEGINNING AT NODE I
C      CAN COMMENCE
C      DIMENSIONS:
C      NA=M= THE NUMBER OF ACTIVITIES IN THE NETWORK
C      NN=N= THE NUMBER OF NODES IN THE NETWORK
C      ORIG(NA),TERM(NA),AORD(NA),CTIME(NA),XNODE(NN),ND(NN),NDD(NN),
C      TIME(NA,MAX),NK(MAX)
C
      IMPLICIT INTEGER*2(A-Z)
      COMMON TIME,CTIME,XNODE,ORIG,TERM,AORD,NK,NN,NA,LMIN,LMAX,TEST1
      DIMENSION ORIG(3000),TERM(3000),AORD(3000),CTIME(3000),
1 XNODE(3000),ND(3000),NDD(3000),TIME(3000,11),NK(3000)
      DO 1 I=1,NA
1      XNODE(I)=0
      DO 2 II=1,NA
      I=AORD(II)
2      IF(XNODE(ORIG(I))+CTIME(I).GT.XNODE(TERM(I)))
1      XNODE(TERM(I))=XNODE(ORIG(I))+CTIME(I)
      CPTIME=XNODE(NN)
      RETURN
      END

```

COST337
 COST337
 COST338
 COST338
 COST339
 COST339
 COST340
 COST340
 COST341
 COST341
 COST342
 COST342
 COST343
 COST343
 COST344
 COST344
 COST345
 COST345
 COST346
 COST346
 COST347
 COST347
 COST348
 COST348
 COST349
 COST349
 COST350
 COST350
 COST351
 COST351
 COST352

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

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

1. ORIGINATING ACTIVITY (Corporate author)		2a. REPORT SECURITY CLASSIFICATION	
Texas A&M University		Unclassified	
		2b. GROUP	
		Unclassified	
3. REPORT TITLE			
STATISTICAL PERT: AN IMPROVED PROJECT SCHEDULING PROGRAM			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates)			
Progress Report			
5. AUTHOR(S) (First name, middle initial, last name)			
Cynthia S. Dunn and Robert L. Sielken, Jr.			
6. REPORT DATE		7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
February 3, 1977		104	2
8a. CONTRACT OR GRANT NO.		9a. ORIGINATOR'S REPORT NUMBER(S)	
N00014-76-C-0038 <i>new</i>		Number 55	
A. PROJECT NO.		9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
NRO47-700			
C.			
d.			
10. DISTRIBUTION STATEMENT			
This document has been approved for public release and sale; its distribution is unlimited.			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY	
		Office of Naval Research	
13. ABSTRACT			
See page following Attachment II			

ATTACHMENT III

Unclassified

Security Classification

14.	KEY WORDS	LINK A		LINK B		LINK C	
		ROLE	WT	ROLE	WT	ROLE	WT
	Project Scheduling Network-flow Algorithm Minimum Cost Schedule PERT						

ATTACHMENT III (continued)

BASIC DISTRIBUTION LIST FOR
UNCLASSIFIED TECHNICAL REPORTS

	Copies		Copies
Operations Research Office of Naval Research (Code 434) Arlington, Virginia 22217	3	Director Office of Naval Research Branch Office 495 Summer Street Boston, Massachusetts 02210 Attn: Dr. A. L. Powell	1
Director, Naval Research Laboratory Attn: Library, Code 2029 (ONRL) Washington, D.C. 20390	2	Director Office of Naval Research Branch Office 219 South Dearborn Street Chicago, Illinois 60604 Attn: Dr. A. R. Dawe	1
Defense Documentation Center Cameron Station Alexandria, Virginia 22314	12	Office of Naval Research San Francisco Area Office 760 Market St. - Room 447 San Francisco, California 94103	1
Defense Logistics Studies Information Exchange Army Logistics Management Center Fort Lee, Virginia 23801	1	Technical Library Naval Ordnance Station Indian Head, Maryland 20640	1
Technical Information Center Naval Research Laboratory Washington, D.C. 20390	6	Bureau of Naval Personnel Navy Department Technical Library Washington, D.C. 20370 STOP 82	2
Director, Office of Naval Research Branch Office 1030 East Green Street Pasadena, California 91101 Attn: Dr. A. R. Laufer	1	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	1
Lt. Dennis Johnston - ED3 Manned Spacecraft Center NASA Houston, Texas 77058	1	Naval Ordnance Station Louisville, Kentucky 40214	1
Mr. William Dejka Code 360 Naval Electronics Lab Center San Diego, California 92132	1	Library Naval Electronics Laboratory Center San Diego, California 92152	1
Professor Lucien A. Schmit, Jr. Room 6731, Boelter Hall School of Engrng. & Applied Sci. University of California Los Angeles, California 90024	1	Naval Ship Engineering Center Philadelphia Division Technical Library Philadelphia, Pennsylvania 19112	1
		Dr. A. L. Slafkosky Scientific Advisor Commandant Marine Corps (Code AX) Washington, D.C. 20380	1

	Copies		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	Dr. Paul Murrill Project Themis Department of Chemical Engineering Louisiana State University Baton Rouge, Louisiana 70803	1
Department of Statistics Syntex Research 3401 Hillview Palo Alto, California 94304 Attn: Prof. Stuart Bessler	1	Director Army Materials & Mechanics Research Center Attn: Mr. J. Bluhm Watertown, Massachusetts 02172	1
University of California Department of Engineering Los Angeles, California 90024 Attn: Prof. R. R. O'Neill	1	Commanding Officer U.S. Army Ballistic Research Laboratories Attn: Dr. John H. Giese Aberdeen Proving Ground Maryland 21005	1
Dr. Thomas Varley Asst. Chief for Technology Office of Naval Research, Code 200 Arlington, Virginia 22217	1	University of Iowa Department of Mechanics and Hydraulics Iowa City, Iowa 52240 Attn: Prof. W. F. Ames	1
Maritime Transportation Research Board National Academy of Sciences 2101 Constitution Avenue Washington, D.C. 20418 Attn: RADM J.B. Oren USCG (RET)	1	Office of Naval Research Resident Representative 1105 Guadalupe Lowich Building Austin, Texas 78701 Attn: Mr. Francis M. Lucas	1
Stanford University Department of Operations Research Stanford, California 94305 Attn: Prof. A. F. Veinott, Jr.	1	Dr. Jerome Bracken Institute of Defense Analyses 400 Army-Navy Drive Arlington, Virginia 22202	1
Case Western Reserve University Cleveland, Ohio 44106 Attn: Prof. B. V. Dean	1	Professor Richard L. Fox School of Engineering Case Western Reserve University Cleveland, Ohio 44106	1
University of California Center for Research in Management Science Berkeley, California 94720 Attn: Prof. W. I. Zangwill	1	Director, National Security Agency Ft. George G. Meade, Maryland 20755 Attn: Dr. Joseph Blum, R44	1
U. S. Naval Postgraduate School Department of Operations Research and Economics Monterey, California 93940 Attn: Prof. C. R. Jones	1		

	Copies		Copies
Naval Applied Science Laboratory Technical Library, Code 222 Flushing & Washington Avenues Brooklyn, New York 11251	1	Harvard University Department of Statistics Cambridge, Massachusetts 02139 Attn: Prof. W. G. Cochran	1
Naval Undersea Warfare Center 3202 E. Foothill Boulevard Pasadena, California 91107 Technical Library	1	Columbia University Department of Industrial Engineering New York, New York 10027 Attn: Prof. C. Derman	1
University of Chicago Statistical Research Center Chicago, Illinois 60637 Attn: Prof. W. Kruskal	1	New York University Institute of Mathematical Sciences New York, New York 10453 Attn: Prof. W. M. Hirsch	1
Stanford University Department of Statistics Stanford, California 94305 Attn: Prof. G. J. Lieberman	1	University of North Carolina Statistics Department Chapel Hill, North Carolina 27515 Attn: Prof. W. L. Smith	1
Florida State University Department of Statistics Tallahassee, Florida 32306 Attn: Prof. I. R. Savage	1	Purdue University Division of Mathematical Sciences Lafayette, Indiana 47079 Attn: Prof. H. Rubin	1
Princeton University Department of Mathematics Princeton, New Jersey 08540 Attn: Prof. J. W. Tukey	1	University of California, San Diego Department of Mathematics P. O. Box 109 La Jolla, California 92038 Attn: Prof. M. Rosenblatt	1
Stanford University Department of Statistics Stanford, California 94305 Attn: Prof. T. W. Anderson	1	Florida State University Department of Statistics Tallahassee, Florida 32306 Attn: Prof. R. A. Bradley	1
University of California Department of Statistics Berkeley, California 94720 Attn: Prof. P. J. Bickel	1	New York University Department of Industrial Engineering & Operations Research Bronx, New York 10453 Attn: Prof. J. H. K. Kao	1
University of Washington Department of Mathematics Seattle, Washington 98105 Attn: Prof. Z. W. Birnbaum	1	University of Wisconsin Department of Statistics Madison, Wisconsin 53706 Attn: Prof. G. E. P. Box	1
Cornell University Department of Mathematics White Hall Ithaca, New York 14850 Attn: Prof. J. Kiefer	1		

Logistic Research Project The George Washington University 707 - 22nd Street, N.W. Washington, D.C. 20037 Attn: Dr. W. H. Marlow	Copies 1	The University of Michigan Department of Mathematics, W.E. Ann Arbor, Michigan 48104 Attn: Prof. R. M. Thrall	Copies 1
International Business Machines Corporation P. O. Box 218, Lamb Estate Yorktown Heights, New York 10598 Attn: Dr. Alan Hoffman	 1	Princeton University Department of Mathematics Princeton, New Jersey 08540 Attn: Prof. A. W. Tucker	 1
University of California Management Sciences Research Project Los Angeles, California 90024 Attn: Dr. J. R. Jackson	 1	Case Western Reserve University Systems Research Center Cleveland, Ohio 44106 Attn: Prof. M. Mesarovic	 1
Harvard University Department of Economics Cambridge, Massachusetts 02138 Attn: Prof. K. J. Arrow	 1	University of Texas Department of Mathematics Austin, Texas 78712 Attn: Dr. A. Charnes	 1
Cowles Commission for Research in Economics Yale University New Haven, Connecticut 06520 Attn: Prof. Martin Shubik	 1	Stanford University Department of Operations Research Stanford, California 94305 Attn: Dr. D. L. Iglehart	 1
Carnegie-Mellon University Graduate School of Industrial Administration Pittsburgh, Pennsylvania 15213 Attn: Prof. G. Thompson	 1	University of Delaware Department of Mathematics Newark, Delaware 19711 Attn: Prof. R. Remage, Jr.	 1
University of California Department of Economics Berkeley, California 94720 Attn: Prof. R. Radner	 1	Stanford University Department of Operations Research Stanford, California 94305 Attn: Prof. F. S. Hillier	 1
University of California Graduate School of Business Administration Los Angeles, California 90024 Attn: Prof. J. Marschak	 1	Dr. Claude-Alain Burdet Asst. Prof. Industrial Admin. Carnegie-Mellon University Pittsburgh, Pennsylvania 15213	 1
University of California Operations Research Center Institute of Engineering Research Berkeley, California 94720 Attn: Prof. D. Gale	 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, Virginia 22217	 1
		Science Librarian Kresge Library Oakland University Rochester, Michigan 48063	 1

	Copies		Copies
University of Connecticut Department of Statistics Storrs, Connecticut 06268 Attn: Prof. H. O. Posten	1	Professor Geoffrey G. Watson, Chairman Department of Statistics Princeton University Fine Hall Princeton, N. J. 08540	1
Prof. Emanuel Parzen Statistics Department State Univ. New York at Buffalo 4230 Ridge Lea Road Amherst, New York 14226	1	Stanford University Department of Mathematics Stanford, California 94350 Attn: Prof. S. Karlin	1
ARCON Corporation Lakeside Office Park North Avenue at Route 128 Wakefield, Massachusetts 01880 Attn: Dr. A. Albert	1	University of Sheffield Department of Probability and Statistics Sheffield 10, ENGLAND Attn: Prof. J. Gani	1
Stanford University Department of Statistics Stanford, California 94305 Attn: Prof. H. Chernoff	1	University of California Operations Research Center Institute of Engineering Research Berkeley, California 94720 Attn: Prof. R. E. Barlow	1
Yale University Department of Statistics New Haven, Connecticut 06520 Attn: Prof. L. A. Savage	1	Stanford University Department of Statistics Stanford, California 94305 Attn: Prof. H. Solomon	1
Rutgers-The State University Statistics Center New Brunswick, New Jersey 08903 Attn: Prof. H. F. Dodge	1	Applied Mathematics Laboratory Naval Ships Research Development Center Washington, D. C. 20007	1
Yale University Department of Statistics New Haven, Connecticut 06520 Attn: Prof. F. J. Anscombe	1	Systems Analysis Division Room BE760, Pentagon Washington, D. C. 20350 Attn: Mr. A. S. Rhodes, Op-964	1
Purdue University Division of Mathematical Sciences Lafayette, Indiana 47907 Attn: Prof. S. S. Gupta	1	Southern Methodist University Department of Statistics Dallas, Texas 75222 Attn: Prof. D. B. Owen	1
Cornell University Department of Industrial Engineering Ithaca, New York 14850 Attn: Prof. R. E. Bechhofer	1	Israel Institute of Technology Technion Haifa, ISRAEL Attn: Prof. P. Naor	1
Mrs. Barbara Eaudi Univ. Program Coordinator, B.E. NASA Johnson Space Center Houston, TX 77058		Department of Statistics University of North Carolina Chapel Hill, North Carolina 27515 Attn: Prof. M. R. Leadbetter	1