A CRITICAL PATH PROGRAM MANUAL

by

JAMES A. CHAMPY

Submitted in Partial Fulfillment
of the Requirements for the
Degree of Bachelor of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June, 1963

Signature redacted

Signature of Author. . . . .
Department of Civil Engineering, May 17, 1963

Signature redacted

Certified by . . . .
Thesis Supervisor

ACKNOWLEDGEMENT

The author wishes to express his gratitude to his thesis advisor, Professor E. F. Bisbee, for his helpful suggestions in the preparation of this work. He is also deeply indebted to Mr. J. A. Currie for the advice and encouragement he so freely provided.

This work has been done in part in the Civil Engineering Systems Laboratory, a research facility of the Civil Engineering Department at the Massachusetts Institute of Technology, and in part at the Computation Center at the Massachusetts Institute of Technology, Cambridge, Massachusetts.

ABSTRACT

A CRITICAL PATH PROGRAM MANUAL

Submitted to the Department of Civil Engineering on May 17, 1963, in partial fulfillment of the requirement for the degree of Bachelor of Science.

This thesis presents a program manual that was written to describe and facilitate the use of a program for critical path scheduling, CPS 300, that was developed in the Civil Engineering Systems Laboratory of the Massachusetts Institute of Technology. In the study and testing of CPS 300, several points of criticism appeared. This resulted in the critique and the suggestions for its revision which are also included herein. In particular, a new flow algorithm was written which may be more efficient than that presently used in CPS 300. This flow algorithm can be integrated into the main program, as is illustrated, but is also presented here as a separate program.

TABLE OF CONTENTS

I INTRODUCTION

Within the past four years there has been considerable attention paid to the methods of critical path scheduling and to adapting CPM to the tools of high speed computation. This interest has been the result of a search for a more exacting and efficient technique to facilitate the planning of large and complex projects in construction, manufacturing, and research and development.

PERT (Program Evaluation and Review Technique) and the IBM LESS are two of the more notable computer programs for scheduling that have evolved. However, many of these programs do not furnish a complete and detailed array of possible project schedules that would benefit management in planning decisions. With the goal of attaining a program that would accomplish this, a considerable amount of work was done in the Civil Engineering Systems Laboratory at M.I.T. in the spring of 1962 under the direction of Mr. J. L. Cutcliffe.

Two major accomplishments in the form of Master's Theses resulted from this work: "A Computer Approach Towards Developing Critical Path Networks" by D. R. Pennell, and "A Digital Computer Solution to the Critical Path Problem and an Approach to Resource Allocation" by J. R. Farmer. The Pennell thesis presented a program that would develop the activity network used as a model in critical path analysis. This was basically done by considering the precedent restrictions of each activity. The Farmer thesis presented a program that would do the actual network computations and would output an array of schedules with their related costs. Both programs were finally combined by Mr. Cutcliffe, and the composite program was called CPS (Critical Path Scheduling) 300. However, no effort was ever made to describe the operation or use of CPS 300, and thus the program manual in Appendix A became the object of this thesis.

As the program manual states, the basic purpose of a critical path analysis is to generate a spectrum of project schedules, each schedule being at a minimum project direct cost for the specified project duration. CPS 300, upon a preliminary investigation, appeared to accomplish this. This indicated that writing a manual for its use would be a profitable undertaking. However, a further study showed some inefficiencies and poor programing techniques existing, which could not be fully corrected because of the time limitations in preparing this thesis.

A considerable amount of study was spent on the flow algorithm used in the network computations, which may be considered the "heart" of the program. It appeared that a more logical and efficient algorithm could be found, and, thus, the program presented in Chapter IV was written.

To maximize the efficiency of CPS 300 a full reorganization and change of format will probably be necessary. As it exists now it is a combination of three chains, each written separately (CHAIN(1,A4) by Pennel, and CHAIN(2,B2) and CHAIN(3,B3) by Farmer) and connected only by output and input statements. Chapter V presents a suggestion for such a reorganization.

III WRITING OF THE MANUAL

The writing of the CPS 300 manual was actually the final step in this study. Before any description could be written, it was necessary to understand the general operation of the whole program plus the detailed operation of its more important parts, particularly the flow algorithm. The testing of the flow algorithm was done on an IBM 1620 computer, the whole program being tested on the IBM 7090. With an understanding of the operation and the technical restrictions of the program, the manual could be written.

The general format followed in the manual is that of the technical publications of the Civil Engineering Systems Laboratory at M.I.T. It was assumed that the user of CPS 300 would have some knowledge of critical path scheduling techniques and a basic understanding of the mathematical methods associated with them. Therefore, no detailed theory of critical path is given, but rather stress is placed on the operation of CPS 300 and how it uses the theory. However, the introduction to each descriptive section briefly explains how critical path techniques function, so that CPS 300 can be understood in the context of the theory.

Since the program is divided into three chains, the description of each chain is given separately. The operation of each chain is presented in the form of a macro flow diagram, with the individual operations discussed in the text. The less important operations are just mentioned, but the main parts of the chains are further discussed. The presentation of the network development method is lengthy and complicated. This was difficult to avoid because of the intricacies of the program. To illustrate the flow algorithm, a micro flow diagram was used. This allows the

CPS 300 user the option of further investigating the program operation without having to search through a lengthy discussion.

After Section 2.0, which describes the operation of the program, the remaining sections deal with the technicalities of using it. Section 4.0 serves as a summary for the more detailed description of the input format that is included in Section 3.0. The Operating Instructions, Section 5.0, are given for the IBM 7090 Fortran Monitor System. These instructions may have to be altered for other systems, particularly with the use of tapes. Section 6.0 presents the output format for each chain.

The Appendix includes a list of references for the CPS 300 user who is not familiar with critical path scheduling techniques. It also includes a micro flow diagram of the flow algorithm, a program listing, and a sample problem with the computer results.

III CRITIQUE OF CPS 300

The study and testing that was done in preparation for writing the CPS 300 program manual revealed several inefficiencies and possible deficiencies in the program. The inefficiencies stemmed both from programing techniques and redundant routines.

One of these techniques is the zero initializing of large, subscripted arrays in the network development chain. This is done, for example, in the statements preceeding statement number 9000, which writes zero values for several subscripted variables on Tape 9. This is unnecessary since values are read onto the tape directly over any values that the tape may already contain. The same technique is further used to initialize the IPRD precedent array in statements 5 through 7.

The efficiency of the flow algorithm in the network computation chain may also be questioned. This concern arose over the large number of transfer statements that appeared in a program trace of the algorithm. The large number of transfers from one IF statement to another suggested that it may be inefficient. Thus the new algorithm presented in the next chapter was written.

Since the chains of CPS 300 were originally written as separate programs, there appears to be some redundancies in their operation. The most evident among these is the arrow renumbering routine that is included in both CHAIN(1,A4) and CHAIN(2,B2). A network that has been developed by CHAIN(1,A4) and whose node numbers have already been put in order for the operation of the network computations, CHAIN(2,B2), is renumbered in the latter chain. This second renumbering is unnecessary and could waste a considerable amount of computer time for a project of many activities.

In the testing of CPS 300 on the IBM 7090, considerable difficulty was encountered with some of the example problems used. The program, as written, is supposed to check for common network logic errors and is to stop if any of these occur. In several test runs, a loop appeared in CHAIN(1,A4). This suggests that there was some form of network logic (or some violation of such) that the program could not accept. A criticism does not arise from the fact that this logic could not be accepted, but from the fact that neither a check nor a warning was provided to tell the CPS 300 user of the error. Because of this, one computer run on the IBM 7090 looped continuously for fifteen minutes and furnished no output.

One further criticism is with the input data format. The fields are arranged so that, in reading the data, it is often difficult to distinguish one word from another. Although this is only a matter of convenience, it could be corrected.

IV NEW FLOW ALGORITHM

The study of CPS 300 lead to a search for a more efficient flow algorithm. The new algorithm was originally written as a separate program that included a scheduling routine. As it is listed in Appendix B, it can perform all the basic functions of the network computation chain of CPS 300 except the check and node renumbering operations.

In order to perform an accurate check of whether this new algorithm is actually more efficient than the one presently used in CPS 300, it would be necessary to replace the main part of CHAIN(2,B2), the forward and backward iterations and the flow assignment, with the new algorithm. Using the same example problem, a comparison of the computer time used for each algorithm within CHAIN(2,B2) could be made. This was attempted, but difficulty encountered in CHAIN(1,A4) with the network logic of the test problems (referred to in the previous chapter) did not allow the program to proceed as far as the algorithm. However, the new algorithm has been prepared to directly replace what the program manual refers to as Operations 8, 9, and 10 of CHAIN(2,B2) so that further testing can be done. A listing of this in substitutable form is given in Appendix C.

The new algorithm was debugged and tested on both the IBM 1620 and 7090 and is available in Fortran language as a separate program for both computers. The listing in Appendix B is for use on the 7090.

Basically, the input format is the same as that used for CHAIN(2,B2) of CPS 300 except that no preferred duration is included. This field is left blank. One extra card is necessary at the beginning of the data giving the number of activities (KMAX) and the number of nodes (LMAX) of the network. A blank card is not necessary at the end of the data. The nodes must be numbered in order, i.e., the number of the node at the beginning of an activity must be less than the one at the end.

The program generates a list of schedules which include the name, starting and ending node, early start, late start, early finish, late finish, total float, and free float of each activity. At the end of each schedule, the project duration and cost for that schedule are printed. The final schedule will have a cost of 1000000, which the 7090 program outputs as infinity. 999999.99 is the final cost on the 1620 computer, the difference being due to rounding. A zero total float indicates a critical activity. A short, theoretical problem is also given in Appendix B to illustrate the program operation.

V SUGGESTIONS FOR THE REVISION OF CPS 300

Even if the problems that have been mentioned with CPS 300 were to be corrected, an efficient CPM program may still have not been developed. It appears that to make the program efficient, it would be necessary to completely reorganize it in order to remove the redundancies caused by combining the chains. This could be done by dividing the operations referred to in the program manual into subroutines. These subroutines could possibly be called by using the new computer languages now being developed in the Civil Engineering Systems Laboratory at M.I.T., COGO and STRESS. The main subroutines would contain the operations for the actual network development and computations. These would be surrounded with subroutines for reading, checking, node renumbering, considering preferred durations, and printing. A subroutine would only be called when required. A suggestion for the division of CPS 300 into these subroutines is shown below with the proposed subroutine names and functions.

SUBROUTINE READ	To read input data.
SUBROUTINE CHECK	To check for network logic errors in the input data.
SUBROUTINE START	To find and check for starting events.
SUBROUTINE DEVEL	To develop activity network from precedent restrictions.
SUBROUTINE END	To find and check for one ending event.
SUBROUTINE RENUM	To renumber the network nodes in order.
SUBROUTINE PRINT1	To print network development data.

SUBROUTINE COMP	To make network computations.
SUBROUTINE COST	To find the minimum total project cost.
SUBROUTINE RANGE	To check for the range of wanted output.
SUBROUTINE PRINT2	To print network computation data (schedules).
SUBROUTINE PREFER	To consider preferred activity durations.
SUBROUTINE PRINT3	To print schedules considering preferred durations.

Such a system has two distinct advantages. First, only those subroutines that are to be used are called. This would avoid redundancy and would conserve computer time and storage space. Second, if a more efficient subroutine were ever developed, it could easily replace the existing one without upsetting the structure of the whole program or of other subroutines.

VI CONCLUSION

CPS 300 cannot be considered an ideal program for critical path scheduling because of its operation. This does not infer, however, that the time to prepare and write the program manual shown in Appendix A was not well spent. Several of separate routines of the program are well written, and credit is due to Messrs Pennell and Farmer for their work. It is hoped that the manual will serve to facilitate the use of CPS 300 until further testing and possible revision can be made. Besides the technical benefit of the manual, this project has certainly provided the writer with an insight into the methods of critical path scheduling and the techniques used in their programing.

BIBLIOGRAPHY

Cutcliffe, J. L.; "Lecture Notes on Critical Path Scheduling," Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, Mass., October 1962.

Farmer, J. R.; "Digital Computer Solution to the Critical Path Problem and an Approach to Resource Allocation," M.I.T. Building Engineering and Construction Master's Thesis, June 1962. Thesis Supervisor, J. Lloyd Cutcliffe.

Pennell, D. R.; "Computer Approach Towards Developing Critical Path Networks," M.I.T. Building Engineering and Construction Master's Thesis, June 1962. Thesis Supervisor, J. Lloyd Cutcliffe.

APPENDIX A

DEPARTMENT OF CIVIL ENGINEERING
CIVIL ENGINEERING SYSTEMS LABORATORY

CPS 300
CRITICAL PATH
PROGRAM MANUAL

by

J. A. Champy

May, 1963

School of Engineering
Massachusetts Institute of Technology

ACKNOWLEDGEMENTS

The author wishes to acknowledge the work done in writing the original parts of the CPS 300 Critical Path Program by D. R. Pennell and J. R. Farmer. Credit is also due to J. L. Cutcliffe, who combined the work of Messrs Pennell and Farmer into its present form.

The author further gratefully acknowledges the help of Professor E. F. Bisbee and Mr. J. A. Currie, who aided in the preparation of this manual.

This work has been done in part in the Civil Engineering Systems Laboratory, a research facility of the Civil Engineering Department at the Massachusetts Institute of Technology, and in part at the Computation Center at the Massachusetts Institute of Technology, Cambridge, Massachusetts.

TABLE OF CONTENTS

CPS 300
CRITICAL PATH PROGRAM

1.0 INTRODUCTION

With the increasing size and complexity of industrial and construction projects there has come a recognition of the need for new management techniques to provide for efficient planning and scheduling. One of the first approaches to this problem was the development of critical path scheduling methods (CPM). This manual describes an electronic computer program, CPS 300, that utilizes the technique of the critical path method and provides an optional algorithm that develops a representative network of the project activities. The manual is written with the assumption that the reader has some understanding of the basic mathematical techniques involved in critical path methods. (See References in Appendix.)

Although CPS 300 is one of the most complete programs that has been developed for project scheduling, it will not make the final decision of the optimal schedule. This is function of the goals and restrictions of the CPS 300 user. However, the program does provide a detailed array of project schedules and their respective durations and costs.

2.0 DESCRIPTION

CPS 300 was written in FORTRAN language primarily for use on the IBM 709/7090 Computer. It is divided into three parts, referred to in the program as chains.

CHAIN(1,A4) is an algorithm that develops a network of project activities, CHAIN(2,B2) does the actual network computations, and CHAIN(3,B3) considers the preferred durations of the project activities. These chains may be used in combination or separately (except CHAIN(3,B3), which is dependent on the output of CHAIN(2,B2)). For example, the CPS 300 user may wish to construct a network diagram by hand methods and would, therefore, only have to call CHAIN(2,B2). He would call CHAIN(3,B3) if he wished to consider any preferred durations of the activities. The signals for calling the different chains are included in the control card which is described under Section 3.0.

2.1 NETWORK DEVELOPMENT - CHAIN(1,A4)

A critical path analysis requires a network of project activities which serves as a mathematical model for computations. A simplified example of such a network is shown in Figure 1. Each link of the network represents an activity of the project, and each node represents an event which may be the start or finish of an activity. (This is often referred to as an arrow diagram, the arrows representing activities and the arrow heads and tails, events.)

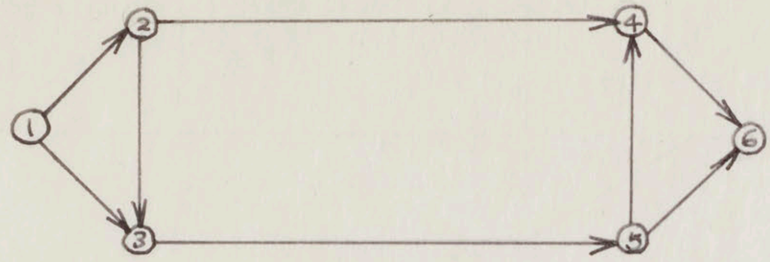


FIGURE 1.

The network is constructed according to restrictions on activities; in other words, the start of one activity is dependent on the completion of one or more other activities. The breakdown of a project into separate activities is determined by several factors: type of work, predominant labor classification, work responsibility, work location, and practical considerations according to how the work is to be accomplished.

Only the breakdown of activities and their precedent restrictions (in any order) are required for the operation of CHAIN(1,A4). (Time and cost data is also included in the input data but only for use in the actual critical path computations.) Each activity is represented by an assigned number. The program then develops a logical network and outputs the node number at the start and finish of each activity. Figure 2 illustrates the general operation of CHAIN(1,A4).

Operation 1 contains the necessary dimension statements and orders the input data to be read.

Operation 2 checks to find if the number of input activities exceeds the capacity of CHAIN(1,A4). If this number is exceeded, the computer will stop and will print out a statement to that

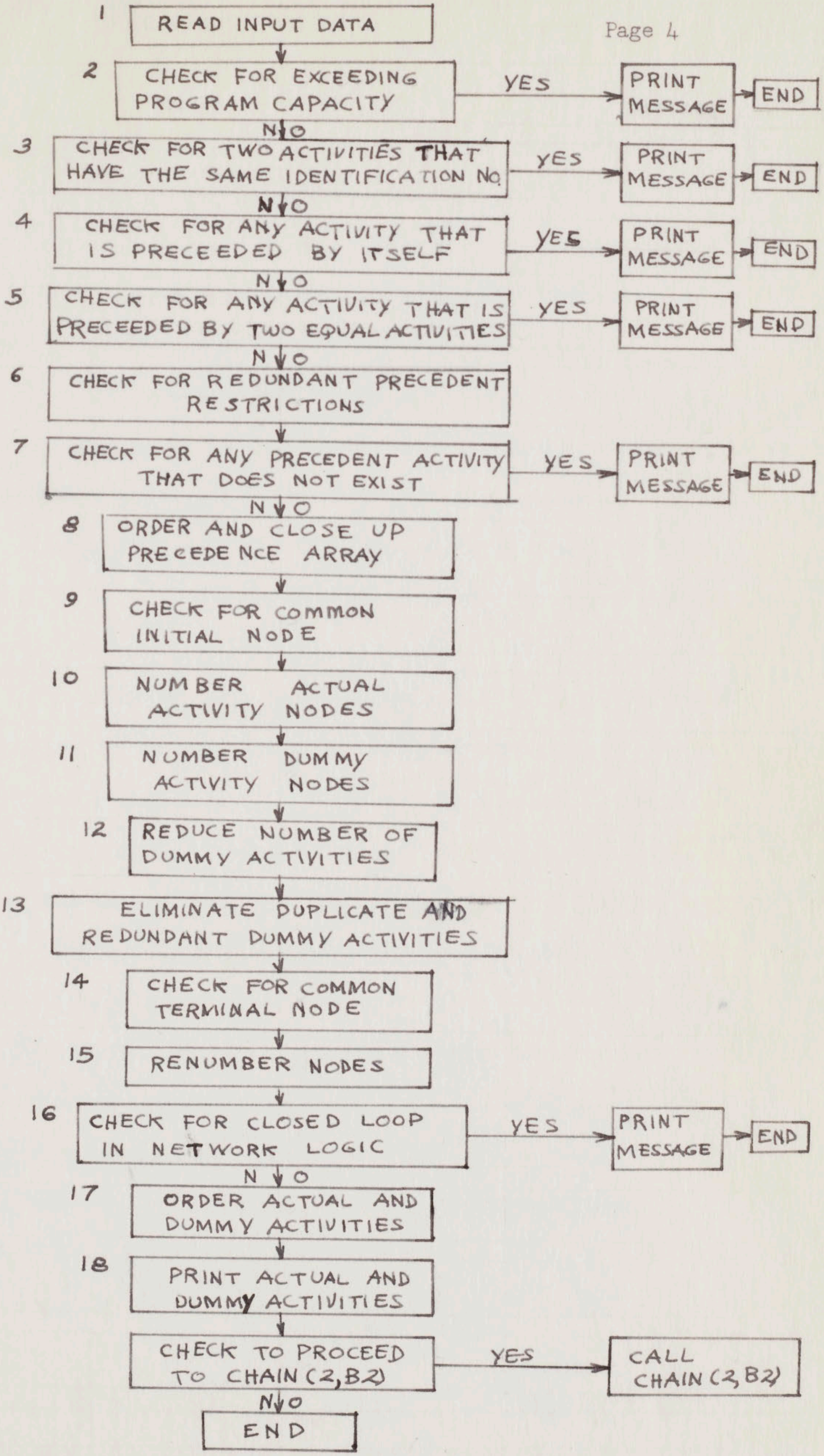


FIGURE 2. MACRO FLOW DIAGRAM FOR CHAIN(1,A4)

effect. The capacity is actually governed by the storage size of the computer being used. For an IBM 7090 Computer the maximum number of activities is 2600.

Operations 3 through 7 search the input data for logic errors. If such an error is found, except in Operation 6, the computer will stop and the error will be printed out. Operation 6, check for redundant precedence restraints, will remove any such restraints which would cause the creation of redundant dummy activities. Although these dummy activities would not cause an erroneous network, they should be avoided to reduce the later work of the program.

Since no restrictions are placed on the order in which precedent activities are listed in the input data and since Operation 6 may have removed some redundant precedent restraints, possible discontinuities will exist in the precedent array. (This array is referred to in the program as the IPRE(N,L) array, where L equals one to seven columns.) Operation 8 proceeds to correct these discontinuities by ordering and closing the array.

Operations 9, 10, 11, and 14 compose the main part of CHAIN(1,A4): find and number the initial nodes and terminal nodes, and number the actual and dummy activity nodes. If an activity is not preceded, the program assumes that it is an activity which commences the project. Any number of activities may fall into this category, and each of their respective starting nodes are labeled as one.

After the initial node(s) has been found, the program proceeds to number the actual activity nodes. In order to assign a beginning node number to an activity and at the same time relate this node to a preceding activity ending node by equating it or separating it by a dummy activity, another array must be established. This array has the same N (activity number) dimension as the precedence array and is called the IPRD(N,L) array, where L equals one to six columns.

As each activity is encountered, the precedence array is scanned to see if any other activity has exactly the same precedence restraints and if it does, then column six of the IPRD(N,L) array is marked and all such activities are known to have the same beginning node. If no other activities have the exact same precedence restrictions, then the precedence array is scanned to find those precedence activities which also precede other activities and to which dummy activities must be drawn. When a preceding activity also precedes an activity other than the one in question, the element in the IPRD(N,L) array corresponding to the preceding activity location is labeled. This process is done for each of the five precedence elements of the array for the activity. Thereupon, node numbers can be generated for the activity. The beginning node takes a value twice the activity number and the ending node takes a value twice the activity number plus one. These two node numbers are then stored in an extended two columns of the precedence array opposite their respective activity. This then increases the IPRE(N,L) array (previously only precedence) to seven columns where L equals one to seven. In the print-out of the network solution, the last two columns of the IPRE(N,L) array namely IPRE(N,6) and IPRE(N,7) are listed as activity beginning and ending nodes with labels of I and J respectively.

Generation of the dummy nodes is dependent on the action taken in the IPRD array. If the preceding job element in this array is designated, a dummy is generated. A dummy activity does not generate its own node numbers. It assumes the preceding activity ending node for its beginning node and takes on the beginning node of the subject activity as its ending node. Assumed dummy activity nodes are then assigned to a two-dimensional array for storage and subsequent reference purposes. This array is labeled IJD(KD,L) where L equals one to two, and KD is a serialized number from one to the maximum number of dummies required.

Operation 14 searches for the terminal node. Each activity that does not function in a precedence capacity at least once may be assumed to have the terminal node of the project as its ending node. All activities with this characteristic are scanned and each activity is assigned an ending node equal to the largest of the ending nodes.

Before the terminal node is found, however, Operations 12 and 13 check for any possible dummy reductions and for any duplicate or redundant dummy activities that may have been created by the program. Operation 12 provides for a case such as shown in Figure 3, in which

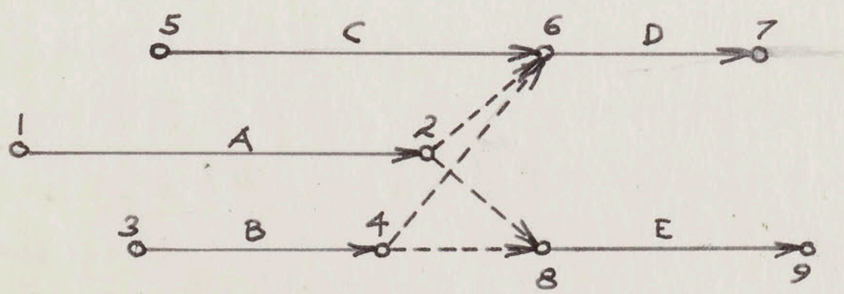


FIGURE 3.

it is possible to reduce the number of dummy activities. The program would equate activity B's ending node to activity A's ending node.

Operation 13 provides for a case such as shown in Figure 4, in which there are duplicate precedence dummy activities. Here dummy activity 2-9 is not necessary as long as dummy activities

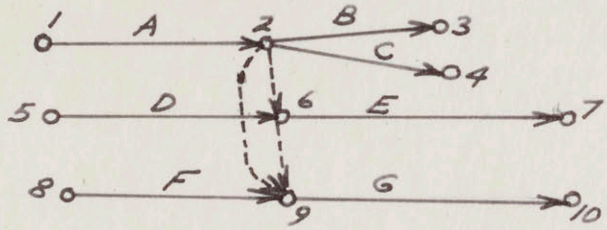


FIGURE 4.

2-6 and 6-9 are in existance and would be eliminated.

Because networks have fewer nodes than activities, the program may produce some large node numbers that are not necessary. Operation 15 reduces these node numbers and reassigns new node numbers so that the beginning node is always less than the ending node. This also prepares the network for use in CHAIN(2,B2).

Operation 16 checks for closed loops within the network logic. A closed loop is not allowable since it would imply that an activity must precede itself. If one occurs, the computer will stop, and an explanatory statement will be printed out.

Operations 17 and 18 order and print the actual and dummy activities. If the CPS 300 user wishes to continue directly to CHAIN(2,B2) it is possible to skip the print-out order or to punch the output if he wishes to use it at some other time for CHAIN(2,B2) input. Directions for doing this are included in the control card.

2.2 NETWORK COMPUTATION - CHAIN(2,B2)

Once an activity network has been developed, whether by using CHAIN(1,A4) or by hand methods, CHAIN(2,B2) may be used for the actual critical path computations. Besides the activity node numbers, CHAIN(2,B2) requires activity time and cost data. This includes the normal time of the activity with its associated cost, the crash time of the activity with its associated cost, and a preferred activity time if CHAIN(3,B3) is to be used. The normal time is defined as the shortest normal time of completing an activity and the crash time as the fastest possible time of completing an activity. The preferred time is a function of the desire of the CPS 300 user.

The basic purpose of a critical path analysis is to generate a spectrum of project schedules, each schedule being at a minimum project direct cost for the specified project duration. CHAIN(2,B2) accomplishes this by using a variation on the Fulkerson flow algorithm. (See Reference (2).) From the time-cost data, a cost curve is developed for each activity. Its slope is considered as flow through the network, and the program searches for the longest path in time through which the flow can travel. This is based on the theory that the minimum project direct cost will result if the duration of each activity is maximized. The program then searches back through the network to determine if a longer possible path exists. This is done several times until, in effect, all the flow has been "used up." In this way, several schedules are generated, each having a critical path. The activities of these paths, if delayed beyond their scheduled duration, would delay the whole project. These are termed "critical activities" and are labeled such by the program. The operation of CHAIN(2,B2) is illustrated in Figure 5.

Operation 1 actually performs several functions as the data is being read into the computer. It adds the normal project cost; it calculates the slope of the cost curve of each activity; and it adds the total number of links and nodes being read in. By performing several operations such as this within the same loop, less subscripted variables are used, and computer storage space is conserved.

As in CHAIN(1,A4), a check for exceeding the program capacity is included in Operation 2. Such a check is provided in the event that CHAIN(1,A4) is not used first. Again, the capacity is 2600 activities.

Operations 4 and 5 check for two starting or ending events. This is done by searching for an event that is at the tail of an arrow but does not appear at the head of any other arrow. Likewise, the reverse is done in checking for two ending events. The program cannot proceed with two starting or ending events, and the error will be printed out. If the network logic is known to be correct, then this check may be skipped by indicating such on the control card.

CHAIN(2,B2) requires that the event at the head of each arrow be numbered greater than at the tail. To assure this Operation 6 renumbers the nodes. Each activity is scanned to find the starting event. Once the starting event has been found, the

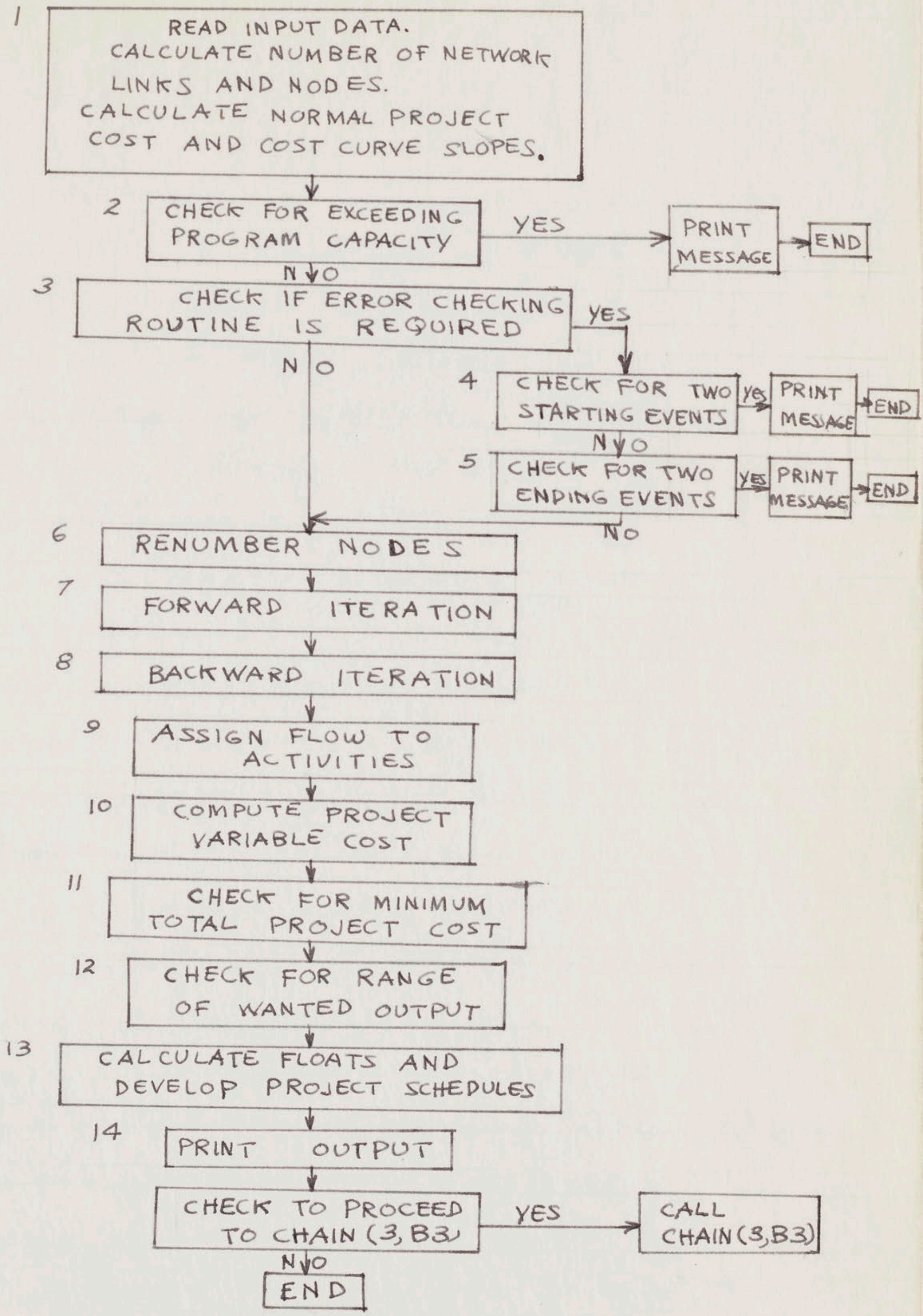


FIGURE 5. MACRO FLOW DIAGRAM FOR CHAIN (2, B2)

tails of all the arrows originating at the starting event are numbered one. At the same time, the activities previously associated with event one have the heads and tails of the arrows changed to the original number of the starting event. The arrow diagram is then scanned to find an event where the tails of all the activities entering the event have been renumbered. This event is numbered two, and all the numbers previously associated with this event are transferred to the event previously numbered two. This process is repeated until each event has been renumbered correctly. If no event can be found to renumber before all events have been renumbered, then a closed loop must be present, and within this loop it is impossible to number the events so the heads of the arrows are numbered greater than the tails. If this situation is encountered, the program will print the error and stop.

The renumbering routine places no restrictions upon the planner in the development of the arrow diagrams. The diagram can be numbered arbitrarily, if done by hand, and numbers can be omitted or activities may be inserted later.

Operations 7, 8, and 9 compose the main part of CHAIN(2,B2). They perform the flow computations described earlier in this section, tracing the flow forward and backward through the network and generating several schedules. Because of the importance of these operations, a micro flow diagram of the flow algorithm is included in the Appendix.

Operation 10 computes the project variable or direct cost for each schedule. This is done by adding to the total normal cost associated with the longest project duration in the first schedule the extra cost computed by multiplying the difference in duration of each schedule by the slope of the project cost curve.

Operation 11 adds the direct and indirect project costs. The data for the slope and duration of the indirect cost curve is included in the control card. The effect of adding these two costs should produce a minimum total project cost. This is illustrated in the cost curves shown in Figure 6.

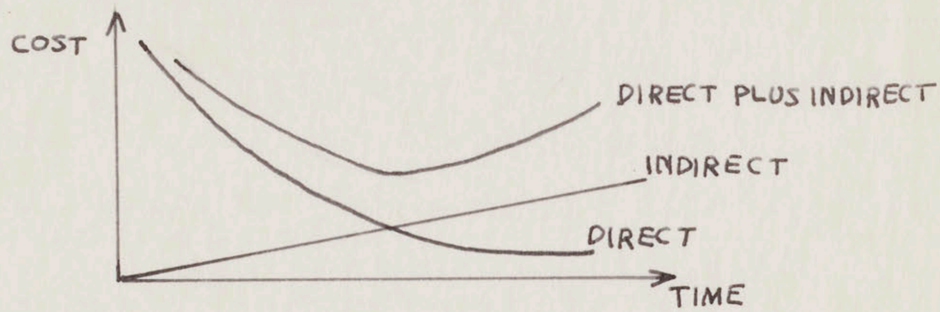


FIGURE 6.

It is conceivable that in a very large network many schedules might be generated that would be unwanted or useless. Operation 12 considers the relevant range as specified on the control card and instructs the computer to print-out only schedules within a desired time period. By using this option, computer time will be reduced.

Operation 13 calculates activity floats and develops project schedules. Floats indicate to the project management the status of all activities in relation to the entire project and the degree of criticalness of all activities. Total, free, and independent floats are calculated.

Total float is defined as the amount of time that an activity can be lengthened without delaying the completion of the project. When the total float of an activity is zero, the activity is critical. By checking for zero total floats, the program determines critical activities.

Free float is defined as the amount of time that an activity can be lengthened without changing the earliest times of all activities immediately following.

Independent float is defined as the amount of time that an activity can be lengthened without changing either the latest starting time of preceding activities or the earliest starting time of all activities immediately following.

The project schedule, consisting of the earliest and latest times for each event, is prepared in the following manner: The earliest time than an activity can be started is the earliest time associated with the event at the tail of the activity in the arrow diagram. The earliest that an activity can be finished is the earliest start plus the scheduled duration of the activity. The latest finish of an activity is the latest time associated with the event at the head of the arrow, as derived from the float computations. The latest start of an activity is the latest finish minus the scheduled duration. This information is calculated for each activity of the project.

The schedules will now be printed out in Operation 16. The program will then halt or will proceed to consider preferred activity durations by calling CHAIN(3,B3).

2.3 PREFERRED DURATIONS - CHAIN(3,B3)

It is conceivable that an individual activity could have a horizontal cost curve as shown in Figure 7. Such a curve would

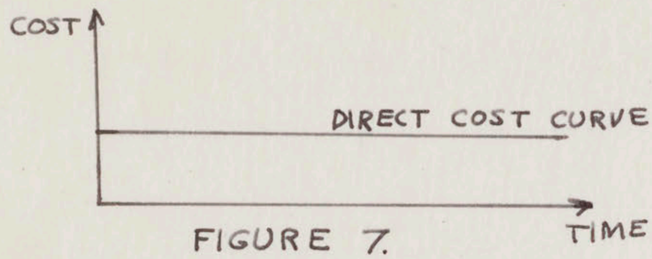


FIGURE 7.

allow planners more flexibility in scheduling, so CHAIN(3,B3) was developed to consider preferred durations of activities with a cost curve of or close to this type. Figure 8 shows a macro flow diagram of CHAIN(3,B3).

In this algorithm the scheduled durations and the preferred duration are the only input information necessary. These are obtained from the output of CHAIN(2,B2) and are read in Operation 1.

The remaining operations proceed with the analysis. Only activities having total float can be considered since an activity having zero float, if lengthened or delayed, would delay the project completion date also. The difference between the scheduled and preferred duration is calculated. The scheduled duration of all non-critical activities is increased by a small percentage of the above calculated difference. The total float is recalculated by a subprogram, and a comparison is made to determine which activities are closest to becoming critical.

The duration of each non-critical activity is lengthened until it becomes critical. If the scheduled duration is greater than the preferred duration, the scheduled duration is reduced to the

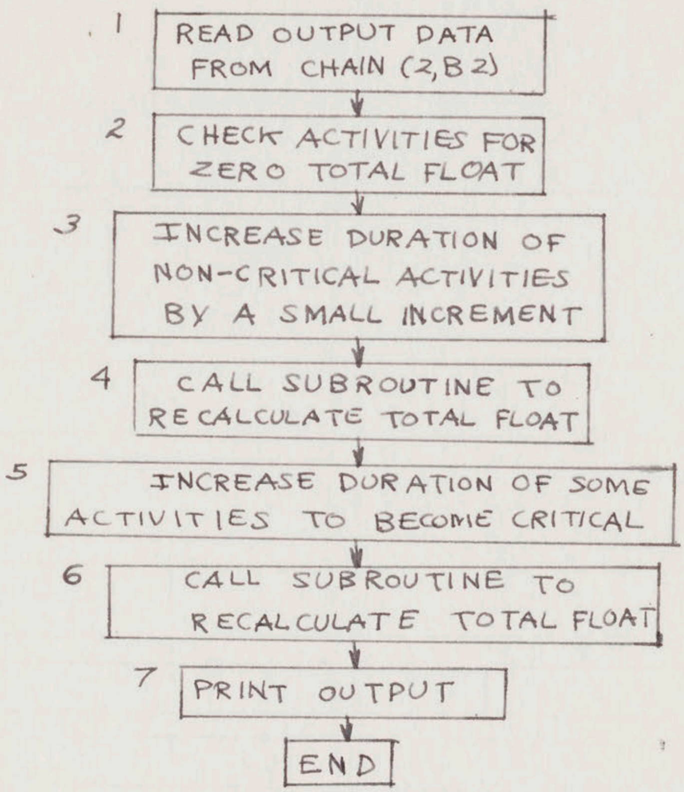


FIGURE 8. MACRO FLOW DIAGRAM FOR CHAIN (3,B3)

preferred duration and the total float recalculated. Many activities will be scheduled to be completed in a duration between the originally scheduled duration and the preferred duration. These activities are then critical and must not be lengthened or the project's completion date will be delayed.

Operation 7 prints the output in the same format as that of CHAIN(2,B2).

3.0 INPUT DATA CARD FORMATS

The 709/7090 composition and format of each data card is now presented. Each word of data must be right adjusted (or trailing zeros used to eliminate the blanks) within the appropriate field. The alphanumeric descriptive words, however, may be left adjusted for appearance sake in the output. Since the 709/7090 computer accepts a decimal punched anywhere within a field, no adherence to card columns, other than for convenience, is necessary within each field.

Identification Data

Card 1

Field	Variable	Description
1-72	Var	Project description. Alphanumeric.

VAR is the name which identifies the project that is being studied. The name is printed on the schedule output sheets.

Control Data

Card 2

Field	Variable	Description
1-6	FIR	Minimum time of range of desired output. Floating point.
7-12	FIN	Maximum time of range of desired output. Floating point.

13-18	FIXA	Slope of initial overhead curve. Floating point.
19-24	FIXB	Slope of final overhead curve. Floating point.
25-30	DURA	Duration of initial overhead curve. Floating point.
31-48	VOR	Description of dummy activity. Alphanumeric.
49-67	blank	
68	IDUM	Signal for printing output for CHAIN(3,B3) if ≠ 0. Fixed point.
69	IERR	Signal for making error network check in CHAIN(2,B2) if ≠ 0. Fixed point.
70	IJACK	Signal for doing CHAIN(2,B2) and CHAIN(3,B3) if ≠ 0. Fixed point.
71	IPRT	Signal for printing output for CHAIN(1,A4) if ≠ 0. Fixed point.
72	IPUN	Signal for punching output for CHAIN(1,A4) if ≠ 0. Fixed point.

The values for FIR, FIN, FIXA, FIXB, and DURA may be omitted if only CHAIN(1,A4), the network development algorithm, is to be used. If CHAIN(2,B2) and CHAIN(3,B3) are used and values for FIR and FIN are omitted, a complete array of schedules will be printed. Values for the overhead curve, FIXA, FIXB, and DURA, may also be omitted if the user does not wish to output indirect costs. An initial and final slope of the overhead curve are provided to allow for an overhead change during the project. This may be caused, for instance, on a construction project by a change in seasons.

A description for dummy activities, VOR, must be included for CHAIN(1,A4) but may be omitted if only CHAIN(2,B2) and CHAIN(3B,3) are used. This allows the program to label a dummy activity. The word "DUMMY" will suffice.

IDUM, IERR, IJACK, IPRT, and IPUN are signals for the different operations of the program. If the operation is not desired, their respective fields may be left blank. IPUN, the signal for punching the network development output, may be used if the network computations are to be made at another time. This will allow network computation data to be read into the computer directly from punched cards. The signal for not doing CHAIN(1,A4) is included in the activity computation data.

Network Development Data

Card 3 through Card 2600

Field	Variable	Description
1-4	I (blank)	Signal for not doing CHAIN(1,A4) if $I \neq 0$ for any activity. Leave blank for network development.
5-8	K(N)	Activity number of Nth activity read in. Fixed point.
9-12	IPRE(N,1)	Activity which immediately precedes K(N). Fixed point
13-16	IPRE(N,2)	Activity which immediately precedes K(N). Fixed point
17-20	IPRE(N,3)	Activity which immediately precedes K(N). Fixed point
21-24	IPRE(N,4)	Activity which immediately precedes K(N). Fixed point

25-28	IPRE(N,5)	Activity which immediately precedes K(N). Fixed point
29-31	TN	Normal duration of an activity. Floating point.
32-34	TCR	Crash time of an activity. Floating point.
35-37	PRT	Preferred time of an activity. Floating point.
38-42	CSTN	Normal cost of an activity. Floating point.
43-47	CSTCR	Crash cost of an activity. Floating point.
48-71	ACT	Activity description. Alphanumeric.

Last Card

1-80	blank	Signifies end of data.

Network Computation Data

If CPS 300 is to be used only for CHAIN(2,B2) or CHAIN(2,B2) and CHAIN(3,B3), network computations, and not for the development of the network also, the input data for Cards 3 through 2600 will be of different composition and format. The new format follows.

Card 3 through Card 2600

Field	Variable	Description
1-4	I	Event number at the start of activity link. (Arrow tail.) Fixed point.
5-8	J	Event number at the finish of activity link, (Arrow head.) Fixed point.
9-28	blank	
29-31	TN	Normal duration of an activity. Floating point.

32-34	TCR	Crash time of an activity. Floating point.
35-37	PRT	Preferred time of an activity. Floating point.
38-42	CSTN	Normal cost of an activity. Floating point.
43-47	CSTCR	Crash cost of an activity. Floating point.
48-71	ACT	Activity description. Alphanumeric.

<u>Last Card</u>

1-80	blank	Signifies end of data.

The presence of a non-zero value in card columns 1 through 4 signals the computer to skip CHAIN(1,A4) and proceed directly with the network computations. If both CHAIN(1,A4) and CHAIN(2,B2) are to be used, the input format of the network development data is necessary. Both types of input require that the last card in the data deck be blank.

4.0 CARD FORMAT SUMMARY

709/7090 Input Formats

The following card formats describe the 709/7090 input data and refer to the FORTRAN variable names explained in the previous section.

Identification Data

Card 1

1-72	VAR

Control Data

Card 2

1-6	FIR
7-12	FIN
13-18	FIXA
19-24	FIXB
25-30	DURA
31-48	VOR
49-67	blank
68	IDUM
69	IERR
70	IJACK
71	IPRT
72	IPUN

Network Development Data

Card 3 through Card 2600

1-4	I (blank)

5-8	K(N)
9-12	IPRE(N,1)
13-16	IPRE(N,2)
17-20	IPRE(N,3)
21-24	IPRE(N,4)
25-28	IPRE(N,5)
29-31	TN
32-34	TCR
35-37	PRT
38-42	CSTN
43-47	CSTCR
48-71	ACT

Last Card

1-80	blank

Network Computation Data

Card 3 through Card 2600

1-4	I
5-8	J
9-28	blank
29-31	TN
32-34	TCR
35-37	PRT
38-42	CSTN
43-47	ACT
48-71	ACT

Last Card

1-80	blank

5.0 OPERATING INSTRUCTIONS

System

Fortran Monitor System

Console Switches

1. Sense Switches 1 not used
   2 not used
   3 not used
   4 not used
   5 not used
   6 not used

Tapes

1. Tape 9 - scratch tape
2. Tape 10 - scratch tape
3. Tape A4 - primary chain tape
4. Tape B2 - secondary chain tape

Order of Input

1. Program Deck
2. Identification Card
   VAR
3. Control Card
   FIR, FIN, FIXA, FIXB, DURA, VOR, IDUM, IERR,
   IJACK, IPRT, IPUN
4. Activity Cards (for CHAIN(1,A4))
   I, K(N), IPRE(N,1), IPRE(N,2), IPRE(N,3), IPRE(N,4),
   IPRE(N,5), TN, TCR, PRT, CSTN, CSTCR, ACT
   or
   Activity Cards (for CHAIN(2,B2) and CHAIN(3,B3) only)
   I, J, TN, TCR, PRT, CSTN, CSTCR, ACT
5. Blank Card

Special Instructions

The capacity of CPS 300 is 2600 activities for all chains and cannot be exceeded using the IBM 7090 Fortran Monitor System.
For CHAIN(1,A4), a maximum of five precedent activities (IPRE(N,L)) are allowed per subject activity (K(N)).

Output List

1. CHAIN(1,A4) - intermediate
   K, I, J
   CHAIN(1,A4) - final
   I, J, TN, TCR, PRT, CSTN, CSTCR, ACT
2. CHAIN(2,B2)
   ACT, I, J, SCDR, BIGG, STLT, FINER, FINLT, FLTO, FLFR
   FLIN, COST
3. CHAIN(3,B3)
   same as CHAIN(2,B2)

6.0 OUTPUT FORMATS

Each chain of CPS 300 provides printed output. CHAIN(1,A4) will also output punched cards upon request on the control card

6.1 OUTPUT OF CHAIN(1,A4)

Printed or punched output may be deleted from CHAIN(1,A4) if the program is to continue directly to CHAIN(2,B2). If printed output is requested on the control card, the activity identification number, K; the tail of the activity arrow, I; and the head of the activity arrow, J, will be printed out and ordered in three different ways: by activity number, activity beginning node number (arrow tail), and activity ending node number (arrow head). This is done to facilitate the drawing of the actual network or arrow diagram. Dummy activities are printed separately with values for K, I, and J.

Punched output is also available from CHAIN(1,A4) with the following information for each activity:

I	activity beginning node number
J	activity ending node number
TN	normal time
TCR	crash time
PRT	preferred time
CSTN	normal time
CSTCR	crash cost
ACT	activity name

This information is in the correct format for use as CHAIN(2,B2) input data.

6.2 OUTPUT OF CHAIN(2,B2)

Output from CHAIN(2,B2) is printed in schedule form as each schedule is generated. The schedule contains a heading with the schedule number, the project duration and the total variable cost for that schedule, and the project identification statement. The following information is printed for each activity:

ACT	activity name
I	activity beginning node number
J	activity ending node number
CRITICAL	indicated by yes or no
SCDR	scheduled duration
BIGG	earliest start
STLT	latest start
FINER	earliest finish
FINLT	latest finish
FLTO	total float
FLFR	free float
FLIN	independent float
COST	activity cost

A summary sheet is also printed for the project giving the schedule number, duration, variable cost, overhead, and total project cost for each generated schedule.

6.3 OUTPUT OF CHAIN(3,B3)

CHAIN(3,B3) prints out schedule sheets with values for the same variables and in the same format as in the print-out of CHAIN(2,B2).

APPENDIX

REFERENCES

(1) Kelley, J. E., Jr. and Walker, M. R.; "Critical Path Planning and Scheduling," Proc. Eastern Joint Computer Conference, 160-173, Boston, December 1-3, 1959.

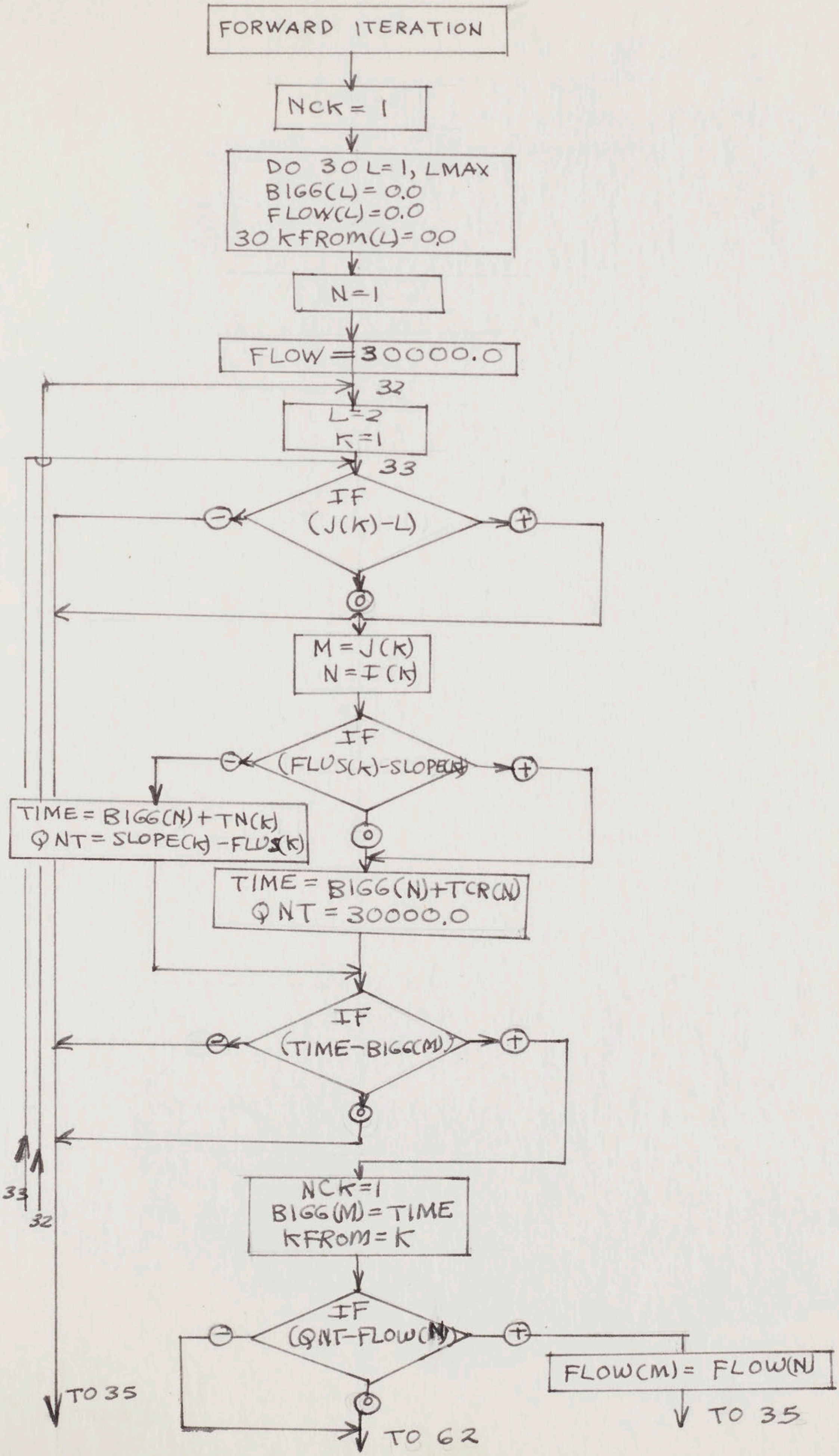
(2) Fulkerson, D. R.,; "A Network Flow Computation for Project Cost Curves," Management Science, Vol. 7, No. 2, January 1961.

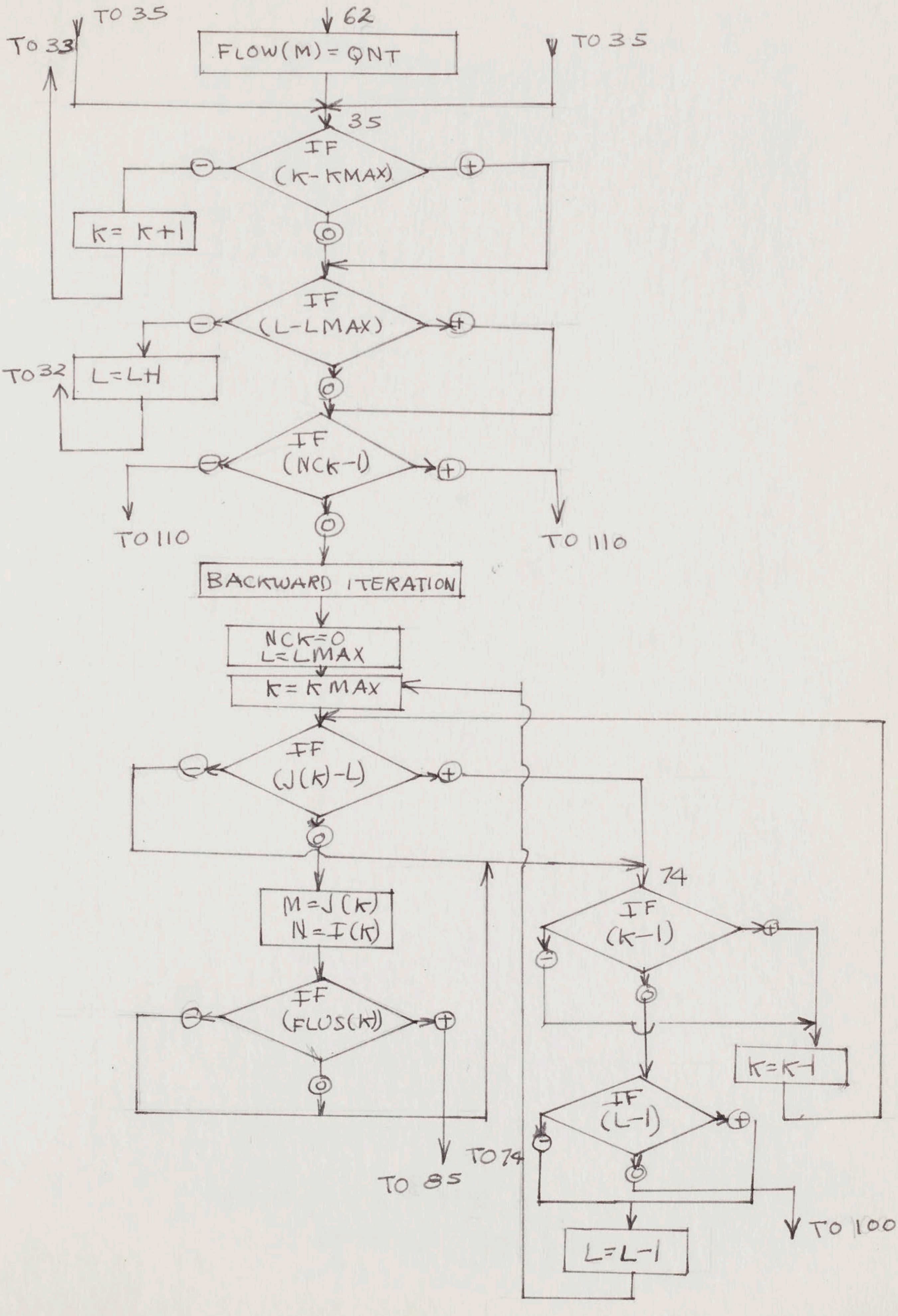
(3) Ford, L. R., and Fulkerson, D. R.; "A Simple Algorithm for Finding Maximal Network Flows and an Application to the Hitchcock Problem," Canadian J. Math., 9, 210-218, 1957.

(4) Kelley, J. E., Jr.; "Critical Path Planning and Scheduling: Mathematical Basis," Operations Research, Vol. 9, No. 3, May-June 1961.

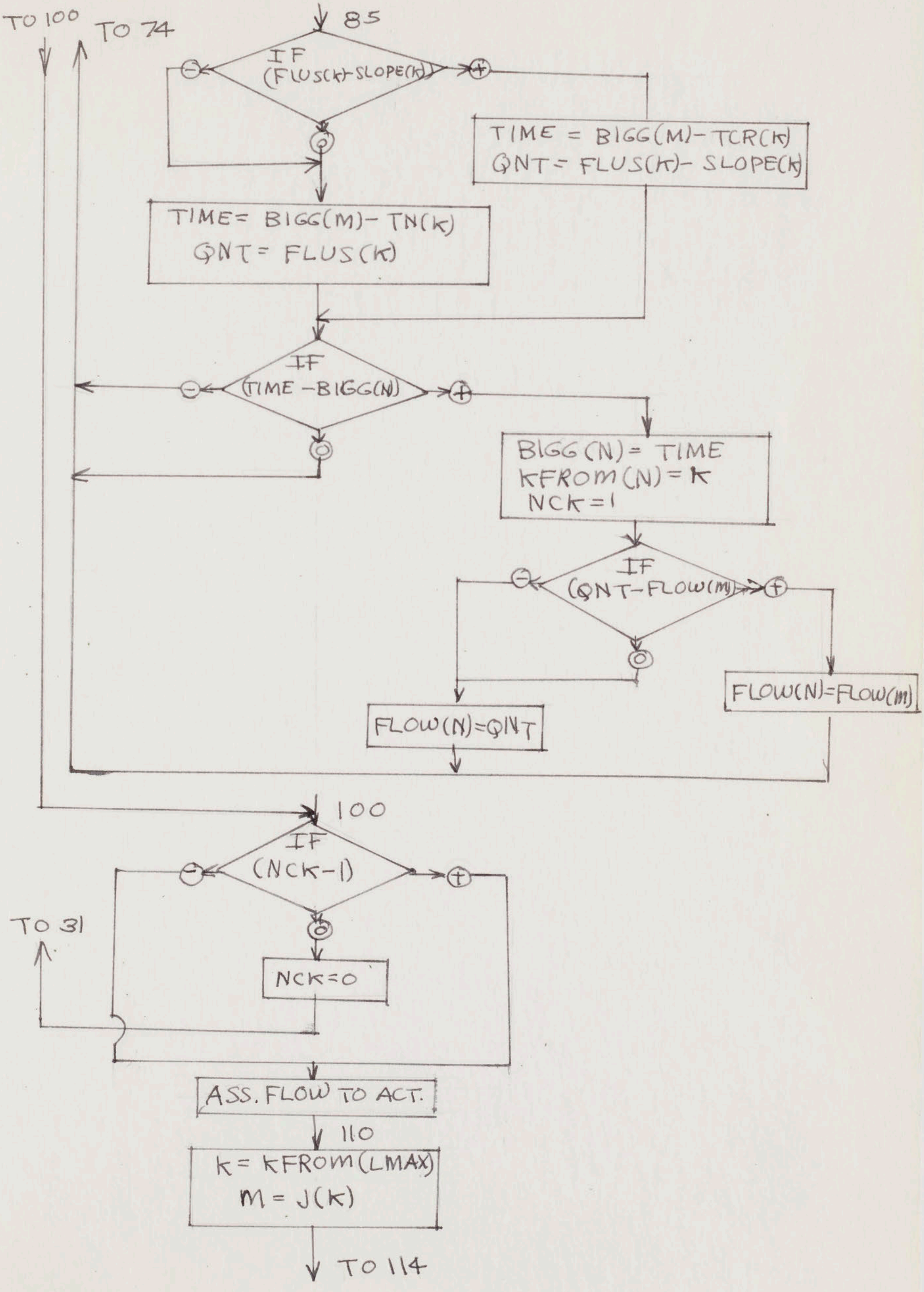
(5) Cutcliffe, J. L.; "Lecture Notes on Critical Path Scheduling," Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, Mass., October 1962.

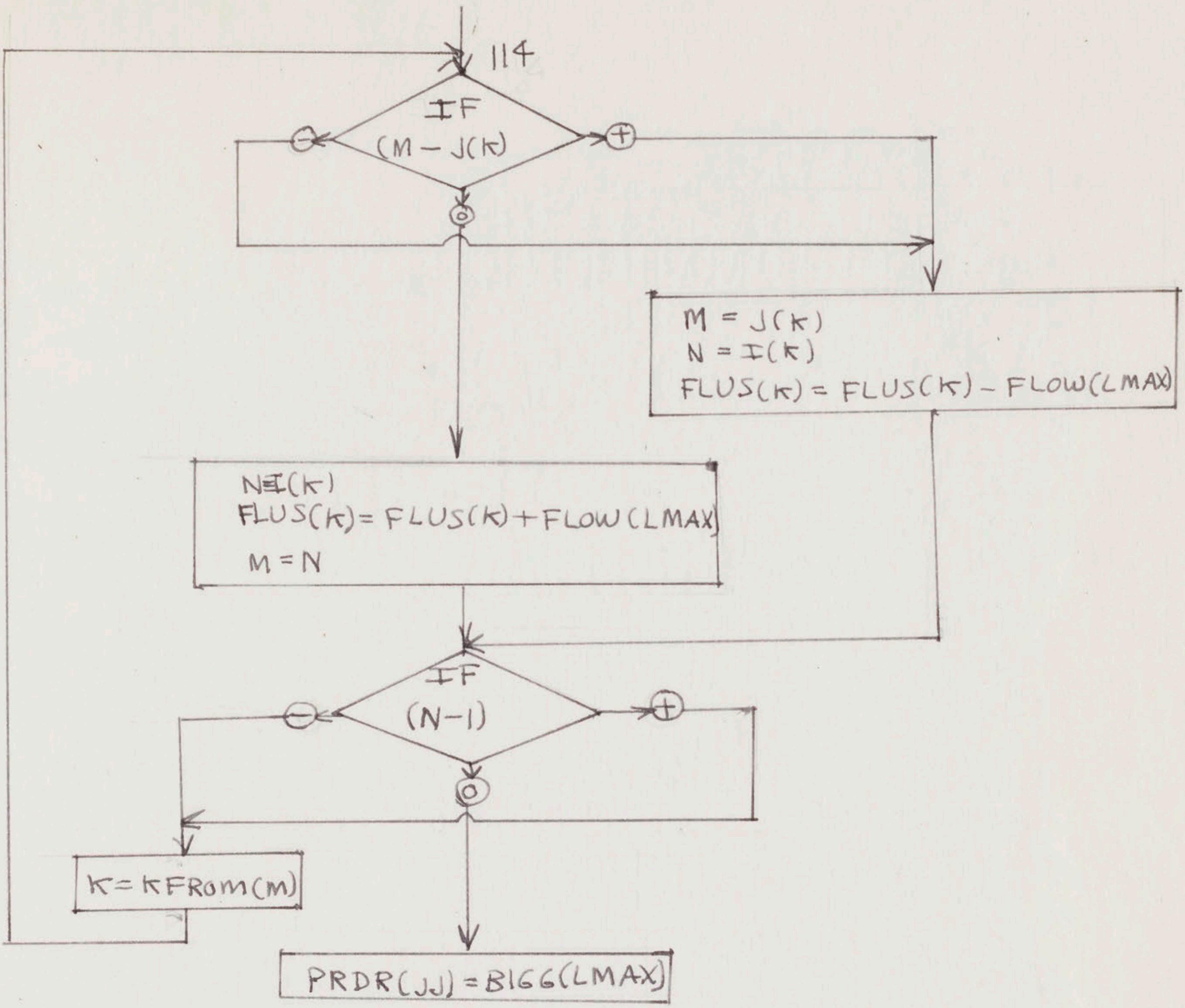
MICRO FLOW DIAGRAM FOR FLOW ALGORITHM


FORWARD ITERATION
NCK = 1
DO 30 L=1, LMAX
BIGG(L) = 0.0
FLOW(L) = 0.0
30 KFROM(L) = 0.0
N=1
FLOW = 30000.0
32
L=2
K=1
33
IF
(J(K)-L)
TIME = BIGG(N) + TN(K)
QNT = SLOPE(K) - FLUS(K)
M = J(K)
N = I(K)
IF
(FLUS(K)-SLOPE(K))
TIME = BIGG(N)+TCR(N)
QNT = 30000.0
IF
(TIME-BIGG(M))
33
32
NCK=1
BIGG(M) = TIME
KFROM = K
IF
(QNT-FLOW(N))
FLOW(M) = FLOW(N)
TO 35
TO 35
TO 62



62
TO 35
TO 33
FLOW(M) = QNT
TO 35
35
IF (K-KMAX)
K = K+1
IF (L-LMAX)
TO 32
L=LH
IF (NCK-1)
TO 110
TO 110
BACKWARD ITERATION
NCK=0
L=LMAX
K = KMAX
IF (J(K)-L)
74
M=J(K)
N=I(K)
IF (K-1)
IF (FLUS(K))
K=K-1
IF (L-1)
TO 74
TO 85
TO 100
L=L-1


85
TO 100
TO 74
IF (FLUS(K)-SLOPE(K))
TIME = BIGG(M)- TCR(K)
QNT = FLUS(K)- SLOPE(K)
TIME= BIGG(M)- TN(K)
QNT= FLUS(K)
IF (TIME -BIGG(N))
BIGG(N)= TIME
KFROM(N)=K
NCK=1
IF (QNT-FLOW(M))
FLOW(N)=FLOW(M)
FLOW(N)=QNT
100
IF (NCK-1)
TO 31
NCK=0
ASS. FLOW TO ACT.
110
K= KFROM(LMAX)
M = J(K)
TO 114


114
IF
(M – J(K)
⊖
⊕
0
M = J(K)
N = I(K)
FLUS(K) = FLUS(K) – FLOW(LMAX)
N=I(K)
FLUS(K)= FLUS(K)+FLOW(LMAX)
M = N
IF
(N–1)
⊖
⊕
0
K=KFROM(M)
PRDR(JJ) = BIGG(LMAX)

PROGRAM LISTING

```
      CHAIN (1,A4)
      LIST
      DIMENSION K(2000),IPRE(1000,7),IJD(500,2), IPRD(1000,6),ACT(4),
     1VAR(12),VOR(3)
      COMMON K,KDMAX,NMAX,IDUM,IOK,VOR,IERR,LMAX,FIXA,FIXB,DURA,VAR,
     1IPRDR,PVC,POH,PTC,ADD,FIR,FIN,IJD,IPRE
      N=1
      NDON=1
      IOK = 0
      REWIND 9
      REWIND 10
 1000 FORMAT(12A6)
      READ 1000, (VAR(I),I=1,12)
 1001 FORMAT(5F6.0,3A6,19X,5I1)
      READ 1001,FIR,FIN,FIXA,FIXB,DURA,(VOR(II),II=1,3),IDUM,IERR,IJACK,
     1IPRT,IPUN
 1002 FORMAT (7I4,3F3.0,2F5.0,4A6)
 1005 READ 1002, I,K(N),(IPRE(N,II),II=1,5),TN,TCR,PRT,CSTN,CSTCR,
     1(ACT(II),II=1,4)
      IF(IOK) 1009,1007,1009
 1007 IF(I) 1008,1009,1008
 1008 IOK = 1000
 1009 WRITE TAPE 9, I,K(N),TN,TCR,PRT,CSTN,CSTCR,(ACT(II),II=1,4)
      IF (K(N)-NDON)  18,18,17
   17 NDON=K(N)
   18 IF (K(N))  3,4,3
    3 N=N+1
      GO TO 1005
    4 NMAX=N-1
      DO 9000 JJ=1,NMAX
      I=0
      J=0
      TN=0.0
      TCR = 0.0
      PRT = 0.0
      CSTN = 0.0
      CSTCR = 0.0
 9000 WRITE TAPE 9, I,J,TN,TCR,PRT,CSTN,CSTCR,(ACT(II),II=1,4)
      REWIND 9
      IF(IOK) 19,19,10000
10000 CALL CHAIN (2,B2)
   19 LMAX=(2*NDON)+1
      CHECK FOR TOO MANY ACTIVITIES IN INPUT DATA DECK
   60 IF (NMAX-2600)  63,63,61
   61 PRINT 62, NMAX
   62 FORMAT (21H0INPUT DATA CONTAINS I5,76H   ACTIVITIES. PROGRAM DESIG
     1NED FOR 1200 ACTIVITIES. PROGRAM CANNOT PROCEED.)
      GO TO 598
   63 DO 87 N=1,NMAX
      NTEST=K(N)
      CHECK TO FIND ANY TWO ACTIVITIES THAT HAVE BEEN GIVEN THE SAME (K)
      VALUE
   67 DO 72  M=1,NMAX
   68 IF (NTEST-K(M)) 72,69,72
   69 IF (N-M) 70,72,70
   70 PRINT 71, K(N)
   71 FORMAT (41H0TWO ACTIVITIES HAVE THE SAME (K) NUMBER I5,28H    . PRO
     1GRAM CANNOT PROCEED.)
      GO TO 598
```

```
 72 CONTINUE
    CHECK TO FIND ANY JOB THAT IS PRECEDING ITSELF
 73 DO 77 II=1,5
 74 IF (K(N)-IPRE(N,II)) 77,75,77
 75 PRINT 76, K(N)
 76 FORMAT (10H0ACTIVITY I5,47H IS PRECEDED BY ITSELF. PROGRAM CANNOT
   1PROCEED.)
    GO TO 598
 77 CONTINUE
    CHECK TO FIND ANY PRECEDING ACTIVITIES THAT ARE EQUAL FOR ONE
    ACTIVITY PRECEDED
 78 DO 86 II=1,5
    LP=IPRE(N,II)
 79 IF (LP) 80,86,80
 80 DO 85 III=1,5
 81 IF (LP-IPRE(N,III)) 85,82,85
 82 IF (II-III) 83,85,83
 83 PRINT 84, K(N), LP
 84 FORMAT (10H0ACTIVITY I5,31H IS PRECEDED BY TWO ACTIVITIES I5,41H W
   1HICH ARE EQUAL. PROGRAM CANNOT PROCEED.)
    GO TO 598
 85 CONTINUE
 86 CONTINUE
 87 CONTINUE
    CHECK FOR REDUNDANT PRECEDENCE RESTRICTIONS
 88 N=1
600 II=1
601 IF (IPRE(N,II)) 605,607,605
607 IF (II-5)   608,602,602
608 II=II+1
    GO TO 601
602 IF (N-NMAX)  603,670,670
603 N=N+1
    GO TO 600
605 NP=N
606 M=IPRE(N,II)
    IBB=0
    IP=II
    DO 610 N=1,NMAX
    IF (K(N)-M) 610,611,610
610 CONTINUE
    CHECK TO FIND ANY PRECEDENCE ACTIVITIES THAT DO NOT HAVE A
    CORRESPONDING ACTIVITY K(N)
612 PRINT 613, IPRE(NP,IP), K(NP)
613 FORMAT (10H0ACTIVITY I5,25H WHICH PRECEDES ACTIVITY I5,65H , IS NO
   1T LISTED AS AN EXISTING ACTIVITY. PROGRAM CANNOT PROCEED.)
    GO TO 598
611 NPN=N
615 IB=1
620 IF (IPRE(N,IB)) 621,625,621
625 IF (IB-5)  626,655,655
626 IB=IB+1
    GO TO 620
621 II=1
640 IF ((IPRE(N,IB))-(IPRE(NP,II))) 645,650,645
650 IF (IPRE(NP,II)) 651,645,651
651 IPRE(NP,II)=0
    GO TO 625
645 IF (II-5)  646,625,625
```

```
646 II=II+1
    GO TO 640
655 IF (IBB-5)  656,627,627
627 N=NP
630 IF (IP-5)  631,602,602
631 II=IP+1
635 IF (IPRE(N,II)) 606,636,606
636 IP=II
    GO TO 630
656 IBB=IBB+1
657 IF (IPRE(NPN,IBB)) 658,655,658
658 M=IPRE(NPN,IBB)
    DO 660 N=1,NMAX
    IF (K(N)-M) 660,615,660
660 CONTINUE
    ORDER AND CLOSE UP PRECEDENCE ARRAY
670 DO 699 N=1,NMAX
671 DO 696 L=1,5
672 IF (IPRE(N,1)) 673,676,673
673 NUMS=IPRE(N,1)
674 IF (IPRE(N,2)) 675,678,675
675 IF (NUMS-IPRE(N,2)) 678,677,677
676 IF (IPRE(N,2)) 677,680,677
677 NUMS=IPRE(N,2)
678 IF (IPRE(N,3)) 679,682,679
679 IF (NUMS-IPRE(N,3)) 682,681,681
680 IF (IPRE(N,3)) 681,684,681
681 NUMS=IPRE(N,3)
682 IF (IPRE(N,4)) 683,686,683
683 IF (NUMS-IPRE(N,4)) 686,685,685
684 IF (IPRE(N,4)) 685,688,685
685 NUMS=IPRE(N,4)
686 IF (IPRE(N,5)) 687,692,687
687 IF (NUMS-IPRE(N,5)) 692,691,691
688 IF (IPRE(N,5)) 691,689,691
689 DO 690 LL=L,5
    IPRD(N,LL)=0
690 CONTINUE
    GO TO 697
691 NUMS=IPRE(N,5)
    IPRE(N,5)=0
    GO TO 695
692 DO 693 II=1,4
    IF (IPRE(N,II)-NUMS) 693,694,693
693 CONTINUE
694 IPRE(N,II)=0
695 IPRD(N,L)=NUMS
696 CONTINUE
697 DO 698 L=1,5
    IPRE(N,L)=IPRD(N,L)
698 CONTINUE
699 CONTINUE
  5 DO 7  N=1,NMAX
  6 DO  7  II=1,6
    IPRD(N,II)=0
  7 CONTINUE
    DO 16  N=1,NMAX
    DO 16  II=6,7
    IPRE(N,II)=0
```

```
 16 CONTINUE
  8 KD=0
    START OF MAIN PROGRAM
  9 DO 350  NL=1,NMAX
 10 IF (IPRD(NL,6)-2)  11,350,11
 11 IF (IPRE(NL,1))  20,12,20
    CHECKING FOR COMMON INITIAL NODE
 12 N=NL
    IPRE(N,6)=1
 13 IF (IPRE(N,7)) 15,14,15
 14 IPRE(N,7) =(2*K(N))+1
 15 IPRD(NL,6)=2
    GO TO 350
    NUMBER ACTUAL ACTIVITY NODES
 20 IF (IPRE(NL,2))  24,21,24
 21 IPRD(NL,6)=1
    GO TO 50
 24 DO 48  NN=1,NMAX
 25 IF (IPRE(NN,1))  26,48,26
 26 IF (NL-NN)  27,48,27
 27 DO 29  IIN=1,5
 28 IF ((IPRE(NL,IIN))-(IPRE(NN,IIN)))  31,29,31
 29 CONTINUE
 30 IPRD(NN,6)=1
    GO TO 48
 31 IIN=1
 32 DO 37  II=1,5
 33 IF ((IPRE(NL,IIN))-(IPRE(NN,II)))  37,34,37
 34 IF (IIN-5)  35,48,48
 35 IIN=IIN+1
 36 IF (IPRE(NL,IIN))  32,48,32
 37 CONTINUE
 38 DO 44  IIN=1,5
 39 IF (IPRE(NL,IIN))  40,48,40
 40 DO 43  II=1,5
 41 IF ((IPRE(NL,IIN))-(IPRE(NN,II)))  43,42,43
 42 IPRD(NL,IIN)=IPRE(NL,IIN)
    GO TO 44
 43 CONTINUE
 44 CONTINUE
 48 CONTINUE
 49 IPRD(NL,6)=1
 50 DO 325   NK=NL,NMAX
 51 IF ((IPRD(NK,6))-1)  325,52,325
 52 IF (NL-NK)  370,53,370
370 IF (IPRE(NK,6))  372,377,372
372 DO 376  JC=1,2
373 DO 376  JN=1,KDMAX
374 IF ((IJD(JN,JC))-(IPRE(NK,6)))  376,375,376
375 IJD(JN,JC)=IPRE(NL,6)
376 CONTINUE
    GO TO 378
377 IPRE(NK,6)=IPRE(NL,6)
378 IF (IPRE(NK,7))  325,379,325
379 IPRE(NK,7)=(2*K(NK))+1
    GO TO 325
 53 DO 320  II=1,5
 54 M=IPRE(NK,II)
    IF (M)  55,325,55
```

```
 55 DO 57   N=1,NMAX
 56 IF (K(N)-M)  57,58,57
 57 CONTINUE
 58 IF ((IPRE(NK,II))-(IPRD(NK,II))) 100,200,100
    NUMBER DUMMY ACTIVITY NODES
200 KD=KD+1
230 IF (IPRE(N,7))  232,231,232
231 ID=(2*K(N))+1
    IPRE(N,7)=ID
    GO TO 233
232 ID=IPRE(N,7)
233 IJD(KD,1)=ID
234 N=NK
240 IF (IPRE(N,6))  242,241,242
241 JD=2*K(N)
    IPRE(N,6) =JD
    GO TO 243
242 JD=IPRE(N,6)
243 IJD(KD,2)=JD
244 KDMAX=KD
250 IF (IPRE(N,7)) 320,251,320
251 IPRE(N,7) =(2*K(N))+1
    GO TO 320
100 IF (IPRE(NK,2))  300,105,300
105 IF (IPRE(N,7))  115,110,115
110 IPRE(N,7)=(2*K(N))+1
115 IPRE(NK,6)=IPRE(N,7)
    GO TO 310
300 IF (IPRE(NK,6))  305,301,305
301 IPRE(NK,6)=2*K(NK)
305 IF ((IPRE(N,7))-(IPRE(NK,6)))  306,310,306
306 IPRE(N,7)=IPRE(NK,6)
    DO 309  JC=1,2
    DO 309  JN=1,KDMAX
307 IF ((IJD(JN,JC))-(IPRE(N,7)))  309,308,309
308 IJD(JN,JC)=IPRE(NL,6)
309 CONTINUE
310 IF (IPRE(NK,7))  320,311,320
311 IPRE(NK,7)=(2*K(NK))+1
320 CONTINUE
325 CONTINUE
330 DO 337  N=1,NMAX
335 IF (IPRD(N,6)-1)  337,336,337
336 IPRD(N,6)=2
337 CONTINUE
350 CONTINUE
    DUMMY REDUCTION
701 DO 875  KD=1,KDMAX
702 IF (IJD(KD,1))  703,875,703
703 DO 705  LP=1,NMAX
704 IF (IJD(KD,1)-IPRE(LP,6))  705,875,705
705 CONTINUE
706 DO 778  LQ=1,NMAX
707 IF (IJD(KD,2)-IPRE(LQ,7))  778,710,778
710 DO 918  KF=1,KDMAX
711 IF (IJD(KF,1)-IJD(KD,1))  918,712,918
712 IF (KF-KD)  913,918,913
913 DO 916  KG=1,KDMAX
914 IF (IJD(KG,2)-IJD(KF,2))  916,915,916
```

```
15 IF (IJD(KG,1)-IJD(KD,2))  916,918,916
16 CONTINUE
   MLP=0
   MLQ=0
   MLR=0
   MLS=0
   MLT=0
   MLP=IJD(KD,1)
   M=KD+1
   DO 722  KPO=M,KDMAX
713 IF ((IJD(KD,2))-(IJD(KPO,2)))  722,714,722
714 IF (MLQ)  716,715,716
715 MLQ=IJD(KPO,1)
    GO TO 722
716 IF (MLR)  718,717,718
717 MLR=IJD(KPO,1)
    GO TO 722
718 IF (MLS)  720,719,720
719 MLS=IJD(KPO,1)
    GO TO 722
720 IF (MLT)  722,721,722
721 MLT=IJD(KPO,1)
722 CONTINUE
    IF (MLQ)   723,875,723
723 MLPA=0
    MLPB=0
    MLPC=0
    MLPD=0
724 MLPE=0
    MLPA=IJD(KD,2)
725 DO 737  KPP=M,KDMAX
726 IF ((IJD(KPP,1))-(IJD(KD,1)))  737,729,737
729 IF (MLPB)  731,730,731
730 MLPB=IJD(KPP,2)
    GO TO 737
731 IF (MLPC)  733,732,733
732 MLPC=IJD(KPP,2)
    GO TO 737
733 IF (MLPD)  735,734,735
734 MLPD=IJD(KPP,2)
    GO TO 737
735 IF (MLPE)  737,736,737
736 MLPE=IJD(KPP,2)
737 CONTINUE
    IF (MLPB)  738,875,738
738 MDRP=MLPB
739 DO 764  KPP=KD,KDMAX
740 IF (IJD(KPP,2)-MDRP)  764,741,764
741 IF (IJD(KPP,1)-MLP)  742,876,742
742 MLP=0
    GO TO 765
876 IF ((KPP+1)-KDMAX)  877,747,747
877 IF (IJD(KPP+1,2)-MDRP)  747,743,747
743 IF (IJD(KPP+1,1)-MLQ)  747,878,747
878 IF ((KPP+2)-KDMAX)  879,749,749
879 IF (IJD(KPP+2,2)-MDRP)  749,744,749
744 IF (IJD(KPP+2,1)-MLR)  749,880,749
880 IF ((KPP+3)-KDMAX)  881,751,751
881 IF (IJD(KPP+3,2)-MDRP)  751,745,751
```

```
745 IF (IJD(KPP+3,1)-MLS)  751,882,751
882 IF ((KPP+4)-KDMAX)  883,753,753
883 IF (IJD(KPP+4,2)-MDRP) 753,746,753
746 IF (IJD(KPP+4,1)-MLT)  753,755,753
747 IF (MLQ)  748,749,748
748 MLQ=0
749 IF (MLR)  750,751,750
750 MLR=0
751 IF (MLS)  752,753,752
752 MLS=0
753 IF (MLT)  754,755,754
754 MLT=0
755 IF (MDRP-MLPB)  758,756,758
756 IF (MLPC)  757,765,757
757 MDRP=MLPC
    GO TO 764
758 IF (MDRP-MLPC)  761,759,761
759 IF (MLPD)  760,765,760
760 MDRP=MLPD
    GO TO 764
761 IF (MDRP-MLPD)  765,762,765
762 IF (MLPE)  763,765,763
763 MDRP=MLPE
764 CONTINUE
765 IF (MLP)  766,875,766
766 IF (MLQ)  767,875,767
767 MLV=MLQ
768 DO 771  KPQ=KD,KDMAX
    IF (IJD(KPQ,1)-MLV)  771,769,771
769 IJD(KPQ,1)=0
770 IJD(KPQ,2)=0
771 CONTINUE
890 IF (MLV-MLQ)  891,772,891
891 IF (MLV-MLR)  892,774,892
892 IF (MLV-MLS)  854,776,854
772 IF (MLR)  773,774,773
773 MLV=MLR
    GO TO 768
774 IF (MLS)  775,776,775
775 MLV=MLS
    GO TO 768
776 IF (MLT)  777,854,777
777 MLV=MLT
    GO TO 768
854 DO 866  LPP=1,NMAX
855 IF ((IPRE(LPP,7))-MLQ)  857,856,857
856 IPRE(LPP,7)=MLP
857 IF (MLR)  858,866,858
858 IF ((IPRE(LPP,7))-MLR)  860,859,860
859 IPRE(LPP,7)=MLP
860 IF (MLS)  861,866,861
861 IF ((IPRE(LPP,7))-MLS)  863,862,863
862 IPRE(LPP,7)=MLP
863 IF (MLT)  864,866,864
864 IF ((IPRE(LPP,7))-MLT)  866,865,866
865 IPRE(LPP,7)=MLP
866 CONTINUE
    GO TO 875
918 CONTINUE
```

```
      GO TO 793
  778 CONTINUE
  779 DO 871  KPO=1,KDMAX
  780 IF (KPO-KD)  781,871,781
  781 IF ((IJD(KD,2))-(IJD(KPO,2)))  871,782,871
  782 DO 870  KPP=1,KDMAX
  783 IF (KPO-KPP)  784,870,784
  784 IF ((IJD(KPO,1))-(IJD(KPP,1)))  925,924,925
  924 MDC=IJD(KPP,2)
      GO TO 785
  925 IF (KPP-KDMAX)  870,926,870
  926 NDC=1
  929 IF (IJD(KPO,1)-IPRE(NDC,6))  928,927,928
  927 MDC=IPRE(NDC,7)
  785 DO 869  KPQ=1,KDMAX
  786 IF (KPP-KPQ)  787,869,787
  787 IF (MDC-IJD(KPQ,2))  869,788,869
  788 IF ((IJD(KD,1))-(IJD(KPQ,1)))  869,789,869
  789 IF (KPQ-KD)  871,869,871
  869 CONTINUE
      IF (KPP-KDMAX)  870,928,870
  928 IF (NDC-NMAX)  930,870,870
  930 NDC=NDC+1
      GO TO 929
  870 CONTINUE
      GO TO 875
  871 CONTINUE
  793 M=KD+1
  794 DO 797  KPS=M,KDMAX
  795 IF ((IJD(KPS,1))-(IJD(KD,1)))  797,796,797
  796 IJD(KPS,1)=IJD(KD,2)
  797 CONTINUE
  850 DO 867  L=6,7
  851 DO 867  LPP=1,NMAX
  852 IF ((IPRE(LPP,L))-(IJD(KD,1)))  867,853,867
  853 IPRE(LPP,L)=IJD(KD,2)
  867 CONTINUE
  798 IJD(KD,1)=0
  799 IJD(KD,2)=0
  875 CONTINUE
      ELIMINATE DUPLICATE AND REDUNDANT DUMMY JOBS
  385 KDRD=1
      KDX=KDMAX
      KD=1
  386 IF (IJD(KD,1))  387,398,387
  387 DO 396  KDA=1,KDX
  388 IF ((IJD(KD,2))-(IJD(KDA,2)))  396,389,396
  389 IF (KD-KDA)  390,396,390
  390 IF ((IJD(KD,1))-(IJD(KDA,1)))  392,391,392
  391 IJD(KDA,1)=0
      IJD(KDA,2)=0
      GO TO 396
  392 DO 395  KDB=1,KDX
      IF ((IJD(KDA,1))-(IJD(KDB,2)))  395,393,395
  393 IF ((IJD(KD,1))-(IJD(KDB,1)))  395,394,395
  394 IJD(KD,1)=0
      IJD(KD,2)=0
      GO TO 398
  395 CONTINUE
```

```
396 CONTINUE
    IJD(KDRD,1)=IJD(KD,1)
    IJD(KDRD,2)=IJD(KD,2)
    IF (KD-KDX)  397,399,399
397 KDRD=KDRD+1
398 KD=KD+1
    GO TO 386
399 KDMAX=KDRD
    CHECKING FOR COMMON TERMINAL NODE
402 LARGE=0
    DO 435  NN=1,NMAX
    DO 425  II=1,5
    DO 420  N=1,NMAX
415 IF ((IPRE(N,II))-K(NN)) 420,435,420
420 CONTINUE
425 CONTINUE
427 IPRE(NN,7) =0
    IF (K(NN)-LARGE) 435,431,431
431 LARGE=K(NN)
435 CONTINUE
437 DO 445  NN=1,NMAX
    IF (IPRE(NN,7))  445,441,445
441 IPRE(NN,7) =(2*(LARGE))+1
445 CONTINUE
    NODE RENUMBERING USING LOWEST NUMBERED DIGITS
450 DO 481  NK=1,NMAX
    KK=NK
452 NM=1
455 IF (IPRE(NM,6)-KK)  482,456,482
482 IF (NM-NMAX)  483,484,484
483 NM=NM+1
    GO TO 455
484 DO 485  KD=1,KDMAX
    IF (IJD(KD,1)-KK)  485,456,485
485 CONTINUE
486 IF (KK-LMAX)  487,490,490
487 KK=KK+1
    GO TO 452
456 DO 459  N=1,NMAX
457 IF (IPRE(N,7)-KK)  459,458,459
458 IF (IPRE(N,6)-NK)  459,486,486
459 CONTINUE
460 DO 480  NN=6,7
    DO 475  N=1,NMAX
461 IF (IPRE(N,NN)-KK)  463,462,463
462 IPRE(N,NN)=NK
    GO TO 475
463 IF (IPRE(N,NN)-NK)  475,464,475
464 IPRE(N,NN)=KK
475 CONTINUE
480 CONTINUE
465 DO 471  M=1,2
466 DO 471  KD=1,KDMAX
467 IF (IJD(KD,M)-KK)  469,468,469
468 IJD(KD,M)=NK
    GO TO 471
469 IF (IJD(KD,M)-NK)  471,470,471
470 IJD(KD,M)=KK
471 CONTINUE
```

```
 481 CONTINUE
     CHECK TO FIND CLOSED LOOP IN ARROW DIAGRAM LOGIC
 490 DO 495  N=1,NMAX
 491 IF ((IPRE(N,7))-(IPRE(N,6)))  492,492,495
 492 PRINT 493, K(N)
 493 FORMAT (10H1ACTIVITY I5,85H IS PART OF A CLOSED LOOP. THIS VIOLATE
    1S ARROW DIAGRAM LOGIC. PROGRAM CANNOT PROCEED.)
     GO TO 598
 495 CONTINUE
     ORDER FINAL ACTIVITY PRINT OUT
1200 IF (IPRT)  1201,1099,1201
1201 PRINT 1202
1202 FORMAT  (37H1CRITICAL PATH ARROW DIAGRAM SOLUTION)
1211 FORMAT(1H ,12A6)
     PRINT 1211,(VAR(II),II=1,12)
     PRINT 1203
1203 FORMAT (93H0K=ACTIVITY IDENTIFICATION NUMBER       I=TAIL OF ACTIVI
    1TY ARROW       J=HEAD OF ACTIVITY ARROW)
 505 L=1
     LR=0
 506 LRG=1
 507 N=0
 508 LRGR=1
 509 M=1
 510 IF (IPRE(M,6)-LRG)  515,520,515
 515 IF (M-NMAX)  516,517,517
 516 M=M+1
     GO TO 510
 517 LRG=LRG+1
     GO TO 509
 520 IF (N-NMAX)  523,521,521
 521 LRGR=LRGR+1
 522 N=0
     GO TO 520
 523 N=N+1
 525 IF (IPRE(N,7)-LRGR)  520,526,520
 526 LR=LR+1
     NLR=1
 527 IF (K(NLR)-LR)  528,534,528
 528 IF (NLR-NMAX)  529,526,526
 529 NLR=NLR+1
     GO TO 527
 534 IPRD(L,1)=NLR
     IPRD(L,2)=M
     IPRD(L,3)=N
 535 IF (L-NMAX)  536,555,555
 536 L=L+1
     GO TO 515
     ORDER FINAL DUMMY PRINT OUT
 555 KD=1
 556 LRG=1
 557 N=0
 558 LRGR=1
 559 M=1
 560 IF (IJD(M,1)-LRG)  565,570,565
 565 IF (M-NMAX)  566,567,567
 566 M=M+1
     GO TO 560
 567 LRG=LRG+1
```

```
      GO TO 559
  570 IF (N-NMAX)  573,571,571
  571 LRGR=LRGR+1
  572 N=0
      GO TO 570
  573 N=N+1
  575 IF (IJD(N,2)-LRGR)  570,576,570
  576 IPRD(KD,4)=M
      IPRD(KD,5)=N
  585 IF (KD-KDMAX)  586,590,590
  586 KD=KD+1
      GO TO 565
  590 PRINT 1204
 1204 FORMAT (104H0   K              ACTIVITY                 I        J
     1        K            I        J               K          I       J)
      PRINT ACTUAL ACTIVITIES
      DO 1206  L=1,NMAX
      M=IPRD(L,2)
      N=IPRD(L,3)
      READ TAPE 9, A,B,C,D,E,F,G, (ACT(II), II=1,4)
 1205 FORMAT (I5,3H    4A6,2I7,I14,I9,I6,I14,I9,I6)
 1206 PRINT 1205, K(L), (ACT(II), II=1,4),    IPRE(L,6), IPRE(L,7),
     1K(M), IPRE(M,6), IPRE(M,7), K(N), IPRE(N,6), IPRE(N,7)
      REWIND 9
      PRINT DUMMY ACTIVITIES
      PRINT 1207
 1207 FORMAT (118H1DUMMY ACTIVITIES    KD=DUMMY ACTY IDENTIFICATION NUMB
     1ER    ID=TAIL OF DUMMY ACTY ARROW    JD=HEAD OF DUMMY ACTY ARROW)
      PRINT 1208
 1208 FORMAT (58H0  KD              ID   JD             KD            ID
     1 JD)
      DO 1210  KD=1,KDMAX
      M=IPRD(KD,4)
      N=IPRD(KD,5)
 1209 FORMAT (I5,9H  DUMMY  ,2I5,I15,9H  DUMMY  ,2I5)
 1210 PRINT 1209,, KD, IJD(M,1), IJD(M,2), KD, IJD(N,1), IJD(N,2)
 1099 IF(IPUN) 1100,1197,1100
 1100 DO 1125  N=1,NMAX
      READ TAPE 9, X,XX,TN,TCR,PRT,CSTN,CSTCR,(ACT(II),II=1,4)
 1124 FORMAT(2I4,20X,3F3.0,2F5.0,4A6)
 1125 PUNCH 1124,IPRE(N,6),IPRE(N,7),TN,TCR,PRT,CSTN,CSTCR,
     1(ACT(II),II=1,4)
      DO 1135 N=1,KDMAX
 1134 FORMAT(2I4,45H                                      DUMMY)
 1135 PUNCH 1134,IJD(N,1),IJD(N,2)
      REWIND 9
 1197 IF(IJACK) 598,598,1198
 1198 DO 1199 N=1,NMAX
 1199 WRITE TAPE 10,IPRE(N,6),IPRE(N,7)
      DO 1250 N=1,KDMAX
 1250 WRITE TAPE 10,IJD(N,1),IJD(N,2)
      CALL CHAIN (2,B2)
  598 CALL EXIT
      END
```

```
      CHAIN(2,B2)
      LIST
      DIMENSION VAR(12),FINLT(2000),FLOW(2000),KFROM(2000),BIGG(2200),
     1I(2000),J(2000),SLOPE(2000),TN(2000),FLUS(2000),TCR(2000),
     2PRDR(200),TVC(200),DAD(2200),ACT(4),VOR(3)
      COMMON FLOW,FINLT,KDMAX,KMAX,IDUM,IOK,VOR,IERR,LMAX,FIXA,FIXB,
     1DURA,VAR,IPRDR,PVC,POH,PTC,ADD,FIR,FIN,KFROM,BIGG,I,J,SLOPE,
     2TN,FLUS,TCR,PRDR,TVC,DAD
      EQUIVALENCE (FLOW,FINLT)
      REWIND 9
      REWIND 10
 READING DATA, CALCULATING LMAX,KMAX, AND NORMAL PROJECT COST
      ADD = 0.0
      COSTO =0.0
      JJ = 1
      HH=0.0
      LLL = 0
      LMAX = 0
      PTC = 10.E+12
      IF(IOK) 2,2,2005
      K=1
      DO 15 K=1,KMAX
      READ TAPE 9,X,XX,TN(K),TCR(K),PRT,CSTN,CSTCR,(ACT(II),II=1,4)
      READ TAPE 10,I(K),J(K)
      ADD = ADD + PRT
      Z = TN(K)-TCR(K)
      SLOPE(K) = (CSTCR-CSTN)/(Z)
      COSTO = COSTO + CSTN
   15 CONTINUE
      K=KMAX
      DO 13 IK=1,KDMAX
      K=K+1
      READ TAPE 10, I(K),J(K)
      TN(K) = 0.0
      TCR(K) = 0.0
   13 SLOPE(K) = 0.0
      KMAX = KMAX+KDMAX
      REWIND 9
      REWIND 10
      GO TO 2020
 2005 K=1
      NUMBER=0
 2006 READ TAPE 9,I(K),J(K),TN(K),TCR(K),PRT,CSTN,CSTCR,(ACT(II),II=1,4)
      ADD = ADD + PRT
      Z=TN(K)-TCR(K)
      SLOPE(K)=(CSTCR-CSTN)/(Z)
      COSTO = COSTO + CSTN
      IF(J(K)-I(K)) 2007,2007,2008
 2007 NUMBER = 1
 2008 IF(J(K)-LMAX) 2013,2013,2012
 2012 LMAX=J(K)
 2013 IF(J(K)) 2014,2015,2014
 2014 K = K+1
      GO TO 2006
 2015 KMAX = K-1
 CHECKING FOR TOO MANY ACTIVITIES IN INPUT DATA DECK
      REWIND 9
      REWIND 10
 2020 IF(KMAX-2600) 2030,2030,17
```

```
16  FORMAT(22H0  THE IMPUT CONTAINS I5, 62H   ACTIVITIES.  THIS  PROGR
   1AM IS DESIGNED FOR 2600 ACTIVITIES.)
7   PRINT 16, KMAX
    GO TO 530
 CHECKING IF ERROR CHECKING ROUTINE IS REQUIRED
030 IF(IERR) 550,2035,550
035 IF(NUMBER) 6175,695,6175
 CHECKING FOR TWO STARTING EVENTS
50   KK=1
     NN=0
51   K=1
52   IF(I(K)-KK) 553,561,553
53   IF(K-KMAX) 554,555,555
54   K=K+1
     GO TO 552
55   IF(KK-LMAX) 556,572,572
56   KK=KK+1
     GO TO 551
61   K=1
611 IF(J(K)-KK) 562,555,562
62   IF(K-KMAX) 563,565,565
63   K=K+1
     GO TO 5611
65   IF(NN) 566,566,567
66   KKS1 = KK
67   KKS2 = KK
     NN = NN +1
     IF(NN-1) 555,555,570
69   FORMAT(67H0  THE ARROW DIAGRAM IS INCORRECT.  TWO STARTING EVENTS
    1WERE FOUND.)
70   PRINT 569
71   FORMAT(9H EVENTS  ,I4,6H AND  ,I4,34H ARE BOTH USED AS STARTING EV
    1ENTS.)
     PRINT 571,KKS1,KKS2
     GO TO 530
 CHECKING FOR TWO ENDING EVENTS
72   KK=LMAX
     NN=0
73   K=1
74   IF(J(K)-KK) 575,580,575
75   IF(K-KMAX) 576,577,577
76   K=K+1
     GO TO 574
77   IF(KK-1) 617,617,578
78   KK=KK-1
     GO TO 573
80   K=1
81   IF(I(K)-KK) 582,577,582
82   IF(K-KMAX) 583,584,584
83   K=K+1
     GO TO 581
84   IF(NN) 594,594,595
94   KKE1=KK
95   NN = NN +1
     KKE2=KK
     IF(NN-1) 577,577,586
85   FORMAT(65H0  THE ARROW DIAGRAM IS INCORRECT.  TWO ENDING EVENTS WE
    1RE FOUND.)
86   PRINT 585
```

```
89  FORMAT(9H EVENTS  ,I4,6H AND  ,I4,32H ARE BOTH USED AS ENDING EVEN
   1TS.)
    PRINT 589,KKE1,KKE2
    GO TO 530
 NODE RENUMBERING ROUTINE
17  IF(NUMBER) 6175,690,6175
175 KN = 1
18  KK=KN
16  K = 1
20  IF(I(K)-KK) 621,631,621
21  IF(K-KMAX) 623,625,625
23  K=K+1
    GO TO 620
25  IF(KK-LMAX-1) 626,629,629
26  KK=KK + 1
    GO TO 616
29  DO 630 K=1, KMAX
    IF(J(K)-I(K)) 6303,6303,630
302 FORMAT(41H0  A  LOOP FOUND.  ERROR IN DIAGRAM LOGIC)
303 PRINT 6302
    GO TO 530
30  CONTINUE
    GO TO 690
31  K=1
35  IF(J(K)-KK) 645,640,645
40  IF(I(K)-KN) 645,625,625
45  IF(K-KMAX) 646,649,649
46  K=K+1
    GO TO 635
49  K=1
50  IF(I(K)-KK) 655,651,655
51  I(K) =KN
    GO TO 660
55  IF(I(K)-KN) 660,656,660
56  I(K) =KK
60  IF(K-KMAX) 661,664,664
61  K=K+1
    GO TO 650
64  K =1
65  IF(J(K)-KK) 670,666,670
66  J(K)=KN
    GO TO 675
70  IF(J(K)-KN) 675,671,675
71  J(K) = KK
75  IF(K-KMAX) 676,680,680
76  K=K+1
    GO TO 665
80  IF(KN-LMAX) 681,629,629
81  KN=KN + 1
    GO TO 618
90  CONTINUE
    LMAX = 1
    DO 695 K=1,KMAX
    IF(J(K)-LMAX) 695,695,692
92  LMAX = J(K)
95  CONTINUE
    DO 28 K=1,KMAX
28  FLUS(K) = 0.0
 FORWARD ITERATION
```

```
29   NCK=1
     DO 30 L = 1,LMAX
     BIGG(L) = 0.0
     FLOW(L)=0.0
30   KFROM(L) = 0
31   N = 1
     FLOW(N)=30000.0
     L=2
32   K=1
33   IF(J(K)-L) 35,43,35
43   M=J(K)
     N=I(K)
45   IF(FLUS(K)-SLOPE(K)) 49,47,47
47   TIME = BIGG(N) + TCR(K)
     QNT = 30000.0
     GO TO 50
49   TIME = BIGG(N)+TN(K)
     QNT = SLOPE(K) - FLUS(K)
50   IF(TIME-BIGG(M)) 35,35,51
51   NCK=1
     BIGG(M) = TIME
     KFROM(M) = K
60   IF(QNT-FLOW(N)) 62,62,61
61   FLOW(M) = FLOW(N)
     GO TO 35
62   FLOW(M) = QNT
35   IF(K-KMAX) 36,37,37
36   K=K+1
     GO TO 33
37   IF(L-LMAX) 38,65,65
38   L=L+1
     GO TO 32
65   IF(NCK-1) 110,70,110
 BACKWARD ITERATION
70   NCK = 0
     L = LMAX
71   K=KMAX
72   IF(J(K)-L)  74,80,74
74   IF(K-1) 75,76,75
75   K=K-1
     GO TO 72
76   IF(L-1) 77,100,77
77   L=L-1
     GO TO 71
80   M=J(K)
     N=I(K)
     IF(FLUS(K)) 74,74,85
85   IF(FLUS(K)-SLOPE(K)) 87,87,86
86   TIME = BIGG(M)-TCR(K)
     QNT = FLUS(K) - SLOPE(K)
     GO TO 89
87   TIME=BIGG(M)-TN(K)
     QNT = FLUS(K)
89   IF(TIME-BIGG(N)) 74,74,92
92   BIGG(N) = TIME
     KFROM(N) = K
     NCK=1
     IF(QNT-FLOW(M)) 96,96,94
94   FLOW(N) =FLOW(M)
```

```
    GO TO 74
96   FLOW(N) = QNT
     GO TO 74
00   IF(NCK-1) 110,101,110
01   NCK=0
     GO TO 31
 ASSIGNING FLOW TO ACTIVITIES
10   K=KFROM(LMAX)
     M=J(K)
14   IF(M-J(K)) 116,117,116
16   M=J(K)
     N=I(K)
     FLUS(K)=FLUS(K)-FLOW(LMAX)
     GO TO 120
17   N=I(K)
     FLUS(K)=FLUS(K)+FLOW(LMAX)
     M=N
20   IF(N-1) 126,128,126
26   K=KFROM(M)
     GO TO 114
28   PRDR(JJ) = BIGG(LMAX)
 COMPUTING PROJECT VARIABLE COST
75   IF(JJ-1) 177,176,177
76   TVC(JJ) = COSTO
     GO TO 190
77   JJ = JJ-1
     BB= PRDR(JJ)
     JJ=JJ+1
     TVC(JJ) = COSTO+(BB-PRDR(JJ))*(HH)
     COSTO = TVC(JJ)
90   HH=HH+ FLOW(LMAX)
CHECKING FOR MINIMUM TOTAL PROJECT COST
     MARIAN = 1
00   IF(FIXA) 201,250,201
01   IF(PRDR(JJ)-DURA) 202,202,203
02   A = TVC(JJ) + (FIXA*PRDR(JJ))
     GO TO 204
03   A = TVC(JJ) +(FIXB*PRDR(JJ))
04   IF(PTC-A) 205,2041,2041
041  PTC = A
     PVC = TVC(JJ)
     POH = PTC-PVC
     IPRDR = BIGG(LMAX)
     MARIAN = 0
05   IF(PRDR(JJ)-DURA) 206,206,207
06   SLT=FIXA-HH
     GO TO 209
07   SLT= FIXB-HH
09   IF(SLT) 210,210,250
10   IF(LLL) 250,211,250
11   LLL=LLL + 1
     GO TO 400
CHECKING FOR RANGE OF WANTED OUTPUT
50   IF(FIR) 255,400,255
55   IF(PRDR(JJ)-FIN) 256,400,500
56   IF(PRDR(JJ)-FIR) 500,400,400
30   FORMAT(11H1SCHEDULE  ,I4,20H PROJECT DURATION = ,F6.0,23H TOTAL VA
    1RIABLE COST = ,F8.0)
140  FORMAT(124H0INPUT                                            CRITICAL
```

```
  1     EARLIEST  LATEST  EARLIEST  LATEST  TOTAL  FREE   INDEP  ACTIV
  2ITY)
240 FORMAT(124H ORDER   ACTIVITY NAME          I      J    PATH   DURATI
  1ON      START   START    FINISH  FINISH  FLOAT  FLOAT  FLOAT      C
  2OST)
41  FORMAT(1H ,I4,4A6,2I6,5H  YES,F11.0,F10.0,F8.0,F10.0,F8.0,
  13F7.0,F10.0)
42  FORMAT(1H ,I4,4A6,2I6,5H   NO,F11.0,F10.0,F8.0,F10.0,F8.0,
  13F7.0,F10.0)
99  FORMAT(1H 12A6)
00  PRINT 130,JJ,PRDR(JJ),TVC(JJ)
    PRINT 399,(VAR(I),I=1,12)
    PRINT 140
    PRINT 240
CALCULATION OF FLOATS AND SCHEDULING
25  DO 426 I=1,LMAX
26  FINLT(I)=BIGG(LMAX)
    L = LMAX
30  K=KMAX
34  M = J(K)
    N = I(K)
40  IF(M-L) 450,441,450
41  IF(BIGG(M)-BIGG(N)-TN(K)) 443,442,442
42  SCDR=TN(K)
    GO TO 444
43  SCDR = BIGG(M)-BIGG(N)
44  START = FINLT(M)-SCDR
45  IF(START-FINLT(N)) 446,450,450
46  FINLT(N)=START
50  IF(K-1) 452,452,451
51  K=K-1
    GO TO 434
52  IF(L-1) 460,460,453
53  L = L-1
    GO TO 430
 OUTPUT PRINTING
460 K=1
    KPAGE=1
    KOUNT=1
461 READ TAPE 9, IB, JB, Y, YY, PRT, CSTN, CSTCR, (ACT(II),II=1,4)
462 IF (SLOPE(K))  464,463,464
463 IF (IDUM)  464,6048,464
464 M=J(K)
    N=I(K)
    IF (BIGG(M)-BIGG(N)-TN(K))  466,465,465
465 SCDR=TN(K)
    COST=CSTN
    GO TO 467
466 SCDR=BIGG(M)-BIGG(N)
    Z=TN(K)-SCDR
    COST=CSTN+(Z*SLOPE(K))
67  FLTO = FINLT(M)-BIGG(N)-SCDR
    FLFR = BIGG(M)-BIGG(N) - SCDR
    FLIN = BIGG(M)-FINLT(N)-SCDR
    FINER = BIGG(N) + SCDR
    STLT = FINLT(M) - SCDR
000 IF (FLIN)  4001,4003,4003
001 FLIN=0.0
003 IF (IOK)  5000,5000,4004
```

```
004 IF (FLTO)  4006,4005,4006
005 PRINT 141,K,(ACT(II),II=1,4),IB,JB,SCDR,BIGG(N),STLT,FINER,
   1FINLT(M),FLTO,FLFR,FLIN,COST
    GO TO 6000
006 PRINT 142,K,(ACT(II),II=1,4),IB,JB,SCDR,BIGG(N),STLT,FINER,
   1FINLT(M),FLTO,FLFR,FLIN,COST
    GO TO 6000
000 IF (SLOPE(K))  5050,5001,5050
001 IF (FLTO)  5006,5004,5006
003 FORMAT(1H ,I4,3A6,6X,2I6,5H  YES,F11.0,F10.0,F8.0,F10.0,F8.0,
   13F7.0,F10.0)
004 PRINT 5003,K,(VOR(II),II=1,3),I(K),J(K),SCDR,BIGG(N),STLT,FINER,
   1FINLT(M),FLTO,FLFR,FLIN,COST
    GO TO 6000
005 FORMAT(1H ,I4,3A6,6X,2I6,5H   NO,F11.0,F10.0,F8.0,F10.0,F8.0,
   13F7.0,F10.0)
006 PRINT 5005,K,(VOR(II),II=1,3),I(K),J(K),SCDR,BIGG(N),STLT,FINER,
   1FINLT(M),FLTO,FLFR,FLIN,COST
    GO TO 6000
050 IF(FLTO) 5065,5055,5065
025 FORMAT(1H ,I4,4A6,2I6,5H  YES,F11.0,F10.0,F8.0,F10.0,F8.0,
   13F7.0,F10.0)
055 PRINT 5025,K,(ACT(II),II=1,4),I(K),J(K),SCDR,BIGG(N),STLT,FINER,
   1FINLT(M),FLTO,FLFR,FLIN,COST
    GO TO 6000
064 FORMAT(1H ,I4,4A6,2I6,5H   NO,F11.0,F10.0,F8.0,F10.0,F8.0,
   13F7.0,F10.0)
065 PRINT 5064,K,(ACT(II),II=1,4),I(K),J(K),SCDR,BIGG(N),STLT,FINER,
   1FINLT(M),FLTO,FLFR,FLIN,COST
000 KOUNT = KOUNT + 1
    IF(KOUNT-50) 6048,6048,6002
001 FORMAT(7H1 PAGE ,I6,10H SCHEDULE ,I6)
002 KPAGE = KPAGE + 1
003 PRINT 6001,KPAGE,JJ
    PRINT 140
    PRINT 240
    KOUNT = 1
048 IF(MARIAN) 6050,6049,6050
049 WRITE TAPE 10,SCDR,COST
050 IF(K-KMAX) 6051,500,500
051 K=K +1
    GO TO 461
00  REWIND 9
    REWIND 10
CHECKING FOR LAST SCHEDULE
    IF(HH-30000.0) 501,505,505
01  JJ=JJ+1
    GO TO 29
05  NJJ=JJ
06  FORMAT(32H1  SCHEDULE OF PROJECT VARIABLES)
    PRINT 506
07  FORMAT(84H0  SCHEDULE        DURATION      VARIABLE COST         OVERH
   1EAD        TOTAL PROJECT COST)
    PRINT 507
PRINT TABLE OF FEASIBLE SCHEDULES
    JJ = 1
071 FORMAT(I8,2F16.0,F17.0,F22.0)
08  IF(PRDR(JJ)-DURA) 509,509,510
09  TOH=PRDR(JJ)*FIXA
```

```
      GO TO 511
510   TOH=PRDR(JJ)*FIXB
511   TPC=TVC(JJ)+TOH
      PRINT 5071,JJ,PRDR(JJ),TVC(JJ),TOH,TPC
      IF(JJ-NJJ) 512,515,515
512   JJ=JJ+1
      GO TO 508
515   CONTINUE
      IF(ADD)  517,530,517
517   CALL CHAIN(3,B3)
530   CALL EXIT
      END
```

```
      CHAIN(3,B3)
      LIST
      DIMENSION FINLT(2000),ACT(4),VAR(12),FLOLD(2000),EARST(2200),
     1I(2000),J(2000),PRT(2000),SCOLD(2000),FLTO(2000),DIFT(2000),
     2SCDR(2000),VOR(3)
      COMMON FINLT,KDMAX,KMAX,IDUM,IOK,VOR,IERR,LMAX,FIXA,FIXB,DURA,VAR,
     1IPRDR,PVC,POH,PTC,ADD,FIR,FIN,FLOLD,EARST,I,J,PRT,SCOLD,FLTO,
     2DIFT,SCDR
      K = 1
  700 READ TAPE 9, A,B,C,D,PRT(K)
      READ TAPE 10, SCDR(K),COST
      IF(SCDR(K)) 702,701,702
  701 PRT(K) = 0.0
  702 IF(K-KMAX) 703,705,705
  703 K = K + 1
      GO TO 700
  705 MM =1
      REWIND 9
      REWIND 10
  712 CALL CALFL(KMAX,LMAX)
      IF(MM-1) 7181,715,7181
  715 DO 718 K=1,KMAX
      FLOLD(K)=FLTO(K)
  718 SCOLD(K)=SCDR(K)
      MM=MM+1
CCHECKING ACTIVITIES TO SEE IF TOTAL FLOAT HAS BEEN REDUCED TO ZERO
 7181 K=1
 7182 IF(PRT(K)-SCDR(K)) 7191,7191,719
  719 IF(FLTO(K)-0.50) 7191,7191,720
 7191 IF(K-KMAX) 7192,790,790
 7192 K=K+1
      GO TO 7182
  720 DO 7200 K=1, KMAX
 7200 DIFT(K)=0.0
C INCREASING DURATION OF NON-CRITICAL ACTIVITIES BY A SMALL INCREMENT
      K =1
 7201 IF(PRT(K)-SCDR(K)) 725,725,721
  721 IF(FLTO(K)-0.50)725,725,722
  722 DIFT(K) =PRT(K)-SCOLD(K)
      SCDR(K)=SCOLD(K) +(0.01*DIFT(K))
  725 IF(K-KMAX) 726,730,730
  726 K=K+1
      GO TO 7201
  730 CALL CALFL (KMAX,LMAX)
C DETERMINING ACTIVITIES THAT ARE TO BE SCHEDULED TO BECOME CRITICAL
      SFAC = 30000.0
      K=1
  734 IF(PRT(K)-SCDR(K)) 739,739,735
  735 IF(FLTO(K)-0.50) 739,739,737
  737 FACTOR=(0.01*FLOLD(K))/(FLOLD(K)-FLTO(K))
      IF(SFAC-FACTOR) 739,739,738
  738 SFAC=FACTOR
  739 IF(K-KMAX) 704,740,740
  704 K=K+1
      GO TO 734
C INCREASING DURATION OF SOME ACTIVITIES TO BECOME CRITICAL
  740 K =1
 7401 IF(PRT(K)-SCDR(K)) 7421,7421,7391
 7391 IF(FLTO(K)-0.50) 7421,7421,742
```

```
42   SCDR(K)= SCOLD(K) + (SFAC*DIFT(K))
421  IF(K-KMAX) 7422,712,712
422  K=K+1
     GO TO 7401
90   K = 1
92   IF(SCDR(K)-PRT(K)) 794,794,793
93   SCDR(K) = PRT(K)
94   IF(K-KMAX)  795,796,796
95   K=K+1
     GO TO 792
96   CONTINUE
97   CALL CALFL(KMAX,LMAX)
00   FORMAT(1H112A6)
     PRINT 800, (VAR(I),I =1,12)
005  FORMAT(80H ANALYSIS UTILIZING THE ACTIVITY PREFERRED DURATION AS A
    1 PARAMETER IN SCHEDULING)
     PRINT 8005
01   FORMAT(20H0PROJECT DURATION = ,I6,23H PROJECT DIRECT COST = ,
    1F8.0,20H PROJECT OVERHEAD = ,F8.0,22H PROJECT TOTAL COST = ,F8.0)
     PRINT 801,IPRDR,PVC,POH,PTC
802  FORMAT(124H0INPUT                                   CRITICAL
    1       EARLIEST LATEST   EARLIEST LATEST  TOTAL  FREE  INDEP  ACTIV
    2ITY)
803  FORMAT(124H ORDER    ACTIVITY NAME              I     J  PATH     DUR
    1ATION     START   START    FINISH  FINISH  FLOAT FLOAT  FLOAT      C
    2OST)
     PRINT 802
     PRINT 803
05   FORMAT(1H ,I4,4A6,2I6,5H  YES,F11.0,F10.0,F8.0,F10.0,F8.0,
    13F7.0,F10.0)
07   FORMAT(1H ,I4,4A6,2I6,5H   NO,F11.0,F10.0,F8.0,F10.0,F8.0,
    13F7.0,F10.0)
08   FORMAT(1H ,I4,3A6,6X,2I6,5H  YES,F11.0,F10.0,F8.0,F10.0,F8.0,
    13F7.0,F10.0)
09   FORMAT(1H ,I4,3A6,6X,2I6,5H   NO,F11.0,F10.0,F8.0,F10.0,F8.0,
    13F7.0,F10.0)
 OUTPUT PRINTING
     K = 1
     KPAGE =1
     KOUNT =1
10   READ TAPE 10,X,COST
     IF(IOK) 811,825,811
11   READ TAPE 9, IB,JB,A,B,C,D,E,(ACT(II),II=1,4)
     IF(COST) 814,812,814
12   IF(IDUM) 814,900,814
14   M=J(K)
     N = I(K)
     FLFR = EARST(M)-EARST(N)-SCDR(K)
     FLIN = EARST(M)-FINLT(N)-SCDR(K)
     FINER = EARST(N) + SCDR(K)
     STLT = FINLT(M)-SCDR(K)
     IF(FLIN) 815,816,816
15   FLIN=0.0
16   IF(FLTO(K)) 818,817,818
17   PRINT 805, K,(ACT(II),II=1,4),IB,JB,SCDR(J),EARST(N),STLT,FINER,
    1FINLT(M),FLTO(K),FLFR,FLIN,COST
     GO TO 900
18   PRINT 807, K,(ACT(II),II=1,4),IB,JB,SCDR(J),EARST(N),STLT,FINER,
    1FINLT(M),FLTO(K),FLFR,FLIN,COST
```

```
      GO TO 900
25    READ TAPE 9, A,B,C,D,E,F,G,(ACT(II),II=1,4)
      IF(COST) 828,826,828
26    IF(IDUM) 828,900,828
28    M=J(K)
      N=I(K)
      FLFR=EARST(M)-EARST(N)-SCDR(K)
      FLIN = EARST(M)-FINLT(N)-SCDR(K)
      FINER = EARST(N)+SCDR(K)
      STLT = FINLT(M)-SCDR(K)
      IF(FLIN) 829,830,830
29    FLIN = 0.0
30    IF(FLTO(K)) 840,835,840
835   IF(COST) 838,836,838
36    PRINT 808,K,(VOR(I),I=1,3),I(K),J(K),SCDR(K),EARST(N),STLT,FINER,
     1FINLT(M),FLTO(K),FLFR,FLIN,COST
      GO TO 900
38    PRINT 805, K,(ACT(II),II=1,4),I(K),J(K),SCDR(K),EARST(N),STLT,
     1FINER,FINLT(M),FLTO(K),FLFR,FLIN,COST
      GO TO 900
40    IF(COST) 845,845,850
45    PRINT 809,K,(VOR(I),I=1,3),I(K),J(K),SCDR(K),EARST(N),STLT,FINER,
     1FINLT(M),FLTO(K),FLFR,FLIN,COST
      GO TO 900
50    PRINT 807, K,(ACT(II),II=1,4),I(K),J(K),SCDR(K),EARST(N),STLT,
     1FINER,FINLT(M),FLTO(K),FLFR,FLIN,COST
900   IF(KOUNT-50) 1000,1000,904
03    FORMAT(7H1 PAGE ,I6)
04    KPAGE = KPAGE +1
05    PRINT 903,KPAGE
      PRINT 8005
      PRINT 802
      PRINT 803
000   KOUNT = KOUNT + 1
      IF(K-KMAX) 1001,1005,1005
001   K = K +1
      GO TO 810
005   CALL EXIT
      END
      LIST
      SUBROUTINE CALFL(KMAX,LMAX)
      DIMENSION FINLT(2000),ACT(4),VAR(12),FLOLD(2000),EARST(2200),
     1I(2000),J(2000),PRT(2000),SCOLD(2000),FLTO(2000),DIFT(2000),
     2SCDR(2000),VOR(3)
      COMMON FINLT,KDMAX,KMAX,IDUM,IOK,VOR,IERR,LMAX,FIXA,FIXB,DURA,VAR,
     1IPRDR,PVC,POH,PTC,ADD,FIR,FIN,FLOLD,EARST,I,J,PRT,SCOLD,FLTO,
     2DIFT,SCDR
00    DO 801 I=1,LMAX
01    EARST(I) = 0.0
      L=2
26    K=1
27    M=J(K)
      N=I(K)
      IF(M-L) 840,830,840
30    TIME=EARST(N)+SCDR(K)
35    IF(EARST(M)-TIME) 836,840,840
36    EARST(M) =TIME
40    IF(K-KMAX) 841,845,845
41    K=K+1
```

```
      GO TO 827
 45   IF(L-LMAX)  846,852,852
 46   L=L+1
      GO TO 826
 52   CONTINUE
 60   L=LMAX
 61   K=KMAX
      DO 863 I = 1, LMAX
 63   FINLT(I)=EARST(LMAX)
 64   M=J(K)
      N=I(K)
 70   IF(M-L) 878,872,878
 72   START=FINLT(M)-SCDR(K)
 74   IF(START-FINLT(N)) 876,878,878
 76   FINLT(N) =START
 78   IF(K-1) 881,881,880
 80   K=K-1
      GO TO 864
 81   IF(L-1) 885,885,882
 82   L=L-1
      K=KMAX
      GO TO 864
 85   CONTINUE
 86   K=1
 87   M=J(K)
      N=I(K)
      FLTO(K)=FINLT(M)-EARST(N)-SCDR(K)
      IF(K-KMAX) 888,890,890
 88   K=K+1
      GO TO 887
 90   CONTINUE
      RETURN
      END
```

PROGRAM INPUT FORMS

CPS 300
PROJECT INPUT FORM

ENGINEER________________

DATE________________

PROJECT________________

JOB NO.________________

PROJECT IDENTIFICATION CARD

VAR
72

CONTROL CARD

FIR	FIN	FIXA	FIXB	DURA	VOR	blank	I DUM	I ERR	I JACK	I PRT	I PUN
6	7 12	13 18	19 24	25 30	31 48	49 67	68	69	70	71	72

CPS 300
NETWORK DEVELOPMENT DATA INPUT FORM

ENGINEER ________________________
DATE ________________________
PROJECT ________________________

I	K	IPRE					TN	TCR	PRT	CSTN	CSTCR	ACT

CPS 300
NETWORK COMPUTATION DATA INPUT FORM

ENGINEER ______________________
DATE ______________________
PROJECT ______________________

I	J	TN	TCR	PRT	CSTN	CSTCR	ACT

EXAMPLE PROBLEM

```
C      INPUT DATA FOR SAMPLE PROBLEM USING CPS 300
*      DATA
AN EXAMPLE TO DEMONSTRATE A METHOD FOR CONSTR. PLANNING AND SCHEDULING
           200.          500.  DUMMY ACTIVITY                              11111
       1                      2. 1. 2. 500. 600. BLDG EXCAVATION
       2   1                  2. 1. 3. 500. 600. BLDG EXCAVATION
       3   1                  3. 2. 3. 1250.1400.FORM POUR-STRIP FN
       4   2   3              3. 2. 5. 1250.1400.FORM POUR-STRIP FN
       5   2   3              3. 1. 3. 450. 650. BACKFULL FOUNDT
       6   4                  2. 2. 2. 2100.2100.PLACE-GRADE GRAVEL
       7   4   5              1. 1. 1. 150. 150. BACKFILL FOUNDTN
       8   6   7              2. 1. 2. 5000.5300.ERECT STEEL
       9   8                  5. 3. 5. 3750.4050.RAFTERS AND RF DCK
      10   9                  1. 1. 1. 500. 500. SET HVAC EQUIP
      11   9                  3. 2. 3. 2250.2400.RAFTERS-ROOF DECK
      12  10                  27.20.27.2150022000HAVC COMPL SYSTEM
      13  11                  8. 5. 10.6000.6300.SHINGLES-ROOF SUB
      14   6   7              9. 7. 12.4000.4300.UNDERSLAB CONDUIT
      15   6   7              9. 7. 15.4000.4200.UNDERSLAB PLUMBING
      16  14  15              1. 1. 1. 4000.4000.POUR SLAB
      17  16                  5. 3. 10.750. 850. CERAMIC-QUARRY TIL
      18  16                  5. 4. 6. 1000.1250.INTERIOR PARTITION
      19  16                  10.8. 15.3900.4150.INTERIOT PLUMBING
      20  16                  10.8. 15.7500.8000.INTERIOR ELECTRIC
      21  17  18              5. 2. 5. 750. 900. CERAMIC QUARRY TIL
      22  18                  15.10.20.3000.3250.PAINTING
      23  18                  5. 4. 8. 1333.1393.FINISH CARPENTRY
      24  18                  5. 3. 5. 2000.2100.SHEETROCK-TAPING
      25  18                  5. 4. 5. 1000.1250.INTERIOR PARTITION
      26  18  19              10.7. 15.3900.4150.INTERIOR PLUMBING
      27  18  20              10.7. 13.1000.1060.INTERIOR EQUIP COT
      28  23  24  25  26  2710.6. 10.2667.2907.FINISH CARPENTRY
      29  24                  10.5. 12.3600.3800.ACOUSTIC TILE
      30  24                  5. 3. 10.1500.1600.RESILIENT TILE
```

TICAL PATH ARROW DIAGRAM SOLUTION

EXAMPLE TO DEMONSTRATE A METHOD FOR CONSTR. PLANNING AND SCHEDULING

CTIVITY IDENTIFICATION NUMBER I=TAIL OF ACTIVITY ARROW J=HEAD OF ACTIVITY ARROW

K	ACTIVITY	I	J	K	I	J	K	I	J
1	BLDG EXCAVATION	1	2	1	1	2	1	1	2
2	BLDG EXCAVATION	2	3	2	2	3	2	2	3
3	FORM POUR-STRIP FN	2	3	3	2	3	3	2	3
4	FORM POUR-STRIP FN	3	4	4	3	4	4	3	4
5	BACKFULL FOUNDT	3	5	5	3	5	5	3	5
6	PLACE-GRADE GRAVEL	4	6	6	4	6	6	4	6
7	BACKFILL FOUNDTN	5	6	7	5	6	7	5	6
8	ERECT STEEL	6	7	8	6	7	8	6	7
9	RAFTERS AND RF DCK	7	8	14	6	11	9	7	8
0	SET HVAC EQUIP	8	9	15	6	11	10	8	9
1	RAFTERS-ROOF DECK	8	10	9	7	8	11	8	10
2	HAVC COMPL SYSTEM	9	61	10	8	9	14	6	11
3	SHINGLES-ROOF SUB	10	61	11	8	10	15	6	11
4	UNDERSLAB CONDUIT	6	11	12	9	61	16	11	12
5	UNDERSLAB PLUMBING	6	11	13	10	61	18	12	13
6	POUR SLAB	11	12	16	11	12	17	12	14
7	CERAMIC-QUARRY TIL	12	14	17	12	14	24	13	15
8	INTERIOR PARTITION	12	13	18	12	13	19	12	16
9	INTERIOT PLUMBING	12	16	19	12	16	20	12	17
0	INTERIOR ELECTRIC	12	17	20	12	17	23	13	18
1	CERAMIC QUARRY TIL	14	61	22	13	61	25	13	18
2	PAINTING	13	61	23	13	18	26	16	18
3	FINISH CARPENTRY	13	18	24	13	15	27	17	18
4	SHEETROCK-TAPING	13	15	25	13	18	12	9	61
5	INTERIOR PARTITION	13	18	21	14	61	13	10	61
6	INTERIOR PLUMBING	16	18	29	15	61	21	14	61
7	INTERIOR EQUIP COT	17	18	30	15	61	22	13	61
8	FINISH CARPENTRY	18	61	26	16	18	28	18	61
9	ACOUSTIC TILE	15	61	27	17	18	29	15	61
0	RESILIENT TILE	15	61	28	18	61	30	15	61

MY ACTIVITIES KD=DUMMY ACTY IDENTIFICATION NUMBER ID=TAIL OF DUMMY ACTY ARROW JD=HEAD OF DUMMY ACTY ARROW

D		ID	JD
1	DUMMY	4	5
2	DUMMY	13	14
3	DUMMY	13	16
4	DUMMY	13	17
5	DUMMY	15	18

KD		ID	JD
1	DUMMY	4	5
2	DUMMY	13	14
3	DUMMY	13	16
4	DUMMY	13	17
5	DUMMY	15	18

EDULE 1 PROJECT DURATION = 50. TOTAL VARIABLE COST = 100100.
EXAMPLE TO DEMONSTRATE A METHOD FOR CONSTR. PLANNING AND SCHEDULING

UT ER	ACTIVITY NAME	I	J	CRITICAL PATH	DURATION	EARLIEST START	LATEST START	EARLIEST FINISH	LATEST FINISH	TOTAL FLOAT	FREE FLOAT	INDEP FLOAT	ACTIVITY COST
1	BLDG EXCAVATION	1	2	YES	2.	0.	0.	2.	2.	0.	0.	0.	500.
2	BLDG EXCAVATION	2	3	NO	2.	2.	3.	4.	5.	1.	1.	1.	500.
3	FORM POUR-STRIP FN	2	3	YES	3.	2.	2.	5.	5.	0.	0.	0.	1250.
4	FORM POUR-STRIP FN	3	4	YES	3.	5.	5.	8.	8.	0.	0.	0.	1250.
5	BACKFULL FOUNDT	3	5	NO	3.	5.	6.	8.	9.	1.	0.	0.	450.
6	DUMMY ACTIVITY	4	6	YES	2.	8.	8.	10.	10.	0.	0.	0.	2100.
7	DUMMY ACTIVITY	5	6	NO	1.	8.	9.	9.	10.	1.	1.	0.	150.
8	ERECT STEEL	6	7	NO	2.	10.	15.	12.	17.	5.	0.	0.	5000.
9	RAFTERS AND RF DCK	7	8	NO	5.	12.	17.	17.	22.	5.	0.	0.	3750.
0	DUMMY ACTIVITY	8	9	NO	1.	17.	22.	18.	23.	5.	0.	0.	500.
1	RAFTERS-ROOF DECK	8	10	NO	3.	17.	39.	20.	42.	22.	0.	0.	2250.
2	HAVC COMPL SYSTEM	9	61	NO	27.	18.	23.	45.	50.	5.	5.	0.	21500.
3	SHINGLES-ROOF SUB	10	61	NO	8.	20.	42.	28.	50.	22.	22.	0.	6000.
4	UNDERSLAB CONDUIT	6	11	YES	9.	10.	10.	19.	19.	0.	0.	0.	4000.
5	UNDERSLAB PLUMBING	6	11	YES	9.	10.	10.	19.	19.	0.	0.	0.	4000.
6	DUMMY ACTIVITY	11	12	YES	1.	19.	19.	20.	20.	0.	0.	0.	4000.
7	CERAMIC-QUARRY TIL	12	14	NO	5.	20.	40.	25.	45.	20.	0.	0.	750.
8	INTERIOR PARTITION	12	13	NO	5.	20.	25.	25.	30.	5.	0.	0.	1000.
9	INTERIOT PLUMBING	12	16	YES	10.	20.	20.	30.	30.	0.	0.	0.	3900.
0	INTERIOR ELECTRIC	12	17	YES	10.	20.	20.	30.	30.	0.	0.	0.	7500.
1	CERAMIC QUARRY TIL	14	61	NO	5.	25.	45.	30.	50.	20.	20.	0.	750.
2	PAINTING	13	61	NO	15.	25.	35.	40.	50.	10.	10.	5.	3000.
3	FINISH CARPENTRY	13	18	NO	5.	25.	35.	30.	40.	10.	10.	5.	1333.
4	SHEETROCK-TAPING	13	15	NO	5.	25.	35.	30.	40.	10.	0.	0.	2000.
5	INTERIOR PARTITION	13	18	NO	5.	25.	35.	30.	40.	10.	10.	5.	1000.
6	INTERIOR PLUMBING	16	18	YES	10.	30.	30.	40.	40.	0.	0.	0.	3900.
7	INTERIOR EQUIP COT	17	18	YES	10.	30.	30.	40.	40.	0.	0.	0.	10000.
8	FINISH CARPENTRY	18	61	YES	10.	40.	40.	50.	50.	0.	0.	0.	2667.
9	ACOUSTIC TILE	15	61	NO	10.	30.	40.	40.	50.	10.	10.	0.	3600.
0	RESILIENT TILE	15	61	NO	5.	30.	45.	35.	50.	15.	15.	5.	1500.
1	DUMMY ACTIVITY	4	5	NO	0.	8.	9.	8.	9.	1.	0.	0.	-0.
2	DUMMY ACTIVITY	13	14	NO	0.	25.	45.	25.	45.	20.	0.	0.	0.
3	DUMMY ACTIVITY	13	16	NO	0.	25.	30.	25.	30.	5.	5.	0.	0.
4	DUMMY ACTIVITY	13	17	NO	0.	25.	30.	25.	30.	5.	5.	0.	0.
5	DUMMY ACTIVITY	15	18	NO	0.	30.	40.	30.	40.	10.	10.	0.	0.

EDULE 2 PROJECT DURATION = 46. TOTAL VARIABLE COST = 1C034C.
EXAMPLE TC DEMONSTRATE A METHOD FOR CCNSTR. PLANNING AND SCHEDULING

UT ER ACTIVITY NAME	I	J	CRITICAL PATH	DURATION	EARLIEST START	LATEST START	EARLIEST FINISH	LATEST FINISH	TOTAL FLOAT	FREE FLOAT	INDEP FLOAT	ACTIVITY COST
1BLDG EXCAVATION	1	2	YES	2.	C.	0.	2.	2.	0.	0.	0.	500.
2BLDG EXCAVATION	2	3	NO	2.	2.	3.	4.	5.	1.	1.	1.	500.
3FCRM POUR-STRIP FN	2	3	YES	3.	2.	2.	5.	5.	0.	0.	0.	1250.
4FCRM POUR-STRIP FN	3	4	YES	3.	5.	5.	8.	8.	0.	0.	0.	1250.
5BACKFULL FOUNDT	3	5	NO	3.	5.	6.	8.	9.	1.	0.	0.	450.
6DUMMY ACTIVITY	4	6	YES	2.	8.	8.	10.	10.	0.	0.	0.	2100.
7DUMMY ACTIVITY	5	6	NO	1.	8.	9.	9.	10.	1.	1.	0.	150.
8ERECT STEEL	6	7	NO	2.	10.	11.	12.	13.	1.	0.	0.	5000.
9RAFTERS AND RF DCK	7	8	NO	5.	12.	13.	17.	18.	1.	0.	C.	3750.
0DUMMY ACTIVITY	8	9	NO	1.	17.	18.	18.	19.	1.	0.	C.	500.
1RAFTERS-ROOF DECK	8	10	NO	3.	17.	35.	20.	38.	18.	0.	0.	2250.
2HAVC COMPL SYSTEM	9	61	NO	27.	18.	19.	45.	46.	1.	1.	0.	21500.
3SHINGLES-ROOF SUB	10	61	NO	8.	20.	38.	28.	46.	18.	18.	0.	6000.
4UNDERSLAB CONDUIT	6	11	YES	9.	10.	10.	19.	19.	0.	0.	0.	4000.
5UNDERSLAB PLUMBING	6	11	YES	9.	10.	10.	19.	19.	0.	0.	0.	4000.
6DUMMY ACTIVITY	11	12	YES	1.	19.	19.	20.	20.	C.	0.	C.	4000.
7CERAMIC-QUARRY TIL	12	14	NO	5.	20.	36.	25.	41.	16.	0.	0.	750.
8INTERIOR PARTITICN	12	13	NO	5.	20.	25.	25.	30.	5.	0.	0.	1000.
9INTERIOT PLUMBING	12	16	YES	10.	20.	20.	30.	30.	0.	0.	0.	3900.
0INTERIOR ELECTRIC	12	17	YES	10.	20.	20.	30.	30.	0.	0.	0.	7500.
1CERAMIC QUARRY TIL	14	61	NO	5.	25.	41.	30.	46.	16.	16.	C.	750.
2PAINTING	13	61	NO	15.	25.	31.	40.	46.	6.	6.	1.	3000.
3FINISH CARPENTRY	13	18	NO	5.	25.	35.	30.	40.	10.	10.	5.	1333.
4SHEETROCK-TAPING	13	15	NO	5.	25.	31.	30.	36.	6.	0.	0.	2000.
5INTERIOR PARTITICN	13	18	NO	5.	25.	35.	30.	40.	10.	10.	5.	1000.
6INTERIOR PLUMBING	16	18	YES	10.	30.	30.	40.	40.	C.	0.	0.	3900.
7INTERIOR EQUIP CCT	17	18	YES	10.	30.	30.	40.	40.	0.	0.	0.	1C000.
8FINISH CARPENTRY	18	61	YES	6.	40.	40.	46.	46.	0.	0.	0.	2907.
9ACOUSTIC TILE	15	61	NO	10.	30.	36.	40.	46.	6.	6.	C.	3600.
0RESILIENT TILE	15	61	NO	5.	30.	41.	35.	46.	11.	11.	5.	1500.
1DUMMY ACTIVITY	4	5	NO	0.	8.	9.	8.	9.	1.	0.	0.	-0.
2DUMMY ACTIVITY	13	14	NO	0.	25.	41.	25.	41.	16.	0.	0.	0.
3DUMMY ACTIVITY	13	16	NO	0.	25.	30.	25.	30.	5.	5.	C.	0.
4DUMMY ACTIVITY	13	17	NO	0.	25.	30.	25.	30.	5.	5.	0.	0.
5DUMMY ACTIVITY	15	18	NO	0.	30.	40.	30.	40.	10.	10.	4.	0.

EDULE 3 PROJECT DURATION = 46. TOTAL VARIABLE COST = 100340.
EXAMPLE TO DEMONSTRATE A METHOD FOR CONSTR. PLANNING AND SCHEDULING

UT ER	ACTIVITY NAME	I	J	CRITICAL PATH	DURATION	EARLIEST START	LATEST START	EARLIEST FINISH	LATEST FINISH	TOTAL FLOAT	FREE FLOAT	INDEP FLOAT	ACTIVITY COST
1	BLDG EXCAVATION	1	2	YES	2.	0.	0.	2.	2.	0.	0.	0.	500.
2	BLDG EXCAVATION	2	3	NO	2.	2.	3.	4.	5.	1.	1.	1.	500.
3	FORM POUR-STRIP FN	2	3	YES	3.	2.	2.	5.	5.	0.	0.	0.	1250.
4	FORM POUR-STRIP FN	3	4	YES	3.	5.	5.	8.	8.	0.	0.	0.	1250.
5	BACKFULL FOUNDT	3	5	NO	3.	5.	6.	8.	9.	1.	0.	0.	450.
6	DUMMY ACTIVITY	4	6	YES	2.	8.	8.	10.	10.	0.	0.	0.	2100.
7	DUMMY ACTIVITY	5	6	NO	1.	8.	9.	9.	10.	1.	1.	0.	150.
8	ERECT STEEL	6	7	NO	2.	10.	11.	12.	13.	1.	0.	0.	5000.
9	RAFTERS AND RF DCK	7	8	NO	5.	12.	13.	17.	18.	1.	0.	0.	3750.
0	DUMMY ACTIVITY	8	9	NO	1.	17.	18.	18.	19.	1.	0.	0.	500.
1	RAFTERS-ROOF DECK	8	10	NO	3.	17.	35.	20.	38.	18.	0.	0.	2250.
2	HAVC COMPL SYSTEM	9	61	NO	27.	18.	19.	45.	46.	1.	1.	0.	21500.
3	SHINGLES-ROOF SUB	10	61	NO	8.	20.	38.	28.	46.	18.	18.	0.	6000.
4	UNDERSLAB CONDUIT	6	11	YES	9.	10.	10.	19.	19.	0.	0.	0.	4000.
5	UNDERSLAB PLUMBING	6	11	YES	9.	10.	10.	19.	19.	0.	0.	0.	4000.
6	DUMMY ACTIVITY	11	12	YES	1.	19.	19.	20.	20.	0.	0.	0.	4000.
7	CERAMIC-QUARRY TIL	12	14	NO	5.	20.	36.	25.	41.	16.	0.	0.	750.
8	INTERIOR PARTITION	12	13	NO	5.	20.	25.	25.	30.	5.	0.	0.	1000.
9	INTERIOT PLUMBING	12	16	YES	10.	20.	20.	30.	30.	0.	0.	0.	3900.
0	INTERIOR ELECTRIC	12	17	YES	10.	20.	20.	30.	30.	0.	0.	0.	7500.
1	CERAMIC QUARRY TIL	14	61	NO	5.	25.	41.	30.	46.	16.	16.	0.	750.
2	PAINTING	13	61	NO	15.	25.	31.	40.	46.	6.	6.	1.	3000.
3	FINISH CARPENTRY	13	18	NO	5.	25.	35.	30.	40.	10.	10.	5.	1333.
4	SHEETROCK-TAPING	13	15	NO	5.	25.	31.	30.	36.	6.	0.	0.	2000.
5	INTERIOR PARTITION	13	18	NO	5.	25.	35.	30.	40.	10.	10.	5.	1000.
6	INTERIOR PLUMBING	16	18	YES	10.	30.	30.	40.	40.	0.	0.	0.	3900.
7	INTERIOR EQUIP COT	17	18	YES	10.	30.	30.	40.	40.	0.	0.	0.	10000.
8	FINISH CARPENTRY	18	61	YES	6.	40.	40.	46.	46.	0.	0.	0.	2907.
9	ACOUSTIC TILE	15	61	NO	10.	30.	36.	40.	46.	6.	6.	0.	3600.
0	RESILIENT TILE	15	61	NO	5.	30.	41.	35.	46.	11.	11.	5.	1500.
1	DUMMY ACTIVITY	4	5	NO	0.	8.	9.	8.	9.	1.	0.	0.	-0.
2	DUMMY ACTIVITY	13	14	NO	0.	25.	41.	25.	41.	16.	0.	0.	0.
3	DUMMY ACTIVITY	13	16	NO	0.	25.	30.	25.	30.	5.	5.	0.	0.
4	DUMMY ACTIVITY	13	17	NO	0.	25.	30.	25.	30.	5.	5.	0.	0.
5	DUMMY ACTIVITY	15	18	NO	0.	30.	40.	30.	40.	10.	10.	4.	0.

EDULE 4 PROJECT DURATION = 45. TOTAL VARIABLE COST = 100440.
EXAMPLE TO DEMONSTRATE A METHOD FOR CONSTR. PLANNING AND SCHEDULING

UT ER	ACTIVITY NAME	I	J	CRITICAL PATH	DURATION	EARLIEST START	LATEST START	EARLIEST FINISH	LATEST FINISH	TOTAL FLOAT	FREE FLOAT	INDEP FLOAT	ACTIVITY COST
1	BLDG EXCAVATION	1	2	YES	1.	0.	0.	1.	1.	0.	0.	0.	600.
2	BLDG EXCAVATION	2	3	NO	2.	1.	2.	3.	4.	1.	1.	1.	500.
3	FORM POUR-STRIP FN	2	3	YES	3.	1.	1.	4.	4.	0.	0.	0.	1250.
4	FORM POUR-STRIP FN	3	4	YES	3.	4.	4.	7.	7.	0.	0.	0.	1250.
5	BACKFULL FOUNDT	3	5	NO	3.	4.	5.	7.	8.	1.	0.	0.	450.
6	DUMMY ACTIVITY	4	6	YES	2.	7.	7.	9.	9.	0.	0.	0.	2100.
7	DUMMY ACTIVITY	5	6	NO	1.	7.	8.	8.	9.	1.	1.	0.	150.
8	ERECT STEEL	6	7	NO	2.	9.	10.	11.	12.	1.	0.	0.	5000.
9	RAFTERS AND RF DCK	7	8	NO	5.	11.	12.	16.	17.	1.	0.	0.	3750.
0	DUMMY ACTIVITY	8	9	NO	1.	16.	17.	17.	18.	1.	0.	0.	500.
1	RAFTERS-ROOF DECK	8	10	NO	3.	16.	34.	19.	37.	18.	0.	0.	2250.
2	HAVC COMPL SYSTEM	9	61	NO	27.	17.	18.	44.	45.	1.	1.	0.	21500.
3	SHINGLES-ROOF SUB	10	61	NO	8.	19.	37.	27.	45.	18.	18.	0.	6000.
4	UNDERSLAB CONDUIT	6	11	YES	9.	9.	9.	18.	18.	0.	0.	0.	4000.
5	UNDERSLAB PLUMBING	6	11	YES	9.	9.	9.	18.	18.	0.	0.	0.	4000.
6	DUMMY ACTIVITY	11	12	YES	1.	18.	18.	19.	19.	0.	0.	0.	4000.
7	CERAMIC-QUARRY TIL	12	14	NO	5.	19.	35.	24.	40.	16.	0.	0.	750.
8	INTERIOR PARTITION	12	13	NO	5.	19.	24.	24.	29.	5.	0.	0.	1000.
9	INTERIOT PLUMBING	12	16	YES	10.	19.	19.	29.	29.	0.	0.	0.	3900.
0	INTERIOR ELECTRIC	12	17	YES	10.	19.	19.	29.	29.	0.	0.	0.	7500.
1	CERAMIC QUARRY TIL	14	61	NO	5.	24.	40.	29.	45.	16.	16.	0.	750.
2	PAINTING	13	61	NO	15.	24.	30.	39.	45.	6.	6.	1.	3000.
3	FINISH CARPENTRY	13	18	NO	5.	24.	34.	29.	39.	10.	10.	5.	1333.
4	SHEETROCK-TAPING	13	15	NO	5.	24.	30.	29.	35.	6.	0.	0.	2000.
5	INTERIOR PARTITION	13	18	NO	5.	24.	34.	29.	39.	10.	10.	5.	1000.
6	INTERIOR PLUMBING	16	18	YES	10.	29.	29.	39.	39.	0.	0.	0.	3900.
7	INTERIOR EQUIP CCT	17	18	YES	10.	29.	29.	39.	39.	0.	0.	0.	10000.
8	FINISH CARPENTRY	18	61	YES	6.	39.	39.	45.	45.	0.	0.	0.	2907.
9	ACOUSTIC TILE	15	61	NO	10.	29.	35.	39.	45.	6.	6.	0.	3600.
0	RESILIENT TILE	15	61	NO	5.	29.	40.	34.	45.	11.	11.	5.	1500.
1	DUMMY ACTIVITY	4	5	NO	0.	7.	8.	7.	8.	1.	0.	0.	-0.
2	DUMMY ACTIVITY	13	14	NO	0.	24.	40.	24.	40.	16.	0.	0.	0.
3	DUMMY ACTIVITY	13	16	NO	0.	24.	29.	24.	29.	5.	5.	0.	0.
4	DUMMY ACTIVITY	13	17	NO	0.	24.	29.	24.	29.	5.	5.	0.	0.
5	DUMMY ACTIVITY	15	18	NO	0.	29.	39.	29.	39.	10.	10.	4.	0.

EDULE 5 PROJECT DURATION = 43. TOTAL VARIABLE COST = 100740.
EXAMPLE TO DEMONSTRATE A METHOD FOR CONSTR. PLANNING AND SCHEDULING

UT ER	ACTIVITY NAME	I	J	CRITICAL PATH	DURATION	EARLIEST START	LATEST START	EARLIEST FINISH	LATEST FINISH	TOTAL FLOAT	FREE FLOAT	INDEP FLOAT	ACTIVITY COST
1	BLDG EXCAVATION	1	2	YES	1.	0.	0.	1.	1.	0.	0.	0.	600.
2	BLDG EXCAVATION	2	3	YES	2.	1.	1.	3.	3.	0.	0.	0.	500.
3	FORM POUR-STRIP FN	2	3	YES	2.	1.	1.	3.	3.	0.	0.	0.	1400.
4	FORM POUR-STRIP FN	3	4	YES	2.	3.	3.	5.	5.	0.	0.	0.	1400.
5	BACKFULL FOUNDT	3	5	YES	3.	3.	3.	6.	6.	0.	0.	0.	450.
6	DUMMY ACTIVITY	4	6	YES	2.	5.	5.	7.	7.	0.	0.	0.	2100.
7	DUMMY ACTIVITY	5	6	YES	1.	6.	6.	7.	7.	0.	0.	0.	150.
8	ERECT STEEL	6	7	NO	2.	7.	8.	9.	10.	1.	0.	0.	5000.
9	RAFTERS AND RF DCK	7	8	NO	5.	9.	10.	14.	15.	1.	0.	0.	3750.
0	DUMMY ACTIVITY	8	9	NO	1.	14.	15.	15.	16.	1.	0.	0.	500.
1	RAFTERS-ROOF DECK	8	10	NO	3.	14.	32.	17.	35.	18.	0.	0.	2250.
2	HAVC COMPL SYSTEM	9	61	NO	27.	15.	16.	42.	43.	1.	1.	0.	21500.
3	SHINGLES-ROOF SUB	10	61	NO	8.	17.	35.	25.	43.	18.	18.	0.	6000.
4	UNDERSLAB CONDUIT	6	11	YES	9.	7.	7.	16.	16.	0.	0.	0.	4000.
5	UNDERSLAB PLUMBING	6	11	YES	9.	7.	7.	16.	16.	0.	0.	0.	4000.
6	DUMMY ACTIVITY	11	12	YES	1.	16.	16.	17.	17.	0.	0.	0.	4000.
7	CERAMIC-QUARRY TIL	12	14	NO	5.	17.	33.	22.	38.	16.	0.	0.	750.
8	INTERIOR PARTITION	12	13	NO	5.	17.	22.	22.	27.	5.	0.	0.	1000.
9	INTERIOT PLUMBING	12	16	YES	10.	17.	17.	27.	27.	0.	0.	0.	3900.
0	INTERIOR ELECTRIC	12	17	YES	10.	17.	17.	27.	27.	0.	0.	0.	7500.
1	CERAMIC QUARRY TIL	14	61	NO	5.	22.	38.	27.	43.	16.	16.	0.	750.
2	PAINTING	13	61	NO	15.	22.	28.	37.	43.	6.	6.	1.	3000.
3	FINISH CARPENTRY	13	18	NO	5.	22.	32.	27.	37.	10.	10.	5.	1333.
4	SHEETROCK-TAPING	13	15	NO	5.	22.	28.	27.	33.	6.	0.	0.	2000.
5	INTERIOR PARTITION	13	18	NO	5.	22.	32.	27.	37.	10.	10.	5.	1000.
6	INTERIOR PLUMBING	16	18	YES	10.	27.	27.	37.	37.	0.	0.	0.	3900.
7	INTERIOR EQUIP CCT	17	18	YES	10.	27.	27.	37.	37.	0.	0.	0.	10000.
8	FINISH CARPENTRY	18	61	YES	6.	37.	37.	43.	43.	0.	0.	0.	2907.
9	ACOUSTIC TILE	15	61	NO	10.	27.	33.	37.	43.	6.	6.	0.	3600.
0	RESILIENT TILE	15	61	NO	5.	27.	38.	32.	43.	11.	11.	5.	1500.
1	DUMMY ACTIVITY	4	5	NO	0.	5.	6.	5.	6.	1.	1.	1.	-0.
2	DUMMY ACTIVITY	13	14	NO	0.	22.	38.	22.	38.	16.	0.	0.	0.
3	DUMMY ACTIVITY	13	16	NO	0.	22.	27.	22.	27.	5.	5.	0.	0.
4	DUMMY ACTIVITY	13	17	NO	0.	22.	27.	22.	27.	5.	5.	0.	0.
5	DUMMY ACTIVITY	15	18	NO	0.	27.	37.	27.	37.	10.	10.	4.	0.

HEDULE 6 PROJECT DURATION = 42. TCTAL VARIABLE COST = 1CC990.
EXAMPLE TO DEMONSTRATE A METHOD FOR CONSTR. PLANNING AND SCHEDULING

PUT DER	ACTIVITY NAME	I	J	CRITICAL PATH	DURATION	EARLIEST START	LATEST START	EARLIEST FINISH	LATEST FINISH	TOTAL FLOAT	FREE FLOAT	INDEP FLOAT	ACTIVITY COST
1	BLDG EXCAVATION	1	2	YES	1.	0.	0.	1.	1.	0.	0.	0.	600.
2	BLDG EXCAVATION	2	3	YES	2.	1.	1.	3.	3.	0.	0.	0.	500.
3	FORM POUR-STRIP FN	2	3	YES	2.	1.	1.	3.	3.	0.	0.	0.	1400.
4	FORM POUR-STRIP FN	3	4	YES	2.	3.	3.	5.	5.	0.	0.	0.	1400.
5	BACKFULL FOUNDT	3	5	YES	3.	3.	3.	6.	6.	0.	0.	0.	450.
6	DUMMY ACTIVITY	4	6	YES	2.	5.	5.	7.	7.	0.	0.	0.	2100.
7	DUMMY ACTIVITY	5	6	YES	1.	6.	6.	7.	7.	0.	0.	0.	150.
8	ERECT STEEL	6	7	YES	2.	7.	7.	9.	9.	0.	0.	0.	5000.
9	RAFTERS AND RF DCK	7	8	YES	5.	9.	9.	14.	14.	0.	0.	0.	3750.
10	DUMMY ACTIVITY	8	9	YES	1.	14.	14.	15.	15.	0.	0.	0.	500.
11	RAFTERS-ROOF DECK	8	10	NO	3.	14.	31.	17.	34.	17.	0.	0.	2250.
12	HAVC COMPL SYSTEM	9	61	YES	27.	15.	15.	42.	42.	0.	0.	0.	21500.
13	SHINGLES-ROOF SUB	10	61	NO	8.	17.	34.	25.	42.	17.	17.	0.	6000.
14	UNDERSLAB CONDUIT	6	11	YES	8.	7.	7.	15.	15.	0.	0.	0.	4150.
15	UNDERSLAB PLUMBING	6	11	YES	8.	7.	7.	15.	15.	0.	0.	0.	4100.
16	DUMMY ACTIVITY	11	12	YES	1.	15.	15.	16.	16.	0.	0.	0.	4000.
17	CERAMIC-QUARRY TIL	12	14	NO	5.	16.	32.	21.	37.	16.	0.	0.	750.
18	INTERIOR PARTITION	12	13	NO	5.	16.	21.	21.	26.	5.	0.	0.	1000.
19	INTERIOT PLUMBING	12	16	YES	10.	16.	16.	26.	26.	0.	0.	0.	3900.
20	INTERIOR ELECTRIC	12	17	YES	10.	16.	16.	26.	26.	0.	0.	0.	7500.
21	CERAMIC QUARRY TIL	14	61	NO	5.	21.	37.	26.	42.	16.	16.	0.	750.
22	PAINTING	13	61	NO	15.	21.	27.	36.	42.	6.	6.	1.	3000.
23	FINISH CARPENTRY	13	18	NO	5.	21.	31.	26.	36.	10.	10.	5.	1333.
24	SHEETROCK-TAPING	13	15	NO	5.	21.	27.	26.	32.	6.	0.	0.	2000.
25	INTERIOR PARTITION	13	18	NO	5.	21.	31.	26.	36.	10.	10.	5.	1000.
26	INTERIOR PLUMBING	16	18	YES	10.	26.	26.	36.	36.	0.	0.	0.	3900.
27	INTERIOR EQUIP CCT	17	18	YES	10.	26.	26.	36.	36.	0.	0.	0.	10000.
28	FINISH CARPENTRY	18	61	YES	6.	36.	36.	42.	42.	0.	0.	0.	2907.
29	ACOUSTIC TILE	15	61	NO	10.	26.	32.	36.	42.	6.	6.	0.	3600.
30	RESILIENT TILE	15	61	NO	5.	26.	37.	31.	42.	11.	11.	5.	1500.
31	DUMMY ACTIVITY	4	5	NO	0.	5.	6.	5.	6.	1.	1.	1.	-0.
32	DUMMY ACTIVITY	13	14	NO	0.	21.	37.	21.	37.	16.	0.	0.	0.
33	DUMMY ACTIVITY	13	16	NO	0.	21.	26.	21.	26.	5.	5.	0.	0.
34	DUMMY ACTIVITY	13	17	NO	0.	21.	26.	21.	26.	5.	5.	0.	0.
35	DUMMY ACTIVITY	15	18	NO	0.	26.	36.	26.	36.	10.	10.	4.	0.

EDULE 7 PROJECT DURATION = 41. TOTAL VARIABLE COST = 101311.
EXAMPLE TO DEMONSTRATE A METHOD FOR CONSTR. PLANNING AND SCHEDULING

UT ER ACTIVITY NAME	I	J	CRITICAL PATH	DURATION	EARLIEST START	LATEST START	EARLIEST FINISH	LATEST FINISH	TOTAL FLOAT	FREE FLOAT	INDEP FLOAT	ACTIVITY COST
1BLDG EXCAVATION	1	2	YES	1.	0.	0.	1.	1.	0.	0.	0.	600.
2BLDG EXCAVATION	2	3	YES	2.	1.	1.	3.	3.	0.	0.	0.	500.
3FORM POUR-STRIP FN	2	3	YES	2.	1.	1.	3.	3.	0.	0.	0.	1400.
4FORM POUR-STRIP FN	3	4	YES	2.	3.	3.	5.	5.	0.	0.	0.	1400.
5BACKFULL FOUNDT	3	5	YES	3.	3.	3.	6.	6.	0.	0.	0.	450.
6DUMMY ACTIVITY	4	6	YES	2.	5.	5.	7.	7.	0.	0.	0.	2100.
7DUMMY ACTIVITY	5	6	YES	1.	6.	6.	7.	7.	0.	0.	0.	150.
8ERECT STEEL	6	7	YES	2.	7.	7.	9.	9.	0.	0.	0.	5000.
9RAFTERS AND RF DCK	7	8	YES	5.	9.	9.	14.	14.	0.	0.	0.	3750.
0DUMMY ACTIVITY	8	9	YES	1.	14.	14.	15.	15.	0.	0.	0.	500.
1RAFTERS-ROOF DECK	8	10	NO	3.	14.	30.	17.	33.	16.	0.	0.	2250.
2HAVC COMPL SYSTEM	9	61	YES	26.	15.	15.	41.	41.	0.	0.	0.	21571.
3SHINGLES-ROOF SUB	10	61	NO	8.	17.	33.	25.	41.	16.	16.	0.	6000.
4UNDERSLAB CONDUIT	6	11	YES	7.	7.	7.	14.	14.	0.	0.	0.	4300.
5UNDERSLAB PLUMBING	6	11	YES	7.	7.	7.	14.	14.	0.	0.	0.	4200.
6DUMMY ACTIVITY	11	12	YES	1.	14.	14.	15.	15.	0.	0.	0.	4000.
7CERAMIC-QUARRY TIL	12	14	NO	5.	15.	31.	20.	36.	16.	0.	0.	750.
8INTERIOR PARTITION	12	13	NO	5.	15.	20.	20.	25.	5.	0.	0.	1000.
9INTERIOT PLUMBING	12	16	YES	10.	15.	15.	25.	25.	0.	0.	0.	3900.
0INTERIOR ELECTRIC	12	17	YES	10.	15.	15.	25.	25.	0.	0.	0.	7500.
1CERAMIC QUARRY TIL	14	61	NO	5.	20.	36.	25.	41.	16.	16.	0.	750.
2PAINTING	13	61	NO	15.	20.	26.	35.	41.	6.	6.	1.	3000.
3FINISH CARPENTRY	13	18	NO	5.	20.	30.	25.	35.	10.	10.	5.	1333.
4SHEETROCK-TAPING	13	15	NO	5.	20.	26.	25.	31.	6.	0.	0.	2000.
5INTERIOR PARTITION	13	18	NO	5.	20.	30.	25.	35.	10.	10.	5.	1000.
6INTERIOR PLUMBING	16	18	YES	10.	25.	25.	35.	35.	0.	0.	0.	3900.
7INTERIOR EQUIP CCT	17	18	YES	10.	25.	25.	35.	35.	0.	0.	0.	10000.
8FINISH CARPENTRY	18	61	YES	6.	35.	35.	41.	41.	0.	0.	0.	2907.
9ACOUSTIC TILE	15	61	NO	10.	25.	31.	35.	41.	6.	6.	0.	3600.
0RESILIENT TILE	15	61	NO	5.	25.	36.	30.	41.	11.	11.	5.	1500.
1DUMMY ACTIVITY	4	5	NO	0.	5.	6.	5.	6.	1.	1.	1.	-0.
2DUMMY ACTIVITY	13	14	NO	0.	20.	36.	20.	36.	16.	0.	0.	0.
3DUMMY ACTIVITY	13	16	NO	0.	20.	25.	20.	25.	5.	5.	0.	0.
4DUMMY ACTIVITY	13	17	NO	0.	20.	25.	20.	25.	5.	5.	0.	0.
5DUMMY ACTIVITY	15	18	NO	0.	25.	35.	25.	35.	10.	10.	4.	0.

HEDULE 8 PROJECT DURATION = 38. TOTAL VARIABLE COST = 102376.
EXAMPLE TO DEMONSTRATE A METHOD FOR CONSTR. PLANNING AND SCHEDULING

PUT DER	ACTIVITY NAME	I	J	CRITICAL PATH	DURATION	EARLIEST START	LATEST START	EARLIEST FINISH	LATEST FINISH	TOTAL FLOAT	FREE FLOAT	INDEP FLOAT	ACTIVITY COST
1	BLDG EXCAVATION	1	2	YES	1.	0.	0.	1.	1.	0.	0.	0.	600.
2	BLDG EXCAVATION	2	3	YES	2.	1.	1.	3.	3.	0.	0.	0.	500.
3	FORM POUR-STRIP FN	2	3	YES	2.	1.	1.	3.	3.	0.	0.	0.	1400.
4	FORM POUR-STRIP FN	3	4	YES	2.	3.	3.	5.	5.	0.	0.	0.	1400.
5	BACKFULL FOUNDT	3	5	YES	3.	3.	3.	6.	6.	0.	0.	0.	450.
6	DUMMY ACTIVITY	4	6	YES	2.	5.	5.	7.	7.	0.	0.	0.	2100.
7	DUMMY ACTIVITY	5	6	YES	1.	6.	6.	7.	7.	0.	0.	0.	150.
8	ERECT STEEL	6	7	YES	2.	7.	7.	9.	9.	0.	0.	0.	5000.
9	RAFTERS AND RF DCK	7	8	YES	5.	9.	9.	14.	14.	0.	0.	0.	3750.
10	DUMMY ACTIVITY	8	9	YES	1.	14.	14.	15.	15.	0.	0.	0.	500.
11	RAFTERS-ROOF DECK	8	10	NO	3.	14.	27.	17.	30.	13.	0.	0.	2250.
12	HAVC COMPL SYSTEM	9	61	YES	23.	15.	15.	38.	38.	0.	0.	0.	21786.
13	SHINGLES-ROOF SUB	10	61	NO	8.	17.	30.	25.	38.	13.	13.	0.	6000.
14	UNDERSLAB CONDUIT	6	11	YES	7.	7.	7.	14.	14.	0.	0.	0.	4300.
15	UNDERSLAB PLUMBING	6	11	YES	7.	7.	7.	14.	14.	0.	0.	0.	4200.
16	DUMMY ACTIVITY	11	12	YES	1.	14.	14.	15.	15.	0.	0.	0.	4000.
17	CERAMIC-QUARRY TIL	12	14	NO	5.	15.	28.	20.	33.	13.	0.	0.	750.
18	INTERIOR PARTITION	12	13	NO	5.	15.	18.	20.	23.	3.	0.	0.	1000.
19	INTERIOT PLUMBING	12	16	YES	10.	15.	15.	25.	25.	0.	0.	0.	3900.
20	INTERIOR ELECTRIC	12	17	YES	10.	15.	15.	25.	25.	0.	0.	0.	7500.
21	CERAMIC QUARRY TIL	14	61	NO	5.	20.	33.	25.	38.	13.	13.	0.	750.
22	PAINTING	13	61	NO	15.	20.	23.	35.	38.	3.	3.	0.	3000.
23	FINISH CARPENTRY	13	18	NO	5.	20.	27.	25.	32.	7.	7.	4.	1333.
24	SHEETROCK-TAPING	13	15	NO	5.	20.	23.	25.	28.	3.	0.	0.	2000.
25	INTERIOR PARTITION	13	18	NO	5.	20.	27.	25.	32.	7.	7.	4.	1000.
26	INTERIOR PLUMBING	16	18	YES	7.	25.	25.	32.	32.	0.	0.	0.	4150.
27	INTERIOR EQUIP CCT	17	18	YES	7.	25.	25.	32.	32.	0.	0.	0.	10600.
28	FINISH CARPENTRY	18	61	YES	6.	32.	32.	38.	38.	0.	0.	0.	2907.
29	ACOUSTIC TILE	15	61	NO	10.	25.	28.	35.	38.	3.	3.	0.	3600.
30	RESILIENT TILE	15	61	NO	5.	25.	33.	30.	38.	8.	8.	5.	1500.
31	DUMMY ACTIVITY	4	5	NO	0.	5.	6.	5.	6.	1.	1.	1.	-0.
32	DUMMY ACTIVITY	13	14	NO	0.	20.	33.	20.	33.	13.	0.	0.	0.
33	DUMMY ACTIVITY	13	16	NO	0.	20.	25.	20.	25.	5.	5.	2.	0.
34	DUMMY ACTIVITY	13	17	NO	0.	20.	25.	20.	25.	5.	5.	2.	0.
35	DUMMY ACTIVITY	15	18	NO	0.	25.	32.	25.	32.	7.	7.	4.	0.

EDULE 9 PROJECT DURATION = 38. TOTAL VARIABLE COST = 102376.
EXAMPLE TO DEMONSTRATE A METHOD FOR CONSTR. PLANNING AND SCHEDULING

UT ER	ACTIVITY NAME	I	J	CRITICAL PATH	DURATION	EARLIEST START	LATEST START	EARLIEST FINISH	LATEST FINISH	TOTAL FLOAT	FREE FLOAT	INDEP FLOAT	ACTIVITY COST
1	BLDG EXCAVATION	1	2	YES	1.	0.	0.	1.	1.	0.	0.	0.	600.
2	BLDG EXCAVATION	2	3	YES	2.	1.	1.	3.	3.	0.	0.	0.	500.
3	FORM POUR-STRIP FN	2	3	YES	2.	1.	1.	3.	3.	0.	0.	0.	1400.
4	FORM POUR-STRIP FN	3	4	YES	2.	3.	3.	5.	5.	0.	0.	0.	1400.
5	BACKFULL FOUNDT	3	5	YES	3.	3.	3.	6.	6.	0.	0.	0.	450.
6	DUMMY ACTIVITY	4	6	YES	2.	5.	5.	7.	7.	0.	0.	0.	2100.
7	DUMMY ACTIVITY	5	6	YES	1.	6.	6.	7.	7.	0.	0.	0.	150.
8	ERECT STEEL	6	7	YES	2.	7.	7.	9.	9.	0.	0.	0.	5000.
9	RAFTERS AND RF DCK	7	8	YES	5.	9.	9.	14.	14.	0.	0.	0.	3750.
10	DUMMY ACTIVITY	8	9	YES	1.	14.	14.	15.	15.	0.	0.	0.	500.
11	RAFTERS-ROOF DECK	8	10	NO	3.	14.	27.	17.	30.	13.	0.	0.	2250.
12	HAVC COMPL SYSTEM	9	61	YES	23.	15.	15.	38.	38.	0.	0.	0.	21786.
13	SHINGLES-ROOF SUB	10	61	NO	8.	17.	30.	25.	38.	13.	13.	0.	6000.
14	UNDERSLAB CONDUIT	6	11	YES	7.	7.	7.	14.	14.	0.	0.	0.	4300.
15	UNDERSLAB PLUMBING	6	11	YES	7.	7.	7.	14.	14.	0.	0.	0.	4200.
16	DUMMY ACTIVITY	11	12	YES	1.	14.	14.	15.	15.	0.	0.	0.	4000.
17	CERAMIC-QUARRY TIL	12	14	NO	5.	15.	28.	20.	33.	13.	0.	0.	750.
18	INTERIOR PARTITION	12	13	NO	5.	15.	18.	20.	23.	3.	0.	0.	1000.
19	INTERIOT PLUMBING	12	16	YES	10.	15.	15.	25.	25.	0.	0.	0.	3900.
20	INTERIOR ELECTRIC	12	17	YES	10.	15.	15.	25.	25.	0.	0.	0.	7500.
21	CERAMIC QUARRY TIL	14	61	NO	5.	20.	33.	25.	38.	13.	13.	0.	750.
22	PAINTING	13	61	NO	15.	20.	23.	35.	38.	3.	3.	0.	3000.
23	FINISH CARPENTRY	13	18	NO	5.	20.	27.	25.	32.	7.	7.	4.	1333.
24	SHEETROCK-TAPING	13	15	NO	5.	20.	23.	25.	28.	3.	0.	0.	2000.
25	INTERIOR PARTITION	13	18	NO	5.	20.	27.	25.	32.	7.	7.	4.	1000.
26	INTERIOR PLUMBING	16	18	YES	7.	25.	25.	32.	32.	0.	0.	0.	4150.
27	INTERIOR EQUIP COT	17	18	YES	7.	25.	25.	32.	32.	0.	0.	0.	10600.
28	FINISH CARPENTRY	18	61	YES	6.	32.	32.	38.	38.	0.	0.	0.	2907.
29	ACOUSTIC TILE	15	61	NO	10.	25.	28.	35.	38.	3.	3.	0.	3600.
30	RESILIENT TILE	15	61	NO	5.	25.	33.	30.	38.	8.	8.	5.	1500.
31	DUMMY ACTIVITY	4	5	NO	0.	5.	6.	5.	6.	1.	1.	1.	-0.
32	DUMMY ACTIVITY	13	14	NO	0.	20.	33.	20.	33.	13.	0.	0.	0.
33	DUMMY ACTIVITY	13	16	NO	0.	20.	25.	20.	25.	5.	5.	2.	0.
34	DUMMY ACTIVITY	13	17	NO	0.	20.	25.	20.	25.	5.	5.	2.	0.
35	DUMMY ACTIVITY	15	18	NO	0.	25.	32.	25.	32.	7.	7.	4.	0.

HEDULE 10 PROJECT DURATION = 36. TOTAL VARIABLE COST = 103269.
EXAMPLE TO DEMONSTRATE A METHOD FOR CONSTR. PLANNING AND SCHEDULING

PUT DER	ACTIVITY NAME	I	J	CRITICAL PATH	DURATION	EARLIEST START	LATEST START	EARLIEST FINISH	LATEST FINISH	TOTAL FLOAT	FREE FLOAT	INDEP FLOAT	ACTIVITY COST
1	BLDG EXCAVATION	1	2	YES	1.	0.	0.	1.	1.	0.	0.	0.	600.
2	BLDG EXCAVATION	2	3	YES	2.	1.	1.	3.	3.	0.	0.	0.	500.
3	FORM POUR-STRIP FN	2	3	YES	2.	1.	1.	3.	3.	0.	0.	0.	1400.
4	FORM POUR-STRIP FN	3	4	YES	2.	3.	3.	5.	5.	0.	0.	0.	1400.
5	BACKFULL FOUNDT	3	5	YES	3.	3.	3.	6.	6.	0.	0.	0.	450.
6	DUMMY ACTIVITY	4	6	YES	2.	5.	5.	7.	7.	0.	0.	0.	2100.
7	DUMMY ACTIVITY	5	6	YES	1.	6.	6.	7.	7.	0.	0.	0.	150.
8	ERECT STEEL	6	7	YES	2.	7.	7.	9.	9.	0.	0.	0.	5000.
9	RAFTERS AND RF DCK	7	8	YES	5.	9.	9.	14.	14.	0.	0.	0.	3750.
10	DUMMY ACTIVITY	8	9	YES	1.	14.	14.	15.	15.	0.	0.	0.	500.
11	RAFTERS-ROOF DECK	8	10	NO	3.	14.	25.	17.	28.	11.	0.	0.	2250.
12	HAVC COMPL SYSTEM	9	61	YES	21.	15.	15.	36.	36.	0.	0.	0.	21929.
13	SHINGLES-ROOF SUB	10	61	NO	8.	17.	28.	25.	36.	11.	11.	0.	6000.
14	UNDERSLAB CONDUIT	6	11	YES	7.	7.	7.	14.	14.	0.	0.	0.	4300.
15	UNDERSLAB PLUMBING	6	11	YES	7.	7.	7.	14.	14.	0.	0.	0.	4200.
16	DUMMY ACTIVITY	11	12	YES	1.	14.	14.	15.	15.	0.	0.	0.	4000.
17	CERAMIC-QUARRY TIL	12	14	NO	5.	15.	26.	20.	31.	11.	0.	0.	750.
18	INTERIOR PARTITION	12	13	NO	5.	15.	16.	20.	21.	1.	0.	0.	1000.
19	INTERIOT PLUMBING	12	16	YES	8.	15.	15.	23.	23.	0.	0.	0.	4150.
20	INTERIOR ELECTRIC	12	17	YES	8.	15.	15.	23.	23.	0.	0.	0.	8000.
21	CERAMIC QUARRY TIL	14	61	NO	5.	20.	31.	25.	36.	11.	11.	0.	750.
22	PAINTING	13	61	NO	15.	20.	21.	35.	36.	1.	1.	0.	3000.
23	FINISH CARPENTRY	13	18	NO	5.	20.	25.	25.	30.	5.	5.	4.	1333.
24	SHEETROCK-TAPING	13	15	NO	5.	20.	21.	25.	26.	1.	0.	0.	2000.
25	INTERIOR PARTITION	13	18	NO	5.	20.	25.	25.	30.	5.	5.	4.	1000.
26	INTERIOR PLUMBING	16	18	YES	7.	23.	23.	30.	30.	0.	0.	0.	4150.
27	INTERIOR EQUIP COT	17	18	YES	7.	23.	23.	30.	30.	0.	0.	0.	10600.
28	FINISH CARPENTRY	18	61	YES	6.	30.	30.	36.	36.	0.	0.	0.	2907.
29	ACOUSTIC TILE	15	61	NO	10.	25.	26.	35.	36.	1.	1.	0.	3600.
30	RESILIENT TILE	15	61	NO	5.	25.	31.	30.	36.	6.	6.	5.	1500.
31	DUMMY ACTIVITY	4	5	NO	0.	5.	6.	5.	6.	1.	1.	1.	-0.
32	DUMMY ACTIVITY	13	14	NO	0.	20.	31.	20.	31.	11.	0.	0.	0.
33	DUMMY ACTIVITY	13	16	NO	0.	20.	23.	20.	23.	3.	3.	2.	0.
34	DUMMY ACTIVITY	13	17	NO	0.	20.	23.	20.	23.	3.	3.	2.	0.
35	DUMMY ACTIVITY	15	18	NO	0.	25.	30.	25.	30.	5.	5.	4.	0.

SCHEDULE OF PROJECT VARIABLES

SCHEDULE	DURATION	VARIABLE COST	OVERHEAD	TOTAL PROJECT COST
1	50.	100100.	10000.	110100.
2	46.	100340.	9200.	109540.
3	46.	100340.	9200.	109540.
4	45.	100440.	9000.	109440.
5	43.	100740.	8600.	109340.
6	42.	100990.	8400.	109390.
7	41.	101311.	8200.	109511.
8	38.	102376.	7600.	109976.
9	38.	102376.	7600.	109976.
10	36.	103269.	7200.	110469.

EXAMPLE TC DEMONSTRATE A METHOD FOR CONSTR. PLANNING AND SCHEDULING
LYSIS UTILIZING THE ACTIVITY PREFERRED DURATION AS A PARAMETER IN SCHEDULING

JECT DURATION = 43 PRCJECT DIRECT COST = 100740. PROJECT OVERHEAD = 8600. PROJECT TOTAL COST = 109340.

UT ER	ACTIVITY NAME	I	J	CRITICAL PATH	DURATION	EARLIEST START	LATEST START	EARLIEST FINISH	LATEST FINISH	TOTAL FLOAT	FREE FLOAT	INDEP FLOAT	ACTIVITY COST
1	BLDG EXCAVATION	1	2	YES	1.	0.	0.	1.	1.	0.	0.	0.	600.
2	BLDG EXCAVATION	2	3	YES	2.	1.	1.	3.	3.	0.	0.	0.	500.
3	FCRM POUR-STRIP FN	2	3	YES	2.	1.	1.	3.	3.	0.	0.	0.	1400.
4	FCRM POUR-STRIP FN	3	4	YES	2.	3.	3.	5.	5.	0.	0.	0.	1400.
5	BACKFULL FOUNDT	3	5	YES	3.	3.	3.	6.	6.	0.	0.	0.	450.
6	PLACE-GRADE GRAVEL	4	6	YES	2.	5.	5.	7.	7.	0.	0.	0.	2100.
7	BACKFILL FOUNDTN	5	6	YES	1.	6.	6.	7.	7.	0.	0.	0.	150.
8	ERECT STEEL	6	7	NO	2.	7.	8.	9.	10.	1.	0.	0.	5000.
9	RAFTERS AND RF DCK	7	8	NO	5.	9.	10.	14.	15.	1.	0.	0.	3750.
0	SET HVAC EQUIP	8	9	NO	1.	14.	15.	15.	16.	1.	0.	0.	500.
1	RAFTERS-ROOF DECK	8	10	NO	3.	14.	30.	17.	33.	16.	0.	0.	2250.
2	HAVC COMPL SYSTEM	9	61	NO	27.	15.	16.	42.	43.	1.	1.	0.	21500.
3	SHINGLES-ROOF SUB	10	61	NO	10.	17.	33.	27.	43.	16.	16.	0.	6000.
4	UNDERSLAB CONDUIT	6	11	YES	9.	7.	7.	16.	16.	0.	0.	0.	4000.
5	UNDERSLAB PLUMBING	6	11	YES	9.	7.	7.	16.	16.	0.	0.	0.	4000.
6	PCUR SLAB	11	12	YES	1.	16.	16.	17.	17.	0.	0.	0.	4000.
7	CERAMIC-QUARRY TIL	12	14	NO	10.	17.	28.	27.	38.	11.	0.	0.	750.
8	INTERIOR PARTITION	12	13	YES	6.	17.	17.	23.	23.	0.	0.	0.	1000.
9	INTERIOT PLUMBING	12	16	YES	10.	17.	17.	27.	27.	0.	0.	0.	3900.
0	INTERIOR ELECTRIC	12	17	YES	10.	17.	17.	27.	27.	0.	0.	0.	7500.
1	CERAMIC QUARRY TIL	14	61	NO	5.	27.	38.	32.	43.	11.	11.	0.	750.
2	PAINTING	13	61	YES	20.	23.	23.	43.	43.	0.	0.	0.	3000.
3	FINISH CARPENTRY	13	18	NO	8.	23.	29.	31.	37.	6.	6.	6.	1333.
4	SHEETROCK-TAPING	13	15	NO	5.	23.	26.	28.	31.	3.	0.	0.	2000.
5	INTERIOR PARTITION	13	18	NO	5.	23.	32.	28.	37.	9.	9.	9.	1000.
6	INTERIOR PLUMBING	16	18	YES	10.	27.	27.	37.	37.	0.	0.	0.	3900.
7	INTERIOR EQUIP COT	17	18	YES	10.	27.	27.	37.	37.	0.	0.	0.	10000.
8	FINISH CARPENTRY	18	61	YES	6.	37.	37.	43.	43.	0.	0.	0.	2907.
9	ACOUSTIC TILE	15	61	NO	12.	28.	31.	40.	43.	3.	3.	0.	3600.
0	RESILIENT TILE	15	61	NO	10.	28.	33.	38.	43.	5.	5.	2.	1500.
1	DUMMY ACTIVITY	4	5	NO	0.	5.	6.	5.	6.	1.	1.	1.	-0.
2	DUMMY ACTIVITY	13	14	NO	0.	23.	38.	23.	38.	15.	4.	4.	0.
3	DUMMY ACTIVITY	13	16	NO	0.	23.	27.	23.	27.	4.	4.	4.	0.
4	DUMMY ACTIVITY	13	17	NO	0.	23.	27.	23.	27.	4.	4.	4.	0.
5	DUMMY ACTIVITY	15	18	NO	0.	28.	37.	28.	37.	9.	9.	6.	0.

APPENDIX B

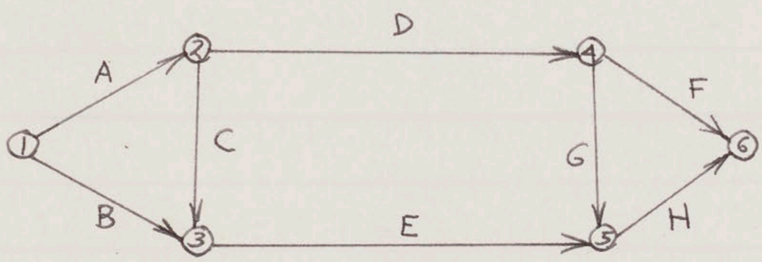
```
C     CRITICAL PATH PROGRAM USING NEW FLOW ALGORITHM
C     SCHEDULING ROUTINE INCLUDED
      DIMENSION I(1000),J(1000),TN(1000),TCR(1000),SLOPE(1000)
      DIMENSION FLUS(1000),FLOW(1000),KFROM(1000),BLET(1000),CSTN(1000)
      DIMENSION CSTCR(1000),BIGG(1000),ACT(1000,4)
1000  FORMAT(2I4)
      READ 1000,KMAX,LMAX
      TFL=0.
      JQAB=KMAX+1
      DO 10 K=1,KMAX
      FLUS(K)=0.
2000  FORMAT(2I4,20X,2F3.0,3X,2F5.0,4A6)
      READ 2000, I(K),J(K),TN(K),TCR(K),CSTN(K),CSTCR(K),
     1(ACT(K,II),II=1,4)
      IF(TN(K))1463,1463,1464
1463  SLOPE(K)=999999.99
      GO TO 10
1464  SLOPE(K)=(CSTCR(K)-CSTN(K))/(TN(K)-TCR(K))
10    CONTINUE
1     DO 12 K=1,LMAX
      BIGG(K)=0.
      FLOW(K)=0.
      BLET(K)=0.
12    KFROM(K)=0.
      J1=1
      FLOW(1)=999999.99
603   DO 201 K=J1,KMAX
      NI=I(K)
      NJ=J(K)
      IF(FLUS(K)-SLOPE(K))5,6,6
6     FLUS(K)=999999.99
5     IF(FLUS(K)-999999.98)301,300,300
300   TEMP=TCR(K)
      GO TO 202
301   TEMP=TN(K)
202   IF(BIGG(NJ)-(BIGG(NI)+TEMP))200,201,201
200   BIGG(NJ)=BIGG(NI)+TEMP
      KFROM(NJ)=K
      QNT=SLOPE(K)-FLUS(K)
      J2=1
      IF(FLOW(NI))100,801,801
801   IF(QNT)101,803,803
803   IF(QNT-FLOW(NI))100,101,101
100   FLOW(NJ)=QNT
      GO TO 201
101   FLOW(NJ)=FLOW(NI)
201   CONTINUE
      DO 30 K=J2,KMAX
      JQA=JQAB-K
      NI=I(JQA)
      NJ=J(JQA)
      IF(NI-1)33,30,33
33    IF(FLUS(JQA))31,30,31
31    TEMP=BIGG(NJ)-TN(JQA)
      IF(TEMP-BIGG(NI))30,30,32
32    BIGG(NI)=TEMP
      FLOW(NI)=FLUS(JQA)
      KFROM(NI)=-JQA
      J2=K
```

```
      DO 600 K=1,KMAX
      IF(I(K)-NI)600,602,602
600   CONTINUE
602   J1=K
      GO TO 603
30    CONTINUE
C     SCHEDULING ROUTINE
       BLET(LMAX)=BIGG(LMAX)
      PRINT 7000
7000  FORMAT(80H ACTIVITY                                          EARLY  LATE   EAR
     1LY  LATE   FREE   TOTAL )
      PRINT  8000
8000  FORMAT(80H                                                   START  START  FIN
     1ISH FINISH FLOAT  FLOAT )
      DO 702 K=1,KMAX
      JQA=JQAB-K
      NI=I(JQA)
      NJ=J(JQA)
      DUR=BIGG(NJ)-BIGG(NI)
      IF(DUR-TN(JQA))701,701,700
700   DUR=TN(JQA)
701   IF(BLET(NI))704,703,704
704   IF(BLET(NI)-(BLET(NJ)-DUR))702,702,703
703   BLET(NI)=BLET(NJ)-DUR
702   CONTINUE
      DO 705 K=1,KMAX
      NI=I(K)
      NJ=J(K)
      DUR=BIGG(NJ)-BIGG(NI)
      IF(DUR-TN(K))813,813,804
804   DUR=TN(K)
813   FES=BIGG(NI)
      FLS=BLET(NJ)-DUR
      FEF=FES+DUR
      FLF=BLET(NJ)
      FFF=BIGG(NJ)-FES-DUR
      FTF=FLF-FES-DUR
705   PRINT 4000,(ACT(K,II),II=1,4),NI,NJ,FES,FLS,FEF,FLF,FFF,FTF
4000  FORMAT(4A6,2I7,6F7.0)
      FLTN=FLOW(LMAX)
      IF(FLOW(LMAX)-999999.98)802,4,4
  802 JQA=LMAX
      TFL=TFL+FLTN
      DO 406 K=1,KMAX
      NJ=KFROM(JQA)
      FISW=1
      IF(NJ)404,404,405
405   IF(FLUS(NJ)-999999.98)402,403,403
404   NJ=-NJ
      FISW=-1
402   FLUS(NJ)=FLUS(NJ)+FLTN*FISW
      IF(FISW)407,403,403
407   JQA=J(NJ)
      GO TO 406
  403 IF(I(NJ)-1)3,3,401
401   JQA=I(NJ)
406   CONTINUE
3     PRINT 6000
6000  FORMAT(32H PROJECT DURATION   PROJECT COST)
```

```
      PRINT 3000,BIGG(LMAX),TFL
3000  FORMAT(2F20.0)
      GO TO 1
4      TFL=999999.99
4444  PRINT 6000
      PRINT 3000,BIGG(LMAX),TFL
      END
```

```
C       INPUT DATA FOR SAMPLE PROBLEM TO TEST PROGRAM WITH NEW FLOW ALGORITHM
*       DATA
    8    6
    1    2                        4. 3.     100. 200. LINK A
    1    3                        8. 6.     300. 400. LINK B
    2    3                        9. 7.     500. 800. LINK C
    2    4                        3. 1.     200. 400. LINK D
    3    5                        7. 6.     150. 350. LINK E
    4    6                        8. 5.     100. 400. LINK F
    4    5                        5. 3.     600. 800. LINK G
    5    6                        7. 5.     100. 300. LINK H
```

```
C       ILLUSTRATION OF SAMPLE PROBLEM NETWORK
```



OUTPUT FOR SAMPLE PROBLEM TO TEST PROGRAM WITH NEW FLOW ALGORITHM

RY POINTS TO SUBROUTINES REQUESTED FROM LIBRARY,
.SETUP (CSHM) (RTN) (SPHM) (FIL)

.65 MINUTES ELAPSED SINCE START OF JOB

EXECUTION

TIVITY			EARLY START	LATE START	EARLY FINISH	LATE FINISH	FREE FLOAT	TOTAL FLOAT
NK A	1	2	0.	0.	4.	4.	0.	0.
NK B	1	3	0.	5.	8.	13.	5.	5.
NK C	2	3	4.	4.	13.	13.	0.	0.
NK D	2	4	4.	12.	7.	15.	0.	8.
NK E	3	5	13.	13.	20.	20.	0.	0.
NK F	4	6	7.	19.	15.	27.	12.	12.
NK G	4	5	7.	15.	12.	20.	8.	8.
NK H	5	6	20.	20.	27.	27.	0.	0.

OJECT DURATION PROJECT COST
27. 100.

TIVITY			EARLY START	LATE START	EARLY FINISH	LATE FINISH	FREE FLOAT	TOTAL FLOAT
NK A	1	2	0.	0.	3.	3.	0.	0.
NK B	1	3	0.	4.	8.	12.	4.	4.
NK C	2	3	3.	3.	12.	12.	0.	0.
NK D	2	4	3.	11.	6.	14.	0.	8.
NK E	3	5	12.	12.	19.	19.	0.	0.
NK F	4	6	6.	16.	14.	24.	10.	10.
NK G	4	5	6.	14.	11.	19.	8.	8.
NK H	5	6	19.	19.	24.	24.	0.	0.

OJECT DURATION PROJECT COST
24. 150.

TIVITY			EARLY START	LATE START	EARLY FINISH	LATE FINISH	FREE FLOAT	TOTAL FLOAT
NK A	1	2	0.	0.	3.	3.	0.	0.
NK B	1	3	0.	2.	8.	10.	2.	2.
NK C	2	3	3.	3.	10.	10.	0.	0.
NK D	2	4	3.	9.	6.	12.	0.	6.
NK E	3	5	10.	10.	17.	17.	0.	0.
NK F	4	6	6.	14.	14.	22.	8.	8.
NK G	4	5	6.	12.	11.	17.	6.	6.
NK H	5	6	17.	17.	22.	22.	0.	0.

OJECT DURATION PROJECT COST
22. 200.

TIVITY			EARLY START	LATE START	EARLY FINISH	LATE FINISH	FREE FLOAT	TOTAL FLOAT
NK A	1	2	0.	0.	3.	3.	0.	0.
NK B	1	3	0.	2.	8.	10.	2.	2.
NK C	2	3	3.	3.	10.	10.	0.	0.
NK D	2	4	3.	8.	6.	11.	0.	5.
NK E	3	5	10.	10.	16.	16.	0.	0.
NK F	4	6	6.	13.	14.	21.	7.	7.
NK G	4	5	6.	11.	11.	16.	5.	5.
NK H	5	6	16.	16.	21.	21.	0.	0.

OJECT DURATION PROJECT COST
21. 1000000.

IO CKECK LIGHT ON FROM LAST WRITE.

APPENDIX C

```
C     NEW FLOW ALGORITHM TO REPLACE OPERATIONS 8, 9, AND 10 OF CPS 300
C     PREPARED FOR DIRECT SUBSITUTION INTO THE MAIN PROGRAM
29    DO 43 K=1,LMAX
      BIGG(K)=0.
      FLOW(K)=0.
43    KFROM(K)=0.
      J1=1
      FLOW(1)=30000.0
31    DO 30 K=J1,KMAX
      NI=I(K)
      NJ=J(K)
      IF(FLUS(K)-SLOPE(K))32,33,33
33    FLUS(K)=30000.0
32    IF(FLUS(K)-30000.0)35,36,36
36    TEMP=TCR(K)
      GO TO 37
35     TEMP=TN(K)
37    IF(BIGG(NJ)-(BIGG(NI)+TEMP))50,30,30
50    BIGG(NJ)=BIGG(NI)+TEMP
      KFROM(NJ)=K
      QNT=SLOPE(K)-FLUS(K)
      J2=1
      IF(FLOW(NI))60,61,61
61    IF(QNT)65,62,62
62    IF(QNT-FLOW(NI))60,65,65
60    FLOW(NJ)=QNT
      GO TO 30
65    FLOW(NJ)=FLOW(NI)
30    CONTINUE
70    DO 74 K=J2,KMAX
      JQA=JQAB-K
      NI=I(JQA)
      NJ=J(JQA)
      IF(NI-1)71,74,71
71    IF(FLUS(JQA))72,74,72
72    TEMP=BIGG(NJ)-TN(JQA)
      IF(TEMP-BIGG(NI))74,74,75
75    BIGG(NI)=TEMP
      FLOW(NI)=FLUS(JQA)
      KFROM(NI)=-JQA
      J2=K
      DO 76 K=1,KMAX
      IF(I(K)-NI)76,77,77
76    CONTINUE
77    J1=K
      GO TO 31
74    CONTINUE
100    FLTN=FLOW(LMAX)
      JQA=LMAX
      TFL=TFL+FLTN
      DO 120 K=1,KMAX
      NJ=KFROM(JQA)
      FISW=1
      IF(NJ)45,45,47
47    IF(FLUS(NJ)-30000.0)114,116,116
45    NJ=-NJ
      FISW=-1
114   FLUS(NJ)=FLUS(NJ)+FLTN*FISW
      IF(FISW)117,116,116
```

```
117    JQA=J(NJ)
       GO TO 120
116    IF(I(NJ)-1)126,126,128
128    JQA=I(NJ)
120    CONTINUE
126    PRDR(JJ)=BIGG(LMAX)
```