THE ACYCLIC DIRECTED NON - STOCHASTIC NETWORK

(SCHEDULING AND RESOURCE LEVELLING)

by

G. VISHWANATH

DISSERTATION SUBMITTED
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF ENGINEERING
IN
CIVIL ENGINEERING (STRUCTURAL ENGINEERING)

UNIVERSITY OF ROORKEE
ROORKEE
APRIL — 1975

C E R T I F I C A T E

Certified that the dissertation entitled, "The Acyclic Directed Non-Stochastic Network-Scheduling and Resource Levelling" which is being submitted by Sri G. Vishwanath, in partial fulfilment of the requirements for the award of the Degree of Master of Engineering in Civil Engineering (Structural Engineering) of the University of Roorkee, is a record of the student's own work carried out by him under my s upervisionand guidance. The matter embodied in this dissertation has not been submitted for the award of any other Degree or Diploma.

This is further to certify that the has worked for a period of over five months, for the preparation of this dissertation for the Master of Engineering Degree at this University.

Dated
April, 12, 1975.

(N. PURI)
Lecturer in Civil Engineering
Civil Engineering Department,
University of Roorkee,
Roorkee.

ACKNOWLEDGEMENTS

There are some unforgettable characters in the life of each individual and one such unforgettable personality, in the author's life is his guide and supervisor, Mr. N.Puri, Lecturer in Civil Engineering, University of Roorkee.

This acknowledgement is only a futile attempt at expressing the author's gratitude for all the personal sacrifices that Mr. Puri has made to expedite the work on thi dissertation.

His thorough knowledge of the subject and his complete mastery over the difficult art of computer programmi have never been in doubt, and whatever programming skills the author has developed, is entirely due to him. From the position of a novice in the field of computer programming, the author is now in a position to confidently tackle any programming problem however challenging.

What is indeed touching is the unselfish nature of Mr. Puri's character and his willingness to go out of his way to help another. The author has no words to describe his patience, and can only recall with gratitude the long and strenuous hours spent in discussing and thrashing out the minutest details of the logic of the computer program, and also his threadbare examination of the write-up.

Where necessary his criticism has been merciless, and in times of emotional strain, his advice has been most soothing, His hardworking nature has inspired the author and in keeping pace with him the author has accomplished much more work than he would have otherwise.

In brief, working with him has been a new and a most rewarding experience in life and the author will forever cherish happy memories of these past few months.

The author also gratefully remembers Mr. S. Sankaran, Mr. T.B.Krishnamoorthy, and Mr. V.Govinda, his respec ted seniors in the Structural Sec tion of the Company where the author is employed and without whose cooperation and blessings, an early submission of this thesis would not have been possible. The author acknowledges with thanks the kind interest shown in this thesis by Mr. Narasimhadu, of the Systems Section of the Office, and his readiness to help. The author also gratefully remembers his friend and colleague in the office Mr. S.Shiva Shanker for his encouraging and morale boosting letters during periods of great emotional strain. The author also records his thanks to Mr. A.C.Das(Overseer) who cheerfully agreed to forgo a quiet holiday at home and accompanied the author to the site to collect data for the case study .

The staff of the Computer Centre at SERC,Roorkee also deserve thanks for their cooperation (specially Mr.G. Kishore and Mr. Sharma who went out of their way to help).

Mr. Narang (Draughtsman) and Mr. Mahendra Pal Goel (Stenographer) deserve special thanks for their cooperation in speeding up the submission of this thesis.

The author also gratefully remembers a host of friends who rallied around during an emergency, particularly Mr. S.S.Ramakrishnan, Mr. Balaguruswamy, Mr. A.K. Aggrawal, Mr. A.D.Pande, Mr. N.S.Murthy, Mr. Bala Subramaniam and others.

Last but not the least the author fondly remembers his parents whose help and encouragement during this period can never be adequately acknowledged.

C O N T E N T S

Contents (Contd..)

--

S Y N O P S I S

The relevance of a systems approach to construction management is pointed out.

Network techniques, as a tool in systems analysis, have been mentioned and the acyclic, directed, non stochastic network (the CPM network) has been discussed in detail.

The mathematical model of the scheduling phase of CPM has been presented and followed by a lucid description of a computer program for the scheduling computations.

The importance of the problem of Res ource Levelling has been discussed and followed by a survey of the work already done in this field. An original algorithm for the Resource scheduling problem has been presented, followed by a detailed description of the computer program.

To illustrate the two computer programs, the hutment renovation project currently in progress at the University, has been taken as a case study, and the results of the CPM Scheduling program and the Resource levelling program have been reported.

CHAPTER - I
INTRODUCTION

NEED FOR A SYSTEM APPROACH TO CONSTRUCTION MANAGEMENT

The field of civil Engineering Construction has certain special features which distinguish it from other industries. Firstly, the range of operations and the processes involved is very wide, and includes excavation, pile driving, dam construction, multi-storeyed structures, tunnelling, bridge construction, marine works, water supply, pavements and a host of other works. Each of these requires special construction techniques, equipment and labour skills.

Secondly the worksites of projec ts are always temporary and often remotely situated and full scale production on any site lasts only for a few months or at best a few years. Hence there is a lack of continuity of operations at one site.

Thirdly the local site management rarely has full control of policy and finance and can never be self reliant.

Lastly, the construction personnel generally fall into two broad groups, viz temporary workers and the more or less permanent supervisory staff which shifts from one project to another.

These features in the construction industry make construction management a challenging task. It is essential that the construction organisation at the field is adaptable to the varied conditions from project to project and is flexible enough to control adequately the works being executed under different site conditions. All these characteristics of construction works have to be taken into account by the management while planning, estimating, organising and executing the works so as to achieve the predetermined objec tives of quality, time and cost.

SYSTEMS ANALYSIS

Though the systems approach is not new, its application to construction management is fairly recent. In simple language, the systems concept envisages that any organisation can be conceived of as a 'system', which is made up of 'segments' or 'sub-systems'. Each of these subsystems have direct or indirect mutual relationships and interdependencies and therefore the performance of the system as a whole will depend upon the interactions between subsystems. While each sub-system may have its own goals and objectives they have to be fitted into the overall objectives of the organisation. The function of the manager therefore is to identify and measure the inter-relationships of these components and to integrate them in a manner which enables the organisation to efficiently pursue its overall objectives. The manager has to look upon the organisation under him as an integrated assembly of interacting components designed to perform, jointly a predetermined purpose.

In a systems approach the function of decision making becomes very complex since the alternatives available and the inter-relationships and combinations of different circumstances become too large to be comprehended in a simple manner. The decision making therefore necessitates objective and scientific analysis of such problems by using mathematical models i.e. by reducing real world systems to a symbolic model which depicts the logical inter-relationships of various parameters. Before taking a decision it is possible to experiment on the model to predict the behaviour or outcome of the system itself. The scientific analysis of a system with the help of models is termed as "systems Analysis". Several other terms which are commonly used are "operations research", "cost -benefit analysis". Net work Techniques (like PERT, CPM, GERT etc) also represent system models which can be analysed in a scientific manner so that the results of the actual field operations can be predicted in advance and remedial action taken.

NETWORK TECHNIQUES

This is a very general term, and whole volumes have been written on network techniques alone. Networks are of several types, and each type has its features. Depending on the situation, one network may be more valuable than the other.

NETWORK CLASSIFICATION

Networks have been classified into

1. Acyclic Directed Networks

The PERT and CPM networks belong to this class. These are networks in which all paths are directed and they contain no cycles or loops. Because of the special nature of these networks we can always number the nodes in such a way that all activities lead from a smaller-numbered node to a node of larger number.

2. Cyclic Directed Networks

These are more general networks and they include cycles.

3. Stochastic Networks

These are networks with logical nodes and probabilistic activity realization. The PERT and GERT networks belong to this class.

The commonly used network techniques are-

1. The critical path method (CPM)
2. Project Evaluation and Review Technique (PERT)
3. The Graphical Evaluation and Review Technique(GERT)
4. Line of Balance Technique (LOB)
5. Flowgraphs
6. Decision trees.

A discussion of the various types of networks is beyond the scope of this dissertation.

Among the network techniques listed above, the CPM technique is the one that is mostly used in construction projects. CPM is a project management technique, and the basis is the activity network, or the project network model. It is essentially an Acyclic , Directed, non-stochastic network, and unlike the PERT network which is a stochastic network, it considers the duration estimates over a range of facility or cost levels, andas a result, provides a range of project durations with an associated range of projec t costs. CPM computations establish the absolute minimum cost of attaining any feasible project duration.

Conventional Methods

Before the advent of CPM, and other network techniques, conventional techniques were used in the construction industry, for planning and scheduling. These included -

i. Gantt Charts ii. Milestone Charts

iii. Flow charts.

GANTT Charts

Before World War I, a consultant by the name Henry L. Gantt, designed for the U.S. Army Ordinance

Bureau, a simple chart which could display the schedule as well as make a comparison between the actual performance and the schedule. A project was divided into activities which were shown by horizontal bars on a time scale drawn from the date of start to the scheduled date of completion. These bars were shaded to indicate the portion of work completed . On any date of reviewing the progress the shaded portion of various bars indicated whether a particular activity, was on time, behind schedule or ahead of time.

Limitation of Gantt Charts

1. It does not indicate the interrelationships between various activities.
2. It does not tell us the effect on the completion time as a whole of one activity falling behind schedule and another activity going ahead of schedule.
3. It does not indicate which activities are critical, and which activities have spare time and resources (and how much).
4. In case the project completion time has to be reduced the Gantt chart does not indicate the effect of this on the cost.

Milestone Charts

The Gantt chart, inspite of its limitations remained the most popular and widely adopted method of planning and scheduling since it is simple and easily understood and because of its visual effect. Further refinement was introduced in this method around 1940 by showing important milestones represented by vertical arrows.

Flow Chart

The first attempt to show sequence of activities by connecting events in order of their relationships was made by Gilberts (a contemporary of Gantt) This chart known as flow chart was perhaps the beginning of Network techniques applied to the field of construction. Management, though the first full-fledged network techniques applied to construction were CPM and PERT.

Advent of CPM/PERT

Due to increasing complexities in Industrial projects resulting from technological advances, the conventional methods were found to be inadequate for planning, scheduling and in monitoring project tasks. In large projec-ts the number of activities ran into thousands, and in the absence of a comprehensive and refined method it became difficult to comprehend the inter-relationships of activities and to schedule them in such a manner as to achieve optimum cost. This provided the background for the development of CPM.

Historical Development

The basis for what is presently called CPM was created during the late 1950's in two more or less parallel developments: The Dupont Company's Refinary Renovation Project and the U.S. Navy's Fleet Ballistic Missile project.

Dupont during the late 50's was interested in finding a way to schedule refinery renovation projects in such a way that a minimum amount of time was required to take a refinery out of service and that a minimum amount of money was required to effect this project speed up. They were looking for the most economical way of scheduling a project considering the increasing direct cost of the project speed up and the indirect cost of taking the refinery out of action which cost goes up as the project is shortened. Morgan. E. Walker, along with James Kelly of Remington Rand Company developed a project planning and scheduling technique based on network analysis. This technique they called Critical Path Planning and Scheduling - CPPS.For short. CPPS was applied by the Dupont Company with great success. The company found that this technique required only about half as much effort as other planning and scheduling techniques, they previously had used. To quote actual figures: the method reduced the shut down period from 125 hours to 93 hours. Expediting and improving labour performance on critical path activities further reduced the shut down time to 78 hours, resulting in millions of pounds of production.

Soon it was realised that this method was easily applicable to most construction projects, and it was hailed as a substantial contribution to management methods.

Since then CPM has grown considerably over the years, it found wider and wider areas of application, and it is today accepted as a basic tool in project management.

Though CPM was not developed exclusively for the construction industry, the fact remains that the construction industry has been the most extensive user. Even outside the construction Industry, CPM has been used with much success. The state of North Carolina used CPM to schedule and control the states biannual budget - an entirely administrative task. The city of Washington DC used CPM to coordinate the 'moving ' operation of the staff of their city hospital with the various phases of its renovation. The U.S. coast Guard used CPM to plan and schedule the production of a cutter, and the Navy's Bureau of ships used the technique to assist them in ship overhaul scheduling.

What started as a planning and scheduling technique, soon developed into a versatile project management technique.Today CPM includes cost control, resource scheduling, monitoring and updating of projects and many more features are being added.

CPM - GeneralPrinciples

CPM is a Management technique that enables a project manager to predict when the activities involved in his projec ts can be expected to occur. It is a set of tools that uses actual job conditions in a continuous process of schedule revision and improvement . It is a technique that through this process of improvement and revision enables the contractor to trust the schedule produced and make most effective use of whatever resources and limitations exist. Also by refining and revising the schedule produced,the project manager is able to maintain control over this typical rapidly changing situation. CPM welcomes a high level of detail,because of this the contractor can break his project into as many activities as necessary to meaningfully describe it.

THE THREE PHASES OF CPM

CPM consists basically of 3 phases, a planning phase, a scheduling phase, and a control -Monitor phase.

1. The Planning Phase

This is by far the most important and time consuming. It is in this phase that all the necessary input data are developed to make CPM work, and since these inputs are crucial, the planning phase must be performed by the contractor, who is the only one completely familiar or knowledgeable about his costs and method of doing business. The contractor can obtain essistance in the accomplishment of the tasks involved in the collection of input data but the basic responsibility for this cannot be delegated to any outside group if CPM is expected to be used successfully.

This planning phase, is one of the vital requirements of CPM and is at the same time, one of its greatest advantages. By performing the work required in this planning and analysis phase one obtains intimate and early knowledge of the job and gains insight into details or anticipated problem areas that are not obtainable in any other fashion. This phase in fact gives the contractor a 'dry run' on the construction. The inputs produced by the planning phase mainly are 1) The Arrow diagram 2) Duration, cost and Resource estimates per activity.

2. The Scheduling Phase

This follows the planning phase and can be conveniently computerised. In this phase the input information is processed and a schedule is produced for the activities in the project. Only simple arithmetic is necessary to prepare the schedule and , for small or medium sized networks it can be done by hand. However, the large networks will require very laborious calculations and it will be cheaper and more accurate to use an electronic computer. Thus the schedule can be produced much faster, Moreover, when a computer is used one can react quickly and inexpensively to unforseen developments in the project whereas manual revision of the schedule when unexpected changes occur, is not so easily accomplished.

The output of the scheduling phase will include a schedule of early and late start and finish times for the activities, the amount of extra time available to an activity , a bar chart if desired, a resource analysis and a cash requirement prediction.

3. <u>The Control Monitor Phase</u>

This is the third and final phase of CPM and for all practical purposes, an electronic computer will be required. Flexibility and prompt recalculations due to changed conditions are the keys to a successful CPM program. One can make attempts to keep the schedule upto-date and meaningful by hand but it is done much more easily and inexpensively with the assistance of the computer.

In the control-monitor phase the contractor is supplied with time status reports indicating the overall status of the project in general and activities in particular . The contractor is also supplied with revised schedules that reflect actual job conditions and the projects status. The contractor can also obtain cost status reports that indicate how much money is being spent for what types of expenses. These reports and revisions can be supplied to the contractor as frequently as desired, by the contractor, in maintaining control over his project.

In this phase information is supplied regarding additions to the project(new activities), deletions from the project, changes in durations, descriptions, trade indicators, cost estimates, or resource estimates, actual start of the activities in progress and finally the actual finish dates of the activities completed. The output of the control monitor phase will include time status reports, revised schedules revised bar charts, revised resource, analysis, revised cash flow predictions and cost status reports.

SUMMARY OF THE THREE PHASES

CPM is thus a planning, scheduling and control technique. It demands reliable input information in order to be effective and the production of these data gives the contractor an opportunity to plan and analyse the project and lends itself to revision and refinement in a continuing process of project control.

BASIC PREMISE OF THIS DISSERTATION

This dissertation focuses attention on the scheduling phase. This includes the basic schedule for activities and also levelling of resources.

Whereas manual computations for CPM scheduling are straight forward and standardised the development of a computer algorithm to perform the scheduling computations offers enough scope for innovation and originality. When using a computer, several aspects of CPM can be considered and included in one compact package program. The methods used, the features included, the limitations the capacity etc are properties which distinguish one CPM program from another. No two computer programs for CPM scheduling will be alike.

This dissertation presents a completely original computerised approach, and a computer program.

The subject of resource scheduling is another challenging one. Several years have been devoted to this particular aspect of CPM, and this dissertation, after studying the efforts so far in this direc tion presents an original algorithm for resource scheduling and levelling. The algorithm for resource levelling experiments with different resource availability levels and calculates the minimum project duration for each level of availability. The program developed is a long range program which allows the management to choose the optimum duration of the project, and the corresponding resource availability, by providing the data which the management needs, to make the crucial decision.

It is also the aim of this dissertation to develop a program which can easily be extended to the control-monitor phase of CPM, and to include PERT networks.

CHAPTER - II

THE SCHEDULING PHASE OF CPM

Fundamentals of the Scheduling Phase

A brief review of the essentials of CPM scheduling is presented under 2 heads viz 1) Basic definitions 2) Mathematical Model.

Basic Definitions

The earliest Occurrence time of an event (IEOT) is the earliest point in time that all activities that precede the event will be completed.

The Latest Occurrence Time of an event (LOT) is the dead line by which time an event must be completed if the project is not to be delayed.

Early Start Time - The first day of the project upon which work on an activity can begin if every preceding activity is finished as early as possible.

Early Finish Time - The first day upon which no work is to be done for an activity assuming that it started on its early start time. It is the duration after the early start for an activity.

Latest Start Time - The latest possible point in time by which an activity must be started if the project is not to be delayed.

Latest Finish Date - The latest point in time by which no further work must be done on an activity if the project is not to be delayed.

Total Float - The difference between the amount of time available to accomplish an activity and the time necessary, OR

The difference between the activity's Latest start time and earliest start time, OR

The amount of extra time available to an activity assuming that all activities preceding have started as early as they can and that all activities following will start as late as they can.

Free Float - The amount of extra time available for an activity if every activity in the project starts as early as possible. It is thus the amount of float that can be allocated to an activity without interfering with subsequent work.

I-node - The node at the tail-end of the arrow representing the activity

J-node - The node at the head-end of the arrow representing the activity.

Mathematical Model

The fundamental steps in the scheduling process are :

Step 1

Set the early occurrence time of the first node equal to zero. Consider each node by turn and compute the early occurrence time of the node as follows : For each activity entering the node, add the activity duration to the early occurrence time of the I-node of the activity. The maximum value obtained is the early occurrence time of the node . Repeat until all the nodes are considered.

STEP 2

Set the latest occurrence time of the last node equal to its early occurrence time. Proceeding backwards, consider each node by turn and establish its latest occurrence time as follows :- For each activity leaving the node, subtract the activity duration from the LOT of the J-node of the activity. The minimum value obtained is the latest occurrence time of the node. Repeat until all the nodes are considered.

STEP 3

Set the Early start time for each activity equal to the early occurrence time of the I-node of the activity, compute the early finish time by adding the duration.

STEP 4

Set the latest finish time of each activity as equal to the latest occurrent time of the J-node of the activity. Compute the latest start time by subtracting the activity's duration.

Step 5

Compute the total float by subtracting the early start time from the latest start time for each activity.

DESCRIPTION OF THE CPM SCHEDULING PROGRAM

Features

1. The aim has been to develop a program which is both simple and versatile, subject to the constraints imposed by the potential of the computer that is readily available (IBM-1620 at the S.E.R.C.Roorkee). The program is flexible and it can be freely modified to suit special needs. The program has been written in the FORTRAN-II Language for IBM-1620.
2. The program first of all reads the number of networks to be scheduled. For each network the program accepts as input data the I-node, the J-node of each activity and its duration. A trade indicator for for each activity can also be fed as data. Other data include the number of sortings of output. The types of sorting desired, the project start date, and the list of holidays if calender dating is desired, and finally counters to decide the types and magnitudes of the output-desired by the user.

3. The output consists of the activity node numbers, the duration, the early start time, the early finish time, the latest start time , the latest finish time, the total float and the trade indicator. This is presented in a convenient tabular form.

4. A special facility that has been provided is that of preparing a CPM calender for the schedule. The early and late start and finish times are converted into actual calender dates by an algorithmic procedure.

5. Another facility that has been provided is that the output sorting can be controlled by the user. This has been provided because of the fact that the way in which the CPM scheduling input data are sequenced affects the utility of the report as a management tool. Some users would prefer a sequence that indicates day-by-day, what activities could start. The early start sequence would be preferred in this case. Others may prefer a sequence by Trade indicator so that all activities with the same trade indicator appear grouped together. This makes it easy to assign tasks to specific groups and to find out just when a certain group or person is needed. Yet another sequence is that of total float, so that we have a list of activities in order of criticality. The program can offer all of these and also additional sortings (including sorting by duration, early finish time, latest start or finish time).

6. The program has built-in checks for errors like looping and dangling etc and the exact nature of the error is described, and the particular node or activity where this occurred is pin-pointed.

7. The program has been written on the assumption that a resource scheduling program will follow, and certain computations are performed early enough so that the resource scheduling program is greatly speeded up. However the program can still be used for a CPM study alone, without a companion resource levelling study.

8. The program assumes that a network is available. It does not generate a network of its own. Thus it is assumed that the planning phase has been completed. There is a school of thought, which subscribes to the view that the preparation of a network is a drudgery and the computer must be programmed to generate a network of its own. But this idea has been vehemently opposed by others who feel that the planning phase (which includes the drawing of the arrow diagram) even though a drudgery, is in fact an essential and a most useful routine that must be gone through if CPM is to be effective.

A detailed discussion on this point is to be found in the text book on CPM by O'Brian (18).

PROGRAM OUTLINE

A broad outline of the program is given below, followed by a detailed description.

The First Stage

The input data is read. This consists of the number of nodes, activities, the number of rearrangements, a counter ISKIP, the types of rearrangements desired, and the I and J nodes of the activities, along with their durations and trade indicators.

The Second Stage

This stage computes the early occurrence times for each node, Simulatneously, two additional arrays NACE , NTAE are generated, which will be used in the resource levelling program. If the counter ISKIP is equal to one, the results so far are printed otherwise the program proceeds to the 3rd stage.

The Third Stage

The backward pass computations are done, starting with the last node. The latest occurrence times are calculated for each node and at the same time two more arrays NTAL and NACL are generated, to be used in the resource levelling program. If the counter ISKIP is equal to one, the results of the stage are printed, otherwise the program continues.

PROGRAM DETAILS

First Stage

The number of nodes, activities and the number of rearrangements is read. Also read is the variable NUMBC, which is the number of columns in the array M. The array M is the central and most important parameter for it includds in its fold almost all the CPM calculation results and the data. M is a double subscripted array (NACTx NUMBC) the input data consisting of the I-node, J-node, the duration and the Trade indicator are read and stored in the first four columns of the array M. Each activity is assigned a row in the array M, and the parameters describing each activity are stored in the columns from 1 to NUMBC.

The first and second columns store the I-node and J-node of the activity. The third stores the duration of the activity. The fourth stores the trade indicator if it is read as input data, and it is immediately transferred to the ninth column.

Also read, is the counter ISKIP, which is to decide whether or not all the calculations results are to be printed.

Second Stage

Purpose: - Calculation of early occurrence time. The EOT for the first node is set equal to zero. Starting from the second node, the program scans the list of activities for their J-nodes and picks up the activities entering the node under consideration (say node K). The duration of the activity is then added to the early occurrence time of the I-node of the activity and the result is stored temporarily in IET. IET is then compared with MAX(which was initially made zero) and if IET is greater than MAX then the value of MAX is updated to IET. The variable MAX helps in maximising the value of the early occurrence time. After all the activities entering the node have been considered, the value of the early occurrence time is set equal to MAX. Then the next node is considered. MAX is once again initialised as zero and the process is repeated until the early occurrence times of all the nodes are calculated. This is accomplished in loop 41.

During the scanning process, as soon as the activities entering a node are picked up. The cumulative total of all activities entering a node so far is stored in the array NTAE and the serial numbers of the activities, in the array NACE. (See List of Symbols for more details). This will be used in the resource levelling program. This stage also includes provisions for the checking of three types of errors (see "Checking for errors" - for details).

If the counter ISKIP is equal to one the arrays NTAE, NACE and IEOT are printed. The counter ISKIP is decided by the user. If it is not equal to one the program proceed to the third stage.

The Third Stage

Calculation of Latest Occurrence Times

This stage is parallel to the second. The operations are similar. Starting from the last node, and proceeding to the first, the latest occurrence times (LOT) of the nodes are calculated, and the arrays NTAL and NACL are generated, as and when an activity leaving a node is encountered. This will be used in the resource levelling program. As before, if ISKIP is equal to one the results of this stage, (LOT, NTAL, NACL) will be printed. This stage includes a provision for the checking of a fourth type of error(see "Checking for errors" - for details).

The Fourth Stage

The values of the early start time, early finish time, latest start time and latest finish times are calculated. The early start time for an activity is set equal to the early occurrence time of the node at the tail end of the activity, and it is stored in the fourth column of the array M. To this is added the duration and the result is stored in the fifth column of M as the early finish time. The latest finish time of an activity is set equal to the latest occurrence time of the J-node, of the activity, and is stored in the 7th column of the array M. The latest start time is computed by subtracting the duration of the activity from the latest finish time and it is stored in the 6th column of the array M.

The total float is computed by subtracting the earliest start time from the latest start time and this is stored in the 8th column. All the above operations are done for each activity. The array M is now fully populated and it contains a complete description of all the activities.

The array M is then printed out in a proper format.

The Fifth Stage

This stage is executed only if rearrangements are desired, i.e., if NUMRE is a positive integer which also denotes the number of re-arrangements.

The array NRENG is read to indicate the types of rearrangement desired by the user. (NRENG stores the column number of the array M). The rearrangement is done according to ascending order of values in these columns of M.

Each of these columns is considered, one at a time, and the column number is stored in NCOL. Two temporary arrays are created ISNO and JTEM. These are single subscripted arrays of dimension NACT. IS NO stores the activity serial numbers and JTEM stores the corresponding values in the column NCOL of the array M. Thus if rearrangement in ascending order of total floats is being done, JTEM will store the total floats of the activities whose serial numbers are stored in ISNO.

The values in the array JTEM are then rearranged in ascending order and as the values are rearranged in JTEM, there is a simultaneous rearrangement of the corresponding serial numbers of the ac-tivities in ISNO.

The array M is then printed out so that the activities are in the new order created in ISNO.

The next rearrangement is considered and the operation are repeated until all the desired rearrangements have been accomplished.

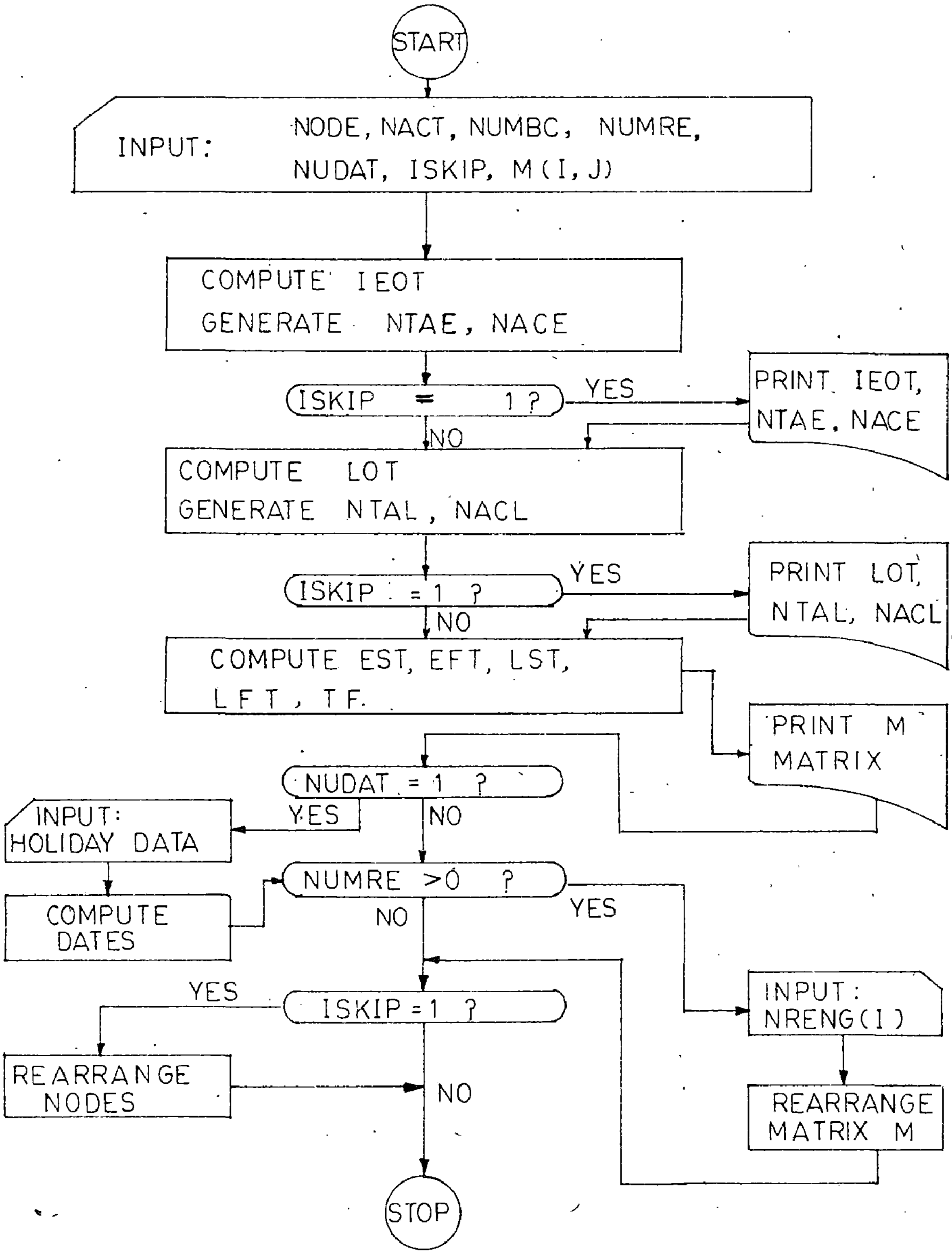
The Sixth Stage

This stage has relevance only for the resource levelling part of the program. The aim is to have the node numbers stored in ascending order of their early occurrence times. The logic used is identical to that used in the previous stage. Thus, the array NDARY of dimension NODE, is generated, and printed.

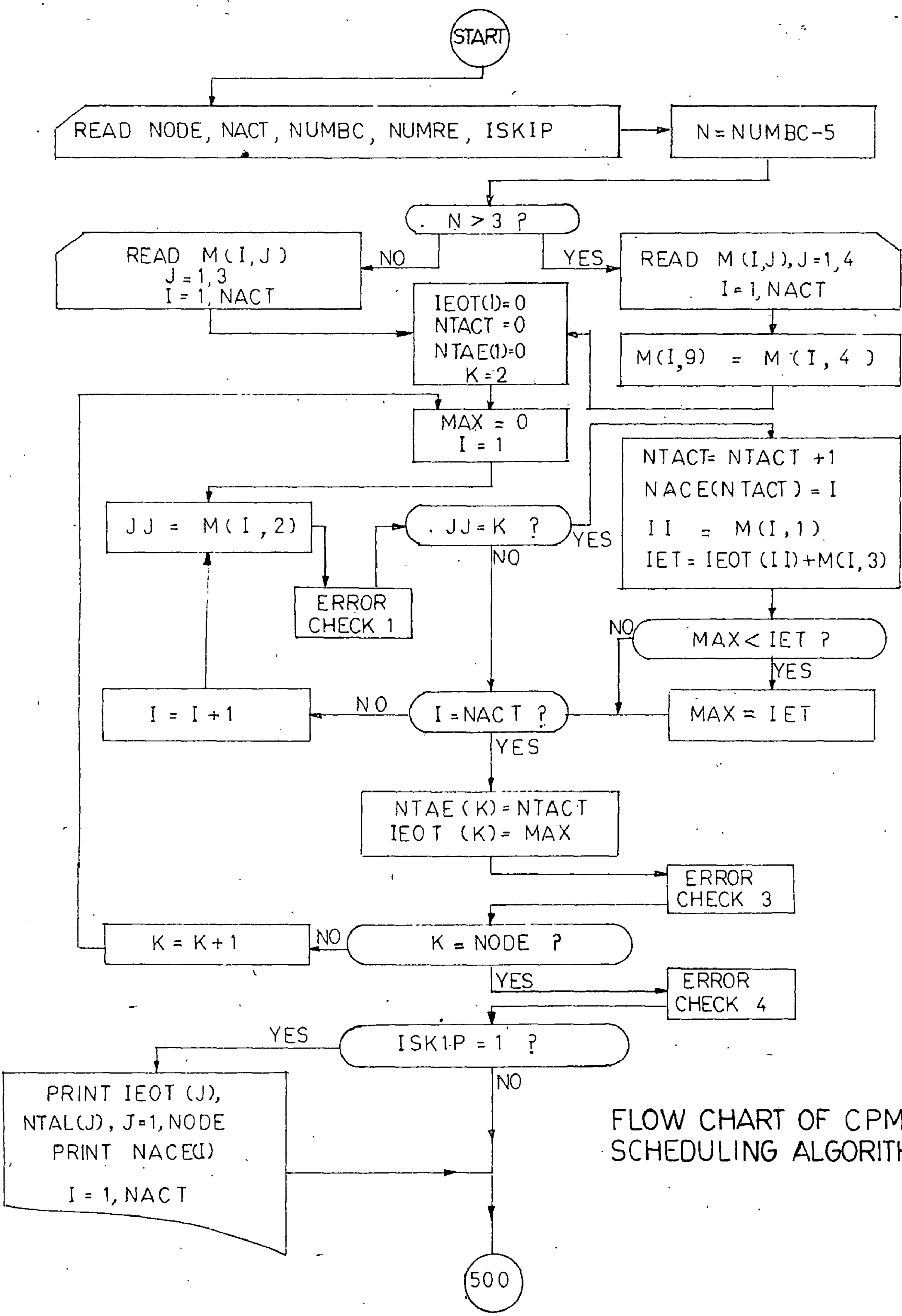
The normal project duration is also determined by setting it equal to the early occurrence time of the last node.

The Seventh Stage

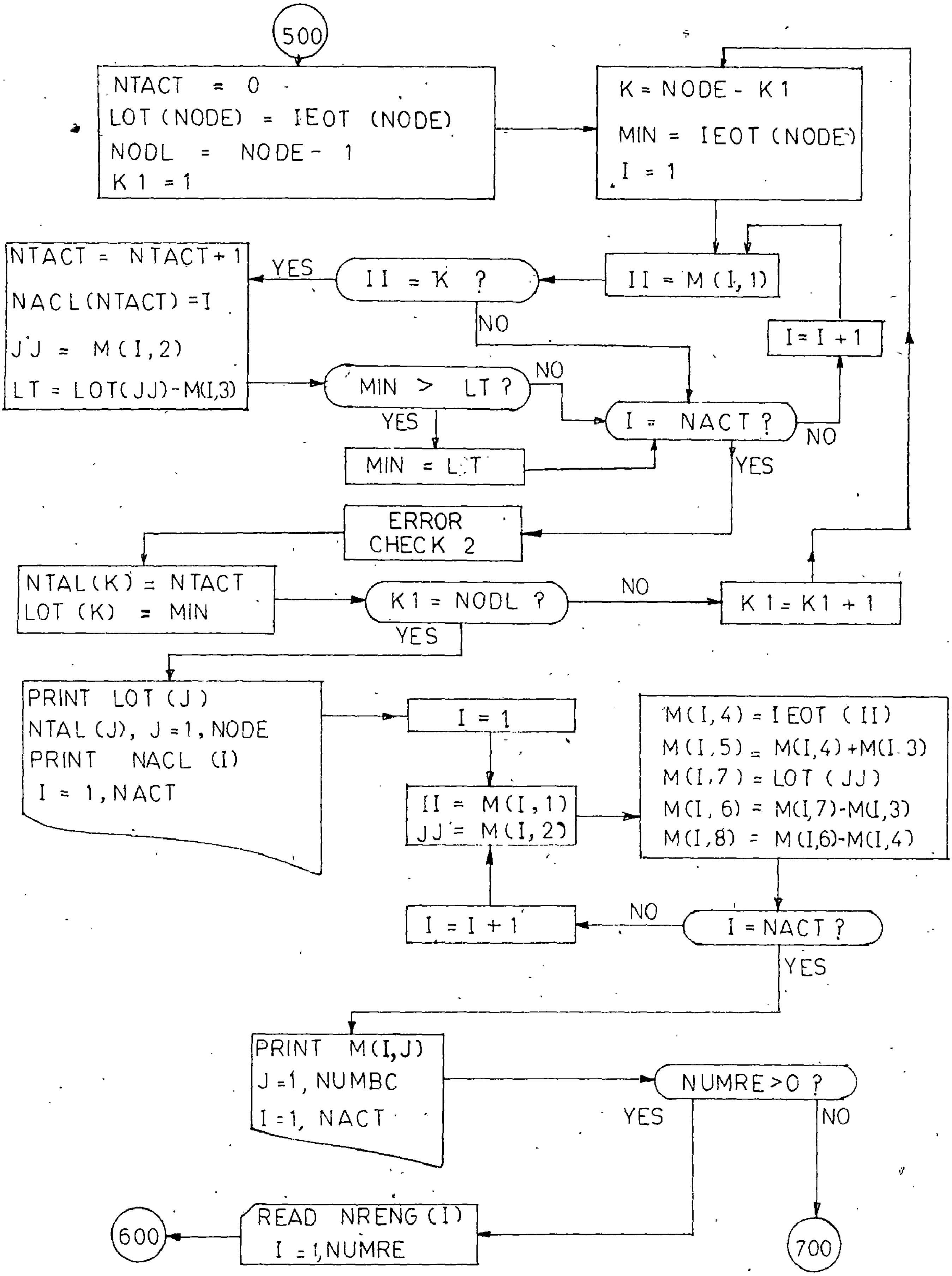
This has been discussed in detail under the heading "The CPM-Calender".


START
INPUT: NODE, NACT, NUMBC, NUMRE, NUDAT, ISKIP, M(I, J)
COMPUTE IEOT
GENERATE NTAE, NACE
ISKIP = 1 ?
YES
PRINT IEOT, NTAE, NACE
NO
COMPUTE LOT
GENERATE NTAL, NACL
ISKIP = 1 ?
YES
PRINT LOT, NTAL, NACL
NO
COMPUTE EST, EFT, LST, LFT, TF.
PRINT M MATRIX
NUDAT = 1 ?
YES
INPUT: HOLIDAY DATA
NO
COMPUTE DATES
NUMRE >0 ?
YES
NO
INPUT: NRENG(I)
REARRANGE MATRIX M
ISKIP = 1 ?
YES
REARRANGE NODES
NO
STOP

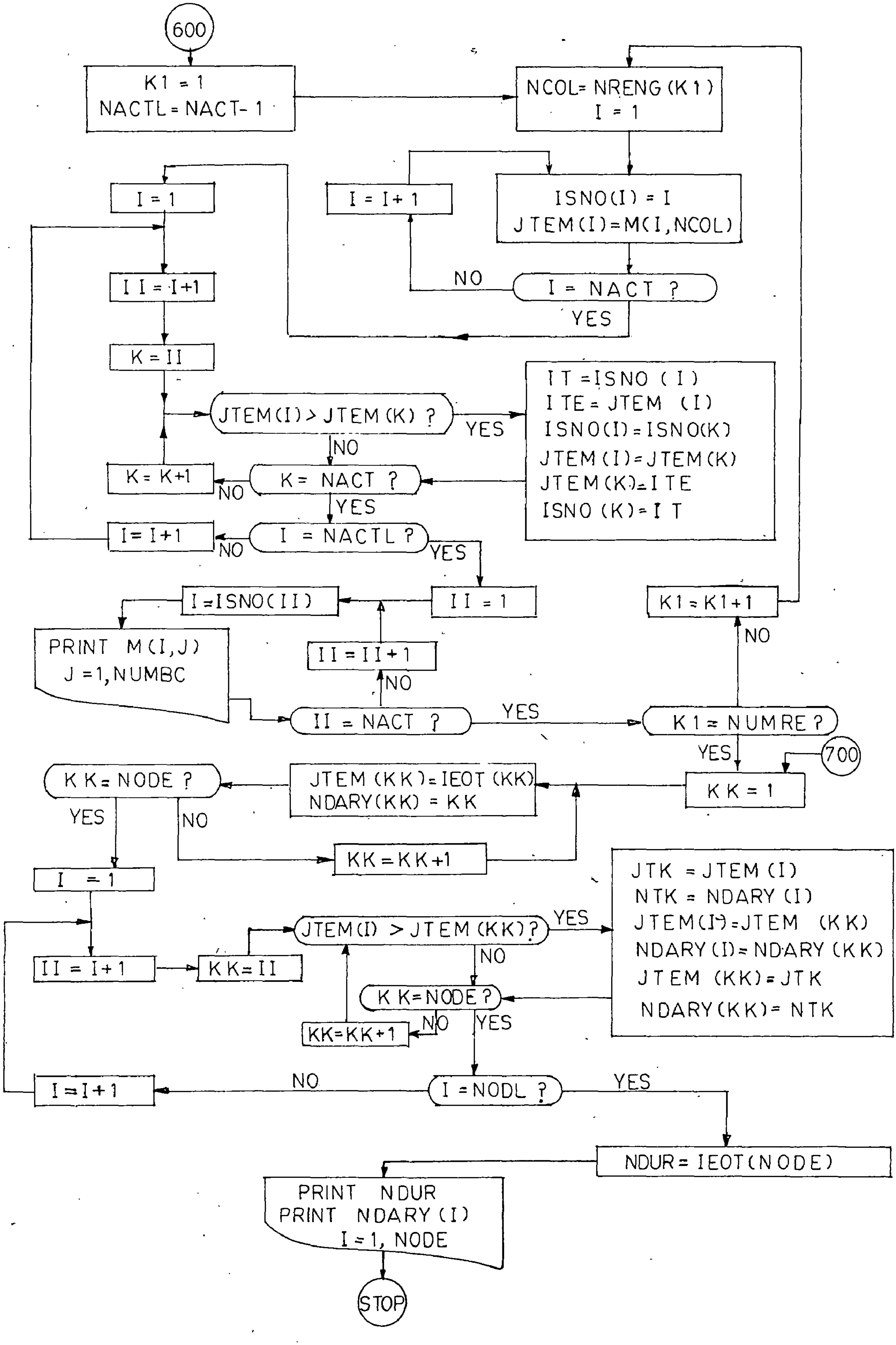
MASTER



FLOW CHART OF CPM
SCHEDULING ALGORITHM



500
NTACT = 0
LOT (NODE) = IEOT (NODE)
NODL = NODE - 1
K 1 = 1
K = NODE - K 1
MIN = IEOT (NODE)
I = 1
NTACT = NTACT + 1
NACL (NTACT) = I
JJ = M (I, 2)
LT = LOT(JJ)-M(I,3)
YES
II = K ?
II = M (I, 1)
NO
I = I + 1
MIN > LT ?
NO
I = NACT ?
NO
YES
MIN = LT
YES
ERROR
CHECK 2
NTAL (K) = NTACT
LOT (K) = MIN
K 1 = NODL ?
NO
K 1 = K 1 + 1
YES
PRINT LOT (J)
NTAL (J), J = 1, NODE
PRINT NACL (I)
I = 1, NACT
I = 1
II = M (I, 1)
JJ = M (I, 2)
M (I, 4) = IEOT (II)
M (I, 5) = M(I, 4) + M(I, 3)
M (I, 7) = LOT (JJ)
M (I, 6) = M(I, 7)-M(I, 3)
M (I, 8) = M (I, 6)-M(I, 4)
I = I + 1
NO
I = NACT ?
YES
PRINT M (I, J)
J = 1, NUMBC
I = 1, NACT
NUMRE > 0 ?
YES
NO
READ NRENG (I)
I = 1, NUMRE
600
700



600
K1 = 1
NACTL = NACT- 1
NCOL= NRENG (K1)
I = 1
ISNO(I) = I
JTEM(I)=M(I,NCOL)
I = I+ 1
I = NACT ?
NO
YES
I = 1
II = I+1
K = II
JTEM(I) > JTEM (K) ?
YES
NO
IT =ISNO (I)
ITE= JTEM (I)
ISNO(I)=ISNO(K)
JTEM (I)=JTEM(K)
JTEM(K)=ITE
ISNO (K)=IT
K= K+1
K= NACT ?
I= I+1
I = NACTL ?
II = 1
I =ISNO(II)
PRINT M(I,J)
J =1,NUMBC
II = II+1
II = NACT ?
K1 = K1+1
K1 = NUMRE ?
700
KK = 1
JTEM (KK)=IEOT (KK)
NDARY(KK) = KK
KK= NODE ?
KK=KK+1
I = 1
II = I+1
KK=II
JTEM(I) > JTEM (KK)?
JTK = JTEM (I)
NTK = NDARY (I)
JTEM(I)=JTEM (KK)
NDARY (I)= NDARY (KK)
JTEM (KK)=JTK
NDARY(KK)= NTK
KK=NODE ?
KK=KK+1
I=I+1
I =NODL ?
NDUR= IEOT(NODE)
PRINT NDUR
PRINT NDARY (I)
I =1, NODE
STOP

The CPM Calender

An important point to be kept in mind is that the CPM schedule yields the start and finish times in terms of WORKING DAYS. Of course, the CPM technique is perfectly general and is independent of the unit of time that has been used . For very large projecsts, extending over several years, a more practical unit would be weeks, rather than days. For smaller projects, working days is convenient, and for industrial processes hours is appropriate.

However, while the working day unit is convenient for the planner, it is not so for the contractor on the field He would naturally prefer actual calender dates.

One of the simplest ways of converting the working days schedule into a calender dated schedule is to prepare the so-called CPM calender.

For small networks, and for short durations, this can be done manually. Starting with zero for the project start date, the dates in the calender, can be numbered in ascending order, skipping out the holidays.

Thus a start or finish time can be converted into the corresponding dates by consulting the CPM calender. But this process is not very elegant. It would be highly desirable if the start and finish dates are made available in a tabular form , against the activity, its duration and other parameters.

If the CPM scheduling has been computerised, then a conversion to calender dates is a must, for the program to be more practical.

This conversion can be accomplished by the computer mainly in 2 ways-

1. By storing the CPM calender in the computer's memory and generating dates in the output by the simple process of matching.
2. By an algorithmic procedure.

The first procedure suffers from serious disadvantages. The first is of course the demands made on the computer's memory, which may be a handicap in case of small size computers. The second is the large number of data cards to be punched. For each working day of the project, a corresponding date, month and year has to be punched as data.

The third and most serious disadvantage is that if there are changes in the project start date or in the list of holidays the earlier data cards punched are all useless, and the CPM calender will have to be prepared once again.

Thus the algorithmic procedure is more rational. If the list of holidays and the project start date are fed as data, the date corresponding to the start or finish time can be calculated and printed in a convenient format by the computer. The advantages are that practically very little storage is required. The data cards are just a few and, if there is a change in the project start date, or in the list of holidays a change of just one or two data cards is all that is needed.

The scheduling program presented in this dissertation includes an algorithm that computes the calender dates also, in addition to the working day schedule.

A brief summary of the algorithm is presented, followed by a detailed description with the help of the flow chart.

The algorithm uses the fact that each start or finish time is essentially an interval of time, using a time of zero (project start date) as base. The algorithm simply "expands" this interval of working days into one of calender days , by including the holidays and the sundays. This expanded interval is then connected to the project start date, and the actual calender date is thus determined.

The project start date is represented by four numbers, one denoting the year, and the others denoting the date, month and day of the week. The holiday list is specified by 2 numbers to each holiday (the date and month numbers). Holidays of the following year are specified by month numbers greater than twelve. Holidays are listed chronologically in the data card. This forms the input . The output consists of the scheduling results in a tabular form. The I-node, J-node, and the duration are also printed along with the early and late start and finish dates.

Details of the Program

The input data consists of the number of activities, the date, month, year and week day number, of the project start date, the list of holidays (other than sundays). Also read is an array NDM which contains the number of days in a month, for each month of the year. The dimensionof NDM is not necessarily 12. The months of the following year are indicated by numbers greater than twelve. For instance 13 indicates January of the following year, 15 indicates March of the following year etc. This procedure is adopted because it eliminates the necessity of storing the year number also. However, in the final output the month number will not exceed twelve and the year number will also be printed.

Finally the first seven columns of the matrix M of the CPM scheduling program is read.

The first step in th operation of the program is to convert the holidays into corresponding calender day intervals, measured from the project start date. Loop 308 accomplishes this by generating an array IHOL(I) of dimension NHOL, where NHOL is the number of holidays. Each holiday is new represented by a single number stored in IHOL, which is the interval in calender days, separating the holiday from the project start date.

Starting from the first row of Matrix M the early start , early finish, latest start and latest finish times are taken by turn and stored temporarily in NDIF. The quantity NDIF is therefore an interval in working days.

If this interval is zero (i.e., the EST, etc is zero) the algorithm directly assigns the date, month and year numbers equal to the starting date, month and year numbers and then considers the next value. Otherwise the program proceeds as follows:

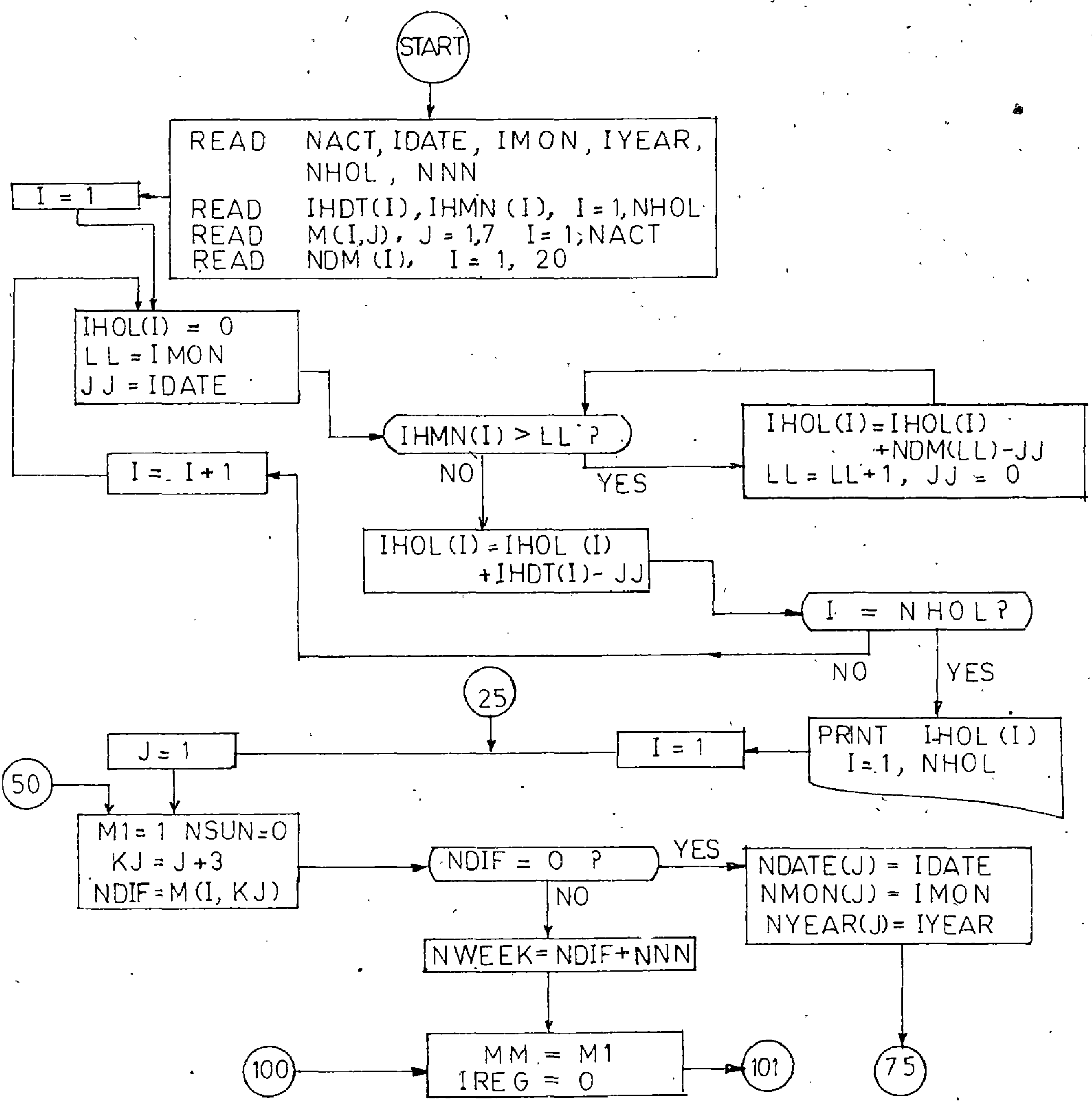
The quantity NDIF is added to NNN which is the week day number of the project start date, (Monday=1, Tuesday =2 etc) to obtain NWEEK. Using the counter MM which is initialised as unity before each date computation, the array IHOL is scanned and the number of holidays coming within the range of NDIF is noted and stored in IREG. As and when a holiday is encountered, the quantities NDIF and NWEEK are updated. This is done in loop 335. NWEEK is the quantity to help keep track of the day of the week. It may be greater than 6 and if so, it will ultimately be reduced to less or equal to 6 (but not zero). simultaneously the counter MM is also updated so that when checking again for holidays, the holidays already considered for expanding NDIF are not included.

After the quantity NDIF has been expanded by taking into account the holidays the program proceeds to compute the number of Sundays in the interval. This is accomplished by simply dividing NWEEK by 6. The value of NWEEK is then made equal to the value of the remainder after dividing by 6. Care is taken to see that NWEEK is not zero (which is a Sunday).

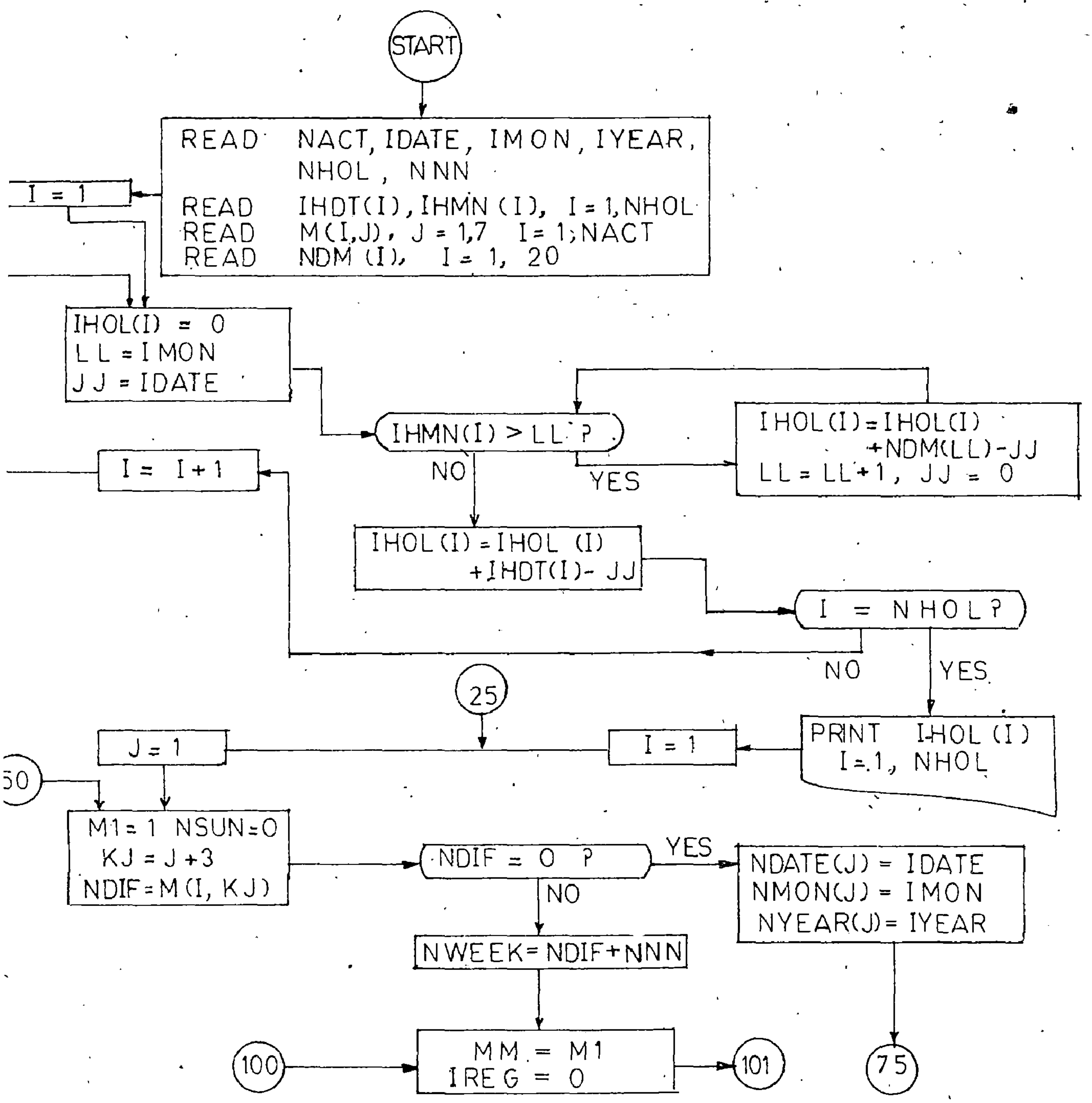
The quantity NDIF is then expanded further by adding the number of s-undays. But, during this process, it may happen that an additional holiday is encountered . The program then goes back to check this. It is here that the counter MM is used. Those holidsys which were included earlier will not be considered. This is possible only when the holidays have been listed in chronological order in the data card, which is the condition imposed by the program on the user.

Once all holidays and sundays have been accounted for the expanded interval NDIF is added to the project start date. If the result exceeds the number of days in the month the month number is incremented and the quantity NDIF is reduced until it is less than the number of days in the month. If the month number exceeds 12, the year number is incremented by one and the month number is decreased by 12.

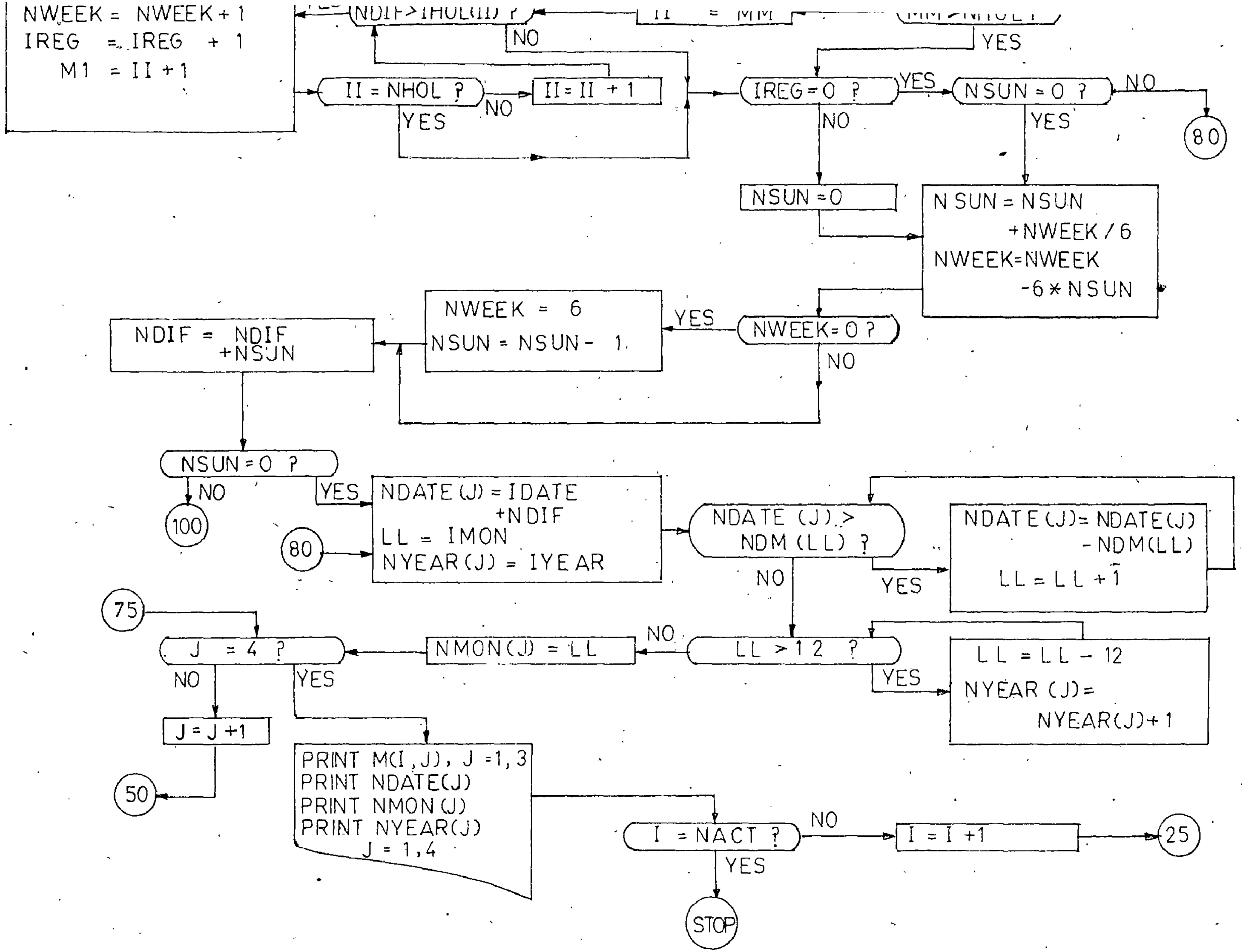
The final date, month and year numbers are stored in NDATE(J), NMON(J) and NYEAR(J) respecitvely, until all the four values in one row of matrix M are considered. They are then printed in proper format before considering the next row (i.e., the next activity).



FLOW CHART FOR CALENDER DATING ALGORITHM



FLOW CHART FOR CALENDER DATING ALGORITHM


NWEEK = NWEEK + 1
IREG = IREG + 1
M1 = II + 1
NO
YES
II = NHOL ?
NO
YES
II = II + 1
IREG = 0 ?
YES
NO
NSUN = 0 ?
NO
YES
80
NSUN = 0
NSUN = NSUN +NWEEK / 6
NWEEK=NWEEK -6 * NSUN
NWEEK= 0 ?
YES
NO
NWEEK = 6
NSUN = NSUN - 1
NDIF = NDIF +NSUN
NSUN = 0 ?
NO
YES
100
80
NDATE (J) = IDATE +NDIF
LL = IMON
NYEAR (J) = IYEAR
NDATE (J) > NDM (LL) ?
YES
NO
NDATE (J)= NDATE(J) - NDM(LL)
LL = LL + 1
75
J = 4 ?
NO
YES
NMON(J) = LL
NO
LL > 12 ?
YES
LL = LL - 12
NYEAR (J)= NYEAR(J)+ 1
J = J +1
50
PRINT M(I, J), J =1, 3
PRINT NDATE(J)
PRINT NMON (J)
PRINT NYEAR(J)
J = 1, 4
I = NACT ?
NO
YES
I = I +1
25
STOP

CHECKING OF ERRORS

An important rule to remember when using the CPM technique, and particularly when using the computer, is "GARBAGE IN-GARBAGE OUT" .

While it is easily realised that CPM can be meaningful only when based on correct data, a common pitfall is the assumption that when a computer is used one need not worry about errors because the computer is always right.

A discussion of errors in the network is presented, because it assumes great importance in computerised scheduling . Often a very well planned arrow diagram, based on realistic data, and using a tested computer program may give absurd results on account of seemingly insignificant errors either in the original network itself, or more commonly, in the preparation of the computer data cards.

A computerised scheduling procedure must therefore necessarily incorporate routine checks for the commonly encountered errors, or el se the program has little value. Of course there are limits to a computer's ability to detect, errors. If the duration of an activity is incorrectly specified, or if the resource requirements are fed incorrectly, there is not much that a computer program can do. But there are certain types of errors which can be detected during the scheduling computations and a good computer program must provide this facility.

Two errors in the network have to be checked in the planning phase itself viz., the wagon wheel error and the waterfall error. These two, being logical errors, are not the computers's responsibility.

The errors that occur due to mispunching of data cards are :

1. The looping error
2. The Dangling error of the first type.
3. The Danglin error of the second type.
4. Error due to a missing activity.

The accompanying figure illustrates these errors and how exactly they arise.

The Looping Error (ERROR CHECK-I)

The Logic

For every activity the J-node is greater than the I-node. Assume that a loop is created by activities A,B,C,... X. The I-node of B is the J-node of A, The I-node of C is the J-node of B etc.

Hence the J-node of B must be greater than I-node of A. Hence it is clear that the J-nodes of each of the activities B,D, D...X are all greater than I node of A. If looping has occurred then J-node of X will be the I-node of A. This violates the condition that the J-node of an activity (in this case X) must be greater than its I-node. Henceto check for looping error, all that is needed is to see that I-node of each activity is always less than J-node for the activity. This is done in the calculation for earliest occurrence times.

The Dangling Error of the First Type (ERROR CHECK-II)

This occurs when a node, other than the last node, has actities entering it, but no activity leaving the node.

This error is checked in the calculations for the latest occurrence times. We make use of the array NTAL which is keeping track of the cumulative total number of activities that leave a node and all its successors. If a dangling error of the first type is present, it will cause the values of NTAL for two successive nodes to be equal. This equality is checked and reported in ERROR CHECK-II.

The Dangling Error of the Second Type (ERROR CHECK-III)

This occurs when a node, other than the first, has activities leaving it but none entering it. This is checked in the calculations for the earliest occurrence times. We make use of the array NTAE which keeps track of the cumulative total number of activities that enter a node and all the preceding nodes. The error is checked in ERROR CHECK-III by ensuring that no two successive nodes have the same value of NTAE, which will be the case if such an error is present.

Error Due to Missing Activity (ERROR CHECK IV)

This is the most troublesome error, and it can seriously upset the scheduling computations . The results of such an error are quite unpredictable, because the computer will try to read the information, For all the activities and if one activity is missing, some other subsequent data card, may be read instead, and the results will be absur d. However, the program will not go far. The error will be detected soon and the computations stop.

This type of error is checked by ensuring that the counter NTACT (which counts the activities entering a node) is equal to NACT (the total number of actigities) after all the nodes have been considered.

When any of these errors are encountered the program immeidately prints an error message and also pinpoints the exact node or activity where the error has been encountered. The nature of the error (whether looping or dangling etc) is also reported.

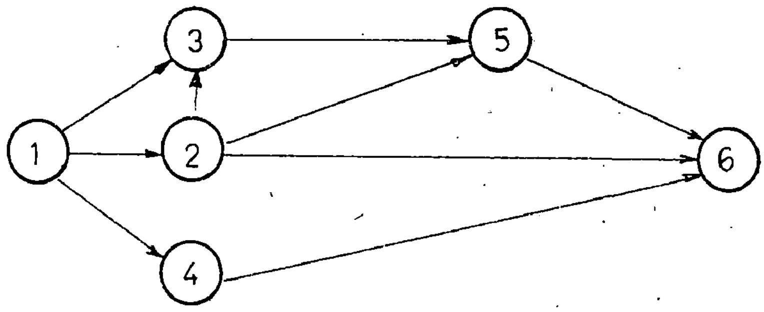
The program then proceeds to the next data set if there is one, otherwise the program terminates.

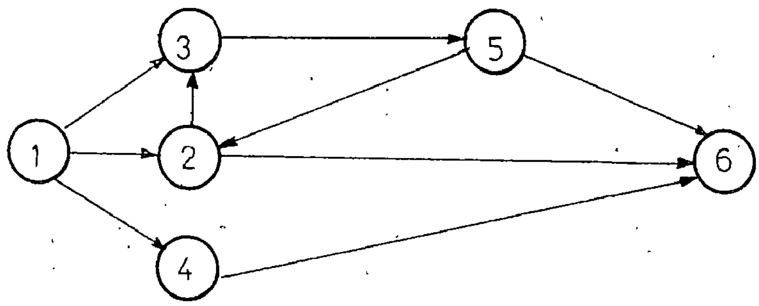
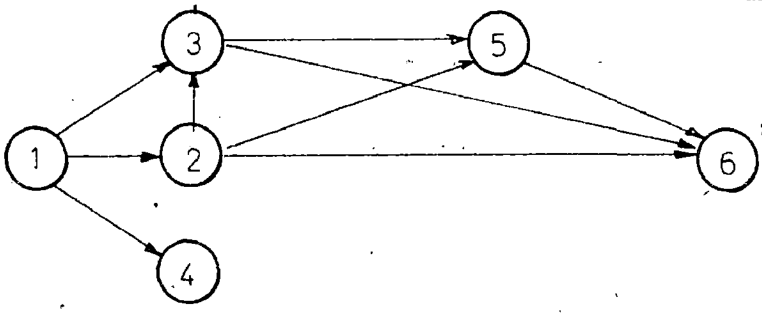
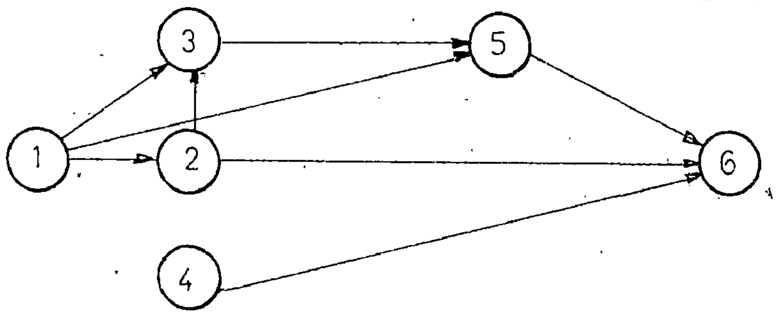
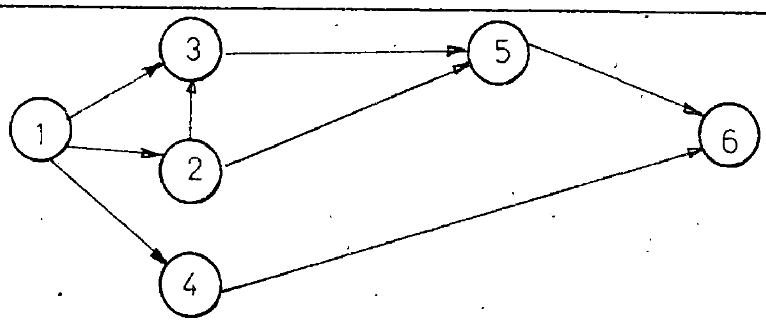
In order to test the working of the error checks in the program, 4 sample data sets were prepared and in each set a data error was intentionally created. Each data set had one of the four errors described previously. The results are attached.

The last error (viz missing activity) needs further discussion. In the fourth data set, one of the activities was left out in the data card. The last data card had space for punching more activities. The space was left unpunched. The computer interpreted the blank card to mean zero values and the three zeros were thus stored as the I-node, J-node and duration of the last activity. This resulted in the computer giving an error message. However the error was reported not as a missing activity error but as an error of duplicate nodes.

This shows that this error is quite unpredictable in its effects. This error has a chance of going undetected but only in very rare cases. Otherwise it will be detected in at least one of the four checks.

ORIGINAL NETWORK



POSSIBLE DATA ERROR	RESULTING INCORRECT NETWORK
ACTIVITY (2)-(5) PUNCHED AS (5)-(2)	 LOOPING ERROR
ACTIVITY (4)-(6) PUNCHED AS (3)-(6)	 DANGLING ERROR OF FIRST TYPE
ACTIVITY (1)-(4) PUNCHED AS (1)-(5)	 DANGLING ERROR OF SECOND TYPE
ACTIVITY (2)-(6) MISSED IN DATA CARD	 ERROR DUE TO MISSING ACTIVITY

DATA AND RESULTS OF ERROI CHECKING PROGRAM

FIRST DATA SET

6	9	8	0	0										
1	2	2	1	3	3	1	4	1	2	3	4	5	2	6
2	6	2	3	5	1	4	6	4	5	6	2			

RESULTS

CHECK ACT 5 AND NODES 5, 2 FOR POSSIBLE LOOPING ERROR IN PROB 1

SECOND DATA SET

6	9	8	0	0										
1	2	2	1	3	3	1	4	1	2	3	4	2	5	6
2	6	2	3	5	1	3	6	4	5	6	2			

RESULTS

DANGLING ERROR OF FIRST TYPE.REFER NODE 4 IN PROB 2

THIRD DATA SET

6	9	8	0	0										
1	2	2	1	3	3	1	5	1	2	3	4	2	5	6
2	6	2	3	5	1	4	6	4	5	6	2			

RESULTS

DANGLING ERROR OF SECOND TYPE.REFER NODE 4 IN PROB 3

FOURTH DATA SET

6	9	8	0	0										
1	2	2	1	3	3	1	4	1	2	3	4	2	5	6
2	6	2	3	5	1	4	6	4						

RESULTS

NODE 0 DUPLICATED IN PROBLEM 4

LIMITATIONS OF THE PROGRAM

The program has been written for a computer of limited potential. Hence, it will naturally have its limitations, which are discussed below-

1. Restrictions in Node Numbering

Node numbering, has received considerable attention in all CPM scheduling programs. The early programs, and the present one also, require care and some discipline in node numbering. The J-node of an activity must always be greater than its I-node. Random node numbering is not per-mitted.

This restriction inhibits the flexibility of the network. However, there are quite a few advantages in the node numbering procedures adopted for this program. The first is that the algorithm is simplified to some extent. Secondly the checking of errors can be done very easily. Thirdly, this system requires much less computer storage. It must be noted that in programs which accept random node numbering, the computer is obliged to renumber the nodes on its own, and perform an internal topological sorting. For a large capacity computer random numbering is desirable. But since this program has been written for s small capacity computer, this luxury of random node numbering has been avoided.

However the present program can be still used as a CORE PROGRAM, for the networks in which the nodes are numbered at random. Provision will have to be made for renumbering the network nodes, and generating a parallel array of nodes, in which the nodes are in ascending order. The present program will then do the scheduling computations and in the output the original random node numbers will appear.

2. Alphabatic Description of Activities

The present program identifies the activities by its node numbers. By a simple modification the input data can be made to include an alphabatic description of the activity and this can be made to appear in the output. This is not a serious limitation and the addition can be made very easily. But a substantial portion of the memory will be taken up by the alphabatic characters. Hence the program has dispensed with this and only the node numbers appear in the output, to describe the activity.

SUGGESTIONS FOR EXTENSION OF THE CPM SCHEDULING PROGRAM

It has been the aim of this dissertation to present a basic program for scheduling computations (including Resource Levelling) for non-stochastic, acyclic directed, networks - particularly CPM networks though it can be extended to other types of networks als o.

Attention has been focussed principally on the scheduling phase of CPM.

The program can be extended to include cost control with CPM. A step in this direction has already been taken and the basic scheduling program of this dissertation was extended to include cost studies (3).

Another possible extension to the program is the inclusion of updating of the network. This aspect of CPM belongs to the Control Monitor phase and presents a challange even to the best programmers.

The updating programs will include as input, the revised start times of the activities in progress, the actual finish times of the completed activities, additions to the network, deletions from the network, and also any changes and in the duration of the activities. The output will be a revised schedule for the remaining activities.

A Stochastic networks like PERT can also be considered and the program can be blown up to include PERT networks also.

LIST OF SYMBOLS IN THE CPM SCHEDULING PROGRAM

IET A temporary storage variable for the earliest occurrence time of a node, during the process of Maximising the early occurrence time of a node (similar to LT)

IEOT A single subscripted array that stores the early occurrence times of the nodes.

ISKIP A counter. If ISKIP is equal to one, it means a resource levelling program will follow and the values of IEOT, LOT, NACE, NACL, NTAE And NTAL will be printed in the result

ISNO This is a single subscripted array and it is used for storing temporarily the serial numbers of the activities during the process of rearrangement.

JTEM This is another parallel single subscripted array which s-tores the quantities that are to be arranged in ascending order.

LOT A single subscripted array storing the latest occurrence times of the nodes.

LT A temporary storage variable for the latest occurrence time of a node during the process of minimising the latest occurrence time of a node (similar to IET.)

M The central and most important array, storing both the input and output of the CPM scheduling program.

The array M is a double subscripted variable, of dimension (NACT X NUMBC).

Each row of M stores information for one activity. The first, second and third columns store the I-node, J-node, and duration of the activity.

The fourth column of M stores the trade indicator in the input. But in the output this is transferred to the ninth column.

In the output, the fourth, fifth, sixth, seventh and eighth columns store the early start, early finish, latest start, latest finish, and total float respectively.

MAX A Variable used for Maximising the value of IEOT. Initially MAX is set equal to zero, and it is progressively increased to its maximum value which is then assigned to IEOT.

MIN This is analogous to MAX. It is used for minimising the value of LOT. Initially it is set equal to a very large value and it is progressively decreased to its minimum value, which is then assigned to LOT.

N A number that indicates how many values for each activity are being read. This is read so that the format is made convenient. If the trade indicator is also fed a format of 16I5 permits 4 activities to be punched on one card. Otherwise the format 15I5 is used to punch 5 activities on one card.

NACE NACL	These are explained later along with NTAE and NTAL
NACT	The total number of activities
NACTL	Defined as NACT-1 .This has no physical meaning.
NCOL	The column in the array M whose values are to be rearranged in ascending order.
NDARY	A single subscripted array of node numbers, arranged so that the corresponding early occurrence times are in ascending order.
NDUR	The normal duration of the project (which is equal to the early occurrence time of the last node)
NRENG	An array containing the column numbers of the array M which are to be arranged in ascending order.
NODE	The total number of nodes in the network.
NODL	Defined as NODE-1 . This has no physical significance
NPROB	The number of problems, i.e., the total number of data sets.
NTAE	A single subscripted array that stores the cumulative total number of activities that enter the node and all preceding nodes. The maximum dimension of this array is NODE . No two values in this array should be equal.
NACE	A single subscripted variable that stores the serial numbers of the activities as and when they are picked up during the search for activities entering a node. The maximumdimension of NACE is [NACT] .

The arrays NTAE, and NACE permit us to determine how many activities enter a node (say K) and also to identify those activities.

Explanation - Consider a node K.

Let LE = Number of activities entering node K

Then LE = NTAE(K) - NTAE(K-1)

LetLE1 = NTAE(K-1) + 1

Let LE2 = NTAE(K-1)+LE

Then the activity serial number entering node K are identified by:

NACE(LE1),NACE(LE1+1),NACE(LE1+2),NACE(LE1+3),.....

...NACE(LE2)

This will be useful in the checking of Dangling Errors, and in Resource Levelling.

NTAL A single subscripted array, analogous to NTAE. This stores the cumulative total number of activities that leave the node and all succeeding nodes. The maximum dimension of NTAL is [NODE] . No two values in this array shouldbe equal.

NACL A single subscripted array that stores the serial numbers of the activities as and when they are picked up during the search for activities leaving a node. The maximum dimension of NACL is [NACT].

The arrays NTAL and NACL permit us to determine how many activities leave a node (say K) and also to identify those activities.

Explanation - Consider a node K

Let LL = Number of activities leaving node K

Then LL = NTAL(K) - NTAL(K+1)

Let LL1 = NTAL(K+1) + 1

Let LL2 = NTAL(K+1)+LL

The activity serial numbers, leaving the node K are

NACL(LL1),NACL(LL1+1),NACL(LL1+2),...NACL(LL2)

NUMBC The number of columns in the array M.

NUMRE The number of ways in which the results are to be arranged.

LIST OF SYMBOLS IN THE CPM CALENDER PROGRAM

IDATE	The date number of the project stort date
IMON	The month number of the project start date
IYEAR	The year number of the project start date
IHDT	A single subscripted array storing the date numbers of the holidays (in chronological order)
IHMN	A single subscripted array storing the month numbers of the holidays in chronological order.
IHOL	A single subscripted array, generated by the program in which are stored the intervals in calender days, separating each holiday from the project start date. The list of holidays are thus represented by one number to each holiday, instead of two, (IHDT,IHMN)
IREG	A counter of keep track of the number of holidays crossed.
M	Adouble subscripted array - same as the array M in the CPM s cheduling program . This is fed as data.
M1, MM	Counters to prevent the inclusion of holidays already considered, during the search for additional holidays.
NDATE	The date number of the EST,EFT, LST or LFT
NMON	The month number of the EST,EFT,LST, or LFT.
NYEAR	The year number of the EST,EFT,LST, or LFT
NDM	An array storing the number of days in a month
NHOL	Number of holidays to be skipped

NSUN	The number of sundays encountered in an interval.
NDIF	The interval in working days, and which is being converted into the corresponding interval in calender days.
NWEEK	A variable to keep track of the day of the week
NNN	The week day number of the projec t start date (Monday = 1, Tuesday = 2 and Saturday=6, Sunday=0)

CHAPTER III

CPM - AND THE PROBLEM OF RESOURCE SCHEDULING

We have seen how the critical path method can be used to plan and schedule projects, and how, in case of large projects, the use of computers makes the job an easy one.

The question now arises: Is the schedule of early start, early finish, latest-start and latest finish, the best schedule? Is it the final schedule?

The answer is No. If resource constraints exist the CPM schedule may be an impracticable one. Even if resources are available, the CPM schedule can still be improved. In almost all projects, resource constraints exist. The management will be keen to see that the best possible use is made of these resources by a judicious assignment of these resources. In the rare (and convenient) situation where sufficient resources can be procured, the management will still be eager to see that the level of resources required is acceptably low. In the more common situation where resources are insufficient, the management will be anxious to see that this causes, the minimum delay in the project.

It is because of these facts, that the CPM schedule is not a final schedule. The completion of the initial CPM computations is merely an indicator that we have now reached a stage when we should decide how to make the best possible use of the floats available to the activities. If we draw up a schedule in which all activities are to start at their early start times, we will find that there is a peaking of resource demand on somedays and hardly any demand on other days. If resources available are limited it will invariably happen that the resource demands exceed the availability on some days and there is surplus resource on other days. Even if resources needed are available on all days, the situation will arise where the projec t manager is obliged to hire 200 men for one day and retrench 150 men the very next day. It is in these situations that the importance of resource levelling is realised. The aim of resource levelling is to make use of the slacks of the activities so that some activities can be postponed and a reduction of peak resource demand is accomplished. If the availability of resources is very low, resource levelling attempts to make optimum use of these resources so that the delay caused to the project is minimum.

Resource levelling is today receiving more attention than any other aspect of CPM, and the significance of the problem is growing rapidly. This is an age of specialisation. We have, by means of Modern technology developed several expensive resources, and the number of 'specialist' personnel is growing. It is not unusual these days to have projects where 50 or more different resources must be considered. Added to this is the fact that big projects include several thousand activities. This has made the sheer magnitude of the problem so great that it is beyond the human intellect to explore even a few of the possible alternatives, for scheduling the project and allocating the resources. Exploring all the possible mathematical combinations contributing to the solution of the problem is beyond even the most sophisticated computers, expept in trivially small networks, and using large computers to explore all possible combinations for a small network is in the words of Wiest, (13) analogous to 'using bull-dozer to move a pebble".

The problem has been made more difficult by the fact that currently there is no mathematical basis for a realistic scheduling procedure. Hence at least for the present, efforts for a precise mathematical treatment have not gone beyond attempts to use linear programming. To add to the difficulties, experts are not agreed on how resource levelling should be done. However, there is general agreement on the need to use computer algorithms based on heuristic techniques. Thus, it is necessary to rely on experience and judgement, to clearly understand the peculiarity of the problem, the practical considerations etc. and come up with an algorithm that narrows our search to a sub-set of 'good schedules", rather than exhaust ourselves by exploring all possible mathematical combinations. Exploring all the possible mathematical combinations is particularly unwelcome because mathematics merely plays around with numbers and the benefits of common sense and engineering and financial judgement is ignored.

It is no surprise therefore, that all the resource levelling and resource allocation algorithms so far use only a minimum of mathematics.

The subjec t of Resource levelling is a challenging one. It is a rich hunting ground for thesis topics. In fact several Ph.D. theses have already come up on resource levelling and many more are on the way. Computer companies are busy developing and improving their package programs for resource levelling. IBM experts have devoted 30man-years already in this field, thus testifying to the magnitude and importance of this aspect of CPM . Infact a CPM study for a construction project is not really complete without a companion levelling study.

A survey of resource levelling algorithms developed so far.

Several researchers have been active in this field and the work of some of the most important is summarised below.

Among the early researchers are Clark (4) Burgess (5), McGee(6) , Kelley(7), and Moshman (8), Then came the papers of Mize (9), Conway(10), Galbraith(11), Brooks(12), Wiest (13), and Fendlay(14).

Among the recent papers is one by Bennet Lawrence (15) and a completely new approach has been presented. Some of these algorithms have been discussed briefly.

Broadly speaking, Resource scheduling problems generally assume that projects are characterised by a technological ordering of the project activities in the form of an arrow diagram. The projects have an assigned start and finish dates (even though tentative) . The resource levels available are specified, along with resource requirements for each activity. Some algorithms permit a relaxation of some of these assumptions.

The algorithms so far developed and published can be roughly classified into 3 broad groups, and the aims of these groups differ basically.

FIRST GROUP

Resource Levelling

Consider a project which must be completed by a specified due date. Assume that the resource requirement on each day of the project is available. The aim is to minimise the resource costs. An ill-planned schedule will result in peaking of resources on some days. It is well known that the cost of hiring and laying off physical resources and personnel can be appreciable. To minimise the resource costs an attempt is made to level the resource demands so that the peaks are reduced.

An ideal resource schedule will be one which is constant, with perhaps an initial build-up period and a terminal tapering-off period. The aim of resource levelling will therefore be a schedule which is as near to the ideal schedule as possible, subject to the crucial condition that the project must not be delayed. The work of Burgess(5), Wiest and Galbraith(11) fall in this category.

SECOND GROUP

Resource Allocation

Consider a project having only limited resource availabilities. The aim is to assign these resources to the activities, so that the duration of the project is a minimum, even if the earlier CPM scheduled date of completion (of the project) is not met. This type of situation is the more commonly encountered one. In these days of advanced technology many specialised resources have to be shared by projects running concurrently.. Hence most of the multi-project resource scheduling problems fall under this category. Most of the recent publications belong to this group like the works of Davis, Mize, Conway etc.

THIRD GROUP

Long term Resource Planning

This is an extension of the Resource Allocation problem, and has no particular resource limitation or fixed project due date as constraints. The aim here is to decide an optimum resource level and the corresponding project duration. It is well known that a reduction in resource levels, will decrease the cost. But the corresponding increase in the project duration will add to the indirect costs of the project. This is in a way similar to the time cost trade off problem.

Long range resource planning is the most difficult of the three, because the constraints are few and the possible combinations can theoretically be limitless, and practically a very large number.

Modern researchers have favoured the third approach. But still the basic idea used to solve the problem is the same, even though the aims are different. All the algorithms developed so far, list the activities in some priority order and schedule them when (i) their predecessors have been completed, and (ii) the required resources are available. The differences arise only in the basic aims of the algorithms and the details.

A survey of resource scheduling algorithms upto 1965 has been published by Davis (16) and hence they are not discussed here.

Two later papers, that of Galbreath (11) and Bennet Lawrence (15), are briefly discussed. One of the earliest resource levelling programs developed by Burgess is briefly elucidated. The algorithm of Burgess is of interest and is discussed to serve as a basis for comparison.

THE BURGESS LEVELLING PROCEDURE

Burgess' procedure belongs to the first category of resource scheduling, algorithms. The project duration is not changed. The peaks are reduced and the valleys are filled as far as possible. The Burgess procedure hinges on the following principle :

"The sum of the daily resource requirements over the project is constant for all complete schedules. But, the sum of the squares of the daily requirements decreases as the resources peaks are clipped to fill in valleys. This sum is a minimum for a schedule which is level or as nearly level as can be obtained for the project in question".

Using the above principle the Burgess procedure can be understood easily. The basic steps are -

1. List the activities in CPM order (ascending J-node subascending I-node) and have the all-early-start schedule available.
2. Shift the last activity to give the lowest total sum of squares of resource requirements for each time unit. If more than one schedule gives the same total sum of squares, schedule the activity as late as possible to

get as much slack as possible for the preceding activities.

3. Holding the last activity fixed, shift the next to last activity, the same way, taking advantage of any slack made by shifting the last activity .Continue the s tep until the first activity is reached. This completes one cycle.
4. Carry out additional cycles until no further reduction in sum of squares is possible.

MERITS AND DEMERITS

Burgess procedure is based on a mathematically sound principle. However, it tends to schedule the activities late and not early. Moreover, the assumption of unlimited resources is not always reasonable. Modern researches do not like to treat the project duration as inflexible . Hence this procedure is not popular now a days.

GALBRAITH'S ALGORITHM

A second resource levelling algorithm is due to Galbraith (11). This algorithm too, does not take liberties with the project duration, which is assumed to be fixed.

The program attempts to solve the resource levelling problem not by any mathematical principle, as in the case of the Burgess procedure, but by using a heuristic technique. The activities are shifted by examining their free floats first and then the total floats.

Another feature is that, unlike the Burgess procedure, the program recognises the fact that resources are limited. It takes note of this and attempts to produce a good schedule within the project duration. When the resource avaialabilities are exceeded, it shifts the activities when it is possible and thus attempts to bring the total recourse requirements within the availability. If recourses required are still greater than the availability, this excess is noted and printed in the results.

MERITS

It has eliminated many of the shortcomings of the Burgess procedure by recognising the fact that resources will be limited. The structure of the program is sound and is easy to follow. The program has provided enough manipulative opportunities that are under the control of the user. A special feature of the program is that it considers the splittability of the activities, and makes use of it during the levelling opperations.

The demerits of the program are -

1. The results are seriously affected if the activities are ordered in a different way. The program considers the ascending order of total float, while deciding the activities to be shifted. But, as the author has himself admitted, the free floats will not be in the same order, and moreover after some floating, of the

activities has taken place the remaining floats will no longer be in ascending order. The author has conceded that a latest start order is more appropriate and this order has been adopted in the algorithm presented in this dissertation.

2. The program succeeds in bringing down the resource peaks, as far as possible but it stops when it has lowered it to the availability level. It does not explore the possibility of further reduction, which may be possible. Thus the levelling may be incomplete.

3. In case the resource demands cannot be brought below the availability level, the program is helpless. It merely reports the fact without offering a solution.

4. Like the early procedures, this program too attaches undue importance to the normal project duration.

THE BENNET LAWRENCE ALGORITHM

A third and a rather interesting algorithm for resource scheduling was developed by Bennet Lawrence(15) The approach to the problem is completely different from the conventional approaches in that, the activity durations are not treated as constants. It can be readily appreciated that if the resources for an activity are increased the duration can be reduced. Also, the same activity can be finished with lesser resources, but it will take more time.

The algorithm of Lawrence Bennet uses a single unit called man-days (or man-weeks, as the case may be) to describe the resource requirements of an activity, instead of specifying the duration and the men required separately. Thus an activity that requires 2 days and 4 men is said to have a resource requirement of 8 man-days The algorithm will permit , if necessary, the activity to extend to 4 days by assigning only 2 men, The algorithm may also vary the assignment of men and will permit 4 men to be assigned the first day and 2 men each for the next two days, thus completing this 8 man-day, activity in 3 days.

The algorithm operates in this manner for all activities, assigning available men of various trades to them based on their requirements and on the number of men available. Input for the algorithm includes the following:

1. The number of men of each trade assumed to be available on each day.
2. The total manpower requirements for each activity for each resource.
3. For each manpower type, 3 values are supplied. The total manpower requirements, the maximum during any one unit of time and the minimum during any unit of time that the trade is scheduled.

Because some activities cannot be worked on by a large number of men without becoming overly inefficient, the maximum number of men which may be practically assigned to this activity is included in the input. Also, because some activities require a minimum number of a particular resource before that activity can start, this minimum is included in the input for each resource. These maximum and minimum values of each resource for an activity can be equated if it is desired to have a constant duration for that particular activity. Otherwise, it is not known beforehand how long each activity will take.

Merits

The algorithm is novel in its approach. Its basic assumption of varying activity durations is very reasonable.

Demerits

1. There are too many decisions to be taken while preparing data. Difficulties may arise in deciding the maximum and minimum number of men that can be assigned. A slight change will drastically alter the schedule alter the schedule.

2. The program cannot be used as a convenient follow up to the CPM scheduling program. The program has no use for the start and finish times of the activities. It can be used only as an independent resource scheduling algorithm. The weakness of the program is that it has no

information, about the criticality of an activity and the resource assignments are made, on a first-come-first served basis, during the scanning of activities for considering eligibility. Thus the order in which the activities are listed will significantly affect the results.

A DISCUSSION OF EARLIER ALGORITHMS

A critical study of the resource scheduling algorithms reveals that the approach has been continuously changing. Initially the emphasis was on levelling the unlimited resources within the fixed project duration, The emphasis then shifted. The duration of the project became a secondary consideration. The resource constraints, and the judicious allocation of these limited resources was the criterion uppermost in the minds of the researchers. But even this has changed. The emphasis these days is optimisation of both resources and duration of the project . Modern programs include a study of cost also and the aim is to come up with an ideal duration and resource combination, for multi project scheduling.

The algorithm developed so far reveal another interesting point, and that is the criterion used to order the activities for scheduling. A change in this criterion will drastically change the resulting schedule Almost all the previous algorithms have recommended the intuitively reasonable criterion of least slack first. Empiricalevidence has shown (and understandably so) that this criterion causes the minimum delay in the project. However it cannot be said that this criterion guarantees the best schedule in all cases. That will depend on the peculiarities of the individual networks. In fact, hypothetical networks can always be contrived to favour some other ordering criteria. However for all practical purposes it will be safe to use the criterion of least slack first which yields the best schedule in almost all cases, and a reasonably good schedule, (if not the best) in exceptional cases.

Another ordering criterion which has found favour with some authors is the ordering with shortest activity duration first. The logic behind this is the fact that if two activities with the same float are eligible for scheduling and resources are available for only one of them, the wait time is minimised by scheduling the shorter activity first.

However, ordering by total slack has its limitations too. As experienced by Galbraith, , who used the free floats and then total floats, if the activities are ordered according to total float, the free float will not fall in the same order. Moreover , floats keep changing.

Moder and Phillips (17) have suggested that the latest start order be used. This has been endorsed by Galbraith (11) tool

Ordering by latest start will be equivalent to ordering by total float.Mareover, the book-keeping is considerably simplified, as the latest start time is fixed and does not keep changing, as in the case of total float.

The relative merits and demerits of the algorithms discussed previously, have been studied. An algorithm has been developed in this dissertation and some of the useful recommendations of the previous researchers have been incorporated.

DESCRIPTION OF THE ALGORITHM

APPROACH

The algorithm can be classified as a long range scheduling program, though as presently written, it does not include a cost study. The algorithm can be considered a long range scheduling program mainly because the aim is not to limit the duration of the project, but to experiment with different resource availabilities and to decide the minimum project duration for each level of resource availability. The managment is thus able to choose its own resourse availability and corresponding duration, based on the results of the scheduling program.

The first trial assumes an unlimited resource availability and this is of course yields an all-early start schedule. This is used as a basis for comparing the results of future trials, which experiment with increasing levels of resource availabilities.

The suggestion of Moder and Phillips (17) has been accepted, and the criterion of latest start has been used to order the activities. However, the duration of the ac tivity has also been given weightage. Hence the activities have first been listed in ascending order of latest start time and within this criterion, in ascending order of activity duration.

FEATURES

1. The program is designed to be a continuation of the CPM scheduling algorithm. Hence it will accept as an essential input, the results of the CPM scheduling program, in addition to the resource availabilities and requirements.
2. The program operates by iterating on trials and within this, by iterating on days, and assigning resources to the eligible activities subject to availability.
3. A special feature of the program is that the process of selecting the eligible activities is very rapid and efficient. Instead of scanning, the activity list to search for activities, the list of nodes is examined and the eligible activities are selected from the eligible nodes. This procedure is more efficient for two reasons -

i. The number of nodes is less than the number of activities

ii. By rejecting one node during the scanning a whole bunch of activities is rejected , thus speeding up the process significantly.

Provision has been made so that even in this scanning of nodes for eligibility, the search is limited to the likely range and not the whole list. Care has been taken to see that no eligible node is missed.

4. Another feature of the program is the inclusion of a "resource scheduling progress report " in the results. The results are in three parts.

I PART

Progress Report.

During the operations for selecting the activities and assigning resources, the program keeps track of the progress and prints out information which will be very useful in deciding resource availabilities for future trials. This progress report will give the list of eligible activities on each day, the days when no activity can start, and its cause (due to ineligibility or due to lack of resources). The report also tells us the days on which critical activities (which are named) could not start due to lack of resources and further the particular resources which were found lacking are also pin-pointed. This will be particularly useful in deciding which resource to increment for use in subsequent trials.

This day-by day progress report is printed for each trial before the other two parts of the results appear.

PART II

This gives the revised project duration for that particular trial, the revised schedule of start and finish dates for each activity, along with their durations and resource requirements, in a convenient tabular form. Also included is the latest start time of the previous CPM schedule, for the purpose of comparison. This is not the revised latest start time, and it is included only to give an idea of the amount of postponement each activity has had to undergo.

PART III

This gives the total daily resource requirements for each resource . This can be used to plot a histogram.

5. The program has been so written that the resources are levelled even below the level of availability, specified, if this is possible.

6. The program is sufficiently flexible to include multiproject scheduling.

OUTLINE OF THE RESOURCE LEVELLING PROGRAM

STEP ONE

Reading input Data

The input data consists of 2 parts. The first is the results of the CPM scheduling program. The second is data regarding the number of resources to be considered, the availabilities of the different resources, the resource requirements of the activities, the number of trials etc.

STEP TWO - Initialisation

The activity and node status are initialised as zero. The resources assigned are also initialised as zero. The counters are given their initial values. These initialisation steps are carried out for each trial and within each trial for each day of the project.

STEP THREE - Checking of Eligibility and Selection of eligible activities.

Here the nodes are scanned, and the eligible nodes for the day are picked up and stored. A node is tested for eligibility by ensuring that all activities entering it have been completed on the day under consideration. Care has been taken to avoid unnecessary scanning of the whole array of nodes by limiting the search to a small but definite range.

The activities leaving the eligible nodes are examined and if they have not already started, they are included in the set of eligible activities for the day. If no eligible activities are selected the output contains a message to that ef-fect and the next day is considered.

STEP FOUR - Deciding Priority Order of Activities

In this step the selected activities are re-arranged in priority order. The criterion for deciding priority is the criticality of the activity. This is accomplished by rearranging the activities' serial numbers in ascending order of their corresponding latest start times, and within this criterion, in ascending order of duration.

STEP FIVE - Checking Resource Availability and Assigning of Resources

Starting with the most critical activity for the day (ie the first in the priority list), the resource requirements are compared with the net availability for the day, for each resource. If resources are not available for a particular activity, and if the activity is critical or has become critical by postponement, the serial number of the activity is noted and printed in the output along with the day under consideration and the scarce resources. If the activity is not critical, then, it is ignored and consideration moves to the next eligible activity for the day.

If all resources required are available, then they are assigned to the activity for the whole of the activity's duration.

The counters for keeping track of the activities scheduled are immediately updated. The day under consideration is noted and s tored as the new starting time of the activity. The new finish time of the activity is also calculated and stored. The cumulative total resources assigened so far, for each resource, is also updated, by adding the resource requirements of the activity that was scheduled. This is done for the entire duration of the activity. Thus, all the eligible activities for the day are considered.

If no activity can start due to lack of resources, a message is printed in the output to that effect.

STEP SIX Updating of Counter M1

This counter is an important one, because this helps in avoiding unnecessary , scanning of nodes for eligibility on the next day. The nodes have been listed in ascending order of their early occurrence times. While scanning for nodes M1 determines the starting point for scanning. Initially we scan from the first node, but later we attempt to eliminate the nodes which have all activities leaving as scheduled.

Hence, in this step, the eligible nodes for the day are examined and if all activities leaving them have started, then by updating counter M1, we eliminate this node from the set of nodes to be scanned on the following day.

If all the project activities have started then the program proceeds to step seven. Otherwise the next day is considered and after the necessary initialisations , steps 3 to 6 are repeated.

STEP SEVEN - Calculation of Projec t Duration and Printing results.

Starting from the normal project duration, the days are examined one by one and the earliest day is determined when no resource assignments have been made. This decides the project duration for this trial and for the considered level of resource availability.

The results of the trial are then printed out. This includes the list of activities, their durations, their resource requirements and their revised start and revised finish times. Also printed are the latest start times of the CPM schedule for comparison.

The output includes the total resources assigned for each day of the project, and for each resource.

Consideration then moves to the next trial until all trials have been carried out . Trial one is computed with unlimited resource availability and will yield an all-early start schedule. Subsequent trials are carried out for different resource availabilities (as specified).

DETAILS OF THE PROGRAM

(Refer Flow Chart and List of Symbols)

STEP ONE - Reading of Data

This step, reads the data . viz the number of nodes, activities, the normal project duration the number of resources, the number of trials and the starting trial number (if it is desired to start straight away from some trial number other than one). The array containing node numbers in ascending order of early occurrence times (NDARY), the early occurrence times of each node, the arrays NTAL,NACL, NTAE and NACE generated in the CPM scheduling program are also read in.

The first six columns of the array M of the CPM scheduling program are read. This will include the node numbers describing the activities, the durations, the EST, EFT and LST . Of these the EST and EFT are of no use in the program and these values will be erased subsequently to make room for the revised EST and EFT. The resource availabilities, in each trial and of each resource is also read in. And finally the resource requirements of each activity of each resource are read in. As presently written, for a particular resource, a constant availability has been assumed throughout the project. If it is desired to vary the availability of the resources on each day of the project, this can be easily accomplished.

STEP TWO - Initialisation

For each trial the following initialisations are done:

1. The status of the first node is made one, while that of others is made zero. The node status (LSTAT) of the nodes will be made equal to one, as and when all the activities entering the node are completed. Initially, in each trial only the starting node satisfies this condition. The initialisation is done in loop 490.

2. The EST and EFT which are stored in the fourth and fifth columns of the array M are made zero. This helps in converting these values automatically into a status indicatur, for each activity. The program has been so written that the values of EST and EFT, retain their usual meaning in the output, but during the program they are put to good use by using them as status indicators. Thus if the EST and EFT are zero it means the activity has not yet started, and it is not yet eligible to start.

   If EST is some positive integer it means the activity is eligible to start, though it may not have started yet.

   If EST and EFT are both positive integers, it means the activity has started though it may not be complete. The completion of the activity is indicated by

EFT being a positive integer (not zero) and less than the day number under consideration.

This initialisation is done in loop 505.

3. A third initialisation is made by equating the total resources assigned to zero. This is done for each resource and for each day of the project. Since itis not known how long the projec t will last, the initialisation is done for a period equal to one and a half times the normal project duration. If need be, this initialisation can be done for an even longer period. This is done in loop 510.

4. The counter M1 is set equal to one, and the cumulative total number of activities that have started (NTASD) is set equal to zero. The day number (IDAY) is set equal to one.

The initialisations described above are made at the commencement of each trial, before the day by day iterations begin.

For each day of the project the following initialisations are made.

1. The number of ac tivities that started on each day (NASD) is made zero.
2. The counter ICT which counts the eligible activities on each day is made zero.
3. The counter MM is set equal to M1, so that the scanning

for nodes that are eligible on the day, is restricted to a small but definite range.

4. The counter ND which counts the eligible nodes for the day is set equal to zero.

STEP THREE - Selection of Eligible Activities

This is done in the following steps.

Loop 520 scans the nodes in NDARY which are in ascending order of their early occurrence times. If the early occurrence time of the node is less than IDAY then it is a potentially eligible node, and the counters ND is incremented by one. The node number is stored in LMN. This will be used in step 6, for updating the counter M1. It is to be noted that loop 520, scans the nodes in NDARY with MM=M1 as the starting point and it stops scanning as soon as a single node with EOT greater than or equal to IDAY is encountered.

After a potentially eligible node is picked up, it is checked for its status (LSTAT) . If it is one it means that all the activities entering this node have been completed and hence this node is eligible. If LSTAT is zero then it means a check will be necessary.
In the firstt trial however this check is dispensed with, since all the resources required are available, the activities start at the early occurrence time of the node at the tail end.

The check for the completion of all activities entering the node is made in loop 526 as follows - The activities entering the node are examined, with the help of the arrays NTAE and NACE (See list of symbols for details).

If the activity is not a dummy it is examined whether its finish time is a positive integer, less than IDAY, which means it is complete.

If the activity is a dummy it is examined whether at least its EST is a positive integer, which indicates that the dummy is eligible to start, and hence passes the test.

If all the activities entering the node, pass the tests, then the node is designated as eligible and its status (LSTAT) is revised to one, so that in future it is not tested again unnecessarily.

If any one of the activities fails the tests the node is discarded and the next node is considered.

Once a node is designated as eligible the program proceeds to select from the activities leaving the node, those which have not yet started and included them in the array ISEL, which stores the eligible activities for the day.

As soon as an eligible activity is selected and stored in ISEL, its latest start time is stored in ITEM. The EST for the activity is then given a positive integral value (the value IDAY) . This should not be taken to be the actual start time of the eligible activity, but must be treated as an indicator that the activity is eligible to start. The actual start time will be assigned only after checking whether resources are available. This selection of eligible activities from the node under consideration is done in loop 527, which is nested within loop 520. The counter ICT counts the eligible activities for the day.

After all the eligible activities for the day have been collected, it is seen whether ICT is zero. If it is zero, a message appears in the progress report and it is made known that no activity was eligible to start on this day. Otherwise the program proceeds to the next step (FOUR) except in the first trial, where it skips the next step (FOUR) and proceeds to step FIVE.

STEP FOUR - Deciding Priority Order

This step is designed to rearrange the activity serial numbers, stored in ISEL, so that their corresponding latest start times are in ascending order. The values in the array ITEM are examined and rearranged. ITEM stores the latest start time. If any two values in the ITEM are equal then the duration of the activities are examined and the activities are arranged in ascending order of their durations. This is accomplished in loop 550.

The eligible activities for the day are then printed in priority order. This forms part of the progress report.

STEP FIVE - Checking Resource Availability and Assigning Resources

Starting with the first activity in ISEL, the program, first checks whether it is a dummy. If it is, the checks for resource availability are omitted and the activity is treated as scheduled, and the next activity is considered. If it is not a dummy a check for resource availability is made for each resource, in the loop 560.

Loop 560 operates as follows. Counter JLACK is first set equal to zero outside the loop. If all resources required for an activity are not available, it is checked whether the activity is critical or has become critical in which case the counter JLACK is updated each time a limitation is encountered. The counter JLACK thus counts the number of resources found lacking for the critical activity. The actual resource serial numbers are stored in the array JLSN.

When all the resources have been examined a message is printed in the output, which will tell us the serial number of the critical activity which could not start on that day due to lack of resources. The parti- cular resources found lacking are printed and so also the day on which it happened.

However, if the activity is not critical, this is dispensed with and the next eligible activity is considered.

If all resources required are available, the resource assignments are made for the duration of the activity. This is done in loop 580. The array JRSAS which stores for each day the resources assigned, is updated. As soon as this resource assignment is made the counters NASD and NTASD are incremented. The activity is assigned its start time and finish time. These are the final and correct start and finish times and they concel any earlier values assigned. Consideration , then moves to the next eligible activity, until all eligible activities have been considered.

The step is accomplished in statement nos 557 to 581 of the program.

STEP SIX

Updating of Counter M1

This step consider the eligible nodes for the day by turn. The activities leaving each node are examined and if all of them have started, the counter

M1 is incremented, and thus in the scanning for nodes on the next day this node is automatically eliminated, as the scanning starts from the $M1^{th}$ node in the array NDARY.It must be noted that this has become possible only because the node numbers were stored in NDARY in ascending order of their E.O.T. Otherwise there would have been no alternative to checking all the nodes.

While examining the node, if even a s ingle activity is found to have not yet started, then this updating process is halted at once and consideration moves to the next day after first checking whether or not NTASD = NACT . If NTASD is not equal to NACT it means all the project acstivities have not yet started and consideration moves to the next day.

If NTASD = NACT, the program moves to the next step and prints the results.

STEP SEVEN - Results

This is the final step of the program, and only when all the activities of the project have started, this step begins.

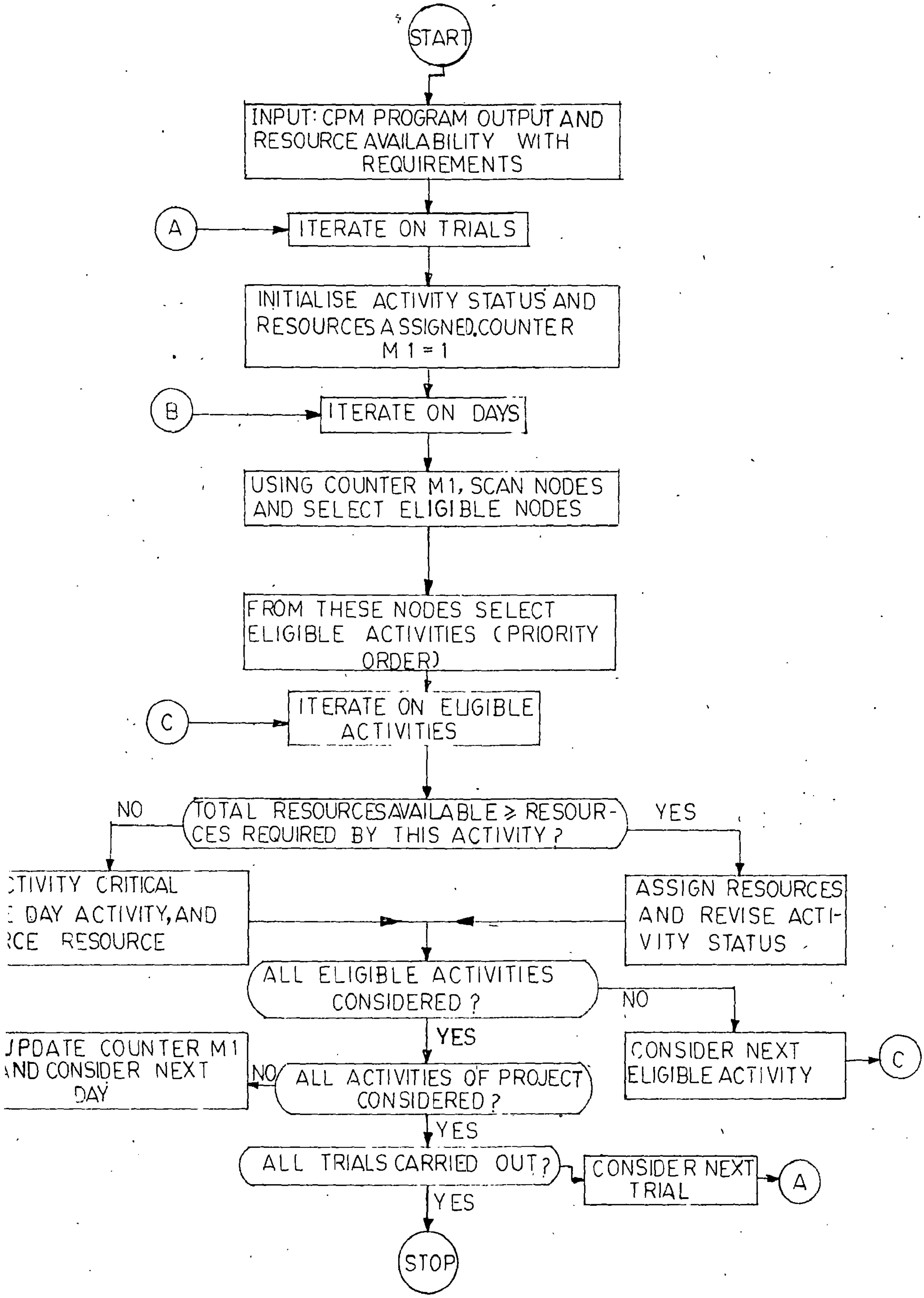
The revised project duration is first determined. Advantage is taken of the fact that at the commencement of the trialthe resources ass igned were initialised as zero. The program starts fromthe last day of the early CPM. schedule and examines each day. The first day on which no resources assignments have been made gives the end of the project. The revised project duration is therefore one day less.

Statements 608 to 610 , which includes the loop 603, will accomplish this.

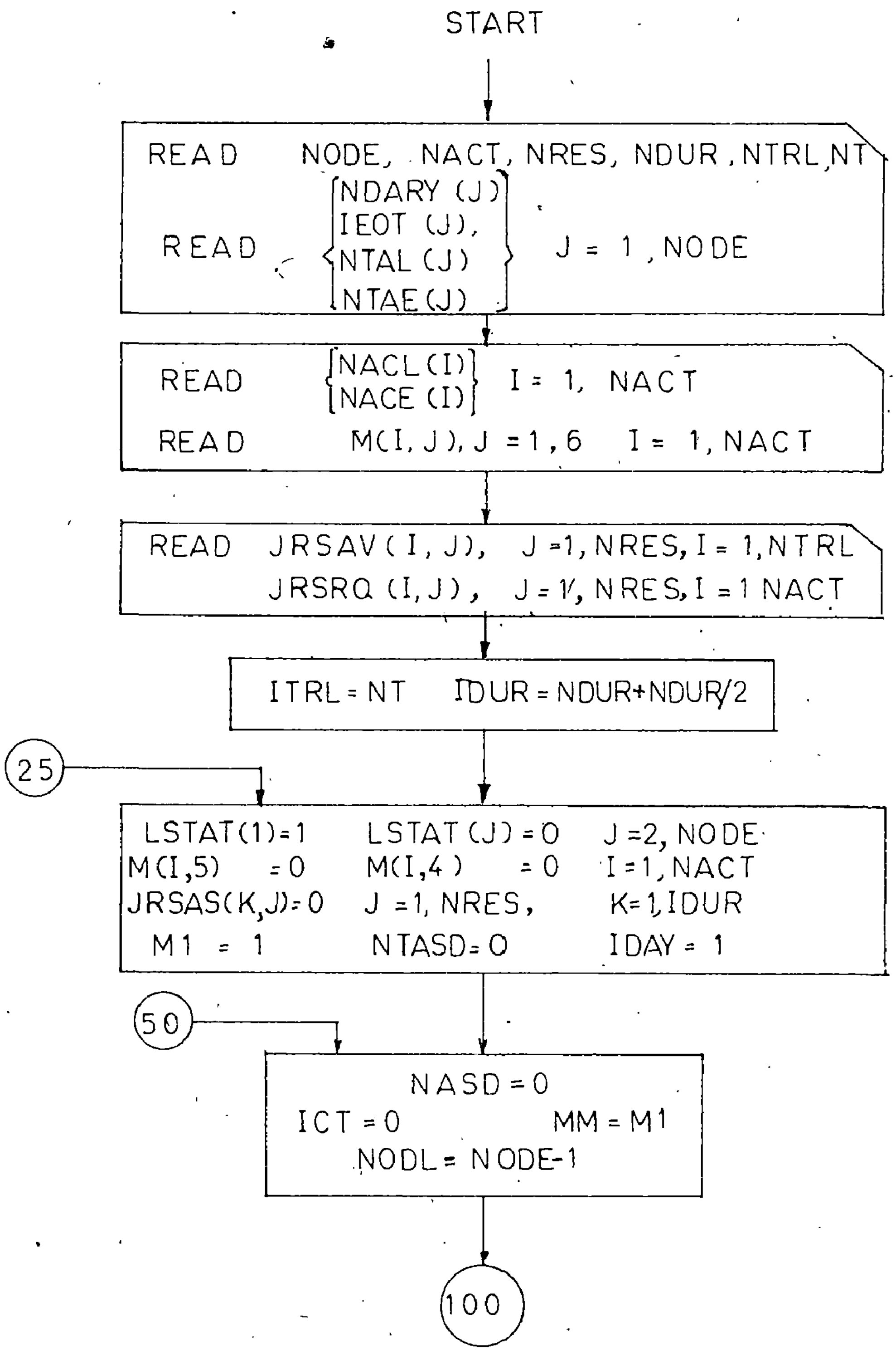
The results of the program are printed. This includes the first six columns of the array M which give the node numbers describing each activity, the duration, the revised early start and early finish times. Also included is the latest start time of the previous CPM schedule. The resource requirements of the activity also appear. This is printed in a tabular form, and each row gives all the relevant information for one particular activity.

This is followed by the total daily resource requirements of each resource, in a convenient tabular form.

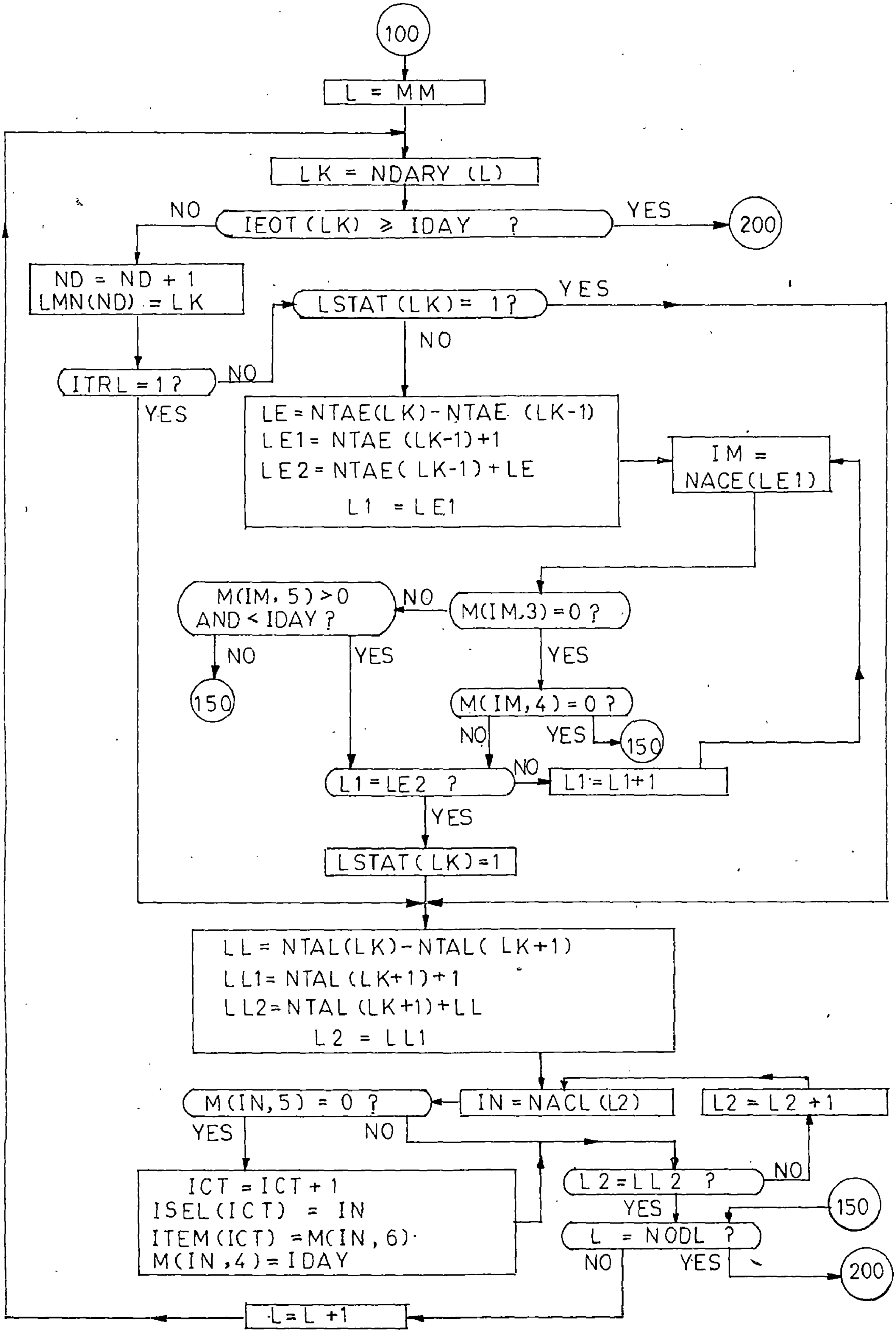
These are the results of one trial. The program then considers the next trial until either the project duration is brought down to the normal project duration, or all the trials have been considered.

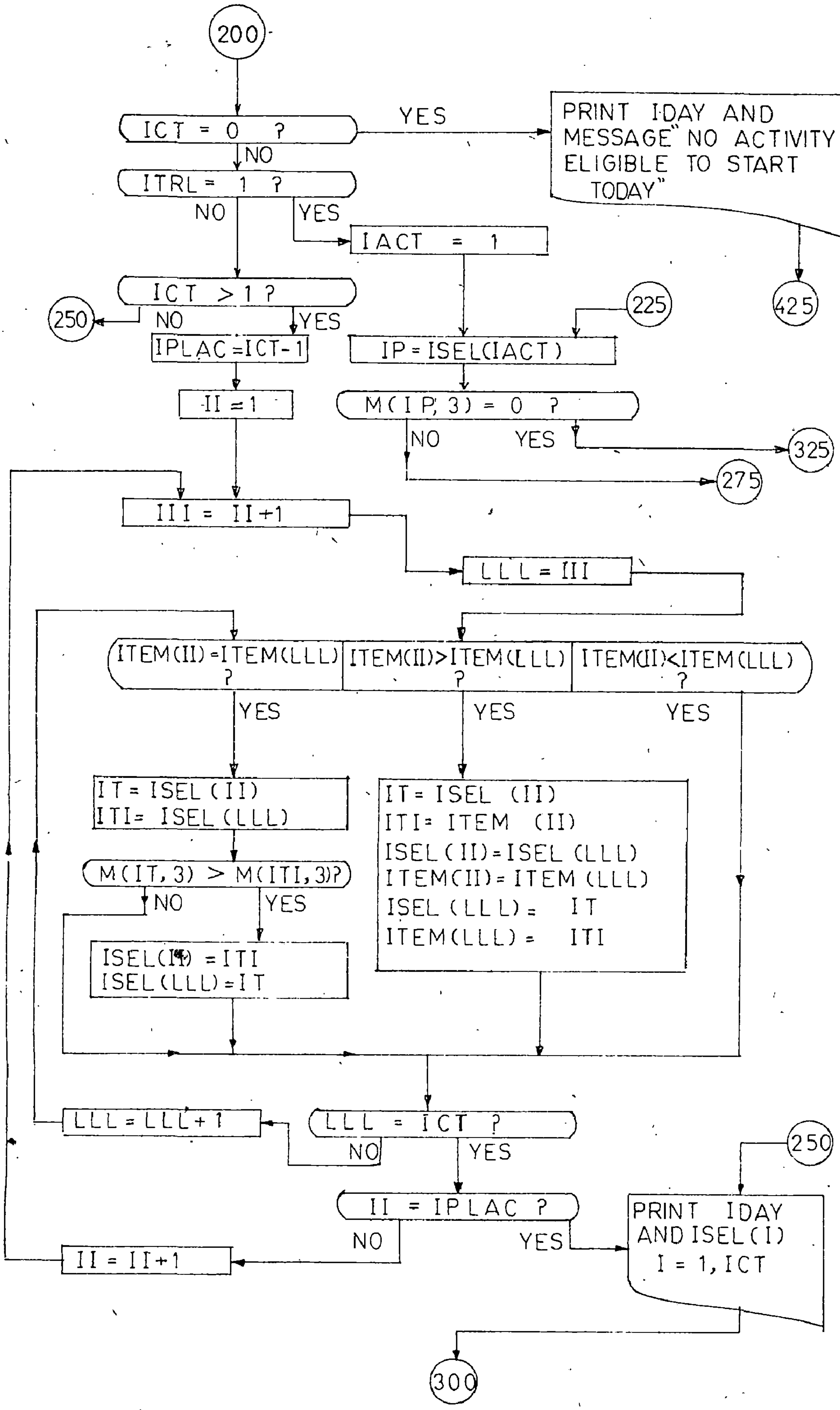


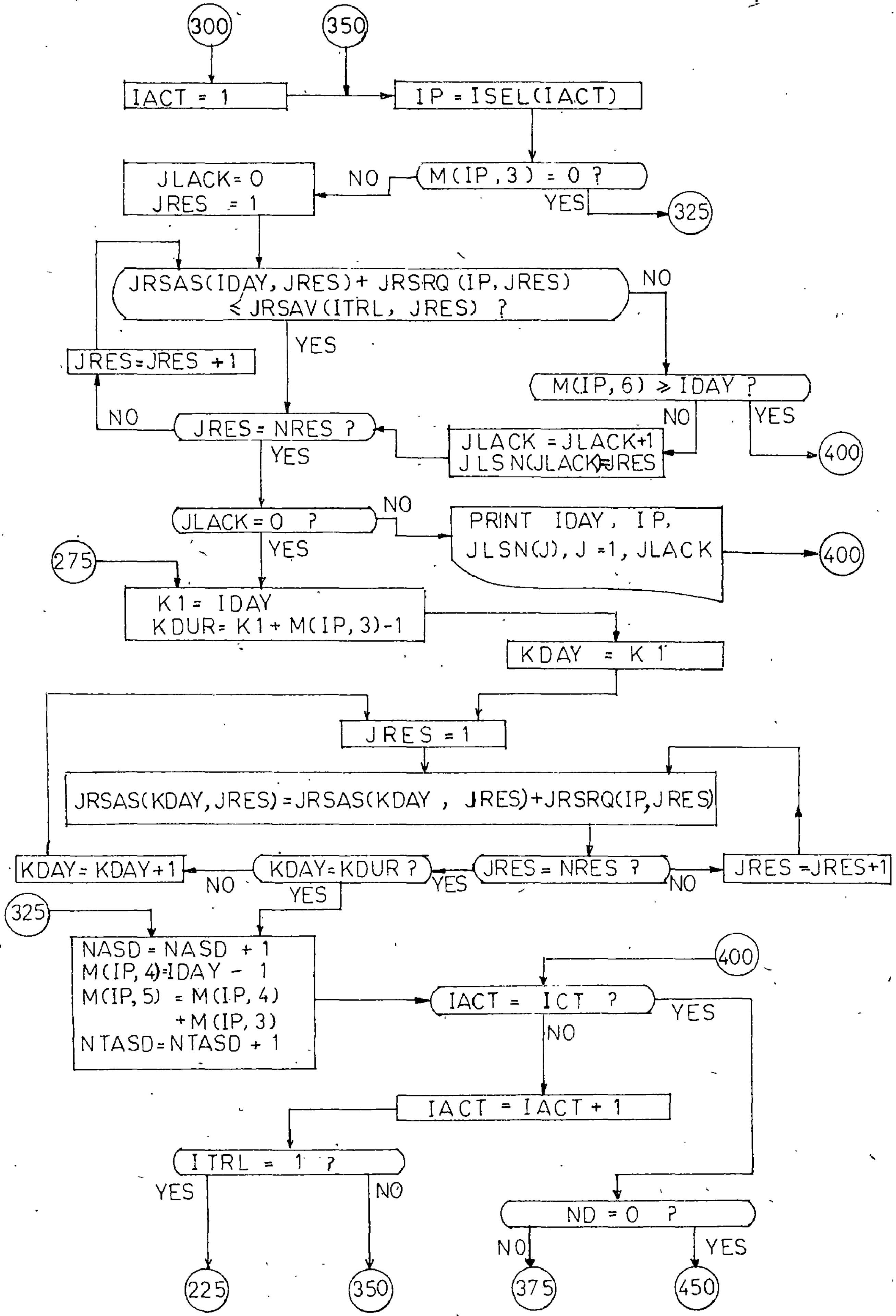
BLOCK DIAGRAM FOR RESOURCE LEVELING ALGORITHM



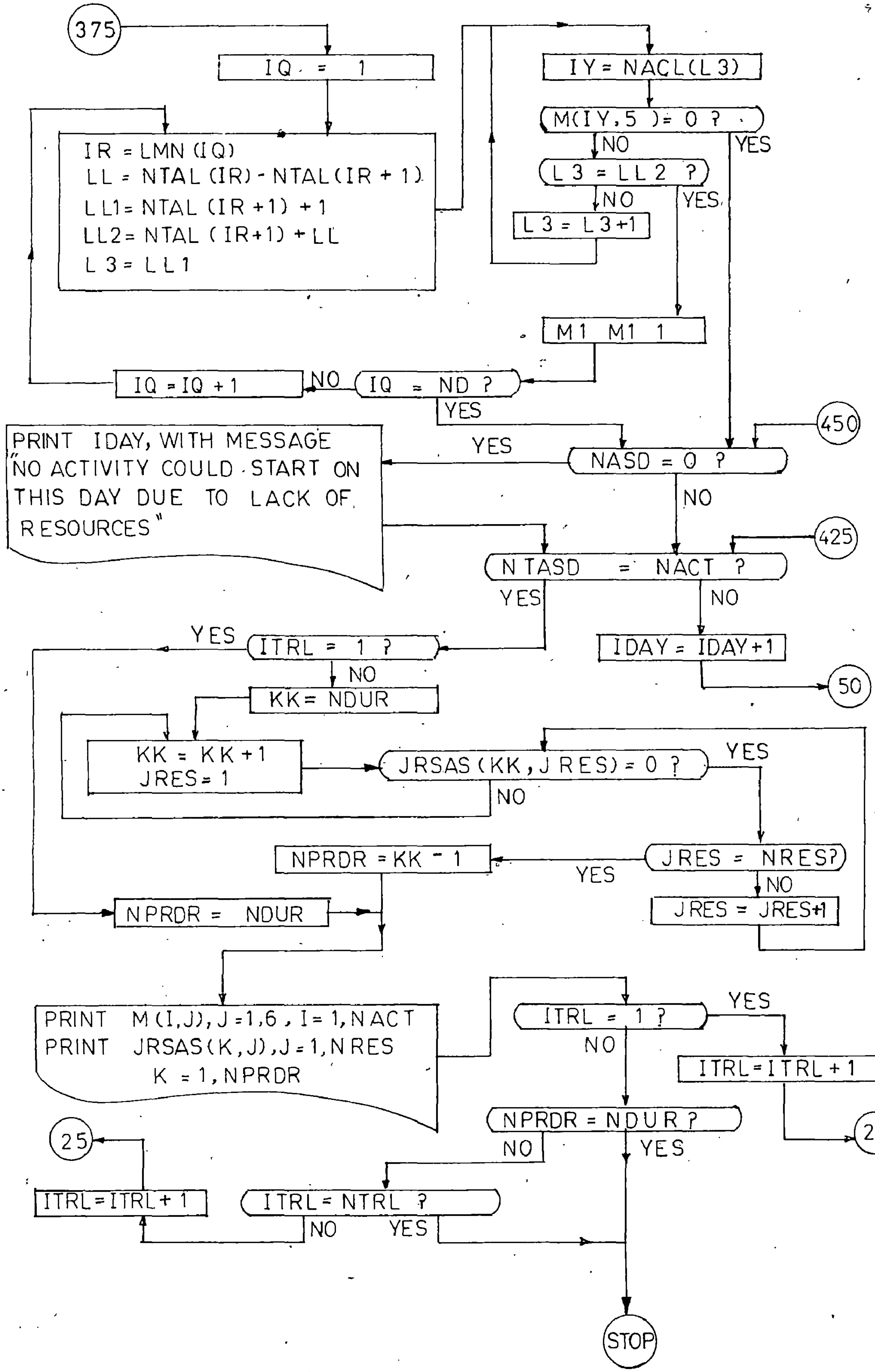
FLOW CHART FOR RESOURCE SCHEDULING ALGORITHM


100
L = MM
LK = NDARY (L)
IEOT (LK) ≥ IDAY ?
NO
YES
200
ND = ND + 1
LMN(ND) = LK
LSTAT (LK) = 1 ?
YES
NO
ITRL = 1 ?
NO
YES
LE = NTAE(LK) - NTAE (LK-1)
LE1 = NTAE (LK-1) + 1
LE2 = NTAE(LK-1) + LE
L1 = LE1
IM = NACE(LE1)
M(IM, 5) > 0 AND < IDAY ?
NO
M(IM,3) = 0 ?
NO
YES
YES
150
M(IM,4) = 0 ?
NO
YES
150
L1 = LE2 ?
NO
L1 = L1+1
YES
LSTAT (LK) = 1
LL = NTAL(LK) - NTAL(LK+1)
LL1 = NTAL (LK+1) + 1
LL2 = NTAL (LK+1) + LL
L2 = LL1
M(IN,5) = 0 ?
IN = NACL (L2)
L2 = L2 + 1
YES
NO
ICT = ICT + 1
ISEL(ICT) = IN
ITEM(ICT) = M(IN, 6)
M(IN ,4) = IDAY
L2 = LL2 ?
NO
YES
150
L = NODL ?
NO
YES
200
L = L + 1


200
ICT = 0 ?
YES
PRINT IDAY AND MESSAGE" NO ACTIVITY ELIGIBLE TO START TODAY"
425
NO
ITRL = 1 ?
NO
YES
IACT = 1
ICT > 1 ?
250
NO
YES
IPLAC = ICT-1
225
IP = ISEL(IACT)
II = 1
M(IP, 3) = 0 ?
NO
YES
325
275
III = II+1
LLL = III
ITEM(II) = ITEM(LLL) ?
ITEM(II) > ITEM(LLL) ?
ITEM(II) < ITEM(LLL) ?
YES
YES
YES
IT = ISEL (II)
ITI = ISEL (LLL)
IT = ISEL (II)
ITI = ITEM (II)
ISEL (II) = ISEL (LLL)
ITEM(II) = ITEM (LLL)
ISEL (LLL) = IT
ITEM(LLL) = ITI
M(IT, 3) > M(ITI, 3)?
NO
YES
ISEL(II) = ITI
ISEL(LLL) = IT
LLL = LLL+1
LLL = ICT ?
NO
YES
250
II = IPLAC ?
NO
YES
PRINT IDAY AND ISEL(I) I = 1, ICT
II = II+1
300



300
350
IACT = 1
IP = ISEL(IACT)
M(IP, 3) = 0 ?
NO
YES
325
JLACK = 0
JRES = 1
JRSAS(IDAY, JRES) + JRSRQ (IP, JRES)
≤ JRSAV(ITRL, JRES) ?
NO
YES
JRES = JRES + 1
M(IP, 6) ≥ IDAY ?
NO
YES
400
NO
JRES = NRES ?
YES
JLACK = JLACK+1
JLSN(JLACK)=JRES
JLACK = 0 ?
NO
YES
PRINT IDAY, IP,
JLSN(J), J = 1, JLACK
400
275
K1 = IDAY
KDUR = K1 + M(IP, 3)-1
KDAY = K1
JRES = 1
JRSAS(KDAY, JRES) = JRSAS(KDAY, JRES)+JRSRQ(IP, JRES)
KDAY = KDAY+1
NO
KDAY = KDUR ?
YES
YES
JRES = NRES ?
NO
JRES = JRES+1
325
NASD = NASD + 1
M(IP, 4)=IDAY - 1
M(IP, 5) = M(IP, 4)
+M(IP, 3)
NTASD = NTASD + 1
400
IACT = ICT ?
YES
NO
IACT = IACT + 1
ITRL = 1 ?
YES
NO
ND = 0 ?
NO
YES
225
350
375
450


375
IQ = 1
IR = LMN (IQ)
LL = NTAL (IR) - NTAL (IR + 1)
LL1 = NTAL (IR + 1) + 1
LL2 = NTAL (IR+1) + LL
L 3 = LL1
IY = NACL(L 3)
M(IY, 5) = 0 ?
NO
YES
L 3 = LL 2 ?
NO
YES
L 3 = L 3+1
M1 M1 1
IQ = IQ + 1
NO
IQ = ND ?
YES
450
PRINT IDAY, WITH MESSAGE "NO ACTIVITY COULD START ON THIS DAY DUE TO LACK OF RESOURCES"
YES
NASD = 0 ?
NO
425
N TASD = NACT ?
YES
NO
YES
ITRL = 1 ?
NO
IDAY = IDAY+1
50
KK = NDUR
KK = KK + 1
JRES = 1
JRSAS (KK, J RES) = 0 ?
YES
NO
NPRDR = KK - 1
YES
JRES = NRES?
NO
NPRDR = NDUR
JRES = JRES+1
PRINT M (I,J), J = 1,6, I = 1, N ACT
PRINT JRSAS(K,J), J = 1, N RES
K = 1, NPRDR
ITRL = 1 ?
YES
NO
ITRL = ITRL + 1
NPRDR = NDUR ?
NO
YES
25
25
ITRL = ITRL + 1
ITRL = NTRL ?
NO
YES
STOP

LIMITATIONS OF THE PROGRAM AND SUGGESTIONS FOR IMPROVEMENT

1. The program initialises the resource assigned, on each day of the project as zero, at the commencement of each trial. But it is not known before hand, how long the project will last. The program at present initialises for an estimated duration which is one and a half times the normal duration, (assuming that the project will not be delayed beyond this). But this procedure has 2 weaknesses.

   Firstly if the resource availability level of a particular trial is indeed very low, the duration may extend to more than 1.5 times the normal duration, and in the absence of initialisation there will be an execution error message in the output.

   Secondly for these trials, in which the duration ofe the project is extended only marginally, this initialisation takes up unnecessary space in the memory which is not used.

2. The program checks an activity for eligibility and then checks for availability of resources. If these conditions are satisfied the algorithm ~~strength~~ straightaway assigns all the resource requirements of this activity for its entire duration. This may seem questionable. It would be better to allow the splitting of the activity on some future day if the need arises, and assign resources day by day.

3. The program prepares a list of eligible activities on each day and arranges them in priority order, for assigning resources.

Consider the following situation -

The first activity in this list is critical and cannot start on the day because the resource requirements marginally exceed the availability. The second activity has sufficient slack and its resource requirementd do not exceed the availability.

The program, as presently written schedules the second activity, and the only consolation is that the critical activity is given top priority next day.

A better way to deal with the situation would be to borrow men from an in-progress activity which has sufficient slack and schedule the critical activity first. The program will have to be modified to accomplish this.

4 The program at present does nothing about unassigned resources on each day. If the suggestion offered earlier is incorporated these unassigned resources can be reduced. Another way of overcoming this problem is to specify the resource availabilities in two parts i. On a regular basis and ii on an overtime basis. As a last resort the unassigned resources may be assigned to the inprogress-activities, on the basis of least slack first, since these resources may have to be paid for whether they are used or not.

LIST OF SYMBOLS USED IN THE RESOURCE SCHEDULING PROGRAM

(note- Those symbols already explained in the CPM Scheduling program are not explained again in this list).

IACT	Counter for iterating an eligible activities
ICT	Number of activities eligible to start on a particular day.
IDAY	The day under consideration
IDUR	An initial 'guessimate' of the new project duration.
IP	Serial number of the eligible activity which is being tested for resource availability and to which resources will be assigned if available.
IPLAC	Defined as ICT-1 . This has no physical signi ficance. It is used for rearranging the activity serial numbers in ascending order of their latest start times.
ISEL	A single-subscripted array that stores the serial numbers of the activities eligible to start on any day.
ITEM	A single subscripted array that stores the latest start times of the activities eligible to start on any dayISEL and ITEM are parallel arrays of maximum dimension equal to ICT. While ISEL stores the serial numbers of the activities ITEM stores the corres-ponding latest start times.

ITRL	Counter for iterating on trials.
JLACK	Conнter for counting the number of resource types found lacking and thus blocking the start of a critical activity.
JLS N	A single subscripted array that stores serial numbers of the resources (thus identifying them) found lacking and thus blocking the start of a critical activity. JLACK AND JLSN Are used in conjunction and they appear in the resource scheduling progress report, in the output.
JRSAV	A double-subscripted array that stores the availabilities of each res urce, in each trial If the need arises JRSAV can be made to store the resource availabilities for each day.
JRSRQ	A double-subscripted array that stores the resource requirements of each activity, of each res urce.
K1 and KDUR	Counters indicating the first and last day, of the period for which resource ass ignments are made to an activity.
LSTAT	A single subscripted array of maximum dimension equal to NODE . This stores the status indicator for each node. A status "One" indicates that all activities entering the node have been completed, and hence all activities leaving

the node are eligible to start.

A status 'zero' indicates that a check will be necessary . It is not sure yet, whether all activities entering the node have been completed.

LMN A single subscripted array for storing the nodes found eligible on each day. These nodes will be examined later to see whether all activities leaving them have started, in which case the counter M1 will be updated.

LE
LE1
LE2 Variables used in identifying the serial numbers of activities entering a node (see NTAE and NACE for details)

LL
LL1
LL2 Variables used in identifying the serial numbers of activities leaving a node. See NTAL and NACL for details.

M A double subscripted array, similar to the one used in the CPM scheduling program, but with the following changes. The number of columns is only 6. The first three columns store the I-node, J-node and duration of the activities. The 4th and 5th columns store the revised start and finish times of the activities. The 6th column is the same as in the case of the CPM scheduling program. It stores the latest start time of the previous CPM schedule . The number of rows in M is equal to NACT. At the start of a new trial the 4th and 5th columns of M are set equal to zero. These act as status indicators for the activity. In the output they have the usual meaning (start and finish times).

M1 and MM	Counters used to reduce the range for scanning of nodes for eligibility, on each day.
NRES	Total number of resources considered
NTRL	Total number of trials considered.
NT	The trial number with which the program is to start. Normally NT is equal to one, and it is read as data. However if in future it is desired to start from the second or third trial NT can be made equal to 2 or 3.
NTASD	Total numberof Activities that have started so far. When NTASD is equal to NACT, the resource allocation procedure stops and the results are printed.
NASD	Number of activities that have started on a particular day.
ND	Counter for counting the eligible nodes for the day.
NPRDR	The new project duration.

CHAPTER - IV

CASE STUDY

Project: Renovation of Malikpur Hutments ,University of Roorkee	
Number of Activities	70
Number of Nodes	52
Normal Duration	63 working days
Resources cons idered	i. Beldars ii. Carpenters iii. Masons
Start Date of Project	5th Feb, 1975.
Finish Date (CPM Schedule)	24th April,1975
Normal Duration of Project (in calender days)	78 days
Number of holidays (excluding Sundays)	4
List of Holidays(other than Sundays)	19th Feb, 1975 26th Feb, 1975 27th March,1975 28th March,1975
Number of Re-arrangements	1
Criterion for Rearrangement	Activity Criticality
Number of Trials for Resource Levelling	3

Resource Availability Levels Considered

First Trial	i. 100 Beldars ii. 100 Carpenters iii. 100 Masons	(Unlimited Resources)
Second Trial	i. 16 Beldars ii. 4 Carpenters iii. 4 Masons	
Third Trial	i. 16 Beldars ii. 6 Carpenters iii. 4 Masons	

RESULTS OF RESOURCE LEVELLING

First Trial

a. Project duration 63 days (Normal duration)

b. Peak Resource Requirements

i. 60 Beldars

ii. 8. Carpenters

iii. 12 Masons.

Second Trial

a. Projec-t Duration = 76 days

b. Peak Resource Requirements

i. 16 Beldars
ii. 4 Carpenters
iii. 4 Masons.

Third Trial

a. Project Duration 69 days

b. Peak Resource Requirements

i. 16 Beldars
ii. 6 Carpenters
iii. 3 Masons

DESCRIPTION OF CASE STUDY

To illustrate the working of the program a case study has been presented. The size of the problem has been limited because the object is only to show that the program achieves the desired results. The program is capable of handling even bigger projects and the upper limit will depend on the memory of the computer that is used.

The case study that is cons-idered is the Hutment Renovation projec t that is in progress at the moment, in the University campus.

The reasons for choosing this project are :

i. It is of a size, just sufficient to illustrate the problem - big enough to test all aspects of the program and small enough to keep the computer expenses low. (The Computer used is the IBM 1620 at SERC -Roorkee which is handicapped by the absence of an on-line printer. Choosing a big project would have resulted in a large output of punched cards without any particular advantage.

ii. The actual activity data is readily available as it is a live problem in progress.

DETAILS OF THE PROJECT

Introduction

The renovation of a single hutment involves 35 activities. Two such hutments are considered, thus bringing, the total number of activities to 70. Of course more hutments could be considered but two are sufficient for illustration.

Resources

The resources considered are primarily only three viz., 1. Skilled labourers ('Beldars')

2. Carpenters

3. Masons.

Other resources like wireman, painter, truck, etc, are not considered because very few activities require them and it is being assumed that they are available when required. The demand for Beldars, Carpenters and Masons is heavy and these are considered for the resource scheduling program.

Data

The data for the case study was obtained by visiting the site and consulting the S.U.E., the overseer in charge, and also the contractor's supervisor at the site.

The data supplied included a list of activities, their durations and resource requirements, for one hutment. This data for one hutment is listed in Table 1 which includes the node numbers assigned to the activity, a description of the activity, an assigned trade indica tor number and the requirements of Beldars, carpenters and Masons.

Arrow Diagram

The Arrow diagram for the project has been attached. This diagram is for two such hutments, on which work could proceed simultaneously. More hutments could have been considered.

The arrow diagram consists of 52 nodes and 70 activities including 6 dummies. The rules of network logic have been kept in mind while drawing up the network and care has been taken to avoid wagon wheel and waterfall errors.

For the sake of convenience, and for ease in book keeping, the arrow diagram has been drawn so that activities specified by even node numbers refer to one hutment and odd node numbers refer to the corresponding activity for the second hutment.For example activity 2-4 refers to 'Disconnect water supply" for the first hutment and activity 3-5 refers to the same for the second hutment. The earliest and latest occurrence time for each node is also shown in the arrow diagram and it is entered in the circles representing the nodes. The activity description and the duration appears alongside the arrow representing the activity.

Trade Indicator -

The trade indicator numbers have the meanings listed in Table 2.

TABLE - 2

Trade Indicator	Corresponding Trade.
0	Dummy Activity
1	Carpentry
2	Brickwork/Plastering
3	Concreting
4	Reinforcement
5	Painting/White washing
6.	Services(Water, Electricity)
7	Demolition
8.	General(Movingin.clearing out etc.)

TABLE 3

Resource No.	Resource Description
1	Beldars
2	Carpenters
3	Masons

Results

The results for the CPM scheduling program and the resource levelling program are given in Appendix 'B'.

The activities have been listed in the order in which they were fed in the data card. But a re-arrangement in ascending order of total float has also been given . The earliest and latest occurrence times of the nodes have been printed, and these values have been entered in the arrow diagram also . The output also gives the arrays NTAE,NACE, NTAL and NAC L, which will be used in the resource scheduling program. The duration of the project according to the scheduling computations is 63 days.

The array NDARY has been printed which gives the nodes in ascending order of their early occurrence times.

CPM Calender

The calender dated schedule is also presented. The project start date has been assumed to be 5th Feb, 1975. All Sundays have been considered holidays . Further a list of four holidays (other than sundays) has been read as data viz 19th February, 26th February, 27th March and 28th March 1975. The results show the early and late start and finish dates. The CPM schedule gives a projec t completion date of 24th April, 1975.

Resource Scheduling

The three resources have been numbered as shown in Table 3. The three trials experimented with three different levels of resource availability.

First Trial

The first trial has been carried out with a resource availability of 100 Beldars 100 carpenters and 100 masons. This is from a practical view point an unlimited resource level (for this project at least) , and, as expected this has produced an all-early start schedule, without changing the project duration.

The activities have been listed with their revised early start and early finish times, and the original latest start time, and also the resource requirements of Beldars, Carpenters and Masons. This trial shows a peak requirement of 60 Beldars, 8 carp enters and 12 masons.

Second Trial

The second trial was carried out with reduced resource availability of 16 Beldars, 4 carpenters and 4 masons. This has produced a revised schedule and the new project duration is 76 days. The daily resource requirements are also within the availability limits.

Third Trial

This trial was carried out with the availabilities of Beldars and masons unchanged (16 and 4 respectively). But an increase of 2 was made in the case of carpenters. This has resulted in a new schedule of start and finish dates. The project duration has been brought down from 76 days to 69 days.

A histogram has been plotted for each resource and it shows the variation of the daily requirements of the resource in different trials.

TABLE 1

DATA FOR ONE HUTMENT RENOVATION

I Node	J Node	Description of Activity	Duration	Trade Indicator	Resources required		
					Beldars	Carpenters	Massons
1	2	Move in with tools, materials and equipment	1	8	5	0	0
2	4	Disconnect water supply	1	6	4	0	0
2	6	Remove wiring, fans, light points etc	1	6	4	0	0
4	6	Dummy	0	0	0	0	0
2	18	Cutting and Bending Reinforcement bars	2	4	2	0	0
6	8	Dismantle asbestos ceiling	3	7	8	2	0
8	10	Take down asbestos Roofing	4	7	4	4	0
10	12	Take down wooden truss	3	7	4	1	0
12	14	Dismantle Brich work	2	7	4	0	1
14	16	Fresh Brick work	3	2	14	0	2
16	18	Centering and s huttering for slab	7	1	2	3	0
18	20	Laying Reinforcement - Phase I	1	4	2	0	0
18	22	Laying Reinforcement - Phase II	1	4	2	0	0
18	24	Laying Reinforcement - Phase III	1	4	2	0	0

I Node	J Node	Description of AGtivity	Duration	Trade Indicator	Resources required		
					Beldars	Carp.	Massons
20	26	Concreting - Phase I	1	3	10	0	2
22	28	Concreting - Phase II	1	3	10	0	2
24	30	Concreting - Phase III	1	3	10	0	2
26	32	Curing Phase I	21	3	1	0	0
28	32	Curing Phase II	21	3	1	0	0
30	32	Curing Phas e III	21	3	1	0	0
32	34	Prepare wooden frame for verandah roof	2	1	1	0	0
32	36	Remove shuttering	4	1	2	2	0
34	38	Lay A.C. Sheet for verandah roof	2	1	1	1	0
36	38	Repairs to wall plaster	2	2	1	0	1
36	40	Dummy	0	0	0	0	0
36	44	Carpentry Repairs to doors and windows	4	1	2	1	0
36	50	Repairs to flooring	2	2	1	0	1
38	42	White Washing	1	5	2	0	0
40	50	Fit pipes, taps and restore water supply	1	6	4	0	0
42	46	Color washing	1	5	2	0	0
44	46	Replace broken glass panes	2	1	1	1	0
46	48	Install fans, lightpoints wiring etc	3	6	4	0	0
46	50	Paint doors and windows	4	5	2	0	0
48	50	Dummy	0	0	0	0	0
50	52	Clean, Clear up and move out	2	8	8	0	0

DISCUSSION OF RESULTS

CPM Scheduling Results

The most important point to be noted is the project duration. The program has shown that the renovation of two hutments can proceed simultaneously and they can be finished in 63 working days, i.e., in about 2 month's time. However , the contractor has been taking 3 months for one hutment, as enquiries show. This is obviously because the contractor is not keen on speeding up the job.If he were keen, CPM shows that 63 days is sufficient for 2 hutments, without resource constraints.

RESOURCES SCHEDULING RESULTS

First Trial

The CPM scheduling results are a good basis for comparison. The CPM schedule has given the shortest project-duration of 63 working days. However, it has also resulted in peaking of resource requirements. The critical activities of concreting, requiring 10 Beldars and 2 masons each have all been scheduled to start on the same day and this has given rise to a requirement of 60 Beldars and 12 masons, on the 26th day of the project(See histogram). This trial also yields a peak requirement of 8 Carpenters on the sixth, seventh, eighth and ninth days.

Providing 60 Beldars, 8 carpenters and 12 masons would be utter waste of resources, particularly in the case of masons and carpenters because, from the histogram it can be seen that the masons and carpenters would have to remain idle for considerable periods. In the first trial the carpenters are not required on 36 days out of a project duration of 63 days, and the Masons are idle for 55 out of 63 days. This is a very unfavourable situation, and it clearly demonstrates the fact that a resources levelling process is essential.

It can be clearly seen that a levelling of resources within the project duration of 63 days is IMPOSSIBLE in the project. This is because the activities that have jointly given rise to the peak resource requirements, are critical and they cannot be postponed without delaying the project.

This point clearly vindicates the approach to the resource scheduling algorithm adopted in this dissertation- viz. of considering an increase in project duration if the peak can be brought down significantly

Second Trial

The results of the second trial have illustrated the merits of the program. A very moderate resource availability of 16 Beldars, 5 carpenters and 4 masons has been considered, as against the peak demands of 60 Beldars, 8 carpenters and 12 masons.

This has given a revised schedule of start and finish times, in which some critical activities have been postponed. The concreting activities have been rescheduled so that only one of them can be scheduled on any day. This has given a much more favourable histogram of resource requirements. The peak demand of 60 Beldars, on the 26th day in the first trial has been chopped down to the available level of 16. Another peak demand of 20 Beldars in the first trial, on the second and third days, has been levelled to 16 Beldars.

The projec t duration has of course been increased by 16 days, and the new project duration is 76 days as against 63 days in the first trial. But the advantages gained by this increase are considerable. In addition to bringing the requirement level of Beldars from 60 to 16, the schedule makes better use of carpenters and masons. The peak demands of 8 cappenters and 12 masons have been brought down to 4 carpenters and 4 masons . Moreover the carpenters and masons are idle on far fewer days than in the schedule produced by the first trial. In the second trial the carpenters are not required on 31 out of 76 days (i.e. 40.08%) as against 36 out of 63 days (i.e. 57%) in the first trial. The masons are not required on 60 out of 76 days (79%) as against 55 out of 63 days (87.3%) in the first trial. The "idle time" has been significantly reduced.

Trial Three

The progress report for the 2nd trial shows that almost all significant postponements have been caused by the lack of resource number two (viz. carpenters). In the third trial the number of carpenters available was increased from 4 to 6, keeping the availability of Beldars and masons unchanged.

This trial has reduced the project duration significantly. From 76 days in the second trial it has been brought down to 69 days. The requirement levels of carpenters and Beldars have been maintained at the availability levels of 6 and 16 respectively. But the requirement level of masons has been reduced to only three even though four are available. This is a very interesting development. It shows that the levelling of resources does not stop at the level of availability that is specified. If possible the requirements are brought down to even below the availability level . This is one of the merits of the program.

SUMMARY

The program has given the user 3 options

1. An all-early -start schedule, with a project duration of 63 days and a peak resource demand of 60 Beldars, 8 carpenters and 12 masons.
2. A revised schedule and a project duration of 76 days, with a maximum resource demand of only 16 beldars, 4 carpenters and 4 masons.
3. Another revised schedule with a project duration of 69 days and a maximum resource demand of 16 beldars, 6 carpenters and 3 masons.

If one makes a cost study, it will be easy to see which of the three schedules is the most favourable.

Without a cost study one can only guess. However, in this problem, trial 3 appears to yield the optimum duration . By providing 2 more carpenters and partly compensating for this increase by providing one mason less, the duration of 76 days in trial II can be brought down to 69 days . This is a significant improvement in the schedule.

REFERENCES

1. Puri N. and G. Vishwanath - "Computerised Scheduling and Resource Levelling of CPM/PERT Networks" Proc. of the National Symposium on "Planning, Design and Construction of Industrial Complexes" Sponsored by BHEL, Institution of Engineers etc, held at Tiruchirapalli (Jan, 1975)
2. Paracer, A.P.," Project Management through PERT/CPM" CPWD Journal Vol.II, October, 1973.
3. Mehra (Mrs) S. " Cost Control with Critical Path Method" Special Problem, Civil Dept., University of Roorkee. October, 1974.
4. Clark C.E., "The Optimum allocation of Resources among the ac-tivities of a network" Journal of Industrial Engineering, Jan-Feb, 1961.
5. Burgess A.R. and J.B. Killebrew , " Variation in Activity Level on a cyclical Arrow diagram" Journal of Industrial Engineering, Vol. 13 No. 2, March-April 1962.
6. Mc Gee A.A. and Markarian M.D., " Optimum allocation of Research Engineering Manpower within a Multiproject Organisational structure" IEEE Trans. on Engineering Management, September, 1962.
7. Kelley J.E., " Scheduling Activities to satisfy Constraints on Resources", Industrial Scheduling, Muth and Thompson, Editors, 1963.
8. Moshman J. Johnson, J. and Larsen M., " RAMPS, A technique for Resource Allocation and Multiproject Scheduling" Proc., 1963 Spring Joint Computer Conference.

9. Mize J.H., " A Heuristic Scheduling Model for Multiproject Organisations" Unpublished Doctoral Dissertation, Purdue University, August, 1964.

10. Conway R.W., " An experimental investigation of Priority Assignment in a job shop", Memorandum RM-3789-PR The RAND Corporation, February 1964.

11. Galbraith Robert V."Computer Program for Levelling of resource Usage",Jr.of Const.Div.Proc.ASCE,May,1965.

12. Davis, Edward W., " An Exact Algorithm for the Multiple Constrained- Resource Projec t Scheduling Problem",Ph.D. Dissertation, Yale University, May, 1968.

13 Wiest Jerome D. " Computer Models for Scheduling of Large Projects" Ph.D. Dissertation, Carnegie Institute of Technology, Sept, 1964.

14. Fendley Larry G, " Toward the development of a Complete Multi Projec t Scheduling system" Journal of Industrial Engineering, Vol. 19, No.10, Oct, 1968 pp505-515.

15. Bennet Lawrence , F. "Critical Path Resource Scheduling Algorithm", Jr. of Const. Div., Proc. ASCE, October, 1968.

16. Davis, Edward W., "Resource allocation in projec-t network models - A Survey" Journal of Industrial Engineering, Vol.XVII No.4, April, 1966, pp177-188.

17. Moder Joseph J. and Phillips, Cecil R., " Project Management with CPM and PERT" Van Nostrand Reinhold Company, Second Edition, 1970.

18. O'Brien" James O., " CPM in Construction Management" Mc Graw Hill 1965.

19. CPM in Construction - A Manual for General Contractors, Prepared and Published by The Associated General Contractors of America , 1965.

APPENDIX-A

COMPUTER PROGRAM

```
C     CPM SCHEDULING  G.VISHWANATH M.E(THESIS)
      DIMENSION M(100,9),IEOT(75),LOT(75),NDARY(75),NTAL(75),NTAE(75)
      DIMENSION NACE(100),NACL(100),JTEM(100),ISNO(100),NRENG(10)
      READ 7,NPROB
    7 FORMAT(I5)
      IPROB=1
    1 READ 8,NODE,NACT,NUMBC,ISKIP,NUMRE
    8 FORMAT(16I5)
      N=NUMBC-5
      IF(N-3)33,36,33
   33 CONTINUE
      READ 8,((M(I,J),J=1,N),I=1,NACT)
      PUNCH8,((M(I,J),J=1,N),I=1,NACT)
      DO  35 I=1,NACT
   35 M(I,9)=M(I,4)
      GO TO 42
   36 CONTINUE
      READ 9,((M(I,J),J=1,N),I=1,NACT)
      PUNCH9,((M(I,J),J=1,N),I=1,NACT)
    9 FORMAT(15I5)
   42 CONTINUE
      IF(NUMRE)44,44,43
   43 CONTINUE
      READ 18,(NRENG(I),I=1,NUMRE)
      PUNCH 18,(NRENG(I),I=1,NUMRE)
   18 FORMAT(5I5)
   44 CONTINUE
      PUNCH 136
  136 FORMAT(37HTHE QUANTITIES PRINTED ABOVE ARE DATA)
```

```
C     CALCULATION OF EARLIEST OCCURENCE TIMES

      IEOT(1)=0
      NTACT=0
      NTAE(1)=0
      DO 41 K=2,NODE
      MAX=0
      DO 40 I=1,NACT
      II=M(I,1)
      JJ=M(I,2)

C     INTRODUCE ERROR CHECK 1

      IF (II-JJ)25,26,27
   26 PUNCH 28,II,IPROB
   28 FORMAT(4HNODE,I4,22H DUPLICATED IN PROBLEM,I3)
      GO TO 1000
   27 PUNCH 29,I,II,JJ,IPROB
   29 FORMAT(9HCHECK ACT,I3,10H AND NODES,I3,1H,,I3,35H FOR POSSIBLE LOO
     9PING ERROR IN PROB,I2)
      GO TO 1000
   25 CONTINUE
      IF(JJ-K)40,37,40
   37 NTACT=NTACT+1
      NACE(NTACT)=I
      IET=IEOT(II)+M(I,3)
      IF(MAX-IET)38,40,40
   38 MAX=IET
   40 CONTINUE
      NTAE(K)=NTACT

C     INTRODUCE ERROR CHECK 3

      IF(NTACT-NTAE(K-1))21,21,22
   21 PUNCH 23,K,IPROB
   23 FORMAT(40HDANGLING ERROR OF SECOND TYPE.REFER NODE,I3,8H IN PROB,I
     93)
      GO TO 1000
   22 CONTINUE
      IEOT(K)=MAX
   41 CONTINUE

C     INTRODUCE ERROR CHECK 4

      IF(NTACT-NACT)12,13,12
   12 PUNCH 15,IPROB
   15 FORMAT(35HACTIVITY LIST INCOMPLETE IN PROBLEM,I3)
      GO TO 1000
   13 CONTINUE
```

```
      IF (ISKIP-1)99,102,99
  102 PUNCH 34
   34 FORMAT(43HPRINTED BELOW ARE THE VALUES NACE,NTAE,IEOT)
      PUNCH 9,(NACE(I),I=1,NACT)
      PUNCH 9,(NTAE(I),I=1,NODE)
      PUNCH 9,(IEOT(I),I=1,NODE)
   99 CONTINUE
C     CALCULATION OF LATEST OCCURENCE TIMES
      NTACT=0
      LOT(NODE)=IEOT(NODE)
      NTAL(NODE)=0
      NODL=NODE-1
      DO 51 K1=1,NODL
      K=NODE-K1
      MIN=IEOT(NODE)
      DO 50 I=1,NACT
      II=M(I,1)
      IF(II-K)50,47,50
   47 NTACT=NTACT+1
      NACL(NTACT)=I
      JJ=M(I,2)
      LT = LOT(JJ)-M(I,3)
      IF(MIN-LT)50,50,78
   78 MIN=LT
   50 CONTINUE
C     INTRODUCE ERROR CHECK 2
      IF(NTACT-NTAL(K+1))53,53,52
   53 PUNCH 54,K,IPROB
   54 FORMAT(40H DANGLING ERROR OF FIRST TYPE.REFER NODE,I3,8H IN PROB,I
     93)
      GO TO 1000
   52 CONTINUE
      NTAL(K)=NTACT
      LOT(K)=MIN
   51 CONTINUE
      IF(ISKIP-1)101,103,101
  103 PUNCH 133
  133 FORMAT(42HPRINTED BELOW ARE THE VALUES NACL,NTAL,LOT)
      PUNCH 9,(NACL(I),I=1,NACT)
      PUNCH 9,(NTAL(I),I=1,NODE)
      PUNCH 9,(LOT(I),I=1,NODE)
  101 CONTINUE
```

```
C     CALCULATION OF EST,LST,EFT,LFT,TOTAL FLOAT
      DO 61 I=1,NACT
      II=M(I,1)
      JJ=M(I,2)
      M(I,4)=IEOT(II)
      M(I,5)=M(I,4)+M(I,3)
      M(I,7)=LOT(JJ)
      M(I,6)=LOT(JJ)-M(I,3)
      M(I,8)=M(I,6)-M(I,4)
   61 CONTINUE
      PUNCH 302
  302 FORMAT(80(1H*))
      PUNCH 153
      PUNCH 302
  153 FORMAT(//,49H    I-NODE   J-NODE DURAT.     EST       EFT       LST,5X,
     C 20HLFT T.FLOAT   TRADE )
      DO 65 I=1,NACT
   65 PUNCH 154,  (M(I,J),J=1,NUMBC)
      PUNCH 302
  154  FORMAT(3X,9(2X,I3,2X,1H*))
      NACTL=NACT-1
      IF (NUMRE)312,312,201
  201 CONTINUE
C     BEGIN REARRANGEMENT OF ACTIVITIES
      DO 301 KI=1,NUMRE
      NCOL=NRENG(KI)
      DO 221 I=1,NACT
      ISNO(I)=I
  221 JTEM(I)=M(I,NCOL)
      DO 241 I=1,NACTL
      II=I+1
      DO 241 K=II,NACT
      IF (JTEM(I)-JTEM(K)) 241,241,231
  231 IT=ISNO(I)
      ITE=JTEM(I)
      ISNO(I)=ISNO(K)
      JTEM(I)=JTEM(K)
      JTEM(K)=ITE
      ISNO(K)=IT
  241 CONTINUE
      PUNCH 251, NCOL
  251 FORMAT(/,8H COLUMN=,I4,5X,9HASCENDING )
      PUNCH 302
      PUNCH 153
      PUNCH 302
      DO 261 II=1,NACT
      I=ISNO(II)
  261 PUNCH 154,(M(I,J),J=1,NUMBC)
      PUNCH 302
  301 CONTINUE
  312 CONTINUE
```

```
      IF(ISKIP-1)1000,313,1000
  313 CONTINUE
C     REARRANGEMENT OF NODES IN ASCENDING ORDER OF E.O.T

      DO 900 KK=1,NODE
      JTEM(KK)=IEOT(KK)
      NDARY(KK)=KK
  900 CONTINUE
      DO 901 I=1,NODL
      II=I+1
      DO 901 KK=II,NODE
      IF(JTEM(I)-JTEM(KK))901,901,902
  902 JTK=JTEM(I)
      NTK=NDARY(I)
      JTEM(I)=JTEM(KK)
      NDARY(I)=NDARY(KK)
      JTEM(KK)=JTK
      NDARY(KK)=NTK
  901 CONTINUE
      NDUR =IEOT(NODE)
      PUNCH 903,NDUR
  903 FORMAT(24HNORMAL PROJECT DURATION=,I4)
      PUNCH 905,(NDARY(I),I=1,NODE)
  905 FORMAT(16I5)
 1000 IF(IPROB-NPROB)998,999,999
  998 IPROB=IPROB+1
      GO TO 1
  999 CONTINUE
      STOP
      END
```

PROGRAM FOR CALENDER DATING OF CPM SCHEDULE

```
C   C   CALENDER
        DIMENSION NDM(20),IHDT(10),IHMN(10),M(75,7),NDATE(4)
        DIMENSION NMON(4),NYEAR(4),IHOL(10)
        READ 297,NACT,IDATE,IMON,IYEAR,NHOL,NNN
        READ 298,(IHDT(I),IHMN(I),I=1,NHOL)
        READ 299,((M(I,J),J=1,7),I=1,NACT)
        READ 300,(NDM(I),I=1,20)
    297 FORMAT(6I5)
    298 FORMAT(8(I2,I3,5X))
    299 FORMAT(3X,7(2X,I3,3X))
    300 FORMAT(20I4)
        DO 308 I=1,NHOL
        IHOL(I)=0
        LL=IMON
        JJ=IDATE
    302 IF(IHMN(I)-LL)305,305,303
    303 IHOL(I)=IHOL(I)+NDM(LL)-JJ
        LL=LL+1
        JJ=0
        GO TO 302
    305 IHOL(I)=IHOL(I)+IHDT(I)-JJ
    308 CONTINUE
        PUNCH 309,(IHOL(I),I=1,NHOL)
    309 FORMAT(/,15I5)
        I=1
    311 J=1
    312 M1=1
        NSUN=0
        KJ=J+3
        NDIF=M(I,KJ)
        IF(NDIF)314,313,314
    313 NDATE(J)=IDATE
        NMON(J)=IMON
        NYEAR(J)=IYEAR
        GO TO 371
    314 CONTINUE
        NWEEK=NDIF+NNN
    315 MM=M1
        IREG=0
        IF(MM-NHOL)317,317,337
```

```
317 DO 335 II=MM,NHOL
    IF(NDIF-IHOL(II))337,330,330
330 NDIF=NDIF+1
    NWEEK=NWEEK+1
    IREG=IREG+1
    M1=II+1
335 CONTINUE
337 IF(IREG)345,340,345
340 IF(NSUN)360,347,360
345 NSUN=0
347 NSUN=NSUN+NWEEK/6
    NWEEK=NWEEK-6*NSUN
    IF(NWEEK)354,355,354
355 NWEEK=6
    NSUN=NSUN-1
354 NDIF=NDIF+NSUN
    IF(NSUN)315,360,315
360 NDATE(J)=IDATE+NDIF
    LL=IMON
    NYEAR(J)=IYEAR
363 IF(NDATE(J)-NDM(LL))370,370,365
365 NDATE(J)=NDATE(J)-NDM(LL)
    LL=LL+1
    GO TO 363
370 IF(LL-12)369,369,368
368 LL=LL-12
    NYEAR(J)=NYEAR(J)+1
    GO TO 370
369 NMON(J)=LL
371 IF(J-4)372,374,374
372 J=J+1
    GO TO 312
374 PUNCH 375,(M(I,J),J=1,3),(NDATE(J),NMON(J),NYEAR(J),J=1,4)
375 FORMAT(3I8,4(2X,I2,1H/,I2,1H/,I2))
    IF(I-NACT)377,378,378
377 I=I+1
    GO TO 311
378 CONTINUE
    STOP
    END
```

RESOURCE LEVELLING PROGRAM

```
C     RESOURCE LEVELLING    G.VISHWANATH  M.E(THESIS)
      DIMENSION M(80,6),NTAE(60),NTAL(60),NACE(80),NACL(80),IEOT(60)
      DIMENSION NDARY(60),LSTAT(60),LMN(40),ISEL(60),ITEM(60),JLSN(8)
      DIMENSION JRSRQ(80,6),JRSAV(5,6),JRSAS(120,6)
      READ 450,NODE,NACT,NRES,NDUR,NTRL,NT
      READ 455, (NDARY(J),J=1,NODE)
      READ 455,(IEOT(J),J=1,NODE)
      READ 455,(NACL(I),I=1,NACT)
      READ 455,(NACE(I),I=1,NACT)
      READ 455,(NTAL(J),J=1,NODE)
      READ 455,(NTAE(J),J=1,NODE)
      READ 460,((M(I,J),J=1,6),I=1,NACT)
      READ 465,((JRSAV(I,J),J=1,NRES),I=1,NTRL)
      READ 470,((JRSRQ(I,J),J=1,NRES),I=1,NACT)
  460 FORMAT(6I5)
  450 FORMAT(6I5)
  455 FORMAT(16I5)
  465 FORMAT(16I5)
  470 FORMAT(15I5)
      ITRL=NT
  503 PUNCH 504, ITRL
  504 FORMAT(//,5HTRIAL,I3,2X,6HBEGINS)

C     INITIALISATION IN EACH TRIAL

      LSTAT(1)=1
      DO 490 J=2,NODE
      LSTAT(J)=0
  490 CONTINUE
      DO 505 I=1,NACT
      M(I,5)=0
      M(I,4)=0
  505 CONTINUE
      IDUR=NDUR+NDUR/2
      DO 510 K=1,IDUR
      DO 510 J=1,NRES
      JRSAS(K,J)=0
  510 CONTINUE
      M1=1
      NTASD=0
      IDAY=1

C     INITIALISATION ON EACH DAY

  512 NASD=0
      ICT=0
      MM=M1
      ND=0
      NODL=NODE-1
```

```
C     SELECTION OF ELIGIBLE ACTIVITIES

      DO 520 L=MM,NODL
      LK=NDARY(L)
      IF(IEOT(LK)-IDAY)521,522,522
  521 ND=ND+1
      LMN(ND)=LK
      IF(ITRL-1)511,523,511
  511 IF(LSTAT(LK)-1)524,523,524
  524 LE=NTAE(LK)-NTAE(LK-1)
      LE1=NTAE(LK-1)+1
      LE2=NTAE(LK-1)+LE
      DO 526 L1=LE1,LE2
      IM=NACE(L1)
      IF(M(IM,3))517,517,518
  517 IF(M(IM,4))520,520,526
  518 IF(M(IM,5))520,520,525
  525 IF(M(IM,5)-IDAY)526,520,520
  526 CONTINUE
      LSTAT(LK)=1
  523 LL=NTAL(LK)-NTAL(LK+1)
      LL1=NTAL(LK+1)+1
      LL2=NTAL(LK+1)+LL
      DO 527 L2=LL1,LL2
      IN=NACL(L2)
      IF(M(IN,5))527,528,527
  528 ICT=ICT+1
      ISEL(ICT)=IN
      ITEM(ICT)=M(IN,6)
      M(IN,4)=IDAY
  527 CONTINUE
  520 CONTINUE
  522 IF(ICT)534,535,534
  534 IF(ITRL-1)540,536,540
  536 IACT=1
  532 IP=ISEL(IACT)
      IF(M(IP,3))577,579,577
  535 PUNCH 537,IDAY
  537 FORMAT(39HNO ACTIVITY IS ELIGIBLE TO START ON DAY,I3)
      GO TO 589
  540 IF(ICT-1)545,552,545
  545 IPLAC=ICT-1
```

```
C     REARRANGEMENT OF ACTIVITIES IN PRIORITY ORDER
      DO 550 II=1,IPLAC
      III=II+1
      DO 550 LLL=III,ICT
      IF(ITEM(II)-ITEM(LLL))550,549,551
  549 CONTINUE
      IT=ISEL(II)
      ITI=ISEL(LLL)
      IF(M(IT,3)-M(ITI,3))550,550,556
  556 ISEL(II)=ITI
      ISEL(LLL)=IT
      GO TO 550
  551 IT=ISEL(II)
      ITI=ITEM(II)
      ITEM(II)=ITEM(LLL)
      ISEL(II)=ISEL(LLL)
      ITEM(LLL)=ITI
      ISEL(LLL)=IT
  550 CONTINUE
  552 PUNCH 555,IDAY,(ISEL(I),I=1,ICT)
  555 FORMAT(6HON DAY,I3,55H THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDE
     9R OF LST IS,/,20I3)
      IACT=1
  557 IP=ISEL(IACT)
      IF(M(IP,3))553,579,553
C     TESTING FOR RESOURCE AVAILABILITY
  553 JLACK=0
      DO 560  JRES=1,NRES
      IF(JRSAS(IDAY,JRES)+JRSRQ(IP,JRES)-JRSAV(ITRL,JRES))560,560,565
  565 IF(M(IP,6)-IDAY)568,585,585
  568 JLACK=JLACK+1
      JLSN(JLACK)=JRES
  560 CONTINUE
      IF(JLACK)572,577,572
  572 PUNCH 574,IP,IDAY,(JLSN(I),I=1,JLACK)
  574 FORMAT(8HACTIVITY,I3,20H CANNOT START ON DAY,I3,29H DUE TO LACK OF
     9 FOLLOWING RES,5I3)
      GO TO 585
C     ALLOCATION OF RESOURCES
  577 K1=IDAY
      KDUR=K1+M(IP,3)-1
      DO 580 KDAY=K1,KDUR
      DO 580  JRES =1,NRES
      JRSAS(KDAY,JRES)=JRSAS(KDAY,JRES)+JRSRQ(IP,JRES)
  580 CONTINUE
  579 NASD=NASD+1
      M(IP,4)=IDAY-1
      M(IP,5)=M(IP,4)+M(IP,3)
      NTASD=NTASD+1
  585 IF(IACT-ICT)581,582,582
  581 IACT=IACT+1
      IF(ITRL-1)532,532,557
  582 IF(ND)584,587,584
```

```
C     UPDATING OF COUNTER M1
  584 DO 586 IQ=1,ND
      IR=LMN(IQ)
      LL=NTAL(IR)-NTAL(IR+1)
      LL1=NTAL(IR+1)+1
      LL2=NTAL(IR+1)+LL
      DO 588 L3=LL1,LL2
      IY=NACL(L3)
      IF(M(IY,5))587,587,588
  588 CONTINUE
      M1=M1+1
  586 CONTINUE
  587 IF(NASD)589,592,589
  592 PUNCH 593,IDAY
  593 FORMAT(30HNO ACTIVITY COULD START ON DAY,I3,19H DUE TO LACK OF RES
     9)
  589 IF(NTASD-NACT)590,591,591
  590 IDAY=IDAY+1
      GO TO 512
  591 PUNCH 594,ITRL
  594 FORMAT(12HEND OF TRIAL,I3,28H GIVES THE FOLLOWING RESULTS)
      IF(ITRL-1)608,609,608
C     CALCULATION OF NEW PROJECT DURATION
  608 KK=NDUR
  602 KK=KK+1
      DO 603 JRES=1,NRES
      IF(JRSAS(KK,JRES))603,603,602
  603 CONTINUE
      NPRDR=KK-1
C     PRINTING OF RESULTS
      PUNCH 610,NPRDR
  610 FORMAT(27HTHE NEW PROJECT DURATION IS,I4)
      GO TO 611
  609 NPRDR=NDUR
  611 DO 607 I=1,NACT
  607 PUNCH 615,I,(M(I,J),J=1,6),(JRSRQ(I,J),J=1,NRES)
  615 FORMAT(I4,1H),10I7)
      PUNCH 628
      PUNCH 620,(JRSAV(ITRL,JRES),JRES=1,NRES)
  628 FORMAT(/,64HFOLLOWING RESOURCE AVAILABILITIES WERE CONSIDERED FOR
     *THIS TRIAL)
  620 FORMAT(5I10)
      PUNCH 629
  629 FORMAT(/,40HLEVELLED RESOURCE ALLOCATION RESULTS ARE)
      PUNCH 630,(K,(JRSAS(K,JRES),JRES=1,NRES),K=1,NPRDR)
  630 FORMAT(I3,1H),3I8,10X,I3,1H),3I8)
      IF(ITRL-1)648,648,632
  632 IF(NPRDR-NDUR)640,640,647
  647 IF(ITRL-NTRL)648,640,640
  648 ITRL=ITRL+1
      GO TO 503
  640 CONTINUE
      STOP
      END
```

APPENDIX-B

RESULTS OF CPM SCHEDULING PROGRAM

VALUES OF NTAE ARE

0	1	2	3	4	6	8	9	10	11	12	13	14	15	16
17	18	20	22	23	24	25	26	27	28	29	30	31	32	33
34	37	40	41	42	43	44	46	48	49	50	51	52	53	54
56	58	59	60	64	68	70								

VALUES OF NACE ARE

1	2	3	6	4	9	7	10	11	12	13	14	15	16	17
18	19	20	5	21	8	22	23	26	24	27	25	28	29	30
31	32	33	34	35	37	39	36	38	40	41	43	42	44	45
47	46	51	48	52	55	56	49	53	59	61	60	62	63	65
50	57	64	67	54	58	66	68	69	70					

VALUES OF EARLIEST OCCURRENCE TIME ARE

0	1	1	2	2	2	2	5	5	9	9	12	12	14	14
17	17	24	24	25	25	25	25	25	25	26	26	26	26	26
26	47	47	49	49	51	51	53	53	51	51	54	54	55	55
57	57	60	60	61	61	63								

VALUES OF NTAL ARE

70	68	65	62	61	60	59	58	57	56	55	54	53	52	51
50	49	48	45	42	41	40	39	38	37	36	35	34	33	32
31	30	28	26	25	24	20	16	15	14	13	12	11	10	9
8	6	4	3	2	1	0								

VALUES OF NACL ARE

70	69	68	67	65	66	63	64	62	61	60	59	58	57	56
55	51	52	53	54	47	48	49	50	46	45	43	44	41	42
40	39	38	37	36	35	34	33	32	31	30	29	26	27	28
23	24	25	22	21	20	19	18	17	16	15	14	13	12	11
10	9	6	7	8	3	4	5	1	2					

VALUES OF LATEST OCCURRENCE TIME ARE

0	1	1	2	2	2	2	5	5	9	9	12	12	14	14
17	17	24	24	25	25	25	25	25	25	26	26	26	26	26
26	47	47	53	53	51	51	55	55	60	60	56	56	55	55
57	57	51	61	61	61	63								

THE CPM SCHEDULE IS PRINTED BELOW

I-NODE	J-NODE	DURAT.	EST	EFT	LST	LFT	T.FLOAT	TRADE
1	2	1	0	1	0	1	0	8
1	3	1	0	1	0	1	0	8
2	4	1	1	2	1	2	0	6
2	6	1	1	2	1	2	0	6
2	18	2	1	3	22	24	21	4
3	5	1	1	2	1	2	0	6
3	7	1	1	2	1	2	0	6
3	19	2	1	3	22	24	21	4
4	6	0	2	2	2	2	0	0
5	7	0	2	2	2	2	0	0
6	8	3	2	5	2	5	0	7
7	9	3	2	5	2	5	0	7
8	10	4	5	9	5	9	0	7
9	11	4	5	9	5	9	0	7
10	12	3	9	12	9	12	0	7
11	13	3	9	12	9	12	0	7
12	14	2	12	14	12	14	0	7
13	15	2	12	14	12	14	0	7
14	16	3	14	17	14	17	0	2
15	17	3	14	17	14	17	0	2
16	18	7	17	24	17	24	0	1
17	19	7	17	24	17	24	0	1
18	20	1	24	25	24	25	0	4

I-NODE	J-NODE	DURAT.	EST	EFT	LST	LFT	T.FLOAT	TRADE
18	22	1	24	25	24	25	0	4
18	24	1	24	25	24	25	0	4
19	21	1	24	25	24	25	0	4
19	23	1	24	25	24	25	0	4
19	25	1	24	25	24	25	0	4
20	26	1	25	26	25	26	0	3
21	27	1	25	26	25	26	0	3
22	28	1	25	26	25	26	0	3
23	29	1	25	26	25	26	0	3
24	30	1	25	26	25	26	0	3
25	31	1	25	26	25	26	0	3
26	32	21	26	47	26	47	0	3
27	33	21	26	47	26	47	0	3
28	32	21	26	47	26	47	0	3
29	33	21	26	47	26	47	0	3
30	32	21	26	47	26	47	0	3
31	33	21	26	47	26	47	0	3
32	34	2	47	49	51	53	4	1
32	36	4	47	51	47	51	0	1
33	35	2	47	49	51	53	4	1
33	37	4	47	51	47	51	0	1
34	38	2	49	51	53	55	4	1
35	39	2	49	51	53	55	4	1
36	38	2	51	53	53	55	2	2
36	40	0	51	51	60	60	9	0
36	44	4	51	55	51	55	0	1
36	50	2	51	53	59	61	8	2
37	39	2	51	53	53	55	2	2
37	41	0	51	51	60	60	9	0
37	45	4	51	55	51	55	0	1
37	51	2	51	53	59	61	8	2
38	42	1	53	54	55	56	2	5
39	43	1	53	54	55	56	2	5
40	50	1	51	52	60	61	9	6
41	51	1	51	52	60	61	9	6
42	46	1	54	55	56	57	2	5
43	47	1	54	55	56	57	2	5
44	46	2	55	57	55	57	0	1
45	47	2	55	57	55	57	0	1
46	48	3	57	60	58	61	1	6
46	50	4	57	61	57	61	0	5
47	49	3	57	60	58	61	1	6
47	51	4	57	61	57	61	0	5
48	50	0	60	60	61	61	1	0
49	51	0	60	60	61	61	1	0
50	52	2	61	63	61	63	0	8
51	52	2	61	63	61	63	0	8

ACTIVITIES ARRANGED IN ASCENDING ORDER OF TOTAL FLOAT

I-NODE	J-NODE	DURAT.	EST	EFT	LST	LFT	T.FLOAT	TRADE
1	2	1	0	1	0	1	0	8
1	3	1	0	1	0	1	0	8
2	4	1	1	2	1	2	0	6
2	6	1	1	2	1	2	0	6
3	5	1	1	2	1	2	0	6
3	7	1	1	2	1	2	0	6
4	6	0	2	2	2	2	0	0
5	7	0	2	2	2	2	0	0
6	8	3	2	5	2	5	0	7
7	9	3	2	5	2	5	0	7
8	10	4	5	9	5	9	0	7
9	11	4	5	9	5	9	0	7
10	12	3	9	12	9	12	0	7
11	13	3	9	12	9	12	0	7
12	14	2	12	14	12	14	0	7
13	15	2	12	14	12	14	0	7
14	16	3	14	17	14	17	0	2
15	17	3	14	17	14	17	0	2
16	18	7	17	24	17	24	0	1
17	19	7	17	24	17	24	0	1
18	20	1	24	25	24	25	0	4
18	22	1	24	25	24	25	0	4
18	24	1	24	25	24	25	0	4
19	21	1	24	25	24	25	0	4
19	23	1	24	25	24	25	0	4
19	25	1	24	25	24	25	0	4
20	26	1	25	26	25	26	0	3
21	27	1	25	26	25	26	0	3
22	28	1	25	26	25	26	0	3
23	29	1	25	26	25	26	0	3
24	30	1	25	26	25	26	0	3
25	31	1	25	26	25	26	0	3
26	32	21	26	47	26	47	0	3
27	33	21	26	47	26	47	0	3
28	32	21	26	47	26	47	0	3
29	33	21	26	47	26	47	0	3
30	32	21	26	47	26	47	0	3
31	33	21	26	47	26	47	0	3
32	36	4	47	51	47	51	0	1
33	37	4	47	51	47	51	0	1
36	44	4	51	55	51	55	0	1
37	45	4	51	55	51	55	0	1
44	46	2	55	57	55	57	0	1
45	47	2	55	57	55	57	0	1
46	50	4	57	61	57	61	0	5

I-NODE	J-NODE	DURAT.	EST	EFT	LST	LFT	T.FLOAT	TRADE
47	51	4	57	61	57	61	0	5
50	52	2	61	63	61	63	0	8
51	52	2	61	63	61	63	0	8
48	50	0	60	60	61	61	1	0
49	51	0	60	60	61	61	1	0
47	49	3	57	60	58	61	1	6
46	48	3	57	60	58	61	1	6
42	46	1	54	55	56	57	2	5
43	47	1	54	55	56	57	2	5
37	39	2	51	53	53	55	2	2
36	38	2	51	53	53	55	2	2
38	42	1	53	54	55	56	2	5
39	43	1	53	54	55	56	2	5
34	38	2	49	51	53	55	4	1
35	39	2	49	51	53	55	4	1
33	35	2	47	49	51	53	4	1
32	34	2	47	49	51	53	4	1
36	50	2	51	53	59	61	8	2
37	51	2	51	53	59	61	8	2
36	40	0	51	51	60	60	9	0
37	41	0	51	51	60	60	9	0
40	50	1	51	52	60	61	9	6
41	51	1	51	52	60	61	9	6
2	18	2	1	3	22	24	21	4
3	19	2	1	3	22	24	21	4

NORMAL PROJECT DURATION= 63

THE NODE NUMBERS ARRANGED IN ASCENDING ORDER OF EOT ARE

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
33	34	35	36	37	40	41	38	39	42	43	44	45	46	47
49	50	51	52											

THE CALENDER DATED SCHEDULE IS PRINTED BELOW

I-NODE	J-NODE	DUR	EST	EFT	LST	LFT
1	2	1	5/ 2/75	6/ 2/75	5/ 2/75	6/ 2/75
1	3	1	5/ 2/75	6/ 2/75	5/ 2/75	6/ 2/75
2	4	1	6/ 2/75	7/ 2/75	6/ 2/75	7/ 2/75
2	6	1	6/ 2/75	7/ 2/75	6/ 2/75	7/ 2/75
2	18	2	6/ 2/75	8/ 2/75	5/ 3/75	7/ 3/75
3	5	1	6/ 2/75	7/ 2/75	6/ 2/75	7/ 2/75
3	7	1	6/ 2/75	7/ 2/75	6/ 2/75	7/ 2/75
3	19	2	6/ 2/75	8/ 2/75	5/ 3/75	7/ 3/75
4	6	0	7/ 2/75	7/ 2/75	7/ 2/75	7/ 2/75
5	7	0	7/ 2/75	7/ 2/75	7/ 2/75	7/ 2/75
6	8	3	7/ 2/75	11/ 2/75	7/ 2/75	11/ 2/75
7	9	3	7/ 2/75	11/ 2/75	7/ 2/75	11/ 2/75
8	10	4	11/ 2/75	15/ 2/75	11/ 2/75	15/ 2/75
9	11	4	11/ 2/75	15/ 2/75	11/ 2/75	15/ 2/75
10	12	3	15/ 2/75	20/ 2/75	15/ 2/75	20/ 2/75
11	13	3	15/ 2/75	20/ 2/75	15/ 2/75	20/ 2/75
12	14	2	20/ 2/75	22/ 2/75	20/ 2/75	22/ 2/75
13	15	2	20/ 2/75	22/ 2/75	20/ 2/75	22/ 2/75
14	16	3	22/ 2/75	27/ 2/75	22/ 2/75	27/ 2/75
15	17	3	22/ 2/75	27/ 2/75	22/ 2/75	27/ 2/75
16	18	7	27/ 2/75	7/ 3/75	27/ 2/75	7/ 3/75
17	19	7	27/ 2/75	7/ 3/75	27/ 2/75	7/ 3/75
18	20	1	7/ 3/75	8/ 3/75	7/ 3/75	8/ 3/75
18	22	1	7/ 3/75	8/ 3/75	7/ 3/75	8/ 3/75
18	24	1	7/ 3/75	8/ 3/75	7/ 3/75	8/ 3/75
19	21	1	7/ 3/75	8/ 3/75	7/ 3/75	8/ 3/75
19	23	1	7/ 3/75	8/ 3/75	7/ 3/75	8/ 3/75
19	25	1	7/ 3/75	8/ 3/75	7/ 3/75	8/ 3/75
20	26	1	8/ 3/75	10/ 3/75	8/ 3/75	10/ 3/75
21	27	1	8/ 3/75	10/ 3/75	8/ 3/75	10/ 3/75
22	28	1	8/ 3/75	10/ 3/75	8/ 3/75	10/ 3/75
23	29	1	8/ 3/75	10/ 3/75	8/ 3/75	10/ 3/75
24	30	1	8/ 3/75	10/ 3/75	8/ 3/75	10/ 3/75
25	31	1	8/ 3/75	10/ 3/75	8/ 3/75	10/ 3/75
26	32	21	10/ 3/75	5/ 4/75	10/ 3/75	5/ 4/75
27	33	21	10/ 3/75	5/ 4/75	10/ 3/75	5/ 4/75
28	32	21	10/ 3/75	5/ 4/75	10/ 3/75	5/ 4/75
29	33	21	10/ 3/75	5/ 4/75	10/ 3/75	5/ 4/75
30	32	21	10/ 3/75	5/ 4/75	10/ 3/75	5/ 4/75
31	33	21	10/ 3/75	5/ 4/75	10/ 3/75	5/ 4/75
32	34	2	5/ 4/75	8/ 4/75	10/ 4/75	12/ 4/75
32	36	4	5/ 4/75	10/ 4/75	5/ 4/75	10/ 4/75
33	35	2	5/ 4/75	8/ 4/75	10/ 4/75	12/ 4/75
33	37	4	5/ 4/75	10/ 4/75	5/ 4/75	10/ 4/75
34	38	2	8/ 4/75	10/ 4/75	12/ 4/75	15/ 4/75

I-NODE	J-NODE	DUR	EST	EFT	LST	LFT
35	39	2	8/ 4/75	10/ 4/75	12/ 4/75	15/ 4/75
36	38	2	10/ 4/75	12/ 4/75	12/ 4/75	15/ 4/75
36	40	0	10/ 4/75	10/ 4/75	21/ 4/75	21/ 4/75
36	44	4	10/ 4/75	15/ 4/75	10/ 4/75	15/ 4/75
36	50	2	10/ 4/75	12/ 4/75	19/ 4/75	22/ 4/75
37	39	2	10/ 4/75	12/ 4/75	12/ 4/75	15/ 4/75
37	41	0	10/ 4/75	10/ 4/75	21/ 4/75	21/ 4/75
37	45	4	10/ 4/75	15/ 4/75	10/ 4/75	15/ 4/75
37	51	2	10/ 4/75	12/ 4/75	19/ 4/75	22/ 4/75
38	42	1	12/ 4/75	14/ 4/75	15/ 4/75	16/ 4/75
39	43	1	12/ 4/75	14/ 4/75	15/ 4/75	16/ 4/75
40	50	1	10/ 4/75	11/ 4/75	21/ 4/75	22/ 4/75
41	51	1	10/ 4/75	11/ 4/75	21/ 4/75	22/ 4/75
42	46	1	14/ 4/75	15/ 4/75	16/ 4/75	17/ 4/75
43	47	1	14/ 4/75	15/ 4/75	16/ 4/75	17/ 4/75
44	46	2	15/ 4/75	17/ 4/75	15/ 4/75	17/ 4/75
45	47	2	15/ 4/75	17/ 4/75	15/ 4/75	17/ 4/75
46	48	3	17/ 4/75	21/ 4/75	18/ 4/75	22/ 4/75
46	50	4	17/ 4/75	22/ 4/75	17/ 4/75	22/ 4/75
47	49	3	17/ 4/75	21/ 4/75	18/ 4/75	22/ 4/75
47	51	4	17/ 4/75	22/ 4/75	17/ 4/75	22/ 4/75
48	50	0	21/ 4/75	21/ 4/75	22/ 4/75	22/ 4/75
49	51	0	21/ 4/75	21/ 4/75	22/ 4/75	22/ 4/75
50	52	2	22/ 4/75	24/ 4/75	22/ 4/75	24/ 4/75
51	52	2	22/ 4/75	24/ 4/75	22/ 4/75	24/ 4/75

RESOURCE LEVELLING PROGRAM RESULTS

RESULTS OF FIRST TRIAL

DAILY RESOURCE REQUIREMENTS FOR TRIAL 1 ARE

DAY	BELDAR	CARPNTR	MASON	DAY	BELDAR	CARPNTR	MASON
1)	10	0	0	2)	20	0	0
3)	20	4	0	4)	16	4	0
5)	16	4	0	6)	8	8	0
7)	8	8	0	8)	8	8	0
9)	8	8	0	10)	8	2	0
11)	8	2	0	12)	8	2	0
13)	8	0	2	14)	8	0	2
15)	8	0	4	16)	8	0	4
17)	8	0	4	18)	4	6	0
19)	4	6	0	20)	4	6	0
21)	4	6	0	22)	4	6	0
23)	4	6	0	24)	4	6	0
25)	12	0	0	26)	60	0	12
27)	6	0	0	28)	6	0	0
29)	6	0	0	30)	6	0	0
31)	6	0	0	32)	6	0	0
33)	6	0	0	34)	6	0	0
35)	6	0	0	36)	6	0	0
37)	6	0	0	38)	6	0	0
39)	6	0	0	40)	6	0	0
41)	6	0	0	42)	6	0	0
43)	6	0	0	44)	6	0	0
45)	6	0	0	46)	6	0	0
47)	6	0	0	48)	6	6	0
49)	6	6	0	50)	6	6	0
51)	6	6	0	52)	16	2	4
53)	8	2	4	54)	8	2	0
55)	8	2	0	56)	2	2	0
57)	2	2	0	58)	12	0	0
59)	12	0	0	60)	12	0	0
61)	4	0	0	62)	16	0	0
63)	16	0	0				

FOLLOWING RESOURCE AVAILABILITIES WERE CONSIDERED FOR THIS TRIAL

BELDARS	CARPENTERS	MASONS
100	100	100

RESULTS OF SECOND TRIAL

PROGRESS REPORT FOR TRIAL 2 IS GIVEN BELOW

```
ON DAY  1 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
   1  2
ON DAY  2 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
   3  4  6  7  5  8
ON DAY  3 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
   9 10 11 12  5  8
ON DAY  4 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
   5  8
NO ACTIVITY COULD START ON DAY  4 DUE TO LACK OF RES
ON DAY  5 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
   5  8
NO ACTIVITY COULD START ON DAY  5 DUE TO LACK OF RES
ON DAY  6 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
  13 14  5  8
ACTIVITY 14 CANNOT START ON DAY  6 DUE TO LACK OF FOLLOWING RES  2
ON DAY  7 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
  14
ACTIVITY 14 CANNOT START ON DAY  7 DUE TO LACK OF FOLLOWING RES  2
NO ACTIVITY COULD START ON DAY  7 DUE TO LACK OF RES
ON DAY  8 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
  14
ACTIVITY 14 CANNOT START ON DAY  8 DUE TO LACK OF FOLLOWING RES  2
NO ACTIVITY COULD START ON DAY  8 DUE TO LACK OF RES
ON DAY  9 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
  14
ACTIVITY 14 CANNOT START ON DAY  9 DUE TO LACK OF FOLLOWING RES  2
NO ACTIVITY COULD START ON DAY  9 DUE TO LACK OF RES
ON DAY 10 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
  14 15
ACTIVITY 15 CANNOT START ON DAY 10 DUE TO LACK OF FOLLOWING RES  2
ON DAY 11 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
  15
ACTIVITY 15 CANNOT START ON DAY 11 DUE TO LACK OF FOLLOWING RES  2
NO ACTIVITY COULD START ON DAY 11 DUE TO LACK OF RES
ON DAY 12 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
  15
ACTIVITY 15 CANNOT START ON DAY 12 DUE TO LACK OF FOLLOWING RES  2
NO ACTIVITY COULD START ON DAY 12 DUE TO LACK OF RES
ON DAY 13 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
  15
ACTIVITY 15 CANNOT START ON DAY 13 DUE TO LACK OF FOLLOWING RES  2
NO ACTIVITY COULD START ON DAY 13 DUE TO LACK OF RES
ON DAY 14 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
  15 16
NO ACTIVITY IS ELIGIBLE TO START ON DAY 15
NO ACTIVITY IS ELIGIBLE TO START ON DAY 16
ON DAY 17 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
  17 18
NO ACTIVITY IS ELIGIBLE TO START ON DAY 18
```

```
ON DAY 19 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 19 20
NO ACTIVITY IS ELIGIBLE TO START ON DAY 20
NO ACTIVITY IS ELIGIBLE TO START ON DAY 21
ON DAY 22 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 21 22
ACTIVITY 22 CANNOT START ON DAY 22 DUE TO LACK OF FOLLOWING RES  2
ON DAY 23 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 22
ACTIVITY 22 CANNOT START ON DAY 23 DUE TO LACK OF FOLLOWING RES  2
NO ACTIVITY COULD START ON DAY 23 DUE TO LACK OF RES
ON DAY 24 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 22
ACTIVITY 22 CANNOT START ON DAY 24 DUE TO LACK OF FOLLOWING RES  2
NO ACTIVITY COULD START ON DAY 24 DUE TO LACK OF RES
ON DAY 25 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 22
ACTIVITY 22 CANNOT START ON DAY 25 DUE TO LACK OF FOLLOWING RES  2
NO ACTIVITY COULD START ON DAY 25 DUE TO LACK OF RES
ON DAY 26 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 22
ACTIVITY 22 CANNOT START ON DAY 26 DUE TO LACK OF FOLLOWING RES  2
NO ACTIVITY COULD START ON DAY 26 DUE TO LACK OF RES
ON DAY 27 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 22
ACTIVITY 22 CANNOT START ON DAY 27 DUE TO LACK OF FOLLOWING RES  2
NO ACTIVITY COULD START ON DAY 27 DUE TO LACK OF RES
ON DAY 28 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 22
ACTIVITY 22 CANNOT START ON DAY 28 DUE TO LACK OF FOLLOWING RES  2
NO ACTIVITY COULD START ON DAY 28 DUE TO LACK OF RES
ON DAY 29 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 22 23 24 25
ON DAY 30 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 29 31 33
ACTIVITY 31 CANNOT START ON DAY 30 DUE TO LACK OF FOLLOWING RES  1
ACTIVITY 33 CANNOT START ON DAY 30 DUE TO LACK OF FOLLOWING RES  1
ON DAY 31 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 31 33 35
ACTIVITY 33 CANNOT START ON DAY 31 DUE TO LACK OF FOLLOWING RES  1
ON DAY 32 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 33 37
ON DAY 33 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 39
NO ACTIVITY IS ELIGIBLE TO START ON DAY 34
NO ACTIVITY IS ELIGIBLE TO START ON DAY 35
ON DAY 36 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 26 27 28
ON DAY 37 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 30 32 34
ACTIVITY 32 CANNOT START ON DAY 37 DUE TO LACK OF FOLLOWING RES  1
ACTIVITY 34 CANNOT START ON DAY 37 DUE TO LACK OF FOLLOWING RES  1
ON DAY 38 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 32 34 36
```

```
ACTIVITY 34 CANNOT START ON DAY 38 DUE TO LACK OF FOLLOWING RES  1
ON DAY 39 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 34 38
ON DAY 40 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 40
NO ACTIVITY IS ELIGIBLE TO START ON DAY 41
NO ACTIVITY IS ELIGIBLE TO START ON DAY 42
NO ACTIVITY IS ELIGIBLE TO START ON DAY 43
NO ACTIVITY IS ELIGIBLE TO START ON DAY 44
NO ACTIVITY IS ELIGIBLE TO START ON DAY 45
NO ACTIVITY IS ELIGIBLE TO START ON DAY 46
NO ACTIVITY IS ELIGIBLE TO START ON DAY 47
NO ACTIVITY IS ELIGIBLE TO START ON DAY 48
NO ACTIVITY IS ELIGIBLE TO START ON DAY 49
NO ACTIVITY IS ELIGIBLE TO START ON DAY 50
NO ACTIVITY IS ELIGIBLE TO START ON DAY 51
NO ACTIVITY IS ELIGIBLE TO START ON DAY 52
NO ACTIVITY IS ELIGIBLE TO START ON DAY 53
ON DAY 54 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 42 41
NO ACTIVITY IS ELIGIBLE TO START ON DAY 55
ON DAY 56 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 45
NO ACTIVITY IS ELIGIBLE TO START ON DAY 57
ON DAY 58 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 49 47 50 48 57
NO ACTIVITY IS ELIGIBLE TO START ON DAY 59
ON DAY 60 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 55
ON DAY 61 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 44 43 59
ON DAY 62 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 61
ON DAY 63 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 46
ON DAY 64 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 64 63
ON DAY 65 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 53 51 54 52 58
NO ACTIVITY IS ELIGIBLE TO START ON DAY 66
ON DAY 67 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 56 67
ON DAY 68 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 60 69
ON DAY 69 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
 62
```

NO ACTIVITY IS ELIGIBLE TO START ON DAY 70
ON DAY 71 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
66 65
NO ACTIVITY IS ELIGIBLE TO START ON DAY 72
NO ACTIVITY IS ELIGIBLE TO START ON DAY 73
ON DAY 74 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
68
ON DAY 75 THE LIST OF ELIGIBLE ACTIVITIES IN ASC ORDER OF LST IS
70

****

END OF TRIAL 2 GIVES THE FOLLOWING RESULTS

THE NEW PROJECT DURATION IS 76

****

THE REVISED SCHEDULE IS GIVEN BELOW

****

S.NO	I-NODE	J-NODE	DURA- TION.	NEW EST	NEW EFT	OLD LST	RESOURCES REQUIRED BELDARS	CARP.	MASONS
1)	1	2	1	0	1	0	5	0	0
2)	1	3	1	0	1	0	5	0	0
3)	2	4	1	1	2	1	4	0	0
4)	2	6	1	1	2	1	4	0	0
5)	2	18	2	5	7	22	2	0	0
6)	3	5	1	1	2	1	4	0	0
7)	3	7	1	1	2	1	4	0	0
8)	3	19	2	5	7	22	2	0	0
9)	4	6	0	2	2	2	0	0	0
10)	5	7	0	2	2	2	0	0	0
11)	6	8	3	2	5	2	8	2	0
12)	7	9	3	2	5	2	8	2	0
13)	8	10	4	5	9	5	4	4	0
14)	9	11	4	9	13	5	4	4	0
15)	10	12	3	13	16	9	4	1	0
16)	11	13	3	13	16	9	4	1	0
17)	12	14	2	16	18	12	4	0	1
18)	13	15	2	16	18	12	4	0	1
19)	14	16	3	18	21	14	4	0	2
20)	15	17	3	18	21	14	4	0	2
21)	16	18	7	21	28	17	2	3	0
22)	17	19	7	28	35	17	2	3	0
23)	18	20	1	28	29	24	2	0	0
24)	18	22	1	28	29	24	2	0	0
25)	18	24	1	28	29	24	2	0	0
26)	19	21	1	35	36	24	2	0	0
27)	19	23	1	35	36	24	2	0	0

S.NO	I-NODE	J-NODE	DURA-TION.	NEW EST	NEW EFT	OLD LST	RESOURCES REQUIRED BELDARS	CARP.	MASONS
28)	19	25	1	35	36	24	2	0	0
29)	20	26	1	29	30	25	10	0	2
30)	21	27	1	36	37	25	10	0	2
31)	22	28	1	30	31	25	10	0	2
32)	23	29	1	37	38	25	10	0	2
33)	24	30	1	31	32	25	10	0	2
34)	25	31	1	38	39	25	10	0	2
35)	26	32	21	30	51	26	1	0	0
36)	27	33	21	37	58	26	1	0	0
37)	28	32	21	31	52	26	1	0	0
38)	29	33	21	38	59	26	1	0	0
39)	30	32	21	32	53	26	1	0	0
40)	31	33	21	39	60	26	1	0	0
41)	32	34	2	53	55	51	1	1	0
42)	32	36	4	53	57	47	2	2	0
43)	33	35	2	60	62	51	1	1	0
44)	33	37	4	60	64	47	2	2	0
45)	34	38	2	55	57	53	1	1	0
46)	35	39	2	62	64	53	1	1	0
47)	36	38	2	57	59	53	1	0	1
48)	36	40	0	57	57	60	0	0	0
49)	36	44	4	57	61	51	2	1	0
50)	36	50	2	57	59	59	1	0	1
51)	37	39	2	64	66	53	1	0	1
52)	37	41	0	64	64	60	0	0	0
53)	37	45	4	64	68	51	2	1	0
54)	37	51	2	64	66	59	1	0	1
55)	38	42	1	59	60	55	2	0	0
56)	39	43	1	66	67	55	2	0	0
57)	40	50	1	57	58	60	4	0	0
58)	41	51	1	64	65	60	4	0	0
59)	42	46	1	60	61	56	2	0	0
60)	43	47	1	67	68	56	2	0	0
61)	44	46	2	61	63	55	1	1	0
62)	45	47	2	68	70	55	1	1	0
63)	46	48	3	63	66	58	4	0	0
64)	46	50	4	63	67	57	2	0	0
65)	47	49	3	70	73	58	4	0	0
66)	47	51	4	70	74	57	2	0	0
67)	48	50	0	66	66	61	0	0	0
68)	49	51	0	73	73	61	0	0	0
69)	50	52	2	67	69	61	8	0	0
70)	51	52	2	74	76	61	8	0	0

DAILY RESOURCE REQUIREMENTS FOR TRIAL 2 ARE

DAY	BELDAR	CARPNTR	MASON	DAY	BELDAR	CARPNTR	MASON
1)	10	0	0	2)	16	0	0
3)	16	4	0	4)	16	4	0
5)	16	4	0	6)	8	4	0
7)	8	4	0	8)	4	4	0
9)	4	4	0	10)	4	4	0
11)	4	4	0	12)	4	4	0
13)	4	4	0	14)	8	2	0
15)	8	2	0	16)	8	2	0
17)	8	0	2	18)	8	0	2
19)	8	0	4	20)	8	0	4
21)	8	0	4	22)	2	3	0
23)	2	3	0	24)	2	3	0
25)	2	3	0	26)	2	3	0
27)	2	3	0	28)	2	3	0
29)	8	3	0	30)	12	3	2
31)	13	3	2	32)	14	3	2
33)	5	3	0	34)	5	3	0
35)	5	3	0	36)	9	0	0
37)	13	0	2	38)	14	0	2
39)	15	0	2	40)	6	0	0
41)	6	0	0	42)	6	0	0
43)	6	0	0	44)	6	0	0
45)	6	0	0	46)	6	0	0
47)	6	0	0	48)	6	0	0
49)	6	0	0	50)	6	0	0
51)	6	0	0	52)	5	0	0
53)	4	0	0	54)	6	3	0
55)	6	3	0	56)	6	3	0
57)	6	3	0	58)	11	1	2
59)	6	1	2	60)	5	1	0
61)	7	4	0	62)	4	4	0
63)	4	4	0	64)	9	3	0
65)	14	1	2	66)	10	1	2
67)	6	1	0	68)	12	1	0
69)	9	1	0	70)	1	1	0
71)	6	0	0	72)	6	0	0
73)	6	0	0	74)	2	0	0
75)	8	0	0	76)	8	0	0

FOLLOWING RESOURCE AVAILABILITIES WERE CONSIDERED FOR THIS TRIAL

BELDARS	CARPENTERS	MASONS
16	4	4

DAILY RESOURCE REQUIREMENTS FOR TRIAL 2 ARE

DAY	BELDAR	CARPNTR	MASON	DAY	BELDAR	CARPNTR	MASON
1)	10	0	0	2)	16	0	0
3)	16	4	0	4)	16	4	0
5)	16	4	0	6)	8	4	0
7)	8	4	0	8)	4	4	0
9)	4	4	0	10)	4	4	0
11)	4	4	0	12)	4	4	0
13)	4	4	0	14)	8	2	0
15)	8	2	0	16)	8	2	0
17)	8	0	2	18)	8	0	2
19)	8	0	4	20)	8	0	4
21)	8	0	4	22)	2	3	0
23)	2	3	0	24)	2	3	0
25)	2	3	0	26)	2	3	0
27)	2	3	0	28)	2	3	0
29)	8	3	0	30)	12	3	2
31)	13	3	2	32)	14	3	2
33)	5	3	0	34)	5	3	0
35)	5	3	0	36)	9	0	0
37)	13	0	2	38)	14	0	2
39)	15	0	2	40)	6	0	0
41)	6	0	0	42)	6	0	0
43)	6	0	0	44)	6	0	0
45)	6	0	0	46)	6	0	0
47)	6	0	0	48)	6	0	0
49)	6	0	0	50)	6	0	0
51)	6	0	0	52)	5	0	0
53)	4	0	0	54)	6	3	0
55)	6	3	0	56)	6	3	0
57)	6	3	0	58)	11	1	2
59)	6	1	2	60)	5	1	0
61)	7	4	0	62)	4	4	0
63)	4	4	0	64)	9	3	0
65)	14	1	2	66)	10	1	2
67)	6	1	0	68)	12	1	0
69)	9	1	0	70)	1	1	0
71)	6	0	0	72)	6	0	0
73)	6	0	0	74)	2	0	0
75)	8	0	0	76)	8	0	0

FOLLOWING RESOURCE AVAILABILITIES WERE CONSIDERED FOR THIS TRIAL

BELDARS	CARPENTERS	MASONS
16	4	4

RESULTS OF THIRD TRIAL

THE NEW PROJECT DURATION IS 69

THE REVISED SCHEDULE IS GIVEN BELOW

S.NO	I-NODE	J-NODE	DURA-TION.	NEW EST	NEW EFT	OLD LST	RESOURCES REQUIRED BELDARS	CARP.	MASONS
1)	1	2	1	0	1	0	5	0	0
2)	1	3	1	0	1	0	5	0	0
3)	2	4	1	1	2	1	4	0	0
4)	2	5	1	1	2	1	4	0	0
5)	2	18	2	5	7	22	2	0	0
6)	3	5	1	1	2	1	4	0	0
7)	3	7	1	1	2	1	4	0	0
8)	3	19	2	5	7	22	2	0	0
9)	4	6	0	2	2	2	0	0	0
10)	5	7	0	2	2	2	0	0	0
11)	6	3	3	2	5	2	8	2	0
12)	7	9	3	2	5	2	8	2	0
13)	8	10	4	5	9	5	4	4	0
14)	9	11	4	9	13	5	4	4	0
15)	10	12	3	9	12	9	4	1	0
16)	11	13	3	13	16	9	4	1	0
17)	12	14	2	12	14	12	4	0	1
18)	13	15	2	16	18	12	4	0	1
19)	14	15	3	14	17	14	4	0	2
20)	15	17	3	18	21	14	4	0	2
21)	16	18	7	17	24	17	2	3	0
22)	17	19	7	21	28	17	2	3	0
23)	18	20	1	24	25	24	2	0	0
24)	18	22	1	24	25	24	2	0	0
25)	18	24	1	24	25	24	2	0	0
26)	19	21	1	28	29	24	2	0	0
27)	19	23	1	28	29	24	2	0	0
28)	19	25	1	28	29	24	2	0	0
29)	20	26	1	25	26	25	10	0	2
30)	21	27	1	29	30	25	10	0	2
31)	22	28	1	26	27	25	10	0	2
32)	23	29	1	30	31	25	10	0	2
33)	24	30	1	27	28	25	10	0	2
34)	25	31	1	31	32	25	10	0	2
35)	26	32	21	26	47	26	1	0	0
36)	27	33	21	30	51	26	1	0	0
37)	28	32	21	27	48	26	1	0	0
38)	29	33	21	31	52	26	1	0	0
39)	30	32	21	28	49	26	1	0	0
40)	31	33	21	32	53	26	1	0	0

S.NO	I-NODE	J-NODE	DURA-TION.	NEW EST	NEW EFT	OLD LST	RESOURCES REQUIRED BELDARS	CARP.	MASONS
41)	32	34	2	49	51	51	1	1	0
42)	32	36	4	49	53	47	2	2	0
43)	33	35	2	53	55	51	1	1	0
44)	33	37	4	53	57	47	2	2	0
45)	34	38	2	51	53	53	1	1	0
46)	35	39	2	55	57	53	1	1	0
47)	36	33	2	53	55	53	1	0	1
48)	36	40	0	53	53	60	0	0	0
49)	36	44	4	53	57	51	2	1	0
50)	36	50	2	53	55	59	1	0	1
51)	37	39	2	57	59	53	1	0	1
52)	37	41	0	57	57	60	0	0	0
53)	37	45	4	57	61	51	2	1	0
54)	37	51	2	57	59	59	1	0	1
55)	38	42	1	55	56	55	2	0	0
56)	39	43	1	59	60	55	2	0	0
57)	40	50	1	53	54	60	4	0	0
58)	41	51	1	57	58	60	4	0	0
59)	42	46	1	56	57	56	2	0	0
60)	43	47	1	60	61	56	2	0	0
61)	44	46	2	57	59	55	1	1	0
62)	45	47	2	61	63	55	1	1	0
63)	46	48	3	59	62	58	4	0	0
64)	46	50	4	59	63	57	2	0	0
65)	47	49	3	63	66	58	4	0	0
66)	47	51	4	63	67	57	2	0	0
67)	48	50	0	62	62	61	0	0	0
68)	49	51	0	66	66	61	0	0	0
69)	50	52	2	63	65	61	8	0	0
70)	51	52	2	67	69	61	8	0	0

FOLLOWING RESOURCE AVAILABILITIES WERE CONSIDERED FOR THIS TRIAL

BELDARS	CARPENTERS	MASONS
16	6	4

THE DAILY RESOURCE REQUIREMENTS FOR TRIAL 3 ARE

DAY	BELDAR	CARPNTR	MASON	DAY	BELDAR	CARPNTR	MASON
1)	10	0	0	2)	16	0	0
3)	16	4	0	4)	16	4	0
5)	16	4	0	6)	8	4	0
7)	8	4	0	8)	4	4	0
9)	4	4	0	10)	8	5	0
11)	8	5	0	12)	8	5	0
13)	8	4	1	14)	8	1	1
15)	8	1	2	16)	8	1	2
17)	8	0	3	18)	6	3	1
19)	6	3	2	20)	6	3	2
21)	6	3	2	22)	4	6	0
23)	4	6	0	24)	4	6	0
25)	8	3	0	26)	12	3	2
27)	13	3	2	28)	14	3	2
29)	9	0	0	30)	13	0	2
31)	14	0	2	32)	15	0	2
33)	6	0	0	34)	6	0	0
35)	6	0	0	36)	6	0	0
37)	6	0	0	38)	6	0	0
39)	6	0	0	40)	6	0	0
41)	6	0	0	42)	6	0	0
43)	6	0	0	44)	6	0	0
45)	6	0	0	46)	6	0	0
47)	6	0	0	48)	5	0	0
49)	4	0	0	50)	6	3	0
51)	6	3	0	52)	5	3	0
53)	4	3	0	54)	11	4	2
55)	7	4	2	56)	7	4	0
57)	7	4	0	58)	9	2	2
59)	5	2	2	60)	10	1	0
61)	10	1	0	62)	7	1	0
63)	3	1	0	64)	14	0	0
65)	14	0	0	66)	6	0	0
67)	2	0	0	68)	8	0	0
69)	8	0	0				