In presenting the dissertation as a partial fulfillment of the requirements for an advanced degree from the Georgia Institute of Technology, I agree that the Library of the Institute shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to copy from, or to publish from, this dissertation may be granted by the professor under whose direction it was written, or, in his absence, by the Dean of the Graduate Division when such copying or publication is solely for scholarly purposes and does not involve potential financial gain. It is understood that any copying from, or publication of, this dissertation which involves potential financial gain will not be allowed without written permission.

3/17/65
b

DEVELOPMENT OF A RESOURCE ALLOCATION TECHNIQUE BASED ON RESOURCE-TIME UNITS

A THESIS

Presented to

The Faculty of the Graduate Division

by

Charles Albert McNeill

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Industrial Engineering

Georgia Institute of Technology

April, 1967

DEVELOPMENT OF A RESOURCE ALLOCATION

TECHNIQUE BASED ON RESOURSE-TIME UNITS

Approved:

Chairman

Date approved by Chairman: 5/15/67

ACKNOWLEDGMENTS

The author is deeply indebted to a number of people whose help and encouragement have made this work possible. First of all, he would like to express his deep appreciation to his advisor, Professor J. Gordon Davis. The helpful suggestions and friendly encouragement of Professor Davis have meant much to this author during the preparation of this thesis. He would like to thank Dr. B. C. Spradlin and Mr. A. P. Jensen, of the Rich Electronic Computer Center, for their advice and service on the reading committee. Also, very special thanks are due to Mr. James A. Hudson whose help and encouragement were invaluable during the preparation of the two computer programs used in this thesis.

Finally, the author would like to express his appreciation to his parents, Mr. and Mrs. L. C. McNeill, whose guidance, encouragement, and support have been invaluable to him in his educational endeavers.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

SUMMARY

Investigation of resource allocation features of project management systems indicated that little attention has been given to activities with variable resource and time requirements. A survey of existing computer programs in the field of project management indicated that only a small percentage of the available programs have a resource allocation feature.

A trial and error method based upon original ideas of the author was used to develop the resource allocation procedure which is called the REST Technique. The original REST Technique was tested and revised a number of times before a satisfactory method was found. The final version of the REST Technique uses resource-time units to measure the work content of all activities. The REST Technique analyzes a network on the basis of path work content rather than the usual path length. The REST Technique uses total remaining path work content as the criteria for determining the priority of scheduling.

Both a manual algorithm and a computer program written in Algol for the Burroughs 5500 computer were developed for this technique. The REST Technique is governed by the following assumptions and restrictions:

1. The resource supply is known and constant.
2. The REST units of each activity can be estimated.
3. A minimum and maximum allowable resource requirement can be determined for each activity.

4. An activity can have different quantities of a resource allocated to it in different time periods.
5. Work on an activity must not necessarily occur in consecutive time periods.
6. Only one resource type is considered by the computer program and although the manual algorithm considers more than one resource type, it considers only one resource type per activity.
7. Only one project is considered.
8. No time-cost trade-off is considered.

The REST Technique has the following advantages over many of the existing resource allocation techniques:

1. The schedule of the project is dependent upon the available resource supply rather than the resource requirement being dependent upon a predetermined project duration.
2. The REST Technique attempts to fully utilize the entire resource supply in each time period.
3. The project duration is dependent upon the available resource supply.
4. The REST Technique is easy for someone untrained in networking techniques to use.
5. The manual algorithm will handle multi-resource allocation problems.
6. The REST Technique is particularly suited to the needs of the many businesses which operate with fixed personnel and resource levels and which cannot afford the high training costs or

consulting fees necessary to benefit from the existing methods.

CHAPTER I

INTRODUCTION

Ever since the conception of modern project management was heralded in 1958 by the introduction of the Critical Path Method (CPM) and the Project Evaluation and Review Technique (PERT), a large number of acronyms have appeared announcing the development of newer and more specialized additions to the store of available management tools.

Today, these techniques are so widely accepted that practically every area of industry and government is affected. Networking techniques are now used everywhere from construction site to hospital room and from factory assembly line to research laboratory. Because of the individual requirements of each group wishing to use these techniques, no one method can be adapted to satisfy the needs of every group.

The original techniques as developed in 1958 were simply methods of project analysis which described the various activities composing the project, indicated the precedence requirements among the activities, and estimated the duration of the project. Although PERT was developed in connection with the Navy's Polaris Program and CPM was developed by DuPont to help control their construction projects, these methods were soon found to be insufficient and new methods were developed to help give the project manager a more complete solution to his problems. Among the areas given added attention by this second generation of techniques were resource allocation, time-cost trade-off, project cost control, and accounting. Resource allocation techniques were developed to schedule limited quantities of one or more resources among the activities of one or more projects. The time-cost trade-off techniques were concerned with the time-cost relationship involved in the alteration of a project duration. Other procedures were developed with the idea of using the networking techniques to give a new dimension to cost control and accounting.

The second generation resource allocation techniques are all extensions of the original networking methods in that they all make use of the network diagram and associated calculations in their procedures. In the network portions of these systems, the activity descriptions are determined while the initial network information is being gathered. A normal crew size or resource assignment for an activity is determined by the estimator and then a time estimate based on the assigned crew size or resource quantity is assigned to the activity. The MAP procedure, developed by Dr. R. L. Martino (1), is the one major exception in that it also assigns resource-time units to some activities.

Of the many resource allocation techniques available, all seem to have their deficiencies. Some of the methods are limited by the types of problems which they can solve while others are limited by the training required for an analyst to use them successfully. There are four basic types of activities which may be found in a network:

(1) Activities requiring a fixed resource level and a corresponding fixed duration.

(2) Activities permitting a variable resource level and a corresponding variable duration.

(3) Activities having a fixed non-zero duration but requiring no resources.

(4) Activities having a zero duration and requiring no resources. Most methods of allocation deal with only the first and last types of activities and thus are seen to be incomplete. However, since it is possible to fix the resource and duration levels of the second type of activities, these activities can be considered to be of the first type and the basic allocation methods will often give reasonable solutions to a wide range of problems. Since the MAP procedure considers the second type of activity, it is possibly the most advanced of the prominent methods.

It is strange to note that none of the major techniques seem to consider the third type of activity, which is basically a forced delay or waiting period. Since this type of activity plays a major role in many projects, its existence should not be ignored. Although it is possible that some existing methods, including MAP, are capable of scheduling activities of this type, no detailed information is offered concerning the methods to be applied to such activities.

Most resource allocation procedures, not excepting MAP, seem to require a trained analyst with a theoretical knowledge of the method to do the actual scheduling. This required training possibly prohibits the use of these methods by groups which do not have a trained analyst and which cannot afford to train or hire one.

Consideration of the advantages, disadvantages, and restrictions of the many resource allocation techniques indicates that there has yet to be designed a practical method for the many businesses which operate with fixed personnel and resource levels and which cannot afford the high training costs or consulting fees necessary to benefit from the existing methods. For the number of allocation procedures in use today, very little attention seems to have been given to activities with variable resource and time requirements, especially since a very large portion of actual activities could be of this type. It has also been noticed that present procedures give no attention to the possibility of intermittent starting and stopping of activities to achieve a more suitable schedule.

In view of the apparent need as indicated by the deficiencies of the existing techniques, further developments in the area of resource allocation seem desirable. The purpose of this thesis, then, is to develop a procedure for the allocation of resources which will consider all types of activities, which will be easy for someone untrained in network procedures and project management techniques to use, and which will produce results which will be consistent and reasonably close to the ideal solution. The method being developed will give primary attention to the use of activities with variable resource requirements and corresponding variable durations and the intermittent starting and stopping of activities to achieve good answers. Would-be users of this method will have to decide if the possible ease of application outweighs the possible deviation of the results from the ideal answers.

In the remaining portion of this paper, the method being developed will be referred to as the REST Technique and the units to specify activity work content will be referred to as REST units where REST is an acronym composed of letters from the words, resource-time. Typical REST units are man-days and crew-weeks.

CHAPTER II

LITERATURE SEARCH

An extensive search of available literature was made in an effort to locate previous work dealing with the use of resource-time units in resource allocation and leveling. It should be realized that not all relevant material was investigated due to the unavailability of some of it. However, a large selection of varied materials was inspected.

A large quantity of material related to resource allocation and leveling was accumulated and studied. Although mention was given to allocation and leveling in many publications and articles, most of the attention was very superficial. Very few authors gave actual details of an allocation or leveling algorithm with enough clarity to permit the reader to make use of them.

Much of the material which is available in the area of scheduling is actually much more closely related to time-cost trade-off than to resource allocation or leveling. An example of this situation is described in the following abstract (2):

> An explanation of Critical-Path Scheduling is given considering its two phases - scheduling and the computation of the project cost curve. A labeling algorithm using network flow theory is developed to solve a linear programming problem of computing the least cost curve for a project composed of many related jobs. The purpose of the routine is to determine which jobs within the project should be accelerated in order to shorten the project duration time at a minimum cost.

While much of the material which is classified as being in the area of scheduling is not concerned with allocation or leveling of resources or is concerned only to a very minor extent, much of the remaining material is of such a nature that a theoretical understanding and much practical experience is required of the reader before the material can be of much value to him. This point is illustrated by the following quote (3): "It is a highly sophisticated extension of network planning and scheduling methods particularly suited to the needs of experienced CPM or PERT users." Although this remark was made about Resource Allocation and Multiproject Scheduling (RAMPS), it could have been directed at numerous other methods just as easily.

The actual allocation and leveling techniques which are available can be classified into two catagories, manual and computerized. While it is imperative to use a computerized procedure on very large problems, the use of the same procedure on a very small or simple problem may be impractical. For this reason, both types of procedures are needed.

Possibly the most common manual procedures include the methods developed by Burgess (4), Brooks (5), and Martino (1). The leveling technique which was developed by Burgess compares alternate schedules using the sum of squares of the resource requirements during each time period. This method equates increasing schedule efficiency with decreasing sums of squares. This procedure may require several passes to acquire the final solution.

The algorithm by Brooks requires only one complete pass to schedule all activities in the network. This seven step algorithm uses as the determining scheduling factor, the maximum remaining path length following each activity. A high priority for scheduling is associated with a large maximum remaining path length.

The MAP technique developed by Martino is probably the best of these methods. This method seems to be capable of scheduling a wider range of problem types than the other procedures and according to the author (1), "The methods outlined here have been tried and proved during the past six years. While newer acronyms have been added to the lexicology of resource allocation and scheduling procedures, they do not offer any better solution." Also according to Martino, the following priority system is used in this method:

> Priorities are assigned to jobs that have the same starting time. Precedence is assigned according to the following tests and in the order indicated:
>
> Least total float (or criticality measure).
> Larger need of overall resource.
> Larger crew size.
> Sequence code.
>
> Each test is used only if the preceding test results in a tie.

All three of these methods are described excellently in a step by step manner in the three references cited. Another method, not so widely known as the three just discussed, is one developed by Collins (6). This method is taken from a book which was written for people lacking extensive training or experience in project management techniques.

The method developed by Collins is called Activity Time Scheduling. This technique is based on the old bar chart methods of scheduling and seems to have some primitive holdovers from those bar chart methods. This method which was developed for the construction industry, is not explained with enough clarity to permit many people in this industry to use the method without additional training. His scheduling procedure which makes an attempt at leveling resources seems to be of a visual trial and error type.

The literature investigation seemed to indicate that the state of the art of resource allocation is still limited. The search found an equally narrow selection of both manual algorithms and computer programs for resource allocation and leveling. Many computer programs for basic project management were inspected. However, most of these programs only have features which are similar to those of the NASA PERT "C" Computer Program (7) and the TIME-PERT-B 5000 Computer Program (8). The following features which were listed as the major output options of the NASA PERT "C" program are representative of the results given by any of the basic project management techniques:

1. Successor event number sort.
2. Critical path sort.
3. Expected date sort.
4. Organizational sort.
5. Master schedule sort.
6. Latest allowable date sort.

An exceptional article by Phillips (9) lists thirty-six different computer programs which cover every aspect of project management. In addition to these, the Burgess manual algorithm which has been mentioned previously has been developed into a computer program. The Phillips article lists the type of computer equipment required, the source of further program information, the program capacity, and the category of the program for all thirty-six programs listed. A comparison of these programs is made on the basis of fifteen factors among which is resource allocation. The article concludes that very few programs have resource allocation capabilities.

The results of the complete literature search indicated that very few techniques, manual or computerized, exist which have resource allocation capabilities. Of the techniques which have resource allocation or leveling capabilities, only Martino's MAP technique seems to make use of REST-type units.

Since Dr. R. L. Martino is a recognized authority in the field of project management, two passages from correspondence dated June 1, 1966, from him to this writer will be quoted to add validity to the thoroughness of the literature search and the uniqueness of this paper.

> So far as I know the material in my book summarizes the state of the art as it exists at this time. I know of no other papers which contain information that is not embodied in the MAP technique. . .
> For your dissertation I might suggest that you concentrate on an automatic procedure for allocating on a variable crew or variable resource quantity bases [sic] with intermittent starting and stopping of non-critical jobs. While I have touched on this subject in my book, I think that expanding on this particular aspect would be a worthwhile thesis topic and would not include duplication.

Using the stated purpose and following Dr. Martino's suggestions, a new method which is described in the following chapters of this paper was developed and tested.

CHAPTER III

ASSUMPTIONS AND CONSIDERATIONS

Deficiencies of Other Methods

Among the aspects of existing techniques which make them unappealing for use by certain groups is that most of the existing scheduling procedures require a trained technician with an overall understanding of the theoretical aspects of the basic project management methods to gain meaningful results from their application. Potential users of scheduling techniques are often thwarted in their attempts to use existing methods because they lack either the time, money, or educational background to avail themselves of training opportunities and cannot locate or afford the necessary consultants to help them.

Another aspect of many existing methods which limits their usefulness to many users is the feature of time-cost trade-off. A large number of the existing scheduling techniques are based on time-cost trade-off theory which associates a changing cost with a changing activity duration. According to this theory, there is associated with each activity in a network, one time and cost or more probably a set of times and associated costs. Every activity has a normal time and normal cost which are the minimum duration and cost of the activity under normal working conditions. In some special cases it is possible for an activity to have a zero time and/or cost associated with it.

There are several other times and costs associated with each activity in a network (see Figure 1). Crash time and the related crash

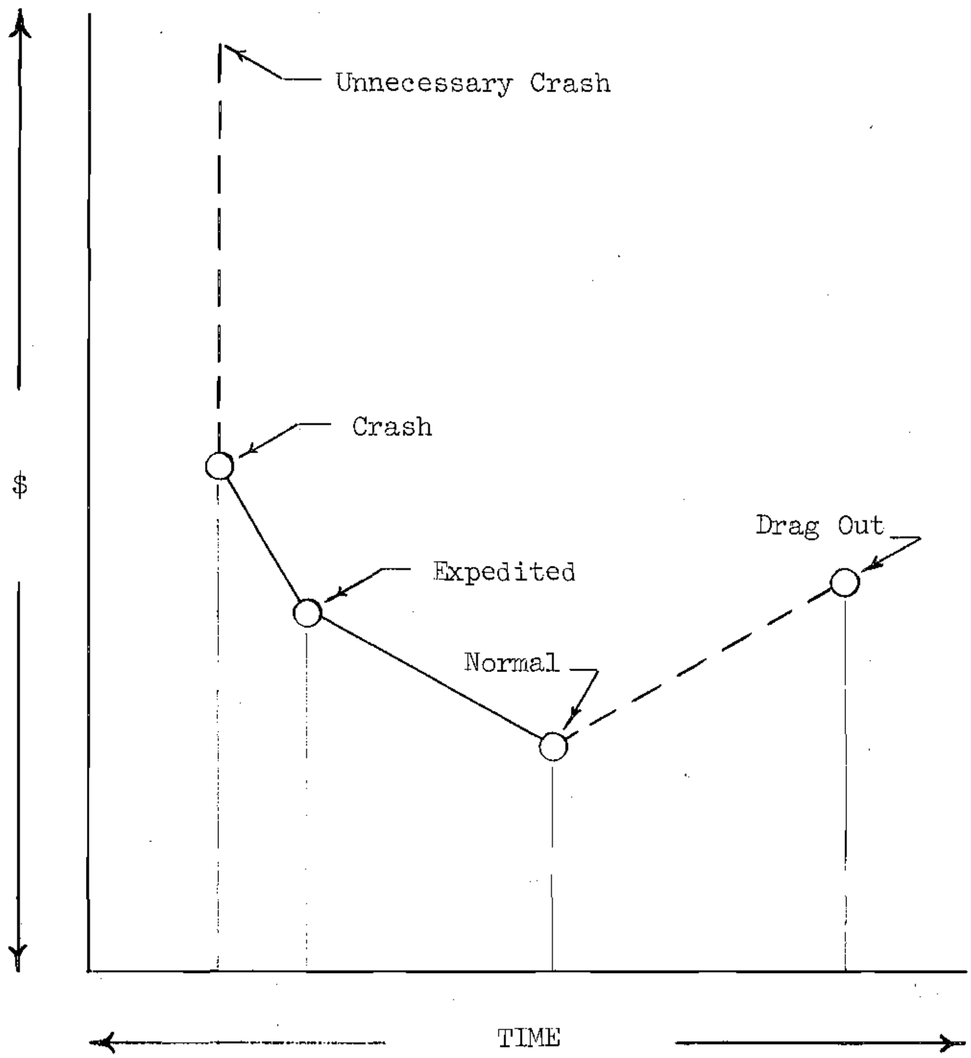

Figure 1. Time-Cost Trade-Off Type Cost Curve

cost are the shortest possible duration of an activity and the minimum cost of performing the activity in that time. Expedited time and cost are a series of possible duration lengths between normal and crash times and the minimum cost for each of the possible times. It is possible that an activity will have only a normal time and cost in which case time-cost trade-off methods are not applicable.

Also associated with each activity of a network is a cost slope. The cost slope of an activity indicates the additional activity cost resulting from a reduction of the activity duration. It is possible to have either a continuous or discrete cost slope. A continuous cost slope indicates that the associated activity may have any duration between normal and crash times. A discrete cost slope is a set of possible activity durations restricted in some way and the additional cost of moving from one possible duration to the next.

Other activity times are unnecessary crash time, which has the same time as crash time but a greater cost due to the use of unnecessary resources, and drag out time, which is a time of greater length than normal time and has a higher cost due to inefficiency.

This time-cost relationship can be caused by several factors. The main cause of this is the added cost associated with the acquisition of additional men or equipment necessary to shorten the duration of the activity. Another cause of this relationship is the learning curve effect which results from the addition of new men to a job. It is also possible that the efficiency will change as the quantities of men or equipment on a job are varied.

It is felt that the time-cost trade-off approach to scheduling would not be applicable to the technique under consideration. Since this technique is assumed to operate on a fixed quantity of men and resources, there will be no additional cost associated with the use of additional resource quantities on a given activity. Also, the learning curve effect is assumed to be invalid since, in many situations, workers are competent in all tasks of their craft and because most jobs have many similarities and vary little among projects. Last, it is assumed that the minimum and maximum permissable resource levels are set so that no inefficiency results from an improper crew size. The assumed time-cost curve for the problem type under consideration is shown in Figure 2. An important difficulty sometimes caused by the application of time-cost trade-off techniques to this type of problem is that the schedule developed by the time-cost trade-off techniques may have resource assignments which exceed the available supply for that time period.

The inevitable inaccuracies of the estimates and the network on which scheduling calculations are based limits the chances that any schedule will be optimal. The fact that most techniques require or recommend frequent updating of the schedule tends to indicate the fluid nature of the problem. These facts make the sophistication of many of the existing methods misleading as to actual value. Because of the uncertainties associated with the raw data, it is felt that it would be permissible to construct a scheduling procedure which at times may result in a schedule efficiency less than that of other methods but which is

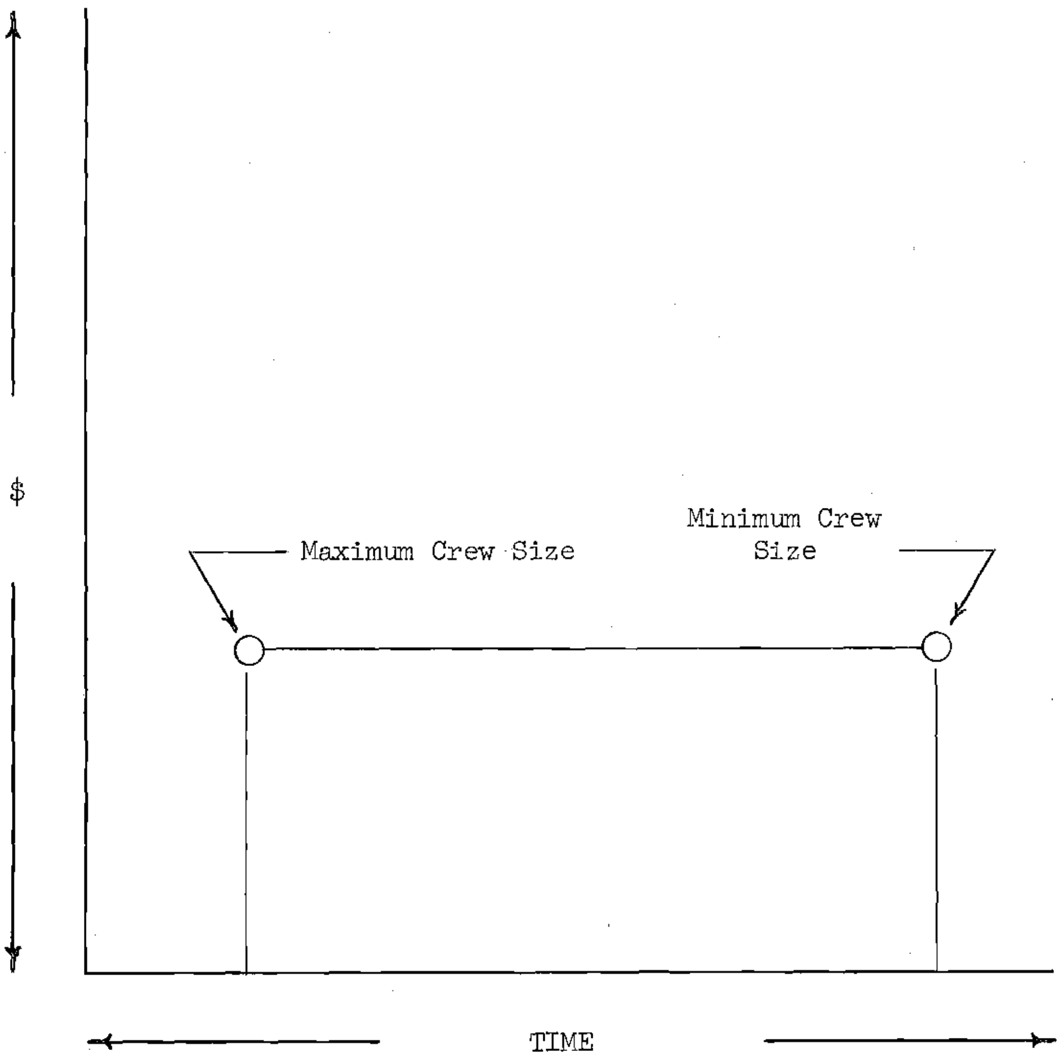

Figure 2. REST Technique Type Cost Curve

easily applied by persons lacking the technical skills necessary for the proper use of existing methods. Schedule efficiency is defined as the ideal project duration divided by the actual duration. In essence, an attempt was made to develop a new resource allocation procedure which would give good answers and would be applicable in the situations previously outlined.

Variables of the Method

The first step in the development was to determine what variables are inherent in the resource scheduling problem. The procedure which was followed began with the listing of factors pertinent to a clear analysis of the problem. This resulted in the following list of considerations:

1. The importance of the ease of application.
2. The possible units of measurement to be used in connection with activity descriptions.
3. The possible types of activities to be considered for scheduling.
4. The number of resource types to be scheduled.
5. The number of projects to share the available resources.
6. The size limitation on the problems to be solved with the procedure.
7. The type of resource availability to be used.

As this list was analysed, a concrete set of specifications for the technique took shape. This specification list, which follows, was the result of a series of decisions in which the choice of variable or assumption always was the choice which seemed most suited to the needs previously stated:

1. Ease of application is very important.

2. Resource-time units will be used to measure the work content of the activities.

3. The proposed resource allocation technique will be capable of scheduling all four activity types.

4. All activities may have intermittent starting and stopping to make possible better schedules.

5. All activities permitting variable resource levels may be assigned any integer number of resource units between the minimum and maximum number permitted.

6. Multi-resource scheduling will be considered but will be limited to problems in which a single activity requires only one resource type while other activities may require different resource types.

7. Only one project at a time will be considered.

8. No size limitations will be imposed.

9. A constant resource availability throughout the duration of the problem will be assumed.

The decision concerning the importance of the ease of application of the planned technique has already been discussed at some length. The choice was made to measure the work content of the activities in resource-time units for several reasons. First and most important was the fact that very little work has been done in this area. It was felt that the information gained from experimentation in this area could be one of the most important results of this work. The decision to use resource-time units was strengthened by the possibility that estimating the work content of an activity using these units might be done easier and more accurately than making two separate estimates, one of resources and the other of time. The old method of estimating work content of activities in which the estimater determined the resource quantity required by each activity often inadvertently resulted in the unconscious attempt of the estimater to schedule resources while defining the activities. This often resulted in a network with incorrect precedence requirements. The use of resource-time units will possibly alleviate this situation by drawing the estimater's attention away from a resource quantity and instead, directing his attention to the actual work content of the activity. Last, it was felt that the calculations in the proposed scheduling procedure would be much easier for untrained persons to perform if one set of values were used instead of the two which would have been necessary with the older methods.

The decision to design a system which would schedule all four activity types resulted from an attempt to impose as few restrictions as necessary upon the procedure. One would be hard put to define an activity which would not be of such types. By permitting all four types, the procedure becomes practical for nearly every type of problem. Also, as in the case of the resource-time units, little attention has been given to activities permitting variable resources and times. The possible value of the experience gained in dealing with this area of project management made this a promising choice.

No existing scheduling procedure considers the possibility of intermittent starting and stopping of activities. For this reason, it seemed desirable to consider this possibility. The possible use of such interruptions permits the formation of a more general procedure than otherwise would have been possible.

It was decided to permit the assignment to activities permitting variable resource levels of any number of resource units between the maximum and minimum permissible values. This assumption made possible a procedure much simpler than would have been possible if only certain resource levels had been permitted.

It was decided to attempt to develop a technique which would consider the possibility of several resource types being available for allocation in a single problem. This seems to be a practical choice since even the smallest company usually makes use of several crafts. By including a plan for the scheduling of several resource types, the entire procedure will increase in usefulness.

It was decided to consider only one project at a time. This does not appear to be a serious limitation since several projects can be combined to form one larger project. This decision also makes possible a much simpler solution to the problem than could have been attained otherwise.

No size limitation as such will be imposed upon the procedure. However, the capabilities of the men or machines performing the procedure may cause practical limits to be set. If such limits are necessary, they will vary according to the capabilities of the scheduler.

It was assumed that there would be a constant resource availability throughout the duration of any problem. This seems to be a very reasonable assumption since with today's labor situation companies will more than likely have a stable work force and will be unable to hire extra workers for short periods of time. Since they have a stable work force, it becomes very important to use as many employees as possible during each time period. If poor scheduling is used, the company will either be forced to pay men for periods not worked or see these men find other employment because their work opportunities were irregular. To prevent either of these situations, an easily applied procedure for scheduling which results in good solutions is needed. This assumption eliminates any consideration of time-cost trade-off since all available resources are being used as much as possible. The only costs which could be considered in connection with this method are the indirect costs of the project and the only way to reduce these costs is to increase the output of the existing resources.

Since the resource supply is limited, this becomes the determining factor of project length. The ideal project length can be found by adding the results of the division of each resource type's work content of the project by that type's availability. Using this figure and the actual duration, the scheduling efficiency can be found by dividing the ideal duration by the actual duration.

It was assumed, and with some justification, that there would be no decrease in efficiency or increase in costs due to intermittent starting and stopping of an activity or changing of the crew size during the duration of an activity.

Description of the Method

After studying the situation which was to be alleviated by the new technique and assimilating the list of assumptions, thought was centered on possible approaches to the development of a suitable allocation method. Ideas, as they presented themselves, were considered, expanded, and tested. This process continued until there existed a technique which was easy to apply and gave reasonable solutions to problems.

An analysis of the new method indicated that only the following four operations would be required:

1. The determination and definition of the activities and their precedence relationships.
2. The determination of all possible paths through the project.
3. The determination of the work content of each path.
4. The use of the actual scheduling algorithm.

Consideration of these four operations indicated that the only one requiring any above-average knowledge or training was the first operation. This operation must be done by someone who is very familiar with the project and who can define all activities found in the project, estimate their work content and determine the precedence requirements associated with the activities. This person also needs sufficient networking knowledge to enable him to assign event numbers to every activity. Once the list of activities, complete with event numbers, has been prepared, a secretary or someone unskilled in the theories and techniques of resource allocation can apply the last three operations and produce a good resource schedule.

CHAPTER IV

RESULTS

Discussion of Results

The initial resource allocation procedure used in the developmental segment of this paper was based upon original ideas of this writer. The initial procedure and succeeding revisions of it were tested and revised until a procedure was had which would fulfill the previously determined specifications. Since this experimental work was in a previously undeveloped area, all revisions were of a trial and error nature based upon further ideas of this writer.

The final procedure which resulted from this experimental work attempts to allocate a fixed resource supply among the activities of a project. This procedure uses the remaining path work content as the criteria for determining the scheduling priority.

This resource allocation procedure was prepared in two forms. First, a manual algorithm was developed and then, based upon this manual algorithm, a computer program was prepared in the Algol computer language.

Manual Algorithm

The manual algorithm and work sheet design which is presented below was tested on numerous example problems and found to be easy to use. The results of this algorithm proved to be more than adequate for the conditions under which it will be used in actual practice. Prior to the presentation of the resource allocation procedure, an algorithm for the determination of all paths of a network will be given.

Path Determination Algorithm

1. Define all activities and determine all dependencies among activities in the proposed project.

2. Assign index numbers (I,J) to each activity of the network letting I represent the begin event of the activity and J represent the end event. It is important that the project starts from only one event and ends with only one. The first event of the network should be numbered zero (0) and no number should be skipped while numbering the events. It is necessary that I always be less than J for a given activity. This assures that the magnitude of event numbers will always be ascending as a path is traced through the network from the start of the project to its finish.

3. Using a form similar to the one illustrated in Figure 3, list all activity index numbers so that the I values will be in ascending order from top to bottom and each subset of index numbers which have the same I value will have the J values in ascending order also.

4. To determine the first path through the network, select the first activity in the list, that is, that activity with I equal to zero which has the smallest J value of the activities in that subset. This is the first activity of the path.

5. Next, go the that subset of activities which has I values equal to the J value of the previous activity. Select the activity from this subset which has the smallest J value. This activity becomes the next activity of the path under consideration. Repeat this step until there is no activity with an I value equal to the J value of the previous

BEGIN EVENT	END EVENT	PATHS 1	2	3	4

Figure 3. Path Determination Work Sheet

activity added to the path.

6. To find each additional path, use must be made of the last previous path. Starting with the last activity of that path and proceeding in the direction of the first activity, examine the index numbers of each activity. If, for any activity (I_1,J_1) in the path under consideration, there exists a second activity (I_2,J_2) such that I_1 equals I_2 and J_1 is less that J_2 then there exists at least one more path which includes this second activity. In applying this step, if a third activity (I_3,J_3) exists such that I_3 equals I_2 and J_3 has a magnitude which is between J_1 and J_2 then replace (I_2,J_2) with (I_3,J_3).

7. The activities of the new path are determined as follows:

a. All activities in the last previous path which come before the activity (I_1,J_1) as determined in the previous step are also found in the new path being developed.

b. The next activity will be the final activity (I_2, J_2) as determined above.

c. The remaining activities of the path are found by using the method described in step 5. Activity (I_2,J_2) should be used as the starting point for this application of the method.

8. All paths have been found when step 6 fails to locate an activity (I_2,J_2) as described in that step.

An example of this algorithm is presented in Figure 4. Since it is often necessary to determine the number of paths of a network and their composition, this algorithm can be of value in situations not associated with the REST Technique developed in this thesis. Since a

BEGIN EVENT	END EVENT	PATHS			
		1	2	3	4
0	1	X			
0	3		X	X	
0	6				X
1	2	X			
2	5	X			
3	4		X		
3	7			X	
4	5		X		
5	8	X	X		
6	7				X
7	8			X	X

Figure 4. Completed Path Determination Work Sheet

great deal of time can be required to determine the paths of a relatively small network, a program in the Algol computer language was prepared which computerizes this algorithm. This program, along with instructions for its use can be found in Appendix A.

<u>REST Technique</u>

Before beginning the REST Technique, it is necessary to use the path determination algorithm described above. In this case, it is necessary to substitute the expanded work sheet illustrated in Figure 5 for the one shown in Figure 3. As soon as the paths of the network have been determined, work can begin on the following steps:

1. Estimate, in REST units, the work content of each activity.

2. If an activity consists of a forced delay or waiting period, there will be no REST units associated with it since no resource units are required. In lieu of REST units, assign to such activities a value equal to the number of time periods involved in the delay. Accompany this value with an 'X' so that it can be differentiated from actual REST units.

3. Dummy activities requiring neither time nor resources should simply be assigned zero REST units.

4. On the work sheet shown in Figure 5, form, to the right of the grid indicating all paths through the network, a column containing the work content of each activity as determined in steps 1, 2, and 3.

5. To the right of the column containing activity work contents, set up two columns. The first column should contain the minimum allowable resource requirement for one time period for each activity while

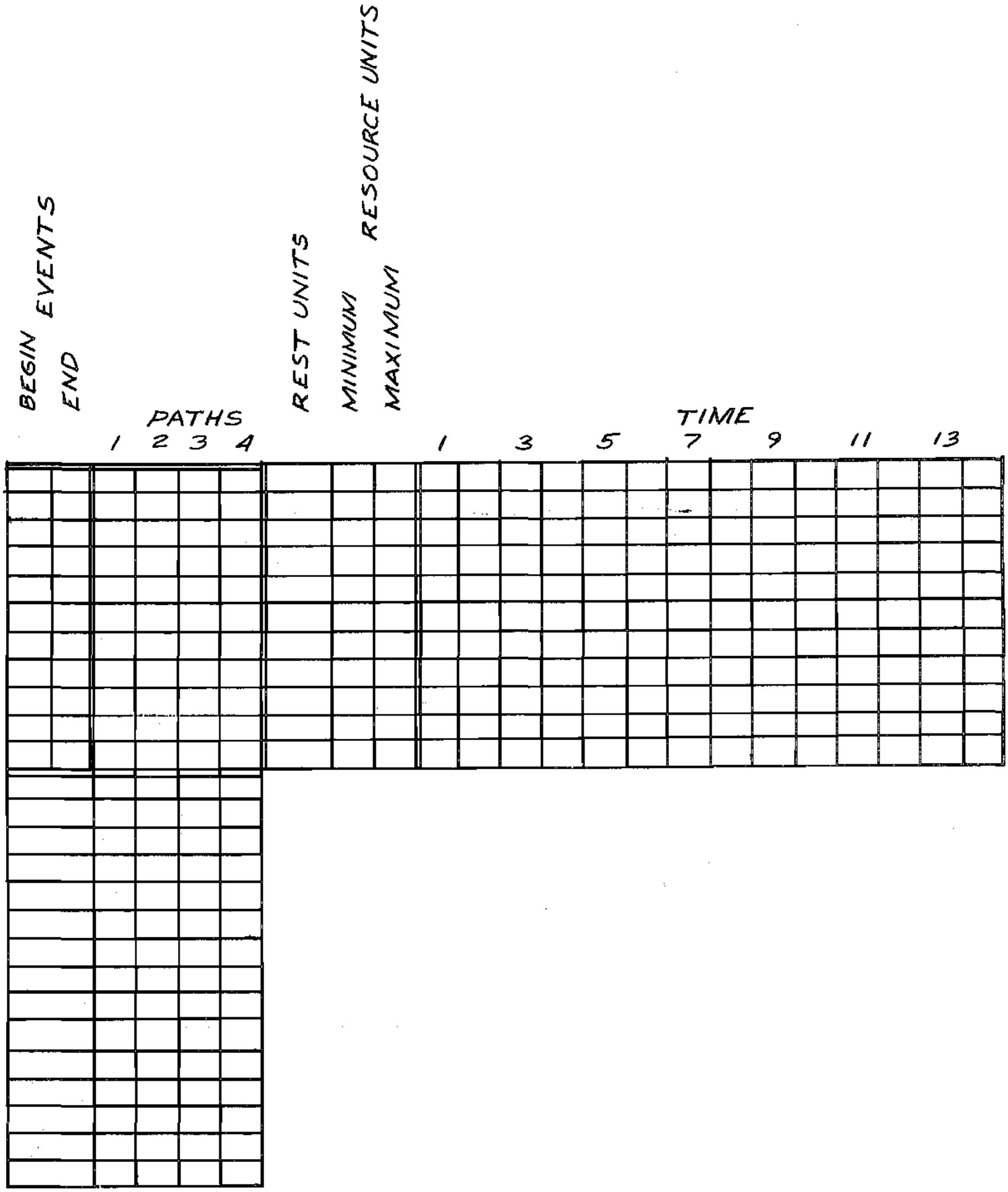

Figure 5. Rest Work Sheet

the second should contain the maximum allowable resource requirement for one time period for each activity.

6. Again, to the right of the columns already completed, set up a series of columns. The number of columns will depend upon the size of the problem. Each column represents one time period in the project duration and these columns will be used to record the resource quantities allocated to each activity during each time period.

7. Below the path columns formed by the path determination algorithm, set up a row in which to record the work content of each path. In computing these quantities, add to the REST units of each path the product of the time periods of forced delay in that particular path times the total resources available in one time period.

8. A grid is then set up below the work content row. In this grid will be recorded the remaining work content of every path at the end of every time period.

A completed work sheet is shown in Figure 7. The network on which it is based is shown in Figure 6.

This iterative portion follows:

9. Examine the total remaining work content of each path and select the largest.

10. In the path designated in step 9, select the uppermost activity which has not been completely scheduled.

11. From the resources available for the time period under consideration, assign resource units to this activity as governed by the following if all predecessor activities have been completely scheduled:

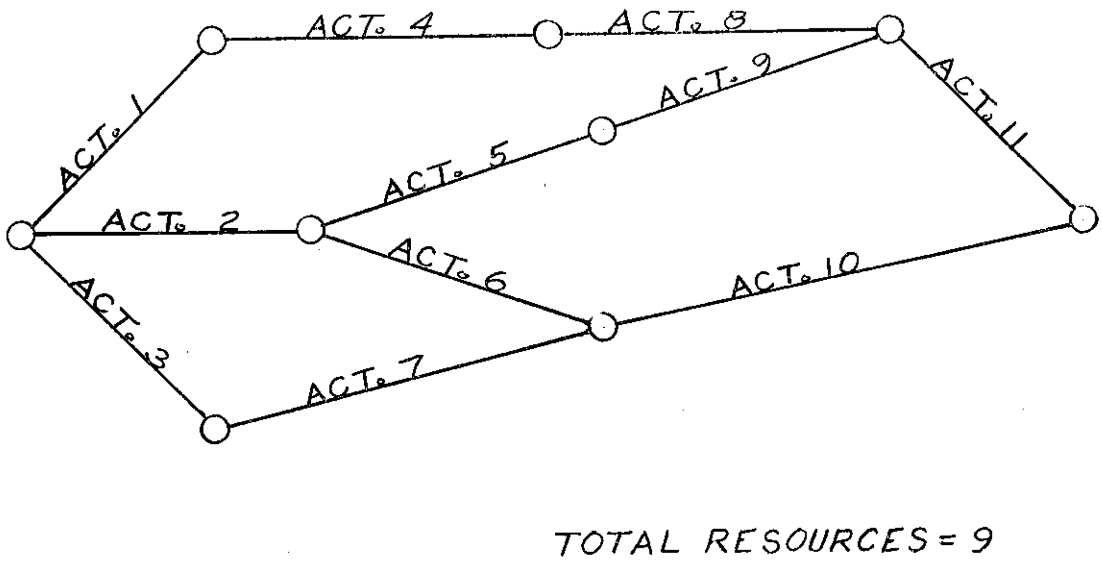

Figure 6. Network

EVENTS: BEGIN	EVENTS: END	PATHS: 1	2	3	4	REST UNITS	RESOURCE UNITS: MINIMUM	RESOURCE UNITS: MAXIMUM	TIME: 1		3		5		7		9		11		13	
0	1	X				6	1	3	3	3												
0	3		X	X		12	6	6	6	6												
0	6				X	4	2	4				4										
1	2	X				13	2	8			8	5										
2	5	X				4	4	4					4									
3	4		X			5X	0	0			S	S	S	S	S							
3	7			X		0	0	0		S												
4	5		X			32	4	9								9	9	9	5			
5	8	X	X			3X	0	0												S	S	S
6	7				X	15	3	5					5	5	5							
7	8			X	X	10	5	5												5	5	
		50	116	22	29																	
		47	110	16	29																	
		44	104	10	29																	
		36	95	10	29																	
		31	86	10	25																	
		27	77	10	20																	
		27	68	10	15																	
		27	59	10	10																	
		27	50	10	10																	
		27	41	10	10																	
		27	32	10	10																	
		27	27	10	10																	
		18	18	5	5																	
		9	9	0	0																	
		0	0																			

Figure 7. Completed Rest Work Sheet

a. Let:

R = Total Remaining Work Content of the Activity

S = Available Resource Supply

M = Maximum Allowable Resource Requirement of the Activity

Min = Minimum Allowable Resource Requirement of the Activity

b. If $S \leq Min$ for the activity in question, then assign zero resource units to the activity. If this condition is not met then one of six other possible conditions must be met.

c. The six scheduling conditions are shown in Figure 8. In this figure, the opposites of each of the conditions are shown to be one of the other five conditions.

i. If condition (1), $R \leq S \leq M$, or condition (2), $R \leq M < S$, is encountered, schedule for that activity a resource quantity equal to R.

ii. If condition (3), $S < R \leq M$, is encountered, schedule for that activity a resource quantity equal to S if $R - S \geq Min$. If $R - S < Min$, let $Z = S - Y$ where Y is initially set equal to 1 and then increased in steps of 1. Schedule for that activity a resource quantity equal to Z if $R - Z = Min$. Schedule for that activity a resource quantity equal to zero if $Z < Min$.

iii. If condition (4), $S \leq M < R$, is encountered, schedule for that activity a resource quantity equal to S if $R - S \geq Min$ and $[R - S / Min] \geq R - S / M$. If $R - S < Min$ or $[R - S / Min] < R - S / M$ let $Z = S - Y$ where Y is initially set equal to 1 and then increased in steps of 1. If $R - Z \geq Min$ and $[R - Z / Min] \geq R - Z / M$, schedule for

- $R \leq S \leq M$ (1)
  - $R > S \leq M$
    - $S \leq M$ &
    - $S < R$ (4)
  - $R \leq S > M$
    - $R \leq S$ &
    - $M > S$ (6)
  - $R > S > M$ (5)

- $R \leq M < S$ (2)
  - $R > M < S$
    - $M < S$ &
    - $M < R$ (5)
  - $R \leq M \geq S$
    - $R \leq M$ &
    - $S \leq M$ (1)
  - $R > M \geq S$ (4)

- $S < R \leq M$ (3)
  - $S \geq R \leq M$
    - $R \leq M$ &
    - $R \leq S$ (1)
  - $S < R > M$
    - $S < R$ &
    - $M < R$ (5)
  - $S \geq R > M$ (6)

- $S \leq M < R$ (4)
  - $S > M < R$
    - $M < R$ &
    - $M < S$ (5)
  - $S \leq M \geq R$
    - $S \leq M$ &
    - $R \leq M$ (1)
  - $S > M \geq R$ (2)

- $M < S < R$ (5)
  - $M \geq S < R$
    - $S < R$ &
    - $S \leq M$ (4)
  - $M < S \geq R$
    - $M < S$ &
    - $R \leq S$ (6)
  - $M \geq S \geq R$ (1)

- $M < R \leq S$ (6)
  - $M \geq R \leq S$
    - $R \leq S$ &
    - $R \leq M$ (1)
  - $M < R > S$
    - $M < R$ &
    - $S < R$ (5)
  - $M \geq R > S$ (3)

Figure 8. Possible Scheduling Conditions

that activity a resource quantity equal to Z. If $Z < Min$, schedule for that activity a resource quantity equal to zero.

iv. If condition (5), $M < S < R$, or condition (6), $M < R \leq S$, is encountered, schedule for that activity a resource quantity equal to M if $R - M \geq Min$ and $[R - M / Min] \geq R - M / M$. If $R - M < Min$ or $[R - M / Min] < R - M / M$ let $Z = S - Y$ where Y is initially set equal to 1 and then increased in steps of 1. If $R - Z \geq Min$ and $[R - Z / Min] \geq R - Z / M$, schedule for that activity a resource quantity equal to Z. If $Z < Min$, schedule for that activity a resource quantity equal to zero.

v. If the activity is of the type which requires a fixed time but no resources as in a forced delay or which requires neither time nor resources, assign a zero resource quantity to it.

A flow diagram is presented in Figure 9 in which an effort is made to simplify the above discussion.

12. Record the resource assignment in the proper activity-time block on the REST work sheet. For the activity types mentioned in section iv above, record an 'S' in the appropriate block to indicate that the activity has been scheduled.

13. Subtract this assigned resource quantity from the total remaining REST units of each path containing the activity. Record these new totals in the next row provided for path totals. If the activity in question is of the forced delay type, subtract a quantity equal to the total resource availability for one time period from the total remaining REST units of each path involved.

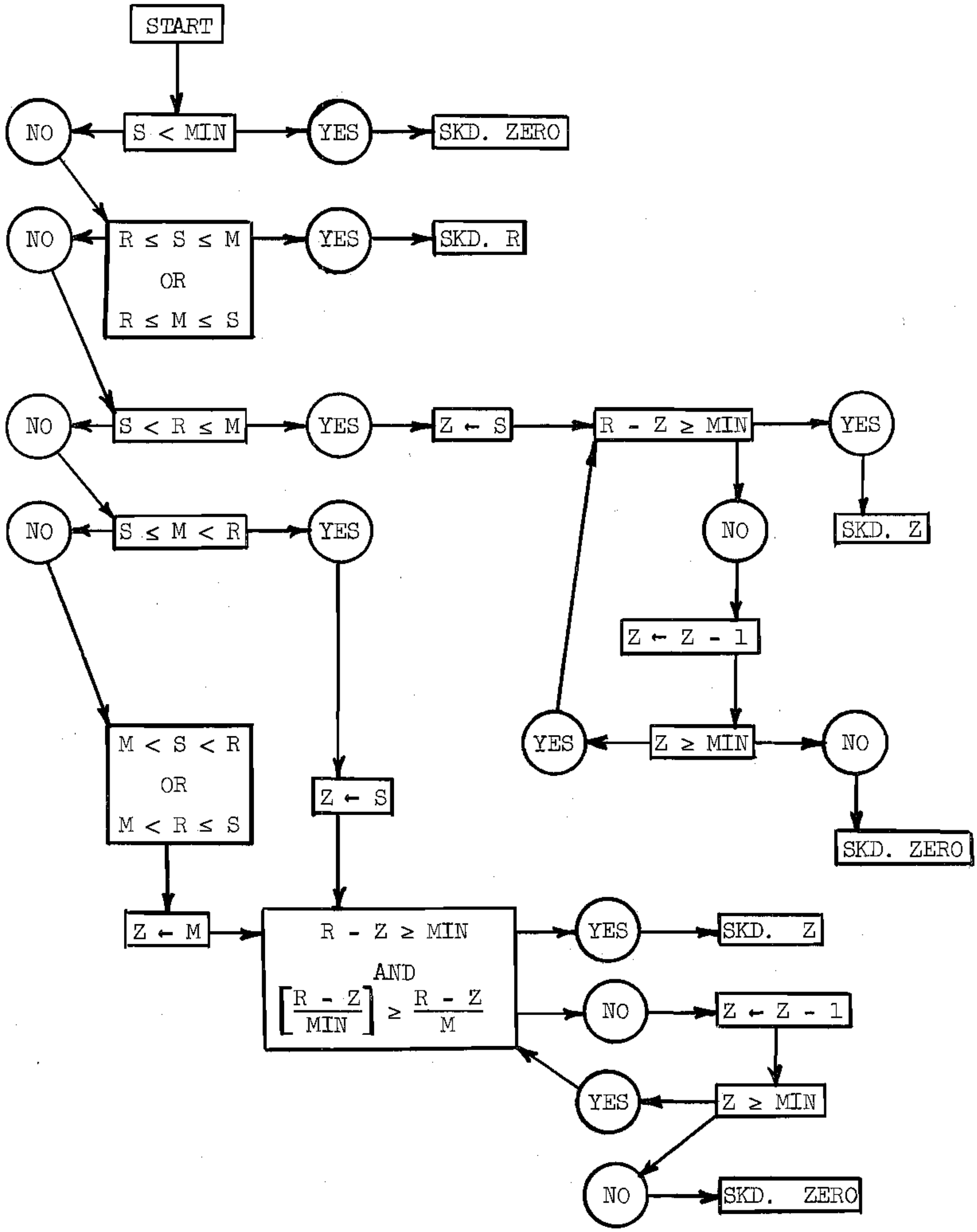

Figure 9. Flow Diagram of Decisions

14. If any resources remain unallocated, go to the next highest ranked path which has not been affected during this iteration. Return to step 10. Continue this procedure until all rescources have been assigned or until all paths have been considered - whichever happens first. If the resource supply is depleted before all paths have been considered, check the remaining paths to be sure that all forced delay and dummy activities which can be scheduled are scheduled.

15. When step 14 has been completed, bring down all path totals unaffected in this iteration into the row with the new totals formed in step 13.

16. Return to step 9 and begin the next iteration.

17. Continue these iterations until all activities have been completely scheduled.

Reasoning in Algorithm

Although the complete algorithm has been presented, it may be wise to discuss further the bases for several of the procedures. The choice of the weighting factor to be used with forced delays is based on the reasoning that the effect of a forced delay, one time period in duration, upon a path length is the same as that of an activity, one time period in duration, which requires the entire resource supply. Both activities add one time period to the duration of the path and with the conditions of the network remaining unchanged, there is no way to shorten either activity. If no weighting system were used on such activities, the REST Technique would indicate that the forced delay had no effect on the network. However, by using the weighting system previously described, the forced delay is shown to be significant and, in fact, is given stress equal to that given the activity which requires the entire resource supply. This is very important because the REST Technique uses remaining work content of the paths of the network to determine the scheduling priority. This weighting system, in effect, gives a work content value to the forced delay.

A short discussion of some of the reasoning associated with the six scheduling conditions has already been given. The third facet of this algorithm which is worthy of further discussion is the greatest integer function used in connection with the scheduling of resources under conditions (4), (5), and (6). Without the characteristic furnished by this greatest integer function, the REST Technique could quite possibly make a series of allocations to an activity such that the remaining activity work content could never be completely scheduled. As an example, consider an activity having fifteen REST units associated with it. This activity also has a maximum allowable resource requirement of five units and a minimum allowable resource requirement of four units. The REST Technique could conceivably schedule four resource units in three time periods. This would leave three REST units to be scheduled. Since the minimum allowable resource requirement is four units for this activity, these remaining REST units can never be scheduled. Therefore, an evident need exists for a method to insure against such possibilities.

The reasoning used in the development of this test is as follows:

After every resource assignment for an activity, it is necessary that the remaining work content, RWC, of the activity is of the form:

$$RWC = A(Min) + B(Max) + C \qquad (1)$$

where A, B, and C are non-negative integers and

$$Min \le C \le Max.$$

This form is assured if

$$\left\lceil \frac{RWC}{Min} \right\rfloor \ge \frac{RWC}{Max}. \qquad (2)$$

If equation (2) holds true then

$$\left\lceil \frac{RWC}{Min} \right\rfloor = I \ge \frac{RWC}{Max},$$

therefore,

$$\frac{RWC}{Min} \ge I \ge \frac{RWC}{Max} ,$$

$$RWC \ge I(Min) ,$$

$$RWC \le I(Max)$$

and

$$I(Min) \le RWC \le I(Max)$$

where I is a non-negative integer.

The lemma, then, can be stated as follows:

If positive integers (Max, Min, I, and RWC) exist such that

$$\text{Min} \leq \text{Max}$$

and

$$I(\text{Min}) \leq \text{RWC} \leq I(\text{Max})$$

then non-negative integers (A, B, and C) exist such that

$$\text{RWC} = A(\text{Min}) + B(\text{Max}) + C$$

where

$$\text{Min} \leq C \leq \text{Max}.$$

Proof: Since

$$I(\text{Min}) \leq \text{RWC}$$

can be written as

$$(I - 1)\text{Min} + \text{Min} \leq \text{RWC} ,$$

there exists an integer

$$C_1 \geq \text{Min}$$

such that

$$(I - 1)\text{Min} + C_1 = \text{RWC} .$$

If

$$C_1 \leq \text{Max} ,$$

then the lemma is proven. If, however,

$$C_1 > \text{Max},$$

then C_2 exists such that

$$(I - 2)\text{Min} + \text{Max} + C_2 = \text{RWC}$$

and

$$C_1 \geq C_2 \geq \text{Min}.$$

Assume that

$$C_2 > \text{Max} ;$$

then a C_3 can be found such that

$$(I - 3)\text{Min} + 2\text{Max} + C_3 = \text{RWC}$$

where

$$C_2 \geq C_3 \geq \text{Min} .$$

This process can be continued (I - 1) times before A becomes zero. Assume, that after (I - 1) cycles, RWC still cannot be written in the form of equation (1). Then

$$(0)\text{Min} + (I - 1)\text{Max} + C' = \text{RWC}$$

where

$$C' > \text{Max} .$$

Therefore

$$(I)\text{Max} + C'' = \text{RWC}$$

where

$$C'' > 0 ,$$

but

$$(I)\text{Max} \geq \text{RWC}$$

and this is a contradiction. Therefore, non-negative integers (A, B, and C) exist such that

$$\text{Min} \leq C \leq \text{Max}$$

and

$$\text{RWC} = A(\text{Min}) = B(\text{Max}) + C.$$

Final Remarks

When working a problem manually using the REST Technique, it is suggested that forms similar to those described in this paper be used. The exact size of the forms needed will naturally depend upon the size of the problem being considered. For small problems, the one work sheet will be sufficient but for larger problems it may be necessary or, at least, desirable to divide portions of the work sheet among several sheets. The work sheets used with this algorithm can take many forms but experience has shown the design presented in this paper to be a good one. A possible expanded work sheet design for larger problems is illustrated in Figures 10, 11, and 12.

Users of this manual algorithm are cautioned that there are two sources of frequent human error in this algorithm. Experience has shown that errors in subtraction are easily made because of the large number of subtraction operations in a normal problem. The second common error is a failure to select for consideration the path with highest priority. To protect against carrying subtraction mistakes through a large number of calculations, it is suggested that the actual remaining path work contents to be periodically calculated and compared with the recorded values. However, a conscientious and careful worker is the best protection against such mistakes. It is also important that the maximum allowable resource requirement never exceeds the total available resources.

Multi-Resource Allocation

Multi-resource allocation, the scheduling of two or more resource types among the activities of a project, is a very desirable characteristic of any resource allocation technique. A moment's reflection on practical problems brings the realization that many, if not most, actual problems are of this multi-resource type. For this reason, an attempt was made to adapt this technique to the multi-resource pro-

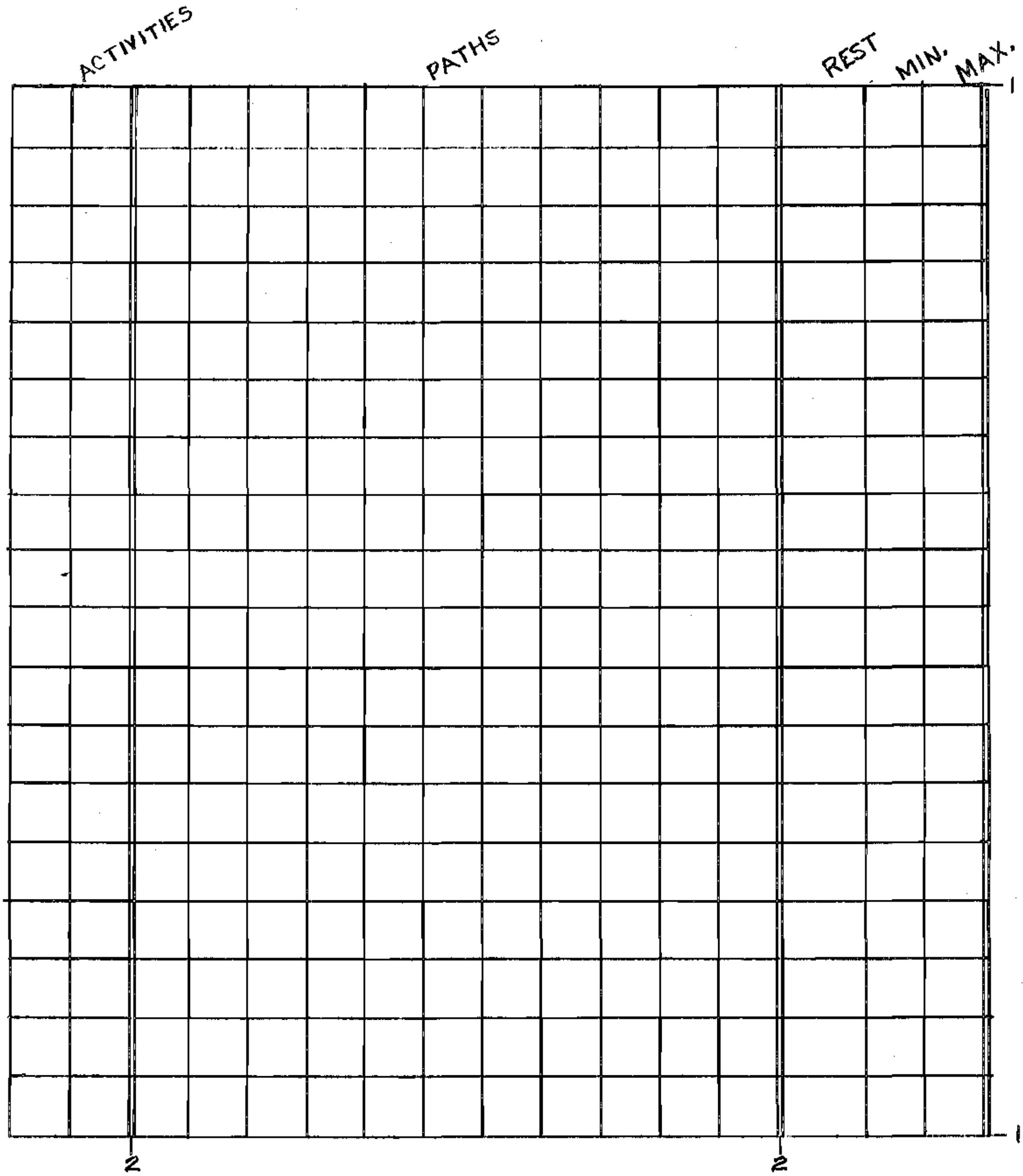

Figure 10. Possible Expanded Work Sheet

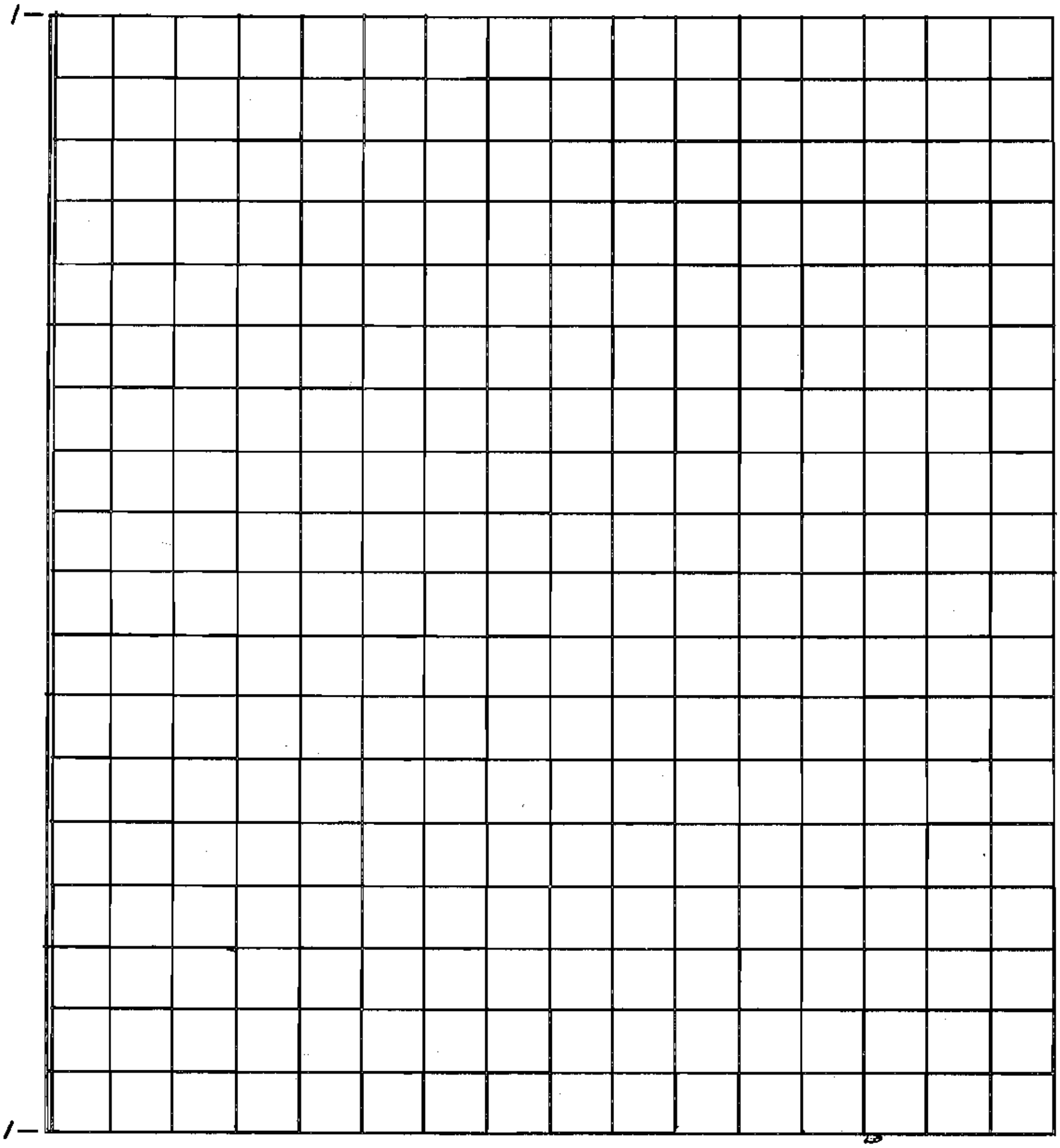

Figure 11. Possible Expanded Work Sheet

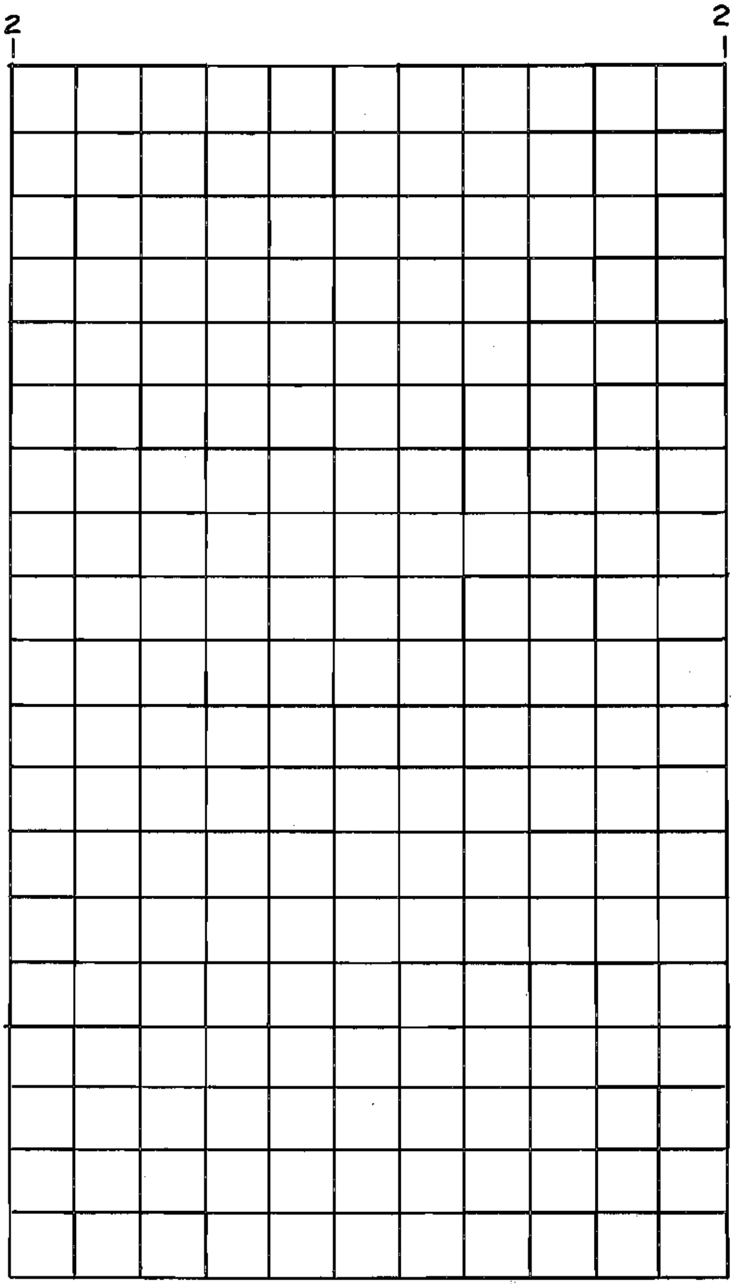

Figure 12. Possible Expanded Work Sheet

blem.

It was desired that the manual algorithm and the computer program give identical solutions when both are correctly applied. Since the storage capacity of the computer was a limiting factor in the computer program, it was not feasible to convert the program to handle more than one resource type. In order to maintain the relationship between manual algorithm and computer program, the algorithm, as presented, was not altered for the multi-resource problem either.

An effort was made later to determine the necessary changes in the manual algorithm to give it multi-resource capabilities.

The REST algorithm was revised for multi-resource scheduling has certain limitations. The main limitation is that it will handle only activities requiring one resource type. As long as any one activity requires only one resource type, the technique will schedule more than one resource type within a project. This restriction may not be as severe as it may seem when it is realized that most activities requiring several resource types can be separated into several simultaneous activities, each of which requires only one resource type.

The only revision in the basic REST Technique are:

1. It is first necessary to enlarge the work sheet so that it will accommodate the enlarged problem. This is accomplished easiest by adding a column next to the one reserved for activity work content. In this column is placed the resource type of each activity.

2. In determining the weighting factor for forced delay or weighting period activities, it is necessary to multiply the number of time periods in the delay by the total of all resources available. This is

necessary since this activity has the same effect on path duration as does a series of activities which require the total available resource supply for the same time period.

3. To determine a path total work content, add the work content of all activities contained in that path regardless of resource type required by each activity. All resource types should be weighted equally.

4. Use the normal method of determining scheduling priorities. As a path is selected, proceed to the uppermost unscheduled activity. Determing its resource type and if any units of that resource type are unscheduled at this point, proceed with the normal tests to determine the quantity of that resource to be shceduled. If no resource units of this type remain, go to the next path. In each time period, continue until all paths have been considered or until all resource units have been scheduled in which case continue the search for dummy and forced delay activities which can be scheduled during this time period.

5. Be sure that the correct resource type is assigned to all activities.

Computer Program

A program written in the Algol computer language for the Burroughs 5500 digital computer was prepared to perform the series of operations described in the manual REST algorithm. This program, which can be found in Appendix B, will produce results identical to those of the manual algorithm when both are properly executed. The details of the use of this program can be discussed best in two sections. The first section will describe the preparation of the input while the second section will describe the interpretation of the output.

Input

The input information describes to the computer the particular project under consideration. The input is processed by the program to produce the output. In order for the program to function properly, it is necessary that a complete set of data arranged in the prescribed sequence be placed immediately behind the program deck. It is imperative that all numbers be right justified on all data cards.

The first data card should have the following form:

columns	content
1-12	a one to twelve character alpha-numeric project identification code
14-17	a one to four character numeric run number to identify different runs of the same project

The second card should contain in columns 1 through 72 any alpha-numeric comment which might be desired. A semicolon must not appear on this card.

columns	content
1-4	the total number of activities in the project
6-9	the highest event number in the project
11-14	the number of resource units available at the beginning of each time period

The remaining data cards will carry information about the individual activities of the project and there should be one card for each activity. These cards should have the following form:

columns	content
1-3	the begin event number of the activity
5-7	the end event number of the activity
9-50	a one to forth-two character alpha-numeric description of the activity (This description is not required.)
54-56	the number of REST units in the activity
58-60	the number of delay or waiting periods in the activity

62-64 a 1 if the activity has no duration or resource requirement, otherwise leave blank

66-68 the minimum number of resource units which can be assigned to the activity during one time period

70-72 the maximum number of resource units which can be assigned to the activity during one time period

It is important to notice the upper limit of several variables which control the size of the projects which can be processed by this program. This program can process a project with a maximum of 100 activities, 100 unique paths, 100 activities per path, and 100 time periods. This capability can be enlarged if necessary by changing the integer array declarations. It should also be noticed that the size of the project is further controlled by the number of characters allowed for each type of input data. It must be remembered also that the project network must have only one start event and only one finish event. The start event number must be zero and the event numbers along each path must be ascending from start to finish.

Output

The first segment of the output will contain the project identification code, the run number, and the comment which appeared on the second data card. The next portion of the output will reproduce the information given on the third data card. N will be the total number of activities in the project, X will be the highest event number in the project, and RESOURCE will be the number of resource units available at the beginning of each time period.

The third section of the output will be an ordered list of the activities. Both alpha-numeric description and event numbers will be given.

The last segment of the output will be one or more arrays indicating the exact allocation of the resource among the activities. Each array will be divided into 25 vertical time columns and N horizontal activity rows. Each time column will be labelled and a note will be printed with each array telling which multiple of one hundred should be added to the time given on the array in order to have the correct time in relation to the actual start of the project. Each activity row will be labelled with the activity event numbers. The interior of the array will be composed of the resource quantities assigned to each activity during each time period. An *S* appearing in an array will indicate that the activity was scheduled but no resources were required.

Comparison of Results

Six basic example problems were considered. The results of this study can be found in Appendix A. Each basic problem was numbered and each variation of one of the basic problems was lettered. Since most of the original problems had the maximum and minimum allowable resource requirements equal to each other, the variations of the basic problems consisted of changing the minimum allowable resource requirements of some activities to make the problems more suited to the REST Technique.

Including the variations of the basic problems, eleven different problems were worked with the REST Technique. Four of the problems were worked with other methods also. It should be noted that although the scheduling efficiency of the REST Technique is sometimes less than that of other methods, the efficiency is still reasonable. It should

Table 1. Comparison of Methods and Results

Network Number	Method Used	Ideal Time	Actual Time	Efficiency	Days Over Ideal Time	Activities in Network
1	REST	14	17	82 %	3	9
1	MAP	14	17	82 %	3	9
1 b	REST	14	15	92 %	1	9
2	REST	19	21	91 %	2	16
2	MAP	19	19	100 %	0	16
2 b	REST	19	20	95 %	1	16
2 c	REST	19	21	91 %	2	16
3	REST	35	36	97 %	1	15
3	MAP	35	36	97 %	1	15
3 b	REST	35	36	97 %	1	15
3 c	REST	35	36	97 %	1	15
4	REST	10	15	67 %	5	11
4	Burgess	10	15	67 %	5	11
5	REST	40	41	98 %	1	27
6	REST	12	13	92 %	1	10

also be realized that the ideal duration used in the efficiency calculations is not always attainable. It is quite possible that some of the efficiencies attained for certain problems by some of the methods are in reality the best possible efficiencies. The efficiency, then, is only an indication of the worst possible deviation from the best possible schedule. The set of examples which were considered indicated that the efficiency of the REST Technique increases as the number of activities in the network increases.

Control and Updating

Although this technique has no sophisticated dynamic control system associated with it, the REST Technique does offer the same basic control features that most other project management methods possess. The nature of the problem involved makes impractical a dynamic feedback loop control system. The project manager is given some control over the project by means of a periodic comparison of actual results with the estimated results indicated in the network and associated calculations. The control in this system results in adjustments made by the project manager when deviations are found between actual and estimated conditions.

This control is periodic in nature and is based on the perceptiveness of the project manager in comparing and adjusting actual and estimated accomplishment levels to give a better system.

When a project is updated as is often necessary when actual work deviates from scheduled work, the REST Technique offers a simple method. If frequent updating is necessary, there is often no need of updating the entire network at each updating period. Instead, it is often desirable to update only over a short period thus eliminating the wasted time and effort associated with updating that portion of the network which will be revised again before any use will be made of it. The REST Technique, unlike other methods which require a complete set of calculations to get a schedule, will permit the scheduling to terminate in any time period desired without affecting the accuracy of the answers already obtained. This updating feature makes this technique particularly suited for situations which require a large number of short range updates due to the length and fluid nature of the project.

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

An investigation of the resource allocation technique developed in this thesis leads to the conclusion that the REST Technique meets all of the specifications for which it was designed. Although the REST Technique satisfied all of the related specifications, no claim of optimality is made. The success of this initial venture in scheduling using REST units indicates that further developmental work in this area would be very beneficial. It is felt that by using the REST Technique as a foundation for further research, a broader and more useful technique can be developed to better fulfill the specifications of a resource allocation procedure as discussed in this thesis.

One last conclusion as to the usefulness of this technique leads to the recommendation that the REST Technique be used in connection with small construction firms and other businesses of like nature which operate with a limited supply of men and equipment and which are financially unable to afford the training costs or consulting fees necessary to benefit from other existing resource allocation methods. The REST Technique seems to be particularly suited to their needs.

The work done in connection with this thesis has suggested the following areas as being particularly worthy of further investigation:

1. It is felt that a study of the REST Technique in actual industrial situations would be worthwhile. Additional testing is needed

to determine if the assumed advantages of the REST Technique are actually advantages in the situations under consideration.

2. An extension of the REST Technique to make it a more competent method for multi-resource scheduling would be a very worthwhile endeaver since most problems of the type considered by the REST Technique are of the multi-resource type. A logical place to begin this extension is with a study of the feasibility of modifying the method used to determine scheduling precedence.

3. The development of a multi-project scheduling technique based upon the theories used in the REST Technique hold promise.

4. A promising modification of the REST Technique seems to be one which would permit the use of predetermine varying resource supplies. If such a method is based on the REST Technique, thought will have to be given to the weighting factor used in connection with activities which are forced delays or waiting periods. Since the present method uses the total resources availability as part of the weighting factor, a new weighting factor will be necessary.

It is hoped that further investigation into the ideas presented in this thesis will be carried out in the near future with outstanding results.

APPENDIX A

APPENDIX A

In order to properly use the Path Determination Computer Program, certain facts concerning its use should be known.

Input

The input for the program should be in the following sequence:

1. The highest event number.
2. The number of activities.
3. The various sets of event numbers which designate the activities.

These event numbers should begin with zero and increase from start to finish of the network. Also no numbers in this series should be skipped. These pieces of data may be placed in any column between one and seventy-two of the data cards as long as each piece of data is followed immediately by a coma. This is the free field input form.

Output

The output of the program will consist of a list of numbered paths. Each path will be presented as a list of the event number sets found in that path.

Both the input and output of this program are very simple to understand and use. Without any alterations, this program will handle 400 activities and 200 paths of 200 activities each. To enlarge this program further, changes must be made in several of the array and format declarations in the program. The actual program follows:

```
BEGIN  COMMENT
                    NETWORK PATH DETERMINATION METHOD
              AN ALGOL PROGRAM FOR THE BURROUGHS B5500 COMPUTER
        C. A. MCNEILL - GEORGIA TECH - ATLANTA, GEORGIA - JUNE, 1966
                                                                            ;
          FILE IN  CARD   (2,10)                                            ;
          FILE OUT LINE  6(2,15)                                            ;
          INTEGER  A,B,C,D,I,J,G,N,U,Y,X                                    ;
     INTEGER  ARRAY   AEVENT,BEVENT[0:400] , P[0:200,0:200]                 ;
      FORMAT OUT  FMA(////X30,"PATH NUMBER ",I4///),
                  FMB(X35,I4," - ",I4)                                      ;
          LIST   LT1(FOR I ← 1 STEP 1 UNTIL N DO [ AEVENT[I],BEVENT[I]])    ;
          LABEL   L1,L2,L3,L4,L5,L6                                         ;
           WRITE(LINE[NO])                                                  ;
           READ(CARD,/,X,N)                                                 ;
           READ(CARD,/,LT1)                                                 ;
           CLOSE(CARD,RELEASE)                                              ;
           FOR I ← 1 STEP 1 UNTIL 200 DO
           FOR J ← 1 STEP 1 UNTIL 200 DO
              P[I,J] ← 0                                                    ;
           A ← 0 ;            B ← U ← Y ← 1                                 ;
L1:        FOR I ← 1 STEP 1 UNTIL N  DO
    BEGIN
           IF  A = AEVENT[I] AND B = BEVENT[I]  THEN
      BEGIN
             P[Y,U] ← I                                                     ;
             IF  BEVENT[I] = X  THEN  GO TO L2 ELSE
         BEGIN
               U ← U + 1 ;  A ← B ; B ← B + 1  ; GO TO L1
         END                                                                ;
      END                                                                   ;
    END                                                                     ;
             IF  B = X  THEN  GO TO L2 ELSE
         BEGIN
                 B ← B + 1 ;  GO TO L1
         END                                                                ;
L2:              C ← AEVENT[P[Y,U]] ; D ← BEVENT[P[Y,U]] + 1                ;
```

```
L6:         FOR I ← 1 STEP 1 UNTIL N  DO
    BEGIN
                IF  C = AEVENT[I] AND D = BEVENT[I]  THEN
       BEGIN
                G ← 1 ;    GO TO L3
       END                                                        ;
    END                                                           ;
              IF D < X  THEN  BEGIN D ← D + 1 ; GO TO L6  END     ;
              IF C ≠ 0  THEN  BEGIN U ← U - 1 ; GO TO L2  END     ;
              A ← 0  ; B ← B + 1                                  ;
              IF B = X + 1  THEN   GO TO L4                       ;
            FOR I ← 1 STEP 1 UNTIL N  DO
     BEGIN
              IF  A = AEVENT[I] AND B = BEVENT[I]  THEN
         BEGIN
                U ← U + 1 ;  GO TO L1
         END                                                      ;
     END                                                          ;
              GO TO L1                                            ;
 L3:        IF  C = AEVENT[P[Y,G]]   THEN
      BEGIN
             A ← C ; B ← D ; Y ← Y + 1 ; U ← G     ; GO TO L1
      END                                                         ;
               P[Y+1,G] ← P[Y,G]  ;  G ← G + 1                    ;
             GO TO L3                                             ;
 L4:        FOR I ← 1 STEP 1 UNTIL Y  DO
    BEGIN
                WRITE(LINE,FMA,I)                                 ;
                J ← 1                                             ;
     L5:        IF BEVENT[P[I,J]] ≠  0   THEN
        BEGIN
                  WRITE(LINE,FMB,AEVENT[P[I,J]],BEVENT[P[I,J]])   ;
                  J ← J + 1                                       ;
                GO TO L5
        END                                                       ;
    END                                                           ;
                                END.
```

APPENDIX B

REST PROGRAM

```
BEGIN  COMMENT
              REST TECHNIQUE OF RESOURCE ALLOCATION
           AN ALGOL PROGRAM FOR THE BURROUGHS B5500 COMPUTER
    C. A. MCNEILL - GEORGIA TECH - ATLANTA, GEORGIA - JUNE, 1966
                                                                    ;
      FILE IN    CARD   (2,10)                                      ;
      FILE OUT   LINE  6(2,15)                                      ;
      INTEGER
                 A,B,C,D,E,F,G,H,I,J,K,L,M,N,Q,R,S,T,U,V,W,X,Y,Z,
                 AB,BC,EE,FF,GG,KM,KT,PP,XX,ZZ,RESOURCE,SOURCE,RIN
                                                                    ;
SAVE
      INTEGER ARRAY
                 AEVENT,BEVENT,RESTA,RESTB,RESTC,RESTD,MIN,MAX,
                 ATOTAL,BTOTAL,TOTAL,TOTALX,TOTALY,RESTE,RESTF
                      [0:100] ,
                 P,MAN
                      [0:100,0:100]
                                                                    ;
      ALPHA ARRAY
                 PIC,COM
                      [0:12] ,
                 TITLE
                      [0:100,0:7]
                                                                    ;
      FORMAT IN
                 DATA1(I4,X1,I4,X1,I4) ,
                 DATA2(I3,X1,I3,X1,7A6,X3,5(I3,X1)) ,
                 DATA7(2A6,X1,I4) ,
                 DATA8(12A6)
                                                                    ;
      FORMAT OUT
                 RESULT1(/X20,"NUMBER OF ACTIVITIES = ",I4//X20,
                      "HIGHEST EVENT NUMBER = ",I4//X20,
                      "RESOURCE AVAILABILITY = ",I4////
                      X20,"EVENT NO.",X5,"ACTIVITY DESCRIPTION"//
                      (X20,I3," - ",I3,X5,7A6)) ,
```

```
          RESULT2(X10,"ACTIVITY",X40,"TIME"/) ,
          RESULT3(X20,25(X1,I3)) ,
          RESULT4(X9,I3," - ",I3) ,
          RESULT5(///X40," ADD ",I3," HUNDRED UNITS TO TIME ",
               "GIVEN BELOW "//) ,
          RESULT10(X20,"PROJECT IDENTIFICATION CODE = ",
               2A6/X20,"RUN NUMBER = ",I4) ,
          RESULT11(X20,12A6)
                                                              ;
SWITCH FORMAT
          SWFORM ← (X20,I4),(X24,I4),(X28,I4),(X32,I4),
          (X36,I4),(X40,I4),(X44,I4),(X48,I4),(X52,I4),
          (X56,I4),(X60,I4),(X64,I4),(X68,I4),(X72,I4),
          (X76,I4),(X80,I4),(X84,I4),(X88,I4),(X92,I4),
          (X96,I4),(X100,I4),(X104,I4),(X108,I4),(X112,I4),
          (X116,I4),
          (X20,A4),(X24,A4),(X28,A4),(X32,A4),(X36,A4),
          (X40,A4),(X44,A4),(X48,A4),(X52,A4),(X56,A4),
          (X60,A4),(X64,A4),(X68,A4),(X72,A4),(X76,A4),
          (X80,A4),(X84,A4),(X88,A4),(X92,A4),(X96,A4),
          (X100,A4),(X104,A4),(X108,A4),(X112,A4),(X116,A4)
                                                              ;
LIST
          DATA3(N,X,RESOURCE) ,
          DATA4(FOR I ← 1 STEP 1 UNTIL N DO [ AEVENT[I] ,
               BEVENT[I] , FOR J ← 1 STEP 1 UNTIL 7 DO
               [ TITLE[I,J]] , RESTA[I] , RESTB[I] ,
               RESTC[I] , MIN[I] , MAX[I] ]) ,
          DATA9(FOR I ← 1 STEP 1 UNTIL 2 DO [PIC[I]],RIN) ,
          DATA10(FOR I ← 1 STEP 1 UNTIL 12 DO [COM[I]]) ,
          RESULT6(N,X,RESOURCE, FOR I ← 1 STEP 1 UNTIL N DO
               [ AEVENT[I],BEVENT[I], FOR J ← 1 STEP 1 UNTIL
               7 DO [ TITLE[I,J]] ])     ,
          RESULT7(FOR H ← 25 × W + 1 STEP 1 UNTIL 25 × W +
               25 DO H) ,
          RESULT8(MAN[100×E + 25 × W + H,I]) ,
          RESULT9(" *S*"),
```

```
                         RESULT12("     ")
                                                                    ;
          SWITCH LIST
                   SWLIST ← RESULT8,RESULT9,RESULT12
                                                                    ;
          LABEL
                   L1,L2,L3,L4,L5,L6,BRP1,BRP2,BRP3,BRP4,BRP5,BRP6,SORT,
                   SORT1,RA1,RA2,RA3,RA4,RA5,RA6,RA7,RA8,RB1,RB2,RC1,RD,
                   ROUT1,ROUT2,NHP1,NHP2,NHP3,NHP4,NHP5,NHP6,NHP7,PRINT
                                                                    ;
          WRITE (LINE[NO])                                          ;
          READ  (CARD,DATA7,DATA9)                                  ;
          READ  (CARD,DATA8,DATA10)                                 ;
          READ  (CARD,DATA1,DATA3)                                  ;
          READ  (CARD,DATA2,DATA4)                                  ;
          CLOSE (CARD,RELEASE)                                      ;
ROUT1:
     BEGIN
               A ← 0                                                ;
               B ← U ← Y ← 1                                        ;
L1:            FOR I ← 1 STEP 1 UNTIL N DO
     BEGIN
               IF A = AEVENT[I] AND B = BEVENT[I] THEN
        BEGIN
               P[Y,U] ← I                                           ;
               IF BEVENT[I] = X THEN GO TO L2 ELSE
           BEGIN
               U ← U + 1                                            ;
               A ← B                                                ;
               B ← B +1                                             ;
               GO TO L1
           END                                                      ;
        END                                                         ;
     END                                                            ;
               IF B = X THEN GO TO L2 ELSE
     BEGIN
               B ← B + 1                                            ;
```

```
               GO TO L1
      END                                                        ;
L2:            C ← AEVENT[P[Y,U]]                                ;
               D ← BEVENT[P[Y,U]] + 1                            ;
L6:            FOR I ← 1 STEP 1 UNTIL N DO
      BEGIN
               IF C = AEVENT[I] AND D = BEVENT[I] THEN
          BEGIN
               G ← 1                                             ;
               GO TO L3
          END                                                    ;
      END                                                        ;
               IF D < X THEN
      BEGIN
               D ← D + 1                                         ;
               GO TO L6
      END                                                        ;
               IF C ≠ 0 THEN
      BEGIN
               U ← U − 1                                         ;
               GO TO L2
      END                                                        ;
               A ← 0                                             ;
               B ← B + 1                                         ;
               IF B = X + 1 THEN GO TO L4                        ;
               FOR I ← 1 STEP 1 UNTIL N DO
      BEGIN
               IF A = AEVENT[I] AND B = BEVENT[I] THEN
          BEGIN
               U ← U + 1                                         ;
               GO TO L1
          END                                                    ;
      END                                                        ;
               GO TO L1                                          ;
L3:            IF C = AEVENT[P[Y,G]] THEN
      BEGIN
               A ← C                                             ;
```

```
                B ← D                                                     ;
                Y ← Y +1                                                  ;
                U ← G                                                     ;
                GO TO L1
    END                                                                   ;
                P[Y+1,G] ← P[Y,G]                                         ;
                G ← G + 1                                                 ;
                GO TO L3                                                  ;
L4: END                                                                   ;
                FOR K ← 1 STEP 1 UNTIL Y DO
    BEGIN
                ATOTAL[K] ← 0                                             ;
                BTOTAL[K] ← 0                                             ;
                FOR L ← 1 STEP 1 WHILE P[K,L] ≠ 0 DO
      BEGIN
                ATOTAL[K] ← ATOTAL[K] + RESTA[P[K,L]]                     ;
                BTOTAL[K] ← BTOTAL[K] + RESTB[P[K,L]]
      END                                                                 ;
    END                                                                   ;
                FOR M ← 1 STEP 1 UNTIL Y DO
                TOTAL[M] ← ATOTAL[M] + BTOTAL[M] × RESOURCE               ;
                T ← 0                                                     ;
BRP1:  BEGIN
                T ← T + 1                                                 ;
                F ← 1                                                     ;
                G ← 2                                                     ;
                FOR I ← 1 STEP 1 UNTIL N DO RESTE[I] ← RESTD[I]           ;
                FOR PP ← 1 STEP 1 UNTIL Y DO
        BEGIN
                TOTALX[PP] ← -1                                           ;
                TOTALY[PP] ← -1                                           ;
        END                                                               ;
                SOURCE ← RESOURCE
      END                                                                 ;
BRP2:           IF TOTAL[F] < TOTAL[G] THEN GO TO BRP4 ELSE
BRP6:           G ← G + 1                                                 ;
                IF G ≤ Y THEN GO TO BRP2 ELSE
```

```
                Q ← F                                                  ;
BRP5:           IF TOTAL[Q] = 0 THEN GO TO PRINT ELSE
BRP3:           R ← 1                                                  ;
SORT1:          IF RESTA[P[Q,R]] > 0 THEN GO TO RA1                    ;
                IF RESTB[P[Q,R]] > 0 THEN GO TO RB1                    ;
                IF RESTC[P[Q,R]] > 0 THEN GO TO RC1                    ;
                R ← R + 1                                              ;
                GO TO SORT1                                            ;
BRP4:           F ← G                                                  ;
                GO TO BRP6                                             ;
RA1:
    BEGIN
                FOR EE ← 1 STEP 1 UNTIL N DO
                IF BEVENT[EE] = AEVENT[P[Q,R]] AND RESTE[EE] = 0 THEN
    BEGIN
RA2:            TOTALX[Q] ← TOTAL[Q]                                   ;
                GO TO ROUT2
    END                                                                ;
                IF SOURCE < MIN[P[Q,R]] THEN
                GO TO RA2                                              ;
                IF RESTA[P[Q,R]] ≤ SOURCE AND
                SOURCE ≤ MAX[P[Q,R]] THEN
    BEGIN
RA6:            MAN[T,P[Q,R]] ← RESTA[P[Q,R]]                          ;
                XX ← RESTA[P[Q,R]]                                     ;
                SOURCE ← SOURCE - RESTA[P[Q,R]]                        ;
                RESTA[P[Q,R]] ← 0                                      ;
                RESTD[P[Q,R]] ← -1                                     ;
                GO TO RA3
    END                                                                ;
                IF RESTA[P[Q,R]] ≤ MAX[P[Q,R]] AND
                MAX[P[Q,R]]  ≤ SOURCE THEN
                GO TO RA6                                              ;
                IF SOURCE < RESTA[P[Q,R]] AND
                RESTA[P[Q,R]] ≤ MAX[P[Q,R]] THEN
    BEGIN
                S ← SOURCE                                             ;
```

```
RA4:            IF RESTA[P[Q,R]] - S ≥ MIN[P[Q,R]] THEN
     BEGIN
                MAN[T,P[Q,R]] ← S                                    ;
                XX ← S                                               ;
                SOURCE ← SOURCE - S                                  ;
                RESTA[P[Q,R]] ← RESTA[P[Q,R]] - S                    ;
                GO TO RA3
     END                                                             ;
                S ← S - 1                                            ;
                IF S < MIN[P[Q,R]] THEN GO TO RA2 ELSE
                GO TO RA4                                            ;
   END                                                               ;
                IF SOURCE ≤ MAX[P[Q,R]] AND
                MAX[P[Q,R]] < RESTA[P[Q,R]] THEN
                GO TO RA5                                            ;
                IF MAX[P[Q,R]] < RESTA[P[Q,R]] AND
                RESTA[P[Q,R]] ≤ SOURCE THEN
   BEGIN
RA8:            M ← MAX[P[Q,R]]                                      ;
RA7:            IF RESTA[P[Q,R]] - M ≥ MIN[P[Q,R]] THEN
     BEGIN
                Z ← RESTA[P[Q,R]] - M                                ;
                ZZ ← ENTIER(Z / MIN[P[Q,R]] )                        ;
                IF (Z / MAX[P[Q,R]]) ≤ ZZ THEN
       BEGIN
                MAN[T,P[Q,R]] ← M                                    ;
                XX ← M                                               ;
                SOURCE ← SOURCE - M                                  ;
                RESTA[P[Q,R]] ← RESTA[P[Q,R]] - M                    ;
                GO TO RA3
       END                                                           ;
     END                                                             ;
                M ← M - 1                                            ;
                IF M < MIN[P[Q,R]] THEN GO TO RA2 ELSE
                GO TO RA7                                            ;
   END                                                               ;
                IF MAX[P[Q,R]] < SOURCE AND
```

```
                    SOURCE < RESTA[P[Q,R]] THEN
                    GO TO RA8                                              ;
                    GO TO RA2                                              ;
RA3:                FOR FF ← 1 STEP 1 UNTIL Y DO
                    FOR GG ← 1 STEP 1 WHILE P[FF,GG] ≠ 0 DO
    BEGIN
                    IF P[FF,GG] = P[Q,R] THEN
       BEGIN
                    TOTALX[FF] ← TOTAL[FF]                                 ;
                    TOTAL[FF] ← TOTAL[FF] - XX                             ;
       END                                                                 ;
    END                                                                    ;
                    GO TO ROUT2                                            ;
RA5:                M ← SOURCE                                             ;
                    GO TO RA7                                              ;
    END                                                                    ;
RB1:
    BEGIN
                    FOR EE ← 1 STEP 1 UNTIL N DO
                    IF BEVENT[EE] = AEVENT[P[Q,R]] AND RESTD[EE] = 0 THEN
                       GO TO RA2                                           ;
RB2:                MAN[T,P[Q,R]] ← -1                                     ;
                    RESTB[P[Q,R]] ← RESTB[P[Q,R]] - 1                      ;
                    IF RESTB[P[Q,R]] = 0 THEN
                       RESTD[P[Q,R]] ← 1                                   ;
                    XX ← RESOURCE                                          ;
                    GO TO RA3
    END                                                                    ;
RC1:
    BEGIN
                    FOR EE ← 1 STEP 1 UNTIL N DO
                    IF BEVENT[EE] = AEVENT[P[Q,R]] AND RESTD[EE] = 0 THEN
                       GO TO RA2                                           ;
                    MAN[T,P[Q,R]] ← -1                                     ;
                    RESTC[P[Q,R]] ← 0                                      ;
                    RESTD[P[Q,R]] ← 1                                      ;
                    RESTF[P[Q,R]] ← 1                                      ;
```

```
                GO TO ROUT2                                              ;
    END                                                                  ;
ROUT2:
    BEGIN
                IF SOURCE = 0 THEN
       BEGIN
NHP7:           FOR AB ← 1 STEP 1 UNTIL Y DO
          BEGIN
                IF TOTALY[AB] < 0 THEN
          BEGIN
                FOR BC ← 1 STEP 1 WHILE P[AB,BC] ≠ 0 DO
               BEGIN
                IF RESTD[P[AB,BC]] ≠ 0 THEN GO TO NHP5                   ;
                IF RESTA[P[AB,BC]] ≠ 0 THEN
                  BEGIN
                TOTALY[AB] ← 1                                           ;
                GO TO NHP4
                  END                                                    ;
                IF RESTB[P[AB,BC]] ≠ 0 AND TOTALX[AB] > TOTAL[AB] THEN
                     BEGIN
                TOTALY[AB] ← 1                                           ;
                GO TO NHP4                                               ;
                     END                                                 ;
                IF RESTB[P[AB,BC]] ≠ 0 AND TOTALX[AB] ≤ TOTAL[AB] THEN
                     BEGIN
                Q ← AB                                                   ;
                R ← BC                                                   ;
                FOR I ← 1 STEP 1 UNTIL N DO
                IF BEVENT[I] = AEVENT[P[Q,R]] THEN
                IF RESTE[I] = 0 AND RESTF[I] = 0 THEN GO TO NHP4         ;
                FOR I ← 1 STEP 1 UNTIL N DO
                IF BEVENT[I] = AEVENT[P[Q,R]] THEN
                IF RESTE[I] = 0 THEN
                        BEGIN
                FOR J ← 1 STEP 1 UNTIL N DO
                IF BEVENT[J] = AEVENT[I] AND RESTE[J] = 0 THEN GO TO NHP4;
                     END                                                 ;
```

```
              GO TO RB2                                       ;
                         END                                  ;
              IF RESTC[P[AB,BC]] ≠ 0 THEN
                   BEGIN
              Q ← AB                                          ;
              R ← BC                                          ;
              FOR I ← 1 STEP 1 UNTIL N DO
              IF BEVENT[I] = AEVENT[P[Q,R]] AND RESTD[I] = 0 THEN
              GO TO NHP4                                      ;
              GO TO RC1                                       ;
                   END                                        ;
NHP5:        END                                              ;
NHP4:     END                                                 ;
          END                                                 ;
              GO TO BRP1
       END                                                    ;
              F ← 1                                           ;
              G ← 2                                           ;
NHP1:         IF TOTALX[F] ≥ 0 THEN
       BEGIN
              F ← G                                           ;
              G ← G + 1                                       ;
              IF F > Y THEN GO TO NHP7 ELSE
              GO TO NHP1
       END                                                    ;
              IF F = Y THEN
       BEGIN
NHP6:         Q ← F                                           ;
              IF TOTAL[Q] = 0 THEN GO TO NHP7                 ;
              GO TO BRP5
       END                                                    ;
NHP2:         IF TOTALX[G] ≥ 0 THEN
       BEGIN
              G ← G + 1                                       ;
              IF G > Y THEN GO TO NHP6 ELSE GO TO NHP2
       END                                                    ;
              IF TOTAL[F] < TOTAL[G] THEN GO TO NHP3 ELSE
```

```
            G ← G + 1                                                   ;
            IF G > Y THEN
      BEGIN
            Q ← F                                                       ;
            IF TOTAL[Q] = 0 THEN GO TO NHP7                             ;
            GO TO BRP5
      END                                                               ;
            GO TO NHP1                                                  ;
NHP3:       F ← G                                                       ;
            G ← G + 1                                                   ;
            IF G > Y THEN
      BEGIN
            Q ← F                                                       ;
            IF TOTAL[Q] = 0 THEN GO TO NHP7                             ;
            GO TO BRP5
      END                                                               ;
            GO TO NHP1
    END                                                                 ;
PRINT:      WRITE(LINE,RESULT10,DATA9)                                  ;
            WRITE(LINE,RESULT11,DATA10)                                 ;
            WRITE(LINE,RESULT1,RESULT6 )                                ;
            E ← 0                                                       ;
L5:         FOR W ← 0 STEP 1 UNTIL 3 DO
    BEGIN
            WRITE(LINE[PAGE])                                           ;
            WRITE(LINE,RESULT5,        E)                               ;
            WRITE(LINE,RESULT2)                                         ;
            WRITE(LINE,RESULT3,RESULT7)                                 ;
            FOR I ← 1 STEP 1 UNTIL N DO
      BEGIN
            WRITE(LINE[NO],RESULT4,AEVENT[I],BEVENT[I])                 ;
            FOR H ← 1 STEP 1 WHILE 100 × E + 25 × W + H < T AND
                 H ≤ 25 DO
        BEGIN
            IF MAN[100 × E + 25 × W + H,I] = -1 THEN BEGIN KT ← H
                + 24                                                    ;
            KM ← 1 END ELSE BEGIN
```

```
                IF MAN[100 × E + 25 × W + H,I] = 0 THEN BEGIN KT ← H
                       + 24                                                   ;
                KM ← 2 END ELSE BEGIN KT ← H − 1                              ;
                KM ← 0 END                                                    ;
                       END                                                    ;
                WRITE(LINE[NO],SWFORM[KT],SWLIST[KM])
           END                                                                ;
                WRITE(LINE)
       END                                                                    ;
                IF 100 × E + 25 × W + 26 ≥ T THEN GO TO RD                    ;
   END                                                                        ;
                E ← E + 1                                                     ;
                GO TO L5                                                      ;
RD:             END.
```

BIBLIOGRAPHY

Literature Cited

1. Martino, R. L., Project Management and Control, Vol. 3, American Management Association, New York, New York, 1965.

2. Roper, D. E., "Critical Path Scheduling", Journal of Industrial Engineering, 15, 51-59, (March/April 1964).

3. Szuprowicz, B. O., "Choosing A Critical-Path Scheduling Program", Engineering News-Record, Vol. 172, No. 25, 78-80, June 18, 1964.

4. Burgess, A. R., and J. B. Killebrew, "Variation in Activity Level on a Cyclical Arrow Diagram," Journal of Industrial Engineering, 13, (March/April 1962).

5. Moder, J. J., and C. R. Phillips, Project Management with CPM and PERT, Reinhold Publishing Corporation, New York, New York, 1964.

6. Collins, F., Network Planning and Critical Path Scheduling, Know How Publications, Berkeley, California, 1964.

7. NASA-PERT "C" Computer Manual, National Aeronautics and Space Administration, GPO Catalog No. NPC 101-2A, Washington: Government Printing Office, 1964.

8. Macek, Anne, TIME-PERT-B 5000: An Algol 60 Program for the Analysis and Planning of Network Flow Problems, Technical Bulletin ORS-033, Burroughs Corporation, Detroit, Michigan, 1964.

9. Phillips, C., "Fifteen Key Features of Computer Programs for CPM and PERT", Journal of Industrial Engineering, 15, 14-20, (January/February 1964).

OTHER REFERENCES

10. Baumgartner, J., Project Management, Richard D. Irwin, Inc., Homewood, Illinois, 1963.

11. Bigelow, C., "Bibliography on Project Planning and Control by Network Analysis: 1959-1961", Operations Research Journal, 10, 728-731, (September/October 1962).

12. Clarke, R., An Introduction to Critical Path Analysis, Graduate School of Business, Stanford University, Stanford, California, 1961.

13. Dooley, A., "Interpretations of PERT", Harvard Business Review, 42, 160-171, (March/April 1964).

14. Lockyer, K., An Introduction to Critical Path Analysis, Pitman Publishing Corporation, New York, New York, 1964.

15. Martino, R. L., Project Management and Control, Vol. 1 and 2, American Management Association, New York, New York, 1964.

16. Miller, R., Schedule, Cost, and Profit Control with PERT, McGraw-Hill Book Company, New York, New York, 1963.

17. Miller, R., "Control with PERT", Bursk, E. and J. Chapman, eds., New Decision-Making Tools for Managers, Harvard University Press, Cambridge, Massachusetts, 1963.

18. Odom, R., and E. Blystone, "A Case Study of CPM in a Manufacturing Situation", Journal of Industrial Engineering, 15, (November/December 1964).

19. Shaffer, L., J. Ritter, and W. Meyer, The Critical-Path Method, McGraw-Hill Book Company, New York, New York, 1965.

20. Stilian, G. and Others, "PERT: A New Management Planning and Control Technique", Paper number 74, American Management Association, New York, New York, 1962.

21. White, G., CPM in Construction, The Associated General Contractors of America, Washington, D.C., 1965.

22. GE-200 Series Critical Path Method Program, Publication No. CPB-198C, Computer Department, General Electric Company, Phoenix, Arizona, 1964.