2

RDTR No. 174

Alternate Methods
of
Project Scheduling
with
Limited Resources

DEPARTMENT OF THE NAVY ORDNANCE SYSTEMS COMMAND

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED

PREPARED BY

RESEARCH & DEVELOPMENT DEPARTMENT

Naval Ammunition Depot, Crane, Indiana

NAVAL AMMUNITION DEPOT
CRANE, INDIANA

(11) 7 Aug 70

(12) 150p.

(6)

ALTERNATE METHODS OF PROJECT SCHEDULING WITH LIMITED RESOURCES.

(14) NAD-CR- RDTR ~~No.~~-174

(9) Doctoral thesis

BY

(10) JAMES [illegible] PATTERSON

Herbert

247 250

ALTERNATE METHODS OF PROJECT SCHEDULING
WITH LIMITED RESOURCES

BY

JAMES H. PATTERSON

176 750

A Dissertation Submitted in Partial Fulfillment of
the Requirements for the Degree of Doctor of
Business Administration in the Graduate
School of Business of Indiana University

INDIANA UNIVERSITY
GRADUATE SCHOOL OF BUSINESS

1970

ACCEPTANCE

This dissertation has been accepted in partial fulfillment of the requirements for the Degree Doctor of Business Administration in the Graduate School of Business of Indiana University.

Date August 7, 1970

Dean, School of Business

Chairman

Member

Member

ACKNOWLEDGEMENTS

This dissertation is the result of the efforts of many people. I am deeply indebted to the members of my dissertation committee: Professor John F. Muth (chairman), Professor William C. Perkins, and Professor James M. Utterback. To Professor Muth I am especially grateful. He not only narrowed down the topic for this research, but provided much of his own time to make constructive criticisms of this dissertation. Had I been able to get as quick a turn-around from the CDC 3600 as I did material submitted to Professor Muth, this study would have been completed much earlier!

Several employees of the U.S. Naval Ammunition Depot, Crane, Indiana, provided both help and moral support throughout the writing of this dissertation. Mr. Stan Fasig, director of the Research and Development Department, provided funding for the project and helped in other ways too numerous to mention. Mr. Jim Werne programmed MPSP, the computer program that implements the scheduling rules treated in this research. He gave freely of his own time, both on weekends and late at night when the author made a discovery he couldn't wait to discuss. Mr. Jerry Kemp and other members of the operations research group in the Crane Research and

Development Department provided much help in formulating a portion of the experimental design as well as interpreting the results. And to Miss Sherry Taylor, who typed all of the initial drafts as well as the final copy, I express amazement. She did a tremendous job under very short notice for most of the chapters.

The final acknowledgement is more personal. To my "dissertation widow" I will be forever grateful. She read a large portion of the final copy of this dissertation while she herself was a student at Indiana University.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The allocation of resources to contemporaneous projects is a problem characteristic of manufacturing firms, contracting firms, and research and development organizations. In allocating resources it is often found that activities demand more resources than are available. When this happens a question arises regarding which jobs to delay and which jobs to schedule.

Mathematical programming techniques and heuristic solution procedures have been employed to solve this resource allocation problem. The mathematical programming methods are generally quite slow and often require more computer memory than is available. These methods do, however, give optimum solutions to resource allocation problems when the problem is small enough to be solved with the methods. Heuristic solution procedures require relatively small amounts of computer memory and are generally quite fast at generating solutions to a problem. The solutions derived when employing these procedures are, however, not necessarily optimum. The object of this study is to statistically assess many of the heuristic solution procedures available for scheduling multiple projects. Mathematical programming methods are

considered only to the extent that they can be employed in implementing the logic of a heuristic solution procedure.

The approach taken in this assessment is to obtain data representing an actual multi-project scheduling problem. The data gathered consists of thirty-four projects each with as many as thirty activities to be scheduled over an eight month interval.

The thirty-four projects are scheduled simultaneously using four variations of five basic scheduling heuristics. The scheduling heuristics examined are as follows: (1) the shortest imminent operation heuristic of the job shop problem, (2) a rule which considers the slack time of an activity, (3) a rule which considers the total resources demanded by an activity, (4) a rule which attempts to schedule around potential bottleneck activities, and (5) a rule which employs a zero-one integer programming algorithm in order to schedule man power efficiently over a small time interval. The variations of these rules are a provision for scheduling activities at random in case of ties on another rule and a reschedule feature which reschedules activities based on resource conflicts exhibited in the first feasible schedule.

Each of the scheduling rules is evaluated on its ability to generate schedules with low total project delays and low

weighted project delays where the weights are determined by the size of the project. In addition, the rules are evaluated on their ability to generate schedules which utilize man power efficiently. Employing analysis of variance tests, multiple ranking procedures, and multiple comparison methods, an attempt is made to distinguish which of the scheduling rules are superior to others in satisfying these measures of organizational performance.

The results of the study demonstrate that scheduling rules which are based on the logic of processing jobs through the system as rapidly as possible produce schedules with low project delays at the expense of a rather inefficient utilization of resources. Those scheduling rules which are based on using resources efficiently do so at the expense of large delays in the completion of projects. The shortest imminent operation heuristic schedules activities with the least total project delays. A scheduling rule based on the total float present in an activity, however, generally schedules projects with less weighted delay.

Attempts to improve upon scheduling rules by adding an element of chance in the scheduling of activities or by rescheduling activities based on resource conflicts exhibited in the first feasible schedule are generally unsuccessful

when a large class of problems is considered. In isolated cases, though, the presence of randomness and rescheduling makes substantial improvements in a schedule.

The difference in computer processing times required for implementing the various scheduling rules is small for the problems considered. The scheduling rule employing the zero-one integer programming algorithm is the only one which requires a significantly larger amount of computer time for developing a schedule.

CHAPTER 1

RESOURCE ALLOCATION IN PROJECT NETWORKS

1.1 Introduction

The scheduling of multiple projects involves organizing vast quantities of diversified resources. The post-World War II era has witnessed the organization of massive projects in building numerous dams, bridges, and interstate highway systems, as well as in developing huge missile complexes and intricate earth satellites. Thus, it is somewhat surprising that until recently little research effort has been expended in developing scheduling systems for multiple projects.

The growing size and complexity of these projects has intensified the need for better methods of planning and scheduling work activity. A great advance was made in 1959 with the development of critical path scheduling,[1] a technique which considers the temporal dimensions of scheduling. The importance of this technique is reflected

---

[1] For the original descriptions of this technique, see: J. E. Kelley and M. R. Walker, "Critical-path Planning and Scheduling," *Proceedings of Eastern Joint Computer Conference*, (December 1959), pp. 160-173 and D. Malcolm, et al., "Application of a Technique for Research and Development Program Evaluation," *Operations Research*, Vol. 7, No. 5 (September-October 1959), pp. 646-669.

in the number of publications on the subject. Over nine texts and 300 articles have been published in the ten-year period since the introduction of the method.[2] The publications have both challenged the assumptions of the technique and increased the number of situations where the method can be applied.

The basis of critical path scheduling in both the single and the multiple project case is the representation of a work program or project as a network of nodes and arcs giving an explicit representation of the interrelationships of the jobs necessary to complete the project. An example network diagram is shown in Figure 1. It is referred to in the literature as a precedence diagram or an activity-oriented project graph, in which the nodes represent jobs to be performed and the arcs denote precedence of jobs.[3] Thus the job or activity[4] assigned the number 3 cannot begin until the first job has been completed. A single project, then, is defined as a partially

---

[2] E. W. Davis, "Resource Allocation in Project Network Models--A Survey," *Journal of Industrial Engineering*, Vol. 17, No. 4, (April 1966), pp. 177-188.

[3] An alternate network representation of a project is the event-oriented diagram where the roles of arcs and nodes is just reversed. Nodes represent events or discrete points in time and arcs represent activities or jobs originating and terminating in a unique pair of nodes. The analysis is similar using either representation.

[4] Job, activity, and task are used synonymously in this study.

ordered set of jobs requiring time and resources, executed according to a predetermined logic. When the terminus or the final job is completed, that project is said to be completed.

The outstanding feature of project activity, as distinguished from the job shop and the assembly line balancing problem, is that each project has never been performed before and probably will never be performed again.[5]

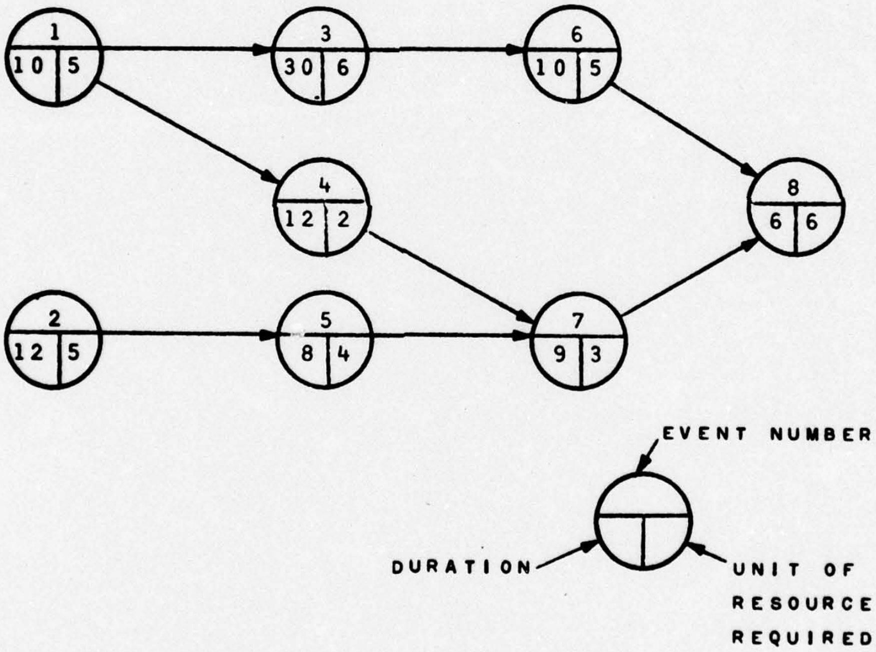


Figure 1: Project Network Model

---

[5] For an excellent discussion of the differences and similarities of the job shop, the assembly line balancing, and the project scheduling problems, see: E. W. Davis, "An Exact Algorithm for the Multiple Constrained-Resource Project Scheduling Problem," (Ph.D. Dissertation, Yale University, 1969), pp. 9-13 and J. D. Weist, "The Scheduling of Large Projects with Limited Resources," (Ph.D. Dissertation, Carnegie Institute of Technology, 1963), pp. 4-6

The representation of a project as a network diagram serves as the basis for the time calculations of the critical path method and determines the earliest (EST) and latest (LST) times that a job can start for the efficient execution of the project. A listing of the definitions and symbols relevant to the critical path method is included in Appendix B. These definitions and symbols are used throughout the study.

The applicability of critical path scheduling is limited by the algorithm's inability to cope with conflicting resource demands. These conflicts arise when two or more jobs simultaneously demand more resources than are available. Either additional resources will have to be obtained, or one job will have to be postponed until sufficient resources become available to schedule it. In the latter instance, the dominant questions become which jobs to schedule first, and which jobs to delay.

Several different algorithms have been developed for assigning scheduled starting dates to jobs within projects where conflicts in resource usage occur. These algorithms are generally composed of a mathematical programming routine or a simple heuristic scheduling rule which establishes priorities for jobs to be subsequently scheduled.

Although massive efforts and large expenditures have been made in developing scheduling algorithms, very little effort has been expended in measuring the relative effectiveness of the heuristics which these algorithms employ. An evaluation of the scheduling power of various heuristic rules for scheduling multiple projects is desirable in order to determine which ones are most likely to develop improved schedules. Such an evaluation is the purpose of the present investigation.

1.2 Problems in Resource Allocation

Four distinct problems related to allocating resources in project networks exist. They are called: Resource accumulation (Fig. 2A), Resource smoothing or leveling (Fig. 2B), Satisfying resource constraints (Fig. 2C), and Time/cost tradeoff procedures (Fig. 3). These problems are outlined below, and the problem with which this study is concerned is indicated. In all problems except the time/cost tradeoff procedures, it is assumed that the resources are used by an activity at a constant rate. Figure 2 gives the resource usage resulting from the application of the constant resource usage methods to the project in Figure 1.

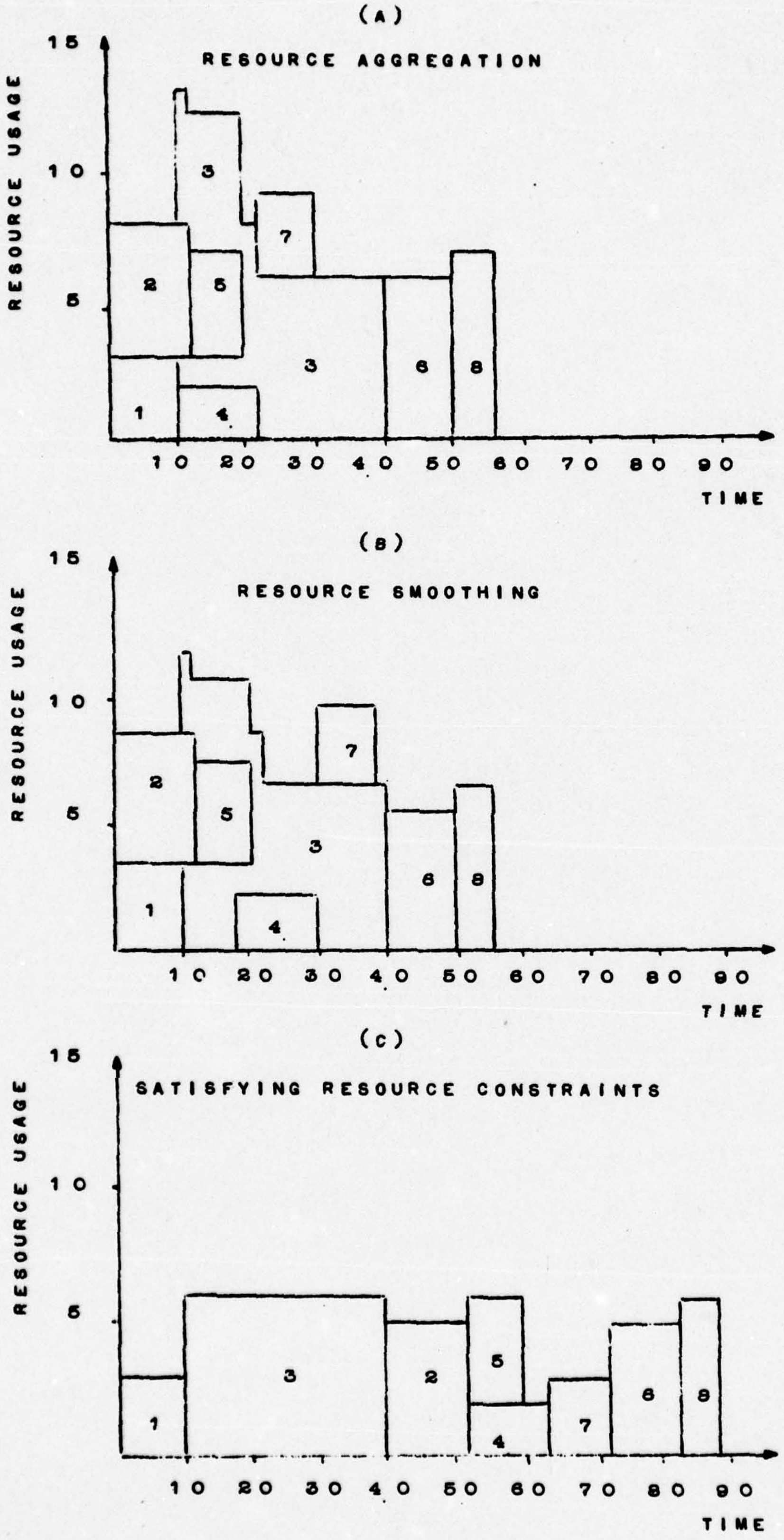

(A)
RESOURCE AGGREGATION
RESOURCE USAGE
15
10
5
3
7
2
5
3
6
8
1
4
10
20
30
40
50
60
70
80
90
TIME
(B)
RESOURCE SMOOTHING
RESOURCE USAGE
15
10
5
7
2
5
3
6
8
1
4
10
20
30
40
50
60
70
80
90
TIME
(C)
SATISFYING RESOURCE CONSTRAINTS
RESOURCE USAGE
15
10
5
3
2
5
1
4
7
6
8
10
20
30
40
50
60
70
80
90
TIME

Figure 2

1. Resource Accumulation[6] (Fig. 2A)

Resource accumulation is the simplest technique and consists of accumulating resources against a time scale determined by the critical path analysis. The essence of the method is that all jobs are started at the earliest possible times. The method is generally used when technical expertise is not available to cope with the widely varying resource usage rates that the method creates.

The completion of the entire project as soon as is technically possible is the over-riding scheduling criterion under resource accumulation. Because all jobs are started at the earliest possible time, the probability of the project's being delayed beyond the completion date indicated by the critical path analysis is less than under the other two scheduling problems which assume a fixed resource usage rate.

2. Resource Leveling (Fig. 2B)

The resource leveling problem assumes resources are used at a constant rate, as long as no delays are introduced in the

---

[6] The title of this method is based on the resource allocation section of the "Report of the Working Party on Large Networks" by D. J. McLeod and C. Staffurth. This reference can be found in: Gail Thornley, (ed.), *Critical Path Analyses in Practice*, London: Tavistock Publications, 1968, pp. 45-47.

completion of the project. This approach may be satisfactory when sufficient resources are available to schedule all jobs at the early start date. Burgess and Killebrew show that solving this problem is equivalent to minimizing the sum of squares of the resource requirements from time period to time period; they present a computer flow chart for doing exactly this.[7]

The importance of this problem is indicated by the number of situations in which it exists. Private contractors, for example, are typically less concerned with resource limitations than they are with meeting contract due-dates. They have flexible resource levels available through union hiring halls, yet they also have a desire to keep the usage of these resources as level or as smooth as possible.

This study does not concern itself with this problem directly, but many of the techniques employed by it are useful in the next problem considered, that of extending the project as little as possible given stated resource constraints. The review by Davis gives a more detailed

---

[7] A. R. Burgess and J. B. Killebrew, "Variation in Activity Level on a Cyclic Arrow Diagram," *Journal of Industrial Engineering*, Vol. 13, No. 2, (March-April 1962), pp. 76-83.

discussion of the resource smoothing problem.[8]

3. Satisfying Resource Constraints (Fig. 2C)

Satisfying resource constraints is concerned with minimizing a project's duration where a lack of resources forces the postponement of some jobs. Many employers, for example, find it both expensive and difficult to vary manpower and physical facilities within a scheduling period. Due to labor contract requirements and the disadvantages of releasing workers (ill-will, increased unemployment compensation contributions, etc.), these firms or individuals are reluctant to vary their usage of resources to any great extent. They are more interested in how the duration of a project can be minimized subject to constraints on resources.

The month-ending project of closing the accounting records in a large firm is an example of the necessity of a firm to extend the duration of a project because of constraints on resource availability.[9] The important point

---

[8] E. W. Davis, op. cit., (1966), pp. 183-185.

[9] J. D. Weist, op. cit., (1963), p. 9.

to note is that when the limits on resources are inflexible, there is a potential problem to a firm which otherwise has flexible resource levels available in other operating departments. Extending the duration of projects minimally under stated resource constraints is the problem of resource allocation in project networks with which this study is concerned.

4. Time/Cost Tradeoff Procedures (Fig. 3)

When projects can be accelerated by the allocation of more resources to an activity, time/cost tradeoff procedures are available for determining the least costly schedule for a given project duration. It is assumed under this method that resources can be used by an activity at a varying rate and that as the number of resources assigned to an activity increases, the direct cost of completing the activity increases also.

If the time-cost relationship of project activities is a continuous convex function which can be accurately approximated by a piecewise linear curve, the problem can be formulated and solved using the Ford-Fulkerson network flow algorithm.[10] Figure 3 illustrates the activity time-cost

---

[10] E. W. Davis, op. cit., (1966), p. 182.

relationships necessary for solving the problem by the Ford-Fulkerson method.

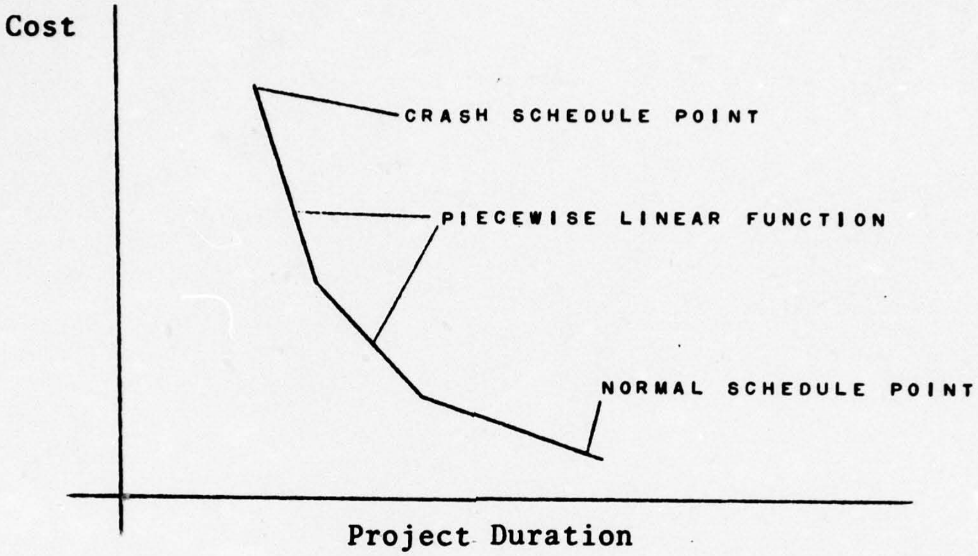


Figure 3: Piecewise Linear Project Cost Curve

Of the four problems of resource allocation discussed, minimizing project duration subject to resource constraints (satisfying resource constraints) has proved the most difficult to solve computationally, and so is the one examined in this research. The next section describes the difficulties that are encountered when one attempts to solve the resource allocation problem. These difficulties stem from an inability to construct and solve a well-defined mathematical system. Heuristic methods of solution are singled out as being the most computationally practicable means of solving the problem today.

1.3 Methods of Solving the Resource Constrained Scheduling Problem

The resource constrained scheduling problem is representative of a class of combinatorial problems; an extremely large number of ways exist in which a series of project schedules can be developed. Being a combinatorial problem, it has not yielded readily to the solution techniques of the usual mathematical programming routines. Attempts to solve the resource constrained scheduling problem by linear,[11] integer,[12] quadratic,[13] and zero-one programming,[14] as well as bounded enumeration[15] have been

---

[11] J. D. Brand, "The Resource Scheduling Problem in Construction," *Civil Engineering Studies, Report No. 5*, (Urbana, Illinois: Department of Civil Engineering, University of Illinois, 1964).

[12] J. D. Weist, op. cit., (1963)

[13] Fred L. Bennett, "Some Approaches to the Critical Path Scheduling Resource Allocation Problem, (Ph. D. Dissertation, Cornell University, 1966).

[14] A.A.B. Pritsker, et al., "Multi-Project Scheduling with Limited Resources: A Zero-One Programming Approach," *Management Science*, Vol. 16, No. 1, (September 1969), pp. 93-109.

[15] E. W. Davis, op. cit., (1969).

successful only on small sets of moderate size projects.[16] For most actual project scheduling problems, the methods are wholly inappropriate, the only exception being the solution of local or one-period scheduling problems. The following are some of the difficulties encountered in solving resource constrained problems with these methods:

1. The first source of difficulty stems from the inability to construct a clear statement of the problem. This is due in part to the fact that the problem is *ill-structured* and is difficult to express in precise mathematical terms.

Managers and project engineers quite often ask about the effects of doubling the number of engineers assigned to a project or of leasing additional equipment during the critical phases of a project. For the most part, they are interested in qualitative answers to their questions or else are interested in the results of differing assumptions on the input parameters. This need for examining alternate assumptions makes a well-defined objective nearly impossible in these instances. Because of the ability of heuristic

---

[16] Small sets of moderate size projects refer to problems containing no more than 4 projects of 30 or less activities each, with each activity demanding less than 5 different resource types.

methods to generate solutions rapidly, they are natural candidates for answering these questions.

The number of alternatives available in many scheduling problems also leads to difficulty in problem statement. Activities can generally be started and maintained for a long time with fewer resources than are ideal. This leads to difficulty in stating just what levels of a resource are needed to complete an activity. The fact that the activities interact in their sharing of these resources only serves to complicate the problem.

2. A second source of difficulty stems from the formulation and solution of the mathematical system, once the problem has been unambiguously stated. This is due in part to the sheer size[17] of the problem and also the number of variables and constraints necessary to correctly describe the problem. Weist cites the following example:

> ....given a project with 55 jobs in 4 shops with a time span of 30 days, and you have some 5275 equations and 1650 variables (not counting slack variables or the additional equations and variables necessary to assure an integer solution). If job splits are not allowed, the number of equations increases to about 6870.[18]

---

[17] For example, scheduling 5 jobs, each being processed on 5 different machines, leads to considering some 25 billion [$(5!)^5$] possible job sequences.

[18] J. D. Weist, op. cit., (1963), p. 24.

To be sure, the problem can be formulated with fewer variables and constraints than those outlined by Weist. The work of Balas,[19] Pritsker,[20] and others have led to more compact formulations of the problem. Even with these reduced formulations of the resource constrained scheduling problem, the number of constraints and variables necessary to describe and solve the problem exceeds the memory capacity of the largest computers available today.

When mathematical analysis of a problem breaks down due to difficulty in stating the problem, difficulty in formulating the problem, or difficulty in computationally solving the problem, one generally resorts to a heuristic procedure for examining a sub-space of his problem-solution domain rich in good, but possibly not optimum solutions. Such has been the case in solving the resource constrained project scheduling problem. To date, the more computationally practicable programs have been those employing heuristic methods of solution. In Weist's latest version of his SPAR-2 program, for example, he is able to accommodate a problem with 1500 jobs, 500 nodes, and

---

[19] E. Balas, "An Additive Algorithm for Solving Linear Programs with Zero-One Variables," *Operations Research*, Vol. 13, No. 3, (July-August 1965), pp. 517-546.

[20] A.A.B. Pritsker, et al., op. cit., pp. 93-109.

practically no limit on the number of resource categories or length of the project in days.[21] Other heuristic programs give optimum or near optimum solutions to certain project scheduling problems in one-fourth or less of the time that it takes their mathematical programming counterparts to arrive at the same solution.

By resorting to heuristic solutions one is, of course, sacrificing a guaranteed optimum solution to his problem. This is the price paid for the computational feasibility that heuristics provide. But the guarantee of an optimum solution is of little real value if it is impossible to find.

1.4 Objective and Scope of the Study

The primary objective of the study is to evaluate the ability of heuristic rules to schedule multiple projects which occur simultaneously within a firm. These rules are evaluated on their ability to generate schedules with low total project delays, low weighted project delays where the weights are determined by the size of the project, and on

---

[21] J. D. Weist, "A Heuristic Model for Scheduling Large Projects with Limited Resources," *Management Science*, Vol. 13, No. 6, (February 1967), pp. 359-399.

their ability to generate schedules which utilize man power efficiently. An assessment of the amount of computer time required for implementing each of the scheduling rules is also given.

Five basic scheduling rules each with four variations are examined in the study. Examples of these rules are the shortest imminent operation heuristic of the job shop problem, a rule which considers the slack present in an activity, and a rule which attempts to *look-ahead* and schedule potential bottleneck activities. The variations of these rules are a provision for rescheduling activities based on resource conflicts exhibited in the first feasible schedule and a provision for switching activities at random in the order in which they are considered for resource assignment when they tie on one of the five basic scheduling rules.

The scheduling rules examined are applied to an actual multi-project scheduling problem existing in a research and development organization which develops pyrotechnic items. Thirty-four projects occur simultaneously in this organization and are scheduled over an eight month interval. Each project contains at most thirty activities and each activity demands as many as thirteen different resource types.

Analysis of variance, multiple range tests, and multiple ranking procedures are employed to assess the abilities of the scheduling rules. In all cases, an attempt is made to distinguish which rule schedules activities with the lowest amount of delay in the completion of the projects and which rule schedules activities so that man power is utilized most efficiently.

The next chapter reviews previous research efforts expended in solving the resource constrained, multi-project scheduling problem by heuristic methods.

CHAPTER 2

REVIEW OF COMPARATIVE STUDIES

2.1 Introduction

In this chapter three previous research efforts which address the constrained resource project scheduling problem are reviewed. The first, by Pascoe[1] provides an example of an experimental design for assessing the efficacy of project scheduling rules. The remaining studies by Mize[2] and Knight[3] suggest some heuristic scheduling rules which appear valuable for scheduling several projects which simultaneously compete for the limited resources of the firm. These research efforts are by no means all of the literature on project scheduling available today; they are simply indicative of the existing literature and are directly relevant to this study.

---

[1] T. L. Pascoe, "An Experimental Comparison of Heuristic Methods for Allocating Resources," (Ph.D. Dissertation, Cambridge University, 1965).

[2] J. H. Mize, "A Heuristic Scheduling Model for Multi-Project Organizations," (Ph.D. Dissertation, Purdue University, 1964).

[3] R. M. Knight, "Resource Allocation and Multi-Project Scheduling in a Research and Development Environment," (M.S. Thesis, Massachusetts Institute of Technology, June, 1966).

Before the pertinent literature is reviewed, it will be useful to have a standard set of definitions and symbols available. Table 1 provides such a list. In addition, the following definitions will be used throughout the remainder of this study.

1. Activity a_j is said to have *priority* over activity a_j' at time t if activity a_j is considered for resource assignment before activity a_j' is considered. Priorities are often based on rules-of-thumb.

2. *Resources* are man power, funding, facilities, and equipment that separately or in combination are expended in accomplishing an activity. Man-hours and resources are used synonomously throughout this research. (See "Assumptions" in Chapter 3.) There are q different types of resources that may be required to complete any given project or activity within a project.

3. A *schedule* consists of an assignment of starting dates and resource usage for all activities comprising the projects of the firm. A schedule is said to be feasible if (1) all precedence relations have been satisfied and (2) sufficient resources are available for assignment to activities at their scheduled occurrence dates.

4. Θ is a vector of subsets of activities that have not been but could be scheduled during the current time interval. The subsets of activities comprising Θ changes throughout the development of a schedule. Θ is p dimensional; $\Theta = (\theta_1, \theta_2, \ldots, \theta_p)$. θ_m exists for all projects ($m = 1, 2, \ldots, p$).

2.2 T. L. Pascoe

Pascoe undertakes the problem of determining the amount of variation in selected project scheduling objectives that can be accounted for by the methods of determining schedules. His experimental medium is a set of single projects (consisting of twenty jobs and three resources) constructed randomly by sampling selected project characteristics. These characteristics are shown in Table 2.

Pascoe notes that various measures of scheduling priority can be implemented through sorting routines. For example, if a scheduling rule based upon the least total float of an activity is desired, all that is needed to obtain this priority is a sort routine which will sort activities on the basis of total float, those activities possessing the least total float being placed first on the sorted list. Scheduling activities on the basis of total float then simply consists of assigning resources to activities in their sorted order until resources are fully

TABLE 1

GLOSSARY OF SYMBOLS

A	The set of all activities to be scheduled
a_j	Activity j ($a_j \varepsilon A$)
b_i^t	Resource i available at time t; $\underline{b}^t$ dimension 1 x q
c_j	$\sum_{i=1}^{q} r_{ij}$
d_j	Duration of a_j
p	Number of projects to be scheduled
q	Number of resource categories available during a schedule
r_i^*	Resource i available during a schedule span; $\underline{r}^*$ dimension 1 x q
R^t	Matrix of resources required by all a_j which can be scheduled at time t
r_{ij}	Resource i required by a_j
Θ	(See definitions preceding)

TABLE 2

PARAMETERS USED BY PASCOE FOR GENERATING PROJECTS

a = aspect ratio = project rank/the maximum number of parallel activities in a project

where:

rank = the number of activities preceding the terminus event along the longest path from the origin of the project

c = complexity = m/n

where:

m = number of activities in a project

n = number of nodes or events in a project

$$d = \text{density} = \frac{\sum_{a_j \varepsilon A} d_j}{\sum_{a_j \varepsilon A} d_j + \sum_{a_j \varepsilon A} FFa_j}$$

$$o = \text{obstruction} = \frac{c \cdot R}{s_r}$$

where:

c = complexity

s_r = the average number of activities that can be accommodated within the availability of resource r.

$$R = \frac{r_{av}}{R_{tot}} = \frac{\text{the average number of resources used per activity}}{\text{the number of resources available to the project}}$$

Source: T. L. Pascoe, op.cit., pp. B-19 - B-21.

expended. Although not all priorities can be implemented in this manner, enough of them can be to make it worth repeating those which Pascoe considers. These are shown in Table 3.

The results of Pascoe's study in the form of an analysis of variance for the criteria Duration of the Project and Percent Utilization of Facilities appear in Tables 4 and 5 respectively. It is interesting to note that heuristics as a whole (f) account for a much smaller sum of squares in the analysis of variance than do various project characteristics (parameters) defined by Pascoe. The minor sort routine (g) used for breaking ties in scheduling rules also accounts for a very limited sum of squares in the analysis.

Relatively few conclusions can be made from Pascoe's study regarding the efficacy of scheduling rules in explaining variation in the duration of projects (Table 4). The effects of project parameters appear to have more influence in the determination of the duration of schedules than do the heuristic rules used to determine them.

A logical question might be raised as to whether the values of the project parameters outlined by Pascoe are indicative of the values found in practical situations. If they are (this would need to be determined through the

TABLE 3

SORTING CRITERIA USED BY PASCOE FOR RANKING ACTIVITIES

1. ESTART = Early Start Time
2. LSTART = Late Start Time
3. EFINISH = Early Finish Time
4. LFINISH = Late Finish Time
5. TF = Total Float
6. Duration = d_j
7. RED = A measure giving high priority to long activities which require a high percentage of the more important resources, the important resources being defined as those resources which constrain a project from being completed within its critical path completion time estimate.
8. REDNRED = A measure giving high priority to an activity if it immediately precedes a set of activities with high RED's.
9. CUMRED = A measure which makes REDNRED long term and includes all logically subsequent activities.
10. RANDOM = A measure which assigns priorities to activities in a project at random.

Source: T. L. Pascoe, *op. cit.*, pp. B-7-B-8.

TABLE 4

PASCOE'S ANALYSIS OF VARIANCE FOR THE DURATION OF PROJECTS

Source*	d.f.	S.S.	M.S.	F	$F_{\alpha=0.05}$
b=[a,c,d,o] bb	10	198,015	19,802	253.87	1.85
f	10	1,232	123	1.58	1.85
g,bg,fg	15	32	2	0.03[a]	1.70
bf	40	1,888	47	0.06[a]	1.42
Residual	628	7,138	11	0.14[a]	1.15
Within cells	704	54,871	78		
Total	1,407	263,176	187		

Source: T. L. Pascoe, *op. cit.*, p. C-14

* Definitions for each of these sources are found in Table 2 and on page 24 of this study. The block sources b = [a,c,d,o] are the parameters used by Pascoe for generating projects, and the block-by-block interactions (bb) are higher order interactions of these parameters with one another. These are highly significant in the analysis of variance (253.87 > 1.85). The sorting criteria (heuristics) used by Pascoe in ranking activities for scheduling (f) as well as the minor sort routines used in breaking ties (g) are insignificant. The higher order interactions of the ranking criteria with blocks and with the minor sort routines (bf and bg, respectively) are also insignificant in the analysis of variance.

[a] These low variance ratios are highly unlikely, and may be the result of rounding or computing error. The analysis of variance performed in this study also gave low variance ratios for several of the insignificant sources of variation. These low variance ratios are felt to be the result of rounding error in the computer.

TABLE 5

PASCOE'S ANALYSIS OF VARIANCE FOR THE PERCENT UTILIZATION OF FACILITIES BY PROJECTS

Source*	d.f.	S.S.	M.S.	F	$F_{\alpha=0.05}$
b=[a,c,d,o] bb	10	486,605	48,661	108.37	1.85
f	10	12,373	1,237	2.75	1.85
g,bg,fg	15	272	18	0.04	1.70
bf	40	34,050	851	1.90	1.42
Residual	628	123,476	197	0.44	1.15
Within cells	704	316,051	449		
Total	1,407	972,827	691		

Source: T. L. Pascoe, op. cit., pp. C-16.

* Definitions for each of these sources are found in Table 2 and on page 24 of this study. The block sources b = [a,c,d,o] are the parameters used by Pascoe for generating projects, and the block-by-block interactions (bb) are higher order interactions of the parameters with one another. These are highly significant in the analysis of variance (108.37 > 1.85). The sorting criteria used by Pascoe in ranking activities for scheduling are also significant (2.75 > 1.85). The minor sort routines used in breaking ties (g) as well as all of the remaining higher order interactions (bg, fg, and bf) are insignificant in the analysis of variance.

statistical sampling of various projects), then means other than the heuristic sorting routines used by Pascoe would have to be devised in order to lend statistical proof to the superiority or inferiority of heuristic rules.

The significance of the scheduling rules Pascoe considers becomes much more apparent when evaluating the percent utilization of facilities by projects (Table 5). This is an indication that statistical ranking procedures may be able to determine which scheduling rules give the most efficient utilization of resources. These ranking procedures would be a useful addition to Pascoe's study. The analysis only indicates that scheduling rules are significant. It does not indicate which rule or group of rules produces schedules which result in low facility idle time.

To further validate Pascoe's work, one needs to determine the number of instances in which a firm devotes all of its resources to a single project. Of more general interest is the case of multiple projects which compete for the available resources in the firm. Devising a parameterization scheme for project characteristics is significantly more difficult in the multi-project scheduling case, however.

2.3 J. H. Mize

Mize examines the multi-project scheduling problem and, like Pascoe, uses simulated projects. His general conclusions, although not statistically assessed, are that:

> The best of the nine scheduling rules tested were those which resolved local schedule conflicts in such a way as to result in minimum critical path violation. It was also found that considerable improvement could be realized in the schedules by employing two 'gap-closing' procedures, after the first feasible schedule was generated.[4]

Results of Mize's study are shown in Table 6. The scheduling rules he considers, because of their length, are included in Appendix A at the end of this chapter.

Although Mize's study considers mainly problems consisting of six projects each of thirty or less activities, much can be learned and applied to the present research from the scheduling rules he examines. In the majority of cases a rule which produces a schedule with low project slippage simultaneously schedules activities so that facility idle time is low. This tends to demonstrate that a different scheduling rule is not needed for every objective; a small set of rules may satisfy the majority of project scheduling objectives of the firm.

Statistical analyses of Mize's results are shown in

---

[4] J. H. Mize

TABLE 6

SUMMARY STATISTICS FOR PROJECT SLIPPAGE AND IDLE FACILITY TIME

Data	Heuristic*											
	1	2	3	4	5	6	7	8	9	10	11	12
Project Set 1	995 3220	849 1681	966 2059	1028 2801	963 1527	812 1592	675 1364	712 915	682 1246	1265 3467	675 815	1006 1514
Project Set 2	593 1870	497 1598	497 1598	686 1925	681 1391	591 2299	474 1402	567 1507	568 1206	1071 2360	609 1645	727 1690
Project Set 3	495 1089	467 854	442 973	511 1089	468 1516	444 1231	306 781	247 973	401 1309	660 1280	374 1129	592 1187
Project Set 4	567 1744	538 1744	523 1966	559 1881	548 2276	443 1585	401 1521	402 1336	346 1428	974 2887	356 1315	953 2544
Project Set 5	144 1096	133 1025	133 1070	300 1170	302 1421	148 983	126 995	129 1094	120 1062	274 1653	291 1120	514 1372
Project Set 6	406 1028	314 848	314 903	256 1099	188 839	152 792	181 617	239 817	244 738	805 1685	240 827	442 1033

Source: J. H. Mize, *op. cit.*, pp. 101.

*See Appendix A for a discussion of the heuristics used.

The top figure in each cell represents program slippage and the bottom figure in each cell represents idle facility time. Since hypothetical data is used in this study, no units can be attached to project slippage and idle facility time. The best obtained project slippage and idle facility time are underlined for each set of data. Heuristics 7, 8, and 9 resolve schedule conflicts in such a manner that critical path violation is minimized.

Tables 7 and 8. These results were obtained by this author by performing a two-way analysis of variance on Mize's data. Duncan's Multiple Range Test was then used to rank the treatment means.

The statistical analysis only partially confirms Mize's conclusions. While the three scheduling rules with the lowest mean values do resolve local schedule conflicts in such a way as to result in minimum critical path violation, Duncan's test is unable to distinguish between these three rules and the first come, first served scheduling rule.

It is somewhat surprising that the SIO heuristic performs as poorly as it does. It is the highest ranking of the twelve scheduling rules. No apparent reason exists for why the SIO rule does not schedule the projects with less slippage. It is perhaps possible that the projects examined contain a few activities of relatively long duration which cause the entire project to be delayed a significant amount.

2.4 R. M. Knight

Knight's investigation of the multi-project scheduling problem differs from Mize's by using actual data (from the Caterpillar Tractor Company); it is similar in that the

TABLE 7

ANALYSIS OF VARIANCE FOR MIZE'S PROJECT SLIPPAGE

Source	d.f.	S.S.	M.S.	F	$F_{\alpha=0.01}$
Treatments	11	1,284,603	116,782	14.3*	2.59*
Blocks	5	3,396,968	679,394	83.2	3.37
Error	55	449,109	8,166		
Total	71	5,130,680			

*Note that treatments are highly significant in this analysis.

TABLE 8

MULTIPLE RANGE TEST FOR MIZE'S PROJECT SLIPPAGE

Significance	Treatment Number	Mean
	10	833
	12	706
│	4	557
│ │	1	533
│ │	5	525
│ │ │	3	479
│ │ │	2	466
│ │	6	432
│ │	11	424
│	9	394
│	8	383
│	7	361

A vertical line enclosing a group of means indicates that the means located within the group cannot be distinguished from one another at the 5% level of results are not subjected to statistical analysis. His general conclusions are as follows:

> The features of the scheduling rules discussed demonstrate that rules based on resource usage by each project are superior to those based on project length alone. Significant results also demonstrate that re-scheduling of projects based on resource conflicts during the first feasible schedule obtained can be used to improve the schedule, especially for the case of widely varying project characteristics.[5]

Knight's results appear in Table 9.

Knight's project set F is the most comparable of the six sets to the types of projects examined by Mize. Although project slippage (Mize) and project length (Knight) are not directly comparable, it is interesting to note that Mize achieves his best results using rules which consider the total float of an activity, while Knight achieves his best results using a rule which considers the usage of resources by a project. The discrepancy may possibly be explained by the fact that Knight assumes all projects are linear in nature and have no parallel activities. This means there is only one path through each network, so that no activity ever has any slack. Thus the results of these two research efforts are not strictly comparable.

The statistical analysis of Knight's results shown in

---

[5] R. M. Knight, *op. cit.*, p. 2.

TABLE 9

PROJECT SET SCHEDULE LENGTHS

Project Set*	Heuristic**			
	SPN	LPN	RUR	RSR
A	26	25	25	24
B	43	32	32	30
C	26	25	25	24
D	38	33	33	31
E	34	34	30	28
F	40	38	33	28

Source: R. M. Knight, op. cit., pp. 45.

*Project sets consist of either ten or eleven projects with varying characteristics. Project Set F is the only set with both varying length projects and varying resource usage.

**SPN = Shortest Project Next

LPN = Longest Project Next

RUR = Resource Utilization Ranking

RSR = Resource Slack Ranking (A schedule is determined using this heuristic only after RUR has first been applied.)

Tables 10 and 11 do support his conclusions. Duncan's multiple range test is, however, unable to distinguish between his *best* scheduling rules, RUR and RSR.

It is somewhat encouraging that Knight's and Mize's data lend statistical support to the hypothesis that the methods of determining schedules significantly affect various indices of organizational performance. This seems to indicate that an investigation of heuristic rules for scheduling multiple projects is more likely to lead to significant results than Pascoe's study indicates.

Other literature on multi-project scheduling has the same format as the studies of Mize and Knight--a set of scheduling rules is applied to a set of projects and the rules which satisfy some criterion are deemed *better* than others.[6] In all instances, no statistical assessment of the scheduling ability of the rules is given; some rules perform better in a majority of cases and so are deemed superior rules.

This points to the need for determining which of the

---

[6] See for example the study by J. D. Brand, et al., "The Resource Scheduling Problem in Construction," *Civil Engineering Studies, Report No. 5*, Urbana, Illinois: Department of Civil Engineering, University of Illinois, 1964. and the text by R. L. Martino, *Project Management and Control: Vol. III, Allocating and Scheduling Resources*, New York: American Management Association, 1965.

TABLE 10

ANALYSIS OF VARIANCE FOR KNIGHT'S PROJECT SCHEDULE LENGTHS

Source	d.f.	S.S.	M.S.	F	$F_{\alpha=0.01}$
Treatments	3	156	52	8.33	5.42
Blocks	5	416	83	13.34	4.56
Error	15	93	6		
Total	23	665			

TABLE 11

MULTIPLE RANGE TEST FOR KNIGHT'S PROJECT SCHEDULE LENGTHS

Significance	Treatment	Mean
	SPN	34.5
	LPN	31.2
	RUR	29.7
	RSR	27.5

A vertical line enclosing a group of means indicates that the means located within the group cannot be distinguished from one another at the 5% level of significance.

available scheduling rules can be statistically described as being able to satisfy certain project scheduling objectives. Given a set of heuristics which have these proven abilities, one is more likely to construct effective scheduling algorithms.

APPENDIX A

APPENDIX A

MIZE'S HEURISTIC SCHEDULING RULES

1. The first decision rule resolves local schedule conflicts according to the lowest EFT discipline. Ties are broken by the lowest float time discipline.

2. The second decision rule begins by generating a feasible schedule using the lowest EFT discipline. An attempt is then made to improve this schedule by a gap-closing procedure.

3. Decision rule 3 begins by scheduling activities according to the lowest EFT discipline. It then attempts to improve the schedule by sequencing all jobs leading into an event to be completed at the largest completion time among all of the jobs (if such movement does not create a schedule conflict). An attempt is then made to apply gap-closing procedures to the schedule.

4. Decision rule 4 begins by calling the logic of the LBOUND routine into action. (The LBOUND is the minimum amount of slippage a particular project can experience without causing some other project to be penalized). Decision rule 4 assigns higher priorities to projects with smaller LBOUNDS. The rule next returns all facility queues to their original states and resolves conflict sets

according to the priorities discussed above. An attempt is then made to improve the schedule as described in Rule 3.

5. Decision rule 5 is the same as rule 4 except that facility queues are not returned to their original statuses. This is an attempt to begin the scheduling process with jobs as near as possible to the "best" positions for them in their respective queues. These are the positions at which the LBOUND values are obtained.

6. Decision rule 6 begins by calling LBOUND and establishing project priorities. However, this rule also gives consideration to the criticality of each job involved in a conflict set. Criticalness is measured by the amount of float a particular job possesses at the time of conflict. The logic is as follows:

1. If two jobs are in conflict and both are critical (i.e., both have zero or negative slack time available) use project lower bound priorities to resolve the conflict.

2. If two jobs are in conflict and neither are critical (i.e., both have positive slack time available), use project lower bound priorities to resolve the conflict.

3. If two jobs are in conflict, one being a critical job and the other a non-critical job, schedule the critical

job first and the non-critical job second.

An attempt is then made to improve the schedule as in rule 4.

7. Decision rule 7 resolves local schedule conflicts to minimize violation of critical paths. When a conflict set is encountered, rule 7 examines all feasible resolutions of the conflict and notes the total amount of exceeded Latest Finish Time resulting from each resolution. That resolution having the lowest such value is selected as best and the jobs are ordered accordingly. After facility queues are disturbed by the resolution of conflicts in other facilities, they are rearranged according to the EFT discipline, i.e., lowest EFT first, next lowest EFT second, etc. The two-stage process of highest EFT conformity and gap-closing procedure as described in rules 3 and 4 is employed as a final attempt to improve the obtained schedule.

8. Decision rule 8 is identical to rule 7 except that it uses the lowest EST discipline in rearranging facility queues after they have been disturbed by conflict resolutions in other facilities.

9. Decision rule 9 begins by calling LBOUND into action. LBOUND computes lower bound values by determining the best

arrangement of jobs within each facility, disregarding between-facility interaction. Decision rule 9 leaves the jobs in the positions determined by LBOUND and then operates exactly as rule 7.

Decision rules 10 through 12 are single attribute decision rules.

10. Shortest operation first--when a schedule conflict is encountered involving two consecutive jobs in a facility queue, always perform first the job whose expected operation time is smaller. This rule has received the most research attention as reported in the literature and has been shown to be optimum for the trivial case of a single machine or facility, and a given set of jobs.

11. First come, first served--when a schedule conflict is encountered, perform first the job with the smallest original EST value.

12. Longest remaining network time--when a conflict is encountered perform first the job which is farthest (in time) from the end of the project network to which it belongs. Although this is considered an "advanced" rule, it is still a simple one since it considers only one attribute of jobs in schedule conflicts.

CHAPTER 3

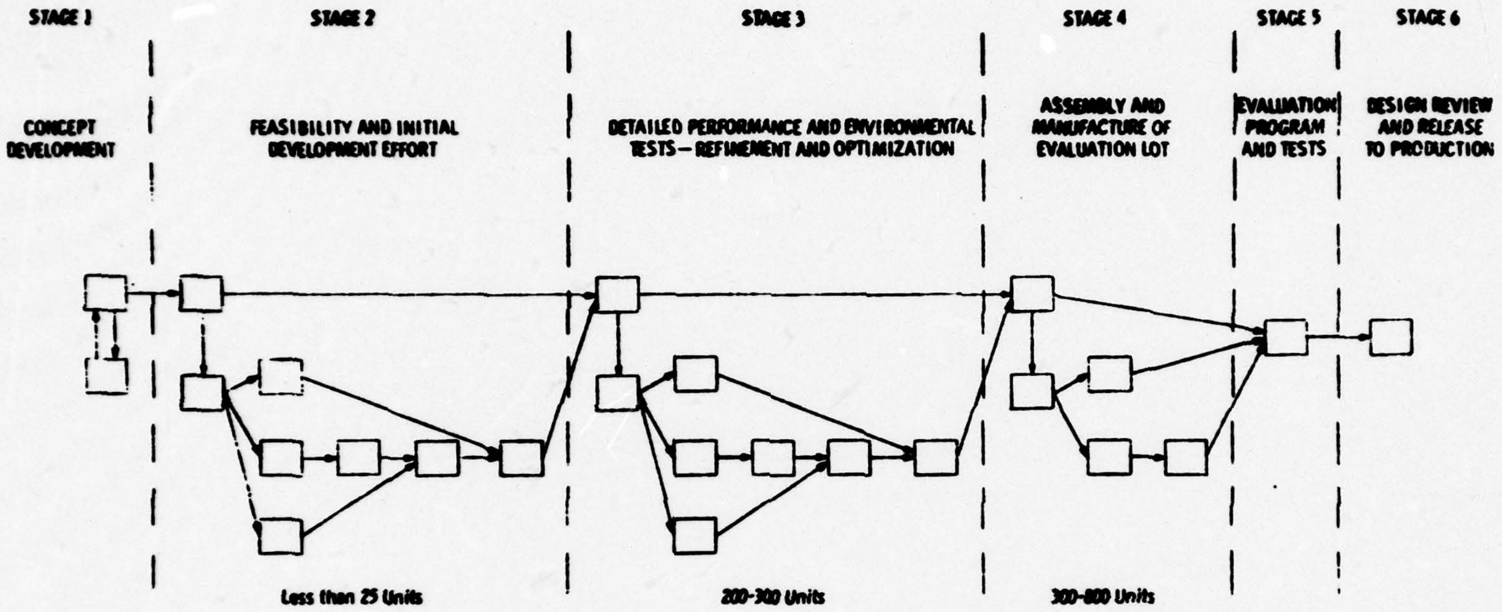
ORGANIZATIONAL SETTING AND ASSUMPTIONS OF THE SCHEDULING MODEL

3.1 Description of Research and Development Operations

Data based on an actual scheduling problem at the Research and Development Department of the U.S. Naval Ammunition Depot, Crane, Indiana, is used in the present study. The use of this data provides a more realistic challenge to the scheduling rules than the alternative of generating fictitious projects on a computer. A network diagram showing the sequence of activities of a typical project in this R&D organization is presented in Figure 4.

Each project consists of approximately six distinct stages, the number of stages fluctuating because of a product or test failure within a stage or because of prior development work in a component item. Within each stage, many items are manufactured and then tested. Depending upon the results of this testing, the entire stage is either repeated or the product moves to the next stage for further developmental work. In the final stage, a report is issued either to release the item for production or to recommend that it not be produced.

R&D PROJECT FLOW



RESOURCES AVAILABLE

Type	Numbers
ENGINEERS	21
CHEMISTS & PHYSICISTS	14
TECHNICIANS	20
*MACHINISTS	8
ASSEMBLERS	30
TOOL DESIGNERS	6
PROCURERS	4
*PERFORMANCE EVALUATORS	10
ELECTRONIC EQUIPMENT DESIGNERS	6

*ADDITIONAL RESOURCES AVAILABLE - NON R&D

Figure 4

The assignment of resources to a particular activity in this organization is complicated by two factors. First, approximately thirty-four projects simultaneously compete for the resources available. This makes the problem of resource assignment more difficult because the effect of a delay on both the activity and the project must be considered. A delay in one activity could possibly affect thirty-four different projects.

This scheduling situation is also complicated by the fact that many of the projects compete for the services of the organization's job shop. The scheduling of projects must then be concerned with the effects of project scheduling on *imbedded* job shop efficiency. This is why the SIO rule is included in the analysis. It is possible that a heuristic which has performed well in a job shop environment will perform well in scheduling projects which consist partially of activities which demand job shop services.

Figures 5 and 6 and Table 12 show the distribution of selected project parameters of the projects considered. These projects represent all the work performed by this organization over an eight month period, the length of time for which data is available. If a total project cannot be

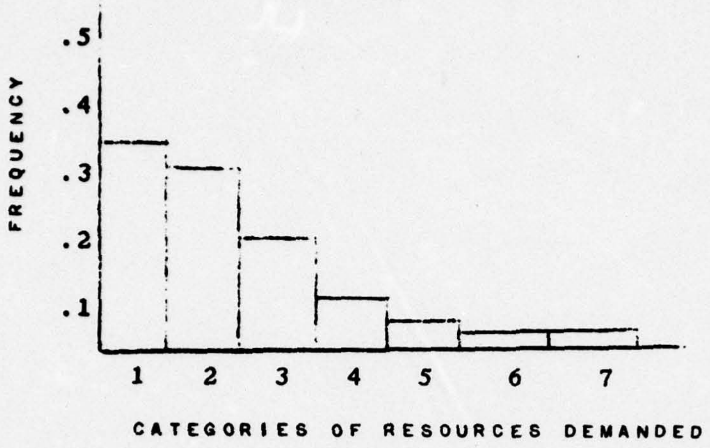


Figure 5: Distribution of Categories of Different Resources Demanded by Each Activity

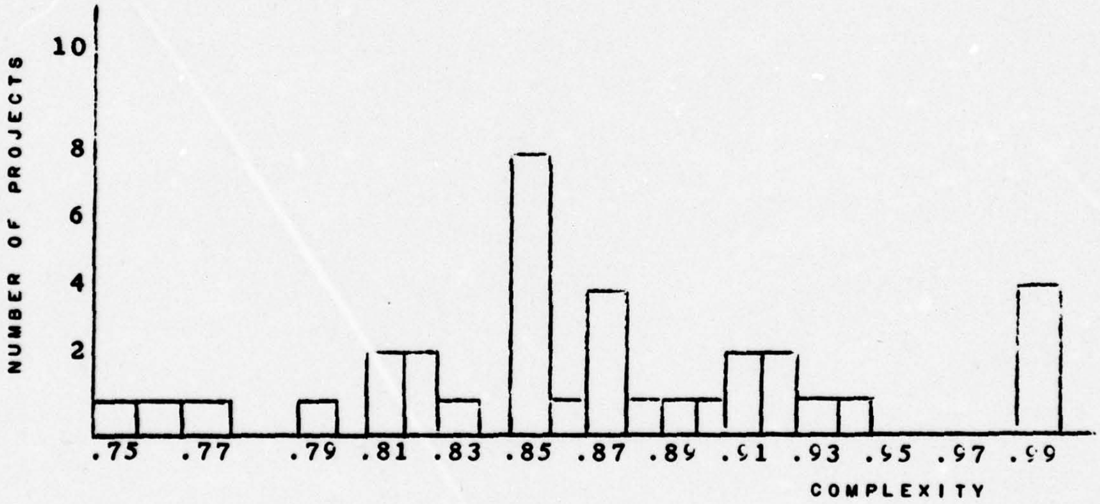


*Figure 6: Distribution of Project Complexity**

*COMPLEXITY IS EQUAL TO THE NUMBER OF NODES (N) IN A PROJECT DIVIDED BY THE NUMBER OF ARCS (M). IN THE RELATED JOB SHOP PROBLEM, THE COMPLEXITY IS ALWAYS EQUAL TO N/(N-1).

TABLE 12

CHARACTERISTICS OF PROJECTS SCHEDULED

Project Number	Number of Activities In Project	Total Man Days Required	Expected Length (In Days)
1	24	1125	136
2	11	728	105
3	22	639	90
4	12	365	54
5	15	153	91
6	13	89	54
7	15	317	101
8	13	447	99
9	7	129	93
10	22	393	90
11	17	393	96
12	12	111	72
13	10	297	45
14	17	512	81
15	7	255	70
16	10	2278	153
17	16	280	70
18	25	355	81
19	14	356	104
20	14	113	55
21	6	109	46
22	15	402	76
23	12	274	52
24	18	527	96
25	29	201	42
26	25	2315	159
27	19	341	97
28	26	382	113
29	30	965	105
30	24	382	156
31	13	259	80
32	20	151	65
33	19	717	134
34	28	462	103

technologically completed within the eight-month period, only the portion which can be completed is considered. Note that project complexity is here defined differently from Pascoe's definition because of the activity orientation used in this study.

The data used in this study is not meant to be representative of every project scheduling situation. The primary interest here is in examining how different scheduling rules affect the performance criteria discussed in Chapter 4. Only by the sampling of heuristic rules and the application of these rules to a variety of actual scheduling problems can it be seen how the rules affect performance criteria. The present study considers a sample of a particular class of scheduling problems, but it is recognized that additional samples will have to be taken from other scheduling situations in order to increase the inference space of conclusions to a universe of project sets.

3.2 Assumptions of the Scheduling Models

The following assumptions are necessary for applying the scheduling rules discussed in section 4.1 to the project set described in section 3.1. The assumptions made when addressing a problem of the magnitude considered in this study are based on real world or practical scheduling considerations and limits on computer hardware/software availability. Of all assumptions listed, the first two are felt to be the most restrictive for the problem considered.

1. No interruptions of jobs[1] are allowed; once a job is begun, it must be continued to completion. This requirement prevents haphazard scheduling of jobs at arbitrary time periods throughout the schedule. For the projects examined in this study, the assumption is valid with the exception of a few of the engineering and drafting jobs. Most operations must, however, be completed once begun. For the operations that can be split, it is necessary to estimate the time and resources required as separate component jobs. This has been done for those engineering and drafting jobs mentioned and constitutes a relaxation of the no interruption assumption.

2. Resources must be used by an activity at a constant rate. Relaxation of this assumption can be accomplished by dividing a job into component jobs that require differing resource amounts. (Future research is directed toward establishing any bias introduced by the assumption of a fixed resource usage rate.)

---

[1] Supra, fn. 4, chap. 1, p. 2.

3. Man power and resources are assumed synonymous. That is, if an activity is assigned a certain type of man power, it is assumed that the men have the equipment available to perform the task. This assumption is valid in a variety of R&D organizations where man power is the limiting resource, and equipment is available as needed.

4. Resources are assumed to be available at a constant rate throughout the length of a schedule. This assumption is inherent in resource constrained scheduling problems and is valid for the problem considered. R&D organizations are frequently reluctant to dismiss workers and generally are unable (except through overtime) to supplement the work force because of the specialization of work they perform.

5. The duration of an activity is assumed to be known with certainty. Although numerous network-based scheduling techniques use probability estimates for job durations, the assumptions underlying the use and interpretation of these estimates limit the extent to which they can be applied in scheduling projects.[2] In addition, Muth has shown in

---

[2] See for example the article by Frank E. Grubbs, "Attempts to Validate Certain PERT Statistics or 'Picking on PERT,'" *Operations Research*, Vol. 10, No. 6, (November-December 1962), pp. 912-915 or the article by Richard M. Van Slyke, "Monte Carlo Methods and the PERT Problem", *Operations Research*, Vol. 11, No. 5, (September-October 1963), pp. 839-851.

a related job shop scheduling problem that the scheduling procedures developed are rather insensitive to moderately large errors in estimated job times.[3]

6. Any set-up or transport times are included in the duration of the job. This assumption is very reasonable and can be included in the definition of a job's duration when estimating time and resource requirements.

---

[3] J. F. Muth and G. L. Thompson, (eds.), *Industrial Scheduling*, New Jersey: Prentice-Hall, Inc., 1963, Chapter 18.

CHAPTER 4

DESCRIPTION OF SCHEDULING RULES AND PERFORMANCE CRITERIA

4.1 Scheduling Rules Examined

A variety of heuristic scheduling rules for multi-project problems can be found in the literature. The studies of Mize[1] and Knight[2] give only a sampling of the number available. Thus it is imperative that some scheme be developed for classifying the available rules; once classified, samples from each group can then be taken and applied to the multi-project scheduling problem, with the classification scheme giving some assurance that the population of available heuristics has been adequately sampled.

Attempts by this author to classify scheduling rules into homogeneous units led to the following conclusions:

1. Numerous scheduling rules are based upon a well-defined objective, such as minimizing the completion time

---

[1] *Supra*, chap. 2, p. 29.

[2] *Supra*, chap. 2, p. 31.

for all projects or maximizing the utilization of employed resources.

2. Other scheduling rules relate to a characteristic of the project set or the activities comprising the project sets. For example, *longest project next* assigns resources to the activity of the project having the longest expected span; any remaining activities are scheduled only if resources permit.

3. Some scheduling rules do not relate to an objective function or to a characteristic of a project set, but rather use other scheduling rules as a basis on which to build. *Switch activities at random in the order in which they are considered for resource assignment when they tie on another priority* is an example of such a scheduling rule.

4. Finally, some scheduling rules base their rationale on learning from knowledge gained in previous scheduling efforts, and thus are applied only after another scheduling rule has been tried. *Reschedule based on resource conflicts exhibited in the first feasible schedule* is an example of such a rule.

In addition to classifying scheduling rules, the actual scheduling of various projects using heuristic methods gives an indication of the scheduling ability and interaction effects of many of the rules available. For example, in repeated attempts to solve a project scheduling problem posed by Martino[3] which does have a known optimum solution,[4] this author found that three heuristics--one based on usage of total resources, one based on the resources required by an activity, and one based on randomness--are necessary if one is to obtain the optimum solution. Problem solving experiences such as this one point up heuristic scheduling rules that seem to be good (1) from an intuitive viewpoint and (2) from the resulting solutions when the heuristics are applied to different project scheduling problems.

Based on a survey of scheduling rules employed in the literature, a method of classifying these rules for adequate sampling, and this author's computational experience, the following heuristic scheduling rules will be evaluated on their ability to schedule multiple projects occurring simultaneously within a firm. All

---

[3] R. L. Martino, *Project Management and Control: Vol. III, Allocating and Scheduling Resources*, New York: American Management Association, 1965.

[4] The critical path length for this project is 19 days and the project can be accomplished with as few as 8 men in a 19-day period. Because the project can be completed in 19 days with no variation in resource usage and no idle resource time, it is considered optimum.

TABLE 13

SCHEDULING RULES EXAMINED IN THIS STUDY

Abbreviation	Title	Basis for Inclusion
LTF	Least Total Float	Take advantage of slack present in an activity
GTRD	Greatest Total Resource Demand	Schedule potential bottleneck activities
GRRD	Greatest Remaining Resource Demand	Look-ahead further than the current activity in scheduling potential bottleneck activities
SIO	Shortest Imminent Operation	Process as many jobs through the system as rapidly as is possible
GRU	Greatest Resource Usage	Schedule all resources that can be scheduled during a time interval
RAN	Randomness	Add an element of chance in the assignment of resources to an activity
RES	Reschedule	Reschedule activities on the basis of resources that had the least idle time in the first schedule

rules are applied in conjunction with the parallel[5] method of scheduling. All rules refer to the choosing of one activity over another for resource assignment at some point in time because of a higher (or lower) value of a priority or other index associated with the activity.

1. Least Total Float (LTF)

Schedule the activities on the basis of total float, those with the least total float being considered first. Mize's results suggest that a rule which results in minimum critical path violation produces shorter duration schedules than do many other scheduling methods.

2. Greatest Total Resource Demand (GTRD)

Schedule each activity on the basis of the total resources demanded by that activity, that with the greatest usage of resources being scheduled first. The rationale behind this rule lies in scheduling potential bottleneck activities because of the high usage of varied resources

---

[5] In the parallel approach to project scheduling, one time interval is considered and all the activities that can be scheduled during this time interval are scheduled. Time is then advanced by one unit and all activities that can be scheduled during the current time interval are considered, etc., until a feasible schedule has been developed.

the activity consumes. The priority index for an activity is calculated as follows:

$$\sum_{i=1}^{q} r_{ij} \cdot d_j$$

3. Greatest Remaining Resource Demand (GRRD)

Schedule the activities on the basis of the total remaining man-hours (resources) of work required on the project, those with the greatest remaining man-hours being scheduled first. This rule is related to many of the *look-ahead* rules receiving attention in the literature today. It attempts to look ahead further than the current activity in locating potential bottlenecks. It is an extension of GTRD and changes values dynamically throughout the development of a schedule. Activity priority is calculated as follows:

$$\sum_{a_j \varepsilon \theta_m} \left[\sum_{i=1}^{q} r_{ij}\right] \cdot d_j$$

4. Shortest Imminent Operation (SIO)

Schedule the activities on the basis of the duration of the activity, those with the shortest duration being scheduled first. This is analogous to the SIO rule of the job shop. The rationale behind this rule is to get as many jobs (activities) through the system in as short a length of time as possible.

5. Greatest Resource Utilization (GRU)

Schedule the activities at time t so that the maximum available man power at time t is utilized, given the constraints imposed by the resources available and the resources required per activity. This rule can be formulated and solved as a zero-one integer programming problem as follows, where $x_j = 1$ if a_j is in solution, and 0 if it is not:

$$\begin{aligned} \max \quad & \sum_j c_j x_j \\ \text{s.t.} \quad & R\,\underline{x} \le \underline{b}^t \\ & x_j = 0,1 \end{aligned}$$

The rationale behind this rule is that to an extent, project completion dates should be minimized if resources are fully employed. There is also the behavioral rationale that man power is usually more productive and more satisifed during periods of low idle time.

The application of this rule demonstrates two important points. First, heuristic scheduling rules are not always based solely on a priority index for an activity and subsequently cannot be implemented simply through sorting routines. Second, although heuristic rules are based on intuitive rules-of-thumb, the application of these rules often requires a greater level of sophistication than that necessary for formulating them. Simple heuristics often require the tools of mathematical programming to institute their logic.

6. Randomness (RAN)

If any activities which are candidates for scheduling during the current time interval tie on any given scheduling rule, switch them at random in the order in which they are considered for resource assignment. Of course, the rule is never applied by itself. Despite the simplicity of randomness, attempts by this author and others[6] to solve certain project scheduling problems without it have been unsuccessful.

In this rule, a list of candidate activities is searched, and if any two, three, four, etc., tie on any given rule, a random number or a string of random numbers is drawn. If a random number is less than 0.05, the corresponding activities are switched in their order for

---

[6] J. D. Weist, "A Heuristic Model for Scheduling Large Projects with Limited Resources," *Management Science*, Vol. 13, No. 6, (February 1967), p. 368.

consideration of resource assignment; if a random number is greater than or equal to 0.05, the activities remain in their previous sorted order.

7. Reschedule (RES)

Determine which class or category of resource had the least total amount of idle time in the first feasible schedule. Develop a new feasible schedule using the amount of this resource required by each activity multiplied by the duration of that activity as the new scheduling priority. Denote this tight resource by r_i'. Then the index for reschedule becomes:

$$r_{ij} \cdot d_j$$

Many researchers claim that improvements can be made in schedules either by rescheduling on the basis of a resource that caused more delays in activities in the first schedule than did other resources or else by including provisions for gap-closing by rescheduling those activities that are scheduled to start before their technological late start time (LST). The reschedule feature used in this research is representative of the former reschedule routine.

4.2 Methods of Evaluating System Performance

There is admittedly a large number of criteria that can be evaluated in assessing the efficacy of the heuristic scheduling rules described in section 4.1. In limiting the number of them actually examined, one must consider (1) criteria that seem reasonable to the firm and (2) criteria that have been evaluated in previous research efforts, in order that comparisons can be made.

In addition to evaluating the system performance, one is usually interested in the cost or the price paid for the system by the firm. In this case, the cost is the price paid (in computer time) for developing the different schedules. One large benefit derived from using heuristic solution methods over strict mathematical programming routines is the ability to generate solutions rapidly and therefore inexpensively. Using heuristic techniques enables one to assess the effects of a change in assumptions or a change in input parameters on the system at a relatively low cost when compared to the cost of a guaranteed optimum solution (when one is available using mathematical programming methods).

In light of the above considerations, the following factors will be used to assess the efficacy of the scheduling rules and scheduling programs (combinations of scheduling rules) considered in this research.

1. Total Slippage

For each scheduling rule and combination of rules used in developing a schedule, evaluate the delay beyond critical path completion time estimates for all projects considered.

It is doubtful that many of the projects considered in a resource constrained scheduling problem will be completed within the critical path completion time estimate. This slippage measure does, however, give an indication of the project slippage due to resource availability and the scheduling rules employed (with comparisons between rules possible by holding resources constant).

2. Weighted Total Slippage

This criterion is similar to the above, except that project slippage is weighted by the total resources demanded by a project. This measure places additional emphasis on the seriousness of a delay in the larger projects.

3. Total Resource Idle Time

Evaluate for all schedules developed the amount of time that resources were idle during the completion of that schedule. This measure indicates the amount of time resource groups were not actively working on a project due to the unavailability of direct project work which in turn is a result of the scheduling method employed.

4. Computer Processing Time

Evaluate the computer processing time expended in developing each schedule. This measure is an indication to management of the expense of using the scheduling rules either singly or in combination with one another.

CHAPTER 5

DESCRIPTION OF MPSP MULTI-PROJECT SCHEDULING PROGRAM

5.1 Introduction

Because of the magnitude of the problem considered in this research, it is necessary to develop a computer program to implement the scheduling rules of Chapter 4. This chapter discusses the computational aspects of such a program and designates the size of problems that it is set up to handle.

The program is called MPSP, an acronym for Multi-Project Scheduling Program. Two versions of MPSP have been developed; the size of core memory available determines which version should be utilized. The simplest version implements all scheduling rules except Greatest Resource Utilization and is suitable for use on a 32K word machine. The second version implements all scheduling rules and requires a minimum of 64K words. The difference in memory requirements between the two versions is a result of whether or not a zero-one integer programming code necessary for implementing the Greatest Resource Utilization heuristic is present.

The program can handle scheduling problems consisting of thirty-nine projects with as many as forty activities per project. In addition, thirteen different resource groups can be utilized by each activity. There is practically no limit to the number of time periods that can be handled by the code.

MPSP is written in FORTRAN IV for several reasons: the author is most familiar with this language; it is one of the most flexible languages; and it is available on most computers. The program is organized in modular form with a main scheduling program and several subroutines. The modular form of the program enhances flexibility in such aspects as the ability to augment or change the scheduling rules considered. Such a form also makes it easier to detect errors when building or running the program.

MPSP has been run on both the Control Data 3600 and the 6600 computers. Minor modifications can make it available for other machines. Chapter 6 will discuss the computer processing times required for implementing the scheduling rules considered. These times will, of course, have to be appropriately adjusted if the code is run on a different computer.

5.2 Structure of MPSP

MPSP is structured around a main ASSIGN routine that receives a list of ranked activities which technologically can be scheduled during the current time interval. The ASSIGN routine then considers the activities in the order in which they were received and assigns scheduled start and finish dates to the activities using the resources currently available. Those activities not scheduled during the current time interval are postponed until the next scheduling interval where they once again compete with other activities for available resources.

The ASSIGN routine is a variable time increment model. In this model, a group of activities is scheduled, and time is then advanced in days to a point where resources become available. This point in time is computed by finding the minimum assigned scheduled finish time of all activities currently being completed. This is the earliest that resources will be freed from ongoing activities to work on the activities not yet scheduled.

The list of activities that is passed from the ASSIGN routine to various subroutines and back is appropriately called a *decision set*. The subroutines involved determine which activities receive priority over other activities by instituting the logic of the scheduling heuristic. Those activities with the higher priority are consequently placed first in the decision set. For example, if the activities possessing the least total float are to be scheduled first, the appropriate subroutine passes the decision set back to the ASSIGN routine in increasing order of least total float. The ASSIGN routine then schedules all activities in the decision set that can be scheduled with the resources available.

This decision set is composed of (i,j) pairs denoting the project number and activity number of the activities that can be scheduled. These (i,j) pairs are stored in the arrays ISET and ISET1 respectively and are shown in Figures 8 and 9.

An EVENT array is common to the main ASSIGN routine and all subroutines. It contains all of the information relevant to an activity--the EST, EFT, resource requirements, assigned scheduled start and finish times, etc. In addition, the EVENT array stores various indices that are used in determining activity priority. For example, a ranking by least total float is accomplished by interchanging EVENT(I,J,10) with EVENT(I',J',10) in the decision set if the total float of the former is greater than that of the latter. Holding the EVENT and decision set arrays in common maintains efficiency of the program.

A flow chart depicting the logic of the main ASSIGN routine is shown in Figure 7. The subroutines that establish the priority of an activity and the critical path of a network are briefly discussed below.

1. Subroutine SORT

Subroutine SORT establishes the ordering of activities in the decision set. It is used in conjunction with all scheduling heuristics except GRRD and GRU.

In this subroutine, an index 'IN' is received from the main scheduling program. This index corresponds to the scheduling heuristic being examined. For example, if a ranking of activities based on activity duration is desired, IN is set equal to 1. Ranking is then accomplished by comparing pairs of EVENT(I,J,1) as shown in Figure 8.

2. Subroutine RANDOM

The logic of subroutine RANDOM is shown in Figure 9. In this subroutine, a list of ranked activities is searched and if any two, three, four, etc., tie on any given rule, a random number or a string of random numbers is drawn. If a random number is less than 0.5, the corresponding activities are switched in their order for consideration of

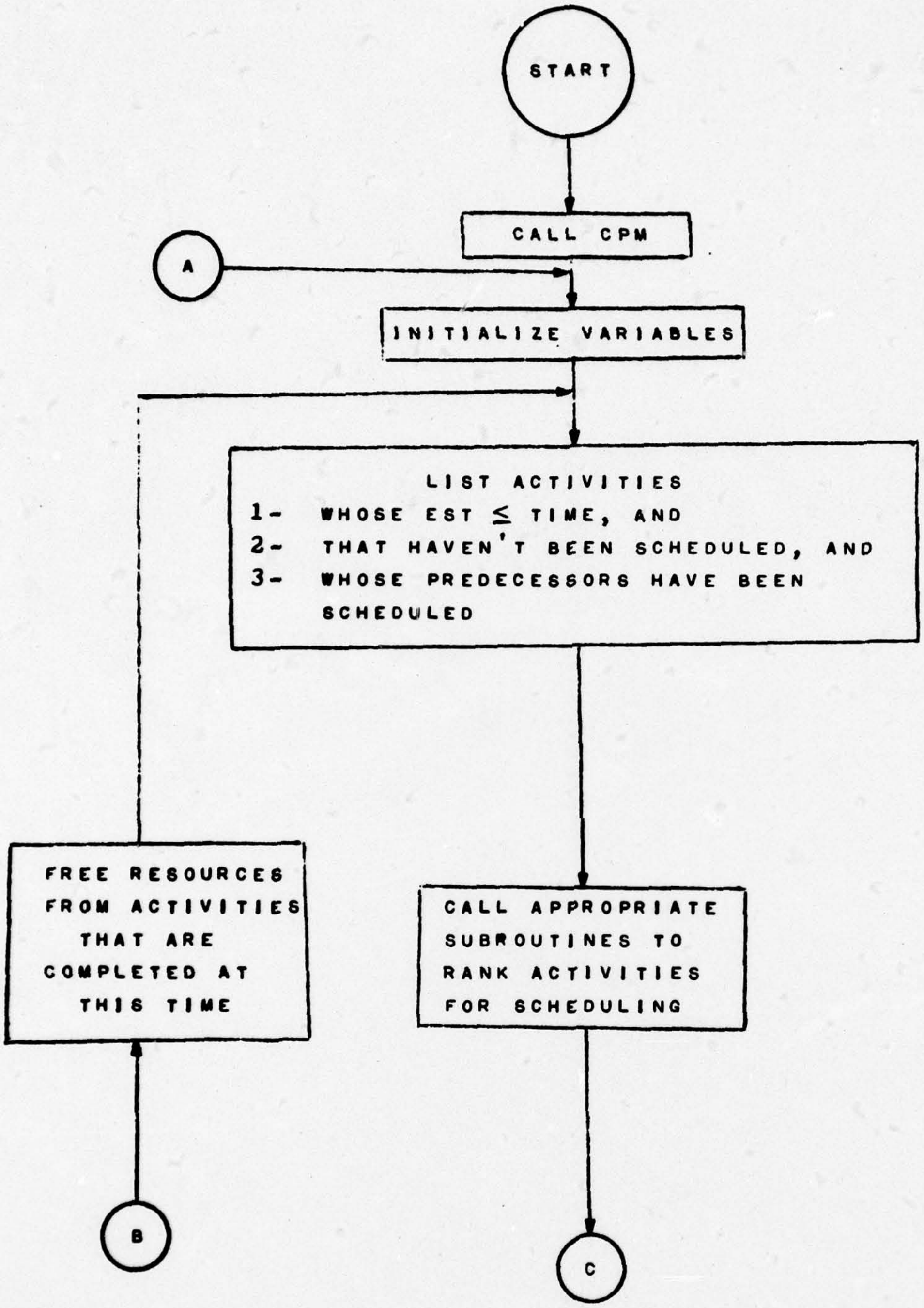


Figure 7: Flow Chart for Program ASSIGN

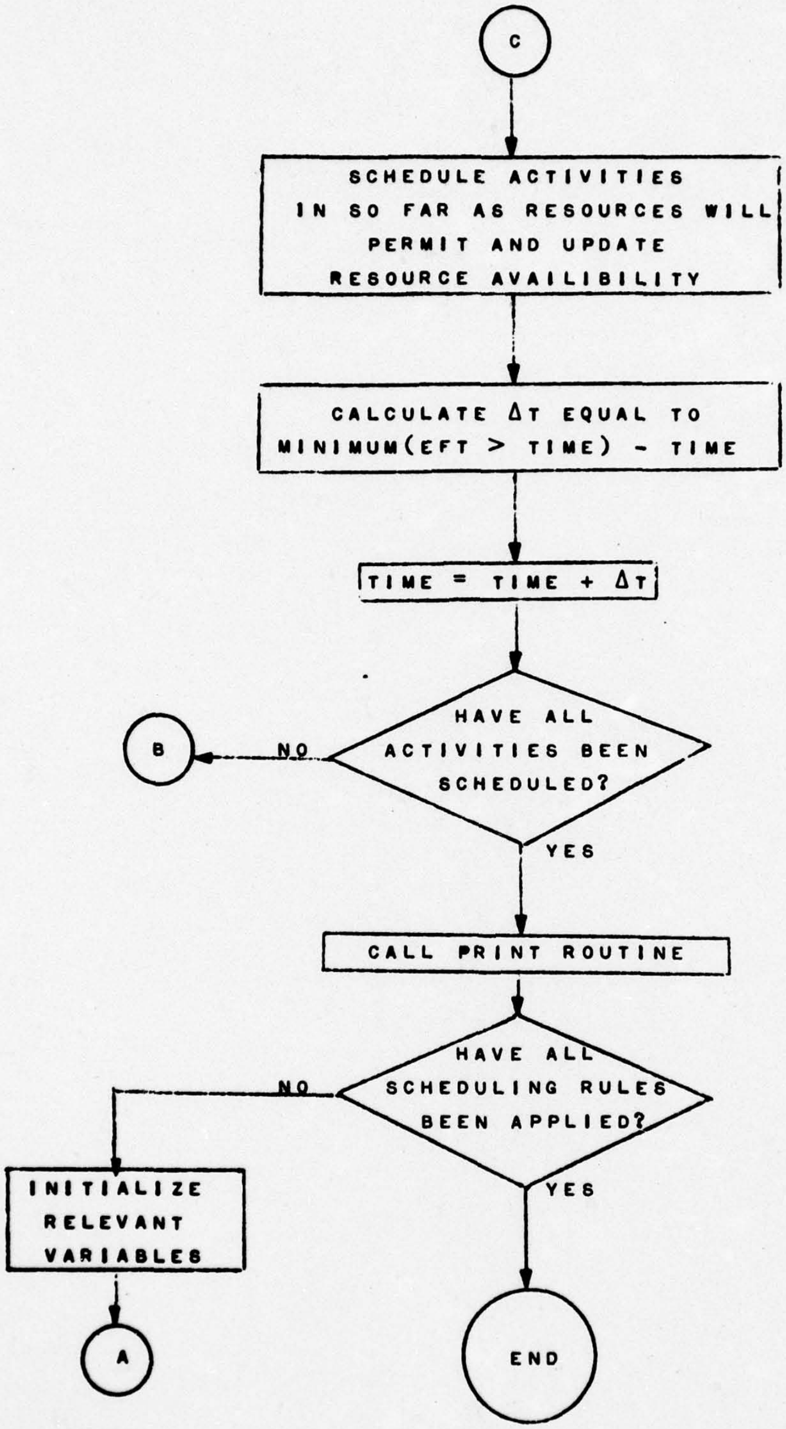


Figure 7: (contd.)

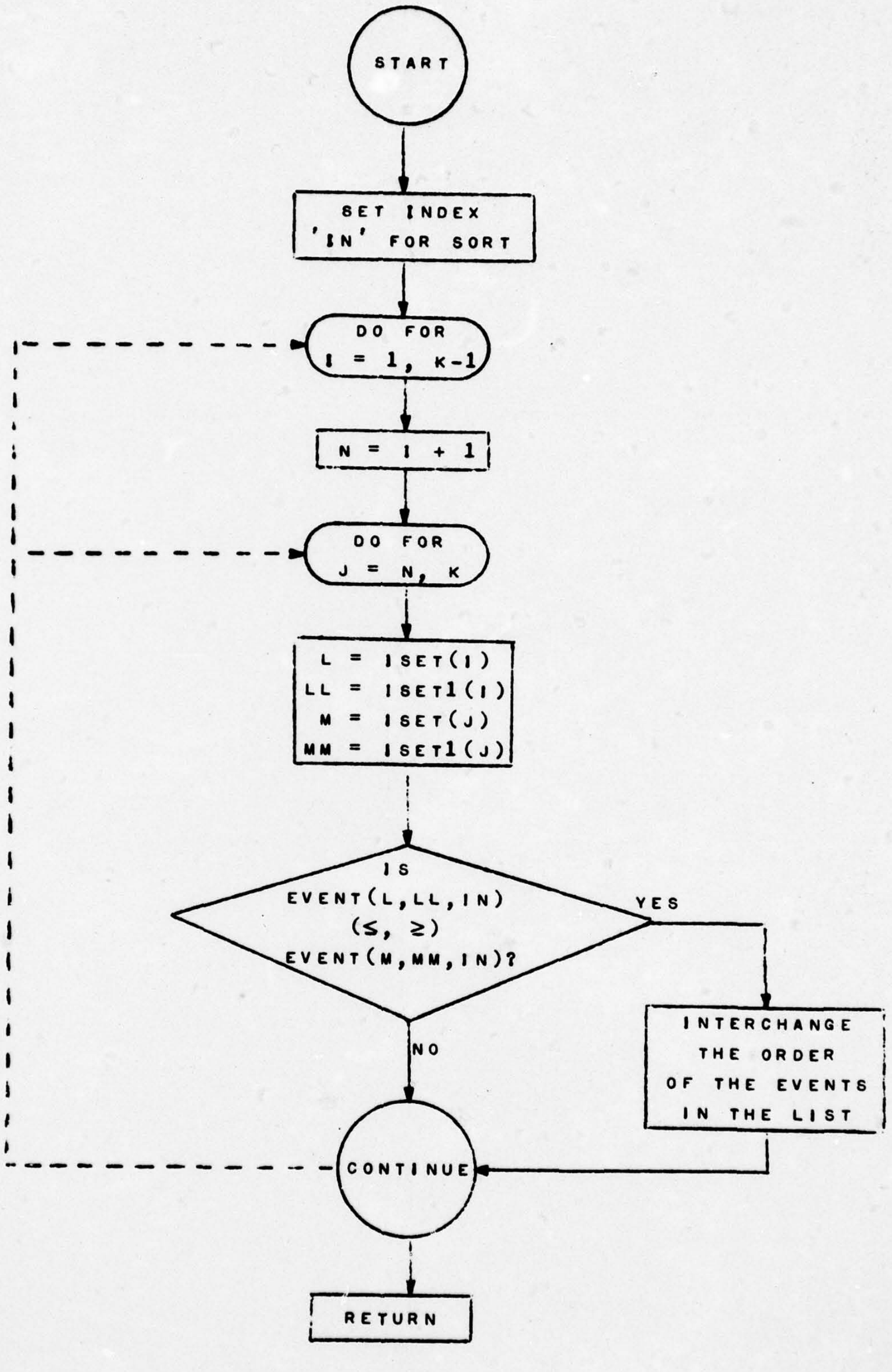


Figure 8: Flow Chart for Subroutine SORT

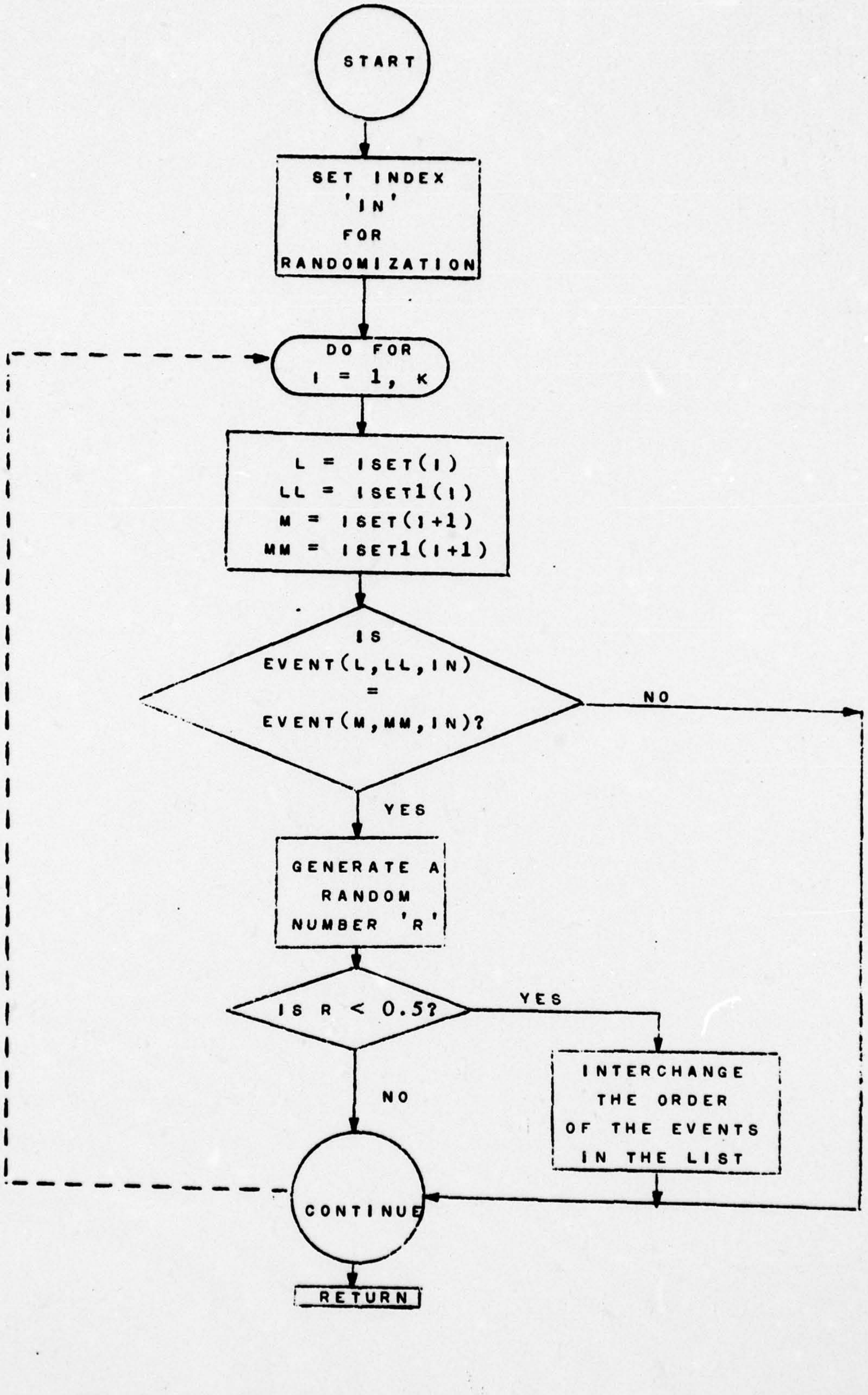


Figure 9: Flow Chart for Subroutine RANDOM

resource assignment; if a random number is greater than or equal to 0.05, the activities remain in their previous order.

3. Subroutine MAXRES

Subroutine MAXRES determines the quantity of resources needed to complete a project. It is required when the GRRD scheduling heuristic is being used to schedule a set of activities.

This subroutine examines all activities within a project; if an activity has not been scheduled and is not a current candidate for scheduling, its resource requirement is added to a running total. This running total then represents the total remaining resources needed to complete a project and becomes a priority index for activities in that project that can currently be scheduled.

The logic of MAXRES is depicted in Figure 10.

4. Subroutine CPM

Subroutine CPM establishes the EST, EFT, LST, LFT, TF, and FF of an activity as well as the critical path through the project network. It also reads the network data into memory prior to determining these critical path estimates.

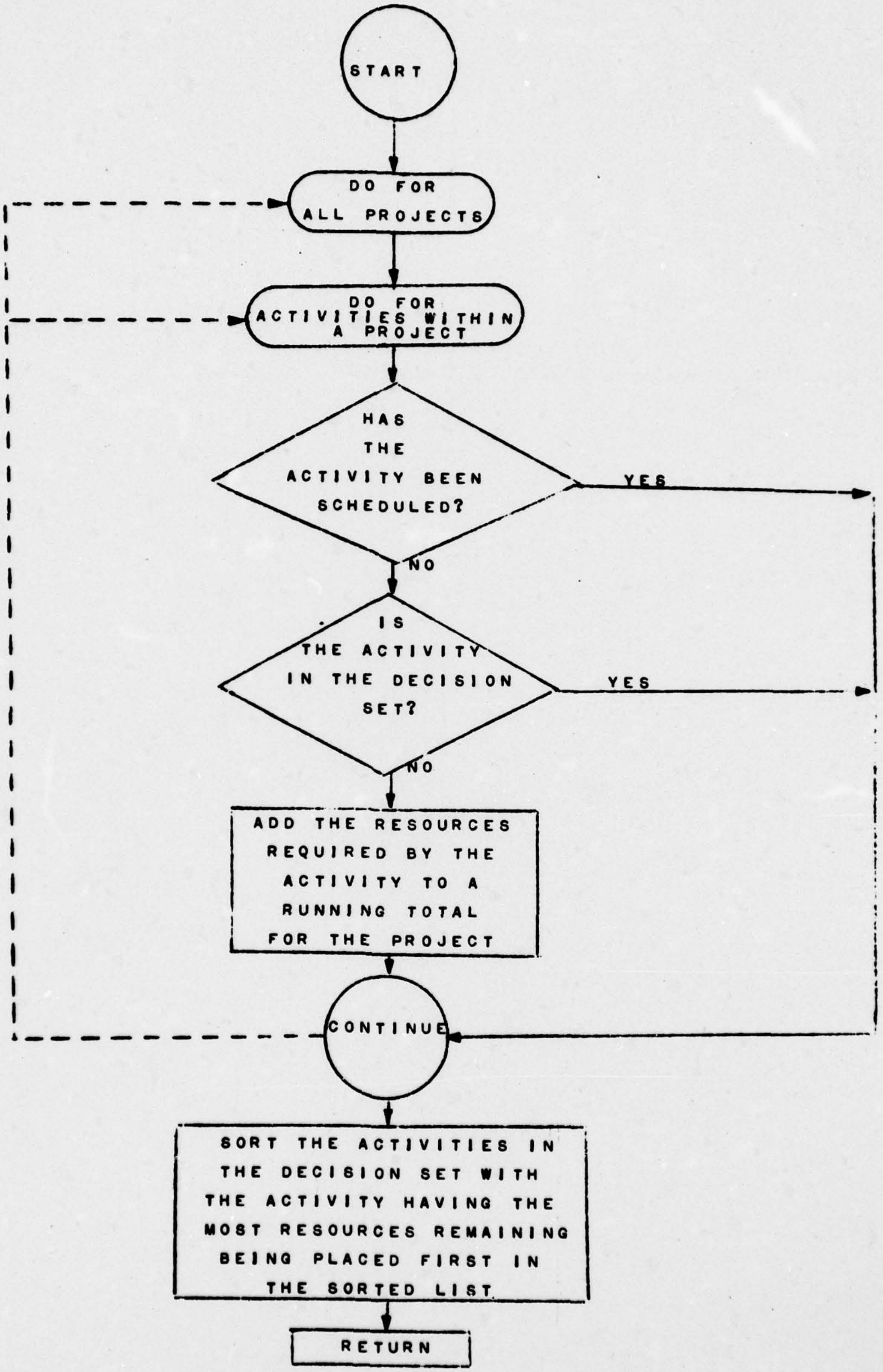


Figure 10: Flow Chart for Subroutine MAXRES

The logic of CPM is shown in Figure 11.

5. Subroutine DYNCRIT

A flow chart for DYNCRIT is not included because its logic is nearly the same as that of CPM. DYNCRIT updates the EST, EFT, LST, LFT, and TF of an activity if it has been postponed because of a lack of resources.

In DYNCRIT, the EST of a delayed activity is set equal to the current value of time, and all subsequent actitities reflect this updated EST. Thus any activity after being sufficiently delayed eventually becomes the most critical and moves to the top of a decision set.

6. Subroutine PRINT

The PRINT subroutine prints the relevant schedules and monitors the clock used in keeping track of the time required to develop a schedule. Output of the PRINT subroutine for one scheduling heuristic and one project is shown in Figure 12.

7. Subroutines SET01 and SIMPLE

Subroutines SET01 and SIMPLE solve the zero-one integer programming problems necessary for instituting the scheduling

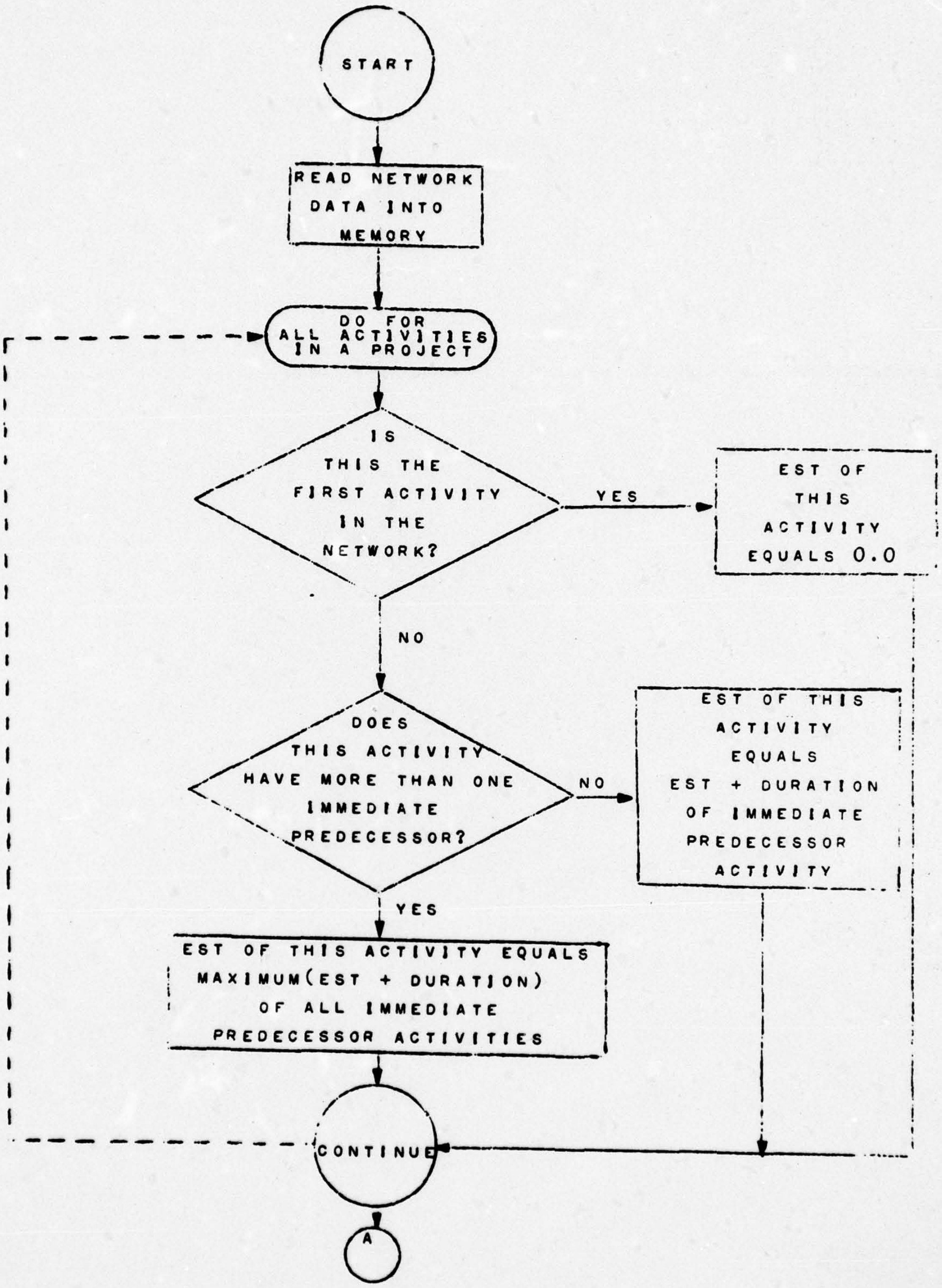


Figure 11: Flow Chart for Subroutine CPM

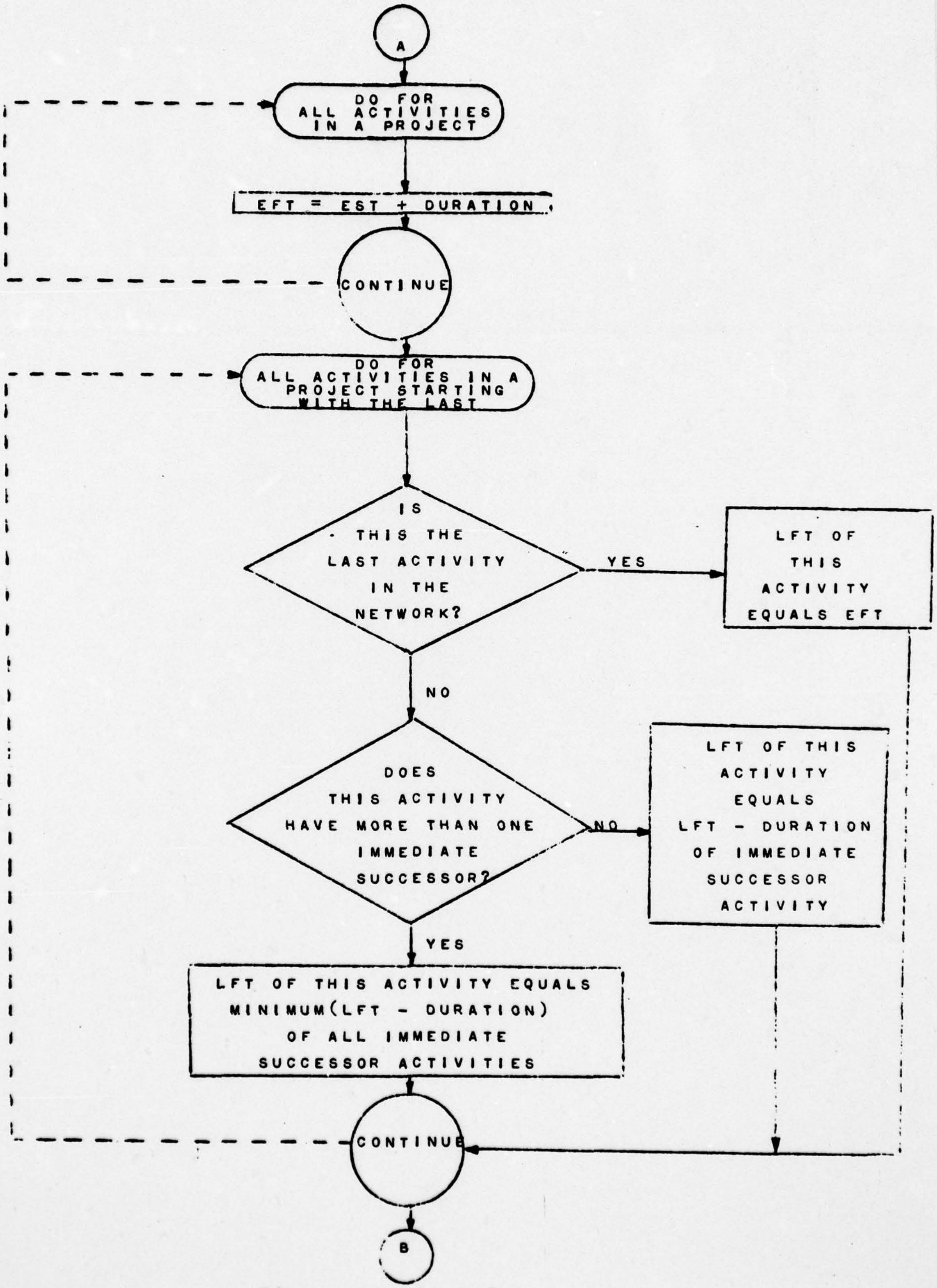


Figure 11: (contd.)

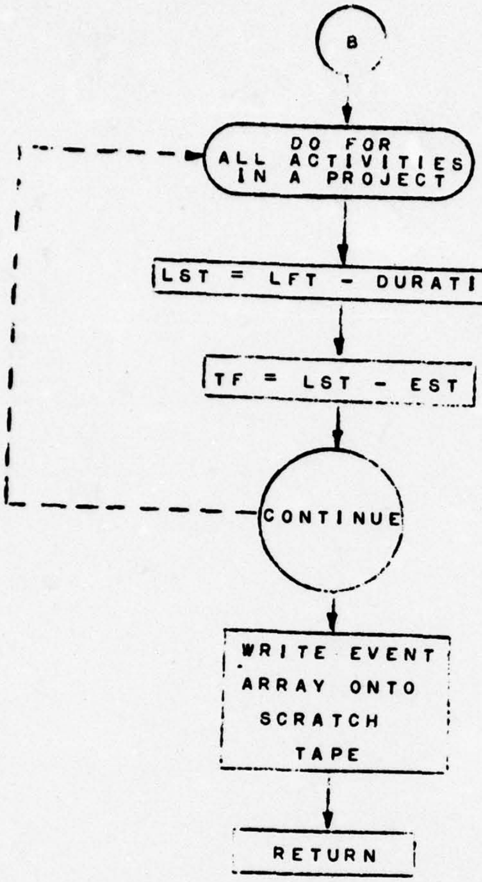


Figure 11: (contd.)

SCHEDULE - GREATEST REMAINING RESOURCE DEMAND WITH RANDOMNESS

PROJECT 31

EVENT NUMBER	DURATION OF EVENT N	EARLIEST STARTING TIME	EARLIEST FINISH TIME	LATEST STARTING TIME	LATEST FINISH TIME	SLACK TIME	ASSIGNED SCHEDULED START TIME	ASSIGNED SCHEDULED FINISH TIME	DELAY
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	10.0	0.0	10.0	0.0	10.0	0.0	0.0	10.0	0.0
3	9.0	10.0	19.0	26.0	35.0	16.0	75.0	84.0	49.0
4	6.0	10.0	16.0	10.0	16.0	0.0	38.0	44.0	28.0
5	10.0	19.0	29.0	35.0	45.0	16.0	87.0	97.0	52.0
6	15.0	16.0	31.0	16.0	31.0	0.0	45.0	60.0	29.0
7	9.0	16.0	25.0	44.0	53.0	28.0	49.0	58.0	5.0
8	11.0	29.0	40.0	45.0	56.0	16.0	97.0	108.0	52.0
9	10.0	31.0	41.0	31.0	41.0	0.0	92.0	102.0	61.0
10	6.0	41.0	47.0	56.0	62.0	15.0	102.0	108.0	46.0
11	12.0	41.0	53.0	41.0	53.0	0.0	102.0	114.0	61.0
12	5.0	53.0	58.0	53.0	58.0	0.0	114.0	119.0	61.0
13	11.0	40.0	51.0	56.0	67.0	16.0	111.0	122.0	55.0
14	2.0	51.0	53.0	67.0	69.0	16.0	122.0	124.0	55.0
15	7.0	47.0	54.0	62.0	69.0	15.0	108.0	115.0	46.0
16	8.0	54.0	62.0	69.0	77.0	15.0	124.0	132.0	55.0
17	12.0	58.0	70.0	58.0	70.0	0.0	119.0	131.0	61.0
18	5.0	62.0	67.0	77.0	82.0	15.0	133.0	138.0	56.0
19	7.0	67.0	74.0	82.0	89.0	15.0	138.0	145.0	56.0
20	6.0	67.0	73.0	83.0	89.0	16.0	138.0	144.0	55.0
21	6.0	74.0	80.0	89.0	95.0	15.0	145.0	151.0	56.0
22	4.0	70.0	74.0	70.0	74.0	0.0	131.0	135.0	61.0
23	5.0	70.0	75.0	83.0	88.0	13.0	131.0	136.0	48.0
24	13.0	74.0	87.0	74.0	87.0	0.0	135.0	148.0	61.0
25	2.0	87.0	89.0	87.0	89.0	0.0	148.0	150.0	61.0
26	11.0	89.0	100.0	89.0	100.0	0.0	150.0	161.0	61.0
27	9.0	75.0	84.0	88.0	97.0	13.0	136.0	145.0	48.0
28	2.0	80.0	82.0	95.0	97.0	15.0	152.0	154.0	57.0
29	3.0	84.0	87.0	97.0	100.0	13.0	154.0	157.0	57.0
30	5.0	100.0	105.0	100.0	105.0	0.0	161.0	166.0	61.0

Figure 12

heuristic Greatest Resource Utilization. Descriptions of these routines are available from the RAND Corporation.[1]

---

[1] A. M. Geoffrion and A. B. Nelson, *User's Instructions for 0-1 Integer Linear Programming Code RIP30C*, Santa Monica, California: The RAND Corporation, RM-5627-PR, (May, 1968).

CHAPTER 6

EXPERIMENTAL DESIGN AND TEST RESULTS

6.1 Introduction

This chapter presents the results of scheduling the project set described in Chapter 3 with each of the heuristic scheduling rules presented in Chapter 4. Questions relating to the design of the experiment are discussed, as is the method of obtaining replications of the data.

6.2 Experimental Design

Factorial experiments permit an investigator to evaluate the combined effect of two or more experimental treatments when they are used simultaneously. This type of experiment is also useful in evaluating test factors that have important causal effects independent of the values associated with other test factors. Since the objective of this study is to assess the scheduling abilities of various heuristic scheduling rules and heuristic scheduling programs (combinations of scheduling rules), a factorial experiment is an appropriate technique for assessing these abilities.

In this study, seven scheduling rules are examined. Five of these (LTF, GTRD, GRRD, SIO, and GRU)[1] are applied

---

[1] *Supra*, chap. 4, p. 54.

to a project scheduling problem independent of one another. These five scheduling rules are collectively called treatment A in the discussion that follows.

The remaining two scheduling rules (RAN and RES) are applied only in conjunction with one of the first five treatments. They are labeled treatments B and C, respectively. These rules attempt to improve upon the schedules developed by the other rules by introducing an element of *chance* in the assignment of resources to an activity or by applying knowledge gained through previous scheduling efforts. These last two rules exist at one of two possible levels: they are either present in a heuristic program or they are not. (This is to be distinguished from treatment A where one level of the factor must always be present in order to develop a schedule.)

Thus, in this project one factor (treatment A) is varied over five levels and each of two factors (treatments B and C) are varied over two levels. The project is then correctly described as a 5x2x2 factorial experiment; from seven scheduling heuristics, twenty (5x2x2) treatment combinations are formed. All twenty combinations are investigated in this study.

The particular factorial experiment used is said to have a randomized complete block design. The blocking performed is achieved by varying the starting times of each of the thirty-four projects. These starting times are generated by drawing random variates which are uniformly distributed over a span of 180 days, the length of a schedule. For each block, a string of thirty-four random variates is drawn. The first random variate is the starting time of the first project, the second is the starting time of the second project, etc. One string of thirty-four random variates corresponds to one block of the experiment. Since each block of the experiment receives all twenty treatment combinations of the scheduling rules, the experiment is said to be a *complete* block design.

For each of the descriptive measures evaluated in this study, a four-way and a two-way analysis of variance is performed in order to measure interactions, main effects, and effects due to blocking or introducing different resource demands. Duncan's Multiple Range Test[2] is then used to rank the various means; the 5% level of significance is used for reporting significant differences.

---

[2] See David B. Duncan, *Multiple Range and Multiple F Tests*, Department of Statistics and Statistical Laboratory, Virginia Polytechnic Institute, Technical Report No. 6 (1953), p. 3 which is referenced in Acheson J. Duncan, *Quality Control and Industrial Statistics*, Revised Edition, Homewood, Illinois: Richard D. Irwin, Inc., 1959, pp. 598-604.

Often in an experiment interest lies in selecting the treatment which has the largest or the smallest (population) mean value. Bechhofer's Multiple Ranking Procedure[3] gives the experimenter a decision rule for selecting the largest or the smallest population, and an operating characteristic which tells him the probability of his making a correct choice if he follows the given decision rule. For example, if the primary interest lies in selecting from the five basic scheduling rules the one which has the smallest (population) mean value, Bechhofer's procedure provides a decision rule for selecting this treatment mean with a specified probability of making a correct decision. With a sample size of thirty observations per treatment, there is a 99% probability of correctly choosing the smallest treatment mean if in fact it has an expected value that is 0.716σ smaller than the next smallest mean (where σ is equal to the population standard deviation). Bechhofer's procedure is used for selecting the smallest treatment mean for each of the descriptive measures.

---

[3] Robert E. Bechhofer, "A Single-Sample Multiple Decision Procedure for Ranking Means of Normal Populations with Known Variances," *Annals of Mathematical Statistics*, Vol. 25, (1954), pp. 16-39.

6.3 Determination of Sample Size

An estimate of the sample size (number of blocks) necessary to assure significant results (given an associated α and β level) is needed in order to establish the number and cost of the computer runs required. In the review of the literature, it was noted that in Pascoe's study, heuristics as a whole (f) are not significant.[4] Thus it is imperative that an estimate be made of the number of blocks required in order to assess the practicality of the present study.

Mize's[5] data described in Chapter 2 more closely resembles the present study than do the studies of Knight and Pascoe; Mize's data therefore is used to provide information for sample size estimation. It might be worthwhile to know, for example, how large n (the number of blocks) should be so that with the power at 0.95 and $\alpha = 0.05$, we are able to reject the hypothesis that Mize's nine multiple attribute decision rules are equal when in fact one of them exceeds the others by σ, the population standard deviation. This criteria for determining n is reasonable in light of Mize's results; it further provides some estimate of the number of blocks required

---

[4] <u>Supra</u>, chap. 2, p. 24.

[5] <u>Supra</u>, chap. 2, p. 30.

in this study.

Examination of power curves for the Non-Central F-Distribution developed by Pearson and Hartley[6] reveals that a sample size of n = 28 is needed to insure the accuracy stated. A sample of thirty blocks was thus selected. This number of blocks is well within the limits on computer time available for this study.

6.4 Test Results

Two-way and four-way analysis of variance, Duncan's Multiple Range Test, and Bechhofer's Multiple Ranking Procedure for each of the four criterion functions are presented in this section. Bartlett's,[7] Cochran's,[8] and Hartley's Short-cut[9] tests for the homogeneity of variances are made on all of the summary measures. Using Bartlett's test the hypothesis that the variances associated with

---

[6] E. S. Pearson and H. O. Hartley, "Charts of the Power Function for Analysis of Variance Tests, Derived from the Non-Central F-Distribution," *Biometrica*, Vol. 38, (1951), pp. 112-130.

[7] Wilfrid J. Dixon and Frank J. Massey, Jr., *Introduction to Statistical Analysis*, New York: McGraw-Hill Book Company, Inc., 1969, pp. 308-310.

[8] Ibid, pp. 310.

[9] H. O. Hartley, "The Maximum F-Ratio as a Short-Cut Test for Heterogeneity of Variance," *Biometrica*, Vol. 37, (1950), pp. 308-312.

total slippage and computer processing time are homogeneous is rejected at the 5% level; the hypothesis for the measures weighted total slippage and total resource idle time are accepted. Using Cochran's and Hartley's tests, the conclusion is drawn that the variances associated with each summary measure are homogeneous at the 5% level of significance. The variances are therefore treated as being homogeneous. Computed values for each of the tests for homogeneity of variances are given in Table 14.

In the discussion that follows, the statistical results of ranking each of the scheduling rules are examined, and an attempt is made to state which scheduling rule is superior for each criterion function. For all criterion functions considered, the data herein reported is coded since it is not intended for public use. This coding does not affect the analysis or the interpretation of the results.

The results of the four-way analysis of variance show that for every criterion function, treatments A and C are highly significant. Treatment B is significant only for the computer processing time criterion. Thus, although randomization (treatment B) appears necessary for solving certain project scheduling problems, it contributes little in explaining the variation in schedules when a wide span

TABLE 14

TESTS FOR HOMOGENEITY OF VARIANCES

Criterion Function	Bartlett's Test Computed Value	Cochran's Test Computed Value	Hartley's Test Computed Value
Total Slippage	1.69*	0.089	3.82
Weighted Total Slippage	1.01	0.082	3.07
Total Resource Idle Time	0.98	0.092	3.78
Computer Processing Time	1.81*	0.091	3.65

*The variances are concluded to be homogeneous at the 5% significance level with two exceptions. The hypothesis of homogeneous variances is rejected at the 5% level using Bartlett's test for the measures of total slippage and computer processing time. The hypothesis of homogeneous variances is accepted using each test and for all of the summary measures at the 1% significance level.

of problems is considered.

1. Total Slippage

The total slippage of a project set is the sum over all projects of the difference between the assigned scheduled finish time and the late finish time determined by critical path methods. Total slippage is measured in days.

The average amount of slippage varies between 998 and 1502 days depending on the scheduling rule chosen. The application of the shortest imminent operation heuristic gives the least slippage; greatest remaining resource demand gives the highest. As can be seen from the four-way analysis of variance given in Table 15, randomness contributes little in explaining the variation present in total project slippage.

The presence of rescheduling generally increases the amount of slippage present in a schedule. Only for the rules greatest remaining resource demand and greatest total resource demand is slippage reduced by the presence of the reschedule rule. Both of these rules, however, represent the worst instances of project slippage.

If the population standard deviation is approximately equal to the square root of the sample error mean square given in Table 16, the standardized value of the difference

TABLE 15

ANALYSIS OF VARIANCE FOR TOTAL SLIPPAGE

Source	d.f.	S.S.	M.S.	F	$F_{\alpha=0.01}$
A	4	6,491,757	1,622,939	386.38	3.35
B	1	6,767	6,767	1.61	6.68
C	1	279,288	279,288	66.49	6.68
AB	4	10,318	2,580	0.61	3.35
AC	4	4,394,954	1,098,738	261.58	3.35
BC	1	5,287	5,287	1.26	6.68
ABC	4	6,455	1,614	0.38	3.35
Blocks	29	1,761,246	60,732	14.46	1.75
Error	551	2,314,408	4,200		
Total	559	15,270,481			

TABLE 16

ANALYSIS OF VARIANCE FOR TOTAL SLIPPAGE
TREATMENT A ONLY

Source	d.f.	S.S.	M.S.	F	$F_{\alpha=0.01}$
Treatments	4	5,126,712	1,281,678	294.62	3.48
Blocks	29	518,316	17,873	4.11	1.86
Error	116	504,627	4,350		
Total	149	6,149,655			

between the two lowest ranking population means associated with a 99% probability of a correct ranking of sample means (using Bechhofer's decision rule) is 47 days. This indicates that if the true (population) difference between the two lowest ranking treatment A means (shortest imminent operation and least total float) is at least 47 days, then 99% of the time the true ranking of the population means will result despite the random statistical fluctuations of sampling. While Bechhofer's test does not permit inferences based on the observed differences in sample means, it appears from the sample results that the shortest imminent operation rule is the lowest ranking scheduling rule for total slippage.

The results of applying Duncan's test to the sample data are shown in Table 17. The shortest imminent operation heuristic is the lowest ranking sample mean for total slippage; greatest remaining resource demand is the highest.

2. Weighted Total Slippage

The weighted total slippage of a project set is the sum over all projects of the total resources demanded by a project multiplied by the slippage of that project as defined above. Weighted slippage is measured in man-days.

TABLE 17

MULTIPLE RANGE TEST FOR TOTAL SLIPPAGE

Significance	Treatment	Mean (In Days)
	GRRD RAN	1,502
	GRRD	1,487
	GTRD RAN	1,351
	GTRD	1,342
	GRRD RAN RES	1,206
	GRRD RES	1,201
	GTRD RAN RES	1,161
	GRU RES	1,155
	GRU RAN RES	1.154
	LTF RAN RES	1,151
	SIO RAN RES	1,150
	SIO RES	1,138
	GTRD RES	1,132
	LTF RES	1,131
	GRU	1,121
	GRU RAN	1,110
	LTF	1,054
	LTF RAN	1,043
	SIO RAN	1,000
	SIO	998

A vertical line enclosing a group of means indicates that the means located within the group cannot be distinguished from one another at the 5% level of significance. Descriptions of each of these treatments are found on pages 55-59 and in Table 13.

The scheduling rule which produces the least weighted slippage is least total float. Duncan's Multiple Range Test is, however, unable to distinguish between this scheduling rule and greatest resource usage. Since least total float requires one-half of the computer processing time that greatest resource usage requires, least total float is the better rule to use in order to minimize the weighted slippage of project sets. Results of applying Duncan's procedure are shown in Table 20.

The four-way analysis of variance given in Table 18 shows treatment A (the basic scheduling heuristic), treatment C (the reschedule feature), and the A-C treatment interaction are significant beyond the 1% level in the analysis of variance. Treatment B (randomness), however, adds little in explaining the variation present in weighted total slippage.

If the population standard deviation is approximately equal to the square root of the error mean square given in Table 19, then the standardized difference in population means associated with Bechhofer's procedure (and a 99% probability of a correct ranking of sample means) is 32,000 man-days. Bechhofer's test does not permit inferences to be made on the observed differences in sample means between the two lowest ranking scheduling rules, least total float and greatest resource utilization. It appears from the sample

TABLE 18

ANALYSIS OF VARIANCE FOR WEIGHTED TOTAL SLIPPAGE

Source	d.f.	S.S. (100,000)	M.S. (100,000)	F	$F_{\alpha=0.01}$
A	4	192,859	48,215	32.95	3.35
B	1	1,699	1,699	1.16	6.68
C	1	19,963	19,963	13.64	6.68
AB	4	4,022	1,005	0.69	3.35
AC	4	130,052	32,513	22.22	3.35
BC	1	953	953	0.65	6.68
ABC	4	3,218	805	0.55	3.35
Blocks	29	417,217	14,387	9.83	1.75
Error	551	806,356	1,463		
Total	599	1,223,573			

TABLE 19

ANALYSIS OF VARIANCE FOR WEIGHTED TOTAL SLIPPAGE
TREATMENT A ONLY

Source	d.f.	S.S. (100,000)	M.S. (100,000)	F	$F_{\alpha=0.01}$
Treatments	4	180,315	45,079	22.50	3.48
Blocks	29	117,411	4,049	2.02	1.86
Error	116	232,388	2,003		
Total	149	530,114			

TABLE 20

MULTIPLE RANGE TEST FOR WEIGHTED TOTAL SLIPPAGE

Significance	Treatment	Mean (In Man-days)
	GRRD RAN RES	723,693
	GTRD	723,395
	GRRD RES	723,127
	SIO	721,692
	GRRD RAN	719,803
	GTRD RAN	714,011
	GRRD	713,060
	SIO RAN	710,927
	LTF RAN RES	703,471
	GRU RAN RES	703,235
	GTRD RAN RES	702,360
	SIO RAN RES	698,327
	GRU RES	697,270
	SIO RES	695,129
	GTRD RES	693,777
	LTF RES	692,352
	GRU RAN	660,190
	LTF RAN	655,869
	GRU	653,746
	LTF	644,648

A vertical line enclosing a group of means indicates that the means located within the group cannot be distinguished from one another at the 5% level of significance. Descriptions of each of these treatments are found on pages 55-59 and in Table 13.

results, however, that these two scheduling rules produce schedules with similar values for weighted total slippage.

3. Total Resource Idle Time

The total resource idle time is the amount of time that resources are idle during a schedule span. Idle time is measured in man-hours.

The scheduling abilities of the shortest imminent operation heuristic and greatest remaining resource demand are reversed when measuring idle time as opposed to project slippage; greatest remaining resource demand produces the least idle time schedule and the shortest imminent operation heuristic produces the greatest idle time schedule. Such a result would be expected. The logic of greatest remaining resource demand lies in the efficient utilization of resources, whereas the logic of the shortest imminent operation rule lies in accomplishing as many jobs as possible in as short a length of time as is possible. What would not be expected in these results is that the greatest remaining resource demand rule performs better than the greatest resource usage rule! The greatest resource usage rule is a local optimization rule, and it might be expected to perform better than a rule which doesn't give a guarantee of the most efficient utilization of resources within a short period. The results of ranking these scheduling rules are shown in Table 23.

The four-way analysis of variance for total resource idle time is shown in Table 21. As with the previous measures for assessing scheduling ability, treatments A and C and the A-C treatment interaction are highly significant in explaining the variation present in schedules, whereas treatment B is insignificant.

The square root of the sample error mean square given in Table 22 is equal to 1,828 man-hours. If this is approximately equal to the population standard deviation and if the greatest remaining resource demand rule has a (population) mean value that is 1,308 man-hours less than the next lowest ranking scheduling rule, then Bechhofer's procedure leads to the conclusion that there is a 99% chance of correctly selecting the lowest ranking treatment despite the random statistical fluctuations of sampling. Although Bechhofer's procedure does not permit these inferences, it appears from the sample results that the greatest remaining resource demand rule produces schedules with substantially less idle time than do the other scheduling rules considered. The difference in observed idle times between the two lowest ranking scheduling rules is felt to be of great economic significance.

TABLE 21

ANALYSIS OF VARIANCE FOR TOTAL RESOURCE IDLE TIME

Source	d.f.	S.S.	M.S.	F	$F_{\alpha=0.01}$
A	4	971,843,896	242,960,974	78.84	3.35
B	1	168,371	168,371	0.05	6.68
C	1	579,515,883	579,515,883	88.06	6.68
AB	4	5,948,609	1,487,152	0.48	3.35
AC	4	639,991,514	159,997,879	51.92	3.35
BC	1	12,595	12,595	0.00	6.68
ABC	4	8,764,936	2,191,234	0.71	3.35
Blocks	29	1,011,357,775	34,874,405	11.32	1.75
Error	551	1,697,964,539	3,081,605		
Total	599	4,915,568,097			

TABLE 22

ANALYSIS OF VARIANCE FOR TOTAL RESOURCE IDLE TIME
TREATMENT A ONLY

Source	d.f.	S.S.	M.S.	F	$F_{\alpha=0.01}$
Treatments	4	790,870,250	197,717,563	59.15	3.48
Blocks	29	274,677,502	9,471,638	2.83	1.86
Error	116	387,737,532	3,342,565		
Total	149	1,453,285,284			

TABLE 23

MULTIPLE RANGE TEST FOR TOTAL RESOURCE IDLE TIME

Significance	Treatment	Mean (In Man-hours)
	SIO	68,203
	SIO RAN	67,694
	GRU RAN RES	66,583
	SIO RAN RES	66,571
	LTF RES	66,557
	GTRD RES	66,502
	GRU RES	66,484
	GTRD RAN RES	66,419
	SIO RES	66,414
	LTF RAN RES	66,394
	GRRD RES	65,732
	LTF RAN	65,681
	GRRD RAN RES	65,599
	LTF	65,067
	GTRD	64,080
	GTRD RAN	63,987
	GRU RAN	63,595
	GRU	63,364
	GRRD	61,193
	GRRD RAN	60,736

A vertical line enclosing a group of means indicates that the means located within the group cannot be distinguished from one another at the 5% level of significance. Descriptions of each of these treatments are found on pages 55-59 and in Table 13.

4. Computer Processing Time

The Multi-Project Scheduling Program was run on the Indiana University CDC 3600 computer. The computer processing times herein reported refer to average time spent in generating a schedule, exclusive of all I/O time.

By far the most expensive scheduling rule in terms of time and core memory required is the greatest resource usage rule. Assuming that *non-prime* CDC 3600 time costs 10¢ a second, the greatest resource usage rule is $10 more expensive than the least cost rule, shortest imminent operation. The high cost of the greatest resource usage rule is due, of course, to the zero-one integer programming code incorporated in it.

The four-way analysis of variance given in Table 24 shows treatments A, B, and C to be significant in explaining the variation in computer processing time. Treatment B is significant for computer processing time because its presence always adds to the time required to generate a schedule.

Duncan's test shown in Table 26 is unable to distinguish a single superior scheduling rule. This is because the six lowest ranking rules differ by at the most, three seconds.

If the population standard deviation is approximately equal to the square root of the error mean square given in Table 25, and if the difference in the two lowest ranking

TABLE 24

ANALYSIS OF VARIANCE FOR COMPUTER PROCESSING TIME

Source	d.f.	S.S.	M.S.	F	$F_{\alpha=0.01}$
A	4	608,183	152,046	8225.61	3.35
B	1	648	648	35.08	6.68
C	1	54,037	54,037	2923.39	6.68
AB	4	394	98	5.33	3.35
AC	4	154	39	2.09	3.35
BC	1	30	30	1.65	6.68
ABC	4	0	0	0	3.35
Blocks	29	2,176	75	4.06	1.75
Error	551	10,185	18		
Total	599	675,808			

TABLE 25

ANALYSIS OF VARIANCE FOR COMPUTER PROCESSING TIME
TREATMENT A ONLY

Source	d.f.	S.S.	M.S.	F	$F_{\alpha=0.01}$
Treatments	4	146,386	36,596	1396.65	3.48
Blocks	29	811	28	1.07	1.86
Error	116	3,040	26		
Total	149	150,237			

TABLE 26

MULTIPLE RANGE TEST FOR COMPUTER PROCESSING TIME

Significance	Treatment	Mean (In Seconds)
	GRU RAN RES	126
	GRU RES	120
	GRU RAN	107
	GRU	102
	LTF RAN RES	66
	LTF RES	65
	LTF RAN	48
	LTF	47
	GRRD RAN RES	44
	GRRD RES	42
	GTRD RAN RES	42
	GTRD RES	40
	SIO RAN RES	40
	SIO RES	38
	GTRD RAN	23
	GRRD RAN	22
	GTRD	22
	GRRD	22
	SIO RAN	21
	SIO	20

A vertical line enclosing a group of means indicates that the means located within the group cannot be distinguished from one another at the 5% level of significance. Descriptions of each of these treatments are found on pages 55-59 and in Table 13.

(population) treatment means is 3.6 seconds, then Bechhofer's procedure assures with 99% probability that the lowest ranking population mean can be correctly chosen despite the random statistical fluctuations of sampling. This difference in mean values is of little economic significance. The difference in cost between the two lowest ranking scheduling rules is only about 20¢. Computer processing time is therefore concluded to be of little economic significance in developing schedules for the types of problems considered in this study.

CHAPTER 7

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS FOR FURTHER RESEARCH

7.1 Summary

The allocation of resources to multiple projects which simultaneously compete for limited resources is a problem characteristic of manufacturing firms, contracting firms, and research and development organizations. In allocating resources, it is often found that activities demand more resources than are available. When this happens, a question arises regarding which jobs to delay and which jobs to schedule. Mathematical programming techniques and heuristic solution procedures have been employed to solve this resource allocation problem. The object of this study is to statistically assess many of the heuristic solution procedures available for scheduling multiple projects.

The approach taken in this assessment is to obtain data representing an actual multi-project scheduling problem. This data is supplied by a research and development organization, and it consists of thirty-four projects to be scheduled over an eight month interval. It is felt that the use of this data provides a more realistic challenge to the heuristic solution procedures evaluated than the alternative of generating fictitious projects on a computer.

The thirty-four projects are scheduled simultaneously using four variations of five basic scheduling heuristics. The basic rules employed are (1) the shortest imminent operation rule, (2) a rule which considers the slack present in an activity, (3) a rule which considers the total resources demanded by an activity, (4) a rule which attempts to schedule around potential bottlenecks, and (5) a rule which schedules man power efficiently over a small time interval. The variations of these rules are a provision for rescheduling activities based on resource conflicts exhibited in the first schedule and a provision for switching activities at random in the order in which they are considered for resource assignment when they tie on one of the five basic scheduling rules.

Each of the scheduling rules is evaluated on its ability to generate schedules with low total project delays and low weighted project delays where the weights are determined by the size of the project. In addition, the rules are evaluated on their ability to generate schedules which utilize manpower efficiently. Employing analysis of variance tests, multiple ranking procedures, and multiple comparison methods, the rules are described as being superior or inferior to other rules in satisfying the above measures of organizational performance.

The shortest imminent operation heuristic schedules activities with the least total project delay while concurrently scheduling man power with the greatest amount of idle time.

The scheduling rule greatest remaining resource demand (which schedules first those activities on projects with the greatest remaining work to be completed) produces schedules that result in the least amount of idle time. These schedules, however, represent the worst possible cases of project slippage.

The least total float rule produces schedules with the least weighted slippage, the weight being determined by the man-days of resources required to complete the project.

Attempts to improve scheduling rules by adding an element of *chance* in the scheduling of activities or by rescheduling activities based on resource conflicts exhibited in the first schedule are generally unsuccessful. In isolated cases, though, the presence of randomness and rescheduling does make an improvement in a schedule.

Finally, the difference in computer processing times required for implementing the various scheduling rules is small for the type of problems considered. The greatest resource usage rule employing a zero-one integer programming algorithm is the only rule which requires a significantly larger amount of computer time for developing a schedule.

7.2 Conclusions and Comparisons with Other Studies

The results of this study demonstrate that for the type of problems considered, scheduling rules which base their logic on moving jobs through the system in as short a length of time as possible generally produce low slippage schedules at the expense of inefficient utilization of resources. Those scheduling rules which concentrate on using resources efficiently, do so at the expense of rather large delays in project completion.

These conclusions are just the opposite of those reached by Mize.[1] For the class of problems he considers, a rule which produces a schedule with low project slippage simultaneously schedules activities so that facility idle time is low. The difference in findings demonstrates that before any scheduling algorithm which employs heuristic techniques is adopted by a firm, a determination should be made concerning the ability of the algorithm to satisfy the specific project scheduling objectives of the firm. It may be that one set of heuristic scheduling rules is unable to satisfy these objectives when different classes or groups of projects are considered.

---

[1] *Supra*, chap. 2, p. 29.

The results of this study also demonstrate that when an actual scheduling problem is examined, the heuristic methods of determining schedules do account for a significant portion of the variation present in project scheduling objectives. These results do not agree with those of Pascoe.[2] This disparity between findings may be a result of the smaller amount of variation present in the projects examined in this study. The disparity may also be accounted for by the difference in scheduling rules examined.

No significant difference in performance exists between scheduling programs that possess an element of randomness and those which do not. Thus, while a certain amount of randomness seems necessary for solving some project scheduling problems optimally, randomness generally adds little in deriving schedules that satisfy various project scheduling objectives. Because the cost of having randomness in a scheduling program is low, however, a firm may run several programs (each with a different set of random numbers) and then pick the schedule that comes closest to meeting its objectives.

While rescheduling based on resource conflicts exhibited in the first schedule is significant in the analysis

---

[2] *Supra*, chap. 2, p. 24.

of variance, scheduling rules not using the reschedule feature out-perform those that do use it for every scheduling criteria considered. These reschedule results do not agree with the reschedule results reported by Knight.[3] Because of his linearization technique which eliminates the slack present in an activity and because of the difference in reschedule features used in the two studies, the two sets of results are not strictly comparable.

One of the benefits claimed for heuristic procedures is their ability to generate solutions rapidly and therefore inexpensively. The results of this study demonstrate that it is economically practical to employ several heuristic scheduling rules and to then choose a schedule which comes closest to meeting the specific objectives of the firm. The computer program developed for this study is able to schedule thirty-four projects over a one year span for a work force of approximately one hundred men. The program requires between twenty and forty seconds of CDC 3600 computer time to develop such a schedule. (The time quoted depends upon the scheduling rule chosen, and does not include the greatest resource utilization scheduling heuristic). With *prime* CDC 3600 time costing 20¢ per second,

---

[3] Supra, chap. 2, p. 33.

a one year schedule can be developed for about $6.00. At *non-prime* time, the same schedule costs about $3.00.

These low cost schedules are beneficial to management because of their ability to assess the effects of accepting more or less direct project work and the effects of varying the work force to any great extent. By using the programs and procedures developed in this study, a large number of input parameters can be varied and the effects assessed. The resultant cost is very low considering the potential benefits derived.

The scheduling programs developed also have great potential as a portion of a management information system. It is an easy matter to judge the effects of taking on an additional project or changing the project mix with the Multi-Project Scheduling Program. When using MPSP, it should be possible to set more accurate completion dates for projects than if such a system is not employed. It is a simple matter to judge the effects of a different project mix on the types and quantities of manpower required by the firm. This ability to assess alternatives inexpensively is of great benefit to a firm deciding whether to accept a job (project) or to change the number or the mix of its personnel.

7.3 Recommendations for Further Research

Several avenues of research have opened as a result of this study. This is especially true with regard to efforts to make the results operational in the firm described. The following suggestions, however, are in no way limited solely to the firm considered in this study.

First, other scheduling rules than those treated in this study could be examined. The ones examined are a sample of those available, and it is indeed possible that other existing rules can out-perform those examined here. Many such rules are suggested by Pascoe[4] and Mize[5] and are not examined in this study. It would also be advantageous to assess the abilities of many of the mathematical programming algorithms available for scheduling moderate size projects.

Second, the present study can be expanded by fitting the project characteristics outlined in Chapter 3 to empirical distributions in order to generate project sets similar to the one described. Numerous Monte Carlo techniques are available for doing this. The examination of project characteristics and subsequent generation of project sets makes it possible to obtain data at a relatively low cost

---

[4] Supra, chap. 2, Table 3, p. 25.

[5] Supra, appendix A, p. 38 .

when compared to the cost of obtaining original data. The generation and analysis of project sets also broadens the inference space over which conclusions regarding these results may be made.

A third area for research lies in the determination of the distributions and moments of project characteristics such as those described by Pascoe. This would involve contacting many firms and/or individuals willing to supply project data. Again, once this data is gathered, the inference space over which conclusions can be made includes a much larger class of problems.

The above two areas for further research suggest a fourth area that would involve determining how characteristics of project sets can be used to determine which scheduling rule is most likely to develop a superior schedule. Research similar to this is being performed in the integer linear programming area where attempts are being made to determine which IP algorithms are most likely to solve certain types of integer programming problems. The output of this study would consist of a group of statistical routines that would examine a particular project set and determine which heuristic rules would be superior for scheduling the project set. The statistical routines would then *call* a scheduling routine consisting of the rules determined to be superior for scheduling the projects under consideration.

One of the assumptions of the present research suggests a fifth and final possibility for future research. This assumption is that resources are used by an activity at a constant rate. Within the context of the resource constrained scheduling problem, there needs to be made a decision on how this assumption affects the scheduling procedures developed. Activities can often be accelerated by the application of additional resources. Whether the ranking of the scheduling procedures developed in this study would remain the same with the assumption of a varying resource usage rate needs to be determined.

APPENDIX B

EST Early Start Time

The earliest that an activity can begin, given the constraints imposed by the precedence relations and the constraints imposed on *the beginning of the entire* project, is denoted by EST. It is computed as follows:

$$EST(a_j) = \max_{\text{all } a_{j-1}} \left\{ \begin{matrix} 0 \\ EST(a_{j-1}) + d_{j-1} \end{matrix} \;\middle|\; a_{j-1} << a_j \right\}$$

where << denotes "immediately precedes"

EFT Early Finish Time

The early finish time of an activity is the earliest that an activity can finish, given the precedence relations of the network and the constraints imposed upon the beginning of the entire project. It is computed as follows:

$$EFT(a_j) = EST(a_j) + d_j$$

LFT Late Finish Time

The latest that an activity can finish without delaying the completion time of the entire project is denoted by LFT. It is computed as follows:

$$LFT(a_j) = \min_{\text{all } a_{j+1}} \left\{ \begin{matrix} \max LFT \\ LFT(a_{j+1}) - d_{j+1} \end{matrix} \;\middle|\; a_{j+1} >> a_j \right\}$$

where >> denotes "immediately succeeds"

LST **Late Start Time**

The late start time of an activity is the latest that an activity can start without delaying the completion of the entire project. It is denoted by LST and is computed as follows:

$$LST(a_j) = LFT(a_j) - d_j$$

TF **Total Float**

The total float of an activity is the amount of time that an activity can be delayed without delaying the overall project. It is determined using one of two different formulas:

$$TF(a_j) = LST(a_j) - EST(a_j)$$

or

$$TF(a_j) = LFT(a_j) - EFT(a_j)$$

Total float is often referred to as total slack or slack in the literature.

FF **Free Float**

The free float of an activity is an indication of the extent to which the activity can be delayed without delaying any subsequent activity in the project. It is computed as follows:

$$FF(a_j) = \left[EST(a_{j+1}) - EFT(a_j) \middle| \; a_j << a_{j+1} \right]$$

CP

Critical Path

The critical path of a network is defined to be that chain of activities from origin to terminus which has the largest sum of durations comprising its length. A project can have more than one critical path. All activities with TF = 0 are termed critical activities and lie on a critical path of a project.

BIBLIOGRAPHY

BIBLIOGRAPHY

Books

Anderson, R. L., and Bancroft, T. A., *Statistical Theory in Research,* New York: McGraw-Hill Book Company, Inc., 1952.

Archibald, R. D., and Villoria, R. L., *Network Based Management Systems,* New York: John Wiley and Sons, Inc., 1967.

Battersby, Albert, *Network Analysis for Planning and Scheduling,* London: Macmillan & Co., Ltd., 1965.

Battersby, Albert, *Network Analysis for Planning and Scheduling,* Second Edition, New York: St. Martin's Press, 1967.

Berge, C., *The Theory of Graphs and Its Applications,* New York: John Wiley and Sons, Inc., 1962.

Berge, Claude, and Ghouila-Houri, A., *Programming, Games and Transportation Networks,* New York: John Wiley and Sons, Inc., 1965.

Bowker, A. H., and Lieberman, G. J., *Engineering Statistics,* Englewood Cliffs, New Jersey: Prentice-Hall Inc., 1959.

Brownlee, K. A., *Statistical Theory and Methodology in Science and Engineering,* New York: John Wiley and Sons, Inc., 1960.

Buffa, Elwood S., *Production-Inventory Systems: Planning and Control,* Homewood, Illinois: Richard D. Irwin, Inc., 1968.

Churchman, C. West, Ackoff, Russel L., and Arnoff, E. Leonard, *Introduction to Operations Research,* New York: John Wiley and Sons, Inc., 1957.

Conway, R. W., Maxwell, W. L., and Miller, L. W., *Theory of Scheduling,* Reading, Massachusetts: Addison-Wesley Publishing Co., Inc., 1967.

Dantzig, G. B., *Linear Programming and Extensions,* Princeton, New Jersey: Princeton University Press, 1963.

Davies, Owen L. (ed.), *The Design and Analysis of Industrial Experiments,* London: Oliver and Boyd, 1960.

Dixon, Wilfrid, J., and Massey, Frank J., Jr., *Introduction To Statistical Analysis*, New York: McGraw-Hill Book Company, Inc., 1969.

Duncan, Acheson J., *Quality Control and Industrial Statistics*, Revised Edition, Homewood, Illinois: Richard D. Irwin, Inc., 1959.

Elmaghraby, S. E., *The Design of Production Systems*, New York: Reinhold Publishing Company, 1966.

Feller, William, *An Introduction to Probability Theory and Its Applications*, Second Edition, New York: John Wiley and Sons, Inc., 1957.

Ford, L. R., and Fulkerson, D. R., *Flows in Networks*, Princeton, New Jersey: Princeton University Press, 1962.

Graybill, Franklin A., *An Introduction to Linear Statistical Models*, New York: McGraw-Hill Book Co., Inc., 1961, Volume 1.

Guenther, William C., *Analysis of Variance*, Englewood Cliffs, New Jersey: Prentice-Hall Inc., 1964.

Hadley, George, "Project Planning and Manpower Scheduling," Sec. 8.8 of *Nonlinear and Dynamic Programming*, Reading, Massachusetts: Addison-Wesley, 1964.

Hahn, Gerald J., and Shapiro, Samuel S., *Statistical Methods In Engineering*, New York: John Wiley and Sons, Inc., 1967.

Hicks, Charles R., *Fundamental Concepts in the Design of Experiments*, New York: Holt, Rinehart, and Winston, 1964.

Hoel, Paul G., *Introduction to Mathematical Statistics*, Third Edition, New York: John Wiley and Sons, Inc., 1962.

Horowitz, Joseph, *Critical Path Scheduling*, New York: Ronald Press Company, 1967.

Kelley, J. E., Jr., *Critical Path Planning and Scheduling: An Introduction*, Ambler, Pennsylvania: Mauchly Associates, Inc., 1959.

Kempthorne, Oscar, *The Design and Analysis of Experiments*, New York: John Wiley and Sons, Inc., 1952.

Levin, R. I., and Kirkpatrick, C. A., *Planning and Control with PERT/CPM*, New York: McGraw-Hill Book Company, 1966.

Lockyer, K. G., *An Introduction to Critical Path Analysis*, London: Pitman Publishing Corporation, 1964.

Lockyer, K. G., *An Introduction to Critical Path Analysis*, Second Edition, London: Issac Pitman and Sons, Ltd., 1967.

Martino, R. L., *Project Management and Control: Vol. III, Allocating and Scheduling Resources*, New York: American Management Association, 1965.

Moder, J. J., and Phillips, C. R., *Project Management with CPM and PERT*, New York: Reinhold Publishing Company, 1964.

Moore, Franklin G., *Production Control*, Revised Edition, New York: McGraw-Hill Book Company, Inc., 1959.

Muth, John F., and Thompson, Gerald L. (eds.), *Industrial Scheduling*, New Jersey: Prentice-Hall, Inc., 1963.

Naylor, Thomas H., et al., *Computer Simulation Techniques*, New York: John Wiley and Sons, Inc., 1966.

Norden, P. V., "Resource Usage and Network Planning," in *Operations Research in Research and Development*, B. V. Dean, (ed.), New York: John Wiley and Sons, Inc., 1963.

Peng, K. C., *The Design and Analysis of Scientific Experiments*, Reading, Massachusetts: Addison-Wesley Publishing Co., Inc., 1967.

PERT Summary Report, Phase I, July 1958, Special Projects Office, Bureau of Naval Weapons, Department of the Navy, Washington, D. C.

Raiffa, Howard, and Schlaifer, Robert, *Applied Statistical Decision Theory*, Boston: Harvard University, 1961.

Riggs, James L., and Heath, Charles O., *Guide to Cost Reduction Through Critical Path Scheduling,* Englewood Cliffs, New Jersey: Prentice-Hall Inc., 1963, 1966.

Scheffe, Henry, *The Analysis of Variance,* New York: John Wiley and Sons, Inc., 1959.

Siegel, Sidney, *Nonparametric Statistics for the Behavioral Sciences,* New York: McGraw-Hill Book Company, Inc., 1956.

Simon, H. A., *The New Science of Management Decision,* New York: Harper and Row, 1960.

Shaffer, L. R., Ritter, J. B., and Meyer, W. L., *The Critical Path Method,* New York: McGraw-Hill Book Company, Inc., 1965.

Sisson, R. L., "Sequencing Theory," in *Progress in Operations Research,* Vol. I, R. L. Ackoff (ed.), New York: John Wiley and Sons, Inc., 1961.

Snedecor, George W., *Statistical Methods,* Ames, Iowa: Iowa State University Press, 1956.

Vajda, S., *Mathematical Programming,* Reading, Massachusetts: Addison-Wesley Publishing Co., Inc., 1967.

Voris, L. P., *Production Control: Text and Cases,* Homewood, Illinois: Richard D. Irwin, Inc., 1961.

Winer, B. J., *Statistical Principles in Experimental Design,* New York: McGraw-Hill Book Company, Inc., 1962.

Articles and Technical Reports

Baker, C. T., and Dzielinski, B. P., "Simulation of a Simplified Job Shop," *Management Science,* Vol. 6, No. 3, (April 1960), pp. 311-323.

Baker, J. J., and Shaffer, L. R., "Staged Decision Theory Applied to the Limited Resource Problem," *Civil Engineering Studies, Report No. 8,* Urbana, Illinois: Department of Civil Engineering, University of Illinois, September, 1965.

Balas, E., "An Additive Algorithm for Solving Linear Programs with Zero-One Variables," *Operations Research,* Vol. 13, No. 3, (July-August 1965), pp. 517-546.

Balinski, M. L., "Integer Programming: Methods, Uses, Computation," *Management Science,* Vol. 12, No. 3, (November 1965), pp. 253-313.

Bechhofer, Robert E., "A Single-Sample Multiple Decision Procedure for Ranking Means of Normal Populations with Known Variances," *Annals of Mathematical Statistics,* Vol. 25, (1954), pp. 16-39.

Beckwith, R. E., "A Cost Control Extension of the PERT System," *IRE Transactions on Engineering Management,* Vol. EM-9, No. 4, (December 1962), pp. 147-149.

Bellman, R., and Kalaba, R., "On the kth Best Policies," *Journal of the Society of Industrial and Applied Mathematics,* Vol. 8, No. 4, (December 1960), pp. 582-588.

Berman, E. B., "Resource Allocations in a PERT Network Under Continuous Time-Cost Functions," *Management Science,* Volume 10, No. 4, (July 1964), pp. 734-746.

Berman, H., "The Critical Path Method for Project Planning and Control," *The Constructor,* (September 1961), pp. 24-29.

Bigelow, Clifford George, "Bibliography on Project Planning and Control by Network Analysis: 1959-1961," *Operations Research,* Vol. 10, No. 5, (September-October 1962), pp. 728-731.

Bildson, R. A., and Gillespie, J. R., "Critical Path Planning--PERT Integration," *Operations Research,* Vol. 10, No. 6, (November-December 1962), pp. 909-912.

Blair, Robert R., "Critical Path Resources, Simulation and Scheduling," *IEEE Transactions on Engineering Management,* Vol. EM-10, No. 3, (September 1963), pp. 100-104.

Blanning, R. W., and Rao, A. G., "A Note On Decomposition of Project Networks," *Management Science,* Vol. 12, No. 1, (September 1965), pp. 145-148.

Bowman, E. H., "Assembly Line Balancing by Linear Programming," *Operations Research*, Vol. 8, No. 3, (May-June 1960), pp. 385-389.

Brand, J. D., Meyer, W. L., and Schaffer, L. R., "The Resource Scheduling Problem in Construction," *Civil Engineering Studies, Report No. 5*, Urbana, Illinois: Department of Civil Engineering, University of Illinois, 1964.

Burgess, A. R., and Killebrew, J. B., "Variation in Activity Level on a Cyclic Arrow Diagram," *Journal of Industrial Engineering*, Vol. 13, No. 2, (March-April 1962), pp.76-83.

Carruthers, J. A., and Battersby, Albert, "Advances in Critical Path Methods," *Operational Research Quarterly*, Vol. 17, No. 4, (December 1966), pp. 359-380.

Charnes, A., and Cooper, W. W., "A Network Interpretation and a Directed Subdual Algorithm for Critical Path Scheduling," *Journal of Industrial Engineering*, Vol. 13, No. 4, (July-August 1962), pp. 213-218.

Charnes, A., Cooper, W. W., and Thompson, G. L., "Critical Path Analysis Via Chance Constrained and Stochastic Programming," *Operations Research*, Vol. 12, No. 3, (May-June 1964), pp. 460-471.

Clark, C. E., "The Optimum Allocation of Resources Among Activities of a Network," *Journal of Industrial Engineering*, Vol. 12, No. 1, (January-February 1961) pp. 11-17.

Clarke, Roderick W., "Activity Costing - Key to Progress in Critical Path Analysis," *IRE Transactions on Engineering Management*, Vol. EM-9, No. 3, (September 1962), pp. 132-136.

Cochran, W. G., "Some Consequences When the Assumptions for the Analysis of Variance Are Not Satisfied," *Biometrics*, Vol. 3, (March 1947), pp. 22-38.

Conway, R. W., Johnson, B. M., and Maxwell, W. D., "An Experimental Investigation of Priority Dispatching," *Journal of Industrial Engineering*, Vol. 11, No. 3, (May 1960), pp. 221-230

Davis, E. W., "Resource Allocation in Project Network Models--A Survey," *Journal of Industrial Engineering,* Vol. 17, No. 4, (April 1966), pp. 177-188.

Demski, Joel S., "Some Considerations in Sensitizing an Optimization Model," *Journal of Industrial Engineering,* Vol. 19, No. 9, (September 1968), pp. 463-467.

Dewitte, L., "Manpower Levelling of PERT Networks," *Data Processing for Science/Engineering,* (March-April 1964).

Dike, S. H., "Project Scheduling with Resource Constraints," *IEEE Transactions on Engineering Management,* Vol. EM-11, No. 4, (December 1964), pp. 155-158.

DOD and NASA Guide, PERT/COST, Office of the Secretary of Defense and National Aeronautics and Space Administration, Washington, D. C., Govt. Printing Office, (June 1962).

Dooley, A. R., "Interpretations of PERT," *Harvard Business Review,* Vol. 42, No. 2, (March-April 1964), pp. 160-168.

Dudek, R. A., and Teuton, O. F., Jr., "Development of M-stage Decision Rule for Scheduling n-Jobs through m-Machines," *Operations Research,* Vol. 12, No. 3, (May-June 1964), pp. 471-498.

Eisner, Howard, "A Generalized Network Approach to the Planning and Scheduling of a Research Project," *Operations Research,* Vol. 10, No. 1, (January-February 1962), pp. 115-125.

Emery, J. C., "An Approach to Job Shop Scheduling Using a Large-Scale Digital Computer," *Industrial Management Review,* Vol. 3, No. 1, (Fall, 1961), pp. 78-96.

Fishman, George S., and Kiviat, Phillip J., "The Analysis of Simulation-Generated Time Series," *Management Science,* Vol. 13, No. 3, (March 1967), pp. 525-557.

Ford, L. R., and Fulkerson, D. R., "A Simple Algorithm for Finding Maximal Network Flows and an Application to the Hitchcock Problems," *Canadian Journal of Mathematics,* Vol. 9, No. 2, (1957), pp. 210-219.

Freeman, Raoul J., "A Generalized PERT," *Operations Research*, Vol. 8, No. 2, (March-April 1960), p. 281.

Fry, B. L., *Network-Type Management Control Systems Bibliography*, RAND Corporation Memo RM-3074-PR, (February 1963).

Fry, B. L., "Selected References on PERT and Related Techniques," *IEEE Transactions on Engineering Management*, Vol. EM-10, No. 3, (September 1963), pp. 150-154.

Fulkerson, D. R., "A Network Flow Computation for Project Cost Curves," *Management Science*, Vol. 7, No. 2, (January 1961), pp. 167-178.

Fulkerson, D. R., "Expected Critical Path Lengths in PERT Networks," *Operations Research*, Vol. 10, No.6, (November-December 1962), pp. 808-817.

Gere, W., "Heuristics in Job-Shop Scheduling," *Management Science*, Vol. 13, No. 3, (November 1966), pp. 167-190.

Geoffrion, A. M., *Implicit Enumeration Using an Imbedded Linear Program*, RAND Corporation Memo RM-5406-PR, (September 1967).

Geoffrion, A. M., *An Improved Implicit Enumeration Approach For Integer Programming*, RAND Corporation Memo RM-5644-PR, (June 1968).

Geoffrion, A. M., and Nelson, A. B., *Users Instructions for 0-1 Integer Linear Programming Code RIP30C*, RAND Corporation Memo RM-5627-PR, (May 1968).

Giffler, Bernard, "Simulation Models in Production Scheduling and Inventory Control," *Production and Inventory Management*, Vol. 7, No. 3, (July 1966), pp. 1-14.

Giffler, B., and Thompson, G. L., "Algorithms for Solving Production Scheduling Problems," *Operations Research*, Vol. 8, No. 4, (July-August 1960), pp. 478-504.

Glover, F., "A Multiphase-Dual Algorithm for the Zero-One Integer Programming Problem," *Operations Research*, Vol. 13, No. 6, (November-December 1965), pp. 879-919.

Gomory, Ralph E., "Outline of an Algorithm for Integer Solutions to Linear Programs," *Bulletin of the American Mathematical Society*, Vol. 64, No. 5, (September 1958), pp. 275-278.

Gorham, William, "Application of Network Flow Model to Personnel Planning," *IEEE Transactions on Engineering Management*, Vol. EM-10, No. 3, (September 1963), pp. 121-124. (Also published as RAND Corporation Memo RM 2587, June 24, 1960).

Grubbs, Frank E., "Attempts to Validate Certain PERT Statistics or 'Picking on PERT'," *Operations Research*, Vol. 10, No. 6, (November-December 1962), pp. 912-915.

Gutjahr, A. L., and Nemhauser, G. L., "An Algorithm for the Line Balancing Problem," *Management Science*, Vol. 2, No. 2, (November 1964), pp. 308-315.

Hartley, H. O., "The Maximum F-Ratio as a Short-cut Test for Heterogeneity of Variance," *Biometrica*, Vol. 37, (1950), pp. 308-312.

Hartley, H. O., and Wortham, A. W., "A Statistical Theory for PERT Critical Path Analysis," *Management Science*, Vol. 12, No. 10, (June 1966), pp. B469-B481.

Hegelson, W. B., and Birnie, D. P., "Assembly Line Balancing Using the Ranked Positional Weight Technique," *Journal of Industrial Engineering*, Vol. 13, No. 6, (November-December 1961), pp. 394-398.

Held, M., and Karp, R. M., "A Dynamic Programming Approach to Sequencing Problems," *Journal of the Society of Industrial and Applied Mathematics*, Vol. 10, No. 1, (March 1962), pp. 196-210.

Held, M., Karp, R. M., and Shareshian, W. S., "Assembly Line Balancing: Dynamic Programming with Precedence Constraints," *Operations Research*, Vol. 10, No. 3, (May-June 1963), pp. 442-460.

Heller, J., and Logemann, G., "An Algorithm for the Construction and Evaluation of Feasible Schedules," *Management Science*, Vol. 8, No. 2, (January 1962), pp. 168-183.

Hoffman, A. J., and Kuhn, H. W., "On Systems of Distinct Representatives, in Linear Inequalities and Related Systems," *Annals of Mathematics Study*, Princeton, New Jersey: Princeton University Press, Vol. 38, (1956), pp. 199-206.

Ignall, E. J., and Schrage, L., "Application of Branch and Bound Techniques to Some Flow-Shop Scheduling Problems," *Operations Research*, Vol. 13, No. 3, (May-June 1965), pp. 400-413.

Ignall, E. J., "A Review of Assembly Line Balancing," *Journal of Industrial Engineering*, Vol. 16, No. 4, (July-August 1965), pp. 244-254.

International Business Machines, *General Information Manual: Improved Job Shop Management Through Data Processing*, 1960.

Jackson, J. R., "A Computing Procedure for a Line Balancing Problem," *Management Science*, Vol. 2, No. 3, (April 1956), pp. 261-272.

Jewell, W. S., "Divisible Activities in Critical Path Analysis," Report No. ORC-64-3 (RR), Operations Research Center, University of California, Berkeley, (February 1964).

Jewell, W. S., "Risk-taking in Critical Path Analysis," *Management Science*, Vol. 11, No. 3, (January 1965), pp. 438-443.

Johnson, S., "Optimal Two- and Three-Stage Production Schedules with Set-Up Times Included," *Naval Research Logistics Quarterly*, Vol. 1, No. 1, (March 1954), pp. 61-68.

Kelley, James E., Jr., "Critical-Path Planning and Scheduling: Mathematical Basis," *Operations Research*, Vol. 9, No. 3, (May-June 1961), pp. 296-320.

Kelley, J. E., and Walker, M. R., "Critical-path Planning and Scheduling," *Proceedings of Eastern Joint Computer Conference*, (December 1959), pp. 160-173.

Kilbridge, M. D., and Webster, L., "A Review of Analytical Systems of Line Balancing," *Operations Research,* Vol. 10, No. 5, (September-October 1962), pp. 626-638.

Klingel, A. R., "Bias in PERT Project Completion Time Calculations for a Real Network," *Management Science,* Vol. 13, No. 4, (December 1966), pp. B194-B201.

Kuhn, H. W., "The Hungarian Method for Assignment Problem," *Naval Research Logistics Quarterly,* Vol. 2, No. 1 and 2, (March-June 1955), pp. 83-97.

Lambourn, S., "RAMPS--A New Tool in Planning and Control," *The Computer Journal,* Vol. 5, (1963), pp. 300-304.

Lawler, E. L., "On Scheduling Problems with Deferral Costs," *Management Science,* Vol. 11, No. 2, (November 1964) 280-288.

Leahy, John P., "A Man-Machine CPM System for Decision Making in the Construction Industry," *Construction Research Series No. 9,* Urbana, Illinois: Department of Civil Engineering, University of Illinois, 1967.

Levy, F. K., and Weist, J. D., "A Simulation Approach to Determining the Stochastic Characteristics of a Project," Graduate School of Industrial Administration, Carnegie Institute of Technology, 1963.

Levy, F. K., Thompson, G. L., and Wiest, J. D., "Multi-Ship, Multi-Shop Workload - Smoothing Program," *Naval Research Logistics Quarterly,* Vol. 9, No. 1, (March 1962), pp. 37-45.

Lomnicki, Z. A., "A Branch and Bound Algorithm for the Exact Solution of the Three-Machine Scheduling Problem," *Operational Research Quarterly,* Vol. 16, No. 1, (March 1965), pp. 89-100.

MacCrimmon, Kenneth R., and Ryavec, Charles A., "An Analytical Study of the PERT Assumptions," *Operations Research,* Vol. 12, No. 1, (January-February 1964), pp. 16-37.

Malcolm, D., Roseboom, J., Clark, C., and Fazar, W., "Application of a Technique for Research and Development Program Evaluation," *Operations Research*, Vol. 7, No. 5, (September-October 1959), pp. 646-669.

Manne, A. S., "On the Job Shop Scheduling Problem," *Operations Research*, Vol. 8, No. 2, (March-April 1960), pp. 219-223.

Mansoor, E. M., "Assembly Line Balancing--An Improvement on the Ranked Positional Weight Technique," *Journal of Industrial Engineering*, Vol. 15, No. 2, (March-April 1964), p. 73.

McGee, A. A., and Markarian, M. D., "Optimum Allocation of Research/Engineering Manpower within a Multi-Project Organizational Structure," *IRE Transactions on Engineering Management*, Vol. EM-9, No. 3, (September 1962), pp. 104-108.

McNaughton, Robert, "Scheduling with Deadlines and Loss Functions," *Management Science*, Vol. 6, No. 1, (October 1959), pp. 1-12.

Meyer, W. L., and Shaffer, L. R., "Extensions of the Critical Path Method Through the Application of Integer Programming," *Construction Research Series No. 2*, Urbana, Illinois: Department of Civil Engineering, University of Illinois, (July 1963).

Miller, Robert W., "How to Plan and Control with PERT," *Harvard Business Review*, Vol. 40, No. 2, (March-April 1962), pp. 93-104.

Minsky, M., "Steps Toward Artificial Intelligence," *Proceedings of the IRE*, Vol. 49, No. 1, (January 1961).

Mitten, L. G., "A Scheduling Problem," *Journal of Industrial Engineering*, Vol. 10, No. 2, (March-April 1959), pp. 30-36.

Moder, J. J., "How to Do CPM Scheduling without a Computer," *Engineering News-Record*, (March 1963), pp. 30-36.

Moodie, C. L., and Mandeville, D. E., "Project Resource Balancing by Assembly Line Balancing Techniques," *Journal of Industrial Engineering*, Vol. 17, No. 7, (July 1966), pp. 377-383.

Moshman, J., Johnson, J., and Larsen, M., "RAMPS--A Technique for Resource Allocation and Multiproject Scheduling," *Proceedings--1963 Spring Joint Computer Conference*, pp. 17-27.

Muth, John F., "The Effect of Uncertainty in Job Time on Optimal Schedules," *O. N. R. Research Memorandum No. 88*, Graduate School of Industrial Administration, Carnegie Institute of Technology, (January 1962).

Naylor, Thomas H., and Finger, J. M., "Verification of Computer Simulation Models," *Management Science*, Vol. 14, No. 1, (October 1967), pp. 92-101.

Noettl, John N. and Brumbaugh, Philip, "Information Concepts in Network Planning," *Journal of Industrial Engineering*, Vol. 18, No. 7, (March 1967), pp. 428-435.

Newell, A., and Simon, H. A., "The Logic Theory Machine," *Transactions on Information Theory*, Vol. IT-2, No. 3, IRE, (September 1956).

Paige, H. W., "How PERT-Cost Helps the General Manager," *Harvard Business Review*, Vol. 41, No. 6, (December 1963), pp. 87-95.

Parikh, Shailendra G., and Jewell, William S., "Decomposition of Project Networks," *Management Science*, Vol. 11, No. 3, (January 1965), pp. 444-459.

Pearlman, J., "Engineering Program Planning and Control Through the Use of PERT," *IRE Transactions on Engineering Management*, Vol. EM-7, No. 4, (December 1960), pp. 125-134.

Pearson, E. S., and Hartley, H. O., "Charts of the Power Function for Analysis of Variance Tests Derived from the Non-Central F-Distribution," *Biometrica*, Vol. 38, (1951), pp. 112-130.

Perk, H. N., "Man-Scheduling Program for the IBM 1620," (Revised) IBM Program Library File No. 10.3.013.

Petersen, C. C., "Computational Experience with Variants of the Balas Algorithm Applied to the Selection of R&D Projects," *Management Science*, Vol. 13, No. 9, (May 1967), pp. 736-745.

Petrovic, Radivaj, "Optimization of Resource Allocation in Project Planning," *Operations Research*, Vol. 16, No. 3, (May-June 1968), pp. 559-569.

Phillips, C. R., "Fifteen Key Features of Computer Programs for CPM and PERT," *Journal of Industrial Engineering*, Vol. 15, No. 1, (January-February 1964), pp. 14-20.

Pocock, J. W., "PERT as an Analytical Aid for Program Planning--It's Payoff and Problems," *Operations Research*, Vol. 10, No. 6, (November-December 1962), pp. 893-904.

Pritsker, A.A.B., and Watters, L. J., *A Zero-One Programming Approach to Scheduling with Limited Resources*, RAND Corporation Memo RM-5561-PR, (January 1968).

Pritsker, A.A.B., Watters, L. J., and Wolfe, P. M., "Multi-Project Scheduling with Limited Resources: A Zero-One Programming Approach," *Management Science*, Vol. 16, No. 1, (September 1969), pp. 93-109.

Pritsker, A.A.B., Watters, L. J., and Wolfe, P. M., "Mathematical Formulation: A Problem in Design," *Proceedings of the 19th Annual Institute Conference of AIIE*, Tampa, Florida, (May 1968).

Reeves, Eric, "Critical Path Speeds Refinery Revamp," *Canadian Chemical Processing*, (October 1960).

Root, J. G., "Scheduling with Deadlines and Loss Functions on k Parallel Machines," *Management Science*, Vol. 11, No. 3, (January 1965), pp. 460-475.

Roper, Don E., "Critical-Path Scheduling," *Journal of Industrial Engineering*, Vol. 15, No. 2, (March-April 1964), pp. 51-60.

Rosenbloom, R. S., "Notes on the Development of Network Models for Resource Allocation in R&D Projects," *IEEE Transactions on Engineering Management*, Vol. EM-11, No. 2, (June 1964), pp. 58-62.

Rowe, A. J., "Toward A Theory of Scheduling," *Journal of Industrial Engineering*, Vol. 11, No. 2, (March 1960), pp. 125-136.

Royer, King, "CPM vs. Cost Control," *Building Construction*, (November 1963), pp. 62-70.

Sauer, Ray N., "Least Cost Estimating and Scheduling (LESS) Scheduling Portion," International Business Machines Corporation, Houston, Texas, (February 1961).

Sayer, Kelley, and Walker, M. R., "Critical Path Scheduling," *Factory*, (July 1960), pp. 74-77.

Schultz, E. F., Jr., "Rules of Thumb for Determining Expectations of Mean Squares in Analysis of Variance," *Biometrics*, Vol. 11, (November 1955), pp. 123-148.

Simms, T. J. G., "The Critical Path Method - a New Approach to Planning," *Engineering and Contract Record*, (June 1961).

Simon, Herbert A., and Newell, Allen, "Heuristic Problem Solving: The Next Advance in Operations Research," *Operations Research*, Vol. 6, No. 1, (January-February 1958), pp. 1-10.

Smith, R. D., and Dudek, R. A., " A General Algorithm for Solution of the n-Job, M-Machine Sequencing Problem of the Flow Shop," *Operations Research*, Vol. 15, No. 1, (January-February 1967), pp. 71-82.

Tonge, Fred, "The Use of Heuristic Programming in Management Science," *Management Science*, Vol. 7, No. 1, (October 1960), pp. 21-42.

Trilling, Donald R., "Job Shop Simulation of Orders that are Networks," *Journal of Industrial Engineering*, Vol. 17, No. 2, (February 1966), pp. 59-72.

Trilling, Donald R., "The Use of a Job-Shop Simulator in the Generation of Production Schedules," *Production and Inventory Management*, Vol. 7, No. 3, (July 1966), pp. 57-76.

Van Slyke, Richard M., "Monte Carlo Methods and the PERT Problem," *Operations Research*, Vol. 11, No. 5, (September-October 1963), pp. 839-851.

Walton, H., "Experience of Application of Critical Path Method to Plant Construction," *Operational Research Quarterly*, Vol. 15, No. 1, (March 1964), pp. 9-16.

Wilson, E. B., and Hilferty, M. M., "The Distribution of Chi-Square," *Proceedings of National Academy Sciences*, USA, Vol. 17, p. 684.

Woodgate, H. S., "Planning Networks and Resource Allocation," *Datamation*, Vol. 14, No. 1, (January 1968).

Wagner, H., "An Integer Linear Programming Model for Machine Scheduling," *Naval Research Logistics Quarterly*, Vol. 6, No. 2, (1959), pp. 131-140.

Watters, L. J., "Reduction of Integer Polynomial Programming Problems to Zero-One Linear Programming Problems," *Operations Research*, Vol. 15, No. 6, (November-December 1967), pp. 1171-1174.

White, W. W., "Comments on a Paper by Bowman," *Operations Research*, Vol. 9, No. 2, (March-April 1961), pp. 274-276.

Wiest, J. D., "Some Properties of Schedules for Large Projects with Limited Resources," *Operations Research*, Vol. 12, No. 3, (May-June 1964), pp. 395-418.

Wiest, J. D., "A Heuristic Model for Scheduling Large Projects with Limited Resources," *Management Science*, Vol. 13, No. 6, (February 1967), pp. 359-377.

Wiest, J. D., "Heuristic Programs for Decision Making," *Harvard Business Review*, Vol. 44, No. 5, (September-October 1966).

Zimmerman, Lawrence S., and Shaffer, L. R., "A Network Approach to Resource Scheduling," *Construction Research Series No. 10*, Urbana, Illinois: Department of Civil Engineering, University of Illinois, (June 1967).

Unpublished Dissertations and Theses

Bennett, Fred L., "Some Approaches to the Critical Path Scheduling Resource Allocation Problem," (Ph.D. Dissertation, Cornell University, 1966).

Davis, E. W., "An Exact Algorithm for the Multiple Constrained-Resource Project Scheduling Problem,"(Ph.D. Dissertation, Yale University, 1969).

Fendley, L. G., "The Development of a Complete Multi-Project Scheduling System Using a Forecasting and Sequencing Technique," (Ph.D. Dissertation, Arizona University, 1967).

Gere, William S., Jr., "A Heuristic Approach to Job Shop Scheduling," (Ph. D. Dissertation, Carnegie Institute of Technology, 1962).

Knight, R. M., "Resource Allocation and Multi-Project Scheduling in a Research and Development Environment," (M.S. Thesis, Massachusetts Institute of Technology, 1966).

McGowan, L. L., "Monte Carlo Techniques Applied to PERT Networks," (M.Sc. Thesis in Statistics, Texas A&M University, 1964).

Mize, J. H., "A Heuristic Scheduling Model for Multi-Project Organizations," (Ph.D. Dissertation, Purdue University, 1964).

Pascoe, T. L., "An Experimental Comparison of Heuristic Methods for Allocating Resources," (Ph.D. Dissertation, Cambridge University, 1965).

Weist, J. D., "The Scheduling of Large Projects with Limited Resources," (Ph.D. Dissertation, Carnegie Institute of Technology, 1963).

VITA

Name: James Herbert Patterson

Born: Denver, Colorado; March 30, 1943

Degrees: B.S., University of Colorado, 1966
M.B.A., Indiana University, 1968
D.B.A., Indiana University, 1970

Teaching Experience: Teaching Associate, Indiana University, 1969-1970.

Fellowships and Awards: Colorado University Fellowship, 1964-1966.
N.D.E.A. Fellowship, Indiana University, 1966-1969.

Professional Societies: The Institute of Management Sciences
The Operations Research Society of America
Beta Gamma Sigma

DOCUMENT CONTROL DATA - R & D

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

1. ORIGINATING ACTIVITY *(Corporate author)* Naval Ammunition Depot Crane, Indiana	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

ALTERNATE METHODS OF PROJECT SCHEDULING WITH LIMITED RESOURCES

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

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

PATTERSON, JAMES H.

6. REPORT DATE 20 November 1970	7a. TOTAL NO. OF PAGES 133	7b. NO. OF REFS

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	RDTR No. 174
c.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	

10. DISTRIBUTION STATEMENT

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

The allocation of resources to contemporaneous projects is a problem characteristic of manufacturing firms, contracting firms, and research and development organizations. In allocating resources it is often found that activities demand more resources than are available. When this happens a question arises regarding which jobs to delay and which jobs to schedule. The object of this study is to statistically assess many of the heuristic solution procedures available for scheduling multiple projects.

A

DD FORM 1 NOV 65 1473 (PAGE 1)
S/N 0101-807-6801