16.2.78

"AN EXPERIMENTAL INVESTIGATION OF SOME HEURISTICS FOR SCHEDULING RESOURCE-CONSTRAINED PROJECTS"

by

DALE F. COOPER, BSc(Hons),

Department of Computing Science,

The University of Adelaide.

Thesis submitted for the degree of Doctor of Philosophy.

8 March 1974.

D. A. A. Mackor.

I give consent to this copy of my thesis, when deposited in the University Library, being available for loan and photocopying.

Date 26 MAR 75 ..... Signed

CONTENTS

		para	page
	Summary		vi.
	Statement		viii.
	Acknowledgements		ix.
PART 1.	INTRODUCTION		
Chapter 1.	Introduction	101	1
	Historical Background	101	1
	Thesis Content	106	4
	Notation	110	6
Chapter 2.	Elements of Graph Theory	201	7
	Graphs	202	7
	Properties of Γ	207	9
	Properties of Graphs	211	11
	Partial Ordering	213	12
	Topological Ordering	216	13
Chapter 3.	Project Scheduling Problems	301	17
	Projects	301	17
	Critical Path Schedules	304	20
	Scarce Resources	308	23
	Problem Requirements	311	25
	Objectives	314	26
	Duration-based Objectives	316	26
	Resource-based Objectives	320	28
	Combined Objectives	325	31

		para	page
Chapter 4.	Related Problem Areas	401	36
	Assembly Line Balancing	403	37
	Job Shop Scheduling	407	40
PART 2.	THE RESOURCE-CONSTRAINED PROJECT SCHEDULING PROBLEM		
Chapter 5.	Introduction to the Resource-Constrained Problem	501	45
	Formulation	503	45
	Schedules and Permutations	507	47
	Solution Methods	515	55
	Integer Linear Programming Procedures	519	58
	Combinatorial Methods	522	60
Chapter 6.	Heuristic Methods	601	63
	Serial and Parallel Methods	602	63
	Priority Rules	606	71
	Sampling Methods	635	90
Chapter 7.	Experimental Procedures	701	97
	Aims	702	97
	Measures of Effectiveness	706	98
	Project Description	710	100
	Size and Structure Parameters	712	100
	Time Parameters	715	104
	Resource Parameters	718	108
	Parameter Values	721	111
	Project Generation	724	113
	Experimental Procedures for the Parallel Methods	737	120
	Examination of Effects and Interactions	743	126

		para	page
	Experimental Procedures for the Sampling Methods	752	131
	Procedures for the Comparison of Methods	759	137
	Significance Levels	764	139
	Programming Methods for Schedule Generation	766	140
Chapter 8.	Results	801	144
	Introduction	801	144
	Outline	803	144
	AOV Results for the Parallel Method	809	146
	Priority Rules with the Parallel Methods - General Results	814	151
	Density Main Effect - Parallel Methods	818	159
	Order Strength Main Effect - Parallel Methods	821	161
	Resource Factor Main Effect - Parallel Methods	825	165
	Rules by Order Strength Interaction - Parallel Methods	827	167
	Rules by Resource Factor Interaction - Parallel Methods	832	174
	Rules by Density Interaction - Parallel Methods	835	177
	Priority Rules with the Parallel Methods - Detailed Discussion	838	183
	Parallel Methods - Summary and Conclusions	858	198

		para	page
	AOV Results for the Sampling Methods	863	199
	Distribution of Sampling Solutions	869	205
	Sampling Methods - Summary and Conclusions	876	216
	Implementation of Sampling Methods	880	218
	Comparison of Parallel and Sampling Methods	883	221
	Measures of Effectiveness	887	224
	Problems from the Literature	891	225
Chapter 9.	Summary and Conclusions	901	227
	Parallel Methods - Results	904	228
	Parallel Methods - Conclusions	911	230
	Serial Methods	915	233
	Sampling Methods	916	233
	Comparison of Methods	920	235
	General Conclusions	922	236
	Final Remarks	925	237
	ANNEXES		
Annex A.	Notation		239
Annex B.	Non-uniqueness of Topological Ordering		248
Annex C.	Size of Combinatorial Problems		250
Annex D.	Generation of the Open Closure Matrix		254
Annex E.	Programming Methods		258
Annex F.	Results		263
Annex G.	Rule 18'		275

		page
Annex H.	Distribution of Sampling Solutions	277
Annex I.	Comparison of Methods	282
Annex J.	Problems from the Literature	284
	BIBLIOGRAPHY	288

SUMMARY

This thesis is concerned with the problem of scheduling a project with constraints on the quantities of resources available to it. We are primarily interested in simple heuristic procedures, since exact solution methods are computationally impracticable for large projects. An experimental investigation of two classes of heuristic is described, and results are presented.

In the first part of the thesis, some historical background is given, and general project scheduling problems are introduced. It is noted that solution techniques from the related areas of assembly line balancing and job shop scheduling have been useful for project scheduling in the past, and similarities between these problems and the project scheduling problem are discussed.

In the second part of the thesis, we formulate the resource-constrained project scheduling problem. Exact solution methods are discussed, and are shown to be unsuitable for practical use. Simple serial and parallel heuristics are described, and a set of priority rules, used in the heuristics, is defined. A different class of approximate solution method, based on a sample of feasible schedules, is described. An experimental investigation of the parallel methods and the sampling methods is detailed; this experiment was designed to examine the effects of different priority rules and different project characteristics on the schedules obtained by each of the methods, and to compare the methods themselves. (It is noted that certain priority rules are static, and produce the same schedules in either serial or parallel methods; thus serial methods are investigated implicitly in

the experiment.)

Results of the experiment are presented, together with the results of some further experiments that were suggested by the initial analyses. From the results we conclude that, for both the parallel and sampling methods, certain of the priority rules are more useful for scheduling than others, and that the characteristics of the project have a significant effect on the schedules. Serial methods are shown to be as effective as parallel methods, if a good static priority rule is chosen. Finally, we show that the sampling methods are better than the parallel methods, and obtain a quantitative measure of the improvement in a project schedule that we might expect by using a sampling technique rather than the corresponding parallel method; even with a small sample size, savings of at least six per cent of project cost can be made.

Some general conclusions are drawn, and possible areas for further research are identified.

I state that this thesis contains no material which has been accepted for the award of any other degree or diploma in any other University and that, to the best of my knowledge and belief, the thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

(DALE F. COOPER)

28 February, 1974.

ACKNOWLEDGEMENTS

I wish to thank my supervisor, Dr. J.L.C. Macaskill, Department of Computing Science, University of Adelaide, for his invaluable assistance and guidance throughout this research.

Guidance in Analysis of Variance and statistical tests of significance, received from Mrs. S.M. Williams and Mr. R.J. Willson, both of the Department of Psychology, University of Adelaide, is gratefully acknowledged.

I also wish to thank Mr. G. Joss, Senior Operator, University of Adelaide Computing Centre, and his staff, for their patient and friendly assistance.

PART 1. INTRODUCTION

CHAPTER 1. INTRODUCTION

CHAPTER 1. INTRODUCTION

Historical Background

101. Project planning methods based on a graphical representation of the activities of a project and their interrelationships first appeared in the late 1950s. Malcolm, Roseboom, Clark and Fazar (1959) introduced PERT (Program Evaluation and Review Technique), and Kelley and Walker (1959), and later Kelley (1961), produced CPM (Critical Path Method). These methods could be applied with ease to large problems, and quickly gained wide acceptance. However, both PERT and CPM assume unlimited resources, which is not a valid assumption in most real problems.

102. When resources are scarce, two problems may be defined. The first of these, of major interest in this thesis, is the problem of scheduling a project so as to minimise some function of its duration, subject to fixed limits on the availability of resources to the project; in this chapter, we shall refer to this as the project scheduling problem. The second problem involves the minimisation of some function of the resource requirements of the project, with the constraint that the project must be completed within some specified project horizon or due date; this is the resource smoothing problem. These two problems, together with the related problems of assembly line balancing and job shop scheduling, belong to the class of scheduling

problems; many of the scheduling techniques in this general area are applicable, with minor modifications, to a range of particular problems, including that of project scheduling. Muth and Thompson (1963) provide an overview of the early state of scheduling problems; in particular, Levy, Thompson and Wiest (1963a, 1963b) and Kelley (1963) deal with the project scheduling problem. Moder and Phillips (1970) cover a range of topics related to the scheduling of projects, with and without resource considerations. Davis (1966), Lerda-Oldberg (1966) and Roy (1966) provide surveys of the early literature, and Woodgate (1973) reviews developments in project planning systems from a practical viewpoint.

103. Many exact solution procedures for the project scheduling problem have been proposed, but in every case the method has been computationally impracticable for all but the smallest projects, either because the model is too large, or because the procedure is too lengthy, or both. Integer linear programming (ILP) formulations of job shop scheduling problems were proposed by Bowman (1959), Wagner (1959) and Manne (1960); Pritsker, Watters and Wolfe (1969) have developed a 0-1 ILP model for multi-project scheduling, but the number of variables and constraints quickly becomes large as the problem size increases. A variety of combinatorial methods, relying principally on some form of branch and bound process to enumerate all possible schedules, have been widely proposed: schedule construction using permutations was first used

by Giffler and Thompson (1960) for the job shop scheduling problem, and applied to the project scheduling problem by Schrage (1970); Johnson (1967) uses a branch and bound method based on a heuristic given by Moder and Phillips (1964); Davis and Heidorn (1970) use an algorithm based on a method by Gutjahr and Nemhauser (1964) for the assembly line balance problem; Mason and Moodie (1971) also use an enumerative approach for minimising cost of changing resource usage in the project scheduling problem. A disjunctive graph model was defined by Balas (1970), and solved by Gorenstein (1972). Bennington and McGinnis (1972) give an excellent review of the state of exact solution procedures.

104. Heuristic procedures provide the only viable solution techniques for large scheduling problems. The simplest methods resolve conflicts between jobs that are competing for the same scarce resources by means of priority rules; many different rules have been used for this purpose in project scheduling, and in similar ways in related scheduling problems. Priority rules of one form or another have been used by Helgeson and Birnie (1961), Heller and Logemann (1962), Fisher and Thompson (1963), Verhines (1963), Pascoe (1965), Tonge (1965), Arcus (1966), Conway, Maxwell and Miller (1967), Wiest (1967), Fendley (1968), Karni (1973) and Thesen (1973); the use of methods of this form is widespread, and the preceding list is by no means comprehensive. Ciobanu (1971) uses a similar heuristic method. Pascoe (1965) describes

an experimental comparison of some of these simple serial and parallel methods, in which a limited number of priority rules are applied to a set of 20-job projects, and presents results.

105. More advanced scheduling methods choose a sample of random schedules, and select the best schedule from the sample. Methods of this kind, in which the sample may be biased by using priority rules to weight the selection of jobs, were suggested by Giffler and Thompson (1960): they were applied to the resource smoothing problem by Levy, Thompson and Wiest (1962) and Ciobanu (1972), to job shop scheduling by Giffler, Thompson and Van Ness (1962), and to the project scheduling problem by Wiest (1967). Arcus (1966) describes an extensive comparison of different priority rules used to bias the sampling process, for the assembly line balance problem. Other methods of biasing the sample, by means of weighted selection of priority rules, have been reported by Heller and Logemann(1962), Fisher and Thompson (1963) and Tonge (1965), but results in this area are inconclusive.

Thesis Content

106. This thesis is concerned with the problem of scheduling a project such that its duration is minimised, subject to limitations on the availability of resources. In Part 1 (Chapters 1 to 4), we examine the general area and some related problems; in Part 2 (Chapters 5 to 9), the particular problem is discussed in more detail, and an experimental investigation is reported.

107. Part 1. As we noted in the first paragraph, project scheduling methods have a historical basis in the graphical representation of projects; in Chapter 2, we formalise some of the relevent graph theoretic concepts that we will find useful later. In Chapter 3, general project scheduling problems are described, and different objectives of the project scheduling process in each case are discussed. We note in Chapter 4 that solution techniques from related problem areas can often be applied, with minor changes, to project scheduling; assembly line balancing and job shop scheduling are compared with project scheduling, and similarities are considered.

108. Part 2. Chapter 5 introduces in detail the resource-constrained project scheduling problem: the problem is formulated, and some preliminary definitions are given; solution techniques are considered, and exact methods are shown to be impracticable. Some heuristic procedures of interest are described in Chapter 6: the serial and parallel methods are outlined, as are sampling methods, and a set of priority rules is defined. In Chapter 7 we describe an experimental investigation of the heuristic methods that were outlined in Chapter 6. Various parameters that describe the properties of projects are defined, and experimental procedures are detailed. The aims of the experiment were, briefly:

a. To examine the parallel methods, and to investigate the effects of changing the priority rules and the project

characteristics. This part of the investigation extends Pascoe's (1965) results to larger projects (60 jobs rather than 20) and a wider range of priority rules (26 instead of 11).

b. To examine the sampling methods, in much the same way as the parallel methods. Here, many of Arcus' (1966) techniques for assembly line balancing are applied to project scheduling.

c. To compare the parallel and sampling methods, and to obtain, if possible, a numerical value for the expected benefits to be gained by using one method instead of the other.

109. Results of the experimental investigation are given in Chapter 8, together with details of some further experiments that were suggested by the initial results. In Chapter 9, we summarise the results and present conclusions.

Notation

110. The symbols and notation used in this thesis are given in Annex A.

CHAPTER 2. ELEMENTS OF GRAPH THEORY

CHAPTER 2. ELEMENTS OF GRAPH THEORY

201. Before proceeding with the detailed formulation of the problems to be considered in this thesis, we shall first introduce some of the relevant graph theoretic concepts that will be necessary later. The notation and definitions in this chapter are based on Berge (1962) and Kaufmann (1967); in particular, we shall be dealing with graphs purely in terms of oriented concepts. (Other authors have used 'directed graph' or 'digraph' to describe what we shall call simply 'graph'.)

Graphs

202. Consider a countable set of points, or vertices, such as {A,B,C,D,E} in Fig 2.1. In the diagram, these points are connected by oriented lines, called arcs. These arcs may be denoted either by their names, or by the ordered pairs of vertices which they connect; thus the arc between vertex C and vertex E may be denoted by its name g or by the pair (C,E). The diagram of Fig 2.1 represents a graph.

203. More formally, a graph is a pair (X,L), such that:

a. X is a countable set; and

b. L is a set of ordered pairs of elements of X.

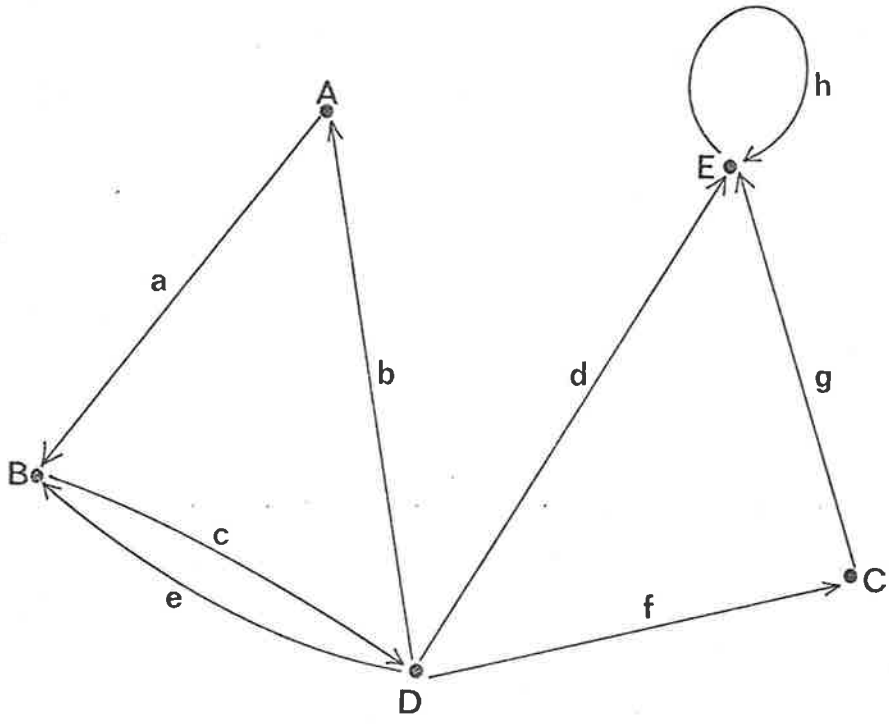

Fig 2.1. Graph G = (X,L), with

X = {A,B,C,D,E}, and

L = {a,b,c,d,e,f,g,h}

204. The set L of arcs defines a relationship amongst the elements of X. We will denote this relationship by a multi-valued function Γ, which maps X into itself. In Fig 2.1 we have: $\Gamma A = \{B\}$, since (A,B) is in L; and $\Gamma D = \{A,B,C,E\}$, since (D,A), (D,B), (D,C) and (D,E) are all in L.

205. Conversely, the function Γ determines the set of arcs L, and hence we can define a graph by the pair (X,Γ), where Γ is a function mapping X into itself.

206. Consider two vertices x and y such that $y \in \Gamma x$, i.e. there is an arc (x,y) in L. We say that y is an immediate successor of x, and x is an immediate predecessor of y.

Properties of Γ

207. Let Γ be a function mapping X into X. If Q is a subset of X, then the image of Q under Γ, written ΓQ, is the set

$$\Gamma Q = \bigcup_{x \in Q} \Gamma x,$$

i.e. the set of immediate successors of all elements x of Q. The functions Γ^2, Γ^3, &c, are defined by

$$\Gamma^2 x = \Gamma(\Gamma x)$$

$$\Gamma^3 x = \Gamma(\Gamma^2 x)$$

.....

In general,

$$\Gamma^k x = \Gamma(\Gamma^{k-1} x),$$

for all integer $k > 1$.

208. The open closure function Γ^* derived from Γ is a function mapping X into X, defined by

$$\Gamma^*x = \bigcup_{k=1}^{\infty} \Gamma^k x.$$

The open closure of x, Γ^*x, is the set of all successors of x. If X is a finite set with n + 1 elements, then

$$\Gamma^*x = \bigcup_{k=1}^{n} \Gamma^k x.$$

209. The inverse of Γ is a function Γ^{-1} defined by

$$\Gamma^{-1}y = \{x \mid y \in \Gamma x\}.$$

$\Gamma^{-1}y$ is the set of immediate predecessors of y. If Q is a subset of X, then

$$\Gamma^{-1}Q = \{x \mid \Gamma x \cap Q \neq \emptyset\}$$

where $\emptyset$ is the empty set, i.e. $\Gamma^{-1}Q$ is the set of all elements of X which have immediate successors in Q. The functions Γ^{-2}, Γ^{-3}, &c, are defined by

$$\Gamma^{-2}x = \Gamma^{-1}(\Gamma^{-1}x)$$

$$\Gamma^{-3}x = \Gamma^{-1}(\Gamma^{-2}x)$$

.....

In general,

$$\Gamma^{-k}x = \Gamma^{-1}(\Gamma^{-k+1}x)$$

for all integer k > 1.

210. The open inverse closure function Γ^{-*} is defined by

$$\Gamma^{-*}x = \bigcup_{k=1}^{\infty} \Gamma^{-k}x.$$

The open inverse closure of x, $\Gamma^{-*}x$, is the set of all predecessors of x. If X is a finite set with n+1 elements, then

$$\Gamma^{-*}x = \bigcup_{k=1}^{n} \Gamma^{-k}x.$$

Example 2.1. Consider the graph of Fig 2.1. We have

$\Gamma A = \{B\}$

$\Gamma^2 A = \Gamma(\Gamma A) = \Gamma B = \{D\}$

$\Gamma^3 A = \Gamma(\Gamma^2 A) = \Gamma D = \{A,B,C,E\}$

$\Gamma^4 A = \Gamma(\Gamma^3 A) = \Gamma\{A,B,C,E\} = \{B,D,E\}$

$\Gamma^* A = \{A,B,C,D,E\}$

$\Gamma^{-1} A = \{D\}$

$\Gamma^{-2} A = \Gamma^{-1} D = \{B\}$

$\Gamma^{-3} A = \Gamma^{-1} B = \{A,D\}$

$\Gamma^{-4} A = \Gamma^{-1}\{A,D\} = \{B,D\}$

$\Gamma^{-*} A = \{A,B,D\}$

Properties of Graphs

211. A path in a graph G = (X,L) is a sequence of vertices $[x_1, x_2, \ldots, x_k]$ such that any pair of vertices x,y which are adjacent in the sequence are connected by an arc, i.e. $(x,y) \in L$, or $y \in \Gamma x$. A circuit is a finite path in which the initial vertex coincides with the final vertex. A loop is a circuit of the form [x,x].

Example 2.2. In Fig 2.1, [A,B,D,E] is a path. [A,B,D,A] is a circuit. The arc h = (E,E) forms a loop.

212. There is a path from vertex x to vertex y if $y \in \Gamma^* x$. Thus x lies in a circuit if $x \in \Gamma^* x$, and there is a loop at vertex x if $x \in \Gamma x$.

Partial Ordering

213. A relation P, defined on a set Y, is a weak ordering if it has the properties:

a. Reflexivity. xPx for all x in Y.

b. Transitivity. If xPy and yPz, then xPz, for all x,y,z in Y.

Example 2.3. Let Y be the set of integers, and let xPy if x is an integer multiple of y. Then P is a weak ordering.

214. A relation P will be associated with a graph $G = (x,\Gamma)$ in the following way:

For two vertices x,y, let xPy if x=y, or if there is a path from x to y (i.e. if $y \in \Gamma^* x$).

This relation is a weak ordering.

215. If (X,Γ) contains no circuits, its associated relation P has the property:

If xPy and yPx, then x=y. In this case, we say that P is a partial ordering, and the set X is partially ordered. Conversely, if the relation associated with a graph is a partial ordering, then the graph does not contain circuits.

Example 2.4. Fig 2.2 shows the graph $G = (X,\Gamma)$ of a partially ordered set X.

Topological Ordering

216. Let X be a finite set containing N elements[1], and let Γ define a partial ordering on X. Let the elements of X be numbered uniquely by the integers 1,2,...,N, and let n(x) represent the number associated with element x of X. Then n is a function mapping X onto the integers 1,2,...,N such that

$$x = y \text{ if and only if } n(x) = n(y),$$

for all x,y in X.

217. Such a numbering of a partially ordered set X is a topological ordering if its associated function n has the property

$$\text{if } x \in \Gamma y, \text{ then } n(x) > n(y),$$

for all x,y in X, i.e. if the element numbered i precedes the element numbered j, then i < j. Fulkerson (1962) gives a simple algorithm for determining a topological ordering of a partially ordered set.

218. A partially ordered set X may often be topologically ordered in more than one way. The associated function n is not unique in general. (See Annex B.)

---

1. We shall denote the number of elements in any set Y by #(Y), so in this case we have #(X) = N.

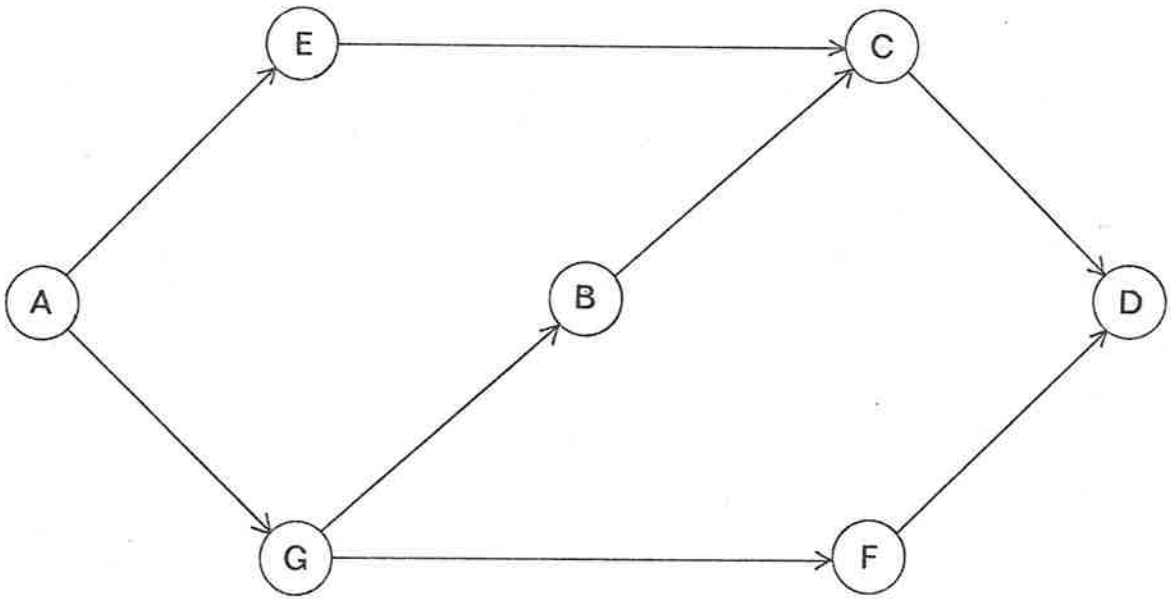

Fig 2.2. Graph G = (X,Γ) of a partially ordered set X.

Example 2.5. Fig 2.3 shows a topological ordering of the graph of Fig 2.2.

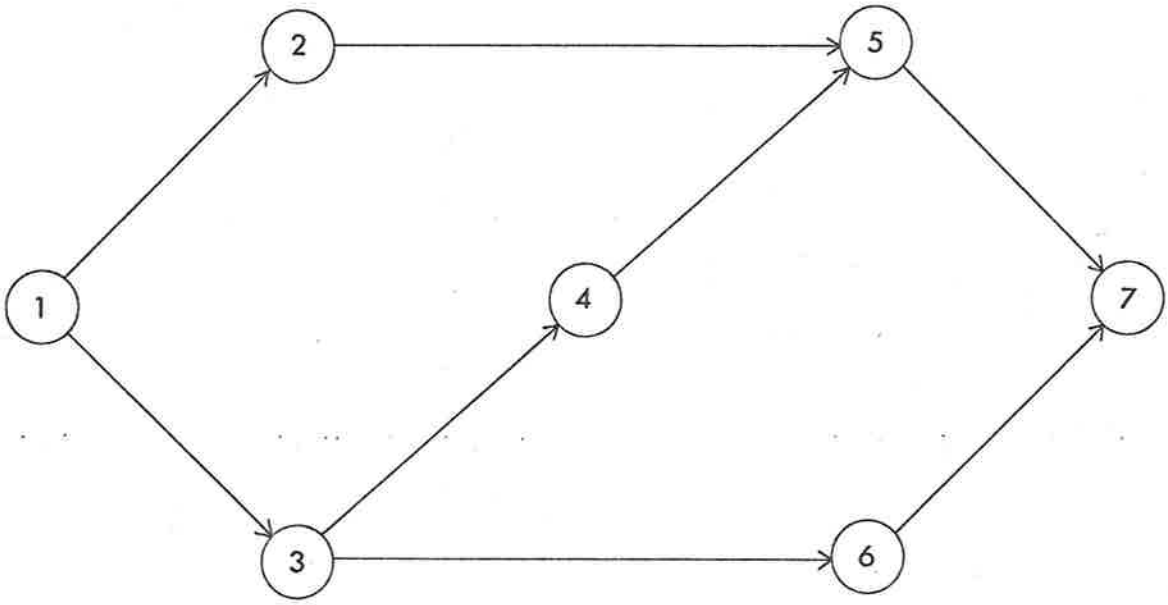

Fig 2.3. A topological ordering of the graph of Fig 2.2.

CHAPTER 3. PROJECT SCHEDULING PROBLEMS

CHAPTER 3. PROJECT SCHEDULING PROBLEMS

Projects

301. A project is a partially ordered set A of activities, or jobs, with each of which is associated a non-negative duration and possibly other quantities describing the job. We can represent a project by a graph $G = (A,\Gamma)$, where Γ is the function determined by the partial ordering on A; the duration of a job and other descriptive information is often shown next to the appropriate vertex of the graph. The partial ordering in a project is derived from the logical and technological relationships between jobs: if job j cannot start until job i has been completed, because of the natures of i and j, then we say that i must precede j, or $j \in \Gamma i$.

Example 3.1. Fig 3.1 shows the graph of a project whose individual jobs are arithmetic calculations; the time required to perform each set of calculations is shown. Jobs 2 and 3 perform calculations on the result of job 1, and must wait for job 1 to finish before either of them can start. Job 5 requires results from jobs 2 and 4, and must wait for both of them to be completed before it can commence. Jobs 5 and 6 are independent of each other, and may be performed one at a time in either order, or concurrently.

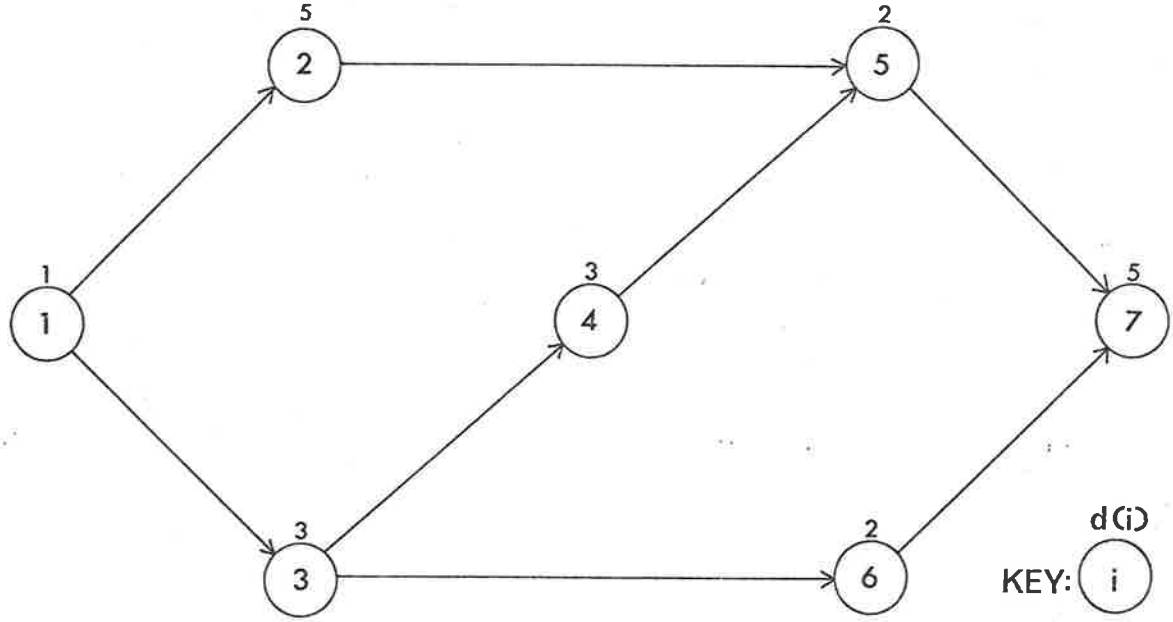

Fig 3.1. A simple project graph, with A = {1,2,3,4,5,6,7} and durations d(i).

302. For an activity i in A, let d(i) be its duration, and let s(i) be the starting time assigned to it by some process. Any set $S = \{s(i) \mid i \in A\}$ is a schedule. For a schedule to be feasible, it must satisfy at least the constraints imposed by the partial ordering; that is, no job may start until all of its immediate predecessors have been completed,

$$s(j) \geq s(i) + d(i), \text{ for all } i \in \Gamma^{-1}j.$$

For such a schedule, the project duration, PDUR, is given by

$$PDUR = \max_{i \in A} \{s(i) + d(i)\} - \min_{i \in A} \{s(i)\},$$

i.e. the project duration is the difference between the finishing time of the latest job and the starting time of the earliest job.

303. In what follows, we shall make the following assumptions:

a. All numerical quantities are integers, unless specifically noted otherwise.

b. Time is divided into unit time periods, each of which is identified by the time instant at the start of the period. Thus time period 5, for example, is the interval between time instants 5 and 6. It will be clear from the context whether a time period or a time instant is meant.

c. The first time instant is time zero, and the first period is period zero.

d. The jobs in A are numbered topologically.

Critical Path Schedules

304. Two schedules of interest are the critical path (CP) early start and late start schedules, in which jobs start as early and as late as possible respectively. The earliest starting time, $\ell(i)$, of job i is given by

$$\ell(i) = 0 \qquad \text{if } \Gamma^{-1}i = \emptyset,$$

$$= \underset{j\in\Gamma^{-1}i}{\text{Max}} \{\ell(j) + d(j)\} \text{ otherwise.}$$

In this case, the project duration is the critical path project duration, CPDUR, given by

$$\text{CPDUR} = \underset{i\in A}{\text{Max}} \{\ell(i) + d(i)\}.$$

(Note that since we have set $\ell(i)$ to zero for jobs i with no predecessors, the term Min $\{s(i)\}$ from the defining equation in para 302 becomes zero.) CPDUR is the minimum time in which the project can be completed.

305. Suppose that the project must be completed within some project horizon, PHOR, or due date. Then the latest starting time, u(i), of job i is given by

$$u(i) = \text{PHOR} - d(i) \qquad \text{if } \Gamma i = \emptyset,$$

$$= \underset{j\in\Gamma i}{\text{Min}} \{u(j)\} - d(i) \text{ otherwise.}$$

In the special case where PHOR = CPDUR, we will denote the late start time of job i by $u^*(i)$.

306. For any job i, we shall define its float, f(i) by

$$f(i) = u(i) - \ell(i).$$

Intuitively, the float of a job i is the maximum amount that the completion of i can be delayed without delaying the entire project. By a critical job we mean a job having minimum float; this minimum float is

$$\min_{i \in A} \{f(i)\} = PHOR - CPDUR$$

There is at least one path $[x,\ldots,y]$ in (A,Γ) in which all jobs are critical and $\Gamma^{-1}x = \Gamma y = \emptyset$; such a path is called a critical path, and the sum of the job durations on such a path is CPDUR. (These definitions and results are adapted from Levy, Thompson and Wiest (1963b), pp 372-373.)

Example 3.2. Table 3.1 shows values of ℓ, u, u^* and f for the project of Fig 3.1. Here,

$$CPDUR = \max_{i \in A} \{\ell(i) + d(i)\} = 14,$$

and PHOR = 20, so

$$\min_{i \in A} \{f(i)\} = PHOR - CPDUR = 6.$$

307. A schedule $S = \{s(i) \mid i \in A\}$ may be represented by a time-scaled bar chart, with job i represented by a line of length d(i) starting at s(i) and extending to the right. (See Battersby (1967), Chapter 5, for further details of time-scaled charts of this nature.) For a schedule S, if we sum the requirements of jobs for resources in each

i	d(i)	ℓ(i)	u(i)	u^*(i)	f(i)	CRITICAL
1	1	0	6	0	6	*
2	5	1	8	2	7	
3	3	1	7	1	6	*
4	3	4	10	4	6	*
5	2	7	13	7	6	*
6	2	4	13	7	9	
7	5	9	15	9	6	*

Table 3.1 CP schedules and float for the project of Fig 3.1, with PHOR = 20.

time period, and plot requirements against time, we obtain a resource profile for the schedule.

Example 3.3. Fig 3.2(a) shows a time-scaled bar chart for the CP early start schedule of the project in Fig 3.1, given in Table 3.1. Fig 3.2(b) shows a resource profile for the same schedule, where each job requires one unit of resource.

Scarce Resources

308. The schedules described above assume unlimited resources. The resource profile for the early start schedule in Example 3.3 (Fig. 3.2(b)) shows that the project requires three units of resource in periods 4 and 5; if, however, there are only two units available, the early start schedule is clearly unfeasible.

309. When resources are scarce, two problems can be defined. One is to minimise some function related to the project duration, subject to fixed resource availabilities; the other is to minimise some function of the project's resource requirements (e.g. minimise the maximum, or peak, resource requirement), while keeping the project duration within the allowed horizon PHOR.

310. The problem requirements and possible objectives will be examined more fully in the following sections.

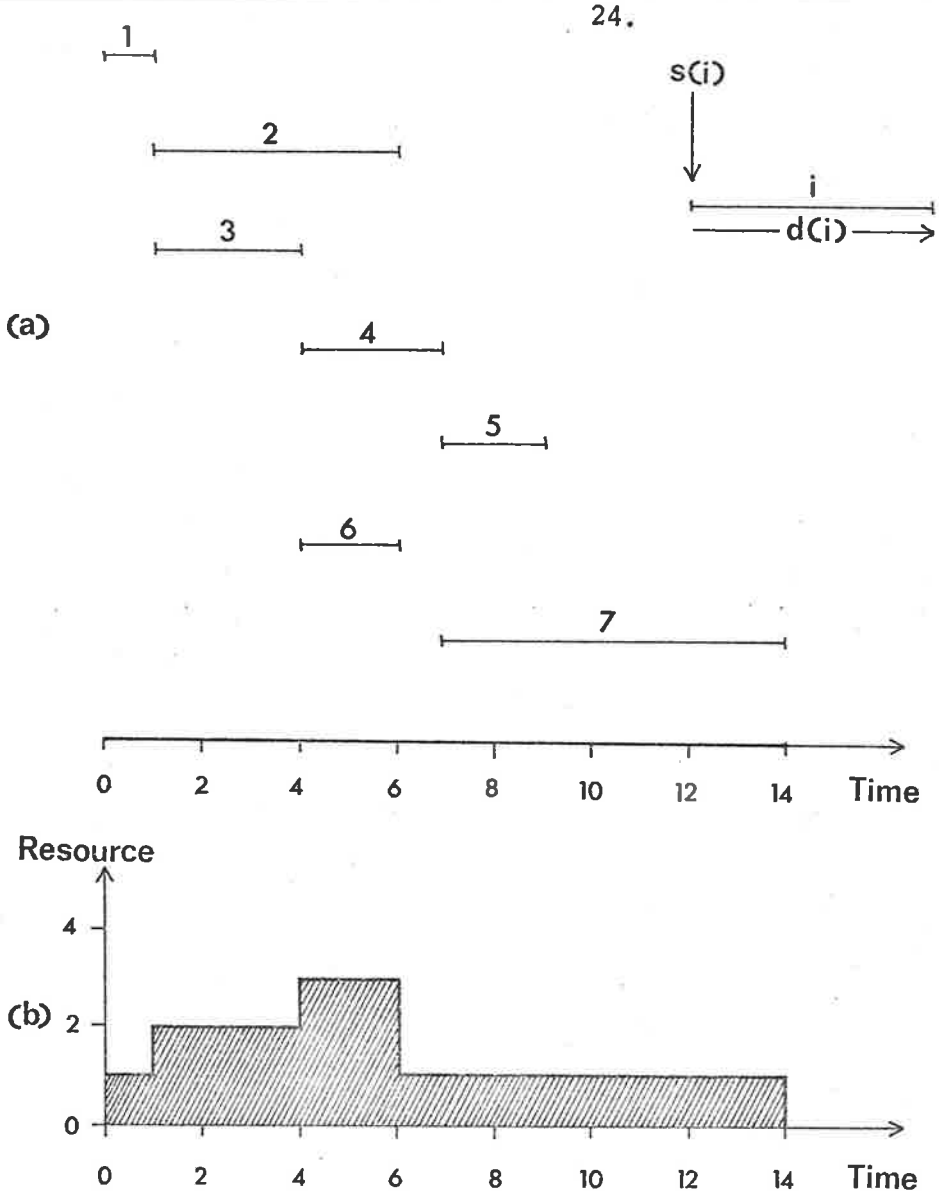

Fig 3.2. CP early start schedule:

a. Time-scaled bar chart.

b. Resource profile.

Problem Requirements

311. There is a partially ordered set A of N jobs. With each job i in A there is associated:

a. A non-negative duration, d(i).

b. A non-negative requirement for r(i,k) units of resource k in each time period in which job i is in progress, for k = 1,2,...,K, where K is the number of distinct kinds of resources used by the project.

c. A set of immediate successor jobs, Γi.

d. A start time, s(i), to be determined.

312. It is required to find start times s(i) such that the following constraints are satisfied:

a. Precedence.

$s(i) \geqslant 0$ if $\Gamma^{-1}i = \emptyset$

$s(i) \geqslant s(j) + d(j)$ for all $j \in \Gamma^{-1}i$, otherwise.

b. Non-preemption. A job, once started, cannot be interrupted, and must run to completion. (A job i with start time s(i) will therefore be in progress in periods s(i), s(i) + 1,...,s(i) + d(i) - 1.)

313. The above constraints are common to both the resource-constrained and the time-constrained problems which are outlined in para 309. Each of these problems requires different additional constraints, which will be considered in the next sections when we discuss the objectives of the scheduling process in each case.

Objectives

314. The scheduling problems may be loosely grouped according to the nature of the objective. This may be related to:

a. Project duration.

b. Resource requirements.

c. Some function of both duration and resources (e.g. cost).

315. Each of these objectives will be discussed in outline in this part. More detailed consideration of the first of these problems will be given in Part 2.

Duration-based Objectives

316. In the single project case, the problem may be stated:

Find the schedule which minimises the project duration while ensuring that the requirements for resources do not exceed a given availability.

317. If R(k,t) is the amount of resource k available in time period t, this problem may be stated more formally as:

Determine that schedule $S = \{s(i) \mid i \in A\}$ which will minimise PDUR

subject to the constraints of para 312 and

$$\Sigma r(i,k) \leqslant R(k,t), \text{ for } k = 1,2,\ldots,K, \text{ and } t=0,1,\ldots,PHOR,$$

where the sum is over all jobs i in progress at time t, i.e. jobs i with

$$s(i) \leqslant t < s(i) + d(i) .$$

318. In the multi-project case, other objectives are possible (see, for example, Pritsker, Watters and Wolfe, 1969):

a. Minimise the total throughput time (sum of the individual project durations) for all projects.

b. Minimise the time by which all projects are completed (minimise makespan).

c. Minimise the total lateness or lateness penalty for all projects.

These objectives will be discussed later in relation to the job shop scheduling problem, but detailed consideration is outside the scope of this thesis.

319. Other objectives may be used to evaluate schedules in different circumstances. Karni (1973) uses schedule evaluation criteria which may have special applications:

a. The amount of delay between the start of one activity and the latest finish of its predecessors (caused by resource restrictions), may be expressed as average delay, AD, to be minimised:

$$AD = \sum_{i \in A} \left(s(i) - \max_{j \in \Gamma^{-1}_i} \{ s(j) + d(j) \} \right) / N.$$

b. The average slippage, AS, between the resource limited schedule and the CP early start schedule, to be minimised:

$$AS = \sum_{i \in A} \left(s(i) - \ell(i) \right) / N.$$

Resource-based Objectives

320. A number of objectives based on resources have been suggested. All are time-constrained, in that they require the project duration not to exceed some specified project horizon or due date, i.e.

$$PDUR \leqslant PHOR.$$

321. A comprehensive set of objectives has been proposed by Wagner, Giglio and Glaser (1964):

a. Minimise the sum of the absolute changes in resource requirements.

b. Minimise the sum of the increases in resource requirements.

c. Minimise the sum of the absolute deviations from the average resource requirement.

d. Minimise the peak resource requirement.

e. Minimise the maximum change in resource requirement.

f. Minimise the sum of squares of changes in resource requirements.

g. Minimise the sum of squares of deviations from the average resource requirement.

A mathematical statement of each of these objective functions for the single-resource case (K = 1) is given in Table 3.2.

SERIAL	PARA	OBJECTIVE
1.	321.a	Min $\Sigma_t \lvert RUSE(1,t+1)-RUSE(1,t)\rvert$
2.	321.b	Min Σ_t Max $\{RUSE(1,t+1)-RUSE(1,t),0\}$
3.	321.c	Min $\Sigma_t \lvert RUSE(1,t)-RAVE(1)\rvert$
4.	321.d	Min $\{\text{Max}_t\ RUSE(1,t)\} \equiv$ Min $RMAX(1)$
5.	321.e	Min $\{\text{Max}_t \lvert RUSE(1,t+1)-RUSE(1,t)\rvert\}$
6.	321.f	Min $\Sigma_t (RUSE(1,t+1)-RUSE(1,t))^2$
7.	321.g	Min $\Sigma_t (RUSE(1,t)-RAVE(1))^2$
8.	322	Max $\Sigma_{i\epsilon A}\ (d(i)*r(i,k))/(PDUR*RMAX(1))$

Table 3.2 Resource-based objectives in the single-resource case (K = 1).

RUSE(k,t) is the quantity of resource k required by the project in period t.

RAVE(k) is the average requirement for resource k over the duration of the project.

RMAX(k) is the maximum (peak) requirement for resource k over the duration of the project.

$$RMAX(k) = \text{Max}_t \{RUSE(k,t)\}$$

322. Each of the objectives of the last paragraph, except d (minimise the peak requirement) attempts to limit the fluctuations in demand for resources. These are particularly suitable if the resource under consideration is manpower, as in the paper by Wagner et al, since they reflect the costs associated with hiring and laying off men. In other situations, it may be more appropriate to minimise the peak requirement; in the case where resources must be maintained at peak levels throughout the duration of the project, this is equivalent to maximising the utilisation of resources. (The utilisation function used by Pascoe (1965), given in line 8 of Table 3.2, shows the correspondence clearly; compare this with line 4 of the table.)

323. Other authors have used similar objectives to those of Wagner et al (1964). Karni (1973) uses a coefficient of variation, COV, which is to be minimised:

$$\text{COV} = \frac{\text{Standard deviation of daily requirements}}{\text{Mean daily requirement}} .$$

Ciobanu (1972) uses as a measure the excess of requirements over some predetermined 'desirable' level.

324. There is a considerable difference in emphasis amongst the resource-based objectives. This reflects the different needs of management in the area of resource usage. Battersby (1967) discusses some of the problems of implementing different management policies in this area.

Combined Objectives

325. The objectives of the project scheduling problem may be formulated in terms of both duration and resources. A simple example of this is the OVERUSE function used by Pascoe (1965), which may be stated:

Minimise the area of the resource profile above some critical resource level R^* or beyond some target duration T. In the single resource case, denoting by RUSE (k,t) the requirement at time t for resource k by the project, this may be expressed:

Minimise OVERUSE

$$= \sum_{t=0}^{T} \text{Max } \{RUSE(1,t)-R^*, 0\}$$

$$+ \sum_{t = T+1}^{PDUR} RUSE\ (1,t).$$

Fig 3.3 shows diagrammatically the effect of this objective, which is to minimise the area of the resource profile outside the rectangle determined by R* and T.

326. Mason and Moodie (1971) have formulated the problem in terms of minimising cost. Costs c_1 and c_2 are linear penalty costs associated with a unit increase and a unit decrease, respectively, in the resource level; c_3 is the cost incurred by each time unit of project delay past some target duration. (Mason and Moodie use the critical path duration, CPDUR, as their target, but any target could be substituted.) Thus, there are costs associated with changes in resource requirements (cf. resource-based objectives)

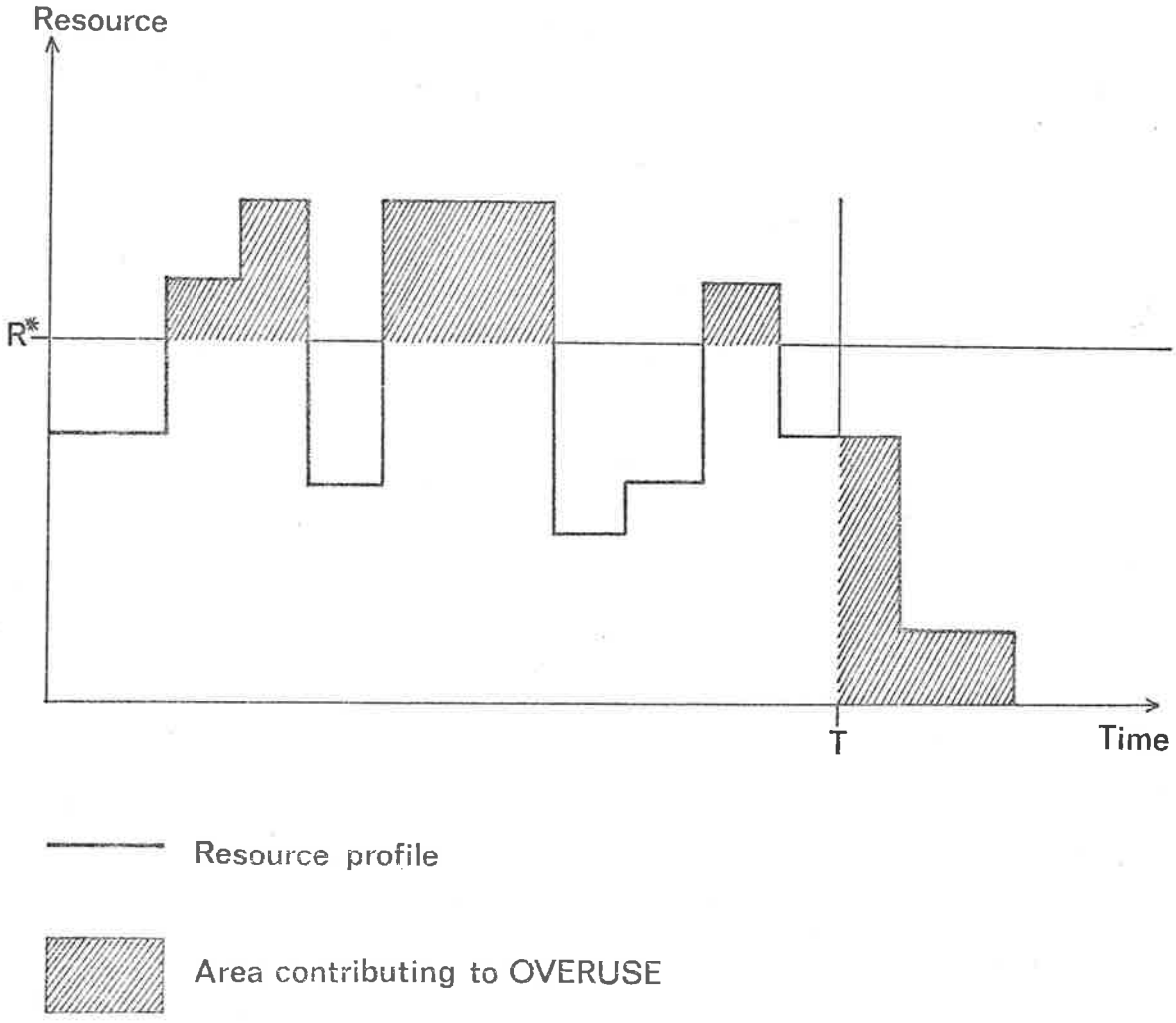

Fig 3.3. The OVERUSE objective, where R* is a critical resource level and T is a target duration.

and with delays in project completion (c f. duration-based objectives).

327. The cost of a change in resource level at time t is, for a single resource,

$$c(t) = c_1 * \text{Max } \{RUSE(1,t+1)-RUSE(1,t),0\} + c_2 * \text{Max } \{RUSE(1,t)-RUSE(1,t+1),0\}.$$

The project delay cost, for target T, is

$$d = c_3 * \text{Max } \{PDUR-T,0\}.$$

Then the total project cost is given by

$$p = \Sigma_{t=0}^{PHOR} c(t) + d.$$

This project cost is to be minimised.

328. The basic model described above may be changed, by appropriate assignment of costs, to give a variety of problems, as may be seen from the next two examples.

Example 3.4. Assign very large costs to increases in resource requirements above a critical level R*, and small costs to increases in project duration over the critical path duration. A cost minimising procedure will first reduce resource requirements below critical levels, since this achieves the greatest cost reduction, and then reduce the project duration. This is equivalent to minimising project duration with resource constraints (para 316). The problem may be stated more formally as

$$\text{Minimise } p = \Sigma_{t=0}^{PHOR} c_1 * \text{Max } \{RUSE(1,t)-R^*,0\}$$
$$+ c_3*(PDUR - CPDUR),$$

where $c_1 >> c_3$.

Example 3.5. Assign very large costs to increases in project duration past the target date T, and small costs to changes in resource requirements. A cost minimising procedure will first reduce the project duration below the target date, and then smooth the resource requirements. If unit changes in resource requirements incur unit cost, i.e. $c_1 = c_2 = 1$, we have

$$\text{Minimise } p = \Sigma_{t=0}^{PHOR} |RUSE(1,t+1)-RUSE(1,t)|$$
$$+ c_3* \text{ Max } \{PDUR-T,0\},$$

where c_3 is very large. This is exactly the problem of minimising the sum of the absolute changes in resource requirements, subject to a constraint on project horizon (para 321.a, and line 1 of Table 3.2). Similarly, if in the formulation we specify $c_1 = 1$, $c_2 = 0$, and c_3 very large, we have the problem of para 321.b, minimise the sum of increases in resource requirements, subject to a constraint on the project duration.

329. Cullingford and Prideaux (1973) use a combined objective in a slightly different context. They use a cost function of the form

$$c_1 * \Sigma_{t=0}^{PHOR} (RUSE(1,t+1)-RUSE(1,t))^2$$
$$+ c_2*PDUR + \text{constant terms},$$

where c_1 and c_2 are cost coefficients. This cost function is generalised to the continuous case, and minimised using variational methods.

330. Formulations of this nature allow a wide variety of policies to be implemented, from entirely duration-bound to entirely resource-bound. However, exact methods for the solution of models of this kind are computationally impractical for problems of a realistic size. (In paras 522 to 525, and Annex C, we discuss the size of combinatorial problems which are formulated in a similar way to the model of Mason and Moodie.) The remainder of this thesis, then, will be concerned with models in which the objectives are functions of either time or resources, but not both, as in these very general formulations.

CHAPTER 4. RELATED PROBLEM AREAS

CHAPTER 4. RELATED PROBLEM AREAS

401. In this chapter, we shall briefly examine some problems that are related to the project scheduling problem, and we shall show that they may be formulated in a similar way. There are several reasons for this:

a. By considering related problems, we are able to gain further insight into the nature of our problem of interest, and the forms that it may take.

b. On a more practical level, we note that solution methods, originally developed in these related areas, have been applied to the project scheduling problem in the past, and therefore are of direct interest. In fact, some of the procedures tested in this thesis are adapted from such methods.

402. The main problems to be considered in this chapter are assembly line balancing and job shop scheduling. The intention of this chapter is to show that these problems may be formulated in a similar way to the project scheduling problem; for this reason, only a few general references will be given, and more detailed reference to the literature of these areas will be made when we consider different kinds of solution methods in Chapters 5 and 6.

Assembly Line Balancing

403. Formulation. There is a set of elemental tasks, or work elements {i}, each requiring a known operation time d'(i) per unit of product. A partial ordering exists among the work elements. A single model assembly line is an assignment of work elements to a sequence of groups, called stations, such that if one work element must precede another, then it is assigned either to the same station or to a preceding station in the sequence. The length, L, of the assembly line is the number of stations, and the cycle time, C, is the maximum of the sums of operation times of tasks assigned to each station.

404. Objectives. There are two closely related optimisation problems:

a. Given a maximum cycle time C', find an assembly line with minimum length.

b. Given a maximum length L', find an assembly line with minimum cycle time.

405. Comparison with project scheduling. The assembly line balancing problem is very similar to the project scheduling problem with a single resource; Table 4.1 shows corresponding concepts in the two areas. The assembly line problem may be solved using project scheduling techniques by making the assignments detailed in Table 4.2, and changing the precedence constraints of para 312.a from

	PROJECT	ASSEMBLY LINE
1	Job, i	Work element, i
2	Resource requirements, r(i,1)	Operation times, d'(i)
3	Start time, s(i)	Station s'(i) to which i is assigned
4	Maximum resource requirement, RMAX(1)	Cycle time, C
5	Project duration, PDUR	Line length, L

Table 4.1. Corresponding concepts in project scheduling and assembly line balancing.

	ASSIGNMENT
1	d(i) ← 1, for all i
2	K ← 1
3	r(i,1) ← d'(i), for all i
4	R(1,t) ← C', for all t
	(Minimise PDUR/minimise L)
5	PHOR ← L'
	(Minimise RMAX(1)/minimise C)

Table 4.2 Conversion of the assembly line balancing problem to the project scheduling formulation of Chapter 3.

$$s(i) \geqslant s(j) + d(j) \qquad \text{for all } j \varepsilon \Gamma^{-1} i$$

to

$$s'(i) \geqslant s'(j) \qquad \text{for all } j \varepsilon \Gamma^{-1} i,$$

where s'(i) is the station to which work element i is assigned. The problem of minimising line length L with a given cycle time C' then becomes a problem of minimising project duration PDUR with a given resource constraint that is constant throughout the duration of the project. The problem of minimising the cycle time C with a given line length L' becomes a problem of minimising the peak resource requirement within a given project horizon PHOR.

406. References. The formulation and objectives given here follow Mastor (1970) who describes an experimental comparison of solution procedures for the assembly line balancing problem, and gives detailed references. Table 4.1 is very like one published by Moodie and Manderville (1966), who use a similar comparison in the application of assembly line balancing techniques to the problem of resource smoothing in projects.

Job Shop Scheduling

407. Formulation[1]. In its most general form, a job shop consists of N jobs that require processing on one or more of M

1. Other formulations of the job shop scheduling problem may be used. This is a general model of a shop where all information is known at the commencement of the scheduling period.

machines, possibly more than once. The k^{th} processing of job j on machine m is called an operation, and is denoted by mjk; such an operation requires a known operation time d(mjk). The operations of a job are partially ordered. A schedule for such a job shop is an assignment of starting times s(mjk) to operations mjk such that the following constraints are satisfied:

a. Precedence. If operation mjk of job j precedes operation pjq, then

$$s(mjk) + d(mjk) \leqslant s(pjq)$$

b. Jobs. There may be only one operation mjk of job j in progress at any time.

c. Machines. No machine may process more than one operation at any time.

408. Objectives. There are a number of possible objectives for the scheduling process, that include:

a. Minimise total throughput, i.e. minimise the sum of the individual job durations, where the duration of a job is the time between the commencement of its earliest operation and the completion of its latest operation.

b. Minimise makespan, i.e. minimise the time by which all operations of all jobs are complete.

c. Minimise total lateness or lateness penalty, in the case where due dates are specified for the jobs.

409. Comparison with project scheduling. The general job shop scheduling problem is a special case of the multi-project scheduling problem with multiple resources; corresponding terms from the two problem areas are shown in Table 4.3. The general job shop is a multi-project with (M + N) resources, of each of which one unit is available in each time period: M of the resources correspond to the M machines, and are required to satisfy the constraints of para 407.c; the remaining N resources correspond to the N jobs, and are artificial resources required to satisfy the constraints of para 407.b. (If each operation on machine m requires one unit of resource m, and only one unit of m is available, then no more than one operation on machine m may be in progress at any time, and 407.c is satisfied; if each operation of job j requires one unit of resource j, and only one unit of j is available, then only one operation of job j may be in progress at any time, and 407.b is satisfied.) In the frequently occurring special case in which each operation has at most one immediate predecessor and at most one immediate successor, the N artificial resources corresponding to the jobs j can be omitted.

410. References. The job shop scheduling problem is treated widely in the literature. Conway, Maxwell and Miller (1967) provide a detailed discussion; Bennington and McGinnis (1972) outline exact solution procedures for the problem in relation to the project

	MULTI-PROJECT SCHEDULING	JOB SHOP SCHEDULING
1	Project	Job j
2	Job, activity i	Operation mjk
3	Job duration d(i)	Operation time d(mjk)
4	Resource requirements	Machines m
5	Artificial resource requirements	Jobs j

Table 4.3 Corresponding concepts in multi-project scheduling and job shop scheduling.

scheduling problem. Similar problems also appear in degenerate form: Thesen (1973), for example, describes the scheduling of computer programs (activities) with multiple resources (core storage, tape units, central processor usage) and very few precedence constraints (i.e. the programs are generally independent of one another).

PART 2. THE RESOURCE-CONSTRAINED PROJECT SCHEDULING PROBLEM

CHAPTER 5. INTRODUCTION TO THE RESOURCE-CONSTRAINED PROBLEM

CHAPTER 5. INTRODUCTION TO THE RESOURCE-CONSTRAINED PROBLEM

501. In this part, we shall consider the problem of scheduling a project with limited resources, the objective being to minimise the duration of the project. Various scheduling methods will be discussed, and the results of an experimental comparison of some of them will be given.

502. We shall initially formulate the problem, and give some preliminary definitions. After discussing some exact solution methods, and concluding that they are impractical for all but the smallest problems, we shall examine some heuristic procedures for obtaining approximate solutions. Next, we describe in detail an experiment designed to test various aspects of the heuristics. Finally, results of the experiment are given, and conclusions are presented.

Formulation

503. There is a partially ordered set A of N jobs, or activities. With each job i in A there is associated:

a. A non-negative duration, d(i).

b. A non-negative requirement for r(i,k) units of resource k in each time period in which job i is in progress, for k=1,2,...,K, where K is the number of distinct kinds of resources used by the project.

c. A set of immediate successor jobs, Γi.

d. A start time, s(i), to be determined.

504. It is required to find start times, s(i), such that the following constraints are satisfied:

a. Precedence. For all jobs i,

$s(i) \geqslant 0$, if $\Gamma^{-1}i=\emptyset$,

$s(i) \geqslant s(j)+ d(j)$, for all $j\varepsilon\Gamma^{-1}i$.

b. Non-preemption. A job, once started, cannot be interrupted, and must run to completion.

c. Limited resources. The demand for resources in any time period must not exceed the availability. Formally,

$\Sigma r(i,k) \leqslant R(k,t)$, for k=1,2,..,K,

and t=0,1,..,PHOR,

where the sum is over all jobs i in progress at time t,

i.e. jobs i with

$s(i) \leqslant t < s(i)+d(i)$.

505. We wish to determine starting times, satisfying the above constraints, which minimise project duration, PDUR, where

$$PDUR = \underset{i\varepsilon A}{Max} \{s(i)+d(i)\}.$$

506. In this part of the thesis, we shall assume that the availability of resource k is constant in all time periods, and we shall represent it by R(k), for all resources k. Thus we have

$R(k,t) = R(k)$, for all time periods t.

(This is not an unreasonable assumption, since such resources as skilled labour, for example, will generally be permanent employees, at least for the duration of the contract; even casual labour levels may not be permitted to vary from period to period, because of practical requirements for continuity of employment. Where resource availabilities can be freely adjusted to match the resource requirements of the project, by such means as hiring or laying off workers, or hiring equipment, then we do not have a resource-constrained problem; in this situation we may be more interested in reducing the fluctuations in demand for resources, or some other resource-based objective similar to those listed in paras 321 to 323.)

Schedules and Permutations

507. We shall now define some of the terms which will be used in this part of the thesis. Many of these definitions follow Schrage (1970); however, Giffler and Thompson (1960) give equivalent definitions, and similar ones are used by Held, Karp and Shareshian (1963).

508. Any set of start times $\{s(i) \mid i \in A\}$ which satisfy the constraints of para 504 is called a feasible schedule. In what follows, we shall be concerned only with feasible schedules, and all schedules will be understood to be feasible.

509. A partial schedule is a specification of start times for some subset (not necessarily proper) of the activities A, that does not violate the constraints, such that if any job has a start time associated with it in the partial schedule, so have all the predecessors of that job. A partial schedule V, with associated start times s(i,V), is said to be active if there does not exist another partial schedule W of the same jobs, with associated start times s(i,W), together with an activity j such that

$$s(i,V) = s(i,W), \text{ for } i \neq j, \text{ and}$$

$$s(j,V) > s(j,W).$$

That is, a schedule is not active if there exists some activity that can be started earlier without changing the start times of any other activities and without violating the constraints. Most of the partial schedules with which we will be concerned will be active.

Example 5.1. Consider the project of Fig 5.1. Table 5.1 shows project data, together with:

a. A feasible schedule for this project (col 4).

b. A non-active partial schedule of jobs 1, 2 and 3 (col 5).

c. An active partial schedule of jobs 1, 2 and 3 (col 6), derived from that of col 5 by starting job 2 at time 2 instead of time 3.

Fig. 5.2 shows resource profiles for these partial schedules.

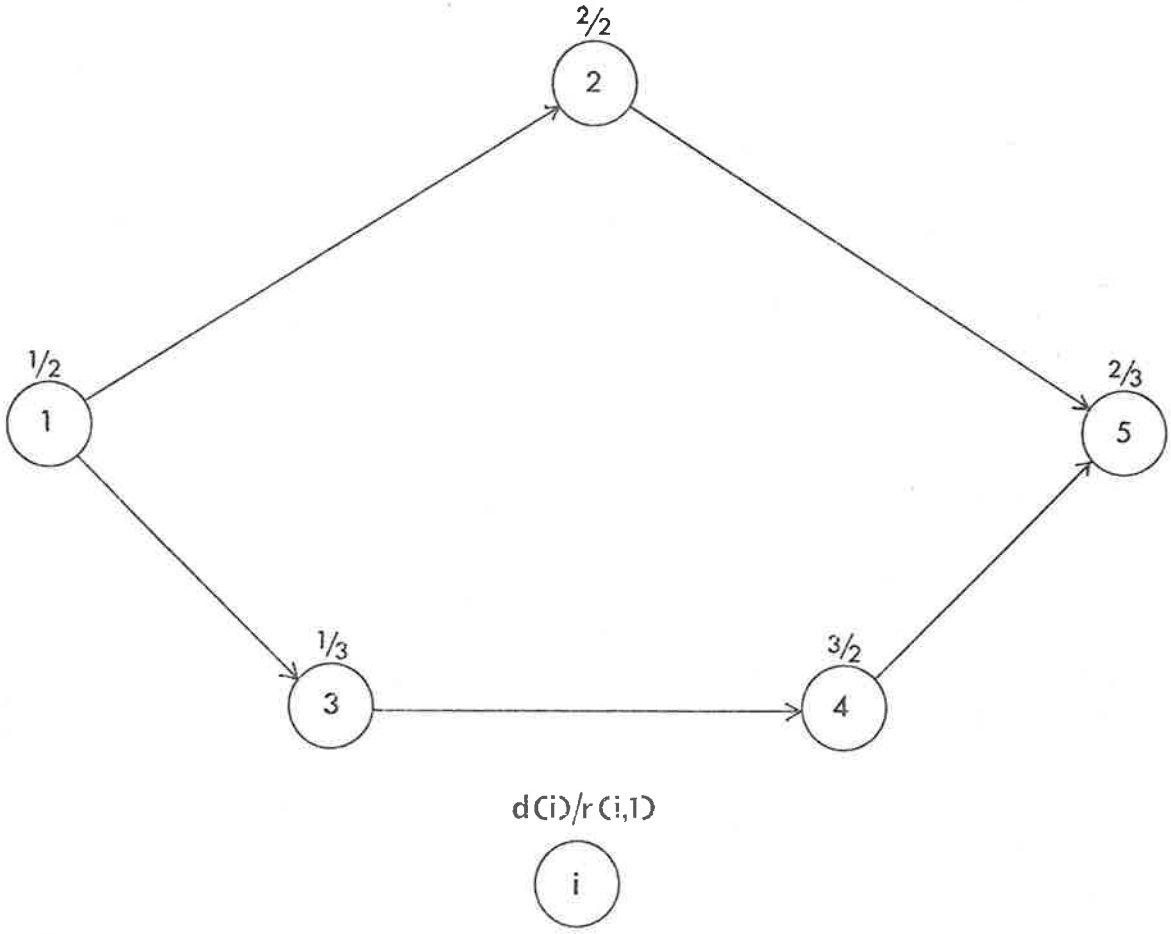

Fig 5.1. The graph of a project for which R(1) = 3.

1	2	3	4	5	6
JOB i	d(i)	r(i,1)	FEASIBLE SCHEDULE	PARTIAL SCHEDULE	PARTIAL SCHEDULE
1	1	2	0	0	0
2	2	2	2	3	2
3	1	3	1	1	1
4	3	2	4		
5	2	3	7		

Table 5.1. Project data and schedules for the project of Fig 5.1.

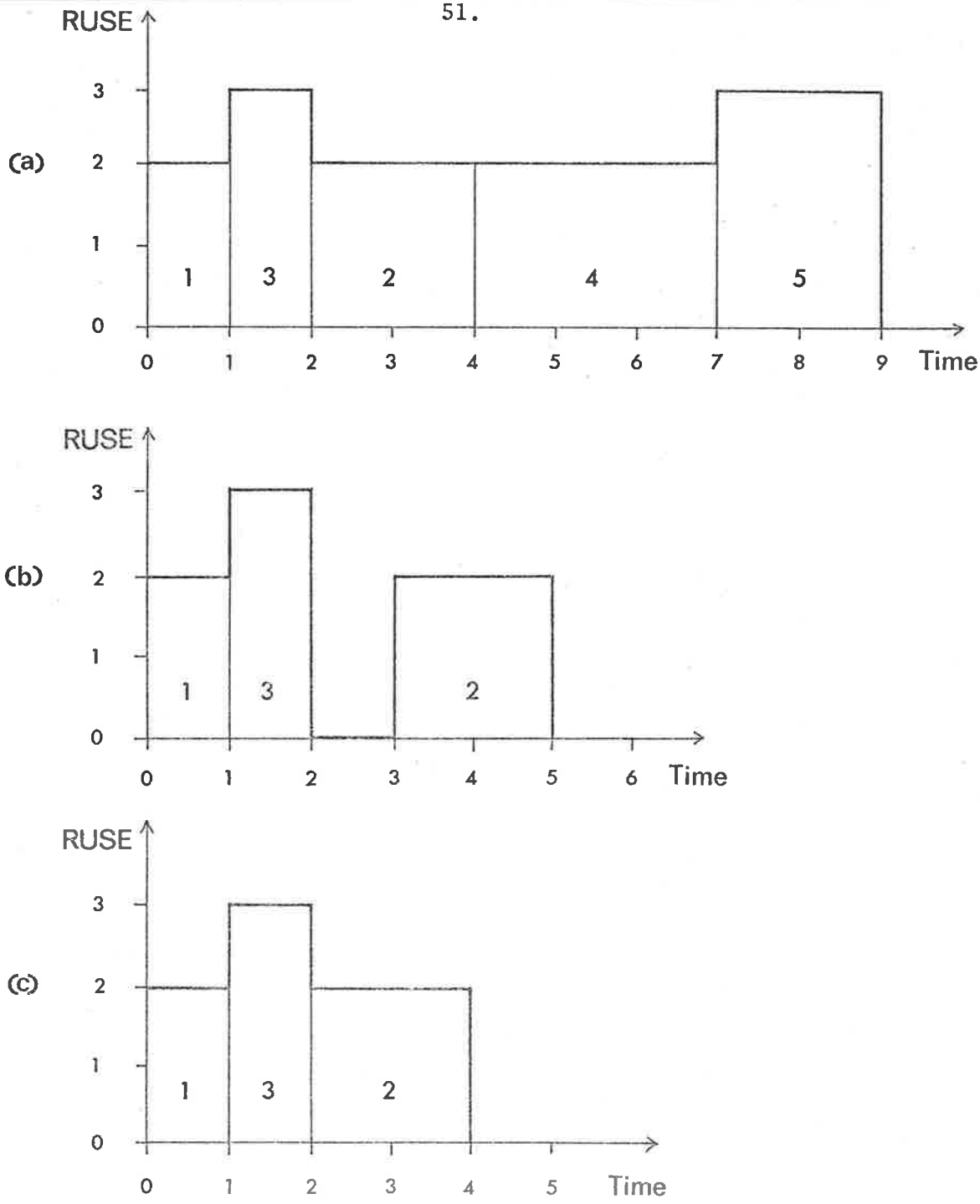

Fig 5.2. Resource profiles for the schedules of Table 5.1:

a. Feasible schedule, col 4.

b. Non-active partial schedule, col 5.

c. Active partial schedule, col 6.

510. Suppose that start times are assigned to jobs in a subset X of A by some partial schedule. Then, from the requirement of a partial schedule to contain all predecessors of any job in that partial schedule, we have

$$\text{whenever } (i \in X), \text{ then } (j \in X, \text{ for all } j \in \Gamma^{-1} i).$$

Conversely, if we are adding jobs to the partial schedule one at a time, the job i to be added next must be such that

$$(i \notin X) \text{ and } (j \in X, \text{ for all } j \in \Gamma^{-1} i).$$

That is, given a partial schedule of a subset X of A, the set AV of jobs available for scheduling next is that set of jobs, not in X, which have all their predecessors in X, i.e.

$$AV = \{ i \mid i \notin X, \{(j \in X, \text{ for all } j \in \Gamma^{-1} i)\}\}.$$

Example 5.2. Consider the project of Fig 5.1, and an active partial schedule in which job 1 has start time 0, and no other jobs have start times assigned. (In terms of para 510, X = {1}.) Then, AV = {2,3}.

511. Given a partial schedule, and an available set AV of jobs which can be added to the partial schedule, we define $\sigma(i)$ to be the earliest time at which job i in AV can start, without violating the constraints. Thus $\sigma(i)$ is the earliest start time of a job, taking into consideration the resource requirements of the partial schedule. This differs from the critical path early start time, $\ell(i)$, which takes no account of resources; in fact, $\sigma(i) \geq \ell(i)$, for

all $i \varepsilon AV$.

Example 5.3. Consider the project of Fig 5.1, and the active partial schedule of col 6 in Table 5.1. Then AV = {4}, and we have

$$\ell(4) = s(3)+d(3) = 2, \text{ since } \Gamma^{-1}4 = \{3\},$$

but

$$\sigma(4) = 4, \text{ due to resource restrictions.}$$

512. Suppose we have a partial schedule, and a set AV of available jobs. By the expression schedule activity i we mean, for a job i in AV, to determine its earliest start time $\sigma(i)$, and add activity i to the partial schedule with start time $s(i)=\sigma(i)$. If the original partial schedule was active, the new partial schedule containing i is also active.

513. A partial permutation is a permutation of some subset of the integers 1,2,..,N, where N is the number of jobs in A. A feasible partial permutation is a partial permutation for which, if i is a member, and j is a predecessor of i, then j is also a member, and j appears before i in the partial permutation. That is, if Y is a feasible partial permutation,

whenever $(i \varepsilon Y)$, then $(j \varepsilon Y$, for all $j \varepsilon \Gamma^{-1} i)$.

(Note that we have established a correspondence between jobs in A and the integers 1,2,..,N by our assumption of topological ordering in para 303.d). The expression above corresponds with the condition

for partial schedules, given in para 510; similarly, jobs i which can be added to a feasible partial permutation Y so that the resulting partial permutation is also feasible are just those jobs in the available set AV, satisfying

$$(i \notin Y) \text{ and } (j \in Y, \text{ for all } j \in \Gamma^{-1} i).$$

The correspondence between partial schedules and partial permutations is made more explicit by the following theorem of Schrage (1970, p265).

Theorem. "Consider an active partial schedule with associated start times s(i), and the partial permutation obtained by ordering the indices of the activities in the schedule lexicographically by the vector [s(i),i]. If the activities are scheduled in order of their appearance in the partial permutation, the partial schedule that generated the partial permutation is obtained. Thus, the partial permutation is a compact means of representing the partial schedule."

Example 5.4. Consider the project of Fig 5.1, and the partial schedule in col 6 of Table 5.1. If activities are ordered lexicographically by the vector [s(i),i], the ordering (1,3,2) is obtained. Scheduling jobs in the order 1 then 3 then 2, we have:

a. Job 1. $\sigma(1) = 0$. $s(1) = \sigma(1) = 0$.

b. Job 3. $\sigma(3) = 1$. $s(3) = \sigma(3) = 1$.

c. Job 2. $\sigma(2) = 2$ ($\neq \ell(2) = 1$, due to resource restrictions). $s(2) = \sigma(2) = 2$.

Thus, scheduling jobs in the order (1,3,2) gives the original active partial schedule.

514. For a finite set of activities, the set of feasible partial permutations, and hence the set of active schedules, is finite. For any objective function that is a non-decreasing function of the start times {s(i)}, there always exists an optimal schedule that is active.

Solution Methods

515. There exists a variety of solution procedures for the resource-constrained project scheduling problem, many of which are adaptations of algorithms developed in related problem areas, most notably assembly line balancing and job shop scheduling. Methods for obtaining an exact solution to the problem are of limited practical usefulness; Bennington and McGinnis (1972), in an excellent review of exact methods, state that "without exception, every [exact] solution procedure proposed has been computationally impractical, either because the model grows too large, or because the solution procedure is too lengthy, or both", and "heuristic methods [are] still the only viable solution techniques".

516. Despite their limitations, exact procedures provide useful insights into the structure of the problem, and several approaches will be discussed briefly in later sections. They can be loosely grouped into three classes:

a. Methods which formulate the problem in terms of linear inequalities.

b. Methods which start with a complete schedule, and progressively modify it.

c. Methods which construct a schedule by means of a sequence of partial schedules.

The latter two classes of methods deal directly with the combinatorial problem; that is, they are essentially enumerative techniques.

517. Heuristic methods cover a wide range, from simple hand-scheduling rules to sampling techniques and 'learning' algorithms. The procedures to be considered in this thesis can be grouped into:

a. Methods which select activities to be scheduled according to a predetermined rule, based on properties of the project and its activities, and on the state of the schedule so far.

b. Methods which, in some fashion, form a sample from the set of feasible schedules, and select the best schedule obtained.

518. The relationships between the various solution methods are shown in Fig 5.3. A detailed description of the different kinds of sampling methods that can be used will be left to a later section.

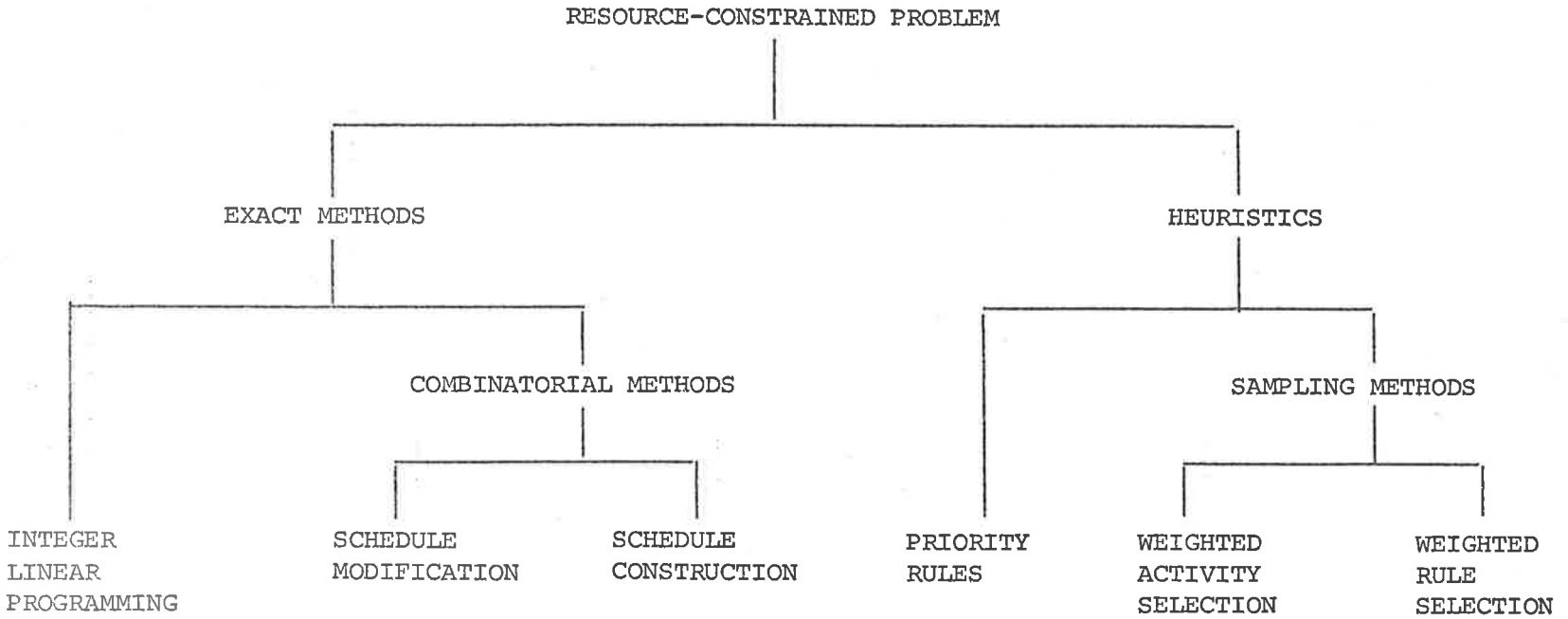

Fig. 5.3 Solution Methods.

Integer Linear Programming Procedures

519. There have been a number of integer linear programming (ILP) formulations of the job shop scheduling problem (principally, Bowman (1959), Wagner (1959), and Manne (1960)), but they were found to be impractical. Pritsker, Watters and Wolfe (1969) have developed a 0-1 ILP model for multi-project scheduling with limited resources which permits great flexibility in problem specification; it allows for job splitting (i.e. selective violation of the non-preemption constraint of para 504.b), due dates, substitution of resources, concurrent and non-concurrent activities, and a variety of objective functions (minimise total throughput time for all projects, minimise makespan, minimise toal lateness or lateness penalty).

520. In the case of a single project, with the constraints of para 504, the model has the form shown in Fig 5.4. The objective (1) is to minimise project duration. Constraints (2) require that every activity be completed. Constraints (3) define the x_t variables. Constraints (4) are the precedence constraints, and constraints (5) are the resource constraints. The number x_{it} variables can be reduced by noting that x_{it} is zero if t is less than the CP early finish time, $a(i)=\ell(i)+d(i)-1$, or greater than the CP late finish time, $b(i)=u(i)+d(i)-1$

521. Despite the flexibility of this model, its use is limited by its size. Pritsker et al give an example problem consisting of three projects with a total of eight activities and using three kinds

$$\text{Maximise} \quad z = \sum_{t = CPDUR}^{PHOR} x_t \qquad (1)$$

Subject to:

$$\sum_{t=a(i)}^{b(i)} x_{it} = 1, \text{ for all } i \in A \qquad (2)$$

$$x_t \leqslant \frac{1}{N} \sum_{i \in A} \sum_{q=a(i)}^{t-1} x_{iq}, \text{ for } t = CPDUR,\ldots,PHOR \qquad (3)$$

$$\sum_{t=a(i)}^{b(i)} t\, x_{it} \leqslant \sum_{t=a(j)}^{b(j)} t\, x_{jt} - d(j),$$

$$\text{for all } j \in \Gamma i,\ i \in A \qquad (4)$$

$$\sum_{i \in A} \sum_{q=t}^{t+d(i)-1} r(i,k) x_{iq} \leqslant R(k,t),$$

$$\text{for } k = 1,2,\ldots,K$$

$$\text{and } t=0,1,\ldots,PHOR \qquad (5)$$

where:

x_{it} = 1 if activity i is completed in period t,

0 otherwise.

x_t = 1 if the project is completed in or before period t-1,

0 otherwise.

Figure 5.4 ILP model.

of resources; the formulation required 33 variables and 37 constraints. The number of variables and constraints increases rapidly as the project horizon, PHOR, is extended; this behaviour is evident in most ILP formulations of this nature.

Example 5.5. For the model of Fig 5.4, we have:

a. The number of x_t variables is (PHOR-CPDUR+1).

b. The number of x_{it} variables is of the order of $\Sigma f(i)$, where f(i) is the float of job i, and the sum is over all jobs i in A. Note that the float of a job is of the order of PHOR (para 306), and so there are of the order of (N*PHOR) x_{it} variables.

c. There are N job completion constraints (2).

d. There are (PHOR-CPDUR+1) project completion constraints (3).

e. There is one constraint (4) per precedence constraint.

f. There are of the order of (K*PHOR) constraints (5).

Combinatorial Methods

522. The combinatorial methods fall into two categories: schedule modification and schedule construction. The schedule modification methods have been applied to machine sequencing, where the problem is formulated in terms of shortest critical paths in graphs which have been augmented by the addition of new precedence constraints representing the different ways of overcoming the limitations on machine availability (Bennington and McGinnis, 1972, give examples

and references). This kind of formulation, however, is not as easy for the project scheduling problem, and only one algorithm of this style has been proposed for its solution (Gorenstein, 1972).

523. Schedule construction using permutations in the enumeration of all feasible active schedules was first used in the context of job shop scheduling by Giffler and Thompson (1960), and applied to the more general case of the project scheduling problem by Schrage (1970). Schrage reduces the number of schedules generated in his implicit enumeration algorithm by using dominance and redundancy criteria to delete infeasible and duplicated permutations; the size of the enumeration, however, is still large.

524. Davis and Heidorn (1970), using an algorithm based on a method by Gutjahr and Nemhauser (1964) for the assembly line balance problem, decompose each activity i into d(i) activities of unit duration, introducing a "must immediately precede" relation to prevent jobs being interrupted. The problem is transformed to one of finding a shortest route in a new "A-network" whose vertices are subsets of unit duration tasks. Various elimination criteria are used to reduce the size of the problem.

525. The enumeration of all feasible schedules is a large problem, even with the addition of dominance and other bound estimators to reduce its size. Annex C gives some examples of the size of the enumeration problem. Generally, combinatorial methods are unaccep-

table in practice because of their large size and the computation time which they require.

CHAPTER 6. HEURISTIC METHODS

CHAPTER 6. HEURISTIC METHODS

601. As we have noted in previous sections, the number of schedules that must be considered in searching for an optimum is extremely large. In most practical cases, then, we must be satisfied with good approximate solutions. We shall be concerned principally with:

a. Serial and parallel methods, procedures which select activities to be scheduled according to a predetermined rule.

b. Sampling methods, procedures which choose a sample of feasible schedules, and select the best schedule obtained.

Serial and Parallel Methods

602. Serial and parallel methods create a schedule via a series of partial schedules, according to the scheme of Fig 6.1. At any stage, this process can be explained thus:

a. Create AV. As discussed in para 510, AV is the set of unscheduled jobs with all predecessors scheduled; any one of these jobs can be scheduled next, yielding a new active partial schedule.

b. If AV is empty, then all jobs have been scheduled; exit from the procedure.

1. Enter, with empty partial schedule.
2. Create the available, set AV.
3. If AV is not empty, then go to 4. Otherwise, all jobs have been scheduled; go to 6.
4. Choose a job i from AV.
5. Schedule job i, with $s(i) = \sigma(i)$. Go to 2.
6. Exit.

Fig 6.1 Serial and parallel methods.

c. If AV is not empty, choose a job i from AV

d. Schedule the chosen job i, by the method of para 512. Then go and create a new set AV, and repeat the process.

603. The methods of choosing a job i from AV in para 602.c above are of major importance in this thesis. Each job i has associated with it a priority, P(i), calculated by some priority rule; that job which has the highest priority of all jobs in AV is chosen for scheduling.

Example 6.1 One possible priority rule is:

$$P(i) = d(i), \text{ for all } i.$$

In this case, the priority of a job is proportional to its duration, and long jobs will be chosen for scheduling in preference to short ones. (Priority rules will be discussed in detail in the next section).

604. The difference between the serial and parallel methods lies in the evaluation priorities $\{P(i)\}$. The serial methods are static: priorities are evaluated for all jobs once only, before any jobs are scheduled, and jobs retain this initial priority throughout the scheduling process. The parallel methods are dynamic: priorities are evaluated as they are required, and the priority associated with a job can change as the scheduling progresses. Pascoe (1965) has shown experimentally that parallel methods are better than serial methods, and it is in parallel methods that we shall be more interested; however,

some of the priority rules that we shall be using are static, and so the serial methods will be investigated indirectly when we apply these rules (Example 6.2, para 701).

Example 6.2. Some priority rules are static in nature, and give the same job priorities whether they are used in a serial or a parallel method. One rule of this kind has been seen in Example 6.1, i.e. $P(i) = d(i)$. Other rules can be used dynamically; consider, for example, the rule "choose the job which can start earliest", i.e.

$$P(i) = 1/(\sigma(i) + 1), \text{ for all } i \varepsilon Av,$$

where $\sigma(i)$ is the earliest time at which job i can start. (A definition of σ is given in para 511.) In the static case (serial method), the best estimate of earliest starting time of a job i is its CP early start time $\ell(i)$. In the dynamic case (parallel method), a job's earliest start time will change if its predecessors are delayed (by resource restrictions, for example), or if it is itself restricted by limited resources; this will be seen more clearly in Example 6.3. (The reciprocal of $\sigma(i)$ is used in the expression for $P(i)$, since we wish to choose the job with the minimum value of σ, but the serial and parallel methods select the job with the maximum priority. $(\sigma(i)+1)$ is used to allow for the case $\sigma(i)=0$.)

605. In both the serial and parallel methods, a second rule may be invoked to break ties; Pascoe has shown that the rule used for this

purpose is not important. As a result of applying a priority rule, and a second rule to break ties, the jobs become (implicitly) sorted in order of preference; for this reason, the effects of these two rules are often referred to in the literature as the major sort and the minor sort respectively. In the experiments with which this thesis is concerned, only one rule was used, and ties were broken by random selection from the set of tied jobs.

Example 6.3. Consider the scheduling of the project of Fig 6.2, using the rule "select the job which can start first, breaking ties in favour of jobs with short durations", i.e.

$$P(i) = a/(\sigma(i)+1) + b/d(i), \text{ for all } i,$$

where a = 60 and b = 2. In this case, with a >> b, early start is the major sort, and duration is the minor sort. (The actual values of a and b have been chosen for arithmetic convenience; the important thing is that a is significantly larger than b, so that incremental changes in $\sigma(i)$ have greater effect on P(i) than the largest possible variation in d(i).) Table 6.1 shows relevant data for the project, including the priorities to be used by the serial method; these priorities are evaluated before the scheduling process commences. Note that in the serial method, $\sigma(i) = \ell(i)$, the CP early start time.

a. Serial method. Table 6.2 shows the stages in scheduling the project using the serial method. The schedule obtained can be described by the permutation (1,2,3,4,5,6).

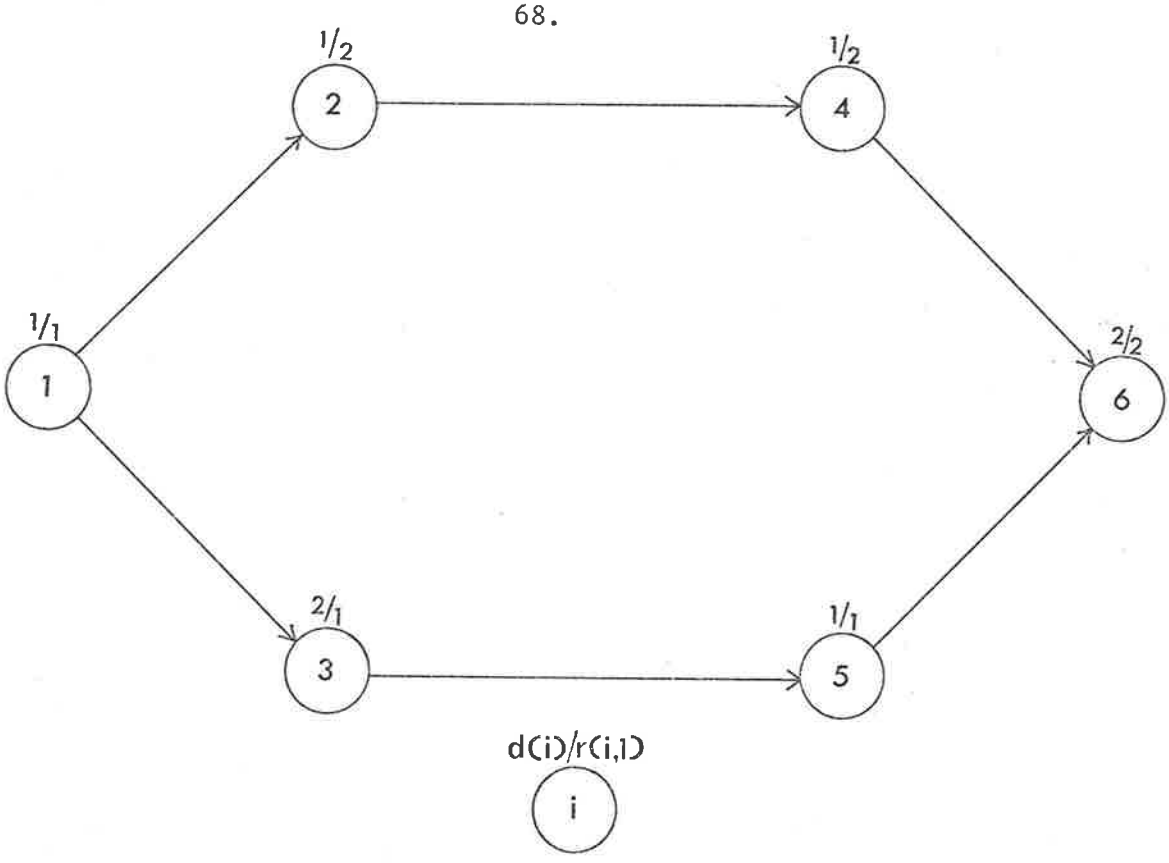

Fig 6.2. The graph of a project with R(1) = 2.

i	d(i)	r(i,1)	ℓ(i)	P(i)
1	1	1	0	62
2	1	2	1	32
3	2	1	1	31
4	1	2	2	22
5	1	1	3	17
6	2	2	4	13

Table 6.1 Project data, and job priorities for the serial method.

STAGE	AV	PRIORITIES	SELECTED JOB, j	SCHEDULED START, s(j)
1	{1}	P(1)=62	1	0
2	{2,3}	P(2)=32		
		P(3)=31	2	1
3	{3,4}	P(3)=31		
		P(4)=22	3	2
4	{4,5}	P(4)=22		
		P(5)=17	4	4
5	{5}	P(5)=17	5	5
6	{6}	P(6)=13	6	6
7	ø			

Table 6.2 Stages in scheduling the project using a serial method.

b. Parallel method. Table 6.3 shows the stages in scheduling the project using the parallel method. Note that priorities change as the schedule progresses (particularly the priority of job 3), and this affects the schedule obtained. The schedule can be described by the permutation (1,2,4,3,5,6).

c. Table 6.4 summarises the schedules obtained by the two methods. Resource profiles are shown in Fig 6.3. Referring to Fig 6.3(b), it can be seen that resource restrictions cause the start times $\sigma(3)$ of job 3 to become progressively later in the parallel method.

Priority Rules

606. As noted in the previous section, available jobs are selected for scheduling on the basis of a calculated priority. Many different priority rules have been used for this purpose at some time or other, in the fields of project scheduling, assembly line balancing, and job shop scheduling. Table 6.5 lists some of the priority rules that can be used; they will be identified either by their number or by their name, as shown in the table. The terms and symbols used in the table are explained in the next paragraph.

607. Rules have been classified by properties of the project on which they depend. The classification used here is based on:

a. Time properties of the project and its jobs (T).

STAGE	AV	EARLY START	PRIORITIES	SELECTED JOB,j	SCHEDULED START,s(j)
1	{1}	$\sigma(1)=0$	P(1)=62	1	0
2	{2,3}	$\sigma(2)=1$	P(2)=32		
		$\sigma(3)=1$	P(3)=31	2	1
3	{3,4}	$\sigma(3)=2$	P(3)=21		
		$\sigma(4)=2$	P(4)=22	4	2
4	{3}	$\sigma(3)=3$	P(3)=16	3	3
5	{5}	$\sigma(5)=5$	P(5)=12	5	5
6	{6}	$\sigma(6)=6$	P(6)=67/7	6	6
7	$\emptyset$				

Table 6.3 Stages in scheduling the project using a parallel method.

i	s(i)	
	SERIAL	PARALLEL
1	0	0
2	1	1
3	2	3
4	4	2
5	5	5
6	6	6

Table 6.4 Schedules obtained by serial and parallel methods.

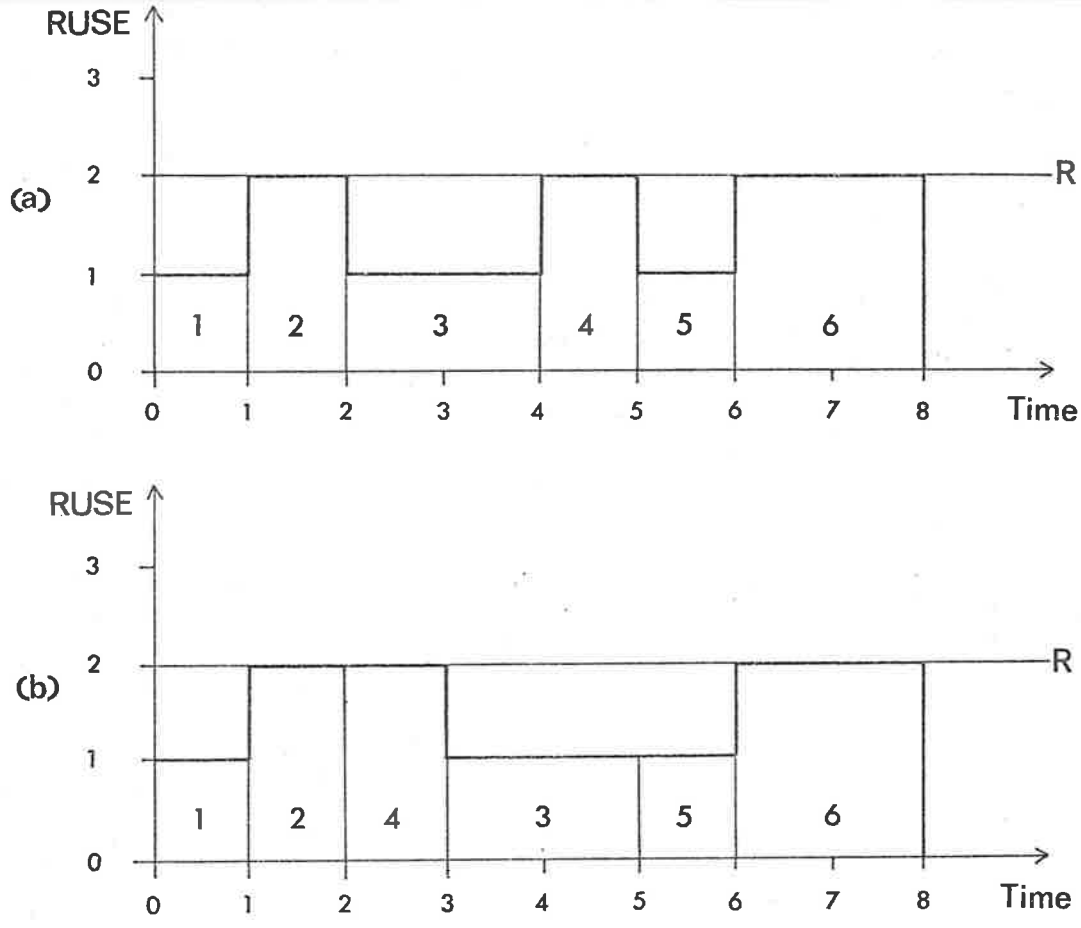

Fig 6.3. Resource profiles:

a. Serial method.

b. Parallel method.

Table 6.5 Priority rules. (Refer to paras 606 to 634.)

RULE NO.	RULE NAME	DESCRIPTION: Select the job which:	DEFINITION: P(i) proportional to:	CLASSIFICATION		REFER TO PARAS:
1	FCFS	Has the earliest CP early start time.	$1/(\ell(i)+1)$	TL	Dynamic	608,610
2	LS	Has the earliest CP late start time.	$1/(u(i)+1)$	TL	Static	609
3	EF	Has the earliest CP early finish time.	$1/(\ell(i)+d(i)+1)$	TL	Dynamic	610
4	LF	Has the earliest CP late finish time.	$1/(u(i)+d(i)+1)$	TL	Static	611
5	FLOAT	Has the least float.	$1/(f(i)+1)$	TL	Dynamic	612
6	SPT	Has the shortest duration.	$1/(d(i)+1)$	TL	Static	613
7	LPT	Has the longest duration.	$d(i)+1$	TL	Static	614
8	NIMSUCC	Has most immediate successors.	$\#(\Gamma i)+1$	SL	Static	615
9	NEXTAV	Will cause AV to be largest at the next stage.	#(Next AV)+1	SL	Dynamic	616
10	RANDOM		Random variable			617

Table 6.5 (contd.)

RULE NO.	RULE NAME	DESCRIPTION: Select the job which:	DEFINITION: P(i) proportional to:	CLASSIFICATION		REFER TO PARAS:
11	ALLSUCC	Has most successors	$\#(\Gamma^{*}i)+1$	SG	Static	618
12	NONREL	Has least non-related jobs	$1/(N-\#(\Gamma^{*}i)-\#(\Gamma^{-*}i))$	SG	Static	619
13	NONDEP	Has least non-dependent jobs remaining	1/(Total unscheduled jobs-$\#(\Gamma^{*}i)$)	SG	Dynamic	620
14	SUCC/LEV	Has most successors per level	$(\#(\Gamma^{*}i)+1)/\text{level}(i)$	SG	Static	621
15	RPOSWT	Has the greatest rank positional weight	RPOSWT(i)+1	TG	Static	622
16	PROD	Has the greatest product of rules 7, 11, 13, 14 and 15		TSG	Dynamic	623
17	RED	Has the greatest RED	RED(i)+1	TRL	Static	625
18	CUMRED	Has the greatest CUMRED	CUMRED(i)+1	TRG	Static	626
19	1/RPOSWT	Has the smallest rank positional weight	1/(RPOSWT(i)+1)	TG	Static	627
20	PRPOSWT	Has the greatest proportional RPOSWT	(RPOSWT(i)+1)/(d(i)+1)	TG	Static	628
21	1/NIMSUCC	Has least immediate successors	$1/(\#(\Gamma i)+1)$	SL	Static	629

Table 6.5 (contd.)

RULE NO.	RULE NAME	DESCRIPTION Select the job which:	DEFINITION P(i) proportional to:	CLASSIFICATION		REFER TO PARAS:
22	1/ALLSUCC	Has least successors.	$1/(\#(\Gamma^{*}i)+1)$	SG	Static	629
23	FIFS	Can start first.	$1/(\sigma(i)+1)$	TRL	Dynamic	630
24	FLOAT/SUC	Has least float per successor	$(\#(\Gamma^{*}i)+1)/(f(i)+1)$	TSG	Dynamic	631
25	LEVELS	Has the greatest level	level (i)	SG	Static	632
26	FIRSTOUT	Can finish first	$1/(\sigma(i)+d(i)+1)$	TRL	Dynamic	633

b. The structure of the project graph (S).

c. Resource requirements (R).

This classification can be further subdivided, according to whether or not the rules are:

d. Local in effect (L); that is, they depend on properties of the job alone, or on the job and its immediate successors and predecessors.

e. Global in effect (G); that is, they depend on the relationship of the job to the rest of the project.

The static or dynamic natures of the rules have also been indicated in Table 6.5. The meanings of the classifications will become clearer when we examine in detail the definitions of the priority rules.

608. Rule 1, FCFS. This rule selects the job with the earliest CP early start time. This is the job which could start first if the resource constraints did not have to be satisfied. (In the terminology of job shop scheduling, this rule selects the job with the earliest arrival time, i.e. first-come-first served, FCFS). This rule depends on the time properties of the job and its immediate predecessors, and is classified TL. We shall discuss this rule later, in particular its dynamic aspects, when we examine rule 3(EF), and rule 23 (FIFS).

609. Rule 2, LS. This rule selects the job with the earliest

CP late start time. It is based on local time properties (TL), and is a static rule.

610. Rule 3, EF. This rule selects the job with the earliest CP early finish time. It is based on local time properties (TL). It should be noted that in the calculation of the CP early start time used in this rule and in rule 1, the early start time of an immediate predecessor j of a job i in the available set AV is its scheduled start time s(j). Thus rules 1 and 3 are made dynamic by changing the definition of CP early start time from that given in para 304 to:

$$\ell(i) = s(i) \text{ if } i \text{ has been scheduled, else}$$

$$0 \quad \text{if } \Gamma^{-1}i = \emptyset, \text{ else}$$

$$\max_{j \in \Gamma^{-1}i} \{\ell(j)+d(j)\} .$$

611. Rule 4, LF. This rule selects the job with the minimum CP early finish time. It is based on local time properties (TL), and is static. This rule was found by Pascoe to be a very good rule.

612. Rule 5, FLOAT. This rule selects the job with the least float, i.e. the most critical jobs will be done first. This rule is based on local time properties (TL). It is a dynamic rule, since $f(i) = u(i)-\ell(i)$, with $\ell(i)$ calculated by the definition in para 610 above.

613. Rule 6, SPT. This rule selects the job with the shortest processing time, i.e. minimum duration. Battersby (1967) suggests that

doing the shortest jobs first will give the earliest opportunity for revising a decision, and that this rule is more appropriate to a minor sort than to a major sort. Rule 6 is classified TL.

614. Rule 7, LPT. This rule selects the job with the longest processing time, i.e. maximum duration, on the basis that long jobs are more difficult to schedule than short ones, which can easily be fitted in later. It is classified TL, and is a static rule.

615. Rule 8, NIMSUCC. This rule selects the job with the greatest number of immediate successors on the basis that this will expand the available set AV, and so make scheduling easier. It depends on the local structure of the project graph (SL), and is a static rule.

616. Rule 9, NEXTAV. This rule selects the job which will cause the available set AV to be largest at the next stage (cf. rule 8). It is classified SL, as it depends on the local structure of the project graph, but unlike rule 8 it is dynamic.

Example 6.4. Fig 6.4 shows a portion of a project graph. Suppose in the scheduling of this project, using a parallel method, all predecessors of jobs a, b, and c have been scheduled, and AV = {a,b,c}. Table 6.6 shows the priorities calculated by rules 8 and 9 for these jobs; using rule 8, jobs are sorted in order of preference b, then c, then a; using rule 9, the order of preference is c, then a, then b. If rule c

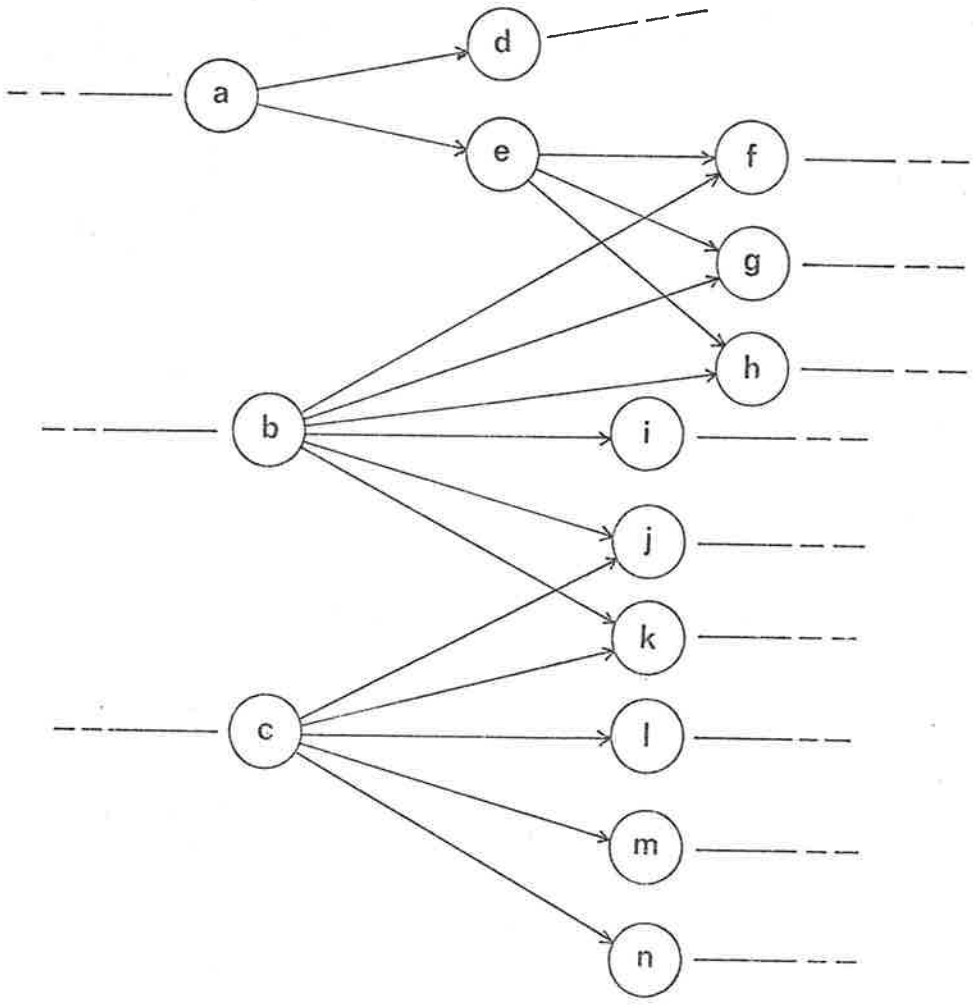

Fig 6.4. A portion of a project graph.

JOB	IMMEDIATE SUCCESSORS	PRIORITY (RULE 8)	AV AT NEXT STAGE	PRIORITY (RULE 9)
a	d,e	3	b,c,d,e	5
b	f,g,h,i,j,k	7	a,c,i	4
c	j,k,l,m,n	6	a,b,l,m,n	6

Table 6.6 Priorities calculated by

rule 8 ($P(i) = \#(\Gamma i)+1$) and

rule 9 ($P(i) = \#(\text{Next AV}) + 1$).

is scheduled next, then the priorities of jobs a and b become 4 and 5 respectively (Table 6.7), i.e. job b is now preferred to job a; the priorities calculated by rule 8, on the other hand, do not change as the schedule progresses.

617. Rule 10, RANDOM. This rule allocates priorities at random, and may be used as a benchmark.

618. Rule 11, ALLSUCC. This rule selects the job with the greatest number of all successors. It depends on the relationship of the job to the rest of the project graph, and is classfied SG. It is a static rule, and is the global equivalent of rule 8, NIMSUCC.

619. Rule 12, NONREL. This rule selects the job with the least number of non-related jobs. (Two jobs are non-related if there is no precedence relationship between them.) This rule is static, global and structure dependent (SG), and was used by Arcus (1966) in a different context for assembly line balancing.

620. Rule 13, NONDEP. This rule selects the job with the least number of non-dependent jobs remaining. (Job j depends on job i if $j \varepsilon \Gamma^{*} i$.) Like rule 12, it was used by Arcus for assembly line balancing.

621. Rule 14, SUCC/LEV. This rule selects the job with the greatest average number of successors per level of following jobs, and is related to rule 11 (ALLSUC) and rule 25 (LEVELS). The level of job i is defined to be the number of vertices in the longest path [i,...]

JOB	AV AT NEXT STAGE	PRIORITY (RULE 9)
a	b,d,e	4
b	a,i,j,k	5

Table 6.7 Priorities of jobs a and b, using rule 9, after job c has been scheduled.

in (A,Γ); in the degenerate case where Γi = ∅, the level of i is 1, corresponding to the 'path' [i].

Example 6.5. In the project shown in Fig 3.1, jobs have the levels indicated in Table 6.8.

622. Rule 15, RPOSWT. This rule selects the job with the greatest rank positional weight, defined for a job i by

$$RPOSWT(i) = d(i) + \Sigma d(j),$$

where the sum is over all successors j of i. In the case where the resource requirements of jobs are much the same, RPOSWT(i) is a fair measure of the remaining work content dependent on job i. This rule is the global equivalent of rule 7 (LPT), and is classified TG; like rule 7, it is a static rule.

623. Rule 16,PROD. This rule selects the job with the greatest product of the priorities calculated by rules 7 (LPT), 11 (ALLSUCC), 13 (NONDEP), 14 (SUCC/LEV), and 15 (RPOSWT). It is included here because Arcus found that it produced very good results in his heuristic for the assembly line balancing problem.

624. Resource usage duration, RUD(k). Rules 17 and 18 were developed by Pascoe (1965); they both depend on a quantity which Pascoe called the resource usage duration of resource k, RUD(k), and which is in fact a coarse estimate of a lower bound on the project duration based on resource k. RUD(k) is the minimum time span in which the project can be completed, considering only the requirements for and

JOB i	LONGEST PATH [i,....]	LEVEL level (i)
1	[1,3,4,5,7]	5
2	[2,5,7]	3
3	[3,4,5,7]	4
4	[4,5,7]	3
5	[5,7]	2
6	[6,7]	2
7	[7]	1

Table 6.8 The levels of jobs in the project of Fig 3.1.

availability of resource k; it is calculated as follows:

a. Find the total excess requirement E(k) for resource k over the availability R(k), when all jobs are scheduled to start at their CP early start times. (This corresponds to the shaded area above the line R* in Fig 3.3, when the profile is that of resource k for the CP early start schedule, and R* is the availability R(k)).

b. Find the last time period L(k) in which there is a non-zero requirement for resource k in the CP early start schedule.

c. Then set

$$RUD(k) = L(k) + E(k)/R(k).$$

Values of RUD are computed for all resources; the maximum value of RUD(k) over all values of k is called RUDMAX.

625. Rule 17, RED. This rule selects the job with the greatest resource equivalent duration, defined for a job i by

$$RED(i) = \Sigma \frac{r(i,k)}{R(k)} \cdot \frac{RUD(k)}{RUDMAX} \cdot d(i),$$

where the sum is over all resources k = 1,2,...,K. RED(i) measures the local importance of a job i in terms of its proportional requirement for resources r(i,k)/R(k), the relative importance of these resources RUD(k)/RUDMAX, and its duration d(i). It is classified TRL.

626. Rule 18, CUMRED. This rule selects the job with the greatest cumulative RED, defined for job i, by

$$CUMRED(i) = RED(i) + \Sigma RED(j),$$

where the sum is over all successors j of i. This rule is the global equivalent of rule 17 (RED), and bears the same relationship to rule 17 as rule 15 (RPOSWT) bears to rule 7 (LPT). Like rule 17, this is a static rule.

627. Rule 19, 1/RPOSWT. This rule selects the job with the smallest rank positional weight. This rule has been included because it is the inverse of rule 15 (RPOSWT), and bears the same relationship to rule 15 as rule 6 (SPT) bears to rule 7 (LPT). Several other rules (21 and 22) have been included for similar reasons.

628. Rule 20, PRPOSWT. This rule, which is another variation on rule 15, selects the job with the greatest proportional RPOSWT, defined for job i by

$$PRPOSWT(i) = RPOSWT(i)/d(i).$$

Like rule 15, it is static, global and time-dependent.

629. Rule 21, 1/NIMSUCC; Rule 22, 1/ALLSUCC. These rules are the inverses of rules 8 and 11 respectively; refer to the comments on rule 19 in para 627.

630. Rule 23, FIFS. This rule selects the job which can start first, considering resource limitations (first-in-first-served). This is a local rule, but it is based on dynamically changing information, and uses most of the available local information about resources. Rule 1 (FCFS) is a similar rule, but it uses only time information.

(Refer to the definition of σ in para 511.)

631. Rule 24, FLOAT/SUC. This rule selects the job with the smallest float per successor. Suppose job i has float f(i). Then there is a path in (A,Γ), containing i, such that all jobs j in the path have $f(j) \leqslant f(i)$ (Frizelle, 1968). Delaying job i by 1 time period will, in general, reduce the float available to those of its successors k which have $f(k) = f(i)$, and larger delays may have correspondingly wider effects. Thus the float of a job i is not related to the job alone, but at least to that path in (A,Γ), containing i, with jobs j having $f(j) = f(i)$, and possibly to other jobs if the proposed reduction in available float is large. Float per successor is a better measure of the flexibility available to a job than is the float. This is a dynamic rule, based on global time and structure properties of the project, and it is classified TSG.

632. Rule 25, LEVELS. This rule selects the job with the greatest number of levels of following jobs; the definition of the level of a job is given in para 621. It is a static rule, based on the global structure of the project graph (SG).

633. Rule 26, FIRSTOUT. This rule selects the job which will finish first, considering the resource limitations. This rule bears a similar relationship to rule 23 (FIFS) as rule 3(EF) bears to rule 1 (FCFS).

634. Summary. Many of the rules used have been included because they have been tried as scheduling rules before, or because their inverses have been used. Table 6.9 lists some references to various priority rules in the literature; these are examples only, and the list is not claimed to be in any way comprehensive. The references have been classified according to their application area:

a. Project scheduling (P).

b. Assembly line balancing (A).

c. Job shop scheduling (J).

Sampling Methods

635. The solution of the project scheduling problem can be viewed combinatorily as a search for an element of the set of optimal schedules, itself a subset of the set of all feasible schedules. (The set of optimal schedules may consist of just one optimal schedule, or there may be more than one schedule that is optimal.) The serial and parallel methods terminate their search for an element of this optimal subset after just one feasible schedule has been generated. Sampling methods, on the other hand, use randomising techniques to select a set of feasible schedules, the best of which becomes an approximation to the optimum.

636. The general scheme of the sampling methods is shown in Fig 6.5. The stage "Form a feasible schedule S_j" is similar to the serial and parallel methods (Fig 6.1), and, as in those methods, the selection of an available job from the set AV to be scheduled

Table 6.9 References to priority rules. (Refer to para 634 and the notes at the end of the table.)

	RULE	REFERENCES
1	FCFS	HL(1962)(J), FT(1963)(J), Pascoe (1965)(P), Tonge (1965)(A), CMM(1967)(J), Wiest(1967)(P), Fendley(1968)(P), Karni(1973)(P).
2	LS	Pascoe(P), Karni(P).
3	EF	Pascoe(P).
4	LF	Verhines(1963)(P), Pascoe(P), Karni(P).
5	FLOAT	Battersby(1966)(P), CMM(J), Pascoe(P), Fendley(P), Karni(P).
6	SPT	FT(J), Pascoe(P), Tonge(A), CMM(J), Fendley(P), Karni(P).
7	LPT	FT(J), Pascoe(P), Tonge(A), Arcus(1966)(A), CMM(J), Karni(P).
8	NIMSUCC	Tonge(A), Arcus(A).
9	NEXTAV	Arcus(A).
10	RANDOM	HL(J), Pascoe(P), Tonge(A), CMM(J).
11	ALLSUCC	Tonge(A), Arcus(A), Fendley(P), Karni(P).
12	NONREL	Arcus(A).
13	NONDEP	Arcus(A), CMM(J).
14	SUCC/LEV	Arcus(A).
15	RPOSWT	HB(1961)(A), Tonge(A), Arcus(A), CMM(J), Karni(P).
16	PROD	Arcus(A).
17	RED	Pascoe(P).

Table 6.9 (contd.)

	RULE	REFERENCES
18	CUMRED	Pascoe(P).
19	1/RPOSWT	Tonge(A), CMM(J).
20	PRPOSWT	CMM(J).
21	1/NIMSUCC	Tonge(A), CMM(J).
24	FLOAT/SUC	CMM(J).
25	LEVELS	Tonge(A).

Notes to Table 6.9. Abbreviations in the table.

CMM Conway, Maxwell and Miller (1967).

HB Helgeson and Birnie (1961).

HL Heller and Logemann (1962).

FT Fisher and Thompson (1963).

next is the crucial step. This step may take two distinct forms:

a. Weighted rule selection.

b. Weighted job selection.

637. Weighted rule selection. At any stage, choose a priority rule from a set of possible rules, using a weighted random selection process. Then select a job from AV using this chosen rule. Methods of this kind have been used by Heller and Logemann (1962), Fisher and Thompson (1963), and Tonge (1965); they are outside the scope of this thesis.

638. Weighted job selection. The sampling methods in which we are most interested select a job from AV at random, possibly using as weighting factors job priorities, calculated by a fixed priority rule. The actual job selection methods can vary considerably - some possible procedures are:

a. Sort the jobs in AV by some fixed priority rule. Select the first job in the sorted sequence with probability r; if this job is not selected, then choose the second job with probability r; continue in this way until a job is chosen. In general, the probability of the n^{th} job in the sequence being selected in the first pass is $(1-r)^{n-1}.r$; if no job in AV is chosen, the process is repeated, starting again with the first job. This method has been used by Levy, Thompson and Wiest (1962) and Wiest (1967).

1. Enter.
2. Set $j \leftarrow 1$.
3. Form a feasible schedule S_j.
4. If j is less than M, the sample size, then go to 5; otherwise, go to 6.
5. Set $j \leftarrow j + 1$. Go to 3.
6. Select the best schedule of $S_1, S_2, \ldots, S_M$.
7. Exit.

Fig 6.5 Sampling methods, for sample of size M.

b. Choose a job from AV at random, so that all jobs in AV have the same probability of being chosen, i.e. probability p = 1/#(AV) (Giffler, Thompson and Van Ness, 1963).

c. Choose a job from AV at random, using the priorities of the jobs to weight the choice, i.e., choose job i with probability p(i) given by

$$p(i) = \frac{P(i)}{\Sigma P(j)}$$

where the sum is over all jobs j in AV.

639. It is with methods of this third kind that we shall be primarily concerned (the second method is a special case, where P(i)=1 for all jobs i). Methods of this nature have been suggested by Giffler and Thompson (1960), and first treated comprehensively by Arcus (1966) for the assembly line balance problem.

640. Sample size. If a scheduling process generates schedules randomly, and if r is the proportion of all feasible schedules that are optimal, then the probability that the first sequence generated will be optimal is r, and the probability that it will not be optimal is (1-r). The probability that no optimal sequence is generated in the first M trials is $(1-r)^M$, and the probability p that at least one is optimal is given by

$$p = 1-(1-r)^M.$$

The probability p increases as either r or M increases. Later, when

we describe the experimental procedures used to examine the sampling methods (para 752), it will be seen that we are using a constant number of trials M, and attempting to improve the probability p of selecting an optimal schedule by forming a biased sample of feasible schedules, in which we intend the proportion of optimal schedules to be greater than r.

CHAPTER 7. EXPERIMENTAL PROCEDURES

CHAPTER 7. EXPERIMENTAL PROCEDURES

701. In this chapter, we describe an experimental investigation of some of the heuristic methods discussed in the previous chapter. We shall be examining parallel methods and sampling methods; although serial methods are not investigated explicitly, we have already noted that static rules produce the same schedules in both serial and parallel methods (para 604, Example 6.2), and hence we shall be able to draw some conclusions about the serial methods from the schedules obtained by using static priority rules in the parallel methods.

Aims

702. The aims of the experimental investigation were to answer the following general questions:

a. How does the choice of priority rule to be used in a parallel method affect the schedule obtained? Is the choice influenced by the kind of project to be scheduled?

b. How does the choice of priority rule to be used in a sampling method (of the form of para 638.c) affect the schedule obtained? Is the choice influenced by the kind of project to be scheduled?

c. Are the sampling methods better than the corresponding parallel methods?

703. The general questions above suggested a number of far more specific requirements. In the case of parallel methods, these were:

a. What is the detailed behaviour of the priority rules over all projects? Are some rules always good or always bad?

b. How do changing project characteristics affect the results of the priority rules overall?

c. How do changing project characteristics affect the results obtained by using specific priority rules?

704. The above questions may also be asked about the sampling methods. In addition, there are further considerations relating to the sample size, the distributions of schedules with different priority rules, and the set of optimal schedules; most of these are outside the scope of this thesis, but we shall touch on some aspects of them later.

705. With respect to the comparison between the parallel and the sampling methods, a quantitative as well as a qualitative answer was required to the question of para 702.c.

Measures of Effectiveness

706. Two measures of scheduling effectiveness were used in the experiments. For each schedule obtained, the measures were:

a. The project duration of the schedule.

b. The utilisation factor of the schedule.

707. Project duration, PDUR, was calculated by

$$PDUR = \max_{i \in A} \{s(i) + d(i)\}$$

(para 505). This was to be minimised, so low values indicated a better schedule than high values.

708. The utilisation factor, UF, was calculated by

$$UF = \sum_{k=1}^{K} \frac{\sum_{i \in A} r(i,k).d(i)}{PDUR.R(k)} \cdot \frac{100}{K} .$$

For fixed availabilities of resources R(k), maximising UF is equivalent to minimising PDUR (cf. the comments in para 322). High values of UF indicate a better schedule than low values. The utilisation factor is sensitive to failure to use resources, and is sensitive to inefficiencies caused by a job being delayed by the lack of, say, one resource.

709. Although these two measures are equivalent in the sense that minimising PDUR has the same effect as maximising UF, as measures of scheduling effectiveness they have quite different properties. The project duration of a project is dependent on its work content and the availability of resources; even if some attempt is made to normalise the critical path durations, for example, of different projects, there are likely to be large variations in the magnitudes of project duration over different schedules and different projects. Utilisation factor, however, does not have the same dependence on work content, as it is a dimensionless quantity (in the sense of dimensional analysis; the numerator and denominator of the summand in para 708 both have the dimension of time x resource k); thus

utilisation factors show less variation between different projects, and are better for statistical analysis, than project durations. Results of the analysis of both project duration data and utilisation factor data will be used in evaluating schedules, but we shall find that utilisation factor, because of the smaller variations in its values, is the more sensitive measure.

Project Description

710. One of the aims of the experimental investigation was to assess the effects of varying project characteristics on the results obtained by different priority rules, for both the parallel and sampling methods. In the following sections we shall describe some of the characteristics relevant to this part of the investigation and the reasons for their inclusion, and we shall then consider the generation of projects having specific characteristics for use in the experiment.

711. Projects will be described by a number of characteristics, based on:

a. The size and structure of the project graph.

b. Time properties of the jobs and the project.

c. Resource properties of the jobs and the project.

Size and Structure Parameters

712. Ideally, projects used in the experiments would have been of a size comparable to that found in industrial applications; however,

the amount of computer time required to generate sufficient data to evaluate the solution procedures with any degree of statistical confidence would have been very large. Therefore, it was decided to test exhaustively a number of projects of moderate size; this size was sixty jobs. There were also practical reasons for the choice of sixty for the number of jobs, N: the critical path calculations and much of the 'housekeeping' in the computer programs use binary (zero-one) matrices and vectors, which are stored as bit matrices within the 60-bit words of the CDC 6400 used for the experiment.

713. The structure of the project graph was characterised by the parameter order strength, S (Mastor, 1970), defined by

$$S = m/h,$$

where m is the total number of ordering relations in the project (including all those implied by the transitivity of the ordering relations), and h is the maximum number of ordering relations possible in a project with the given number of jobs. More formally, we have

$$m = \sum_{i \in A} \#(\Gamma^* i),$$

and

$$h = N(N-1)/2,$$

where Γ^* is the open closure function defined in para 208, and #(X) is the number of elements in the set X. These definitions will be clearer after we consider the following examples.

Example 7.1. Strong ordering. Fig 7.1.a shows a project graph in which the number of ordering relations is a maximum. Here we have six ordering relations (1 precedes 2, 3 and 4; 2 precedes 3 and 4; 3 precedes 4), i.e. N = 4, and N(N-1)/2 = 6. The order strength is

$$S = (3 + 2 + 1 + 0)/6 = 1,$$

and there is only one possible active schedule, described by the permutation (1,2,3,4).

Example 7.2. Weak ordering. Fig 7.1.b shows a project graph in which the ordering is a minimum; there are no ordering relations and S = 0/6 = 0. The number of feasible permutations is 24 (i.e. 4!).

Example 7.3. Intermediate ordering. Fig 7.1.c shows a project graph with seven ordering relations (1 precedes 2, 3, 4 and 5; 2 precedes 4 and 5; 3 precedes 4). Here we have N = 5, N(N-1)/2 = 10, and so S = (4 + 2 + 1 + 0 + 0)/10 = 0.7. There are five feasible permutations: (1,2,3,4,5), (1,2,3,5,4), (1,2,5,3,4), (1,3,2,4,5), and (1,3,2,5,4).

714. Thus $0 \leqslant S \leqslant 1$. We also see from the above examples that the strength of the partial ordering affects the number of alternative schedules that may be generated. If a partial ordering is strong, there are few alternative ways to schedule jobs, and if it is weak, there are a large number of possible schedules. When S is large, as

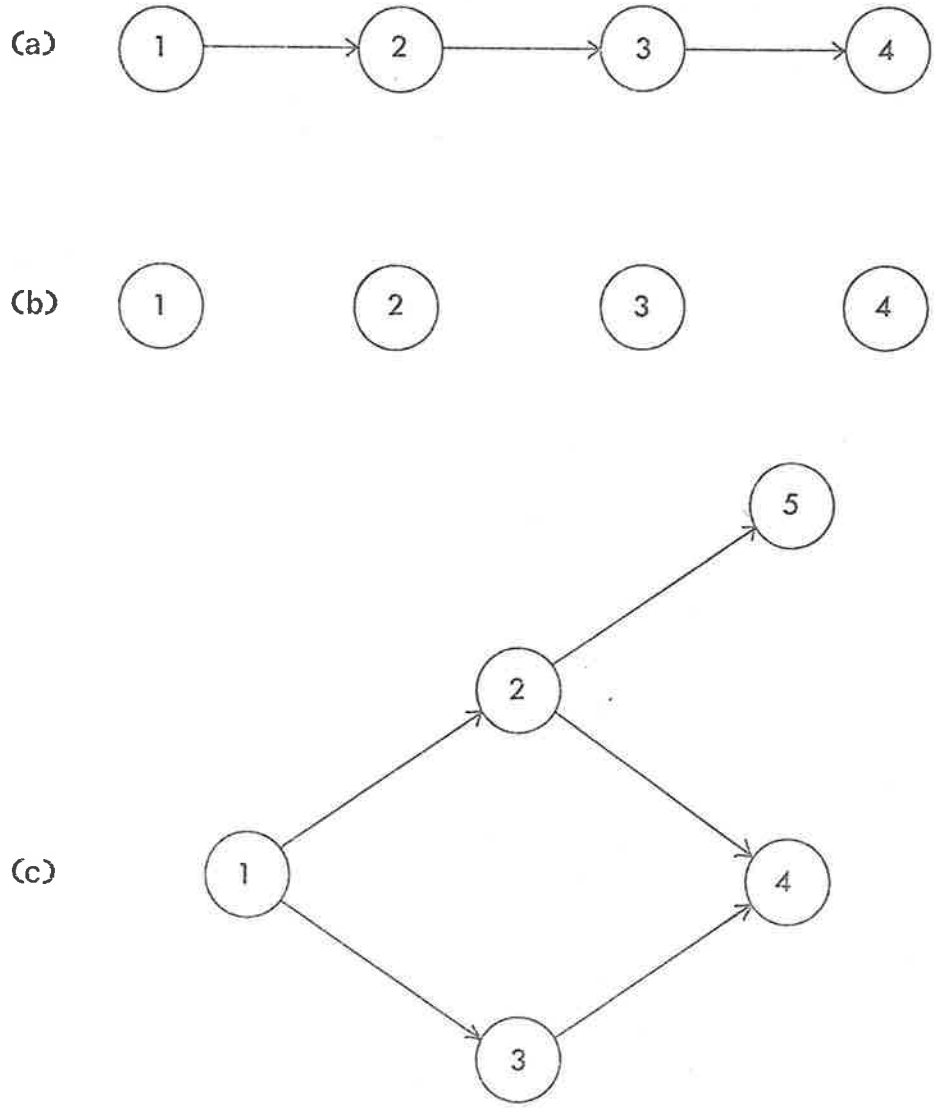

Fig 7.1. Graphs with:

a. Strong ordering (S=1).

b. Weak ordering (S=0).

c. Intermediate ordering (S=0.7).

in Fig 7.1.a, the project graph is long and narrow; when S is smaller, the graph becomes shorter and fatter (we are following the convention that arcs are directed from left to right; in this context, the terms 'long', 'narrow', 'short' and 'fat' have intuitive meanings). In summary, varying the order strength generates classes of project that represent the variation in technological ordering constraints occurring in applications.

Time Parameters

715. We have already come across one parameter related to time properties of the project, the CP duration CPDUR (para 304). This parameter was set to 1000 for all projects, in an attempt to normalise projects and reduce the variation in scheduled durations between different projects. Although this did not have the desired effect of reducing variations in PDUR, it did provide a certain amount of control that later became useful in the detailed analysis of the results (in particular with respect to the effects of order strength on the results, para 822).

716. A more important time parameter is related to the free float of the jobs of the project. We define the free float, ff(i), of job i, to be the float associated with job i when all jobs start as early as possible; the free float of a job is the difference between its earliest finish time and the earliest of the early start times of its successors, i.e.

$$ff(i) = 0 \text{ if } \Gamma i = \emptyset$$

$$= \text{Min}_{j \in \Gamma i} \{\ell(j)\} - (\ell(i)+d(i)) \text{ otherwise.}$$

Thus, free float is a measure of the ability of a job to be moved in time without affecting any other jobs. Increasing the free float of jobs in the project increases the flexibility available to overcome resource limitations locally without upsetting other jobs; this is likely to cause a reduction in the scheduled project duration, and a decrease in the utilisation factor, due to the less efficient use of resources. (We note here that if a job j has only one predecessor i, then that predecessor will have zero free float, since $\ell(j) = \ell(i) + d(i)$ and this is the minimum value of ℓ for any successor of i, by definition. Hence $ff(i) = 0$. The distribution of free float in a project depends on the structure of the project graph.)

Example 7.4. Table 7.1 shows free float calculations for the project of Fig 7.2. For comparison, the float f(i) of jobs is also shown, for the case PHOR = CPDUR (see paras 305 and 306).

717. We shall describe the free float in a project by the parameter density, D, defined by

$$D = \frac{\Sigma d(i)}{\Sigma(d(i)+ff(i))},$$

where the sums are over all jobs i in A. Then $0 < D \leqslant 1$. In a project with density $D = 1$, all jobs are critical.

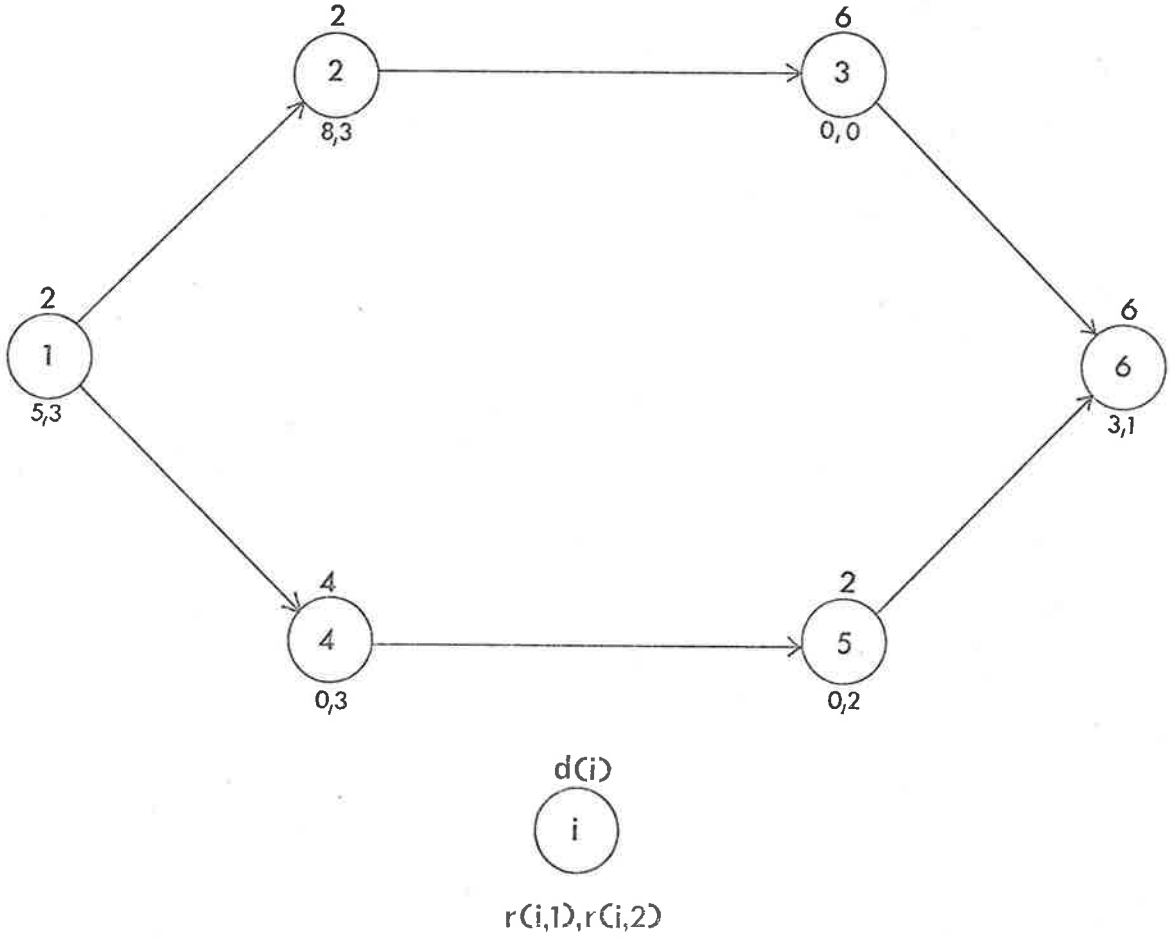

Fig 7.2. A simple project with N=6 and K=2.

i	d(i)	ℓ(i)	ℓ(i)+d(i)	ff(i)	$u^*(i)$	f(i)
1	2	0	2	0	0	0
2	2	2	4	0	2	0
3	6	4	10	0	4	0
4	4	2	7	0	4	2
5	2	6	8	2	8	2
6	2	10	12	0	10	0
TOTAL	18			2		

Table 7.1. Free float and float for the project of Fig 7.2.

Example 7.5. For the project of Example 7.4, we have

$$\Sigma d(i) = 18,$$

$$\Sigma(d(i) + ff(i)) = 20,$$

and so D = 0.9.

Resource Parameters

718. All the projects used in the experiment had five different kinds of resources available to them (K = 5). This was felt to be a realistic, yet manageable, number. The other resource parameters related to the number of different kinds of resources required by individual jobs, and to the magnitude of these requirements.

719. Denote by q(i) the number of different kinds of resources used by job i, that is, the number of non-zero values of r(i,k) over all resources k. Then the relative number of different kinds of resources used by individual jobs in a project is described by the parameter resource factor, rf, defined by

$$rf = \frac{\frac{1}{N} \sum_{i \in A} q(i)}{K},$$

where the numerator is the average number of kinds of resources required by jobs in the project, and the denominator is the total number of kinds of resources available to the project. If all jobs require at least one kind of resource (i.e. $q(i) \geqslant 1$, for all i), then

$$\frac{1}{K} \leqslant rf \leqslant 1.$$

Example 7.6. Table 7.2 gives details of the resource requirements of the project in Fig 7.2. In this case K = 2. Values of q(i) are shown; the average number of kinds of resource required per job is 8/6. The resource factor is given by

$$rf = (8/6)/2 = 2/3.$$

720. The magnitude of the requirements for resource k is measured by the parameter resource strength, rs(k). This quantity is defined as the number of average activities which could be accommodated within the availability R(k) of resource k. That is,

$$rs(k) = R(k)/(\text{Mean requirement for resource k per job})$$

$$= R(k)/\left\{\frac{(\text{Total requirement for resource k over all jobs})}{N}\right.$$

$$= \frac{R(k).N}{\sum_{i\varepsilon A} r(i,k)} .$$

Example 7.7. If the resource availabilities of the project of the previous example are R(1) = 8 and R(2) = 4, we have

$$\Sigma r(i,1) = 16,$$

$$\Sigma r(i,2) = 12,$$

and so

$$rs(1) = 8 \times 6/16 = 3,$$

$$rs(2) = 4 \times 6/12 = 2.$$

i	r(i,1)	r(i,2)	q(i)
1	5	3	2
2	8	3	2
3	0	0	0
4	0	3	1
5	0	2	1
6	3	1	2

Table 7.2 Resource data for the project of Fig 7.2.

Parameter Values

721. A set of projects was generated for use in the experiments. The values of the parameters used to describe these projects are given below; they are summarised in Table 7.3. A description of the methods used to generate projects with specific parameter values is given in the next section.

722. Constant parameters. The values of parameters that were constant in the experiment were:

a. Project size, N. Each project had sixty jobs.

b. Resources, K. Each project had five different kinds of resources available to it.

c. CP duration, CPDUR. The CP duration of each project was fixed at 1000.

d. Resource strength, rs(k). The resource strength was set to 5 for all resources.

723. Varying parameters. The values of the other parameters were allowed to take a range of values:

a. Order strength, S. Projects were generated with order strengths of 0.1, 0.25, 0.5 and 0.75.

b. Density, D. Two values of density were used, 0.5 and 0.75.

c. Resource factor, rf. Two values of resource factor were used: 0.5, when jobs used, on average, half the kinds of resources available, and 1.0, when all jobs used all kinds

PARAMETER	SYMBOL	VALUES
Project size	N	60
Number of resources	K	5
CP duration	CPDUR	1000
Resource strength	rs(k)	5, for all k
Order strength	S	0.1, 0.25, 0.5, 0.75
Density	D	0.5, 0.75
Resource factor	rf	0.5, 1.0

Table 7.3 Parameter values.

of resources.

Project Generation

724. The generation of a project with a specific set of parameter values proceeds sequentially:

a. Structure. Create a project graph with sixty vertices (jobs), having the desired order strength S.

b. Time. Assign durations to the jobs of the project and adjust them so that CPDUR is 1000. Further adjust the durations of the non-critical jobs to obtain the required density D.

c. Resources. Assign resources to the jobs, giving the required resource factor rf. Set the availability R(k) of each resource such that the desired resource strength is obtained.

725. Structure. Order strength, S, is given by

$$S = m/h,$$

where

$$m = \sum_{i \in A} \#(\Gamma^* i),$$

and

$$h = N(N-1)/2 \qquad \text{(para 713)}.$$

In this case we have N = 60, and so h = 1770; for a given order strength S we require

$m = 1770\ S.$

726. We shall represent the open closure relations Γ^* by an open closure matrix, C, in which

$C_{ij} = 1$ if $j \varepsilon \Gamma^* i$,

$= 0$ otherwise.

The matrix C has the properties:

a. Transitivity. If $C_{ij} = 1$ and $C_{jk} = 1$, then $C_{ik} = 1$. This follows from the transitivity of Γ^*, i.e. if there is a path from i to j, and a path from j to k, then there is a path from i to k.

b. Upper Triangular. Since the graph of a project is topologically ordered (para 303.d), if $j \leqslant i$, then there is no path from i to j, and $C_{ij} = 0$.

727. The problem of generating a project graph with order strength S becomes one of creating a strictly upper triangular open closure matrix C with $m(=1770S)$ one-elements (since $\#(\Gamma^* i) = \Sigma_j C_{ij}$). A suitable generation method is described in Annex D.

728. The open closure matrix contains all the information necessary to create the project graph. However, because we have required transitivity, the matrix C contains much redundant information; we shall now simplify C by removing this redundancy, forming from C a precedence matrix W in which

$W_{ij} = 1$ if $j \varepsilon \Gamma i$,

0 otherwise.

We shall call a relationship $j \varepsilon \Gamma i$ redundant if and only if there exists a job k such that $k \varepsilon \Gamma^* i$ and $j \varepsilon \Gamma^* k$ (graphically, an arc (i,j) is redundant if and only if there exists a path from i to j containing some other vertex k). We shall now show how to construct W from C.

729. Since the elements of C are either zero or one, we may regard C as a column vector whose elements are binary row vectors of 60 elements. For two such row vectors x and y, with elements $(x_1, x_2, \ldots, x_{60})$ and $(y_1, y_2, \ldots, y_{60})$ respectively, define

$$x \wedge y = (x_1 \wedge y_1, x_2 \wedge y_2, \ldots, x_{60} \wedge y_{60})$$

$$\neg x = (\neg x_1, \neg x_2, \ldots, \neg x_{60}),$$

where $\wedge$ and $\neg$ are product and negation operators defined by the relationships in Table 7.4. Denote row i of C by C_i, and similarly for W.

730. Construction. Set

$$W_i = C_i \wedge \left(\bigwedge_{j \varepsilon \Gamma^* i} \neg C_j \right),$$

for i = 1, 2, ..., 60.

731. Theorem. The matrix W constructed above has no redundancy.

Proof. Consider the k^{th} element of row i of C (element C_{ik}). Then $W_{ik} = 1$ if and only if $C_{ik} = 1$ and $C_{jk} = 0$ for all successors j of i; that is, the relationship $k \varepsilon \Gamma i$ is preserved if and

i	j	$i \wedge j$	$\neg i$
0	0	0	1
0	1	0	1
1	0	0	0
1	1	1	0

Table 7.4 Definition of $\wedge$ and $\neg$.

(If we regard 0 as FALSE and 1 as TRUE, then $\wedge$ and $\neg$ correspond with the logical operators AND and NOT.)

only if it exists in C and there is no job j which is both a successor of i and a predecessor of k. If such a job j exists, then $C_{jk} = 1$, and $W_{ik} = 0$. Thus, if $k \in \Gamma i$ is redundant, $W_{ik} = 0$ and the redundancy has been removed; if non-redundant, then $W_{ik} = 1$ and the relationship is preserved.

732. The matrix W now contains a non-redundant description of the project, which has sixty jobs (N=60) and the desired order strength S.

733. Time. Initially, jobs are assigned durations in the range 1 to 10. The CP duration of the project is then calculated, and the individual job durations are normalised such that the CP duration becomes 1000. (If the initial value of CPDUR is x, normalisation involves changing the duration of job i from d to 1000 d/x, for all jobs i, truncating this value to an integer, and adding one time unit to the durations of randomly chosen jobs until CPDUR = 1000. Time units will generally have to be added, due to the truncation process.)

734. After normalisation, the durations are adjusted to give the desired density D. It is necessary to alter the durations of the non-critical jobs (so that CPDUR is not changed) by a total amount x, where

$$D = \frac{\Sigma d(i) + x}{\Sigma(d(i)+ff(i))} .$$

This amount x must be added to the durations of non-critical jobs, so that it can be taken up in free float, leaving the denominator of the above expression unchanged.

Example 7.8. Consider the project of Fig 7.2, which has density D = 0.9 (Table 7.1, Example 7.5). If we wish to change the density to a value of, say, 0.8, we have

$$0.8 = \frac{\Sigma d(i)+x}{\Sigma(d(i)+ff(i))}$$

$$= \frac{18 + x}{20},$$

i.e. $x = -2$.

Thus, the durations of non-critical jobs must be reduced by a total of 2 time units. Suppose we reduce the durations of jobs 4 and 5 by one unit each. The new values of d(i) and ff(i) are shown in Table 7.5. We now have

$$D = \frac{\Sigma d(i)}{\Sigma(d(i)+ff(i))}$$

$$= 16/20$$

$$= 0.8, \text{ as required.}$$

735. Resources. In comparison with the generation of projects having specific values of order strength and density, generating data to obtain specific values of the resource parameters is easy. For a given resource factor rf, the average number of kinds of resources used per job is 5rf, since 5 kinds of resource are available to the entire project (K=5). We then have sixty jobs i, each capable of using 5 kinds of resources k, giving 300 job-resource positions (i,k), of which a total of 300rf must be filled with non-zero requirements, and the remainder with zero entries. So to generate resource requirements

i	d(i)	ℓ(i)	d(i)+ℓ(i)	ff(i)
1	2	0	2	0
2	2	2	4	0
3	6	4	10	0
4	3 (=4-1)	2	5	0
5	1 (=2-1)	5	6	4 (=2+2)
6	2	10	12	0
TOTAL	16			4

Table 7.5 Adjusted durations for the project of Fig 7.2.

The project now has density D = 0.8.

r(i,k) for a project, such that the desired value of rf is obtained, we proceed as follows:

a. Select (300*rf) job-resource positions from the 300 available, and set the remainder to zero.

b. For a selected position (i,k), allocate a resource requirement r(i,k) in the range 1 to 10.

736. The parameter resource strength, rs(k), is fixed at 5.0 for all resources k. It is used to set the availability R(k) of resource k, by

$$R(k) = \text{Max}\left\{\begin{matrix} rs(k).\Sigma_i r(i,k)/60 \\ \text{Max}_j \{r(j,k)\} \end{matrix}\right\}$$

$$= \text{Max}\left\{\begin{matrix} \Sigma_i\ r(i,k)/12 \\ \text{Max}_j \{r(j,k)\} \end{matrix}\right\},$$

for k = 1,2,..., 5. (Obviously, if the maximum requirement for resource k exceeds the availability, then no feasible schedule exists; hence the second term in the definition. The first term is derived from the definition of rs(k) in para 720.)

Experimental Procedures for the Parallel Methods

737. The parallel methods were examined by means of a factorial experiment. Thirty-two projects were generated, two with each combination of values of order strength (S = 0.1, 0.25, 0.5 and 0.75), density (D = 0.5 and 0.75), and resource factor (rf = 0.5 and 1.0);

each of these projects was scheduled, using a parallel method, with each of the twenty-six priority rules listed in Table 6.5. This gave a total of 832 schedules (4 order strengths x 2 densities x 2 resource factors x 2 replications x 26 priority rules), for each of which the project duration and the utilisation factor were calculated.

738. The experiment falls into the class of multi factor experiments with nested factors and repeated measures; it may be represented schematically by the table of Fig 7.3. In this context, we are regarding a project graph (fixed S) with a set of resource requirements (fixed rf) as an experimental unit, and priority rules and density as experimental treatment conditions. With this in mind, in Fig 7.3, G1 represents a group of two replicated experimental units, namely the two with parameter values S = 0.1 and rf = 0.5; G2 represents a second group of two units, and so on. To specify a particular experimental unit, we must first specify values of order strength and resource factor, to identify the group, and within that group, the particular element, or replication; in this situation we say that the replication is nested within the order strength and resource factor, and the experiment is partially hierarchical. The nesting arises because only four project graphs were generated for each value of order strength, two of these being used with one value of resource factor and two with the other value. However, each was used with both values of density, and scheduled with all rules; where

		DENSITY, D=0.5				DENSITY, D=0.75			
ORDER STRENGTH	RESOURCE FACTOR	PRIORITY RULE 1	2	.....	26	1	2	.....	26
0.1	0.5	G1	G1	. . . .	G1	G1	G1	. . . .	G1
0.1	1.0	G2	"	. . . .	G2	G2	. . . . .		G2
0.25	0.5	G3	"						"
....	..	"	"						"
....	..	"	"						"
0.75	1.0	G8	"	. . . . .		. . . . .			G8

Fig 7.3. Multifactor experiment with nested factors and repeated measures.

the same experimental unit (project graph with resources) is observed (scheduled) under several treatment conditions (priority rules and density), the experiment has repeated measures.

739. Experiments of this form may be examined by Analysis of Variance (AOV) techniques, for which Winer (1970) is a comprehensive reference. Winer discusses nested factors (section 5.12), and multi factor experiments with repeated measures (Chapter 7); in section 7.3 he discusses an experiment with repeated measures on two factors (e.g. priority rules and density), and in section 7.4 he discusses a repeated measures experiment in which the replication effect is nested under two other factors (e.g. order strength and resource factor).

740. In this experiment the factors were priority rules, density, order strength, resource factor and replications; Table 7.6 lists these factors, with the symbols and indices used to represent them in the AOV model, and the number of levels of each. The AOV was based on a linear model which had the form

$$X(LJIKM) = U + R(L) + D(J) + S(I) + F(K) + N(M(IK)) + RD(LJ) + RS(LI) + RF(LK) + RN(LM(IK)) + DS(JI) + DF(JK) + DN(JM(IK)) + SF(IK) + RDS(LJI) + RDF(LJK) + RDN(LJM(IK)) + RSF(LIK) + DSF(JIK) + RDSF(LJIK) + E(LJM(IK)),$$

where X is the criterion measure (project duration or utilisation

SER	FACTOR	SYMBOL	INDEX	NUMBER OF LEVELS
1	Rules	R	L	26
2	Density	D	J	2
3	Order Strength	S	I	4
4	Resource Factor	F	K	2
5	Replication	N	M	2

Table 7.6 Factors in the AOV model for the parallel methods.

factor), U is the grand mean of this measure over all projects and all rules, and E is the experimental error. The notation N(M(IK)) indicates that the replication effect is nested under both factors S and F.

741. This model is not a simple one. Its complexity arose historically: in the initial stages of the research, some limited experiments were planned to test a range of priority rules, in the hope that some rule or set of rules would be of outstanding merit; this did not eventuate, and later when it became necessary to design a more comprehensive experiment for detailed investigation, the experimental investment was already large enough to warrant the inclusion of the previous results in the new design. Thus, the experimental investigation grew as the research progressed, and its emphasis changed as more results became available; the model is somewhat cumbersome in consequence. However, the results it produces are entirely adequate for the aims of the experiment. The interactions which are omitted from the model due to the partial hierarchical design (for example, the model contains no replication x order strength or replication x resource factor interactions, because of the nesting of replications within order strength and resource factor) are not of major importance in this context; they all involve the replication factor, and their variances would have been error terms in the calculation of F statistics (see the next paragraph, and para 810).

742. From the AOV we will obtain a partitioning of the total experimental variation into variations attributable to specific effects, namely the main effects and interactions of the AOV model. The variations and their corresponding degrees of freedom are used to derive the variances attributable to elements of the model; from these variances, and the appropriate error variance in each case, are derived F statistics, which may be used to test hypotheses about the population means for the elements of the model. Thus, we can examine the variance attributable to the priority rules, for example, and determine whether this is more likely to be due to chance or to real differences in effects between the rules.

Examination of Effects and Interactions

743. The AOV described in the previous section can be used to indicate main effects and interactions in which there are significant differences between population means, but no information is provided about the specific nature of the differences. In this section, we shall outline some simple test procedures for investigating significant effects and interactions, using examples to illustrate the methods.

744. We first introduce dot notation. For the criterion measure X(LJIKM), we make the convention that if an index is replaced by a dot, then the criterion has been averaged over all values of that index. (Actual parameter values associated with each factor level are listed

in Table 7.7.)

Example 7.9.

a. $X(.....) = U$

$$= \frac{\Sigma\ \Sigma\ \Sigma\ \Sigma\ \Sigma\ X(LJIKM)}{26 \times 2 \times 4 \times 2 \times 2},$$

the grand mean.

b. $$X(i....) = \frac{\Sigma\ \Sigma\ \Sigma\ \Sigma\ X(iJIKM)}{2 \times 4 \times 2 \times 2},$$

the mean value of the criterion, generated with rule i, over all levels of the factors D, S, F, and N.

c. $$X(i2...) = \frac{\Sigma\ \Sigma\ \Sigma\ X(i2IKM)}{4 \times 2 \times 2},$$

the mean value of the criterion, generated with rule i, over all projects with density 0.75.

745. Main effects. We shall illustrate the procedure for examining a main effect by considering the analysis of a priority rules main effect. This method can be used in a similar way to examine the main effects of order strength, density, and resource factor (but not the replications main effect, which is an error term in the AOV).

246. Step 1. Form the group means X(i....) for all rules i. For any rule i, values of X(i....) are formed by averaging X(iJIKM) over all values of J, I, K, and M; denote the variance of this popu-

FACTOR R		FACTOR D		FACTOR S		FACTOR F	
INDEX L	RULE	INDEX J	DENSITY	INDEX I	ORDER STRENGTH	INDEX K	RESOURCE FACTOR
1	1	1	0.5	1	0.1	1	0.5
:	:	2	0.75	2	0.25	2	1.0
i	i			3	0.5		
:	:						
26	26			4	0.75		

Table 7.7 Factor levels and corresponding parameter values, parallel methods.

lation by S_i^2, and its size by n. (For all main effects, n is constant over different levels of the corresponding factor. In the case of priority rules, all the X(i....) are over 32 values of the criterion, and n = 32).

747. Step 2. Rank the rules in order of decreasing group means.

748. Step 3. Test the significance of the difference between the means X(i....) and X(j....), for all pairs of rules i and j in the ranking, using a t Test. This may be performed as follows (see Winer, section 2.3, for the theory of this procedure, and section 2.4 for computational formulae): the pooled estimate of the common population variance is

$$s_p^2 = (S_i^2 + S_j^2)/2,$$

with 2n-2 degrees of freedom; the t statistic is then given by

$$t = \frac{|X(i....) - X(j....)|}{(S_p^2(1/n + 1/n))^{\frac{1}{2}}}$$

$$= \frac{|X(i....) - X(j....)|}{((S_i^2 + S_j^2)/n)^{\frac{1}{2}}}$$

also with 2n-2 degrees of freedom. If i is ranked before j, the t statistic calculated above may be used in a one-sided test for X(i....) = X(j....) against the alternative X(i....) > X(j....), rather than the corresponding equal-tails test for X(i....) =

X(j....) against the alternative X(i....) ≠ X(j....).

749. In the above procedure, we have assumed that the samples are drawn from populations with the same variance, the best estimate of which is the pooled estimate S_p^2. Winer, section 2.5, indicates the robustness of the t Test with respect to moderate departures from the hypothesis of homogeneity of variance, and describes an F statistic for testing the equality assumption. If $S_i^2 > S_j^2$, then calculate

$$F = S_i^2/S_j^2$$

with (n-1, n-1) degrees of freedom, and reject the null hypothesis that the parent population variances are equal, at significance level 2α, if F exceeds the tabled value $F_\alpha(n-1, n-1)$ for $P(F) = \alpha$. (It will be shown later, in para 817, that the hypothesis of homogeneity of variance is in fact justified, at least in the case of the priority rules main effects.)

750. Interactions. We shall illustrate the procedure for examining an interaction by considering the analysis of a priority rule by resource factor interaction. The interaction between level L of factor R and level K of factor F is a measure of the extent to which the criterion mean for the combination of levels L and K, i.e. X(L..K.), cannot be predicted from the sum of the corresponding main effects X(L....) and X(...K.). In this case, we are interested in the RF(LK) term in the AOV model; an estimate of this term is given

by

$$\text{est RF(LK)} = X(L..K.) - X(L....) - X(...K.) + X(.....)$$

(Winer, section 5.10). Here, X(L..K.) contains both main effects and interactions; from this are subtracted the main effects X(L....) and X(...K.); the term X(.....), the grand mean, is a correcting factor.

751. The term est RF(LK) indicates whether rule L performs better or worse than would be expected from its overall effect, at level k of resource factor. Plotting values of est RF(LK) against, say, rules, we can obtain a 'picture' of the interaction; we shall be examining such graphical representations when we consider the experimental results. The detailed theory of this process, and numerical examples, may be found in Winer, sections 5.10 and 5.11.

Experimental Procedures for the Sampling Methods

752. The sampling methods used in the investigation had a sample size of 100. This particular sample size had a number of advantages:

a. It was large enough for any effects due to the sampling method, in comparison with the parallel method, to become apparent.

b. It was large enough for the effect of the method on the relative behaviour of different rules to be assessed.

c. It was small enough for a limited experiment to be computationally practicable. The sampling method, in generating a schedule, forms M feasible subschedules, where M is the sample size (see Fig 6.5). Thus, it takes about M times as long to form a schedule with the sampling method as it does with the corresponding parallel method.

753. Like the parallel methods, the sampling methods were examined by means of a factorial experiment. A total of eight projects was used, one with each combination of values of order strength (S = 0.25 and 0.5), density (D = 0.5 and 0.75), and resource factor (rf = 0.5 and 1.0); each of these projects was scheduled, using a sampling method with sample size 100, with each of a set of fourteen priority rules (rules 1, 2, ..., 13 and 15). This gave a total of 112 schedules (2 order strengths x 2 densities x 2 resource factors x 14 priority rules), for each of which project duration and utilisation factor were calculated.

754. This would appear on the surface to be a far smaller experiment than was performed with the parallel methods. However, the experimental effort involved is far greater, due principally to the far greater computation time required to form a schedule, noted in para 752.c above. As the investigation progressed, the general nature of the results became apparent: there was very little difference between the priority rules, no matter what the project characteristics

were. In view of this, it was not considered worthwhile to investigate a wider range of rules and project parameters, at a cost of a large amount of computation and experimental effort, with very little likelihood of obtaining further results, relevant to the investigation, than those already obtained.

755. The experiment is a 2 x 2 x 2 x 14 factorial experiment, with one observation per call; it is shown schematically in Fig 7.4, where the notation is similar to that used in Fig 7.3 for the parallel methods, but G1, G2,etc, each represent only one project rather than two. Because there is only one observation for any combination of levels of factors, there can be no within-cell variation, and hence no direct estimate of experimental error. The experiment was investigated with AOV techniques, based on the linear model.

$$X(LJIK) = U + R(L) + D(J) + S(I) + F(K) + RD(LJ)$$
$$+ RS(LI) + RF(LK) + DS(JI) + DF(JK) + SF(IK)$$
$$+ RDS(LJI) + RDF(LJK) + RSF(LIK) + DSF(JIK)$$
$$+ E(LJIK),$$

where the notation is the same as that used for the parallel model (para 740, Table 7.6), with the numbers of levels of the factors R and S reduced to 14 and 2 respectively. This model is an additive model, and its use requires some justification.

ORDER STRENGTH	DENSITY	RESOURCE FACTOR	PRIORITY RULE 1	2 13	15
0.25	0.5	0.5	G1	G1 . . . . G1	G1
0.25	0.5	1.0	G2	G2	G2
0.25	0.75	0.5	G3		..
0.25	0.75	1.0	G4		..
0.5	0.5	0.5	G5		..
0.5	0.5	1.0	G6		..
0.5	0.75	0.5	G7		..
0.5	0.75	1.0	G8	. . . . .	G8

Fig 7.4 Experimental design for sampling methods.

756. Two models may be postulated to underlie the observed data in this case, the additive model above, and the corresponding non-additive model

$$X(LJIK) = U + R(L) + D(J) + S(I) + F(K) + RD(LJ)$$
$$+ RS(LI) + RF(LK) + DS(JI) + DF(JK) + SF(IK)$$
$$+ RDS(LJI) + RDF(LJK) + RSF(LIK) + DSF(JIK)$$
$$+ RDSF(LJIK) + E(LJIK),$$

in which an interaction term RDSF(LJIK) is postulated. In the additive model, no such term is included, and variation from this source has been considered as part of the experimental error. Winer discusses the choice of models of this nature in section 5.20, and details a test, due to Tukey (1949), to aid in the choice. Tukey's test divides the residual variation due to sources other than the main effects and the two- and three-way interactions into two parts: one part measures non-additivity, and corresponds to the linear components of the RDSF interaction; the other part is the balance of the variation. If the component associated with non-additivity is significantly larger than the balance, then a decision is made against the additive model.

757. The sum of squares associated with non-additivity, SS_N, is given by

$$SS_N = \frac{\left[\sum_L \sum_J \sum_I \sum_K (X(L...)-U)(X(.J..)-U)(X(..I.)-U)(X(...K)-U)X(LJIK)\right]^2}{\sum_L (X(L...)-U)^2 \sum_J (X(.J..)-U)^2 \sum_I (X(..I.)-U)^2 \sum_K (X(...K)-U)^2}$$

If SS_R is the residual variation (i.e. the sum of squares not attributable to the main effects or the two- and three-way interactions of the additive model), then the balance of the variation, SS_B, is given by

$$SS_B = SS_R - SS_N.$$

The residual sum of squares has thirteen degrees of freedom (i.e. (2-1)x(2-1)x(2-1)x(14-1)), and one degree of freedom is required for SS_N. Thus, SS_B has 12 degrees of freedom, and we are able to calculate an F statistic

$$F = \left(\frac{SS_N}{1}\right) / \left(\frac{SS_B}{12}\right)$$

with (1,12) degrees of freedom. This statistic may be used to test the significance of the non-additivity component; if F exceeds the critical value defined by the level of significance of the test, then the hypothesis that the additive model is appropriate is rejected.

758. For the data in this experiment, the additive model is appropriate. Numerical details of Tukey's test as applied to this data will be considered in the discussion of the results (para 864).

Procedures for the Comparison of Methods

759. The projects used in the investigation of the sampling methods were used also in the investigation of the parallel methods; thus it was possible to compare directly results obtained by scheduling the same project, using the same priority rule, but with different heuristic methods. To determine just how much better were the sampling methods than the parallel methods, two kinds of tests were performed: one was a set of non-parametric tests, using a Sign Test (see, for example, Snedecor and Cochran, 1967, section 5.4); the other was a parametric test, based on a t Test (Crow, Davis and Maxfield, 1960, Section 2.2). (The assumptions on which these tests are based are not the same. The t Test assumes that the sample being tested has been drawn from a normally distributed parent population, whereas the Sign Test makes no assumptions about the distribution of the parent population. The t Test is more efficient than the Sign Test, i.e. for the same sample size in each case, the t Test is more sensitive. Although the normality assumption was not tested directly, it will be shown when we consider the results that the confidence limits derived from the t Test lie within those derived from the Sign Test, as expected if the parent population was indeed normally distributed.) The criterion tested was project duration; as will be seen in the next paragraphs, we are testing proportional differences between the methods, and for fixed availabilities of resources, project duration is inversely proportional to

utilisation factor, the other possible criterion that could be used (para 708), i.e. the percentage relative differences are the same for each criterion.

760. Let h(i) be the project duration obtained by scheduling a project with some priority rule in a sampling method, and let g(i) be the corresponding project duration obtained by scheduling the same project using the same rule in a parallel method. Further, let $h'(i,\delta) = h(i)(1+\delta/100)$. When $g(i) = h'(i,\delta)$, then

$$\delta = \left[\frac{g(i)}{h(i)} - 1\right]100,$$

the percentage relative difference between the results obtained by the two methods; if $g(i) > h'(i,\delta)$, we can say that for this rule and this project, the result of the sampling method is at least δ% better than that of the parallel method.

761. We shall use a Sign Test (Snedecor and Cochran, section 5.4) to examine the null hypothesis that there is no difference between items in the pairs $(g(i), h'(i,\delta))$, for a fixed value of δ(i.e. to examine the hypothesis that the relative difference between the methods is δ%). We calculate the chi-square statistic, χ^2, with one degree of freedom, by

$$\chi^2 = (|x-y| - 1)^2/n,$$

where x is the number of cases in which $g(i)-h'(i,\delta) > 0$, y is the number of cases in which $g(i)-h'(i,\delta) < 0$, and $n = x+y$ (the case $g(i)-h'(i,\delta) = 0$ is ignored), and we reject the no-difference hypothesis

at significance level α if χ^2 exceeds the tabled value of χ^2_α with one degree of freedom.

762. A set of such tests, for different values of δ enables us to determine, at significance level α in each case, limits δ_1 and δ_2 on the range of values of δ for which the no-difference hypothesis cannot be rejected. Outside this range, we reject the no-difference hypothesis, and the hypothesis that the relative difference between the methods is δ%. Then we can say that the sampling methods are better than the parallel methods by a percentage δ in the range (δ_1, δ_2), the significance level of each limit being α, i.e. δ_1 and δ_2 are the $(1-2\alpha)100$% confidence limits for the relative difference δ.

763. We can obtain similar confidence limits by applying a t Test to the population $\{(g(i)/h(i)-1)100\}$ of percentage relative differences between the results obtained by the two methods. If this population contains n items, and has mean m and standard deviation s, then the $(1-2\alpha)100$% confidence limits for the mean m are given by

$$m \pm t(\alpha,n-1)s/\sqrt{n}$$

(Crow et al, Section 2.2.3).

Significance Levels

764. The level of significance of a statistical test defines the probability level that is considered too low to warrant support

of the hypothesis being tested (Winer, Section 1.3); we shall usually be testing a null hypothesis. For tests associated with the parallel methods, we shall follow conventional practice and require a significance level of less than 0.05 (5%) or, preferably, less than 0.01 (1%), before rejecting an hypothesis.

765. The sampling methods, however, show great variability in the schedules they produce, even for replications involving the same priority rule and the same data; this occurs because each result is the extremum of a sample of results. Thus, the results of the sampling methods are subjected to errors which are not otherwise estimated in the various statistical tests. For this reason, we shall require a significance level of 0.01 for tests associated with the sampling methods, i.e. we shall be far more conservative than in the case of the parallel methods, because we are less sure of the original data. (We shall see examples of variability of the results from the sampling methods in the next chapter.)

Programming Methods for Schedule Generation

766. A general outline of the computer programs used to generate project schedules is given in Annex E. The method of storing resource information is of some interest, and will be discussed here.

767. Where the project horizon is large, it becomes impractical to store information about the resources available to the project and the resource usage of the current partial schedule by individual time

units; this is a similar problem to that encountered in the ILP formulations discussed in Chapter 5 (paras 519 to 521). Instead, we record the times at which there are changes in the resource availability or requirements, and the resource levels at those times.

Example 7.10. Table 7.8 shows the information needed to record the resource requirements of the parallel schedule of Table 6.4, whose resource profile appears in Fig 6.3.b.

768. The addition to the schedule of a job with non-zero resource requirements[1] will, in general, require four more items to be stored; see Example 7.11. (This requirement is reduced if either or both ends of the job coincide with a time at which a change in requirements occurs; coincidence is likely when scheduling jobs i to start at their earliest time $\sigma(i)$.) Thus, the maximum storage needed to record the resource requirements of a schedule for a project with N jobs is (4N+2) words per resource. The store requirement now depends linearly on the number of jobs in the project, instead of depending linearly on the project horizon; this generally provides a big reduction in the computer memory that is necessary. (See Example 7.12.)

Example 7.11. Suppose that a job j with s(j)=2, d(j)=3 and r(j,1)=3 is added to the schedule whose resource requirements are given in Table 7.8; the resource requirements of the new schedule are given

---

1. The resource requirements of a job are constant over the duration of that job.

in Table 7.9, which requires four more items.

Example 7.12. The projects used in the experiment each had five resources, sixty jobs and PHOR=10000. Storing resource requirements by time units, 50000 words of computer memory would have been necessary. However, by only recording changes in requirements, this was lowered to only 1210 words, a reduction of over 97%.

TIME	RUSE
0	1
1	2
3	1
6	2
8	0

Table 7.8

TIME	RUSE
0	1
1	2
2	5
3	4
5	1
6	2
8	0

Table 7.9

CHAPTER 8. RESULTS

CHAPTER 8. RESULTS

Introduction

801. As has been described in Chapter 7, a number of projects with different characteristics were scheduled, using parallel and sampling methods, with different priority rules. In this chapter, we describe the results of the analysis of these schedules, using both project duration and utilisation factor as measures of scheduling effectiveness. The numerical values of the project durations and the utilisation factors, on which the analyses are based, are tabulated in Annex F.

802. We shall give a brief outline of the results initially, and then describe them in depth. A detailed summary of the results is presented in Chapter 9.

Outline

803. In this section, we give a brief résumé of the results, in order to provide a frame of reference within which the detailed results that follow can be viewed.

804. Parallel methods. The AOV indicated that there were significant differences between the priority rules, and that there were significant interactions between the priority rules and the project characteristics (that is, the comparative behaviour of the rules is affected by the values of the project characteristics). Further analy-

sis indicated that the best rules were 24(FIFS), 18(CUMRED), 15(RPOSWT), 11(ALLSUCC) and 24(FLOAT/SUC), and that these rules were amongst the best for all combinations of project parameter values used in the experiment. Various classes of priority rules were compared; in particular, we note that global rules produce far better schedules than their local equivalents.

805. Serial methods. Although serial methods were not examined explicitly, we have noted that static rules produce the same schedules in both serial and parallel methods. We have the result that serial methods can be just as good as the parallel methods, if the priority rule is chosen appropriately, and more economical of computational effort; rules 18, 15 and 11, amongst the best rules, are static rules. We deduce that some previous results related to the comparison of serial and parallel methods have been misleading.

806. Sampling methods. The AOV did not indicate any significant differences between the priority rules, nor any significant interactions between the priority rules and the project parameters. The reasons for this unexpected result are discussed, and further experimental investigation of the sampling methods is described. From this additional investigation, it is concluded that there are significant differences between the samples produced by different rules, and that, for one particular project, rule 16(PROD) produces a sample of schedules with a mean project duration significantly less than that produced by

any other rule tested. The distribution of project durations in a sample of schedules is also discussed.

807. Comparison of methods. The sampling methods are shown to be significantly better than the parallel methods, and a quantitative measure of the difference is obtained. The 99% confidence limits for the percentage relative difference between corresponding results from the two methods are 6% and 11%, that is, the sampling methods, with a sample size of 100, produce results that are between six and eleven per cent better than the corresponding results obtained by the parallel methods, with 99% certainty. In obtaining this quantitative result, we have shown that there is a considerable advantage to be gained in using sampling methods rather than parallel methods for project scheduling, even with sample sizes as small as 100, and we are able to give a numerical value to the savings likely to be made in project cost.

808. Measures of effectiveness. We note that project duration was not as useful a measure of scheduling effectiveness as utilisation factor, in this experiment.

AOV Results for the Parallel Method

809. The AOV described in paras 737 to 742 was performed with both project duration and utilisation factor as the criterion measure; Tables 8.1 and 8.2 give the AOV tables in each case. In the tables are listed the sources of variation, i.e. the elements of the AOV

ANALYSIS OF VARIANCE TABLE

SOURCE	SUMS OF SQUARES	DEGREES OF FREEDOM	MEAN SQUARES	F-RATIOS		PROBABILITIES
R	1.14132E+07	25.000	4.56527E+05	17.869	(25, 200)	.0000
D	2.95200E+08	1.0000	2.95200E+08	119.62	(1, 8)	.0000
S	1.43249E+09	3.0000	4.77498E+08	107.10	(3, 8)	.0000
F	4.97191E+07	1.0000	4.97191E+07	11.152	(1, 8)	.0102
N	3.56659E+07	8.0000	4.45824E+06			
RD	8.02800E+05	25.000	32112	1.3923	(25, 200)	.1103
RS	2.88928E+06	75.000	38524	1.5078	(75, 200)	.0128
RF	2.82250E+06	25.000	1.12900E+05	4.4190	(25, 200)	.0000
RN	5.10980E+06	200.00	25549			
DS	1.31735E+08	3.0000	4.39115E+07	17.794	(3, 8)	.0007
DF	1.62973E+06	1.0000	1.62973E+06	.66040	(1, 8)	.4399
DN	1.97422E+07	8.0000	2.46777E+06			
SF	1.91875E+06	3.0000	6.39584E+05	.14346	(3, 8)	.9310
RDS	1.58174E+06	75.000	21090	.91440	(75, 200)	.6679
RDF	6.47506E+05	25.000	25900	1.1230	(25, 200)	.3196
RDN	4.61284E+06	200.00	23064			
RSF	2.09448E+06	75.000	27926	1.0931	(75, 200)	.3100
DSF	6.89685E+05	3.0000	2.29895E+05	9.31588E-02	(3, 8)	.9617
RDSF	1.70831E+06	75.000	22777	.98757	(75, 200)	.5144

Table 8.1. AOV table with criterion project duration - parallel methods.

ANALYSIS OF VARIANCE TABLE

SOURCE	SUMS OF SQUARES	DEGREES OF FREEDOM	MEAN SQUARES	F-RATIOS		PROBABILITIES
R	3875.2	25.000	155.01	30.198	(25, 200)	.0000
D	3128.7	1.0000	3128.7	21.531	(1, 8)	.0017
S	30797	3.0000	10266	141.26	(3, 8)	.0000
F	42423	1.0000	42423	583.76	(1, 8)	.0000
N	581.38	8.0000	72.673			
RD	240.93	25.000	9.6372	2.1464	(25, 200)	.0020
RS	550.78	75.000	7.3438	1.4307	(75, 200)	.0260
RF	300.78	25.000	12.031	2.3439	(25, 200)	.0006
RN	1026.6	200.00	5.1329			
DS	2135.4	3.0000	711.80	4.8983	(3, 8)	.0322
DF	32.402	1.0000	32.402	.22298	(1, 8)	.6494
DN	1162.5	8.0000	145.31			
SF	758.84	3.0000	252.95	3.4806	(3, 8)	.0703
RDS	412.67	75.000	5.5022	1.2255	(75, 200)	.1347
RDF	99.883	25.000	3.9953	.88983	(25, 200)	.6194
RDN	897.99	200.00	4.4900			
RSF	388.29	75.000	5.1772	1.0086	(75, 200)	.4706
DSF	291.99	3.0000	97.329	.66978	(3, 8)	.5940
RDSF	391.53	75.000	5.2205	1.1627	(75, 200)	.2051

Table 8.2. AOV table with criterion utilisation factor - parallel methods.

model given in para 740, and for each of these sources is shown:

a. The variation (sum of squares).

b. The degrees of freedom.

c. The variance (mean squares, calculated by dividing the degrees of freedom into the sum of squares).

d. F statistics, where applicable, calculated by dividing the variance attributable to the source by the appropriate error variance.

810. The error variances used in calculating the F statistics are determined by the nature of the AOV model and the kind of factors involved. Here, the replication factor N is a random factor; the variances for this factor and all interactions involving it are used as error terms. For example, the variance attributable to the RN interaction is the error variance in the F statistic used for testing the priority rules main effect (factor R).

811. The probabilities listed in the tables are the probabilities of the observed values of F, with the degrees of freedom indicated, arising by chance. For example, in Table 8.1, the probability of the value of F corresponding to the factor R (i.e. 17.869 with (25,200) degrees of freedom) arising by chance is 0.000, to four significant figures; in other words, it is virtually certain that the factor R, priority rules, has had a significant effect on the project duration results.

812. Taking 5% as an acceptable level of significance, we have the following results from the AOV tables:

a. For both criteria, priority rules have a significant effect on the results.

b. For both criteria, each of the project characteristics varied in the experiment has a significant effect on the results.

c. There are significant interaction effects between priority rules and each of the project characteristics, when utilisation factor is the criterion, and between priority rules and order strength, and rules and resource factor, when project duration is the criterion. In these cases, changing the value of a project parameter changes the effects of the priority rules, i.e. the rules behave differently at different levels of the particular characteristic.

d. There is a significant interaction between density and order strength; changing the value of order strength has a different effect on the results at the different values of density used in the experiment. The interaction between these two parameters is due to the relationship between free float, and hence density, and the structure of the project graph, outlined in para 716.

e. There are no significant three-way interactions. Even though changing the values of each of the characteristics order strength and resource factor, for example, changes the effects of the priority rules, this result indicates that the changes are in a sense additive, and that the effects which are peculiarly due to, say, order strength are not altered significantly by the value of resource factor.

813. These results will be examined in more detail in the following sections.

Priority Rules with the Parallel Methods - General Results

814. The AOV indicated that the priority rules have a significant effect on the scheduling results with both criteria. Following the procedure detailed in paras 745 to 748, group means X(i....) were calculated for all rules i, for each criterion. Values of X(i....) and s_i, the standard deviation, are given in Table 8.3; Table 8.4 gives the corresponding rankings. (In the table, the 'best' rule, with rank order 1, has the lowest value of the group means with project duration as the criterion, and the highest value with the criterion utilisation factor.)

815. The rules were tested by pairs, using a t Test as described in para 748, to determine the significance of the rankings. In

RULE	PROJECT DURATION		UTILISATION FACTOR	
i	X(i....)	S_i	X(i....)	S_i
1	2864.5	1537.0	55.99	10.83
2	2827.2	1555.6	56.77	9.68
3	2878.0	1515.2	55.53	10.74
4	2849.0	1537.6	56.20	9.44
5	2966.3	1619.3	54.30	10.04
6	3155.7	1695.1	50.88	10.10
7	2983.3	1557.0	53.51	10.39
8	2976.0	1717.3	54.83	10.13
9	2834.8	1566.8	56.67	9.55
10	2905.2	1491.6	54.79	10.99
11	2796.0	1533.0	57.36	9.31
12	3004.0	1581.6	53.38	10.38
13	2795.5	1538.9	57.36	9.24
14	2887.7	1566.9	55.68	9.98
15	2794.2	1581.4	57.74	9.67
16	2815.9	1574.9	57.26	10.26
17	2940.1	1531.3	54.24	10.89
18	2731.6	1486.7	58.61	9.97
19	3144.5	1683.6	51.13	10.82
20	2911.3	1613.9	55.12	9.95

Table 8.3. Group means and standard deviations - parallel methods.

RULE	PROJECT DURATION		UTILISATION FACTOR	
i	X(i....)	S_i	X(i....)	S_i
21	3087.4	1649.7	52.23	10.27
22	3091.2	1652.9	51.89	11.08
23	2713.5	1450.9	59.01	11.78
24	2797.6	1532.1	57.33	9.36
25	2848.0	1520.3	56.32	10.78
26	2893.8	1539.2	55.23	11.72

Table 8.3. Group means and standard deviations - parallel methods (contd.).

RANK ORDER	RULE	
	PROJECT DURATION	UTILISATION FACTOR
1	23	23
2	18	18
3	15	15
4	13	13
5	11	11
6	24	24
7	16	16
8	2	2
9	9	9
10	25	25
11	4	4
12	1	1
13	3	14
14	14	3
15	26	26
16	10	20
17	20	8
18	17	10

Table 8.4. Rules ranked under each criterion measure - parallel methods.

RANK ORDER	RULE	
	PROJECT DURATION	UTILISATION FACTOR
19	5	5
20	8	17
21	7	7
22	12	12
23	21	21
24	22	22
25	19	19
26	6	6

Table 8.4. Rules ranked under each criterion measure - parallel methods (contd.).

this case, there are 62 degrees of freedom for the t statistic, and the critical values are 2.388 (for a significance level of 0.01) and 1.670 (for a significance level of 0.05). No significant differences were observed between any pairs of rules when project duration was the criterion; this occurred because the range of values of project durations is wide, and the variance of each group is large, compared with the differences between group means. Significant differences between pairs of rules when utilisation factor is the criterion are shown in Table 8.5.

Example 8.1. When project duration is the criterion, comparing the group means for rules 6 and 23, we have

$$t = \frac{|3155.7 - 2713.5|}{(((1695.1)^2 + (1450.9)^2)/32)^{\frac{1}{2}}}$$

$$= 1.121.$$

This is less than 1.670, the critical value for the 0.05 significance level, and so the difference between the group means is not significant at this level. However, when utilisation factor is the criterion, we have

$$t = \frac{|50.88 - 59.01|}{(((10.10)^2 + (11.78)^2)/32)^{\frac{1}{2}}}$$

$$= 2.964,$$

which exceeds 2.338, the critical value for the 0.01 significance level. Thus, the difference between the group means of rules 6 and 23, when utilisation factor is the criterion, is significant at the

	2 3	1 8	1 5	1 3	1 1	2 4	1 6	2 9	2 5	2 4	1 1	1 4	2 3	2 6	2 0	1 8	1 0	1 5	1 7	1 7	1 2	2 1	2 2	1 9	6
23																		5	5	5	5	1	1	1	1
18																		5	5	5	5	1	1	1	1
15																				5	5	5	5	1	1
13																						5	5	1	1
11																						5	5	1	1
24																						5	5	1	1
16																						5	5	5	1
2																						5	5	5	5
9																						5	5	5	5
25																								5	5
4																							5	5	5
1																								5	5
14																								5	5
3																									5
26																									
20																									5
8																									
10																									
5	5	5																							
17	5	5																							
7	5	5	5																						
12	5	5	5																						
21	1	1	5	5	5	5	5	5	5																
22	1	1	5	5	5	5	5	5	5		5														
19	1	1	1	1	1	1	5	5	5	5	5	5	5												
6	1	1	1	1	1	1	1	5	5	5	5	5	5	5		5									

Table 8.5. Significant differences between rules when utilisation factor is the criterion - parallel methods. (For example, rule 23 is better than rule 6 at the 1% level, better than rule 7 at the 5% level, and not significantly better than rule 10 at the 5% level.)

1% level.

816. From Table 8.5, we note that rule 23(FIFS), which makes most use of dynamic information, and rule 18(CUMRED), which makes most use of the information about the project's resource requirements, are very good rules. Rules 6(SPT) and 19(1/RPOSWT), at the bottom of the ranking, are very poor. A detailed discussion of the priority rules and their comparative behaviour will be given in a later section, after we have considered the other significant main effects and interactions of the AOV model. (It should be noted that Tables 8.4 and 8.5 do not give the same results in different forms. In Table 8.4, rules are ranked according to the group means X(i....); in Table 8.5, the differences between rules depend not only on the means, but also on the variances s_i^2 of the groups. For example, rules 3, 26 and 20 are respectively in rank order positions 14, 15 and 16 when utilisation factor is the criterion; comparing these rules with rule 6, we obtain values of t of 1.784, 1.590 and 1.692. Of these values, the first and third indicate significant differences at the 5% level; the second value is smaller, and not significant at the 5% level, reflecting the larger variance of the results obtained with rule 26.)

817. In para 749, we discussed the assumption of homogeneity of variance which was made in the formulation of the t Test used to obtain significant differences between rules. With the criterion utilisation factor, and referring to Table 8.3, we find that the

maximum and minimum standard deviations are 11.78 and 9.24 respectively; thus the maximum value of the F statistic defined in para 749 is

$$F = (11.78)^2/(9.24)^2$$

$$= 1.63,$$

with (31,31) degrees of freedom. This is the worst case, but it is not significant at the 5% level, so we accept the assumption for all pairs of rules.

Density Main Effect - Parallel Methods

818. The parameter density has a significant effect on the scheduling results with both criteria. The group means X(.i...) and the corresponding standard deviations are shown in Table 8.6. From this table, we see that with increasing density:

a. The mean project duration and the standard deviation of the project durations are increased.

b. The mean utilisation factor increases, and the standard deviation of the utilisation factors decreases.

819. The lower project durations at the lower value of density reflect the flexibility in scheduling provided by high free float, which allows jobs to be scheduled over a wider range of times without affecting the remaining jobs in the project. If a job must be delayed, because of resource restrictions, then in the case of high density it is more likely to delay other jobs, because of the smaller quantity of free float available. This results in increased project durations,

FACTOR	LEVEL	PROJECT DURATION		UTILISATION FACTOR	
		GROUP MEAN	STD.DEVN.	GROUP MEAN	STD.DEVN.
DENSITY	1	2307.9	970.1	53.42	10.87
	2	3499.2	1781.2	57.30	9.49
ORDER STRENGTH	1	5005.4	1411.2	63.44	6.71
	2	2917.8	602.9	57.57	8.75
	3	2184.8	495.6	53.75	8.91
	4	1506.3	391.3	46.68	9.09
RESOURCE FACTOR	1	3148.0	1563.7	48.22	7.87
	2	2659.1	1503.4	62.50	7.18

Table 8.6. Group means and standard deviations - parallel methods.

with a wider range (and hence a larger standard deviation).

820. The flexibility in scheduling at low density results in inefficient use of resources, and hence lower utilisation factors. The lower standard deviation of utilisation factors at high density will be discussed in the section on the rules by density interaction (para 836).

Order Strength Main Effect - Parallel Methods

821. The parameter order strength has a significant effect on the scheduling results with both criteria. The group means X(..i..) and the corresponding standard deviations are shown in Table 8.6. From this table, we see that with increasing order strength:

a. The mean project duration and the standard deviation of the project durations are decreased.

b. The mean utilisation factor is decreased, but the standard deviation of the utilisation factors is increased.

822. We have already seen, in Examples 7.1, 7.2 and 7.3, how the order strength is related to the structure of the project graph; at low values of order strength, there are fewer ordering relations, and the graph contains more parallel paths than at higher values of S, when it is much narrower. Since the CP duration, CPDUR, is fixed at 1000 for all projects, increasing the order strength, and so decreasing the number of parallel paths in the project graph, has

the effect of reducing the mean duration of the jobs in the project.

Example 8.2. Fig 8.1 shows two project graphs, G and G', both with the same number of jobs (N=6), the same critical path duration (CPDUR=60), and the same density (D=7/8); Table 8.7 gives details. Graph G has a relatively high order strength (S=14/15), and the mean duration of its jobs is 11.7. Graph G', on the other hand, has a lower order strength (S=9/15), and the mean duration of its jobs is 17.5.

823. For projects with the same value of resource factor, a project's work content is generally proportional to the sum of the durations of its jobs, since resource requirements r(i,k) are allocated by the project generation process in the same range [1,10], for all resources k and all jobs i which require them. A project with a high value of order strength, then, will have a lower mean job duration, and a lower work content, than a project with a low value of order strength (other things being equal). However, the resource strength, rs(k), which measures the number of average activities that could be accommodated within the availability R(k) of resource k, is constant for all projects. The reduced work content at high order strength, together with the constant resource strength, has a number of effects:

a. The scheduled durations of projects with high order strength will be shorter, and their standard deviation

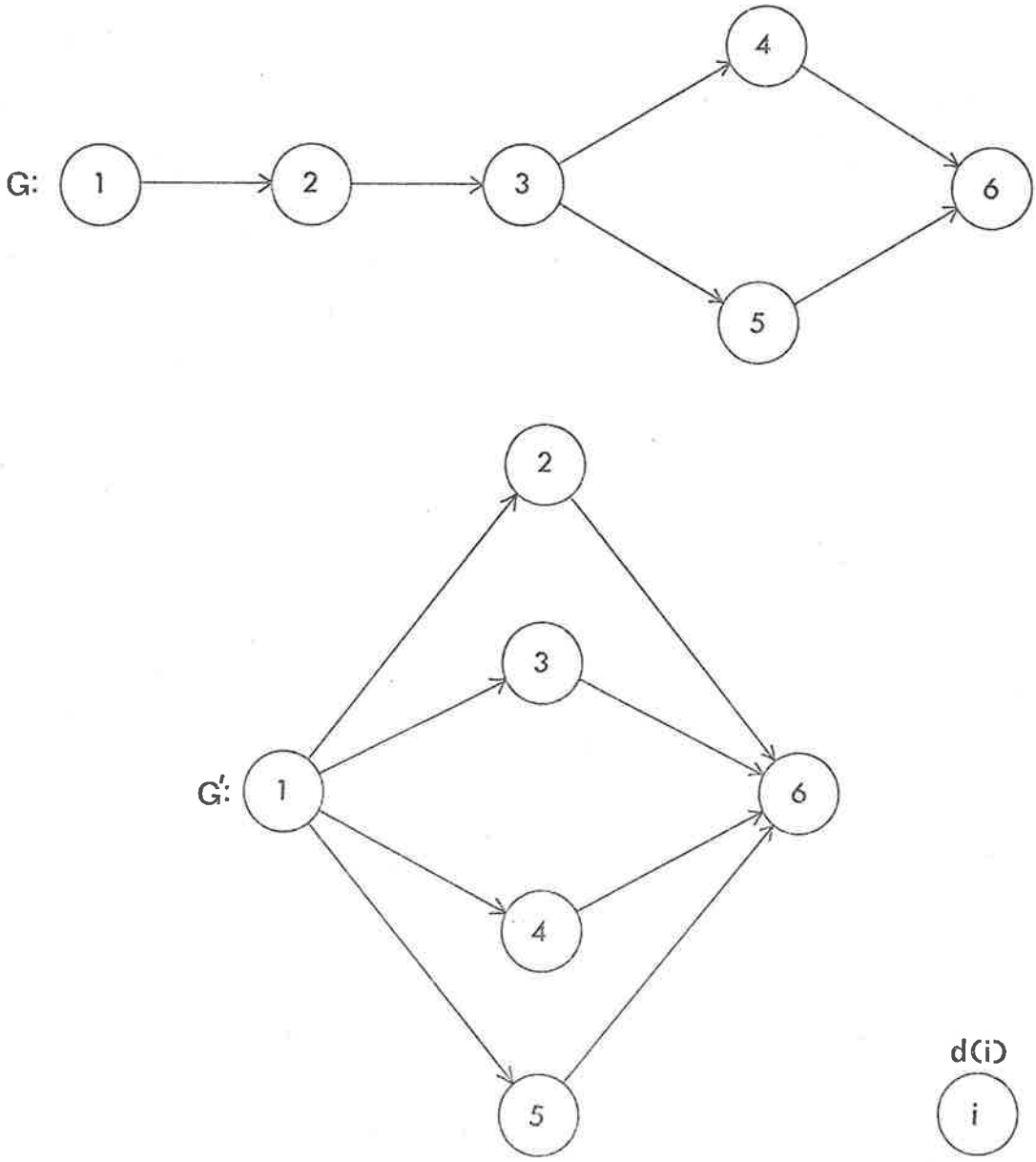

Fig 8.1. Graphs for Example 8.1:

a. For both G and G', N=6, and so N(N-1)/2=15.

b. For G, we have

S = (5+4+3+1+1+0)/15=14/15.

c. For G', we have

S = (5+1+1+1+0)/15=9/15.

G					G'				
i	d(i)	ℓ(i)	d(i)+ℓ(i)	ff(i)	i	d(i)	ℓ(i)	d(i)+ℓ(i)	ff(i)
1	10	0	10	0	1'	20	0	20	0
2	10	10	20	0	2'	20	20	40	0
3	10	20	30	0	3'	15	20	35	5
4	10	30	40	10	4'	20	20	40	0
5	20	30	50	0	5'	10	20	30	10
6	10	50	60	0	6'	20	40	60	0
	70			10		105			15

Table 8.7. Details of projects in Fig 8.1. For both G and G', CPDUR = 60.

a. For G, D = 70/(70+10) = 7/8,

$\bar{d}$ = 70/6 = 11.7.

b. For G', D = 105/(105+15) = 7/8,

$\bar{d}$ = 105/6 = 17.5.

smaller, than for projects with low order strength.

b. The resource availabilities, R(k), determined from the resource strengths, are less restrictive at high order strength. This means that resources are not used to their fullest extent, and there is a decrease in the mean utilisation factor. (For example, at S=0.75, the highest value of order strength used, there were some optimal schedules generated, with PDUR=CPDUR=1000. In these cases, the utilisation factor was only 46.7%.)

824. The order strength also has a more direct effect on the utilisation factors. Because the number of parallel paths in the project graph is reduced at high values of order strength, the number of jobs in the set AV at any stage is likely to be less, on average, at these values. Thus, the range of choice at any stage in the scheduling process is small, for high order strength, and the number of possible schedules is limited; this lack of scheduling flexibility results in less efficient use of resources, and lower utilisation factors with greater standard deviations.

Resource Factor Main Effect - Parallel Methods

825. The parameter resource factor has a significant effect on the scheduling results with both criteria. The group means X(...i.) and the corresponding standard deviations are shown in Table 8.6. From this table, we see that at the higher value of resource factor

we have:

a. A shorter mean project duration and a higher mean utilisation factor.

b. A reduction in the standard deviations of both the project durations and the utilisation factors.

826. These effects are related to the resource strength, rs(k), which was fixed for all resources k, and which was used to determine the availability R(k) of resource k (para 736). Projects with low values of resource factor will have smaller overall requirements for resources (since jobs use, on average, only half of the available kinds of resources for rf=0.5), and hence their resource availability will be less, than projects with high values of resource factor. However, individual jobs have requirements for resources in much the same range (i.e. [1,10]), irrespective of the resource factor of the project to which they belong. So we have the situation where, compared with jobs in projects having rf=1.0, jobs in projects with low resource factor (rf=0.5) require on average only half as many kinds of resources, but in relation to the resource availabilities, they require twice the quantity of these resources; in this case, scheduling is more difficult, and we find longer project durations, lower utilisation factors, and greater variability in the schedules obtained, than is the case with projects having high resource factor.

Rules by Order Strength Interaction - Parallel Methods

827. The AOV results indicate that there is a significant interaction between the priority rules and the order strength, for both criterion measures. That is, we expect to find that the priority rules behave differently at different values of order strength.

828. Fig 8.2 shows means X(i.j..) when project duration is the criterion, plotted against rules, ranked in order of decreasing project duration means X(i....) (see Table 8.4). Here we note that as the order strength increases, the differences between rules become smaller: because the limitations on resources are not so restrictive in the case of high order strength, and since the number of possible schedules is smaller (Examples 7.1, 7.2 and 7.3), rules tend to behave similarly. At low order strength, where there are a large number of parallel paths in the project graph (and hence a large set AV at any stage), and where the resource limitations are more restrictive, the differences between rules become more apparent. This is reflected in the larger standard deviation of project durations at low order strength, which we noted in the discussion of the order strength main effect (paras 821 to 824, and Table 8.6).

829. Fig 8.3 shows the interaction terms est RS(LI) for criterion project duration (paras 750 and 751), plotted against the ranked rules. Apart from rules 18 and 23, we note a general tendency for the interaction terms to become smaller as the rules become better overall, i.e.

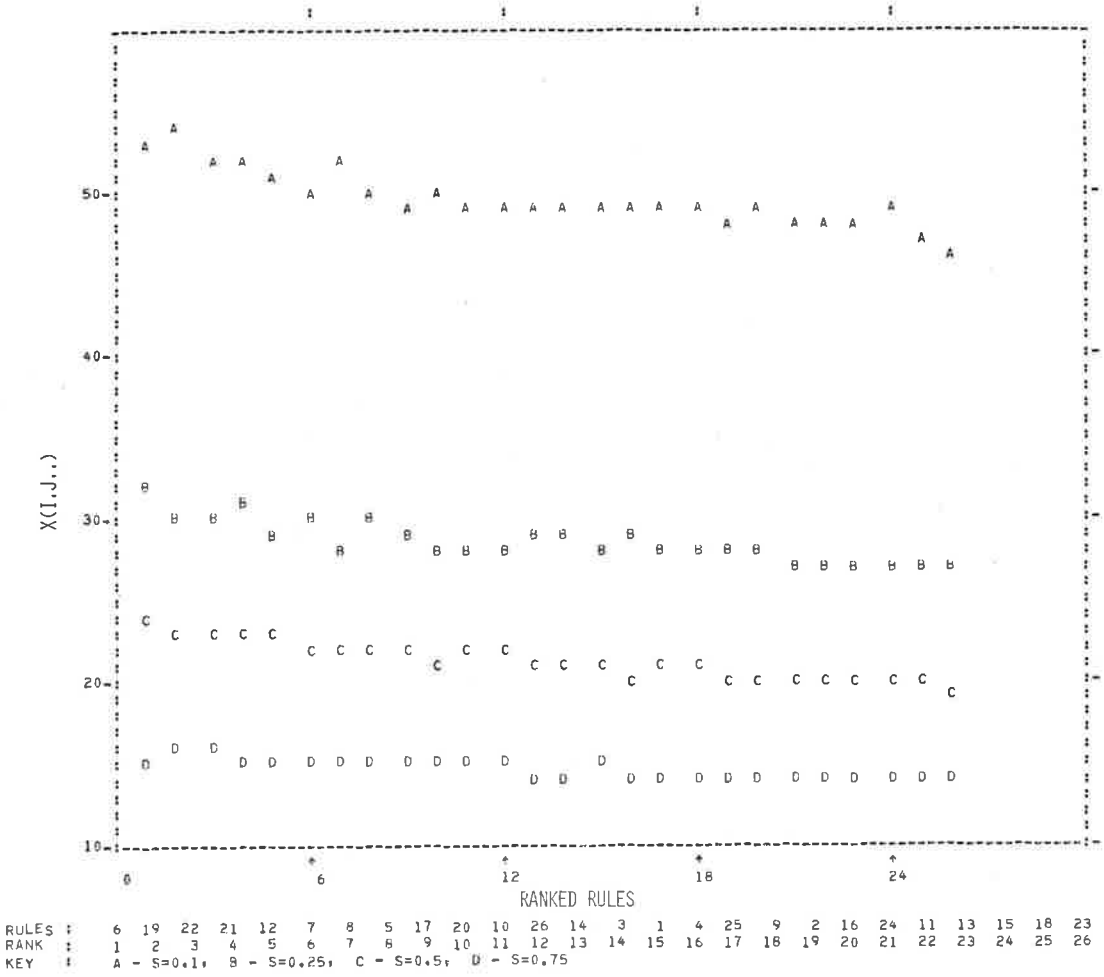

Fig 8.2 Rules of order strength interaction: X(i.j..) for criterion project duration.

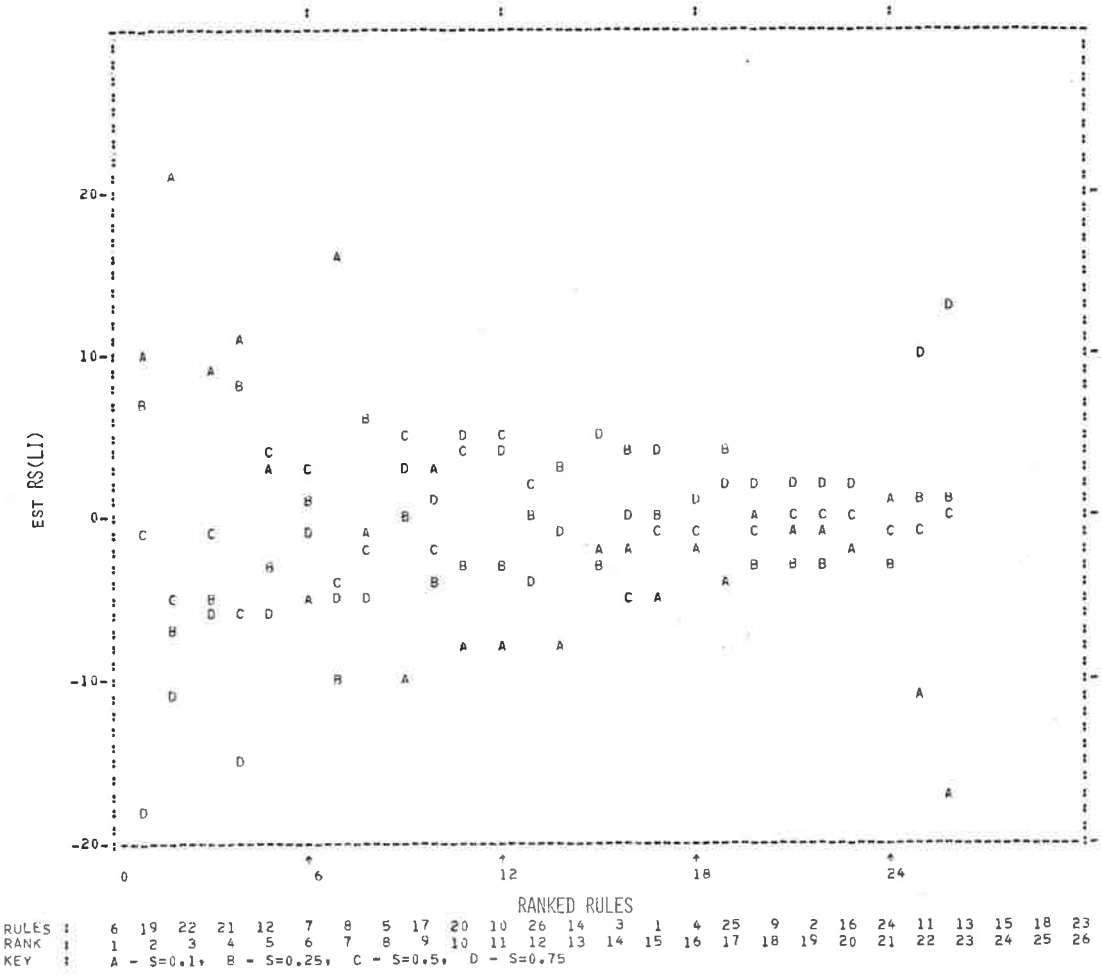

Fig 8.3 Rules by order strength interaction: est RS(LI) for criterion project duration.

the better rules are not affected by variations in order strength as much as those at the lower end of the ranking. Rules 18 and 23 are exceptional: both these rules make a great deal of use of information about resources, rule 23(FIFS) using local but dynamically changing information, and rule 18(CUMRED) using static global data. At low order strength, when resource limitations are most restrictive and the available set AV is generally larger, these rules make better use of the available resource information, and they produce much better results than they normally do. On the other hand, at high order strength (S=0.75), when resources are not as limiting, rules 18 and 23 do not perform as well as we might expect from their overall means X(i....). (But rule 18 is still one of the best five rules at S=0.75.)

830. Fig 8.4 shows means X(i.j..) when the criterion is utilisation factor, plotted against rules, ranked in order of decreasing utilisation factor means X(i....) (see Table 8.4). As order strength decreases, there is a tendency for the differences between most of the rules to decrease, this tendency being reflected in the decreasing standard deviations of utilisation factors (Table 8.6); as the resource restrictions become tighter (S decreases), the majority of the rules, which do not depend on the resource properties of the project, become more nearly the same in their effect on the utilisation factor results. However, the rules at the extremes of the ranking become more extreme, as may be seen by comparing the ranges of values

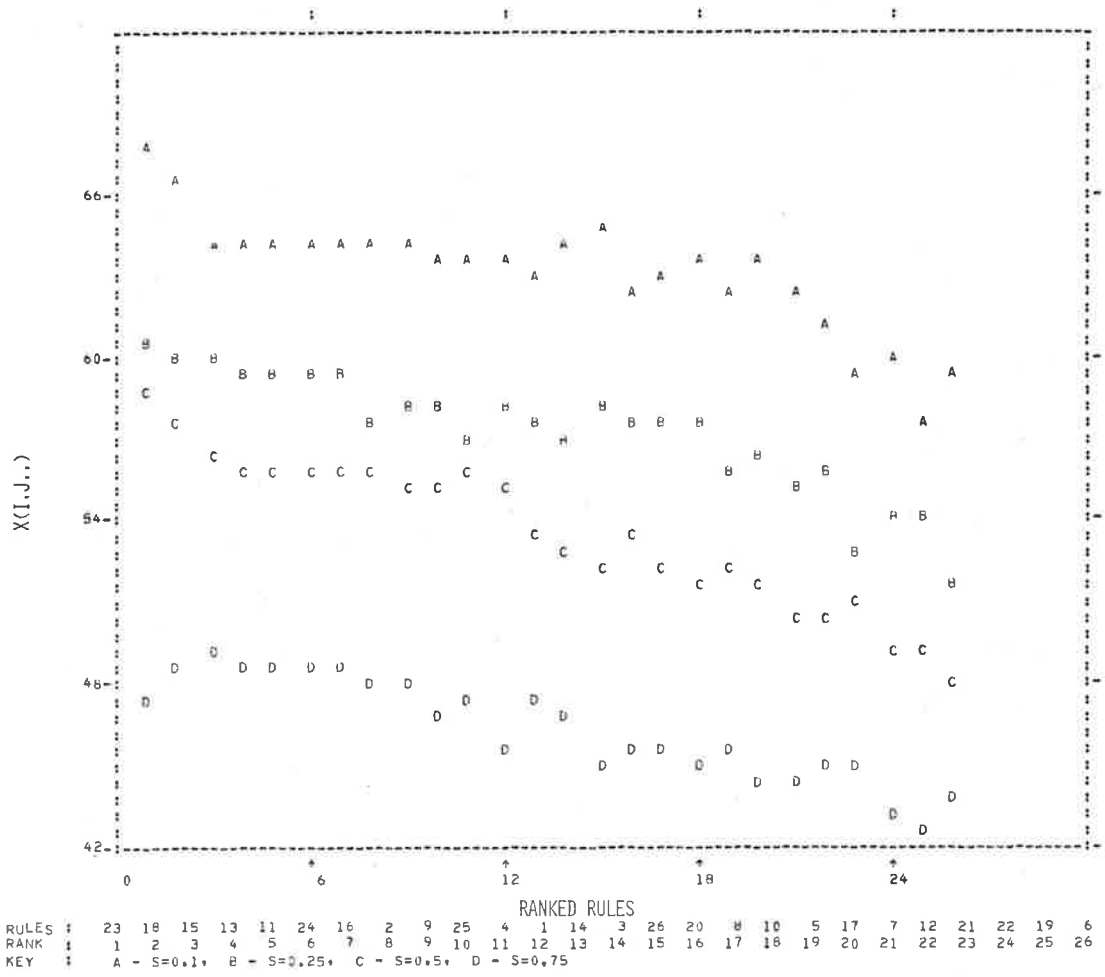

Fig 8.4 Rules by order strength interaction: X(i.j..) for criterion utilisation factor.

of X(i.j..) for S=0.1 and S=0.75 in Fig 8.4; at S=0.1, rules 18 and 23, which use most information about resources, are far better than the rest of the rules. Fig 8.5 shows the interaction term est RS(LI) for this criterion, plotted against the ranked rules; here we see that rules 18 and 23 produce schedules with utilisation factors that are higher than we should expect, from their overall means X(i....) and the order strength means X(..j..), at low order strength, and that the rules next in the ranking (rules 15, 13, 11 and 16) produce schedules that have utilisation factors which are lower than expected. As was the case when project duration was the criterion, rules 18 and 23 do not perform as well as we might expect at high order strength (S=0.75).

831. In summary, at low order strength, when resources are most restrictive, those rules which make most use of information about the resources (rules 18 and 23) perform very well (better than the other rules, and better than they normally perform). However, at high order strength, when resources are relatively less important and the number of possible schedules is smaller, these rules do not perform as well as we might expect; in this case, other considerations are more important. (In fact, rule 15(RPOSWT), which depends on global duration information, seems to be the best rule at S=0.75, but the differences between rules are not statistically significant in general.)

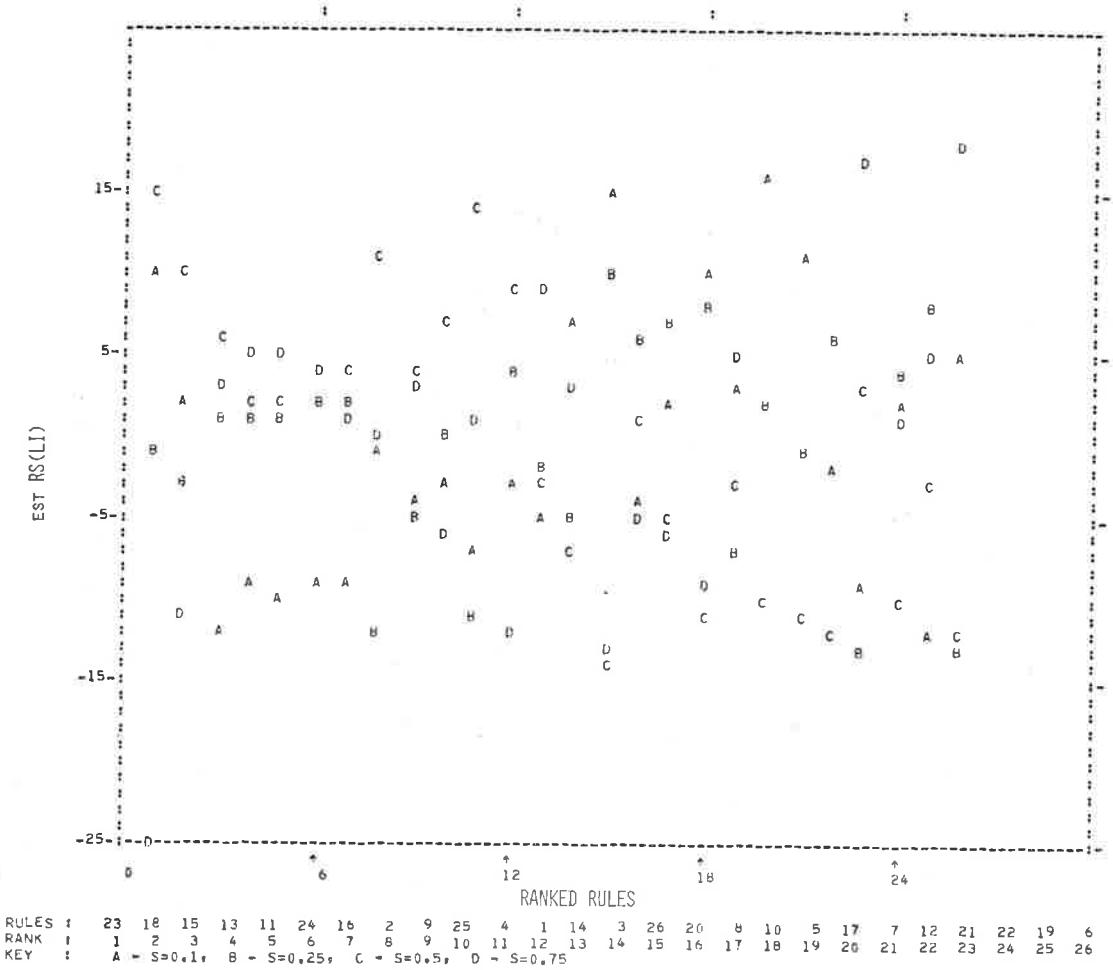

Fig 8.5 Rules by order strength interaction: est RS(LI) for criterion utilisation factor.

Rules by Resource Factor Interaction - Parallel Methods

832. Results of the AOV indicate that the priority rules and the resource factor interact significantly, with each of project duration and utilisation factor as the criterion measure.

833. Fig 8.6 shows means X(i..j.) when project duration is the criterion, plotted against rules, ranked in order of decreasing project duration means X(i....) (see Table 8.4). Here we see that for low resource factor the differences between rules are much greater than for high resource factor. This effect occurs, but not so obviously, when the criterion is utilisation factor (Fig 8.7); in this case, we find that the standard deviation of utilisation factor results, from Table 8.6, is significantly less for rf=1.0 than for rf=0.5. (In fact, using the procedure detailed in para 749,

$$F = (7.87)^2/(7.18)^2$$

$$= 1.20,$$

with (415,415) degrees of freedom, which is significant at the 1% level.) At high resource factor, there is more homogeneity amongst the activities, and hence the rules perform more uniformly. However, at low resource factor, when the scheduling is more difficult (for the reasons noted in the discussion of para 826), there is more opportunity for the differences between rules to become apparent.

834. Fig 8.8 shows the interaction terms est RF(LK) when project duration is the criterion, plotted against the rules, ranked in order

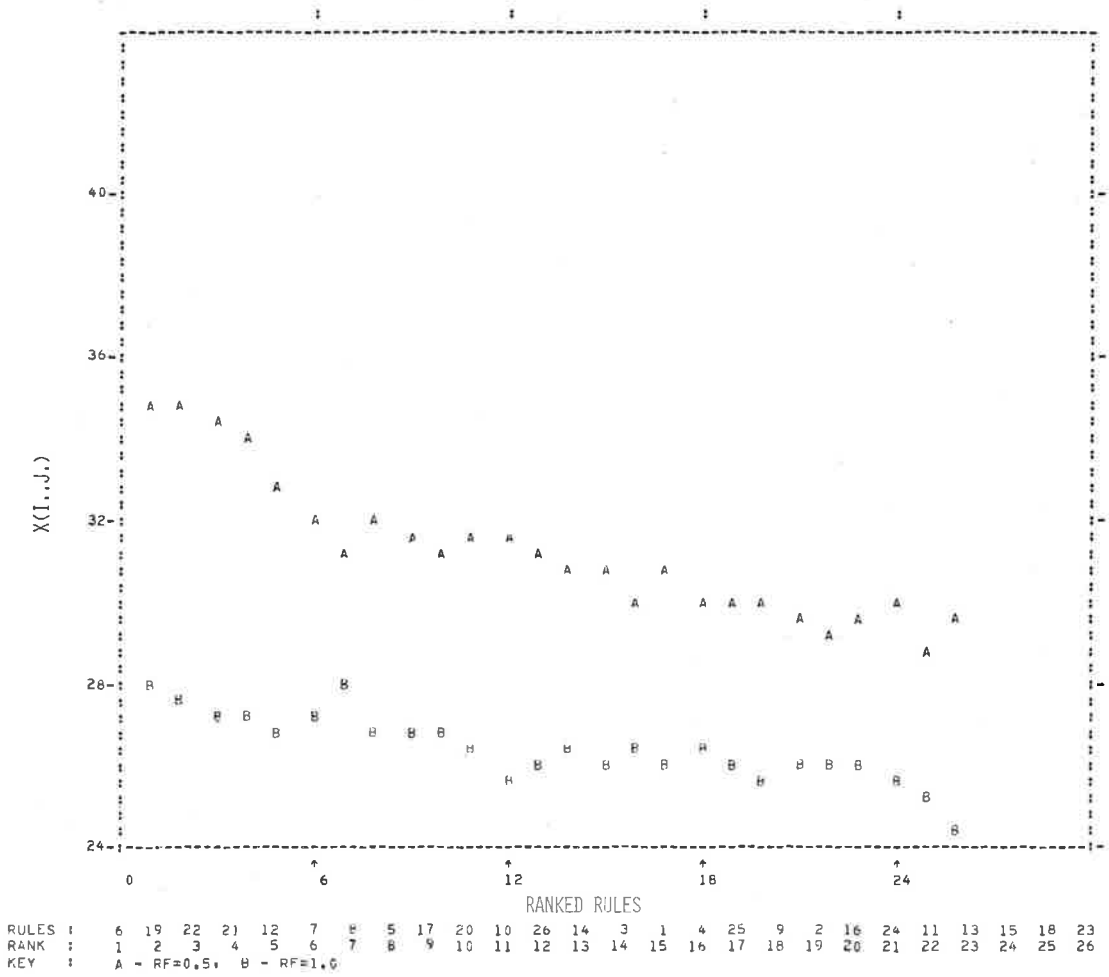

Fig 8.6 Rules by resource factor interaction: X(i..j.) for criterion project duration.

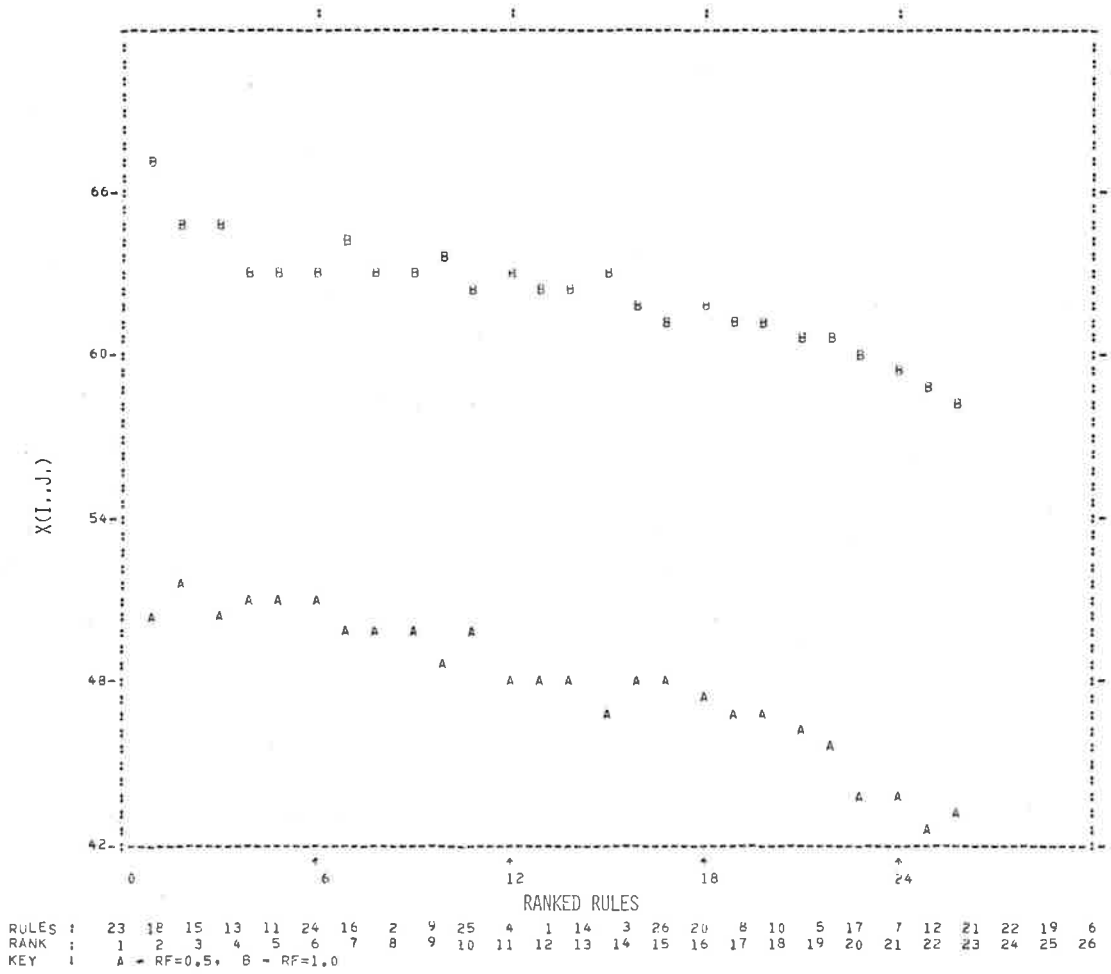

Fig 8.7 Rules by resource factor interaction: X(i..j.) for criterion utilisation factor.

of decreasing project duration means X(i....); Fig 8.9 shows the corresponding terms when the criterion is utilisation factor. For both criteria, there is a tendency for the better rules to show an improved performance, with reference to the overall rule means X(i....), at the lower value of resource factor, and for the rules at the lower end of the ranking to perform better than they do overall when resource factor is high (refer to the definition of est RF(LK) in para 750). Rule 15 (RPOSWT) is affected very little by the resource factor, and performs equally well at rf=1.0,and rf=0.5. Rule 23(FIFS) behaves contrarily; particularly when utilisation factor is the criterion, rule 23 shows improved performance when rf=1.0, i.e. when there are more local resource constraints to be satisfied before a job can be scheduled (see paras 630 and 511); however, it is still amongst the five best rules when rf=0.5.

Rules by Density Interaction - Parallel Methods

835. The AOV results indicate that there is a significant interaction between the priority rules and the density when the criterion is utilisation factor, but not when it is project duration.

836. In Fig 8.10, utilisation factor means X(ij...) are plotted against rules, ranked in order of decreasing rule means X(i....) (i.e. in the rank order shown in Table 8.4, when utilisation factor is the criterion). Although the range of the means in Fig 8.10 does not

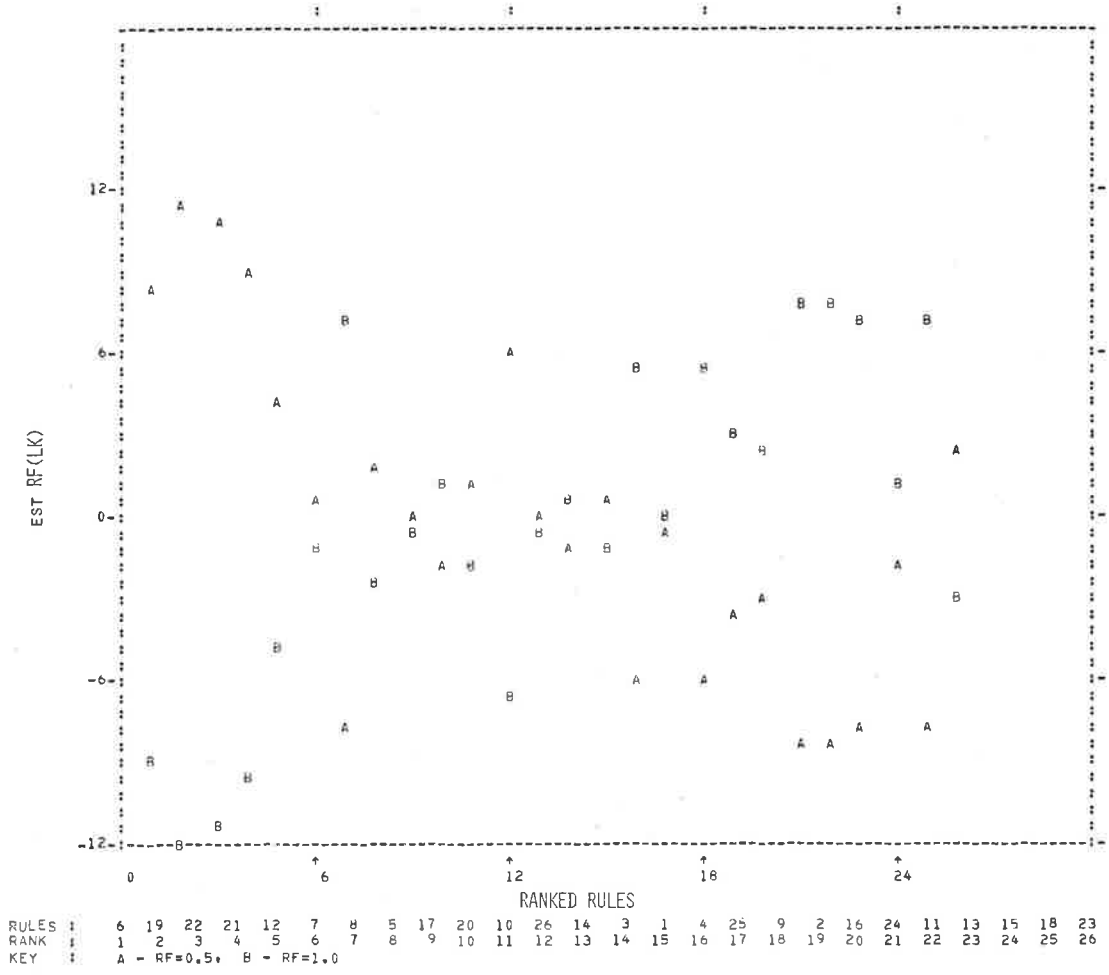

Fig 8.8 Rules by resource factor interaction: est RF(LK) for criterion project duration.

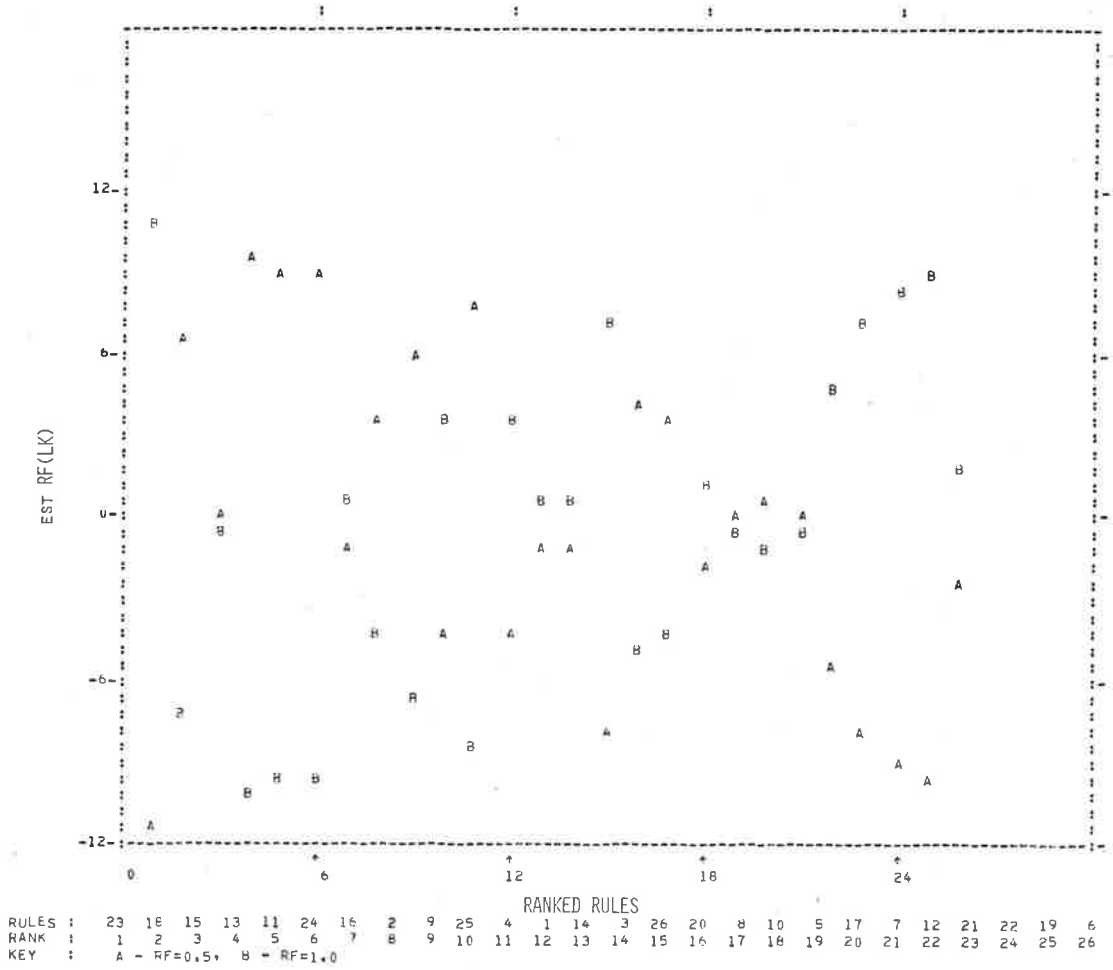

Fig 8.9 Rules by resource factor interaction: est RF(LK) for criterion utilisation factor.

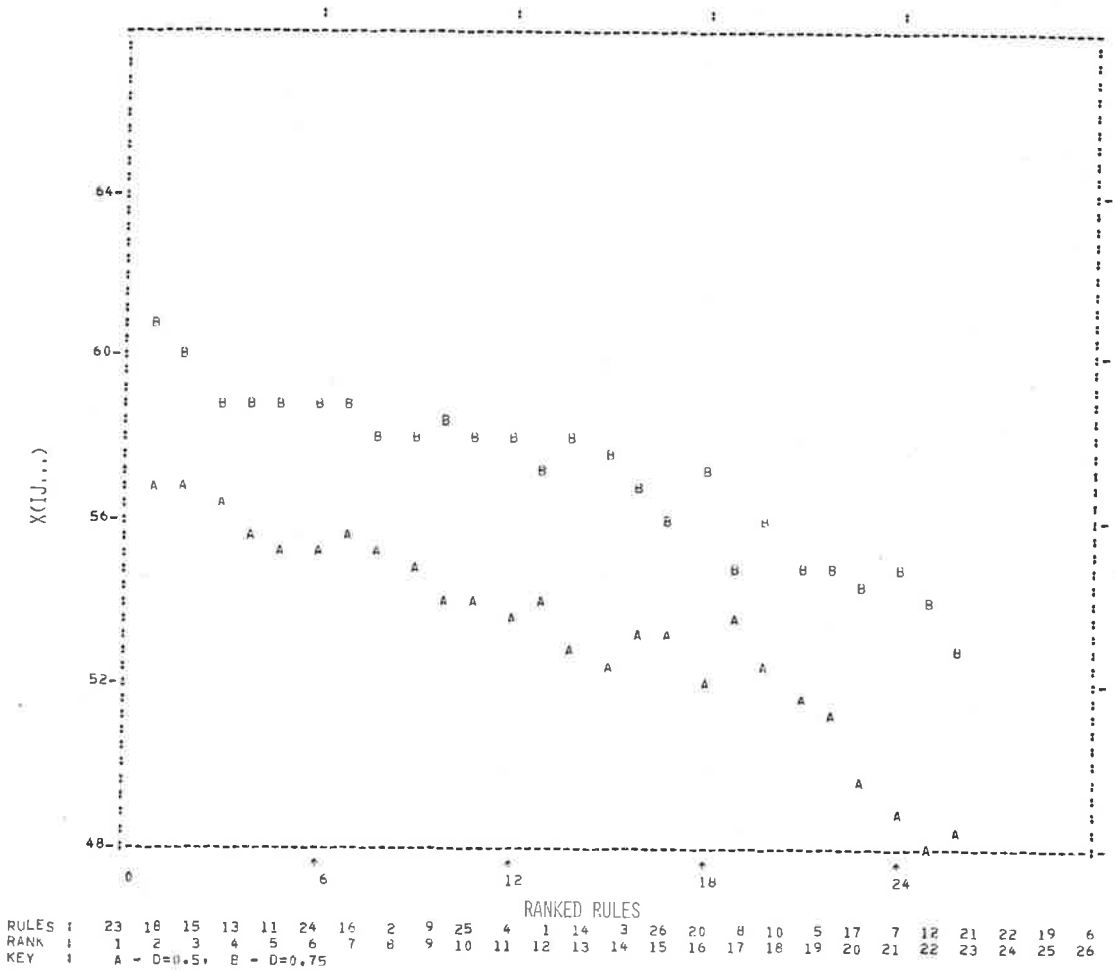

Fig 8.10 Rules by density interaction: X(ij...) for criterion utilisation factor.

differ much from one value of density to the other, we have already noted a difference between the standard deviations of the results at the two values (Table 8.6), which is significant at the 1% level. (In fact, using the procedure detailed in para 749,

$$F = (10.87)^2/(9.49)^2$$

$$= 1.31,$$

with (415,415) degrees of freedom.) At high values of density, there is a reduction in scheduling flexibility due to the smaller quantity of free float available; this gives less opportunity for differences between rules to become apparent.

837. Fig 8.11 shows the interaction terms est RD(LJ) plotted against the ranked rules. Here we see the greatest interaction effect with rule 5(FLOAT): with low free float (high density), sub-critical jobs become relatively more important than is the case when free float is high (low density); a rule which consistently selects against these jobs does not produce such good schedules at high density as at low density. Amongst the better rules, the tendency is for performance to be improved at low density, when the increased flexibility provided by the greater quantity of free float allows the good rules to produce better schedules than would be expected from their overall means. Rule 23(FIFS) is exceptional: it performs better when density is high; the dynamic resource information on which this rule is based is not as important at lower density, because of the increased free

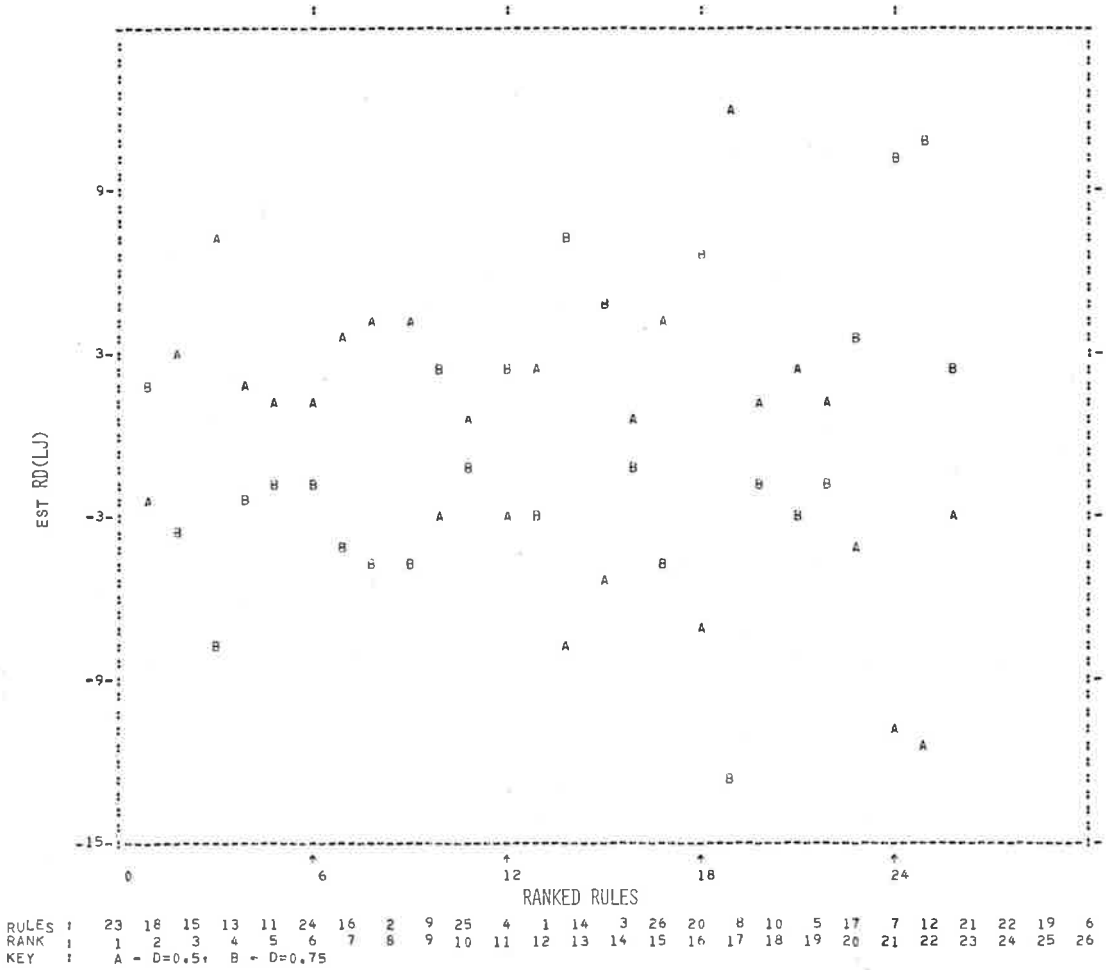

Fig 8.11 Rules by density interaction: est RD(LJ) for criterion utilisation factor.

float. Rule 15(RPOSWT) is also sensitive to the value of density, performing well at the low value.

Priority Rules with the Parallel Methods - Detailed Discussion

838. In this section we shall look more closely at the priority rules. Rather than discuss each individual rule separately, we shall first examine some different classes of rules, and then return in detail to a few rules of special interest.

839. Duration-based rules. Table 8.8A shows the principal duration-based rules used in the experiment, with their ranks according to each criterion (from Table 8.4). Generally, rules based on the selection of long jobs (rule 7,LPT), or jobs with long successors (rule 15,RPOSWT), are better than the corresponding rules based on the selection of short jobs (rules 6,SPT, and 19,1/RPOSWT). This result is also true in assembly line balancing (see, for example, Tonge, 1965, Fig 6), but in job shop scheduling there are some occasions, depending on the scheduling objectives, where the selection of short jobs first is better (and sometimes optimal; see Conway and Maxwell, 1962). In the absence of other information, RPOSWT(i) is a measure of the work dependent on job i - rules which favour jobs on which the largest quantity of work depends produce good schedules (see also para 850).

840. Rules based on the CP schedule. Rules related to the CP schedule are shown in Table 8.8B. We see that rules based on late

		PRIORITY RULE		RANK ORDER	
				PROJECT DURATION	UTILISATION FACTOR
A	DURATION-BASED RULES	6	SPT	26	26
		7	LPT	21	21
		15	RPOSWT	3	3
		19	1/RPOSWT	25	25
		20	PRPOSWT	17	16
B	RULES BASED ON CP SCHEDULE	1	FCFS	12	12
		2	LS	8	8
		3	EF	13	14
		4	LF	11	11
		5	FLOAT	19	19
		23	FIFS	1	1
		26	FIRSTOUT	15	15
C	FLOAT-BASED RULES	5	FLOAT	19	19
		24	FLOAT/SUC	6	6
D	STRUCTURE-BASED RULES	8	NIMSUCC	20	17
		9	NEXTAV	9	9
		11	ALLSUCC	5	5
		12	NONREL	22	22
		13	NONDEP	4	4
		14	SUCC/LEV	14	13
		21	1/NIMSUCC	23	23
		22	1/ALLSUCC	24	24
		25	LEVELS	10	10
E	RESOURCE-BASED RULES	17	RED	18	20
		18	CUMRED	2	2
		23	FIFS	1	1
		26	FIRSTOUT	15	15

(contd.)

Table 8.8. Classes of priority rules.

		PRIORITY RULE		RANK ORDER	
				PROJECT DURATION	UTILISATION FACTOR
F	LOCAL AND GLOBAL RULES	8	NIMSUCC	20	17
		11	ALLSUCC	5	5
		7	LPT	21	21
		15	RPOSWT	3	3
		17	RED	18	20
		18	CUMRED	2	2
		5	FLOAT	19	19
		24	FLOAT/SUC	6	6

Table 8.8. Classes of priority rules.

start (rule 2,LS) and late finish (rule 4,LF) are better than the corresponding rules based on early start (rule 1, FCFS) and early finish (rule 3,EF), but the differences are small; this result is compatible with results obtained by Pascoe (1965) in a similar, but smaller, experiment. (Battersby, 1967, states that early start "is not a very useful measure of priority, because there is often a flood of initial jobs demanding resources, all having the same earliest start".) These rules do not depend on resources; rule 23(FIFS), which does depend on resources, and corresponds with rule 1(FCFS), is a far better rule. There is not much difference, however, between the corresponding rules 26(FIRSTOUT), which depends on resources, and 3(EF), which does not.

841. Float-based rules. The two rules based on float are rules 5(FLOAT) and 24(FLOAT/SUC) (Table 8.8C). We see that FLOAT/SUC is a better rule than FLOAT, as it more nearly estimates the flexibility of the job in relation to the remainder of the project (refer to the discussion of these two rules in para 631). As noted in para 837, FLOAT is highly sensitive to the density, producing better schedules than would be expected from its overall mean at D=0.5 than at D=0.75; FLOAT/SUC shows the same tendency, but to a lesser degree.

842. Structure-based rules. Table 8.8D shows those rules primarily dependent on the structural properties of the project graph. Choosing a job with a large number of successors (rules 11,ALLSUCC,

9,NEXTAV, and 8,NIMSUCC) is better than choosing a job with a small number of successors (rules 21,1/NIMSUCC, and 22,1/ALLSUCC). The number of successors of a job is a rough measure of the work that depends on that job,and so, as in the case of the duration-based rules, those rules which favour jobs that are likely to release the most work for early scheduling produce better schedules than those rules which tend to defer high work loads to the end of the schedule. Rule 25(LEVELS) is also a crude measure of remaining work, and is a fairly good rule.

843. Rules 11(ALLSUCC) and 13(NONDEP) are virtually the same when they are used in the parallel methods: at any stage, the number of unscheduled jobs is constant for all jobs in the set AV, and so the orderings of jobs by values of the priority calculated by each of these rules are the same. However, at low values of order strength, there are a number of cases in which two or more jobs in AV have the same number of successors; in these cases, the job selected is chosen at random from those jobs with the same priority value, and it is this random selection which produces the minor differences in results between these two rules. Rule 11 is faster than rule 13, since it depends only on static information, and does not require recalculation as the schedule progresses; it can therefore be used in a serial method, producing the same results as it would in a parallel method, but with considerable saving in calculation time. Although these rules sort

jobs in the same priority order in the parallel methods, the relative magnitudes of the priority values are different, and so, for a given set AV, the probabilities of jobs being selected for scheduling next are generally different with the two rules in the sampling methods; see Example 8.3. The differences between these two rules in the sampling methods is the reason for the inclusion of both of them in the experiment, despite their similarity in the parallel methods; we shall see a corresponding effect in reverse in para 872 and Example 8.4, when we discuss rules 2 and 10, which produce similar results in the sampling methods, but different results in the parallel methods.

Example 8.3. Suppose at some stage in the scheduling of a project we have AV={k,j,...}, with 19 unscheduled jobs. If job k has 4 successors, and job j has 9 successors, then the priorities calculated by rules 11 and 13 are as shown in Table 8.9. We note that j is selected in preference to k with both rules in the parallel case. However, in the sampling methods, the relative probabilities of selection of the jobs changes: for rule 11, we have

$$P(k):P(j) = 5:10$$
$$= 1:2,$$

but for rule 13, we have

$$P(k):P(j) = (1/15):(1/10)$$
$$= 2:3.$$

844. Resource-based rules. The rules used in the experiment

i	#(Γ*i)	RULE 11	RULE 13
		P(i)	P(i)
k	4	5	1/15
j	9	10	1/10

Table 8.9. Priorities with rules 11 and 13, when there are 19 unscheduled jobs.

which depend on resources are listed in Table 8.8E. This group contains the two best rules, rules 18 and 23, and one of the worst rules, rule 17. Rule 23(FIFS) makes most use of dynamic resource information, and rule 18(CUMRED) makes most use of the global information about resources required by and available to the entire project; we shall discuss these two rules in detail in paras 846 to 849. Rule 19(RED), however, uses only local information, and is not much better than rule 7(LPT), which uses the duration as a coarse estimate of the work content of a job.

845. Local and global rules. There were a number of pairs of rules used in the experiment in which one rule was a local rule, and depended only on the properties of a job and (sometimes) its immediate neighbours, and the other rule used the same properties, but related the job to the rest of the project. Table 8.8F lists these pairs; we note that the global rules are, in every case, far better than the local rules. In the global rules, peculiarities of the individual jobs are less important than in the local rules, where they often dominate the overall properties of the project as a whole; this means that the scheduling process is oriented more towards project completion than towards individual job scheduling when global rules are used, and better results are obtained. We conclude that global rules should be used where possible, in preference to the corresponding local rules.

846. Rule 23, FIFS. This rule, which selects for scheduling the job that can start first, taking into account all the constraints, is the best rule of those used in the experiment. It uses only local information, but it relates jobs to the current state of the partial schedule, and makes very good use of dynamically changing information. As we have seen, local rules are generally inferior to global rules: a global equivalent of rule 23 is "select the job which will cause the project to finish earliest", i.e. "select the job which will lead to an optimal schedule"; we are using heuristic procedures because optimal methods are impractical, and hence no global equivalent of rule 23 was suitable for use in the experiment. Rule 23 had one disadvantage - the computer time required to form a schedule was considerably larger than that required by any other rule except rule 26(FIRSTOUT), since there are a large number of constraints to be satisfied in the calculation and recalculation of values of $\sigma(i)$; we shall return to this point later in para 853.

847. Rule 23 interacts with the project characteristics, producing better schedules at low order strength (paras 829 and 830), at high resource factor (para 834), and at high density (para 837).

848. Rule 18, CUMRED. This rule selects jobs on the basis of their cumulative resource equivalent duration (para 626), and it is the second best rule used. CUMRED(i) is a measure of the work content of job i and its successor activities; this rule favours jobs with the

greatest amount of work dependent on them. Rule 18 is a static rule, since the value of CUMRED(i) does not change as the partial schedule is expanded. RUD(k), used in calculating RED(i) (para 625), is a coarse lower bound on the project duration, based on the requirements for and availability of resource k (para 624); it was conjectured that in this experiment, where all resources are roughly equivalent, there would be very little difference between values of RUD(k) for different resources k, and that the factor RUD(k)/RUDMAX in the definition of RED(i) would be very nearly 1 for all resources. Thus, it was felt that this factor RUD(k)/RUDMAX was somewhat redundant in this experiment, and that similar results might have been obtained if the resource equivalent duration had been defined alternatively by

$$RED'(i) = \Sigma \frac{r(i,k)}{R(k)} \cdot d(i),$$

where the sum is over all resources k. (This turned out to be the case. A further experiment was performed, in which the 32 projects used in the investigation of the parallel methods were each scheduled with rule 18', the modified version of rule 18 suggested above; project durations and utilisation factors for these 32 schedules are tabulated in Annex G. The rule mean X(i....) lies between the corresponding means of rules 18 and 15, for both criterion measures. Using a Sign Test similar to that detailed in para 761, we find that there is no significant difference between the results produced by this rule and the unmodified rule 18.) This may not be the case where there are heavy demands on a few kinds of scarce resources; in this situation

some bias in favour of jobs on which there is dependent a large workload of scarce resources may be an advantage.

849. As we have already seen, rule 18 interacts with the project characteristics to produce generally better results at low order strength (paras 829 and 830), at low resource factor (para 834), and at low density (para 837).

850. Rule 15, RPOSWT. This rule selects jobs on the basis of their rank positional weight, and is ranked third of the rules used in the experiment. As noted in para 622, RPOSWT(i) is a fair measure of the work dependent on job i, if the resource requirements of the individual jobs are homogeneous overall. Thus, we expect this rule to behave in a similar manner to rule 18, possibly interacting differently with the project characteristics. Except for the utilisation factor results at S=0.1, when it does not perform as well as expected from its overall mean X(i....), rule 15 is relatively insensitive to changes in values of order strength and resource factor; however, it interacts strongly with the parameter density, producing better results at low values (D=0.5). The differences between the interactions of this rule, and those of rule 18, reflect the difference in emphasis between the two: rule 15 is duration dependent, and interacts most strongly with density; rule 18 is oriented more towards resources, and interacts far more with resource factor and order strength, to which is related the restrictiveness

of the resource limitations (para 823); generally, however, rule 18 is the better rule.

851. Rules 13(NONDEP) and 11(ALLSUCC). These rules are respectively fourth and fifth in the ranking. As we noted in para 842, the number of successors of a job is a rough measure of the work that depends on that job; these rules perform similarly to rules 18 and 15, and the results they produce are only slightly worse. Their interactions are like those of rule 18 with respect to resource factor and density, and like those of rule 15 with respect to order strength.

852. Implementation of rules. This experiment was not designed to test the ease of implementation of the priority rules, nor to obtain detailed comparisons of the computer time required to form a schedule using them. Instead, a general-purpose program was written, which would accept the number of a priority rule as input and produce as output a schedule formed using the designated rule in a parallel method, and the rules were implemented within this broad framework. Consequently, schedules were not formed as efficiently on the computer as they might have been if a separate program had been written to form a schedule using each rule, and taking advantage of the specific characteristics of individual rules; the additional programming effort that would have been necessary for this task was considered to be excessive. However, some statements can be made about the behaviour in practice of certain rules.

853. In the experiment, the generation of schedules with rules 23(FIFS) and 26(FIRSTOUT) took far longer than with any of the other rules. This need not have been the case, as the following discussion indicates. Rules 23 and 26 both used $\sigma(i)$, the earliest time at which job i can start without violating the constraints, given the current state of the partial schedule (para 511), in their calculation of a priority P(i) for job i. This involved checking whether, for each time period t commencing at the job's CP early start time $\ell(i)$, job i could be started at time t so that it could be completed in period (t+d(i)-1) without violating the resource constraints, i.e.

$$\sigma(i) = \text{Min}\{t \mid t \geqslant \ell(i), \text{ and } r(i,k) + \Sigma r(j,k) \leqslant R(k,T), \text{ for all T such that } t \leqslant T < t+d(i)\},$$

where the sum is over all jobs j in the partial schedule that are in progress at time T (i.e. jobs j with $s(j) \leqslant T < s(j)+d(j)$). The calculation and recalculation of $\sigma(i)$'s by this method involved a large amount of computation, and rules 23 and 26 took longer to form a schedule than the other rules. We can modify the above definition of σ by observing that if, at some stage, job i has an earliest start time $\sigma(i)=t$, then at no subsequent stage of the scheduling can the earliest start time of i be less than t. Thus, the computational definition becomes

$$\sigma(i) = \text{Min}\{t \mid t \geqslant \ell'(i), \text{ and } r(i,k) + \Sigma r(j,k) \leqslant R(k,T), \text{ for all } T \text{ such that } t \leqslant T < t+d(i)\},$$

where $\ell'(i)$ is the maximum of $\ell(i)$ and any values of $\sigma(i)$ calculated at previous stages of the scheduling procedure. A special-purpose implementation, making use of this new form of the definition of σ, should be considerably more efficient than the one used in the general-purpose program written for this experiment, and the computational requirements of rules 23 and 26 should be of the same order as the requirements for the other rules.

854. It is generally true that rules which are static in nature, and depend on quantities that need be calculated only once at the beginning of the scheduling process, can be used in methods that are faster in terms of computer time than methods needed when rules are used which depend on dynamically changing information. Serial methods are computationally more efficient than parallel methods for such rules.

855. In practice, the cost of implementation of the parallel methods on a computer will be of relatively minor importance. Of far greater concern will be the costs associated with the collection and preparation of the large amount of data necessary for any scheduling to be worthwhile, and the usefulness of the schedule that is produced. Compared with the costs of data acquisition, and the

increases in project cost due to the use of poor scheduling methods, computational costs for the parallel methods are trivial; in most applications, the cost of producing the many reports required by various levels of management for overall project control will be greater than the cost of the computer processor time required to generate the project schedule on which these reports are based.

856. Summary. Rule 23(FIFS) is the best of the priority rules used in this experiment, for the set of projects used. Rules which use a measure of the work dependent on a job as a measure of the job's priority are very good (rules 18, 15, 13,and 11), even if the measure of work is very coarse, as is the case with rules 13 and 11.

857. Perhaps the most interesting result is that rules based on static information can be as good or better than those based on dynamic information, if they are well chosen. We have the result that serial methods using rules 18, 15 or 11, are at least as good as the parallel methods using any rules other than rule 23. Thus, the statement that serial methods are not as good as parallel methods, while it is generally true in the case of rules that have both static and dynamic equivalents, can be highly misleading if it is unqualified by any mention of rules such as CUMRED, RPOSWT and ALLSUCC. In these cases, the serial methods produce the same schedules as the parallel methods, but they are much more economical of computation time, since there is no necessity to recalculate priorities during the scheduling

process.

Parallel Methods - Summary and Conclusions

858. There are significant differences between the priority rules used in the experiment. Rule 23(FIFS), based on dynamically changing resource information, is the best rule overall.

859. Rules which select jobs on which there depends a large amount of work are among the best rules, even if they use a very coarse estimate of a job's work content. Thus, rules 18(CUMRED), 15(RPOSWT) and 11(ALLSUCC), using progressively cruder estimates of work remaining, are better than all other rules tested except FIFS (rules 11 and 13 are equivalent, as explained in para 843). Rule 24(FLOAT/SUC) is also a good rule, and far better than the corresponding rule 5(FLOAT).

860. The effects of the priority rules are generally dependent on the characteristics of the project, but the interaction between specific rules and characteristics depends on the particular combination. The best rules (rules 23, 18, 15, 11 and 24) are good for all combinations of parameter values used in the experiment.

861. Global rules are better than the corresponding local rules, and should be used in preference to them whenever possible.

862. Serial methods with static priority rules can produce schedules that are as good as those obtained by parallel methods, with

a saving in computation time, if the priority rules used are selected carefully.

AOV Results for the Sampling Methods

863. The AOV described in para 755 was performed with both project duration and utilisation factor as the criterion measure; the AOV tables in each case are given in Tables 8.10 and 8.11. Before examining the AOV results in detail, we shall first look at our choice of model.

864. Choice of additive model. As was noted in paras 756 and 757, Tukey's test may be used to test the hypothesis that the additive model is appropriate for the particular data in this experiment. Results of this test, described in para 757, are given in Table 8.12; in this case, with a required significance level of 1%, the critical value of F with (1,12) degrees of freedom is 9.33. For both criteria, the calculated value of F is less than the critical value, and so the hypothesis that the additive model is appropriate is not rejected; we shall use the additive model in preference to the alternative nonadditive model in the AOV. (In Table 8.12, the value of SS_R is obtained from the appropriate AOV table; it is the residual sum of squares.)

865. AOV results. From the AOV tables, we find that neither the priority rules main effect, nor any of the interactions between priority rules and the project characteristics, is significant at the required level of 1%. There are significant effects due to the project

SOURCE	SUMS OF SQUARES	DEGREES OF FREEDOM	MEAN SQUARES	F-RATIOS		PROBS
R	6.4597E+04	13	4.9690E+03	2.3540E+00	(13,13)	.0678
F	3.6205E+06	1	3.6205E+06	1.7152E+03	(1,13)	.0000
D	1.7659E+07	1	1.7659E+07	8.3660E+03	(1,13)	.0000
S	1.8276E+07	1	1.8276E+07	8.6582E+03	(1,13)	.0000
RF	2.8169E+04	13	2.1669E+03	1.0265E+00	(13,13)	.4815
RD	2.2625E+04	13	1.7404E+03	8.2450E-01	(13,13)	.6335
RS	3.8023E+04	13	2.9249E+03	1.3856E+00	(13,13)	.2825
FD	1.0863E+04	1	1.0863E+04	5.1461E+00	(1,13)	.0410
FS	9.1029E+04	1	9.1029E+04	4.3125E+01	(1,13)	.0000
DS	2.1479E+04	1	2.1479E+04	1.0175E+01	(1,13)	.0071
RFD	2.4563E+04	13	1.8895E+03	8.9513E-01	(13,13)	.5776
RFS	3.2483E+04	13	2.4987E+03	1.1838E+00	(13,13)	.3828
RDS	2.0710E+04	13	1.5931E+03	7.5470E-01	(13,13)	.6904
FDS	1.4465E+05	1	1.4465E+05	6.8526E+01	(1,13)	.0000
RESIDUAL	2.7441E+04	13	2.1108E+03			

Table 8.10. AOV table for project duration data from the sampling methods.

SOURCE	SUMS OF SQUARES	DEGREES OF FREEDOM	MEAN SQUARES	F-RATIOS		PROBS
R	3.9865E+01	13	3.0665E+00	2.7814E+00	(13,13)	.0382
F	6.7868E+03	1	6.7868E+03	6.1558E+03	(1,13)	.0000
D	1.1391E+02	1	1.1391E+02	1.0332E+02	(1,13)	.0000
S	7.0958E+02	1	7.0958E+02	6.4361E+02	(1,13)	.0000
RF	1.1968E+01	13	9.2062E-01	8.3503E-01	(13,13)	.6250
RD	1.5774E+01	13	1.2134E+00	1.1006E+00	(13,13)	.4327
RS	2.1040E+01	13	1.6185E+00	1.4680E+00	(13,13)	.2492
FD	2.3415E+01	1	2.3415E+01	2.1238E+01	(1,13)	.0005
FS	4.2083E+00	1	4.2083E+00	3.8170E+00	(1,13)	.0726
DS	6.7522E-02	1	6.7522E-02	6.1245E-02	(1,13)	.8084
RFD	1.4635E+01	13	1.1257E+00	1.0211E+00	(13,13)	.4853
RFS	1.3821E+01	13	1.0632E+00	9.6433E-01	(13,13)	.5256
RDS	1.5579E+01	13	1.1984E+00	1.0870E+00	(13,13)	.4414
FDS	5.5301E-01	1	5.5301E-01	5.0160E-01	(1,13)	.4913
RESIDUAL	1.4332E+01	13	1.1025E+00			

Table 8.11. AOV table for utilisation factor data from the sampling methods.

	PROJECT DURATION	UTILISATION FACTOR	DEGREES OF FREEDOM
SS_N	$6.116*10^3$	4.203	1
SS_R	$27.441*10^3$	14.332	13
SS_B	$21.325*10^3$	10.129	12
F	3.44	4.98	(1,12)

Table 8.12. Details of Tukey's test.

characteristics and their interactions, as we would expect from the discussion of their effects in the case of the parallel methods; the arguments are similar and will not be repeated here.

866. The lack of any significant priority rule effects is of major interest. This result indicates that it does not matter greatly what rule is used in a sampling method; the schedule produced will be much the same for all rules. This was an unexpected result, for which there are a number of possible explanations:

a. There is no difference between the priority rules when used in a sampling method with sample size 100.

b. There is a difference between rules, but the experiment failed to detect it.

867. The first of these reasons was not thought to be plausible, in view of the marked significance obtained in the case of the parallel methods, and the results of Arcus'(1966) investigations. (Consequently, a further set of experiments, described in the next section, was performed to investigate further the processes that occur in the sampling methods.) We are left with the alternative hypothesis, that the experiment failed to detect a difference.

868. There are two factors which might account for the non-detection of a difference that in fact exists: the size of the

experiment, and the experimental procedure with respect to the methods under investigation. The factorial experiment used in the examination of the sampling methods was small, despite the large amount of experimental effort involved (paras 753 and 754); small experiments detect fewer significant effects than large ones, in general, due to the generally large relative magnitude of the error variances. Possibly of more importance was the experimental procedure with respect to the particular methods. As we have noted in para 765, the extrema of samples can be highly variable, even for samples drawn in the same way from the same parent population; it became apparent only when the results were being analysed that to investigate the sampling methods, the schedules produced with particular priority rules applied to particular projects (that is, the best schedule of the 100 schedules generated in each case) were not of prime importance, but of far greater interest were the distributions of the sets of 100 schedules produced as intermediate results[1]. This fact, which was not at all obvious until late in the experimental program, together with the hypothesis that there were significant effects due to priority rules, motivated a further set of experiments, described in the next section.

---

1. Here we are referring to the importance of the best schedule and the set of schedules in this experimental investigation; the practical user of the sampling methods generally will be far more interested in the best schedule obtained than in the distribution of schedules in the sample.

Distribution of Sampling Solutions

869. As we have noted in the previous section, further investigation of the sampling methods was indicated. This took the form of an examination of the intermediate results obtained by the sampling methods, and was intended to determine:

a. Whether or not there was a difference in the sample means between different priority rules.

b. The distribution of schedules in a sample.

870. Because of limitations in the amount of experimental time available, only one project was investigated. This had parameter values density D=0.5, order strength S=0.5, and resource factor rf= 0.5; it was scheduled with a number of priority rules and a variety of sample sizes. For each sample, all intermediate schedules were saved, and their project durations were calculated. (For a single project, project duration is an equivalent measure to utilisation factor; it has the practical advantage of being an integer-valued quantity.)

871. Table 8.13 shows details of the samples. Since the samples are large, the sample means may be compared using a z statistic (Crow et al, 1960, section 2.4), calculated by

$$z = |x_1 + x_2| \Big/ \left\{ \frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2} \right\}^{\frac{1}{2}}$$

where x_i, σ_i^2 and n_i are respectively the mean, variance and sample

RULE	SAMPLE SIZE	MEAN	VARIANCE	BEST
2	200	1544.1	3849.7	1392
10	1200	1544.6	4132.7	1388
11	100	1479.4	2497.4	1350
13	500	1495.5	3211.0	1376
14	100	1481.6	3139.8	1369
15	600	1474.7	2742.7	1335
16	600	1439.0	2315.9	1337
17	100	1582.9	2057.7	1425
18	100	1478.2	3111.5	1375

Table 8.13. Sample results. BEST is the minimum project duration obtained.

size of sample i; if the calculated value of z exceeds the critical value for the significance level of the test, reject the hypothesis that there is no difference between the means of the populations from which the samples were drawn. (This test assumes that the samples are drawn from normal populations. As we shall see in para 875 and Figs 8.12, 8.13 and 8.14, the distribution of project durations in a sample of schedules is very nearly normal.) Table 8.14 shows values of z for all pairs of rules; also shown are significance levels of the differences between the means. (The critical value of z at the 0.001, i.e. 0.1%, significance level is 3.291. The unmarked entries are not significantly different at the 0.2, i.e. 20%, significance level, for which the critical value of z is 1.282. The rules have been ranked in order of increasing sample means.)

872. From Table 8.14, we see that there are differences between the samples selected by different priority rules. In terms of our discussion in para 640, by using different priority rules we in fact do bias the sample of feasible schedules obtained by the sampling methods. For this project, rule 16(PROD), which Arcus (1966) found so successful for the assembly line balance problem, has a lower mean project duration than the other rules tested, and rule 17(RED) has a higher mean than the others. Rules 10(RANDOM) and 2(LS) produce samples of schedules with very similar means; this occurred because the project horizon, PHOR, was large, and hence the values of the CP late start times had relatively similar values for all jobs (para 305); this means

	16	15	18	11	14	13	10	2	17
16		12.295	6.628	7.524	7.174	17.620	39.074	21.863	29.110
15	0.001		0.586	0.865	1.150	6.273	24.690	14.220	21.576
18	0.001			0.160	0.430	2.824	11.295	9.286	14.562
11	0.001				0.293	2.873	12.231	9.729	15.335
14	0.001					2.260	10.673	8.782	14.051
13	0.001	0.001	0.01	0.01	0.05		15.632	9.592	16.820
10	0.001	0.001	0.001	0.001	0.001	0.001		0.105	7.815
2	0.001	0.001	0.001	0.001	0.001	0.001			6.148
17	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	

Table 8.14. Values of z for comparison of sample means (upper half of table) and significance levels of the differences (lower half of table). (For example, comparing the means of samples from rules 18 and 10, z = 11.295, and this is significant at the 0.001 level.) Unmarked entries are not significant at the 0.05 level.

that the job priorities are much the same for the LS rule over all jobs, and we are scheduling with very little bias, as can be seen in the following example.

Example 8.4. Table 3.1 shows CP schedules for the project of Fig 3.1, including CP late start times when PHOR=20 (Example 3.2). Table 8.15 shows the priorities calculated by rule 2 in this case, and in the case where PHOR=1000. It can be seen that, although the priority order remains the same, indicating that the serial and parallel methods are not affected by the change in PHOR, the relative magnitudes change drastically. (Here, we have set $P(i)=1000/(u(i)+1)$; i.e. the constant of proportionality for the definition of Table 6.5 is 1000.) If at some stage in the scheduling we have AV $=\{2,4,6\}$, then with PHOR=20, the corresponding probabilities p(i) of selecting jobs i using rule 2 are

$$p(2) = 0.41,$$

$$p(4) = 0.33,$$

$$p(6) = 0.26,$$

while if PHOR=1000, the corresponding probabilities are very nearly equal. (Here we are evaluating p(i) by the method of para 638.c, i.e.

$$p(i) = \frac{P(i)}{\Sigma P(j)},$$

where the sum is over all jobs j in AV.) Thus, using rule 2 is equivalent to using rule 10, if PHOR is large; for rule 2 to be effective, PHOR should be closer to the final project duration PDUR.

i	PHOR = 20		PHOR = 1000	
	u(i)	P(i)	u(i)	P(i)
1	6	142.9	986	1.013
2	8	111.1	988	1.011
3	7	125.0	987	1.012
4	10	90.9	990	1.009
5	13	71.4	993	1.006
6	13	71.4	993	1.006
7	15	62.5	995	1.004

Table 8.15. Priority values calculated by rule 2(LS).

873. We also note from Tables 8.13 and 8.14 that rules 11(ALLSUCC), 15(RPOSWT) and 18(CUMRED) all have similar means. This confirms the conclusions drawn in the case of the parallel methods, that priority rules based on a measure of the remaining work dependent on a job perform equally well, even if the measure used is fairly coarse (para 859). However, comparing the best schedule obtained in 100 trials for rules 11 and 18, we note a large difference in their project durations, despite the similarity of the sample means; the extreme value (i.e. the best schedule in the sample) is not necessarily a good indicator of the performance of a priority rule.

874. Rules 11 and 13, which produced the same schedules in the parallel methods in most cases, produce samples whose means are significantly different, since the priority values calculated by each of them, while having the same ordering, have different relative magnitudes, as we have noted in para 843 and Example 8.3. The mean project duration of a sample from rule 14(SUCC/LEV) is not significantly different from that of rule 11(ALLSUCC).

875. The differences between the samples produced by some of the rules can be seen in Figs 8.12, 8.13, 8.14 and 8.15. In Fig 8.12, the distribution of project durations with rule 16(PROD) is shown, together with a normal curve having the same mean and standard deviation; in the histogram, the width of the class intervals is half

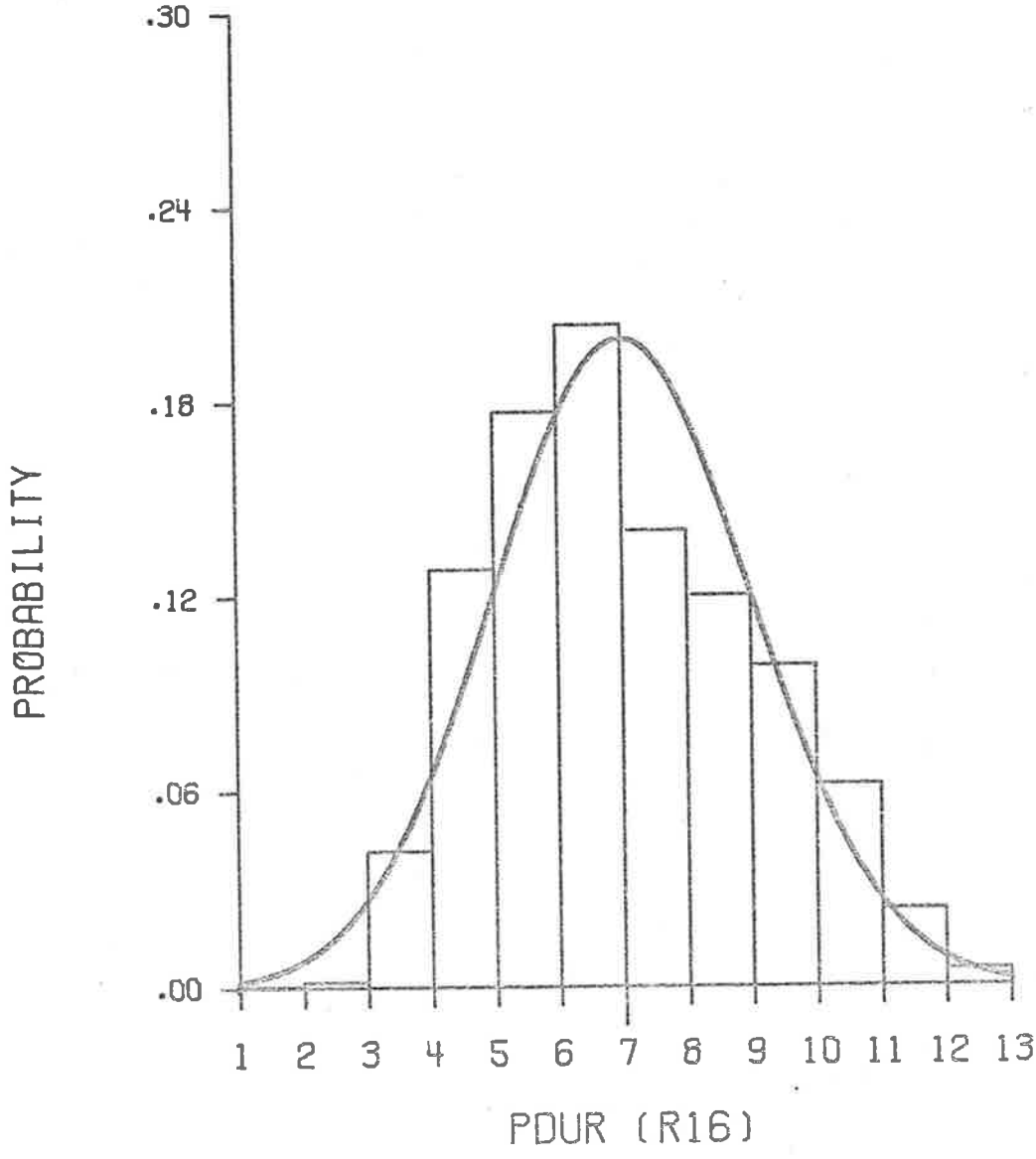

Fig 8.12. Distribution of 600 intermediate schedules with rule 16(PROD).

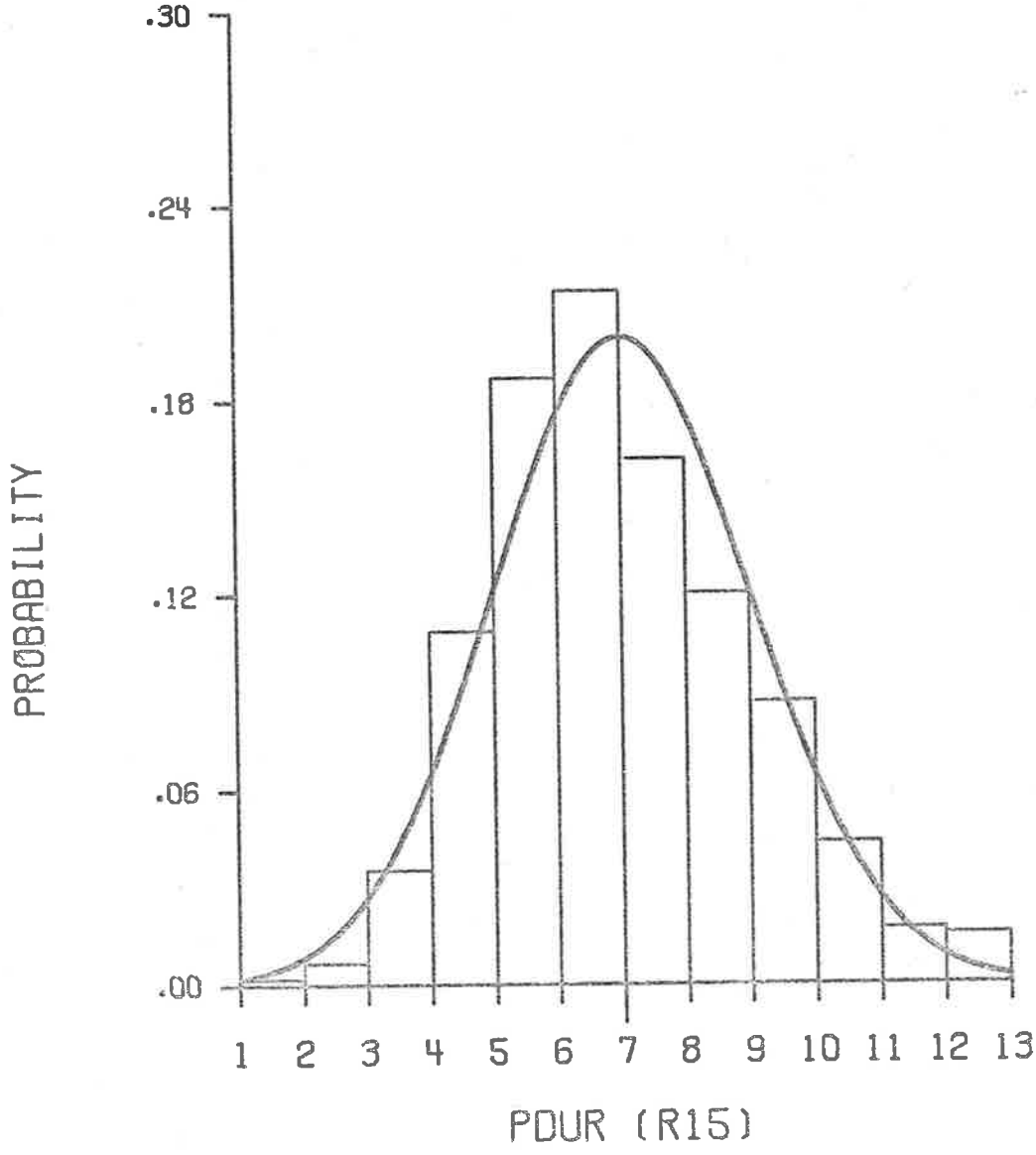

Fig 8.13. Distribution of 600 intermediate schedules with rule 15(RPOSWT).

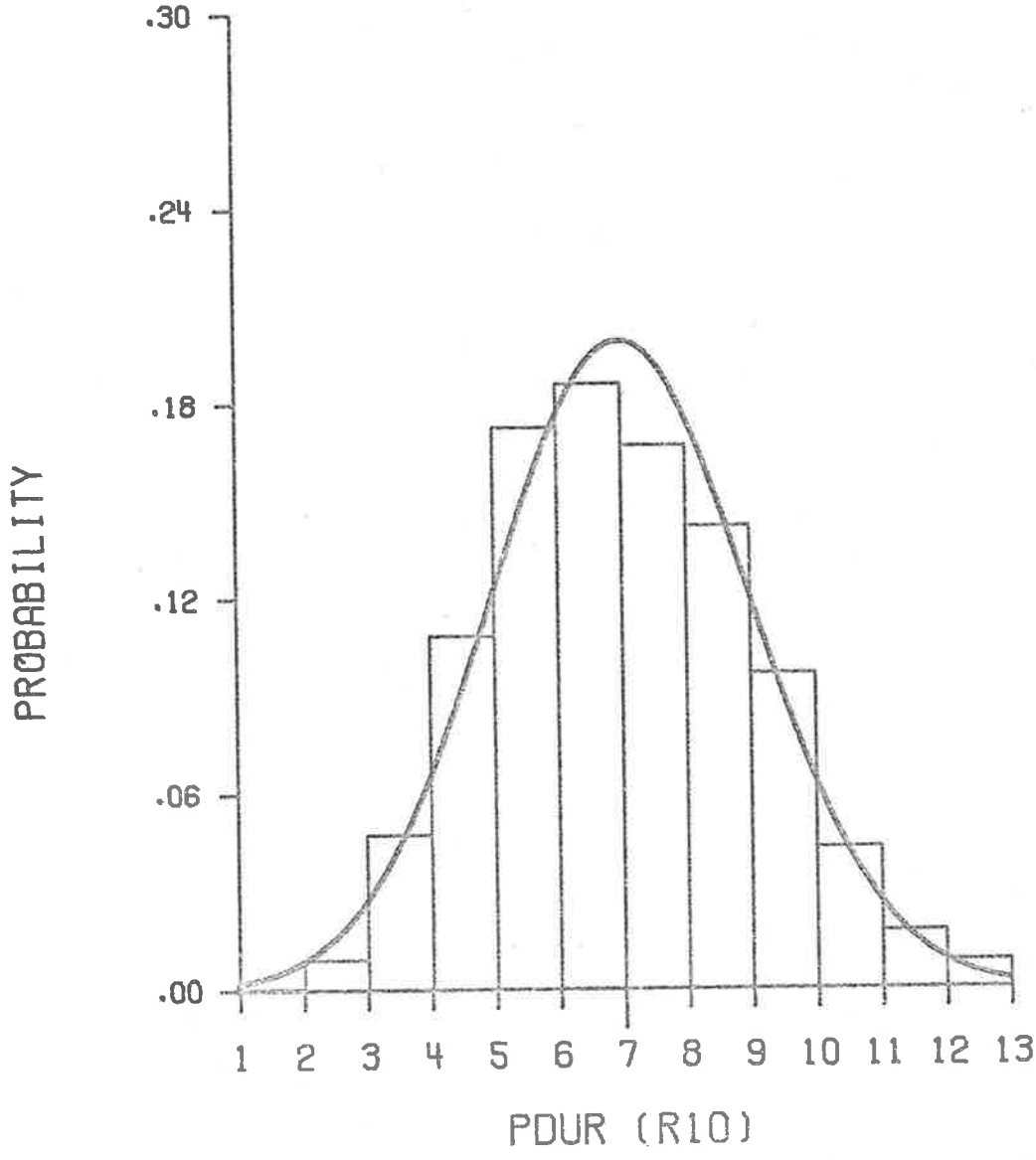

Fig 8.14. Distribution of 1200 intermediate schedules with rule 10(RANDOM).

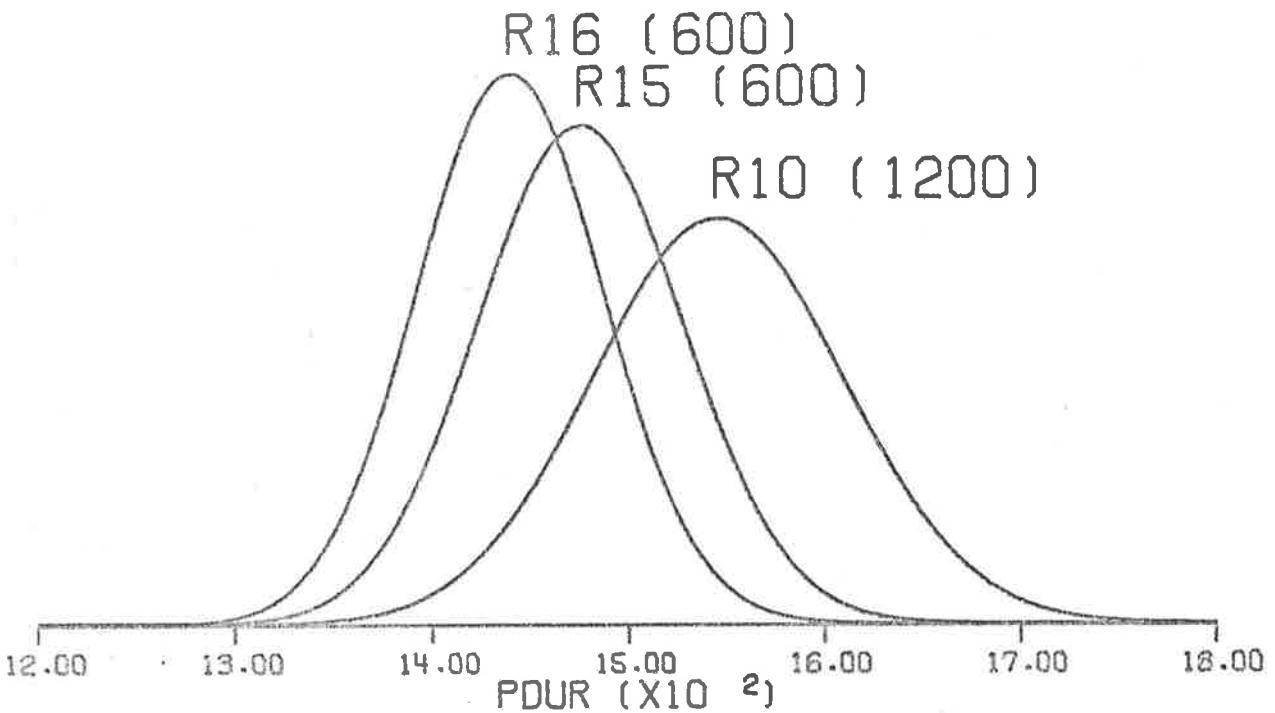

Fig 8.15. Normal approximations from Figs 8.12, 8.13 and 8.14, plotted on the same axis.

the standard deviation, so that point 7 is the mean of the distribution, points 6 and 8 are half a standard deviation from the mean, points 5 and 9 are one standard deviation from the mean, and so on. Figs 8.13 and 8.14 show the distributions of samples from rules 15(RPOWST) and 10(RANDOM); Table 8.16 gives the numerical values of the bounds on the class intervals (i.e. the points 1,2,...,13 on the horizontal axes) in each case. The distributions resemble the normal distribution, but they are positively skewed. Fig 8.15 shows the approximating normal curves plotted on the same axes and to the same scale. The values of project durations for these three samples are given in Appendix H.

Sampling Methods - Summary and Conclusions

876. Although the AOV did not indicate that the priority rules have a significant effect on the results of the sampling methods with a sample size of 100, further investigation of the distribution of scheduling solutions within a sample showed that there were differences between the rules. In the project which was investigated, rule 16(PROD) produced a sample whose mean value was significantly less than that of a sample produced by any other rule examined. We conclude from these results that, despite the lack of detectable significant differences between the extrema of the samples, the priority rules cause the sampling methods to sample from, effectively, different populations; thus, we will expect to obtain

POINT		RULE 16	RULE 15	RULE 10
1	m-3s	1294.7	1317.6	1351.7
2	m-5s/2	1318.7	1343.8	1383.9
3	m-2s	1342.8	1370.0	1416.0
4	m-3s/2	1366.9	1396.2	1448.2
5	m-s	1390.9	1422.3	1480.3
6	m-s/2	1415.0	1448.5	1512.4
7	m	1439.0	1474.7	1544.6
8	m+s/2	1463.1	1500.9	1576.7
9	m+s	1487.2	1527.1	1608.9
10	m+3s/2	1511.2	1553.3	1641.0
11	m+2s	1535.3	1579.5	1673.2
12	m+5s/2	1559.4	1605.6	1705.3
13	m+3s	1583.4	1631.8	1737.4
MEAN, m		1439.042	1474.708	1544.586
STANDARD DEVIATION, s		48.124	52.371	64.286
OUT OF RANGE		0	3	3

Table 8.16. Bounds on the class intervals.
(m is the mean, and s the standard deviation of the distribution.)

better results in the long term by using rules which sample from populations with the lowest mean project duration.

877. We have seen in Example 8.4 how the priorities calculated by rule 2(LS) depend on the project horizon. It is concluded that if this rule is to be effective, the project horizon must be closer to the actual project duration; otherwise a simple random rule (rule 10) produces similar results with a slight reduction in computational effort.

878. As was the case with the parallel methods, rules 11(ALLSUCC), 15(RPOSWT) and 18(CUMRED), all of which use priorities based on estimates of the work dependent on a job, perform in a similar way to one another. For scheduling purposes, these rules are equivalent when used in a sampling method.

879. The overall conclusion from the examination of the sampling methods is that, although no significant differences between the best schedules produced by different priority rules were detected in this experiment and for this sample size, there may be benefits in using a rule which produces a sample of schedules with a low mean project duration.

Implementation of Sampling Methods

880. In practice, the sampling methods might not be implemented in the manner in which they have been formulated here (paras 635 et seq). One possible modification would assess partial schedules as

they were being created, and compare them with the best schedule obtained so far; partial schedules which could not lead to better schedules than the current 'best' schedule would then be discarded. (Mansoor (1964) uses a similar approach in his algorithm for the solution of the assembly line balance problem. Bounding and dominance criteria used by, for example, Schrage (1970), are based on identical principles.) Implementations of this nature would have a number of effects:

a. By discarding partial schedules which could not produce better schedules than any already obtained, the amount of computation time is reduced. This permits either cheaper scheduling (i.e. the same number of schedules implicitly generated in a shorter time and at a lower cost) or larger sample sizes (i.e. more schedules implicitly generated for the same computational cost). As we noted in para 640, increasing the sample size increases the probability of obtaining an optimal schedule, so an approach of this kind could be of practical value, especially if the possible saving in project cost is much larger than the computation cost, as is usually the case. McRoberts (1971) gives a decision model to aid in determining whether further sampling is likely to be worthwhile.

b. If the project horizon PHOR is set to the current

'best' project duration PDUR, rules like rule 2(LS) become more meaningful than if PHOR is large and fixed (see Example 8.4).

881. There is one major disadvantage of implementations of this nature. In practice it is often desirable to have a number of different schedules, with similar project durations, from which to choose, the choice being based on criteria outside the realm of the project scheduler. Thus, there may be a management requirement for a set of good feasible schedules, which will prohibit this kind of implementation. (The sampling methods used in the experimental investigation were programmed in this way. Consequently, only the best schedules were saved, and new experiments were required to obtain the data discussed in the previous section.)

882. The assessment of partial schedules in order to compare them with the best schedule obtained so far can be done in a number of ways. Basically, these all depend on the formation of a lower bound on the project duration, given the current partial schedule; if the bound exceeds the best project duration obtained so far, then the partial schedule is discarded. Often the cheapest bound will be required (i.e. cheapest in terms of computation time), so that any savings made by discarding a partial schedule are not used up in the process of deciding whether or not to discard it. This requirement will usually mean that the bound used is based on the CP duration,

which we shall define by

$$CPDUR = \underset{i \in A}{\text{Max}} \{\ell(i) + d(i)\},$$

where $\ell(i)$ is the CP early start time of job i, calculated by the dynamic method of para 610 and used in rules 1(FCFS), 3(EF) and 5(FLOAT). This is a simple quantity to obtain, as values of $\ell(i)$ are used in other parts of the scheduling process. Other bounds based on the resource requirements of the project are possible; Zaloom (1971) gives a set of lower bounds which could be adapted for dynamic calculation, and Fernandez and Bussell (1973) give bounds in a special case.

Comparison of Parallel and Sampling Methods

883. The tests described in paras 759 to 763 were performed on a set of 141 paired project durations, being the results of the 112 schedules used in the analysis of the sampling methods, and some other preliminary sampling results, paired with the corresponding project durations obtained by the parallel methods (Annex I). Critical values of the chi-squared statistic and the critical differences $|x-y|$ for the case n=141 are given in Table 8.17.a; results of the Sign Test for different integer values of δ are presented in Table 8.17.b. From this simple non-parametric test we obtain the result that the 99% confidence limits for the percentage relative difference between the parallel and sampling methods are 6% and 11%; in other words, the results of the sampling methods are between six and eleven

P	0.100	0.050	0.010	0.005
χ^2	2.706	3.841	6.635	7.879
$\lvert x-y \rvert$ (n=141)	21	25	32	35

(a)

δ	x	y	n	$\lvert x-y \rvert$	P
6	95	46	141	49	<0.005
7	83	58	141	25	0.050
8	72	69	141	3	>0.100
9	60	81	141	21	0.100
10	50	91	141	41	<0.005

(b)

Table 8.17. Sign Test:

a. Critical values.

b. Results.

(P is the probability of the observed results occurring by chance.)

per cent better than the results of the corresponding parallel methods, with 99% confidence.

884. In the confirmatory t Test, described in para 763, the critical value of t(0.005,140) for the 99% confidence limits is 2.611. The population (g(i)/h(i)-1)100 has mean m and standard deviation s given by

$$m = 8.706$$

$$s = 6.265.$$

Then the 99% confidence limits on the percentage relative differences are 7.3% and 10.1%; these limits lie within those obtained by the Sign Test.

885. Implications. These confidence limits have very important implications. Firstly, we note that the sampling methods, even with the relatively small sample size of 100, generally produce schedules that are at least 6% better than the corresponding schedules produced by the parallel methods. Thus, for a small increase in computation cost, we are practically guaranteed a large reduction in project cost. The computation cost is in any case only a small part of the overall planning cost; once an investment has been made in data collection and preparation, the increase in computation cost to achieve a markedly better schedule is trivial in comparison.

886. On the other hand, the upper confidence limit is a bound on the improvement in scheduling results expected when we use a

sampling method with sample size 100 instead of the corresponding parallel method. We should not anticipate a saving of much more than 10%, with a sample size of 100.

Measures of Effectiveness

887. The two measures of scheduling effectiveness used in this experiment, project duration and utilisation factor, while they are equivalent for a single project, have different properties. Each has its advantages, depending on the scheduling environment.

888. Project duration. When we are considering the scheduling of a single project, project duration is often more useful than the utilisation factor. In this case, it is:

a. Easier to calculate,

b. An integer-valued quantity, and

c. Equivalent to utilisation factor.

However, it gives no indication of the efficiency of the schedule, which may be a major disadvantage if resources are very scarce. When a number of projects are to be compared, as in this experiment, project duration is of limited use, since the ranges of project durations, and their variances, are large, due to the dependence of the project durations on the work content and CP duration of the individual projects.

889. Utilisation factor. When we are considering the scheduling of a single project, utilisation factor is not as useful as project

duration in many cases; however, it does give a measure of the efficiency of the resource usage of the schedule. In cases where a number of different projects are to be scheduled, utilisation factor is a far more useful measure than project duration. Because it is a dimensionless quantity (in the sense of dimensional analysis), and because it lies within a limited range, the variance of utilisation factors is small. This low variance makes the utilisation factors far more useful for statistical purposes than the project duration.

890. Conclusions. In the case of a single project, the two measures of effectiveness are equally useful. Project duration has some practical advantages over the utilisation factor in this case, but it gives no indication of the efficiency of the schedule; for major projects it may be desirable to use both measures. In experimental comparisons of the kind described in this thesis, the statistical properties of the utilisation factor, with its generally smaller variance over a number of different projects, make it a far more useful criterion than project duration.

Problems from the Literature

891. The parallel and sampling methods were applied to a limited set of scheduling problems that have appeared in the open literature; details of the problems, and scheduling results, are presented in Annex J. The problems were scheduled with each of the 26 priority rules in

a parallel method, and with rules 10(RANDOM) and 16(PROD) in a sampling method, with sample size 100. For each problem, at least one priority rule in a parallel method produced a schedule that was as good as that produced by the sampling methods, but no rule was uniformly good over all problems when used in a parallel method; however, rule 18(CUMRED) did well on the smaller problems. The sampling methods were better overall.

CHAPTER 9. SUMMARY AND CONCLUSIONS

CHAPTER 9. SUMMARY AND CONCLUSIONS

901. In Part 1 of this thesis, we introduced the general project scheduling problem, and the related problems of assembly line balancing and job shop scheduling. In Part 2, we treat the specific problem of interest, that of scheduling a project for minimum project duration, subject to constraints on the availability of resources.

902. In Chapter 5 the problem was formulated; we indicated that exact solution procedures were impracticable for all but the smallest projects, and, in Chapter 6, we described some simple heuristic methods, based on priority values, for obtaining approximate solutions. Chapter 7 was devoted to a detailed description of an experimental investigation of two particular classes of heuristic methods (the parallel methods and the sampling methods), and the effects of different priority rules and different project characteristics on the schedules obtained by them. We noted also that serial methods produced the same schedules as the parallel methods when the priority rule was a static rule, and so serial methods would be implicitly investigated. The results of the investigation were presented in Chapter 8, together with the results of some further experiments that were suggested by the initial analyses.

903. In this chapter, we summarise the results of the experiments and draw conclusions where appropriate.

Parallel Methods - Results

904. The AOV of results from the parallel methods (paras 809 et seq) for both criteria indicated that, within a 5% significant level:

a. Priority rules have a significant effect on the schedules obtained by the parallel methods.

b. Each of the project characteristics varied in the experiment has a significant effect on the results.

c. There are significant interaction effects between the priority rules and each of the project characteristics, except that there is no rules by density interaction when project duration is the criterion.

d. There are no significant three-way interactions.

905. Priority rules effects. The initial examination of the priority rules effects is reported in paras 814 to 817. There, we obtained group means X(i....) for each criterion, and the corresponding standard deviations (Table 8.3); using these values, we were able to rank the rules under each criterion (Table 8.4), and compare the rules, in pairs, with a t Test. We found that there were no significant differences between the group means when the criterion was project duration, since the standard deviations were large, but that when utilisation factors were considered, there were significant differences (Table 8.5). We noted that rules 23 (FIFS) and 18 (CUMRED) were very good rules, and that rules 6 (SPT) and 19 (1/RPOSWT) were very poor.

906. Effects of project characteristics. In paras 818 to 826 we discussed the effects of the project characteristics, which are summarised in Table 8.6. Briefly, we noted a general increase in project durations with increasing density, with decreasing order strength, and with decreasing resource factor; utilisation factors increased with increasing density, with decreasing order strength, and with increasing resource factor.

907. Interaction effects. In paras 827 to 837 we examined the interactions between the priority rules and each of the project characteristics. We noted that rules 18 and 23 produced good schedules at low order strength, while rule 15 did well at high values; rule 23 also produced good schedules at high values of resource factor and density, and rule 15 was sensitive to density, producing better schedules than expected at low values.

908. The absence of significant three-way interactions was noted in para 812.e. Even though changing the values of each of the parameters order strength and resource factor, for example, each changes the effects of the priority rules, this result indicates that the changes are in a sense additive, and that the effects which are peculiarly due to, say, order strength are not altered significantly by the value of resource factor.

909. Priority rules. The priority rules were examined in more detail in paras 838 to 857. We discussed several classes of rules,

and concluded that:

a. Global rules are better than local rules.

b. Rules which select jobs on which there depends a large amount of work produce good schedules, even if they use a very coarse estimate of the work content of a job.

910. The practical implementation of the parallel methods on a computer was briefly discussed; we noted that the computational cost of the parallel methods was small in comparison with the costs of data acquisition and report generation.

Parallel Methods - Conclusions

911. From the results we conclude that there are significant differences between the priority rules; the best rules are 23(FIFS), 18(CUMRED), 15(RPOSWT), 11(ALLSUCC)[1] and 24(FLOAT/SUC). Despite the overall significance of the interaction effects between the priority rules and the project characteristics, we are unable to say with any confidence "given a project with these particular parameter values, use this particular priority rule"; however, rules 23, 18, 15, 11 and 24 are among the best rules for all combinations of parameter values used in the experiment.

912. Given the relatively minor cost of scheduling with the parallel methods, it is economically feasible to schedule a

1. Rule 13 (NONDEP) is equivalent to rule 11 in the parallel methods; see para 843.

project several times, once with each of the above rules; this approach has the advantage of producing a set of good schedules. Such a set of schedules may be desirable for a number of reasons:

a. If one good schedule is all that is required, the best schedule of the set can be selected, and this will generally be better than the single schedule obtained by using just one rule.

b. In planning situations, when the project data may be limited (e.g. only details of major resources, such as items of heavy plant, may be included) or subject to change (for political, economic or contractual reasons), a set of alternative schedules may be required by management, to give it the greatest flexibility in meeting its own objectives, which may well be different from those of the project planner.

913. Rules based on a measure of the work dependent on a job produce good schedules, even if they use a very coarse estimate of the work content of a job. This result may be important in situations where a project is in progress and the schedule must be periodically revised as actual starting and completion times become known; if the project must be rescheduled with updated information (for example, better estimates of the likely durations of activities, or changes in their resource requirements), then all priorities must be recalculated

if rules 18 or 15 are used. However, the structure of the project graph is less subject to change than the duration and resource information, and so priorities under rule 11 remain fixed. This result indicates that rule 11 may be used instead of rules 18 or 15 in this situation, generally with no necessity to recalculate priorities from update to update, and that the schedule produced will not be much poorer than that obtained by rules 18 and 15. The ability to use priority values that remain constant from revision to revision, while it may seem to be of minor interest, can have profound effects on the implementation of a scheduling system: the necessity to recalculate global priorities periodically may require quite a different and more complex data structure from the cast where priorities are calculated once only and thereafter remain fixed, when a simple form of data structure may be adequate; considerations of this nature affect all priority rules that are based on global properties of the project.

914. Global rules are better than the corresponding local rules; with global rules, the scheduling process is oriented more towards project completion than towards individual job scheduling, and better results are obtained. It is concluded that global rules should be used in preference to local rules whenever possible.

Serial Methods

915. We have noted in several places previously that the static priority rules produce the same schedules in both serial and parallel methods, but that the serial methods are computationally more efficient for such rules, since priorities need not be recalculated. Rules 18, 15 and 11, among the best rules of those used in the experiment, are static rules; only rule 23 produces better schedules, and rule 13 is equivalent to rule 11 with the parallel methods. The statement that parallel methods are better than serial methods, while it is generally true for rules with static and dynamic equivalents, can be highly misleading. It is concluded that the serial methods can be as good as the parallel methods, and computationally more efficient, if a good static priority rule is used: suitable rules are CUMRED, RPOSWT and ALLSUCC, which produce better schedules than all rules except rule 23(FIFS), with either serial or parallel methods.

Sampling Methods

916. Results. The AOV of the results obtained by the sampling methods (para 863 et seq) did not indicate any significant priority rule effects, nor any interactions between rules and the project parameters. This was an unexpected result, possible reasons for which were discussed in para 866. It was concluded that the schedules produced by the sampling methods (i.e. the best schedule in the sample of schedules) were not as important in this investigation as the

distributions of the individual schedules within the samples.

917. Accordingly, one project was subjected to further experimental analysis (paras 869 to 875). This project was scheduled with a sampling method, using a variety of rules and sample sizes, and all intermediate schedules were saved. The results obtained (Table 8.13) indicated that there were significant differences between the means of the samples produced by different rules (Table 8.14), and that rule 16(PROD), which was found by Arcus (1966) to be so successful for the assembly line balance problem, produced a sample whose mean was significantly lower than the mean of a sample produced by any other rule tested. As was the case with the parallel methods, rules 18, 15 and 11 performed in a similar fashion to one another, but rules 11 and 13 produced samples with significantly different means (para 843 and Example 8.3). The distributions of schedules in a sample resembled the normal distribution (Figs 8.12, 8.13 and 8.14), but they were positively skewed.

918. Conclusions. Although the experiment was unable to detect significant differences between the extrema of the samples generated by the sampling methods, the sample means do differ significantly with different priority rules. We conclude that there may be long term benefits in using a priority rule which causes the sampling methods to select from a population with a low mean project duration; in other words, it is likely that different priority rules change the

proportion of optimal schedules in a sample (para 640) by effectively changing the population from which the sample is drawn, and so change the probability of selecting an optimal schedule in a sample of any given size.

919. For the particular project and priority rules tested, rule 16 (PROD) produced a sample with the lowest mean project duration; this may be a general result (c.f. Arcus' results, 1966), or it may depend on the project characteristics. We conclude that further research effort in this area could be directed to the study of samples and their properties in relation to the project characteristics; however, the experimental effort required would be considerable.

Comparison of Methods

920. Comparing corresponding schedules obtained by the parallel and sampling methods (paras 883 et seq), we noted that the sampling methods with sample size 100 produced significantly better results, as was expected. We obtained the 99% confidence limits for the percentage relative difference between the two methods; they were 6% and 11%.

921. We conclude with 99% confidence that savings of at least six per cent of project cost can be made by using a sampling method rather than a parallel method, even with a sample size as small as 100; in most cases the associated increase in computation cost will be trivial in comparison. Thus, we are able to give a numerical value

to the cost reductions that are possible by using sampling methods instead of parallel methods.

General Conclusions

922. Where possible, sampling methods should be used for project scheduling, rather than serial or parallel methods, as they are likely to produce far better schedules, even with relatively small sample sizes. In this case, there is very little difference between priority rules, but there may be advantages in scheduling with a rule which produces a sample with a low mean project duration. (Rule 16(PROD) performed well in this regard with one project tested, and has given similarly good results in assembly line balancing.) Bounding procedures which, early in the scheduling process, discard partial schedules that cannot lead to better schedules than the best obtained so far, should be implemented; methods of this kind allow larger sample sizes for the same computational effort, or, alternatively, they permit samples of the same size to be generated more efficiently.

923. If parallel methods are to be used, then one of rules 23(FIFS), 18(CUMRED), 15(RPOSWT), 11(ALLSUCC) or 24(FLOAT/SUC) is a good choice. It may be feasible to schedule the project with each of the above rules, and then choose the best schedule (i.e. a limited form of sampling). Global rules are generally better than local rules, but their use may have important effects on the data structure if the project is subject to periodic revision; in this case, too, a serial method using a good

static rule (e.g. ALLSUCC) may be advantageous, particularly if the priority values remain fixed from one updating period to the next.

924. Whatever scheduling method is used, resource requirements of the schedule should not be stored by time periods, unless the project horizon is very small. Instead, methods of the kind outlined in paras 767 and 768 should be used to reduce the storage requirements to the order of the number of jobs in the project.

Final Remarks

925. The experimental investigation described in this thesis has led to conclusions based on the particular projects used, with the particular set of parameter values. However, care has been taken to make the analyses statistically rigorous, and, where it was thought appropriate, conservative; it is likely that these conclusions apply to a wide range of industrial problems.

926. The parallel methods have been examined in depth, over a wide range of priority rules and parameter values, and the sampling methods have been shown to produce schedules that are at least 6% better than those of the parallel methods, even for sample sizes as small as 100. It is concluded that further research in this area should be directed towards the sampling methods rather than the parallel methods, since the sampling methods have the most potential for producing good schedules. Possible research areas include:

a. Detailed study of the samples of schedules produced by the sampling methods.

b. The effects of sample size.

c. Decision models for further sampling and estimation of the optimal solution (see, for example, McRoberts, 1971).

d. Other methods of biasing the sampling process.

927. Other research areas of interest are related to the problem of optimising some function of resources subject to a constraint on the project duration (paras 320 et seq). Helgeson and Birnie (1961) use an iterative approach for minimising the cycle time in an assembly line with a given number of stations; a similar approach may be suitable for project scheduling with given project horizon, using either the parallel or sampling methods described in this thesis to obtain the intermediate schedules at each iteration.

ANNEXES

ANNEX A. NOTATION

ANNEX A. NOTATION

A1. In this annex, the principal terms and symbols used in the thesis are listed. Table A.1 gives the terms, with the corresponding symbol, where appropriate, and the paragraph in which the term is introduced. Table A.2 lists the symbols and abbreviations used, with definitions and paragraph references. It should be noted that the names of the priority rules (Table 6.5) and the factors in the analysis of variance model (Table 7.6) have not been included.

Table A.1. Terms used in the thesis

TERM	PARA	SYMBOL
active partial schedule	509	
activity	301	i,j,...
arc	202	
assembly line	403	
available resources	317	R(k,t)
	506	R(k)
available set	510	AV
average delay	319	AD
average resource requirement	Table 3.2	RAVE(k)
average slippage	319	AS
circuit	211	
coefficient of variation	323	COV
critical job	306	
critical path	306	CP
critical path duration	304	CPDUR
CP early finish time	510	a(i)
CP early start time	304	ℓ(i)
CP late finish time	520	b(i)
CP late start time	305	u(i), u^*(i)
critical resource level	325	R^*
cumulative RED	626	CUMRED(i)

Table A.1 (contd.)

TERM	PARA	SYMBOL
cycle time	403	C
density	717	D
dot notation	744	X(.....), &c
duration	301	d(i)
earliest start time	511	σ(i)
feasible schedule	508	
feasible subset	C1	
float	306	f(i)
free float	716	ff(i)
graph	202	G,(X,L),(X,Γ)
immediate predecessor	206	
immediate successor	206	
interaction term	750	est RF(LK), &c
job	301	i,j,..
job shop	407	
lateness	318	
level	621	level(i)
line length	403	L
loop	211	
major sort	605	
makespan	318	
minor sort	605	
open closure	208	
open closure matrix	726	C

Table A.1 (contd.)

TERM	PARA	SYMBOL
order strength	713	S
partial ordering	215	
partial permutation	513	
partial schedule	509	
path	211	
peak resource requirement	Table 3.2	RMAX(k)
precedence matrix	728	W
priority	603	P(i)
project	301	
project duration	302	PDUR
project horizon	305	PHOR
rank positional weight	622	RPOSWT(i)
resource availability	317	R(k,t)
	506	R(k)
resource equivalent duration	625	RED(i)
	848	RED'(i)
resource factor	719	rf
resource requirement	311	r(i,k)
resource strength	720	rs(k)
resource usage	Table 3.2	RUSE(k,t)
resource usage duration	624	RUD(k)
sample size	640	M

Table A.1 (contd.)

TERM	PARA	SYMBOL
schedule	302	S
schedule activity i	512	
start time	302	s(i)
station	403	s'(i)
target duration	325	T
throughput time	318	
topological ordering	216-218	
vertex	202	
weak ordering	213	

Table A.2 Symbols used in the thesis

SYMBOL	PARA	DEFINITION
A	301	set of jobs
a(i)	520	CP early finish
AD	319	average delay
AOV	739	analysis of variance
AS	319	average slippage
AV	510	available set
b(i)	520	CP late finish
C	403	cycle time
C	726	open closure matrix
COV	323	coefficient of variation
CP	306	critical path
CPDUR	304	CP duration
CUMRED(i)	626	cumulative RED
D	717	density
d(i)	301	duration
est RF(LK),&c	750	interaction term
f(i)	306	float
ff(i)	716	free float
G	202	graph
g(i)	760	project duration of schedule calculated by parallel method

Table A.2 (contd.)

SYMBOL	PARA	DEFINITION
h	713	max number of ordering relations
h(i)	760	project duration of schedule calculated by sampling method
$h'(i,\delta)$	760	
i	301	activity, job
ILP	519	integer linear programming
j	301	activity, job
K	311	number of resources
k	311	resource
L	403	line length
$\ell(i)$	304	CP early start time
level(i)	621	level of a job
M	640	sample size
m	713	number of ordering relations
N	311	number of jobs
P(i)	603	priority
PDUR	302	project duration
PHOR	305	project horizon
q(i)	719	number of kinds of resource used
R(k)	506	resources available
R(k,t)	317	resources available
R^*	325	critical resource level

Table A.2 (contd.)

SYMBOL	PARA	DEFINITION
r(i,k)	311	resource requirement
RAVE(k)	Table 3.2	average resource requirement
RED(i)	625	resource equivalent duration
RED'(i)	848	resource equivalent duration (modified)
rf	719	resource factor
RMAX(k)	Table 3.2	peak resource requirement
RPOSWT(i)	622	rank positional weight
rs(k)	720	resource strength
RUD(k)	624	resource usage duration
RUSE(k,t)	Table 3.2	resource usage
S	302	schedule
S	713	order strength
s(i)	302	start time
s'(i)	403	station
T	325	target duration
U	740	grand mean
u(i), $u^*(i)$	305	CP late start time
W	728	precedence matrix
X(.....), &c	744	means (dot notation)
Γ (defn.)	204	precedence relation
Γ (props.)	207-210	

Table A.2 (contd.)

SYMBOL	PARA	DEFINITION
$\sigma(i)$	511	earliest start time
#(X)	216 (footnote)	number of elements in set X

ANNEX B. NON-UNIQUENESS OF TOPOLOGICAL ORDERING

ANNEX B. NON-UNIQUENESS OF TOPOLOGICAL ORDERING

B1. In para 218 it was stated that a partially ordered set may be topologically ordered in more than one way, in general. Table B.1 shows the functions associated with the eleven topological orderings of the partially ordered set X = {A,B,C,D,E,F,G} with ordering function Γ illustrated in the graph of Fig 2.2. The function n_1 is associated with the topological ordering illustrated in Fig 2.3 (Example 2.5).

VERTEX	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8	n_9	n_{10}	n_{11}
A	1	1	1	1	1	1	1	1	1	1	1
B	4	4	5	4	4	5	3	3	3	5	4
C	5	6	6	5	6	6	5	6	6	6	6
D	7	7	7	7	7	7	7	7	7	7	7
E	2	2	2	3	3	3	4	4	5	4	5
F	6	5	4	6	5	4	6	5	4	3	3
G	3	3	3	2	2	2	2	2	2	2	2

Table B.1 Functions associated with the topological orderings of the graph of Fig 2.2.

ANNEX C. SIZE OF THE COMBINATORIAL PROBLEM

ANNEX C. SIZE OF THE COMBINATORIAL PROBLEM

C1. A feasible subset X of a partially ordered set A is a subset with the property

whenever ($i \in X$) then ($j \in X$, for all $j \in \Gamma^{-1} i$).

Referring to para 513, we note that the set of jobs which appear in a feasible partial permutation form a feasible subset, since they have the above property.

Example C.1. Consider the set of jobs with partial ordering defined by the graph of Fig 2.2. There are fourteen feasible subsets of these seven jobs, listed in Table C.1.

C2. The number of feasible subsets of a partially ordered set A is less than the number of partial permutations on A, since in general we can form different feasible partial permutations of the same feasible subset X. (Table B.1 of Annex B shows eleven different feasible permutations of the same partially ordered set; this set consists of the seven jobs in Example C.1.) Held, Karp and Shareshian (1963) give an algorithm for determining the number of feasible subsets of a partially ordered set; this number is a lower bound on the number of feasible partial permutations and from it we can obtain an indication of the size of the combinatorial problem.

C3. Table C.2 lists the number of feasible subsets of some partially ordered sets that have appeared in the literature. It can be seen that the size of the combinatorial problem increases

1	ø
2	{A}
3	{A,E}
4	{A,G}
5	{A,B,G}
6	{A,E,G}
7	{A,F,G}
8	{A,B,E,G}
9	{A,B,F,G}
10	{A,E,F,G}
11	{A,B,C,E,G}
12	{A,B,E,F,G}
13	{A,B,C,E,F,G}
14	{A,B,C,D,E,F,G}

Table C.1. Feasible subsets of the graph of Fig 2.2.

NUMBER OF JOBS	NUMBER OF FEASIBLE SUBSETS	REFERENCE
7	14	Fig 2.2
11	52	Jackson (1956)
13	72	Mason and Moodie (1971)
16	392	Wiest (1967)
53	$2*10^7$	Karni (1973

Table C.2. Size of some problems from the literature.

rapidly as the number of jobs increases, and the amount of computer time required for enumeration soon becomes excessive. For example, suppose we conservatively estimate that there are ten feasible partial permutations of each feasible subset of Karni's 53 job problem; if a different partial permutation can be generated and evaluated every one millisecond, say, then the calculation time alone would be of the order of 55 hours. Even with the addition of bounding and dominance techniques (which themselves require a considerable amount of computation if they are to be effective) to reduce its size, the combinatorial problem is large, and enumerative methods are impracticable for all but the smallest projects.

ANNEX D. GENERATION OF THE OPEN CLOSURE MATRIX.

ANNEX D. GENERATION OF THE OPEN CLOSURE MATRIX

D1. In paras 725 to 727, we showed how the problem of generating a project with a given value of order strength S became a problem of creating a strictly upper triangular open closure matrix C, with m = 1770S one-elements. In this Annex, we describe a suitable generation method, shown in outline in Fig D.1.

D2. After initiallising the matrix C to all zero-elements, and setting a counter, TOTAL, of the number of one-elements in C to zero, a pair of integers (i,j) are randomly generated. These are both in the range 1 to 60, and are such that $i < j$ and $C_{ij} = 0$, i.e. they correspond to a valid ordering relation $j \varepsilon \Gamma^* i$ which is not already represented in the open closure matrix. Having found a zero-element (i,j) of C above the main diagonal, set C_{ij} to 1, and set to one such other elements of C as may be required by the transitivity condition of para (using the procedure INSERT, described in the next paragraph). In TOTAL is maintained a count of the number of elements of C changed from zero to one; while the number of one-elements in C, as recorded by TOTAL, is less than the required number m, the process cycles as new pairs (i,j) are formed and new elements are added to C.

D3. Procedure INSERT. Once a relation $j \varepsilon \Gamma^* i$ has been chosen to be added to the project, the element C_{ij} must be changed from zero to one, and the transitivity conditions must be checked. The procedure INSERT handles this; it is shown in outline in Fig D.2. All elements

1. Enter.

2. Set TOTAL=0; set C_{ij}=0, for all i and j.

3. Generate random integers i and j in the range [1,60].

4. If i is not less than j, then go to 3.

5. If C_{ij} is not 0, then go to 3.

6. INSERT(i,j,TOTAL). (See Fig D.2 for details of this procedure.)

7. If TOTAL is less than m, then go to 3.

8. Exit.

Fig D.1. Generation of the matrix C.

1. Enter INSERT(i,j,TOTAL).
2. Set C_{ij}=1, TOTAL=TOTAL+1, k=0.
3. Set k=k+1.
4. If k is equal to i, then go to 8.
5. If C_{ki} is not equal to 1, then go to 3.
6. If C_{kj} is not equal to 0, then go to 3.
7. INSERT(k,j,TOTAL); go to 3.
8. Set k=k+1.
9. If k is greater than 60, then go to 13.
10. If C_{jk} is not equal to 1, then go to 8.
11. If C_{ik} is not equal to 0, then go to 8.
12. INSERT(i,k,TOTAL); go to 8.
13. Exit.

Fig D.2. Procedure INSERT.

k of the project which could validly precede i (i.e. $k < i$) are first inspected: if k does in fact precede i ($C_{ki} = 1$), then k must also precede j ($C_{kj} = 1$); if this is not the case, then the relation $j \varepsilon \Gamma^* k$ must be added to the project. Similarly, all elements k which could validly succeed j are inspected: if k is a successor of j ($C_{jk} = 1$), then k must also be a successor of i ($C_{ik} = 1$); if this is not the case, then the relation $k \varepsilon \Gamma^* i$ must be added to the project. The addition of new relations to the project requires the use of procedure INSERT, which is therefore a recursive procedure.

ANNEX E. PROGRAMMING METHODS

ANNEX E. PROGRAMMING METHODS

E1. Figs E.1, E.2 and E.3 show, in outline, the main program and the procedures for the parallel and sampling methods. These procedures were coded in FORTRAN and COMPASS, the machine language of the CDC 6400 used in the experiment. (In the parallel and sampling methods, values of $\ell(i)$ for unscheduled jobs i are used in later computation of σ.)

E2. Matrix methods. The calculation of the available set, AV, and the CP early and late start time, ℓ and u, was facilitated by the use of matrix methods; the ordering relations Γ were represented by a precedence matrix W (para 728), any row of which can be stored within one 60-bit computer word. Then, by utilising the logical operations available in either FORTRAN or COMPASS, calculations can be performed on entire rows of W, enabling large savings to be made in computation times. Procedures to exploit this property were investigated by the author as partial requirement for the Honours degree of B.Sc. (Cooper, 1969), and similar techniques have been reported by Macaskill (1969, 1973).

E3. Implementation of rules and heuristics. Aspects of the implementation of the priority rules and the heuristic methods are discussed in various places in Chapters 8 and 9. In particular, in paras 880 to 882 we discuss an implementation of the sampling methods which was slightly different from the outline procedure of Fig 6.5.

Other comments on implementation are given in para 848, and paras 852 to 855.

1. Initiallise.
2. Read project data.
3. Read rule number (NRULE), and scheduling method.
4. Schedule the project with the specified method.
5. Print results.
6. Stop.

Fig E.1. Main program.

1. Enter.
2. Set SCHED = $\emptyset$.
3. Form $\ell(i)$, for all unscheduled jobs i.
4. Form the available set, AV.
5. Calculate priorities P(i), for all jobs i in AV, by rule NRULE.
6. Select job j from AV with maximum priority.
7. Calculate $\sigma(j)$; set(j) = $\sigma(j)$; set SCHED = SCHED $\cup \{j\}$.
8. If #(SCHED) is equal to N, then go to 10.
9. If $\sigma(j)$ is equal to $\ell(j)$, then go to 4; otherwise, go to 3.
10. Exit.

Fig E.2. Parallel method. (SCHED is the set of scheduled jobs.)

1. Enter

2. Set SCHEDNUM = 1.

3. Set SCHED = $\emptyset$.

4. Form $\ell(i)$, for all unscheduled jobs i.

5. Form the available set, AV.

6. Calculate priorities P(i), for all jobs i in AV, by rule NRULE.

7. Select job j from AV at random, such that the probability of any job i in AV being selected is

$$p(i) = \frac{P(i)}{\Sigma P(k)} ,$$

where the sum is over all jobs k in AV.

8. Calculate $\sigma(j)$; set $s(j) = \sigma(j)$; set SCHED = SCHED $\cup \{j\}$.

9. Evaluate the partial schedule. If this partial schedule cannot produce a better schedule than the best schedule obtained so far, then go to 12.

10. If #(SCHED) is equal to N, then save this schedule as the new current best schedule, and go to 12.

11. If $\sigma(j)$ is equal to $\ell(j)$, then go to 5; otherwise, go to 4.

12. If SCHEDNUM is equal to M, then go to 13; otherwise, get SCHEDNUM = SCHEDNUM + 1, and go to 3.

13. Exit.

Fig E.3. Sampling method. (SCHEDNUM is the current schedule number; SCHED is the set of scheduled jobs; M is the sample size.)

ANNEX F. RESULTS

ANNEX F. RESULTS

F1. Table F.1 shows project identifiers for the 32 projects used in the experiment, together with values of the parameters order strength, density and resource factor.

F2. Parallel methods. Tables F.2 and F.3 show values of project durations and utilisation factors for the parallel methods.

F3. Sampling methods. Tables F.4 and F.5 show values of project durations and utilisation factors for the sampling methods.

PROJECT	ORDER STRENGTH	DENSITY	RESOURCE FACTOR
1	.10	.50	.5
2	.10	.50	.5
3	.10	.50	1.0
4	.10	.50	1.0
5	.10	.75	.5
6	.10	.75	.5
7	.10	.75	1.0
8	.10	.75	1.0
9	.25	.50	.5
10	.25	.50	.5
11	.25	.50	1.0
12	.25	.50	1.0
13	.25	.75	.5
14	.25	.75	.5
15	.25	.75	1.0
16	.25	.75	1.0
17	.50	.50	.5
18	.50	.50	.5
19	.50	.50	1.0
20	.50	.50	1.0
21	.50	.75	.5
22	.50	.75	.5
23	.50	.75	1.0
24	.50	.75	1.0
25	.75	.50	.5
26	.75	.50	.5
27	.75	.50	1.0
28	.75	.50	1.0
29	.75	.75	.5
30	.75	.75	.5
31	.75	.75	1.0
32	.75	.75	1.0

Table F.1 Project identifiers and parameters.

PROJECT DURATIONS

RULE	PROJECT 1	2	3	4	5	6	7	8
1	4269	3469	3357	3732	7014	5959	6382	5404
2	4089	3533	3412	3393	7381	5723	6238	5412
3	3997	3516	3461	3545	6723	6066	6369	5585
4	4147	3787	3567	3555	6991	5479	6525	5465
5	4173	3606	3541	3519	7462	6170	6613	5389
6	4944	3872	3489	3704	8226	6362	6674	5628
7	4155	3946	3333	3646	7343	5819	6572	5534
8	4185	3712	3465	3634	7298	5921	8294	5472
9	3922	3703	3399	3490	7362	5417	6666	5381
10	3869	4399	3388	3691	7239	5502	6111	5296
11	4349	3332	3434	3602	6807	5582	6542	5474
12	4753	4075	3524	3730	7441	5562	6502	5537
13	3817	3437	3539	3719	7015	5819	6466	5266
14	4277	3704	3514	3572	7095	6091	6381	5294
15	3905	3655	3334	3516	7491	5932	6253	5215
16	3956	3413	3315	3599	7136	5955	6614	5393
17	4092	3575	3504	3493	7211	5845	6562	5275
18	3844	3282	3345	3513	6954	5468	6174	5208
19	4885	4156	3874	3823	8003	6854	6628	5443
20	4021	3669	3544	3618	7986	5549	6527	5510
21	4429	4390	3821	3874	7514	6572	6468	5330
22	4852	3902	3526	3894	8475	6020	6401	5230
23	4118	3423	3132	3346	7004	5487	5722	5004
24	4349	3332	3434	3602	6807	5582	6542	5474
25	4204	3406	3427	3836	6921	5783	6307	5376
26	3873	3979	3299	3360	7689	5928	5961	5270

Table F.2. Project durations for the parallel methods.

PROJECT DURATIONS

RULE	PROJECT							
	9	10	11	12	13	14	15	16
1	2119	2935	2208	2316	3429	3690	2951	3219
2	2245	2674	2236	2306	3408	3628	3328	3264
3	2558	2899	2231	2189	3720	3529	3162	3157
4	2344	2691	2253	2234	3450	3617	3276	3414
5	2098	2760	2273	2267	3747	4432	3282	3501
6	2578	3325	2348	2613	3645	4401	3452	3591
7	2267	3057	2249	2337	3721	3627	3203	3628
8	2467	2503	2210	2279	3694	3439	3167	3395
9	2152	2736	2203	2218	3538	3404	3299	3326
10	2293	2970	2092	2279	3537	3422	3335	3247
11	2138	2610	2143	2219	3364	3338	3207	3298
12	2214	2833	2263	2289	3918	3805	3359	3263
13	2145	2610	2149	2219	3381	3338	3204	3298
14	2127	2719	2236	2190	3683	3728	3286	3314
15	2038	2684	2167	2139	3525	3346	3118	3265
16	2226	2657	2121	2112	3331	3672	3177	3179
17	2292	2703	2167	2377	3830	3579	3080	3660
18	2021	2708	2142	2120	3139	3491	3225	3239
19	2821	2877	2384	2471	3850	3782	3279	3257
20	2163	2715	2301	2234	3446	3898	2956	3429
21	2379	3360	2373	2330	3943	4353	3376	3407
22	2314	2910	2458	2479	3859	3943	3243	3312
23	2298	2715	2069	2094	3362	3539	2885	3019
24	2138	2610	2143	2219	3364	3338	3207	3298
25	2306	2697	2207	2154	3631	3663	3154	3130
26	2302	2709	2191	2243	3388	4107	3009	3099

Table F.2. Project durations for the parallel methods (contd.)

PROJECT DURATIONS

RULE	PROJECT							
	17	18	19	20	21	22	23	24
1	1779	2264	1407	1575	2686	2896	2198	2208
2	1658	2061	1398	1563	2668	2694	2271	2266
3	1797	2400	1520	1691	2793	2853	2286	2240
4	1684	2127	1381	1609	2555	2844	2213	2288
5	1800	2487	1464	1673	2828	2822	2416	2376
6	2103	2533	1714	1831	3369	2902	2448	2526
7	1990	2097	1608	1845	3146	2954	2418	2316
8	1713	2372	1494	1639	2619	3030	2576	2352
9	1554	2120	1441	1647	2810	2729	2241	2320
10	1904	2352	1599	1763	2711	2837	2388	2262
11	1735	2065	1424	1581	2719	2614	2208	2318
12	2062	2384	1452	1708	3264	2857	2486	2418
13	1735	2065	1424	1581	2719	2614	2208	2318
14	1744	2300	1393	1674	3172	2625	2287	2327
15	1669	1984	1382	1522	2775	2725	2199	2270
16	1650	2071	1463	1533	2773	2762	2193	2265
17	1772	2429	1518	1847	3245	2803	2267	2310
18	1625	1970	1351	1514	2634	2458	2207	2304
19	2279	2623	1625	1747	3050	2918	2464	2370
20	1681	2271	1497	1703	2842	2729	2258	2411
21	1894	2821	1525	1690	3025	2852	2456	2266
22	2260	2543	1751	1788	3050	2918	2287	2366
23	1504	2019	1371	1494	2656	2764	2114	2049
24	1735	2065	1424	1581	2719	2614	2208	2318
25	1749	2240	1432	1598	2652	2969	2189	2146
26	2032	2463	1568	1625	2724	2724	2300	2440

Table F.2. Project durations for the parallel methods (contd.)

PROJECT DURATIONS

RULE	PROJECT							
	25	26	27	28	29	30	31	32
1	1437	1631	1122	1032	2235	2047	1406	1289
2	1325	1513	1057	1044	2020	2019	1425	1217
3	1452	1582	1148	1073	2021	1924	1359	1252
4	1320	1454	1080	1067	2065	1984	1410	1293
5	1374	1524	1105	1009	2146	2261	1558	1246
6	1598	1677	1183	1048	2084	2170	1586	1359
7	1490	1771	1071	1034	2112	2319	1539	1319
8	1524	1561	1088	1028	2159	2111	1465	1367
9	1401	1420	1108	1024	2196	1937	1389	1160
10	1477	1573	1097	1057	2329	2182	1435	1331
11	1224	1511	1096	1000	2072	1898	1363	1206
12	1461	1621	1111	1042	2257	2151	1475	1312
13	1224	1511	1096	1000	2072	1898	1363	1206
14	1348	1599	1103	1018	1859	2110	1393	1243
15	1219	1495	1077	1000	2085	1852	1325	1252
16	1406	1564	1099	1009	2071	1905	1274	1215
17	1458	1726	1179	1075	2264	2186	1453	1306
18	1305	1584	1086	1000	2051	1919	1302	1229
19	1581	1859	1153	1121	2247	2333	1527	1341
20	1487	1676	1125	1033	2109	2021	1483	1271
21	1400	1545	1126	1080	2144	2209	1493	1351
22	1581	1801	1128	1105	2436	2267	1486	1334
23	1416	1475	1112	1032	2034	1935	1364	1275
24	1224	1511	1096	1000	2072	1950	1363	1204
25	1425	1589	1096	1032	2181	2012	1334	1289
26	1464	1599	1204	1117	2183	2027	1383	1340

Table F.2. Project durations for the parallel methods (contd.).

UTILISATION FACTORS

RULE	PROJECT							
	1	2	3	4	5	6	7	8
1	54.6	60.9	69.6	65.5	63.9	55.9	70.5	70.1
2	57.0	59.8	68.4	72.1	60.7	58.2	72.1	70.0
3	58.3	60.1	67.5	69.0	66.6	54.9	70.6	67.8
4	56.2	55.8	65.5	68.8	64.1	60.8	68.9	69.3
5	55.8	58.6	65.9	69.5	60.0	54.0	68.0	70.3
6	47.1	54.6	66.9	66.0	54.5	52.4	67.4	67.3
7	56.1	53.6	70.1	67.1	61.0	57.2	68.4	68.4
8	55.7	57.0	67.4	67.3	61.4	56.3	71.4	69.2
9	59.4	57.1	68.7	70.1	60.9	61.5	67.5	70.4
10	60.2	48.1	68.9	66.2	61.9	60.5	73.6	71.5
11	53.6	63.5	68.0	67.9	65.8	59.7	68.7	69.2
12	49.0	51.9	66.3	65.6	60.2	59.9	69.2	68.4
13	61.0	61.5	66.0	65.7	63.9	57.2	69.6	71.9
14	54.5	57.1	66.4	68.4	63.2	54.7	70.5	71.5
15	59.7	57.8	70.0	69.5	59.8	56.2	71.9	72.6
16	58.9	61.9	70.4	67.9	62.8	55.9	68.0	70.2
17	56.9	59.1	66.6	70.0	62.1	57.0	68.5	71.8
18	60.6	64.4	69.8	69.6	64.4	60.9	72.8	72.7
19	47.7	50.9	60.3	64.0	56.0	48.6	67.8	69.6
20	57.9	57.6	65.9	67.6	56.1	60.0	68.9	68.7
21	52.6	48.2	61.1	63.1	59.6	50.7	69.5	71.1
22	48.0	54.2	66.2	62.8	52.9	55.3	70.2	72.4
23	56.6	61.8	74.6	73.1	64.0	60.7	78.6	75.7
24	53.6	63.5	68.0	67.9	65.8	59.7	68.7	69.2
25	55.4	62.1	68.1	63.7	64.7	57.6	71.3	70.4
26	60.2	53.1	70.8	72.8	58.3	56.2	75.4	71.9

Table F.3. Utilisation factors for the parallel methods.

UTILISATION FACTORS

RULE	PROJECT							
	9	10	11	12	13	14	15	16
1	51.2	47.5	68.1	62.4	53.2	48.6	70.8	67.2
2	48.3	52.1	67.3	62.6	53.5	49.5	62.7	66.3
3	42.4	48.1	67.4	66.0	49.0	50.8	66.0	68.5
4	46.3	51.8	66.8	64.7	52.8	49.6	63.7	63.3
5	51.7	50.5	66.2	63.7	48.6	40.5	63.6	61.8
6	42.1	41.9	64.1	55.3	50.0	40.8	60.5	60.2
7	47.8	45.6	66.9	61.8	49.0	49.5	65.2	59.6
8	44.0	55.7	68.1	63.4	49.3	52.2	65.9	63.7
9	50.4	50.9	68.3	65.1	51.5	52.7	63.3	65.0
10	47.3	46.9	71.9	63.4	51.5	52.4	62.6	66.6
11	50.7	53.4	70.2	65.1	54.2	53.8	65.1	65.6
12	49.0	49.2	66.5	63.1	46.5	47.2	62.2	66.3
13	50.6	53.4	70.0	65.2	53.9	53.8	65.2	65.6
14	51.0	51.2	67.3	65.9	49.5	48.1	63.5	65.3
15	53.2	51.9	69.4	67.5	51.7	53.6	67.0	66.2
16	48.7	52.4	70.9	68.4	54.7	48.9	65.7	68.0
17	47.3	51.5	69.4	60.8	47.6	50.1	67.8	59.1
18	53.7	51.5	70.2	68.1	58.1	51.4	64.7	66.8
19	38.5	48.4	63.1	58.5	47.3	47.4	63.7	66.4
20	50.2	51.3	65.4	64.7	52.9	46.0	70.6	63.1
21	45.6	41.5	63.4	62.0	46.2	41.2	61.8	63.5
22	46.9	47.9	61.2	58.3	47.2	45.5	64.4	65.3
23	47.2	51.3	72.7	69.0	54.2	50.7	72.4	71.6
24	50.7	53.4	70.2	65.1	54.2	53.8	65.1	65.6
25	47.0	51.7	68.2	67.1	50.2	49.0	66.2	69.1
26	47.1	51.4	68.7	64.4	53.8	43.7	69.4	69.8

Table F.3. Utilisation factors for the parallel methods (contd.)

UTILISATION FACTORS

RULE	PROJECT							
	17	18	19	20	21	22	23	24
1	44.3	44.0	63.5	62.0	53.3	43.4	67.6	64.5
2	47.5	48.4	63.9	62.5	53.7	46.6	65.4	62.8
3	43.8	41.5	58.8	57.7	51.3	44.0	65.0	63.6
4	46.8	46.9	64.7	60.7	56.0	44.2	67.2	62.2
5	43.8	40.1	61.0	58.4	50.6	44.5	61.5	59.9
6	37.5	39.3	52.2	53.3	42.5	43.3	60.7	56.4
7	39.6	47.5	55.6	52.9	45.5	42.5	61.5	61.5
8	46.0	42.0	59.8	59.6	54.7	41.5	57.7	60.5
9	50.7	47.0	62.0	59.3	51.0	46.0	66.3	61.4
10	41.4	42.4	55.9	55.4	52.8	44.3	62.2	63.0
11	45.4	48.3	62.8	61.8	52.7	48.1	67.3	61.4
12	38.2	41.8	61.6	57.2	43.9	44.0	59.8	58.9
13	45.4	48.3	62.8	61.8	52.7	48.1	67.3	61.4
14	45.2	43.3	64.2	58.3	45.1	47.9	65.0	61.2
15	47.2	50.2	64.7	64.2	51.6	46.1	67.6	62.7
16	47.7	48.1	61.1	63.7	51.6	45.5	67.8	62.9
17	44.5	41.0	58.9	52.9	44.1	44.8	65.6	61.7
18	48.5	50.6	66.2	64.5	54.4	51.1	67.3	61.8
19	34.6	38.0	55.0	55.9	46.9	43.1	60.3	60.1
20	46.9	43.9	59.7	57.3	50.4	46.0	65.8	59.1
21	41.6	35.3	58.6	57.8	47.3	44.1	60.5	62.8
22	34.8	39.2	51.0	54.6	46.9	43.1	65.0	60.2
23	52.4	49.4	65.2	65.4	53.9	45.5	70.3	69.5
24	45.4	48.3	62.8	61.8	52.7	48.1	67.3	61.4
25	45.0	44.5	62.4	61.1	54.0	42.3	67.9	66.4
26	38.8	40.5	57.0	60.1	52.6	46.1	64.6	58.4

Table F.3. Utilisation factors for the parallel methods (contd.)

UTILISATION FACTORS

RULE	PROJECT							
	25	26	27	28	29	30	31	32
1	36.1	34.4	50.9	45.2	40.2	43.6	59.2	59.6
2	39.1	37.1	54.1	44.7	44.4	44.2	58.5	63.1
3	35.7	35.4	49.8	43.5	44.4	46.4	61.3	61.4
4	39.3	38.6	52.9	43.8	43.5	45.0	59.1	59.4
5	37.7	36.8	51.7	46.3	41.8	39.5	53.5	61.7
6	32.5	33.4	48.3	44.6	43.1	41.2	52.5	56.5
7	34.8	31.7	53.4	45.2	42.5	38.5	54.1	58.3
8	34.0	35.9	52.5	45.4	41.6	42.3	56.9	56.2
9	37.0	39.5	51.6	45.6	40.9	46.1	60.0	66.2
10	35.1	35.7	52.1	44.2	38.5	40.9	58.0	57.7
11	42.4	37.1	52.1	46.7	43.3	47.1	61.1	63.7
12	35.5	34.6	51.4	44.8	39.8	41.5	56.5	58.6
13	42.4	37.1	52.1	46.7	43.3	47.1	61.1	63.7
14	38.5	35.1	51.8	45.9	48.3	42.3	59.8	61.8
15	42.5	37.5	53.1	46.7	43.0	48.2	62.9	61.4
16	36.9	35.8	52.0	46.3	43.3	46.9	65.4	63.2
17	35.6	32.5	48.5	43.4	39.7	40.9	57.3	58.8
18	39.7	35.4	52.6	46.7	43.8	46.5	64.0	62.5
19	32.8	30.2	49.6	41.7	40.0	38.3	54.5	57.3
20	34.9	33.5	50.8	45.2	42.6	44.2	56.2	60.5
21	37.0	36.3	50.7	43.2	41.9	40.4	55.8	56.9
22	32.8	31.1	50.7	42.3	36.8	39.4	56.1	57.6
23	36.6	38.0	51.4	45.2	44.1	46.2	61.1	60.3
24	42.4	37.1	52.1	46.7	43.3	45.8	61.2	63.8
25	36.4	35.3	52.1	45.2	41.2	44.4	62.4	59.6
26	35.4	35.1	47.5	41.8	41.1	44.1	60.2	57.3

Table F.3. Utilisation factors for the parallel methods (contd.)

PROJECT DURATIONS

RULE	PROJECT							
	10	12	13	15	17	19	22	24
1	2554	2082	3262	2916	1666	1414	2499	2127
2	2619	2122	3306	2883	1602	1392	2599	2105
3	2569	2140	3315	2889	1609	1372	2521	2143
4	2507	2083	3253	3014	1654	1414	2692	2124
5	2629	2125	3395	2893	1658	1430	2485	2104
6	2623	2120	3300	2998	1552	1412	2499	2179
7	2546	2119	3289	2912	1566	1416	2570	2119
8	2526	2133	3363	2933	1670	1454	2586	2161
9	2665	2128	3328	2949	1561	1407	2553	2150
10	2683	2107	3266	2924	1672	1436	2565	2095
11	2553	2104	3160	2953	1594	1350	2456	2153
12	2686	2155	3340	2937	1584	1390	2474	2137
13	2488	2078	3295	2895	1586	1394	2437	2135
15	2447	2036	3223	2897	1618	1357	2444	2130

Table F.4. Project durations for the sampling methods.

UTILISATION FACTORS

RULE	PROJECT							
	10	12	13	15	17	19	22	24
1	54.5	69.4	55.9	71.6	47.3	63.2	50.3	66.9
2	53.2	68.1	55.1	72.4	49.2	64.2	48.3	67.7
3	54.2	67.5	55.0	72.3	49.0	65.2	49.8	66.4
4	55.6	69.3	56.0	69.3	47.6	63.2	46.7	67.1
5	53.0	68.0	53.7	72.2	47.5	62.5	50.6	67.7
6	53.1	68.1	55.2	69.7	50.7	63.3	50.3	65.3
7	54.7	68.2	55.4	71.7	50.3	63.1	48.9	67.2
8	55.2	67.7	54.2	71.2	47.2	61.5	48.6	65.9
9	52.3	67.9	54.8	70.8	50.5	63.5	49.2	66.2
10	51.9	68.6	55.8	71.4	47.1	62.2	49.0	68.0
11	54.6	68.7	57.7	70.7	49.4	66.2	51.2	66.1
12	51.9	67.0	54.6	71.1	49.7	64.3	50.8	66.6
13	56.0	69.5	55.3	72.1	49.7	64.1	51.6	66.7
15	56.9	70.9	56.5	72.1	48.7	65.9	51.4	66.9

Table F.5. Utilisation factors for the sampling methods.

ANNEX G. RULE 18'

ANNEX G. RULE 18'

G1. Table G.1 shows results obtained by rule 18', the modified version of rule 18 (CUMRED), in a parallel method. Details of rule 18' are given in para 848.

PROJECT	PROJECT DURATION	UTILISATION FACTOR
1	3850	60.5
2	3409	62.0
3	3345	69.8
4	3513	69.6
5	7059	63.5
6	5551	60.0
7	6662	67.5
8	5312	71.3
9	2114	51.3
10	2708	51.4
11	2121	70.9
12	2120	68.1
13	3139	58.1
14	3927	45.7
15	3191	65.4
16	3218	67.2
17	1625	48.5
18	1970	50.6
19	1351	66.2
20	1531	63.8
21	2708	52.9
22	2458	51.1
23	2207	67.3
24	2271	62.7
25	1270	40.8
26	1584	35.4
27	1086	52.6
28	1000	46.7
29	2051	43.8
30	1919	46.5
31	1302	64.0
32	1229	62.5

Table G.1. Results for rule 18' in a parallel method.

ANNEX H. DISTRIBUTION OF SAMPLING SOLUTIONS

ANNEX H. DISTRIBUTION OF SAMPLING SOLUTIONS

H1. Tables H.1, H.2 and H.3 show project durations obtained by the sampling methods with rules 16(PROD), 15(RPOSWT) and 10(RANDOM) respectively. The project used had parameter values order strength S = 0.5, density D = 0.5 and resource factor rf = 0.5. (This project had identifier 19 in Table F.1.) Para 875 gives further details of the samples.

1491	1486	1439	1494	1573	1421	1344	1512	1444	1486	1382	1511	1537	1524	1471	1470	1420	1517
1479	1398	1400	1472	1388	1420	1410	1497	1523	1503	1479	1403	1463	1492	1473	1502	1477	1470
1462	1466	1489	1511	1436	1420	1435	1500	1485	1469	1467	1520	1515	1463	1478	1482	1429	1502
1363	1461	1455	1512	1447	1484	1499	1510	1487	1388	1475	1385	1395	1489	1426	1510	1440	1425
1506	1399	1486	1438	1401	1381	1385	1421	1485	1429	1361	1399	1432	1365	1488	1407	1407	1497
1389	1471	1524	1473	1436	1481	1417	1510	1500	1383	1456	1430	1390	1427	1421	1444	1416	1466
1394	1452	1435	1505	1394	1486	1489	1410	1488	1504	1453	1386	1392	1428	1378	1399	1400	1376
1391	1391	1359	1406	1483	1455	1465	1374	1417	1496	1419	1432	1467	1440	1432	1439	1400	1501
1427	1408	1415	1381	1439	1518	1437	1516	1503	1390	1413	1367	1459	1410	1452	1481	1408	1553
1461	1411	1517	1478	1443	1384	1397	1476	1429	1456	1388	1409	1466	1419	1410	1413	1455	1396
1430	1388	1421	1403	1387	1449	1511	1546	1365	1368	1392	1409	1391	1469	1448	1461	1403	1420
1501	1387	1489	1409	1539	1439	1449	1422	1421	1419	1441	1397	1412	1431	1442	1539	1390	1474
1440	1459	1427	1470	1410	1492	1459	1427	1478	1381	1501	1390	1432	1358	1504	1446	1421	1389
1392	1432	1378	1423	1377	1413	1454	1408	1401	1377	1419	1379	1533	1523	1390	1407	1378	1406
1418	1482	1408	1351	1514	1536	1406	1489	1501	1387	1497	1432	1429	1364	1424	1464	1395	1469
1433	1493	1416	1491	1460	1466	1414	1415	1556	1402	1451	1367	1387	1461	1407	1403	1493	1387
1455	1428	1376	1409	1539	1398	1533	1393	1494	1496	1366	1418	1390	1414	1374	1414	1549	1403
1407	1395	1502	1433	1386	1403	1526	1449	1374	1444	1442	1419	1466	1423	1418	1475	1531	1429
1430	1455	1364	1401	1461	1446	1370	1391	1417	1468	1502	1382	1461	1482	1368	1482	1386	1475
1536	1512	1411	1377	1481	1427	1524	1383	1553	1448	1402	1490	1437	1431	1453	1390	1515	1355
1441	1413	1421	1446	1375	1430	1365	1416	1461	1446	1511	1437	1411	1394	1455	1485	1448	1482
1531	1522	1357	1346	1434	1551	1428	1361	1443	1372	1424	1525	1445	1465	1365	1460	1492	1519
1427	1415	1380	1384	1417	1424	1520	1520	1441	1377	1442	1437	1395	1406	1417	1365	1457	1468
1417	1403	1379	1525	1401	1500	1379	1423	1471	1413	1510	1419	1473	1468	1369	1463	1454	1552
1445	1404	1394	1376	1388	1417	1398	1398	1495	1417	1385	1469	1442	1504	1480	1380	1375	1421
1427	1427	1401	1402	1445	1532	1433	1489	1467	1379	1493	1464	1376	1404	1460	1384	1337	1440
1452	1415	1359	1460	1347	1380	1464	1396	1415	1432	1453	1406	1403	1513	1379	1509	1405	1500
1478	1425	1404	1427	1360	1386	1396	1450	1404	1440	1449	1405	1393	1415	1436	1418	1530	1571
1391	1441	1399	1390	1495	1354	1451	1476	1422	1432	1408	1473	1465	1416	1456	1515	1413	1523
1417	1440	1445	1423	1407	1365	1446	1369	1418	1435	1421	1487	1462	1389	1425	1387	1471	1495
1477	1512	1496	1424	1387	1375	1419	1409	1429	1423	1476	1428	1385	1431	1441	1448	1456	1432
1485	1414	1578	1391	1528	1519	1357	1410	1497	1521	1385	1539	1422	1399	1415	1476	1398	1391
1435	1412	1464	1439	1458	1444	1404	1439	1455	1439	1452	1495	1472	1406	1367	1449	1415	1436
1403	1401	1514	1436	1362	1388												

Table H.1. Project durations with rule 16(PROD).

1431	1462	1494	1502	1615	1476	1452	1486	1434	1443	1456	1574	1491	1432	1536	1503	1485	1559
1463	1441	1532	1525	1534	1465	1607	1486	1495	1427	1419	1491	1437	1555	1528	1449	1427	1419
1377	1367	1559	1484	1444	1541	1444	1544	1389	1526	1474	1422	1458	1480	1606	1531	1604	1432
1491	1442	1463	1493	1518	1467	1616	1448	1407	1413	1468	1459	1492	1483	1387	1355	1421	1429
1453	1485	1419	1433	1477	1436	1598	1380	1481	1653	1445	1451	1623	1430	1472	1363	1460	1442
1475	1445	1484	1563	1545	1491	1398	1477	1519	1520	1425	1522	1433	1420	1504	1450	1454	1439
1476	1403	1447	1543	1467	1439	1427	1495	1401	1440	1474	1427	1476	1438	1533	1539	1480	1420
1536	1432	1543	1436	1379	1543	1538	1462	1434	1417	1416	1523	1528	1437	1489	1531	1519	1559
1488	1558	1407	1441	1506	1451	1535	1587	1535	1460	1497	1376	1508	1532	1458	1458	1440	1427
1473	1534	1480	1440	1404	1448	1472	1510	1536	1479	1429	1528	1440	1417	1458	1521	1464	1433
1465	1472	1485	1473	1470	1421	1434	1441	1602	1417	1464	1448	1478	1430	1440	1476	1426	1417
1511	1458	1559	1422	1607	1426	1398	1564	1445	1469	1448	1443	1434	1416	1419	1479	1467	1470
1474	1457	1561	1442	1395	1447	1458	1522	1499	1437	1447	1608	1421	1488	1468	1495	1510	1394
1457	1469	1464	1554	1421	1541	1453	1460	1469	1508	1460	1447	1523	1457	1436	1419	1408	1443
1478	1536	1452	1489	1517	1576	1424	1453	1508	1522	1449	1391	1445	1523	1488	1614	1472	1465
1436	1458	1475	1528	1459	1576	1469	1576	1460	1492	1486	1439	1461	1469	1417	1420	1414	1508
1429	1448	1469	1504	1459	1422	1412	1546	1556	1544	1499	1486	1553	1511	1511	1401	1481	1399
1465	1542	1475	1464	1541	1437	1377	1590	1564	1401	1485	1493	1513	1431	1427	1478	1505	1486
1401	1475	1473	1428	1453	1504	1499	1434	1447	1530	1552	1425	1441	1532	1437	1431	1513	1465
1392	1428	1445	1448	1467	1414	1508	1504	1474	1424	1532	1406	1577	1451	1526	1472	1475	1602
1514	1443	1585	1474	1511	1504	1511	1478	1446	1460	1544	1510	1545	1520	1542	1460	1451	1422
1449	1472	1516	1448	1489	1394	1452	1469	1482	1491	1460	1487	1455	1527	1436	1477	1443	1465
1419	1422	1457	1539	1495	1537	1514	1463	1477	1549	1481	1537	1420	1476	1415	1450	1475	1426
1449	1456	1417	1507	1411	1435	1385	1438	1615	1435	1561	1559	1504	1487	1386	1490	1587	1375
1418	1408	1478	1434	1454	1422	1640	1570	1511	1436	1498	1407	1433	1458	1484	1419	1401	1389
1445	1467	1335	1545	1524	1421	1512	1546	1401	1494	1604	1477	1477	1556	1452	1491	1545	1570
1405	1491	1469	1523	1424	1473	1577	1460	1471	1491	1434	1502	1497	1424	1463	1407	1412	1509
1512	1488	1507	1491	1539	1651	1495	1517	1402	1462	1456	1588	1466	1407	1512	1430	1439	1478
1484	1400	1476	1464	1456	1441	1494	1370	1458	1520	1442	1431	1490	1463	1456	1545	1460	1454
1500	1408	1444	1506	1440	1429	1463	1462	1529	1468	1482	1463	1443	1455	1523	1466	1474	1463
1501	1478	1520	1437	1433	1514	1527	1468	1393	1471	1515	1483	1555	1453	1514	1453	1466	1477
1421	1356	1438	1567	1474	1512	1460	1427	1388	1483	1515	1519	1535	1415	1519	1432	1559	1427
1443	1465	1393	1476	1540	1389	1492	1482	1477	1451	1446	1537	1471	1503	1476	1493	1467	1539
1417	1456	1461	1509	1505	1398												

Table H.2. Project durations with rule 15(RPOSWT).

1577	1457	1467	1571	1509	1519	1584	1555	1688	1529	1471	1634	1590	1632	1583	1479	1500	1682
1461	1578	1481	1545	1534	1607	1419	1472	1629	1535	1581	1644	1638	1494	1528	1486	1516	1581
1577	1508	1501	1723	1595	1575	1487	1497	1481	1613	1594	1560	1441	1537	1499	1487	1505	1659
1635	1515	1554	1504	1568	1586	1501	1467	1521	1467	1472	1514	1460	1553	1522	1498	1415	1541
1474	1596	1532	1515	1604	1562	1591	1445	1518	1649	1518	1708	1486	1630	1650	1762	1622	1631
1544	1544	1596	1537	1534	1560	1522	1653	1464	1557	1741	1462	1480	1551	1504	1484	1584	1445
1583	1586	1497	1638	1513	1490	1500	1599	1459	1647	1466	1574	1563	1547	1439	1525	1592	1611
1565	1490	1440	1585	1552	1492	1547	1556	1578	1565	1508	1622	1603	1455	1549	1512	1629	1472
1717	1592	1441	1484	1649	1684	1590	1553	1668	1645	1494	1496	1555	1496	1496	1557	1640	1461
1485	1610	1470	1542	1644	1562	1561	1621	1571	1563	1625	1541	1585	1544	1644	1568	1684	1647
1526	1479	1421	1550	1512	1462	1480	1513	1559	1601	1592	1548	1723	1482	1550	1472	1534	1513
1585	1484	1665	1505	1626	1577	1576	1447	1625	1645	1443	1573	1618	1520	1471	1541	1608	1497
1619	1496	1520	1504	1515	1631	1662	1607	1460	1656	1559	1481	1618	1487	1528	1591	1594	1575
1638	1551	1572	1540	1482	1579	1666	1589	1574	1739	1591	1459	1549	1455	1477	1587	1537	1590
1485	1606	1547	1509	1520	1588	1568	1491	1605	1651	1543	1599	1454	1467	1606	1555	1537	1650
1492	1581	1518	1475	1494	1570	1488	1639	1576	1597	1608	1615	1496	1506	1529	1592	1620	1531
1577	1455	1506	1463	1624	1522	1521	1555	1636	1574	1509	1564	1492	1625	1515	1557	1490	1582
1430	1451	1483	1566	1408	1602	1567	1438	1481	1604	1562	1612	1606	1489	1513	1507	1539	1512
1525	1478	1611	1505	1468	1638	1621	1525	1534	1527	1613	1519	1476	1532	1509	1539	1558	1523
1643	1496	1477	1585	1490	1536	1412	1576	1643	1565	1614	1507	1609	1487	1509	1497	1588	1515
1464	1633	1459	1426	1520	1479	1678	1551	1599	1556	1470	1441	1515	1523	1599	1464	1491	1452
1561	1525	1465	1494	1493	1631	1519	1592	1506	1626	1574	1509	1521	1442	1424	1594	1484	1635
1570	1592	1603	1554	1548	1464	1604	1540	1527	1714	1638	1496	1505	1522	1524	1591	1493	1661
1514	1469	1631	1583	1488	1611	1465	1560	1514	1582	1519	1426	1579	1495	1675	1679	1525	1492
1489	1605	1493	1535	1595	1603	1481	1592	1647	1662	1545	1576	1546	1508	1470	1572	1604	1572
1425	1497	1531	1508	1561	1625	1483	1580	1470	1530	1511	1624	1565	1661	1399	1537	1598	1500
1540	1582	1553	1605	1488	1613	1432	1590	1422	1457	1639	1459	1604	1586	1530	1550	1633	1516
1447	1665	1526	1489	1500	1588	1666	1654	1560	1573	1578	1444	1616	1491	1503	1495	1514	1630
1569	1591	1466	1619	1560	1531	1626	1437	1523	1548	1636	1518	1602	1425	1612	1511	1555	1525
1550	1648	1403	1584	1454	1515	1553	1513	1551	1602	1557	1580	1514	1543	1631	1551	1486	1467
1585	1640	1561	1646	1616	1462	1647	1564	1432	1589	1598	1483	1562	1452	1504	1477	1600	1678
1567	1507	1499	1567	1502	1645	1611	1504	1515	1657	1471	1446	1503	1670	1488	1545	1589	1565
1528	1581	1563	1624	1465	1545	1500	1518	1538	1673	1580	1535	1524	1394	1643	1578	1556	1570
1620	1505	1609	1470	1619	1542	1634	1540	1525	1510	1438	1461	1637	1516	1486	1607	1634	1616

Table H.3. Project durations with rule 10(RANDOM).

1603	1551	1578	1461	1484	1653	1436	1583	1494	1559	1417	1570	1560	1548	1681	1499	1463	1652
1496	1609	1557	1613	1565	1525	1541	1538	1601	1519	1509	1434	1501	1506	1423	1452	1567	1575
1561	1443	1525	1501	1523	1512	1483	1562	1598	1417	1595	1628	1441	1642	1516	1520	1458	1485
1482	1508	1522	1631	1463	1516	1479	1492	1503	1576	1477	1558	1575	1522	1566	1552	1535	1570
1498	1431	1503	1546	1535	1451	1583	1510	1567	1546	1523	1626	1592	1618	1639	1511	1600	1470
1648	1538	1440	1565	1534	1465	1605	1460	1499	1515	1566	1585	1665	1493	1601	1486	1539	1546
1562	1541	1612	1480	1467	1571	1539	1634	1425	1641	1457	1603	1539	1679	1436	1444	1535	1459
1618	1500	1542	1494	1509	1502	1614	1613	1493	1466	1586	1460	1576	1573	1514	1666	1480	1523
1488	1568	1510	1538	1479	1515	1501	1473	1516	1601	1494	1635	1496	1602	1528	1536	1475	1697
1628	1519	1547	1451	1450	1518	1547	1735	1477	1522	1526	1699	1544	1482	1536	1575	1427	1442
1532	1536	1639	1542	1530	1513	1497	1581	1622	1562	1576	1484	1480	1530	1438	1565	1485	1570
1464	1553	1617	1494	1602	1632	1502	1556	1445	1489	1514	1567	1602	1550	1598	1652	1607	1530
1501	1569	1549	1444	1510	1682	1513	1532	1514	1544	1534	1475	1574	1545	1487	1476	1539	1525
1409	1509	1496	1463	1617	1516	1582	1560	1573	1570	1688	1462	1495	1480	1580	1558	1490	1561
1523	1481	1446	1516	1484	1582	1568	1609	1529	1480	1547	1622	1603	1536	1567	1626	1455	1507
1591	1628	1581	1564	1549	1548	1573	1585	1498	1428	1596	1549	1668	1480	1548	1492	1519	1519
1552	1654	1485	1481	1542	1536	1541	1530	1589	1522	1492	1561	1504	1490	1563	1580	1478	1678
1585	1511	1540	1621	1573	1587	1461	1575	1512	1548	1637	1439	1542	1596	1503	1490	1623	1515
1575	1574	1479	1620	1543	1640	1504	1540	1543	1541	1632	1536	1590	1522	1533	1493	1532	1559
1545	1456	1547	1602	1554	1585	1576	1614	1637	1562	1477	1438	1508	1508	1467	1533	1630	1574
1566	1546	1553	1549	1602	1589	1500	1710	1629	1694	1589	1722	1650	1440	1581	1470	1493	1531
1632	1463	1598	1586	1591	1528	1559	1431	1444	1575	1620	1512	1609	1537	1562	1614	1646	1497
1499	1590	1542	1485	1487	1513	1528	1562	1597	1526	1554	1560	1567	1602	1627	1579	1501	1524
1683	1499	1590	1583	1549	1470	1519	1480	1473	1507	1688	1522	1446	1452	1593	1479	1415	1517
1606	1529	1459	1487	1634	1573	1648	1521	1523	1499	1599	1445	1492	1517	1490	1452	1592	1567
1464	1499	1461	1602	1583	1536	1486	1603	1639	1609	1615	1528	1538	1542	1473	1633	1681	1488
1524	1502	1572	1570	1559	1532	1586	1499	1533	1492	1684	1566	1560	1538	1539	1541	1490	1518
1563	1528	1584	1586	1512	1497	1522	1470	1611	1468	1559	1543	1581	1452	1607	1544	1644	1520
1520	1422	1458	1474	1530	1605	1464	1415	1410	1570	1533	1573	1559	1510	1534	1616	1594	1474
1534	1677	1574	1533	1585	1482	1581	1486	1455	1533	1634	1579	1528	1577	1537	1580	1578	1520
1650	1554	1562	1568	1479	1509	1726	1673	1484	1587	1528	1512	1574	1496	1423	1483	1613	1436
1388	1598	1492	1494	1528	1530	1641	1671	1510	1480	1600	1475	1521	1548	1429	1540	1588	1454
1560	1477	1518	1527	1467	1710	1558	1489	1547	1527	1478	1597						

Table H.3. Project durations with rule 10(RANDOM) (contd.)

ANNEX I. COMPARISON OF METHODS

ANNEX I. COMPARISON OF METHODS

I1. Table I.1 shows project durations of corresponding schedules obtained by the parallel and sampling methods. Within each pair, the same project was scheduled, using the same priority rule, by both methods.

PAIRED PROJECT DURATIONS

PARL	SAMP	PARL	SAMP	PARL	SAMP	PARL	SAMP
2935	2554	2674	2619	2899	2569	2691	2507
2760	2629	3325	2623	3057	2546	2503	2526
2736	2665	2970	2683	2610	2553	2833	2686
2610	2488	2719	2534	2684	2447	2682	2438
2316	2082	2306	2122	2189	2140	2234	2083
2267	2125	2613	2120	2337	2119	2279	2133
2218	2128	2279	2107	2219	2104	2289	2155
2219	2078	2190	2023	2139	2036	2116	2066
3429	3262	3408	3306	3720	3315	3450	3253
3747	3395	3645	3300	3721	3289	3694	3363
3538	3328	3537	3266	3364	3160	3918	3340
3381	3295	3683	3172	3525	3223	3229	3183
2951	2916	3328	2883	3162	2889	3276	3014
3282	2893	3452	2998	3203	2912	3167	2933
3299	2949	3335	2924	3207	2953	3359	2937
3204	2895	3286	2932	3118	2897	3510	2944
1779	1666	1658	1602	1797	1609	1684	1654
1800	1658	2103	1552	1990	1566	1713	1670
1554	1561	1904	1672	1735	1594	2062	1584
1735	1586	1744	1591	1669	1618	1650	1570
1772	1609	1625	1594	1407	1414	1398	1392
1520	1372	1381	1414	1464	1430	1714	1412
1608	1416	1494	1454	1441	1407	1599	1436
1424	1350	1452	1390	1424	1394	1393	1369
1382	1357	1463	1344	1518	1425	1351	1375
2896	2499	2694	2599	2853	2521	2844	2692
2822	2485	2902	2499	2954	2570	3030	2586
2729	2553	2837	2565	2614	2456	2857	2474
2614	2437	2725	2444	2757	2409	2803	2546
2458	2511	2208	2127	2266	2105	2240	2143
2288	2124	2376	2104	2526	2179	2316	2119
2352	2161	2320	2150	2262	2095	2318	2153
2418	2137	2318	2135	2327	2154	2270	2130
2265	2155	2310	2108	2304	2104	1398	1403
1398	1408	1599	1415	1599	1421	1424	1389
1382	1355						

Table I.1. Project durations for parallel methods and corresponding sampling methods.

ANNEX J. PROBLEMS FROM THE LITERATURE

ANNEX J. PROBLEMS FROM THE LITERATURE

J1. A set of problems that have appeared in the open literature were scheduled with parallel and sampling methods. Details of the problems are given in Table J.1. Scheduling results are given in Table J.2; all results quoted are project durations.

PROBLEM	REFERENCE	JOBS	RESOURCES
1	Heller and Logemann (1962)	9	5
2,3,4*	Martino (1965), quoted in Wiest (1967)	16	1
5	Davis and Heidorn (1971)	5	3
6	Moodie and Mandeville (1966)	7	1
7,8	Moder and Phillips (1970)	11	2,3
9	Mason and Moodie (1971)	11	1
10	Ciobanu (1971)	27	2
11,12,13**	Karni (1973)	53	3

Table J.1. Details of problems.

* These three problems had different resource availabilities (7,8 and 9), but otherwise they were the same.

** These three problems had different resource availabilities (7,8 and 9, with R(1)=R(2)=R(3) in each case), but otherwise they were the same.

METHOD	RULE	PROBLEM 1	2	3	4	5	6	7	8	9	10	11	12	13
PARALLEL	1	9	29	19	22	9	9	20	22	23	40	46	41	40
	2	9	29	21	19	7	8	18	22	23	44	49	45	44
	3	10	27	25	20	9	9	20	20	23	42	52	47	47
	4	9	27	22	21	7	9	20	23	24	43	44	40	40
	5	9	32	31	26	8	9	20	23	23	43	48	44	44
	6	12	29	29	24	9	11	29	20	23	50	53	50	50
	7	9	29	26	24	7	8	20	23	24	57	55	49	49
	8	9	28	22	21	7	9	22	24	23	43	58	42	41
	9	9	29	24	21	7	8	22	24	25	43	49	44	42
	10	9	30	25	26	7	8	20	22	25	47	51	47	47
	11	9	28	22	21	7	9	22	24	23	43	44	42	41
	12	9	26	24	23	8	9	27	27	24	43	54	51	41
	13	9	28	22	21	7	9	22	24	23	43	44	42	41
	14	9	29	22	21	8	9	27	27	24	43	48	41	40
	15	9	26	21	20	8	8	18	21	24	44	51	47	44
	16	9	26	21	20	8	8	18	22	24	43	46	43	41
	17	9	29	24	22	7	8	20	23	23	45	55	49	49
	18	9	26	19	19	10	8	18	20	22	45	49	47	47
	19	12	29	26	25	9	14	19	20	26	53	68	42	47
	20	9	28	22	21	7	9	22	23	24	46	44	43	41
	21	12	29	31	24	9	14	19	20	23	53	60	42	41

METHOD	RULE	PROBLEM 1	2	3	4	5	6	7	8	9	10	11	12	13
PARALLEL	22	12	29	26	25	9	14	19	20	26	53	60	42	41
	23	9	31	22	21	9	11	19	24	24	40	44	45	40
	24	9	28	22	21	7	9	22	24	23	43	44	42	41
	25	9	29	24	22	9	9	22	22	23	44	49	40	39
	26	12	29	29	20	9	11	29	20	23	48	52	49	48
SAMP-LING	10	9	26	19	19	7	8	18	19	22	40	45	40	39
	16	9	26	19	19	7	8	18	19	22	41	44	41	39
BEST PARALLEL		9	26	19	19	7	8	18	20	22	40	44	40	39
BEST SAMPLING		9	26	19	19	7	8	18	19	22	40	44	40	39
REPORTED OPTIMUM		9	25	19	19	7	8		19*		39			

Table J.2. Problem solutions.

* Moder and Phillips (1970, Exercise 2, pp 178-179) quote 21 as the optimum. However, a schedule with duration 19 exists, and this can be proven optimal.

BIBLIOGRAPHY

BIBLIOGRAPHY

ARCUS, A.L. (1966), "COMSOAL - A Computer Method of Sequencing Operations for Assembly Lines", International Journal of Production Research, Vol. 4, p. 259.

BALAS, E. (1970), "Project Scheduling with Resource Constraints", Chapter 5 of "Applications of Mathematical Programming Techniques", E.M.L. BEALE(Ed.), American Elsevier, New York.

BATTERSBY, A. (1967), "Network Analysis for Planning and Scheduling", 2nd edition, Macmillan, London.

BENNINGTON, G.E., and McGINNIS, L.F. (1972), "A Critique of Project Planning with Constrained Resources", North Carolina State University at Raleigh, Operations Research Report No. 81, May 15.

BERGE, C. (1962), "The Theory of Graphs and Its Applications", Methuen, London.

BOWMAN, E.H. (1959), "The Schedule-Sequencing Problem", Operations Research, Vol. 7, p. 621.

CIOBANU, G. (1971), "Project Scheduling with Resource Allocation", Economic Computation and Economic Cybernetics Studies and Research, No. 1, p. 81.

CIOBANU, G. (1972), "A Multi-Project Programming Model with Resources Levelling", Economic Computation and Economic Cybernetics Studies and Research, No. 3, p. 61.

CONWAY, R.W. and MAXWELL, W.L. (1962), "Network Dispatching by the Shortest Operation Discipline", Operations Research, Vol. 10, p. 51.

CONWAY, R.W., MAXWELL, W.L. and MILLER, L.W. (1967), "Theory of Scheduling", Addison-Wesley, Reading, Mass.

COOPER, D.F. (1969), "Matrix Solutions of Project Networks without Topological Ordering", unpublished Honours project, Department of Computing Science, University of Adelaide.

CROW, E.L., DAVIS, F.A. and MAXFIELD, M.W. (1960), "Statistics Manual", Dover, New York.

CULLINGFORD, G. and PRIDEAUX, J.D.C.A. (1973), "A Variational Study of Optimal Resource Profiles", Management Science, Vol. 19, p. 1067.

DAVIS, E.W. (1966), "Resource Allocation in Project Network Models - A Survey", Journal of Industrial Engineering, Vol. XVII, p. 177.

DAVIS, E.W. and HEIDORN, G.E. (1971), "An Algorithm for Optimal Project Scheduling under Multiple Resource Constraints", Management Science, Vol. 17, p. B-803.

FENDLEY, L.G. (1968), "Towards the Development of a Complete Multi-project Scheduling System", Journal of Industrial Engineering, Vol. 19, p. 505.

FERNANDEZ, E.B. and BUSSELL, B. (1973), "Bounds on the Number of Processors and Time for Multiprocessor Optimal Schedules",

IEEE Transactions on Computers, Vol. C-22, p. 745.

FISHER, H. and THOMPSON, G.L. (1963), "Probabilistic Learning Combinations of Local Job Shop Scheduling Rules", Chapter 15 of Muth and Thompson (1963).

FRIZELLE, G. (1968), "The Concept of Paths in Networks of the Metra Potential Method Type", Metra, Vol. 7, p. 75.

FULKERSON, D.R. (1962), "Expected Critical Path Lengths in PERT Networks", Operations Research, Vol. 10, p. 808.

GERE, W.S., Jr. (1966), "Heuristics in Job Shop Scheduling", Management Science, Vol. 13, p. 167.

GIFFLER, B. and THOMPSON, G.L. (1960), "Algorithms for Solving Production Scheduling Problems", Operations Research, Vol. 8, p. 487.

GORENSTEIN, S. (1972), "An Algorithm for Project (Job) Sequencing with Resource Constraints", Operations Research, Vol. 20, p. 835.

GUTJAHR, A.L. and NEMHAUSER, G.L. (1964), "An Algorithm for the Line Balancing Problem", Management Science, Vol. 11, p. 308.

HELD, M., KARP, R.M. and SHARESHIAN, R. (1963), "Assembly Line Balancing - Dynamic Programming with Precedence Constraints", Operations Research, Vol. 11, p. 442.

HELGESON, W.B. and BIRNIE, D.P. (1961), "Assembly Line Balancing using the Ranked Positional Weight Technique", Journal of Industrial Engineering, Vol. XII, p. 394.

HELLER, J. and LOGEMANN, G. (1962), "An Algorithm for the Construction and Evaluation of Feasible Schedules", Management Science, Vol. 8, p. 168.

JOHNSON, T.J.R. (1967), "An Algorithm for the Resource Constrained Project Scheduling Problem", Ph.D. Thesis, Massachusetts Institute of Technology.

KARNI, R. (1973), "Heuristic Resource Analysis", Management Informatics, Vol. 2, p. 57.

KAUFMANN, A. (1967), "Graphs, Dynamic Programming and Finite Games", Academic Press, New York.

KELLEY, J.E., Jr. (1961), "Critical Path Planning and Scheduling: Mathematical Basis", Operations Research, Vol. 9, p. 296.

KELLEY, J.E., Jr. (1963), "The Critical Path Method: Resources Planning and Scheduling", Chapter 21 of Muth and Thompson (1963).

KELLEY, J.E., Jr. and WALKER, M.R. (1959), "Critical Path Planning and Scheduling", Proceedings of the Eastern Joint Computer Conference, December 1959, p. 160.

KILBRIDGE, M.D. and WESTER, L. (1961), "A Heuristic Method of Assembly Line Balancing", Journal of Industrial Engineering, Vol. XII, p. 293.

LERDA-OLDBERG, S. (1966), "Bibliography on Network-Based Project Planning and Control Techniques: 1962-1965", Operations Research, Vol. 14, p. 925.

LEVY, F.K., THOMPSON, G.L. and WIEST, J.D. (1962), "Multiship, Multishop, Workload Smoothing Program", Naval Research Logistics Quarterly, Vol. 9, p. 37.

LEVY, F.K., THOMPSON, G.L. and WIEST, J.D. (1963a), "Introduction to the Critical Path Method", Chapter 20 of Muth and Thompson (1963).

LEVY, F.K., THOMPSON, G.L. and WIEST, J.D. (1963b), "Mathematical Basis of the Critical Path Method", Chapter 22 of Muth and Thompson (1963).

MACASKILL, J.L.C. (1969), "Application of Computers to the Analysis of Mixed-product Assembly Lines", Proceedings of the Fourth Australian Computer Conference, Adelaide, South Australia, p. 61.

MACASKILL, J.L.C. (1973), "Diagonal-matrix Formulation of Networks for Storage Economies", Australian Computer Journal, Vol. 5, p. 18.

MALCOLM, D.G., ROSEBOOM, J.H., CLARK, C.E. and FAZAR, W. (1959), "Applications of a Technique for Research and Development Program Evaluation", Operations Research, Vol. 7, p. 646.

MANNE, A.S. (1960), "On the Job Shop Scheduling Problem", Operations Research, Vol. 8, p. 219.

MANSOOR, E.M. (1964), "Assembly Line Balancing - An Improvement in the Ranked Positional Weight Technique", Journal of Industrial Engineering, Vol. XV, p. 73.

MASON, A.T. and MODDIE, C.L. (1971), "A Branch and Bound Algorithm for Minimising Cost in Project Scheduling", Management Science, Vol. 18, p. B-158.

MASTOR, A.A. (1970), "An Experimental Investigation and Comparative Evaluation of Production Line Balancing Techniques", Management Science, Vol. 16, p. 728.

McROBERTS, K.L. (1971), "A Search Model for Evaluating Combinatorially Explosive Problems", Operations Research, Vol. 19, p. 1331.

MODER, J.J. and PHILLIPS, C.R. (1964), "Project Management with CPM and PERT", Van Nostrand Reinhold, New York.

MODER, J.J. and PHILLIPS, C.R. (1970), "Project Management with CPM and PERT", 2nd edition, Van Nostrand Reinhold, New York.

MODDIE, C.L. and MANDEVILLE, D.E. (1966), "Project Resource Balancing by Assembly Line Balancing Techniques", Journal of Industrial Engineering, Vol. XVII, p. 377.

MUTH, J.F. and THOMPSON, G.L. (Eds.) (1963), "Industrial Scheduling", Prentice-Hall, Englewood Cliffs, N.J.

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

PRITSKER, A.A.B., WATTERS, L.J. and WOLFE, P.M. (1969), "Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach", Management Science, Vol. 16, p. 93.

ROY, B. (1966), "Prise en compte des contraintes disjonctives dans les

mêthodes de chemin critique", Revue Francaise de Recherche Operationelle, Vol. 38, p. 69.

SCHRAGE, L. (1970), "Solving Resource-Constrained Network Problems by Implicit Enumeration - Nonpreemptive Case", Operations Research, Vol. 18, p. 263.

SNEDECOR, G.W. and COCHRAN, W.G. (1967), "Statistical Methods", 6th edition, Iowa State University Press, Ames, Iowa.

THESEN, A. (1973), "Scheduling of Computer Programs for Optimal Machine Utilisation", BIT, Vol. 13, p. 206.

TONGE, F.M. (1965), "Assembly Line Balancing using Probability Heuristics", Management Science, Vol. 11, p. 727.

TUKEY, J.W. (1949), "One Degree of Freedom for Non-Additivity", Biometrics, Vol. 5, p. 232.

VERHINES, D.R. (1963), "Optimum Scheduling of Limited Resources", Chemical Engineering Progress, Vol. 59, p. 65.

WAGNER, H.M. (1959), "An Integer Linear Programming Model for Machine Scheduling", Naval Research Logistics Quarterly, Vol. 6, p. 131.

WAGNER, H.M., GIGLIO, R.J. and GLASER, R.G. (1964), "Preventive Maintenance Scheduling by Mathematical Programming", Management Science, Vol. 10, p. 316.

WIEST, J.D. (1967), "A Heuristic Model for Scheduling Large Projects with Limited Resources", Management Science, Vol. 13, p. B-359.

WINER, B.J. (1970), "Statistical Principles in Experimental Design", McGraw-Hill, London.

WOODGATE, H.S. (1973), "Trends and Developments in Network Planning Systems", Management Informatics, Vol. 2, p. 106.

ZALOOM, V. (1971), "On the Resource Constrained Project Scheduling Problem", AIIE Transactions, Vol. 3, p. 302.