Management Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Multi-Project Scheduling: Categorization of Heuristic Rules Performance

I. Kurtulus, E. W. Davis,

MANAGEMENT SCIENCE
Vol 28, No 2, February 1982
Printed in U S A

MULTI-PROJECT SCHEDULING: CATEGORIZATION OF HEURISTIC RULES PERFORMANCE*

I. KURTULUS† AND E. W. DAVIS‡

Application of heuristic solution procedures to the practical problem of project scheduling has previously been studied by numerous researchers. However, there is little consensus about their findings, and the practicing manager is currently at a loss as to which scheduling rule to use. Furthermore, since no categorization process was developed, it is assumed that once a rule is selected it must be used throughout the whole project. This research breaks away from this tradition by providing a categorization process based on two powerful project summary measures. The first measure identifies the location of the peak of total resource requirements and the second measure identifies the rate of utilization of each resource type The performance of the rules are classified according to values of these two measures, and it is shown that a rule introduced by this research performs significantly better on most categories of projects.
(PROJECT MANAGEMENT, RESEARCH AND DEVELOPMENT)

1. Introduction

Ever since the development of critical path methods in the 1950s research has been conducted on resource-constrained multi-project scheduling problems ([3], [5], [6], [8]–[10], [18]). While this is the most complex of the various versions of project scheduling problems which have been studied, it is also among the most realistic. Its importance has increased in recent years with the increasing costs of both direct labor resources and indirect costs which tend to increase the cost of project delay, often disproportionately.

A multi-project problem consists of a number of projects, and a project is defined as a collection of activities, which consume resources, and events which constitute points in time. There are two possible ways of representing each project: activity-on-arc (A-O-A) and activity-on-node (A-O-N). In this paper, an A-O-N representation is adopted with deterministic activity times and precedence relationships; that is, "deterministic activity networks" as defined in Elmaghraby [7] are used. It is assumed that once an activity is started, its progress is not interrupted. Also the number of different resources required and the amount required of each resource by an activity are assumed to be constant. The amount available per period in each resource category is also constant, but there is no limit on the total amount that can be expended.

Then, a "constrained" project schedule is an assignment of a start time for each activity in the network such that the precedence and resource requirements are satisfied. However, since the resources available are limited in amount, the start time of a number of activities may be delayed beyond the "unconstrained" schedule, possibly causing delays beyond the original duration of the project critical path.

Using the definitions given above, the resource-constrained multi-project scheduling problem in this paper is defined as one of scheduling two or more projects simultaneously (i.e., starting at the same point in time and continuing thereafter), under the objective of minimizing total project delay.

* Accepted by Burton V Dean; received February 1, 1980 This paper has been with the authors 10 months for 2 revisions.

† Virginia Commonwealth University
‡ University of Virginia.

0025-1909/82/2802/0161$01 25

The projects are limited only through their dependence upon a common pool of available resources of each category. If, in contrast, the separate projects are artificially combined into one large project for scheduling purposes (by the addition of dummy "start" and "end" activities) this approach is termed a "single-project" approach. The two approaches will produce different schedules with some heuristic sequencing rules. For example, if the MINSLK rule is used, there will be only one critical path under the single-project approach, but there will be multiple critical paths under the multi-project approach.

Various optimizing techniques have been used to attack the single-project resource-constrained problem (Davis [2], Patterson and Huber [18], Stinson [19]), but these techniques have generally been successful only on small problems containing at most 50 activities (Talbot [20]). Thus major efforts have been directed towards developing heuristic procedures which produce only "good" feasible solutions. There are in existence today literally hundreds of different heuristic-based scheduling rules applicable to the single-project problem (Davis and Patterson [6], Jennett [11]). In contrast, relatively little research has been published on rules developed specifically for the multi-project problem, and the results vary widely. Among the research discussed in the open literature, for example, Mize [14] found that FCFS (First Come First Served) produced minimum total delay among his simple rules. In contrast, Fendley [8] used eight different objective functions and found that MINSLK (Minimum Slack First)[1] produced best results under five of them, including total delay. In another published work Patterson [16] using a single-project approach, tested four different objective functions and found that SOF (Shortest Operation First) minimized total delay. While the reasons for these different findings have not been thoroughly analyzed, we suspect a primary one could be differences in the characteristics of the projects used in these studies, since each researcher used a unique set of projects.

In this paper we present a method for classifying the performance of multi-project scheduling rules according to project characteristics. First, we will describe two new project summary measures which were developed in the course of this research. Then the design of the experiment is discussed, including the generation of project data, the underlying experimental model and the statistical tests. Finally, the results are presented, along with the categorization process.

2. Summary Measures

The concept of a project summary measure for categorization of scheduling results' was first introduced by Pascoe [15] and utilized by Johnson [12]. Davis [5] extended these concepts and proposed an associated scheme for predicting the performance of scheduling rules, as did Patterson [17]. However, most of these measures were limited to the single-project case and were not useful for our purposes here. Thus, to allow for the multi-project nature of the problem, we propose two new project summary measures based on resource profiles generated by a "time-only" analysis.[2] The first of these measures is termed the Average Resource Load Factor (ARLF), and it identifies whether the peak of total resource requirements is located in the first or second half of the project's original, critical path duration. It is defined as follows:

Let,

$$Z_{ijt} = \begin{cases} 1 & \text{if} \quad t \geqslant CP_i/2, \\ -1 & \text{if} \quad t < CP_i/2, \end{cases}$$

[1] Slack is determined as in standard critical path analysis.

[2] By definition, this is the resource profile obtained by scheduling each activity at its early start time. Hence another name for this schedule is an "ALL-EST" schedule.

and

$$X_{ijt} = \begin{cases} 1 & \text{if activity } j \text{ of project } i \text{ is active at time } t, \\ 0 & \text{otherwise,} \end{cases}$$

where CP_i is the critical path duration of project i. If K_{ij} denotes the number of types of resources required by activity j of project i, then the ARLF for project i is defined as;

$$\mathrm{ARLF}_i = \frac{1}{CP_i} \sum_{t}^{CP_i} \sum_{k=1}^{K_{ij}} \sum_{j=1}^{N_i} Z_{ijt} X_{ijt} \left(\frac{r_{ijk}}{K_{ij}} \right) \tag{1}$$

where $Z_{ijt} X_{ijt} = (-1, 0, 1)$ and N_i is the number of activities in project i and r_{ijk} is the amount of resource of type k required by activity j of project i. The value of ARLF for a collection of projects is simply the average of the individual project values. If all r_{ijk} are represented in dollar terms, then it is easy to see that ARLF can also be used for locating the peak of the project's total cash requirements.

Our second summary measure, termed the Average Utilization Factor (AUF), indicates the average "tightness" of the constraints on each required resource. The Utilization Factor (UF)[3] is calculated for each problem (i.e., project set) and each resource type as the ratio of the total amount required to the amount available in each time period based on a time-only analysis over the original CP durations of all projects. It is easy to see that if UF $\leqslant 1$ for each resource in each time period, then zero project delay will be obtained. Also if two problems have the same UF value for each resource in each time period and if the resources are constraining, then the same delay should be obtained for each problem when the same sequencing rule is used.[4] This is an important property since if really different multi-project problems are to be used in testing various sequencing rules, the problems should have different UF values.

But one major disadvantage of the UF as defined above is that severe computational problems will be encountered if it is used on real-life (large) projects. So a form of averaging is adapted to make it computationally feasible. It may be observed that natural "break points" in the total resource usage profile of a mutli-project problem occur as individual projects are completed. In order to define the resource requirements between these break points the critical path durations of the projects are arranged in ascending order, and the following intervals are created: $S_1 = CP_1$, $S_2 = CP_2 - CP_1, \ldots,$ and $S_M = CP_M - CP_{M-1}$ (where M denotes the number of projects in the problem). Then the total amount of resource of type k required over any interval L is given by $W_{S_L,k}$ where:

$$W_{S_L,k} = \sum_{t=a}^{b} \sum_{i=1}^{M} \sum_{j=1}^{N_i} r_{ijk} X_{ijt}$$

where $a = CP_{L-1} + 1$ and $b = CP_L$, and

$$\mathrm{AUF}_k = 1/M \sum_{L=1}^{M} W_{S_L,k} / (R_{\max,k} \times S_L). \tag{2}$$

The test projects used in this study had a range of 0.6 to 1.6 for the AUF factor, which is not inconsistent with values encountered in actual industry projects, according to the limited available evidence (Davis [5]).

[3] This term was first defined by Davis [5].

[4] One qualification is necessary: if different tie-breaking rules are used, a different schedule *may* be obtained.

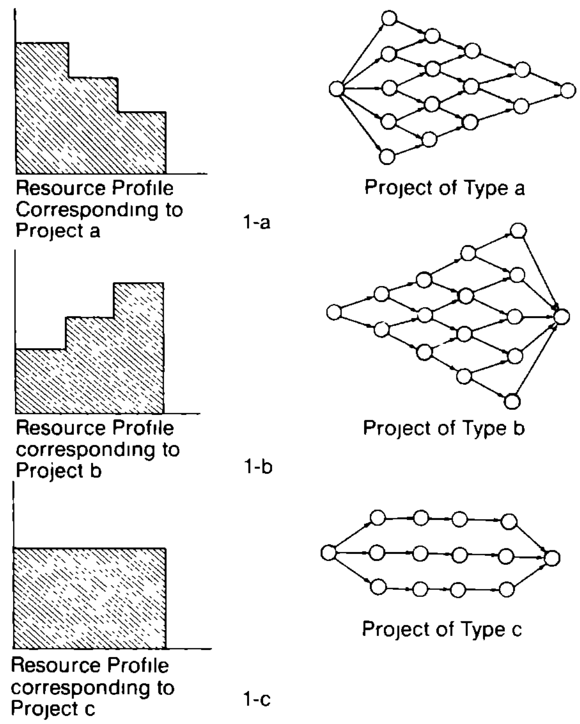

FIGURE 1. Categories of Project Shapes and the Corresponding Resource Profiles.

3. Generation of Test Projects

In order to generate test projects for this study the important characteristics of projects used in previous research (i.e., Davis [2], Mize [14], Pascoe [15], Patterson [16]), were first examined and summarized. These summary data included such measures as the number of activities, number of parallel paths, activity time and resource requirement distributions and complexity (as defined by Davis [5]).

The range of values for the ARLF was then generated by using selected values of project summary measures with the 3 categories of project shapes given in Figure 1. Since the time and resource requirements of the activities were "compatible,"[5] when all projects were chosen from the first category (Figure 1a) a minimum value for ARLF was determined as -3.0 and when they were chosen from the second category (Figure 1b), a maximum value of ARLF of $+3.0$ was determined. In the next step, this range was arbitrarily divided into 7 equal intervals, the midpoints of which were set at -3, -2, -1, 0, 1, 2, 3. As each project set ("problem") was generated, the data was modified to meet one of these desired ARLF values. Once the target value of ARLF was reached, the values of $W_{S_L,k}$, S_L and M in (2) above were determined. Finally, resource availabilities ($R_{\max,k}$) were obtained by setting AUF equal to values between 0.6 and 1.6 and solving. Since some initial tests suggested that project delay was much more sensitive to changes in the value of the AUF than ARLF, the range of AUF was divided into eleven equal intervals, with midpoints 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6. Among the 77 project sets generated by repeating 11 levels of AUF for each level of ARLF, the total number of activities per problem varied between 34 and 63 activities.

[5]Compatible here implies a range of values without tremendous variation, e.g., a uniform distribution between 2 and 6 units for different types of resources satisfied this requirement. Similar values were used by Davis [2] and Patterson [16].

TABLE 1

List of the Scheduling Rules Used in the Study

SOF = (Shortest Operation First) By definition, SOF = Min d_{ij} where d_{ij} is the duration of the jth activity from the ith project (Conway [1]), Patterson [16]). Tie-breaker: FCFS

MINSLK = (Minimum Slack First) MINSLK = Min SLK_{ij}, where $\text{SLK}_{ij} = \text{LST}_{ij} - \text{Max}(\text{EST}_{ij}, \text{TIME})$ and terms are as defined in standard CPM (Wiest [21], Fendley [8], Davis and Patterson [6]). Tie-breaker: FCFS.

SASP = (Shortest Activity from Shortest Project) SASP = Min f_{ij} and $f_{ij} = CP_i + d_{ij}$ where CP_i is the duration of critical path of the project (Kurtulus [13]). Tie-breaker: FCFS.

LALP = (Longest Activity from the Longest Project) LALP = Max f_{ij} and $f_{ij} = CP_i + d_{ij}$ as defined in SASP (Kurtulus [13]). Tie-breaker:

$$\text{GRES} = \max \sum_{k=1}^{K} r_{ijk}.$$

MOF = (Maximum Operation First) MOF = Max d_{ij} (Kurtulus [13]). Tie-breaker: GRES.

MAXSLK = (Maximum Slack First) MAXSLK = Max SLK_{ij} (Kurtulus [13]). Tie-breaker: GRES.

MINTWK = (Minimum Total Work Content) MINTWK = Min f_{ij}. Let r_{ijk} denote the resource required by the activity, then,

$$f_{ij} = \text{TWK}_i + d_{ij} \sum_{k=1}^{K} r_{ijk}$$

and

$$\text{TWK}_i = \sum_{k=1}^{K} \sum_{j \in AS_i} d_{ij} r_{ijk}$$

defines the total work content of the activities already scheduled (i.e., AS_i) from project i (Kurtulus [13]). Tie-breaker. FCFS

MAXTWK = (Maximum Total Work Content) MAXTWK = Max f_{ij} and f_{ij} is as defined in MINTWK (Kurtulus [13]). Tie-breaker: FCFS.

FCFS = (First Come First Served) First eligible activity is assigned the highest priority (Mize [14]). Tie-breaker: random.

4. Sequencing Rules Used

The sequencing rules tested in this study included 3 popular rules used by previous researchers (i.e., SOF, MINSLK, FCFS) as well as 6 new rules developed in this research especially for the multi-project problem (i.e., SASP, LALP, MOF, MAXSLK, MINTWK, MAXTWK). A definition of each rule is given in Table 1.

5. The Underlying Model and the Statistical Testing Procedure

The underlying experimental design for this research can be summarized in terms of a three-factor model. The factors consist of the rules, and the two summary measures ARLF and AUF. If we let y_{ijl} denote the delay obtained when rule i is used in conjunction with the lth and the jth levels of the two summary measures, then

$$y_{ijl} = u + \text{Rule}_i + \text{ARLF}_j + \text{AUF}_l + \epsilon_{ijl}.$$

In the above model, it is assumed that the errors (ϵ) are mutually independent and that each ϵ is drawn from the same continuous but not necessarily normal population.

We had two reasons for choosing a nonparametric testing procedure. First, the results of nonparametric tests were based on more general assumptions and hence they would also be true under the more restrictive assumption of normality. Second, to our knowledge there is no evidence that project delays are normally distributed in real-world scheduling problems.

The experimental model shown above consists of a three-way layout, but there is no known nonparametric statistic for testing such a model for treatment effects in its present form. However, if it can be reduced to a two-way layout, then Friedman's distribution-free S-statistic[6] for two-way layout can easily be applied. In our case a simple transformation did the job: in order to test for ARLF (i.e., $\mathrm{ARLF}_1 = \ldots \mathrm{ARLF}_7 = 0$ versus $\mathrm{ARLF}_1 \neq 0 \ldots \neq \mathrm{ARLF}_7 \neq 0$) a simple average over AUF was taken and to test for AUF a simple average over ARLF was taken. This produced the desired two-way layout and allowed use of nonparametric testing. Since variety was introduced into the data by varying the levels of the two summary measures, a test of significance of ARLF and AUF also indicated whether our goal with respect to variety was accomplished.

6. Results of the Experiment and the Tests of Verification of the Model

Tables 2 and 3 provide a summary of the findings of this study. They show that two new rules introduced by this research (SASP and MAXTWK) outperformed the other rules tested in a majority of the cases and produced lowest average project delay across all test cases.[7]

TABLE 2

The Best Eight Rules Based on Overall Results

Ranking Based on Average Delay[a]	
1st SASP	71.05
2nd MAXTWK,	74 39
3rd FCFS	87.60
4th MINSLK	87.90
5th LALP	88.60
6th MOF	90.56
7th MAXSLK	91.40
8th SOF	92.38
Ranking Based on Average Rank[b]	
1st SASP	1.81
2nd MAXTWK	2.87
3rd MINSLK	3.92
4th FCFS	4.27
5th MAXSLK	4.53
6th MOF	4.80
7th SOF	4.96
8th LALP	4.99
Ranking Based on Number of Times First[c]	
1st SASP	43
2nd MAXTWK	18
3rd MINSLK	13
4th MAXSLK	6
5th MOF	5
6th FCFS	3
7th SOF	1

[a] These figures were obtained by averaging total delay per problem over all values of ARLF and AUF (i.e., all problems).

[b] These figures were obtained by first ranking the performance of the rule under each value of ARLF and AUF and then by averaging over all problems.

[c] No other rule was ranked first.

[6] See Hollander and Wolfe [10, p. 140].

[7] In 52 out of 75 possible cases one of these two rules performed best.

TABLE 3

Categorization According to Both ARLF *and* AUF*

	ARLF						
AUF	−3	−2	−1	0	1	2	3
0.6	MINSLK[a]	MAXSLK	SASP	MINSLK	MAXSLK[d]	MINSLK	[e]
0.7	MINSLK	SASP	SASP	MINSLK	MAXTWK	MINSLK	[e]
0.8	MINSLK	MAXTWK	SASP	MINSLK	SASP	MINSLK	MINSLK[f]
0.9	SASP	MAXTWK[b]	SASP	MINSLK	SASP	MAXTWK	MINSLK
1.0	SASP	SASP	MAXTWK	MAXSLK	MAXTWK[d]	MOF	MINSLK[b]
1.1	SASP	SASP	MAXTWK[b]	MAXSLK	SASP	SASP	SASP
1.2	SASP	SASP	MAXTWK	SOF[c]	SASP	SASP	SASP
1.3	MAXSLK	MAXTWK	MAXTWK	SASP	SASP	MOF	SASP
1.4	SASP	SASP	MAXTWK	SASP	SASP	MAXTWK	SASP
1.5	SASP	SASP	MAXTWK	MAXTWK	SASP	SASP	SASP
1.6	MAXTWK	SASP	MAXTWK	SASP	SASP	MOF[b]	SASP

*This table contains the best performing rule under each value of ARLF and AUF, based on total average delay.

[a] Tied with MOF and MAXTWK.

[b] Tied with SASP.

[c] Tied with MAXSLK and SASP.

[d] Tied with FCFS.

[e] Omitted because zero delays were produced by this combination of AUF and ARLF.

[f] Tied with MOF, MAXTWK, FCFS, SASP.

An interesting conclusion from these results is that the "single project" approach of artificially combining projects; followed by some previous researchers and used in some commercially-available software packages, is apparently inferior to a "true" multi-project approach because of the two best performing rules SASP and MAXTWK were based on the multi-project approach.

Another interesting finding of this study, also seen in Tables 2 and 3, is that two rules which were reported as best performers in most single-project studies, SOF and LALP, performed very poorly. It may be noted that the schedule generated by the LALP rule will be very similar to the schedule generated by the MINSLK rule on the combined problem. The LALP rule will pick the activities from the longest project first, from the second longest project second and so forth. But this is usually how the MINSLK rule will behave on the "combined" version of the multi-project problem: the longest project determines the first critical path and its activities will be scheduled first; the second longest project will constitute the second critical path and so forth.

In verifying the experimental model employed, an assumption of lack of interaction between the ARLF and AUF factors was verified visually through profile plots. In Figure 2, a profile plot is constructed for each value of AUF by plotting average delays (i.e., averaged over all rules) over values of ARLF. As can be seen in Figure 2, average delays increase with the value of AUF, but plots stay parallel over ARLF, which confirms an hypothesis of lack of interaction between AUF and ARLF. Next, Friedman's test was applied to assess the significance of the ARLF. Using average delays, the value of the S-statistic was found to be significant at the $\alpha = 0.10\%$ level.[8] Using average ranks, the value of the S-statistic was found to be significant at the $\alpha = 10\%$ level. Thus there is at least 90% confidence that the ARLF effect is

[8] To account for the different number of projects, total delays were first divided by the number of projects in the problem.

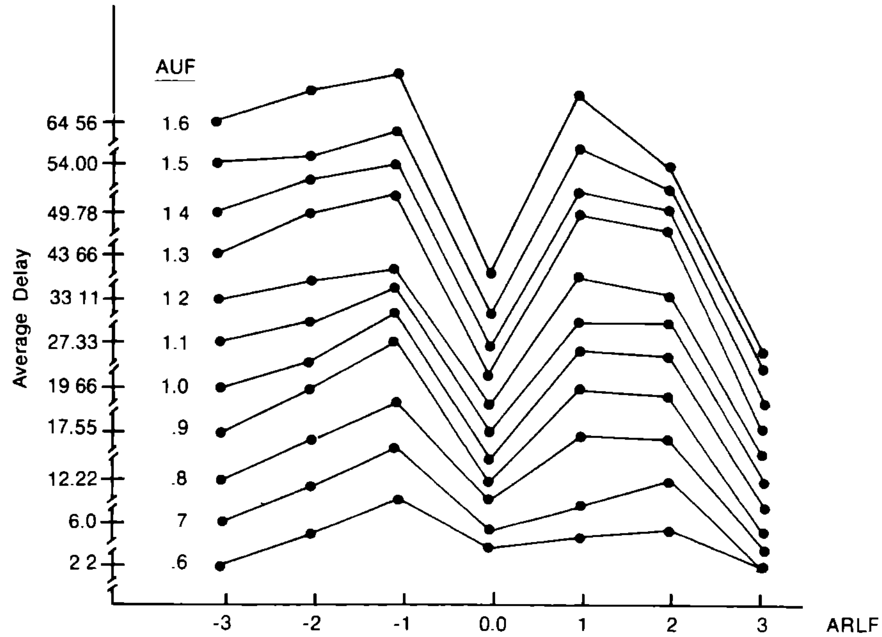

FIGURE 2. Plots of Profiles between AUF and ARLF.

significantly different from zero. Similarly, using delays to test for AUF, the S-statistic was found to be significant at the $\alpha = 0.1\%$ level. Using ranks the value of the S-statistic was significant at $\alpha = 2.5\%$ level. Thus there is at least 97.5% confidence that the AUF effect is also significantly different from zero.

7. Categorization of the Results

The performance of the rules was categorized according to values of the summary measures taken separately. This meant that, in case of ARLF, the performance of any pair of rules was tested over all levels of AUF (and vice versa).[9] Categorization proceeded by first testing the performance of the best rule against the second best rule. If the former performed significantly better, the testing procedure was discontinued and the first rule was judged the best rule under the value of the summary measure. On the other hand, if the test did not reveal significant results, then the two rules were put into the same "decision set"[10] and tested against the third best rule. Testing continued until either significance of difference in performance was established or the list of rules was exhausted. For ARLF, the order of the tests (WSRT) conducted for this purpose, was first based on the number of times a rule was ranked first, then on total average delay, and finally on average rank (see Table 4). As can be seen in Table 4, in most cases the rankings were the same and few additional tests had to be made (the results are summarized in Table 5).

In the case of AUF, the WSRT did not produce significant results due to small sample sizes (i.e., ARLF had only seven levels and some observations had to be thrown away due to ties). However, the categorization based on AUF provided us with

[9] For pairwise testing of the rules, Wilcoxon Signed Ranks Test (WSRT) was used and the test was based on total delays. See Hollander and Wolfe [10, p. 27].

[10] By definition, a decision set contains those rules whose performance is not significantly different under a value of the summary measure.

TABLE 4

Summary of Performance of the Rules under ARLF *Categorization*

	Rank	Average Total Delay	Total Average Rank	Number of Times First
ARLF = −3.0	1st	SASP	SASP	SASP
	2nd	MAXTWK	MAXSLK	MINSLK
	3rd	MAXSLK	MAXTWK	MAXTWK
	4th	MINTWK	MINTWK	MAXSLK[a]
ARLF = −2.0	1st	SASP	SASP	SASP
	2nd	MAXTWK	MAXTWK	MAXTWK
	3rd	LALP	LALP	MAXSLK
	4th	MINTWK	MINSLK	*
ARLF = −1.0	1st	MAXTWK	MAXTWK	MAXTWK
	2nd	SASP	SASP	SASP
	3rd	LALP	LALP	*
	4th	FCFS	FCFS	
ARLF = −0.0	1st	MAXTWK	MAXTWK	MINSLK[b]
	2nd	SASP	SASP	MAXSLK
	3rd	MAXSLK	MAXSLK	SOF[c]
	4th	MINSLK	MINSLK	*
ARLF = 1.0	1st	SASP	SASP	SASP
	2nd	FCFS	FCFS	FCFS[c]
	3rd	MAXTWK	MAXTWK	MINSLK[d]
	4th	LALP	SOF	*
ARLF = 2.0	1st	MOF	MOF	SASP
	2nd	SASP	SASP	MINSLK[a]
	3rd	MAXTWK	MINSLK	MAXTWK
	4th	MINSLK	MAXTWK	*
ARLF = 3.0	1st	SASP	SASP	SASP
	2nd	MINSLK	MINSLK	MINSLK
	3rd	FCFS	FCFS	FCFS
	4th	MINTWK	MINTWK	MOF[c]

* Blank spaces indicate no rule qualified for this position.
[a] Tied with MOF.
[b] Tied with SASP.
[c] Tied with MAXTWK.
[d] Tied with MAXSLK.

a basis for comparing our work with some previous research. As can be seen in Table 6, SASP is consistently the best rule for AUF = 0.9–1.6. But for AUF = 0.6 and 0.7, MINSLK outperforms SASP. It may be noted that the AUF values of 0.6 and 0.7 produce a not-very-constraining set of resource restrictions, and one common approach of setting resource availabilities as a percentage of peak resource requirements (i.e., Conway [1], 70, 80, 90%; Gordon [9], 50 and 70%) also has the same effect. Thus we tested the performance of four of the rules (i.e., SOF, MINSLK, SASP, MAXTWK) on these less constraining values of UF (i.e., 0.6–0.9). Again, the SASP was first, and with an average rank of 1.75, but this time MINSLK was second (1.89) and MAXTWK was third (2.21) and SOF was fourth (2.79). Thus, the performance of MINSLK improved as resource constraints became *less* constraining, which is an interesting observation about this popular rule.

TABLE 5

The Rules of the Tests of Significance Based on ARLF *Categorization*

Level	Rule
$-2.5 < \text{ARLF} < -3.5$	SASP, better than MINSLK, MAXSLK and MAXTWK at $\alpha = 0.027$, 0.001, and 0.009, respectively.
$-1.5 < \text{ARLF} < -2.5$	SASP, better than MAXTWK at $\alpha = 0.019$.
$-0.5 < \text{ARLF} < -1.5$	MAXTWK, better than SASP at $\alpha = 0.042$.
$-0.5 < \text{ARLF} < 0.5$	There is no best rule. MAXTWK, SASP, MINSLK, MAXSLK, SOF performed insignificantly different in that order.
$0.5 < \text{ARLF} < 1.5$	SASP, better than FCFS and MAXTWK at $\alpha = 0.012$ and 0.0005, respectively.
$1.5 < \text{ARLF} < 2.5$	MOF, better than MINSLK and MAXTWK at $\alpha = 0.021$ and 0.036, respectively.[a]
$2.5 < \text{ARLF} < 3.5$	SASP, better than MINSLK at $\alpha = 0.025$.

[a]SASP also performed significantly better than MINSLK, but failed to do so over MAXTWK.

TABLE 6

The Two Best Rules under AUF *Categorization*[a]

AUF	Rank	Based on Delay	Based on Rank	Based on Number First
0.6	1.	MINSLK	MINSLK	MINSLK
	2.	FCFS	FCFS	MAXSLK
0.7	1.	MINSLK	MINSLK	MINSLK
	2.	SASP	SASP	SASP
0.8	1.	SASP	SASP	MINSLK
	2.	MINSLK	MINSLK	SASP
0.9	1.	SASP	SASP	SASP
	2.	MAXTWK	MINSLK	MINSLK
1.0	1.	MAXTWK	SASP	SASP
	2.	SASP	MAXTWK	MAXTWK
1.1	1.	SASP	SASP	SASP
	2.	MAXTWK	MAXTWK	MAXTWK

[a]For AUF = 1.2–1.6 the rankings were SASP first and MAXTWK second, for all cases.

8. Test on Patterson's Data

The last comparison of the results was made using Patterson's data [16][11] which consists of thirty 34-project problems, each problem containing a total of 610 activities and 13 different resource types. For thirty of these problems, ARLF varied between

[11]We are grateful to Professor Patterson for lending us his data for this research.

0.01 and 0.6, which corresponds generally to our fourth level of ARLF. For the 13 different resource types, the Average Utilization Factor varied between 0.1 and 0.96 (which corresponds to generally less constraining values of AUF). We tested the performance of our top six rules on this data, and found that SASP again performed best, with its performance being better than that of SOF (one of Patterson's two best-performing rules) at the $\alpha = 0.001$ level of significance (using the WSRT). MINSLK performed third and MAXTWK was fourth. We speculate that the superior performance of MINSLK over MAXTWK here may be due to the generally less constraining values of AUF which characterizes Patterson's problems.

9. Suggested Implementation Procedure

The results of this research can be implemented in a number of ways. If the most important variable to management is the occurrence and the timing of "peaks" in resource requirements (such as cash), then the choice of rule should be based on Table 5. According to Table 5, if ARLF is between -3.5 and -1.5, or equivalently, if the peak of the total requirements occurs on or before the 29% of the longest project's expected completion time, then SASP should be used.[12] If it is between 29 to 43% ($0.43 = 3/7$) of the projects' expected completion time then MAXTWK should be used, and so forth. On the other hand, if management is primarily concerned with the tightness of resources, then either one primary resource (e.g., labor) can be picked and decisions made based on the tightness of this resource (i.e., MINSLK if AUF = 0.6–0.9 and SASP otherwise) or the average of the AUF values for all resources can be used. If both the location of the peak and the tightness of the resources are important, then we recommend that the choice be based on Table 7.

TABLE 7

Suggested "One Best Rule" for Various ARLF *and* AUF *Categories*

ARLF Range	AUF Range 0.6 to 0.8	AUF Range 0.9 to 1.6
− 3.5 to −2.5	MINSLK	SASP
− 2.5 to − 1.5	MAXTWK	SASP
− 1.5 to −0.5	SASP	MAXTWK
− 0.5 to 0.5	MINSLK	SASP, SOF, MAXTWK[a]
0.5 to 1.5	SASP	SASP
1.5 to 2.5	MINSLK	MOF, SASP
2.5 to 3.5	MINSLK	SASP

[a] More than one rule is recommended. The choice depends on secondary criterion.

10. Summary and Suggests for Further Research

This research provides one possible scheme for categorizing the performance of scheduling rules in a multi-project environment. It also shows that rules based on a true multi-project approach perform better than rules based on the single-project approach of artificially combining projects. Future researchers may wish to replicate the cell values (i.e., ARLF/AUF combinations) used in this study and to test the strength of our ARLF/AUF classification. However in this process they will also have to develop their own nonparametric statistical methodology. In general the research

[12] Twenty-nine percent is obtained in the following manner: the total range of ARLF is equal to 7 (i.e., 3.5–(−3.5)). The range over which SASP is successful initially is 2 (i.e., 3.5–1.5), and the ratio 2/7 is 29%.

topic of multi-project scheduling is one which has been surprisingly neglected in recent years—a situation we hope will not continue.

References

1. Conway, R. W., "Priority Dispatching and Job Lateness in a Job Shop," *J. Indust. Engineering*, Vol. 16, No. 4 (1965), pp. 228–237.
2. Davis, E. W., "An Exact Algorithm for the Multiple Constrained-Resource Project Scheduling Problem," Unpublished Ph.D. Thesis, Yale University, 1969.
3. ———, "Resource Allocation in Project Network Models—A Survey," *J. Indust. Engineering*, Vol. 17, No. 2 (1966), pp. 177–188.
4. ———, "Project Scheduling under Resource Constraints. Historical Review and Categorization of Procedures," *AIIE Trans.*, Vol. 5, No. 4 (1973), pp. 297–313.
5. ———, "Project Network Summary Measures Constrained-Resource Scheduling," *AIIE*, Vol. 7, No. 2 (1975), pp. 132–142.
6. ——— and Patterson, J. H., "A Comparison of Heuristic and Optimum Solutions in Resource Constrained Project Scheduling," *Management Sci.*, Vol. 21, No. 8 (1975), pp. 944–955.
7. Elmaghraby, S. E., *Activity Networks*, Wiley, New York, 1977.
8. Fendley, L. G., "Toward the Development of a Complete Multiproject Scheduling System," *J. Indust. Engineering*, Vol. 19, No. 10 (1968), pp. 505–515.
9. Gordon, J. H., "Project Network Categorization in an Evaluation of Heuristic Resource Allocation Techniques," Unpublished Ph.D. Thesis, University of Birmingham, 1976.
10. Hollander, M. and Wolfe, D., *Nonparametric Statistical Methods*, Wiley, New York, 1973.
11. Jenett, E., "Availability of CPM Programs," *Project Management Quart.*, Vol. 1, No. 2 (1970), pp. 10–13.
12. Johnson, T. J. R., "An Algorithm for the Resource-Constrained, Project Scheduling Problem," Unpublished Ph.D. Thesis, Massachusetts Institute of Technology, 1967.
13. Kurtulus, I. S., "An Analysis of Scheduling Rules for Multi-Project Scheduling," Unpublished Ph.D. Thesis, University of North Carolina, 1978.
14. Mize, H. H., "A Heuristic Scheduling Model for Multi-Project Organizations," Unpublished Ph.D. Thesis, Purdue University, August 1964.
15. Pascoe, T. L., "An Experimental Comparison of Heuristic Methods for Allocating Resources," Unpublished Ph.D. Thesis, Cambridge University, 1965.
16. Patterson, J. H., "Alternative Methods of Project Scheduling with Limited Resources," *Naval Res. Logist. Quart.*, Vol. 20, No. 4 (1973), pp. 767–784.
17. ———, "Project Scheduling: The Effects of Problem Structure on Heuristic Performance," *Naval Res. Logist. Quart.* (1976).
18. ——— and Huber, W. D., "A Horizon-Varying, Zero-One Approach to Project Scheduling," *Management Sci.*, Vol. 20, No. 6 (1974), pp. 990–998.
19. Stinson, Joel P., "A Branch and Bound Algorithm for a General Class of Resource-Constrained Scheduling Problems," *AIIE Conf. Procs.*, Las Vegas, Nevada (1975), pp. 337–342.
20. Talbot, Brian F., "An Integer Programming Algorithm for the Resource-Constrained Project Scheduling Problem," Unpublished Ph.D. Thesis, Pennsylvania State University, 1976.
21. Wiest, J. D., "The Scheduling of Large Projects with Limited Resources," Ph.D. Thesis, Carnegie Institute of Technology, May 1963.