Comments Welcome

Resource-Constrained Multi-Project Scheduling: Priority Rule Performance Revisited

Tyson R. Browning*
Neeley School of Business
Texas Christian University
TCU Box 298530
Fort Worth, TX 76129
t.browning@tcu.edu

Ali A. Yassine
Department of Industrial & Enterprise
Systems Engineering (IESE)
University of Illinois at Urbana-Champaign
Urbana, IL 61801
yassine@uiuc.edu

Texas Christian University
Neeley School of Business
Working Paper

Current version: April 2, 2008

*Corresponding author

The authors are grateful to Sönke Hartmann for helpful comments on the paper, Ibrahim Kurtulus for discussions on multi-project scheduling, and Terry Dielman for additional statistical expertise. Any errors in this paper are our own. The first author is grateful for support from the Neeley Summer Research Award Program from the Neeley School of Business at Texas Christian University.

Resource-Constrained Multi-Project Scheduling: Priority Rule Performance Revisited

Abstract

Managers of multiple projects with limited resources face difficult decisions. How should they allocate resources to minimize the average delay per project, or to minimize the time to complete the whole set of projects? Such is the basic *resource-constrained multi-project scheduling problem* (RCMPSP), which is rather intractable and has attracted much research on optimization and meta-heuristic techniques. Nevertheless, decision heuristics using *priority rules* remain crucial for several reasons, especially when project managers do not (or cannot) build a complete network model of their project. Unfortunately, past research has reported conflicting results on priority rule performance and has not provided managers with clear guidance on which rule to use in various situations. This paper addresses the static RCMPSP with two lateness objectives, project lateness and portfolio lateness. We developed new and improved measures for RCMPSP characteristics and conducted a comprehensive analysis of 20 popular priority rules on 12,320 test problems. We considered the performance of these rules subject to various project-, activity-, and resource-related characteristics—including network complexity and resource distribution and contention. We found several situations in which widely-advocated decision rules perform poorly. We also confirmed that portfolio managers and project managers will prefer different decision rules depending on their local or global objectives. We summarize our results in two decision tables, the practical use of which requires managers to do only a rough, qualitative characterization of their projects in terms of complexity, degree of resource contention, and resource distribution.

Keywords: *Project Management; Multi-Project Scheduling; Resource Constraints; Heuristic Priority Rules*

1. Introduction

As projects have become ever-more-common structures for organizing work in contemporary enterprises, issues involving the simultaneous management of multiple projects (or a portfolio of projects) have become more pervasive and acute. For example, studies have shown that managers typically deal with up to four projects at once (Liberatore and Pollack-Johnson 2003; Maroto *et al.* 1999). In this paper, we address the case of a portfolio of concurrent projects with identical start times. Each project consists of an activity network that draws from common pools of multiple types of resources which are typically not large enough for all of the activities to work concurrently. In such cases, which activities and projects should get priority? The goal is to prioritize them so as to optimize an objective function, such as minimizing the delay to each project, or to the overall portfolio. Such is the basic *resource-constrained multi-project scheduling problem* (RCMPSP), a common decision problem faced by managers. According to Payne (1995), up to 90% of the value of all projects accrues in the multi-project context, so the impact of even a small improvement in their management could provide an enormous benefit. A significant portion of this improvement could come from making better resource allocation decisions.

Most research on resource-constrained project scheduling has dealt with single projects—the *resource-constrained project scheduling problem* (RCPSP). When dealing with multiple projects, two approaches have been used: (1) a single-project approach, using dummy activities and precedence arcs to combine the projects into a single mega-project, thereby reducing the RCMPSP to a RCPSP with a single critical path, or (2) a multi-project (MP) approach, maintaining the RCMPSP and a separate critical path per project (Kurtulus and Davis 1982—hereafter, K&D 1982). In this paper, we take the second approach, because (1) it is more common in practice and realistic, (2) it has received less attention in past research, (3) it presents a greater opportunity for improvement (Herroelen 2005), and, critically, (4) the decision guidance for managers is inconclusive.

Both the RCPSP and the RCMPSP are strongly NP-hard, meaning there are no known algorithms for finding optimal solutions in polynomial time (Lenstra and Kan 1978). Hence, most research has sought efficient heuristics and meta-heuristics. While meta-heuristics have been shown to outperform *priority rule* heuristics, the latter continue to be crucial for several reasons: (1) meta-heuristics' improved performance comes at greater computational expense; (2) priority rules are a component of other (local search-based and sampling) heuristics (Kolisch 1996b) and some meta-heuristics; (3) priority rules are used extensively by commercial project scheduling software due to their speed and simplicity (Herroelen 2005); and (4) priority rules are the primary tool for very large problems (Kolisch 1996a). However, perhaps the most important argument for priority rules is that *they are (and will, for quite some time, most likely continue to be) the decision support tool most likely to be used in practice*. For a variety of reasons, most project managers do not (or cannot) actually build the formal activity network models to which meta-heuristics are applied. When faced with a resource allocation decision, project managers will often make a quick call based on intuition or simple rules of thumb. Therefore, the question of which priority rule to use is discussed in many project management textbooks (e.g., Meredith and Mantel 2006), but without conclusive guidance for project managers. This is because few comprehensive, systematic studies have been reported in the literature (Herroelen 2005), and these few studies have dealt with relatively small subsets of the common priority rules. Moreover, these studies have presented conflicting results on priority rule performance, because this varies based on portfolio, project, resource, and activity characteristics. Thus, while we were not sanguine at first about the value of another study of priority rules, numerous recent experiences with and observations of real projects led us to realize that a more comprehensive study of a larger set of rules and *RCMPSP characteristics*—that will give managerial decision makers firmer guidance—is of great value and much needed.

In this paper, we address the static RCMPSP with two lateness objectives, making several contributions. First, we describe the salient characteristics of the RCMPSP in terms of its constituent projects, activities, and resources, and we introduce five new or modified measures of RCMPSP characteristics and use these to define a multi-dimensional problem space. Second, using a full factorial experiment with 12,320 randomly generated problem instances, we demonstrate the superiority of the new measures and analyze the performance of 20 priority rules. We find significant differences in the performance of the rules—implying that the choice of rule does indeed matter—and that several widely-advocated rules generally do not perform well. Third, we organize these results for managers, distinguishing the project and portfolio management perspectives.

The rest of the paper is organized as follows. After showing the mathematical formulation of the basic RCMPSP in the next section, Section 3 reviews related literature. In §4 we discuss characteristics of the RCMPSP, including objective functions, network characteristics, and resource characteristics. §5 provides our scheduling algorithm, and §6 reports our efforts to replicate the results of K&D (1982), the salient prior study in this area. These discussions motivate our more comprehensive study, which we detail in §7. §8 distills the study's implications for managers, and §9 concludes the paper.

2. Basic Problem Statement

The static RCMPSP can be stated as follows. A set of $l = 2,\ldots,L$ projects are to be performed. Each project consists of $i = 1,\ldots,N_l$ activities with deterministic, non-preemptable duration d_{il}. The activities are interrelated by predecessor and resource constraints. Predecessor constraints keep activity i from starting until all of its predecessors have finished. Each activity requires r_{ik} units of resource type $k \in K$ during every period of its duration. Resource k has a renewable capacity of R_k. At any time, if the set of eligible (precedence unconstrained) activities requires more than R_k for any k, then some activities will be delayed. The RCMPSP entails finding a schedule for the activities (i.e., determining the start or finish times) that optimizes a performance measure, such as minimizing the average delay in all projects. Each project is associated with a due date, set by its resource-unconstrained duration, which is used to measure delays. Let F_{il} represent the finish time of activity i in project l, such that a schedule can be represented by a vector of finish times (F_{1l}, ..., F_{il}, ..., $F_{N_l l}$). Let $A(t)$ be the set of activities in work at time instant t. P_{il} is the set of all immediate predecessors of activity i in project l. $\hat{i} \in P_{il}$. With these definitions, the problem can be formally stated as:

$$\text{Optimize: Performance Measure } (\forall i \in N_l, l \in L\text{: } F_{1l}, ..., F_{il}, ..., F_{N_l l}) \quad (1)$$

$$\text{Subject to:} \quad \forall i \in N_l, \hat{i} \in P_{il}, l \in L: \quad F_{\hat{i}l} \leq F_{il} - d_{il} \quad (2)$$

$$\forall i, l \in A(t): \quad \sum_{i,l \in A(t)} r_{ilk} \leq R_k \quad k \in K; t \geq 0 \quad (3)$$

$$\forall i \in N_l, l \in L: \quad F_{il} \geq 0 \quad (4)$$

The objective function (1) seeks to optimize a pre-specified performance measure. Constraints (2) impose the precedence relations between activities; constraints (3) limit the resource demand imposed by the activities being processed at time t to the capacity available; and constraints (4) force the finish times to be non-negative.

The basic (static) problem can be expanded in several ways. New projects might arrive at various rates (the dynamic problem).[1] Project interdependencies (beyond common resources) might exist. Activities could be performed in various modes, each requiring different types and/or amount of resources (e.g., Tseng 2004). Activity preemption might be allowed, perhaps implying switching or restart costs (e.g., Ash 2002). Activity durations could be stochastic. Resource transfer times could be non-zero, and resources might be non-renewable. To maximize our insights from the basic RCMPSP, we do not address these additional features in this paper, although our approach could be extended to do so.

[1] In a static RCMPSP environment (e.g., K&D 1982; Lawrence and Morton 1993; Lova and Tormos 2001; Pritsker *et al.* 1969), all projects within the portfolio and their associated activities are known prior to scheduling, unlike in the dynamic case (e.g., Bock and Patterson 1990; Dumond and Mabert 1988; Kim and Leachman 1993; Yang and Sum 1993; Yang and Sum 1997).

3. Literature Review

This paper addresses the static RCMPSP and maintains the distinction between projects (the MP approach mentioned in §1). While an abundant amount of research and progress has been reported in the literature for the (single-project) RCPSP (several reviews are available—see, e.g., Brucker *et al.* 1999; Herroelen 2005; Kolisch and Hartmann 2005), the single-project approach to solving RCMPSPs has several drawbacks (K&D 1982; Chiu and Tsai 1993). First, it is a less accurate representation of reality, as it implicitly assumes equal delay penalties for all projects (Kurtulus 1985). Second, independent project analysis becomes difficult when all projects are bound together—e.g., it is hard to reveal the degree of concurrency among different projects and to maintain the distinction in their critical paths. In many realistic situations, each project has its own manager who is interested in an individual project's performance characteristics. Third, optimal procedures for RCPSPs cannot solve large problem instances, which are the typical result of aggregating multiple projects.[2]

Using the MP approach, two general approaches are exact methods and heuristic procedures. Exact methods (e.g., Chen 1994; Deckro *et al.* 1991; Pritsker *et al.* 1969; Vercellis 1994) are limited to solving small problem instances and impractical for solving large RCMPSPs (Herroelen 2005; Özdamar and Ulusoy 1995). On the other hand, heuristic procedures can be divided into four groups: priority-rule-based X-pass heuristics, classical metaheuristics, non-standard metaheuristics, and miscellaneous heuristics (Kolisch and Hartmann 1999; Kolisch and Hartmann 2005). Classical metaheuristics include simulated annealing (e.g., Bouleimen and Lecocq 2000), genetic algorithms (GAs) (e.g., Gonçalves *et al.* 2004; Kim *et al.* 2005; Kumanan *et al.* 2006), and swarm optimization (e.g., Linyi and Yan 2007). Non-standard metaheuristics include agent-based and non-GA population-based approaches (e.g., Confessore *et al.* 2007; Homberger 2007). Miscellaneous heuristics include forward-backward improvement and others (e.g., Lova and Tormos 2002). While these latter three categories of heuristics can outperform simple priority rules, good priority rules remain very important for the RCMPSP for the reasons stated in the Introduction.

Priority-rule-based heuristics, also known as X-pass methods, include single- and multi-pass methods (Hartmann and Kolisch 2000). Single-pass priority rules prioritize the activity that maximizes (or minimizes) a particular value. Multi-pass methods include *multi-priority rules*, which employ more than one priority rule in succession (e.g., Lova and Tormos 2001), and *sampling methods*, which generally make use of a single rule with

[2] Examples of optimal RCPSP procedures include dynamic programming (e.g., Elmaghraby 1993), 0-1 programming (e.g., Patterson and Roth 1976; Pritsker *et al.* 1969), and branch & bound (e.g., Demeulemeester and Herroelen 1992; Demeulemeester and Herroelen 1997; Mingozzi *et al.* 1998; Sprecher 2000).

the insertion of some degree of randomness (Hartmann and Kolisch 2000).

Priority rules can also be classified on the basis of the information they use: (a) activity-related, (b) project-related, and (c) resource-related (Kolisch 1996a).[3] Activity-related rules assign high priority to an activity based on a parameter or characteristic of the activity itself, such as its duration (e.g., shortest operation first—SOF) or slack (e.g., minimum slack first—MINSLK). Project-related rules assign priorities to activities based on the project they belong to, or characteristics of that project (e.g., shortest activity from shortest project first—SASP). Resource-related rules assign priority in terms of an activity and/or project's resource demands, scarcity of resources used, or some combination. High priorities are usually assigned to potential bottleneck activities. An example is the maximum total work content (MAXTWK) rule. Some rules combine elements of information about the activity, the project, and/or the resources (Hartmann and Kolisch 2000). For each priority rule addressed in our study, we note its "Basis" according to this classification in Tables 1 and 5.

Davis and Patterson (1975) noted that successful priority rules generally incorporate some measure of *time* or *resource usage*; they also isolated three important characteristics in the RCPSP: an activity's resource utilization, the ratio of average slack per activity to the critical path length, and project complexity. Interestingly, project size (number of activities) has *not* been found to be a significant determinant of priority rule performance (e.g., Pascoe 1966). Ulusoy and Özdamar (1989) noted the importance of the following measures in deciding upon successful priority rules: percentage of critical activities, network complexity and resource measures, obstruction value, and utilization factor.

Although uncounted studies have been conducted on the performance of a myriad of priority rules for the RCPSP, only a relative handful of rules have been developed for and studied in a MP environment (Herroelen 2005). It is important to note that the single- and MP approaches often produce different schedules with the same priority rule (Kurtulus 1978; Lova and Tormos 2001), especially if the rule depends on the critical path—e.g., the MINSLK rule. While the single-project approach is more efficient for minimizing a single project's duration, priority rules based on the MP approach perform better when minimizing the average delay in several projects (K&D 1982). RCMPSP studies have disagreed about which rule performs best and under which conditions, although MINSLK has generally performed well (Cohen *et al.* 2004; Davis and Patterson 1975; Fendley 1968). K&D (1982) developed six priority rules for the MP environment, and along with three single-

[3] These classes are neither exclusive nor exhaustive. This taxonomy is merely one of many possible characterizations. For instance, considering whether a priority rule returns the same value regardless of the stage it is performed in, we may characterize this rule as static, compared to a dynamic rule that changes value depending on the stage (Kolisch 1996a).

project priority rules (see the list in Table 1), analyzed these with the objective of minimizing total project delay, finding that SASP was best under most conditions. We discuss their study in detail in our pilot study in §6, but before doing so we must discuss some of the important RCMPSP characteristics.

Table 1: Priority rules analyzed by K&D (1982)

Priority Rule (* = multi-project)	**Basis**	**Formula**	**Original Tie-Breaker**	**Comments**
1. **FCFS**—First Come First Served	Activity	Min(ES_{il}), where ES_{il} is the early start time of the i^{th} activity from the l^{th} project	Random	Best in studies by Mize (1964) Dumond and Mabert (1988)[b] and Bock and Patterson (1990)[b]
2. **SOF**—Shortest Operation First	Activity	Min(d_{il}), where d_{il} is the duration of the i^{th} activity from the l^{th} project	FCFS	Best in study by Patterson (1973)
3. **MOF**—Maximum (longest) Operation First	Activity	Max(d_{il})	Greatest resource requirements (*GRES*), where $GRES = \text{Max}\left(\sum_{k=1}^{K} r_{ilk}\right)$	
4. **MINSLK***—Minimum Slack	Activity	Min(SLK_{il}), where $SLK_{il} = LS_{il} -$ Max(ES_{il}, t), LS_{il} is the late start time of the i^{th} activity from the l^{th} project, and t is the current time step[4]	FCFS	Best in studies by Fendley (1968), Davis and Patterson (1975)[a], and Boctor (1990)[a]
5. **MAXSLK***—Maximum Slack	Activity	Max(SLK_{il})	*GRES*	
6. **SASP***—Shortest Activity from Shortest Project	Activity, Project	Min(f_{il}), where $f_{il} = CP_l + d_{il}$ and CP_l is the critical path duration of the l^{th} project without resource constraints	FCFS	Best in studies by K&D (1982), Kurtulus (1985), Kurtulus and Narula (1985), Tsubakitani and Deckro (1990), and Maroto *et al.* (1999)
7. **LALP***—Longest Activity from Longest Project	Activity, Project	Max(f_{il})	*GRES*	
8. **MINTWK***—Minimum Total Work content	Activity, Resource	$\text{Min}\left(\sum_{k=1}^{K}\sum_{i \in AS_l} d_{il} r_{ilk} + d_{il}\sum_{k=1}^{K} r_{ilk}\right)$, where AS_l is the set of activities already scheduled (i.e., in work) in project l	FCFS	
9. **MAXTWK***—Maximum Total Work content	Activity, Resource	$\text{Max}\left(\sum_{k=1}^{K}\sum_{i \in AS_l} d_{il} r_{ilk} + d_{il}\sum_{k=1}^{K} r_{ilk}\right)$	FCFS	Best in studies by Kurtulus (1985), Kurtulus and Narula (1985), and Lova and Tormos (2001)

[a]Discusses a *single* project (RCPSP) [b]Refers to the *dynamic* RCMPSP

In summary, while various studies have identified potentially important characteristics of the RCMPSP and proposed various priority rules, the variety of results and their disagreements have left project managers lacking clear guidance on which rule to use in a particular situation.

4. RCMPSP Characteristics

In this section we discuss four important characteristics of the RCMPSP—objective function, network complexity, resource distribution, and resource contention—that distinguish problem and project situations. We

[4] t is relevant when using the parallel schedule generation scheme (SGS), where an activity's slack will diminish the longer it is delayed. We discuss the SGS in §5.

also discuss an appropriate measure for each, including our proposals for some new measures.

4.1 Objective Function

A variety of objective functions have been used for the RCMPSP. The most widely used has been the minimization of project duration (Baker 1974). Other MP objective functions include: minimize total project delay, lateness, or tardiness (K&D 1982), minimize average project delay (Lova and Tormos 2001), minimize total lateness or lateness penalty (Kurtulus 1985), minimize overall project cost (Talbot 1982), minimize the cost of delay (Kurtulus 1978; Kurtulus and Narula 1985), or maximize resource leveling (Woodworth and Willie 1975).[5] Several studies have shown that priority-rule performance depends on the chosen objective (e.g., K&D 1982; Kurtulus 1985). In this study, we seek to minimize project or portfolio delay (tardiness). We do this by defining a due date for each project, based on the length of its resource-*unconstrained* critical path (CP), and then measuring the delay beyond that point. Project and problem delay can be measured in at least five ways, as defined in the following equations based on the three-project example problem in Figure 1:

$$\text{Total Delay} = a + b + c \qquad \text{(R1)}$$

$$\text{Average Delay} = (a + b + c) / 3 \qquad \text{(R2)}$$

$$\text{Average Percent Delay} = \frac{\frac{a}{A} + \frac{b}{B} + \frac{c}{C}}{3} \qquad \text{(R3)}$$

$$\text{Total Delay} = \text{Max}(A + a, B + b, C + c) - \text{Max}(A, B, C) \qquad \text{(R4)}$$

$$\text{Percent Delay} = \frac{\text{Max}(A + a, B + b, C + c) - \text{Max}(A, B, C)}{\text{Max}(A, B, C)} \qquad \text{(R5)}$$

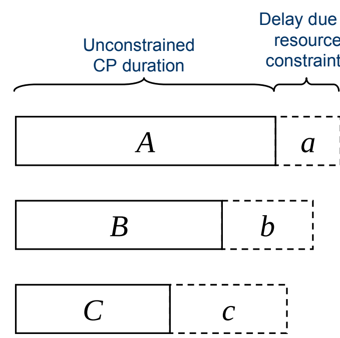


Figure 1: Example problem composed of three projects

The first three of these measures are project measures; the latter two are problem (portfolio) measures. R1 and R2 are essentially equivalent as discriminating measures. We focus on R3 and R5, since taking delay as a percentage of duration allows comparison of projects and problems with different durations. For example, R3 recognizes that a ten-day delay on a one-day project is probably worse than a ten-day delay on a 100-day project. Furthermore, R3 and R5 represent the individual project manager's and the program or portfolio manager's respective points of view. That is, while a project manager will care about the effects of delays on his or her individual project, a program manager might choose to focus on delays to the entire portfolio of projects. R5 is a less sensitive measure than R3, because in most cases it is affected only by delays to the longest project in a problem.

[5] Other single-project objective functions include: maximize resource utilization (Neumann and Zimmermann 1999), minimize cost of resource requirements (Möhring 1984), and maximize net present value (Doersch and Patterson 1977; Elmaghraby and Herroelen 1990).

4.2 Network Complexity

Network complexity is an important factor in project scheduling, because low-complexity networks are less precedence-constrained. Many measures of network complexity have been proposed since the coefficient of network complexity (CNC) was first introduced by Pascoe (1966). The CNC was simply defined as the ratio of the number of arcs (precedence relationships), *A*, over the number of nodes (activities), *N*. Elmaghraby and Horroelen (1980) questioned the CNC's usefulness for truly reflecting the complexity of project networks. A review of ten complexity measures (Browning and Yassine 2008) led us to use an adapted form of the average number of non-redundant arcs per node, which has been used extensively in the RCPSP literature (Kolisch *et al.* 1995) and in the few computational studies performed on the RCMPSP (Kurtulus and Narula 1985, Lova & Tormos 2001).

To adapt this complexity measure, we normalize it over [0,1]:

$$C = \frac{A' - A'_{\min}}{A'_{\max} - A'_{\min}} \tag{5}$$

where A' is the number of non-redundant arcs, $A'_{\min}$ is the minimum number of non-redundant arcs in a network of *N* nodes ($A'_{\min} = N - 1$), and $A'_{\max}$ is the maximum number of non-redundant arcs ($A'_{\max} = \frac{N^2}{4}$) (Browning and Yassine 2008).[6] In terms of *N*, *C* then becomes:

$$C = \frac{A' - (N-1)}{\left(\frac{N^2}{4}\right) - (N-1)} = \frac{4A' - 4N + 4}{(N-2)^2} \tag{6}$$

It is important to note that the level of analysis in the network complexity literature is a single project. In the MP environment, we hesitate to use a simple averaging or some other composite complexity measure, because the impact of individual network complexity on the MP portfolio remains unclear. A problem with three medium-complexity projects is probably not the same as a problem with one high- and two low-complexity projects, even though both might average to the same number. Therefore, we maintain a distinction between projects and use a vector of constituent project complexities, $\boldsymbol{C} = \{C_1, C_2, \ldots, C_L\}$, as a MP complexity measure.

4.3 Resource Distribution

Several measures of the availability and distribution of project resources have been developed for the RCPSP. Two early ones are the resource factor, *RF*, which indicates the average number of resources used by an

[6] We assume that a dummy start node is fully connected to all activities (in the individual project) without a predecessor and that a dummy finish node is fully connected to all activities without a successor, but we do *not* include these dummy nodes or arcs in *N* or *A*.

activity (Cooper 1976; Kolisch *et al.* 1995; Pascoe 1966), and the resource strength, *RS*, which expresses the relationship between resource requirements and resource availability (Cooper 1976; Kolisch *et al.* 1995). However, K&D (1982) noted that these measures are not as useful in a MP environment and proposed an alternative measure for the RCMPSP, the *average resource loading factor*, *ARLF*, which identifies whether the bulk of a project's total resource requirements are in the front or back half of its critical path duration[7] and the relative size of the disparity. For project *l*, it is defined as:

$$ARLF_l = \frac{1}{CP_l} \sum_{t=1}^{CP_l} \sum_{k=1}^{K_{il}} \sum_{i=1}^{N_l} Z_{ilt} X_{ilt} \left(\frac{r_{ilk}}{K_{il}} \right) \tag{7}$$

where $Z_{ilt} = \begin{cases} -1 & t \le CP_l/2 \\ 1 & t > CP_l/2 \end{cases}$, $X_{ilt} = \begin{cases} 1 & \text{if activity } i \text{ of project } l \text{ is active at time } t \\ 0 & \text{otherwise} \end{cases}$, $Z_{ilt}X_{ilt} \in \{-1, 0, 1\}$, N_l is the number of activities in project *l*, K_{il} is the number of types of resources required by an activity *i* in project *l*, and r_{ilk} is the amount of resource type *k* required by task *i* in project *l*.[8] Projects with $ARLF < 0$ are "front-loaded" in their resource requirements, while projects with $ARLF > 0$ are "back-loaded." Since K&D (1982) proposed *ARLF*, we are not aware of any additional work to improve upon a measure of the resource distribution in the RCMPSP.

However, *ARLF* provides only a rough indicator of a project's resource distribution. It can fail to distinguish between significantly different cases. For example, Figure 2 illustrates some stylized resource distributions and their *ARLF*s. In the first row, both distributions are front-loaded (relative to the mid-point of the project's critical path duration, indicated by the dashed, vertical line) and have negative *ARLF*. Despite their different shapes, they could have the same *ARLF* value. Similarly, both distributions in the second row might have the same, positive *ARLF* value. More problematically, both distributions in bottom row have $ARLF \approx 0$.

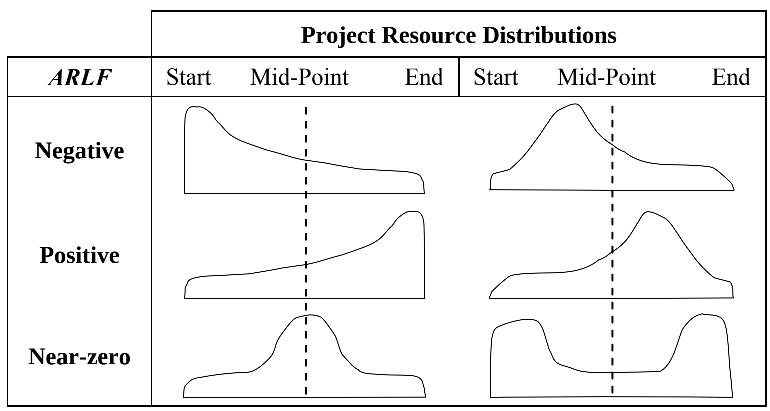


Figure 2: Example resource distributions over project time

Observation 1: Projects with $ARLF_l \approx 0$ can have dramatically different shapes.

---

[7] Based on scheduling each activity at its early start time (the "all EST" schedule)

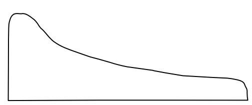
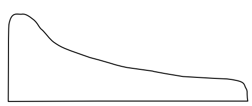
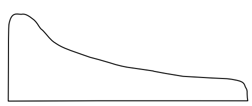
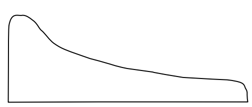
[8] The original definition of Z_{nlt} in (K&D, 1982) and (Kurtulus 1978, p. 59) assumes activities are indexed from 0 to $N_l - 1$ and therefore puts the equal sign in the second case rather than the first—i.e., $t \ge CP_l / 2$. However, it seems more intuitive to index activities from 1 to N_l. For example, in a 10-day project, with days numbered 1-10, we assign activities on day 5 (10 / 2 = 5) to the first half of the project.

Furthermore, the above definition of *ARLF* pertains to a single project. K&D (1982) define the *ARLF* for a problem as the average of its constituent projects' *ARLF*s:

$$ARLF = \frac{1}{L}\sum_{l=1}^{L} ARLF_l \tag{8}$$

However, this simple averaging loses a lot of important information in a MP context. Consider the following two observations.

Observation 2: As demonstrated in Figure 3, a problem containing *l* projects with identical *ARLF*s can have the same *ARLF* as a problem containing *l* projects with very different *ARLF*s. Thus, observation 1 applies at the *problem* level as well as the *project* level. Looking only at the mean *ARLF* of the constituent projects may fail to distinguish between dramatically different types of problems.

Project	*ARLF*	Distribution
1	-2	
2	-2	
3	-2	
Problem	**-2**	

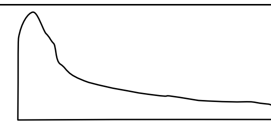
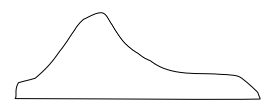
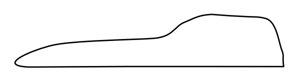
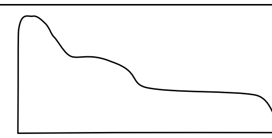
Project	*ARLF*	Distribution
1	-5	
2	-2	
3	1	
Problem	**-2**	

Figure 3: Two examples of *ARLF* calculation for an overall problem (using equation 8)

Observation 3: Equation (8) also fails to account for differences in the durations of the constituent projects. For example, consider the three projects in Figure 4, where the first project is twice as long as the second and third projects. According to equation (8), this problem's *ARLF* = 1.33, even though visual analysis confirms a highly front-loaded distribution of resources and an *ARLF* that should be highly negative. By calculating each project's *ARLF* on the basis of its own duration, rather than the duration of the overall problem (which is dictated by the longest of its projects), equation (8) provides misleading results.

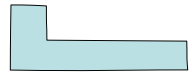
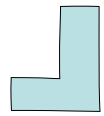
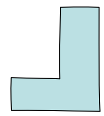
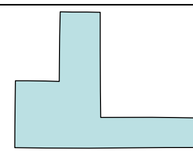
Project	*ARLF*	Distribution
1	-2	
2	3	
3	3	
Problem	**1.33**	

Figure 4: Example *ARLF* calculation for a problem using equation (8)

To address this issue, we introduce a new way to measure the distribution of resources in a MP problem by normalizing over the *problem's* critical path duration:

$$NARLF = \frac{1}{L \cdot CP_{Max}} \sum_{l=1}^{L} \sum_{t=1}^{CP_l} \sum_{k=1}^{K_{il}} \sum_{i=1}^{N_l} Z_{ilt} X_{ilt} \left(\frac{r_{ilk}}{K_{il}} \right) \tag{9}$$

where *NARLF* stands for *normalized ARLF* and $CP_{max} = \text{Max}(CP_1, \ldots, CP_L)$.[9]

We also introduce an additional measure, the variance in the *ARLF*s of a problem's constituent projects from its *NARLF*:

$$\sigma^2_{ARLF} = \frac{1}{L} \sum_{l=1}^{L} \left(ARLF_l - NARLF \right)^2 \tag{10}$$

The second moment of the resource distribution supplements the information about the characteristics of a RCMPSP.

4.4 Resource Contention

As a further complication with *ARLF* and *NARLF* as effective resource distribution measures, refer to the last row in Figure 2, about which we make the following observation.

Observation 4: As |*ARLF*| and |*NARLF*| → 0, they become less effective measures of the size of a resource distribution. That is, since *ARLF* (*NARLF*) provides a relative comparison of the resource load in the front-half of a project (problem) with the resource load in the back-half, this comparative value diminishes as the load

[9] Since *ARLF* and *NARLF* assume fungible resources, they are sensitive to disparities in the number of types, *K*. For example, in a three-project problem, if one of the projects uses four types of resources and the other two projects use only one of those types, then *K* = 4. If all three projects use four types of resources, then *K* = 4 also. For this reason, our experiments will use a constant *K* for all projects.

moves towards the mid-point of the project (problem).

This observation has important implications for the RCMPSP. Consider the four projects with *ARLF* = *NARLF* = 0 in Figure 5. Using the metaphor of a pipe to represent the resource constraints, a problem containing the two projects whose resource distributions are shown on the left (the upper one of which has been flipped vertically to emphasize its complementarity with its lower counterpart) is much easier to fit through the pipe without delaying some of its activities than a problem containing the two projects on the right. Neither *ARLF* nor *NARLF* captures the important difference between these two problems. Interestingly, K&D (1982) had trouble distinguishing the best priority rule from their experiments when $ARLF \approx 0$.[10] Therefore, we need another measure to distinguish the amount of resource contention.

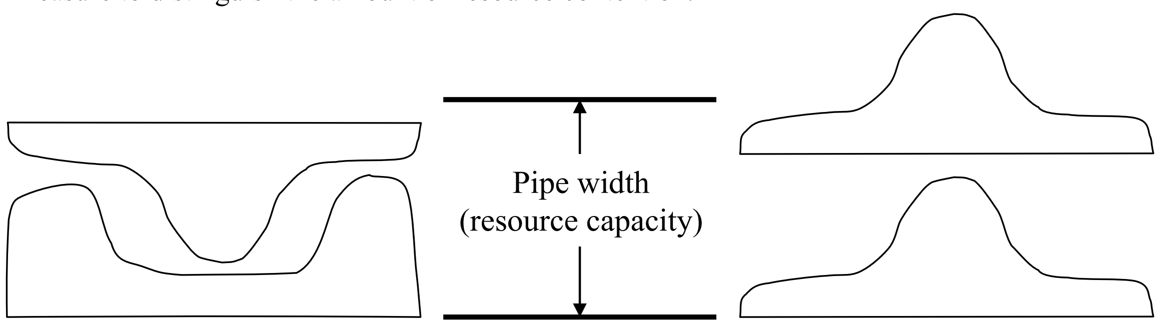


Figure 5: Two example problems, each containing two projects with *ARLF* = 0

To measure resource contention, Davis (1975) proposed the *utilization factor*, *UF*, which is calculated for each resource type as the ratio of the total amount required to the amount available in each time period, based on the problem's critical path duration. If $UF_k < 1\ \forall\ k$ in each time period, then there is no resource contention. To reduce computational intensity, K&D (1982) proposed averaging *UF* over intervals to get an *average utilization factor*, *AUF*. Using Figure 6 as an example, they proposed using $S = 3$ intervals, where $S_1 = CP_1$, $S_2 = CP_2 - CP_1$, and $S_3 = CP_3 - CP_2$, once the projects have been sorted from shortest to longest, such that $CP_1 \leq CP_2 \leq CP_3$.[11] The total amount of resource *k* required over any interval *s* is given by:

$$W_{sk} = \sum_{t=a}^{b} \sum_{l=1}^{L} \sum_{i=1}^{N_l} r_{ilk} X_{ilt} \tag{11}$$

where $a = CP_{s-1} + 1$, $b = CP_s$, r_{ilk} is the amount of resource *k* required by the i^{th} activity in project *l*, and *X* is defined as in equation (7). The *AUF* indicates the average tightness of the constraints on (i.e., the average

[10] We conjecture that their results in this case would be the most difficult to replicate with a randomly generated problem. By the way, their *ARLF* = 0 problem, given in (Kurtulus 1985), contains three projects with *ARLF*s of 2.875, -0.259, and -2.583, respectively. The *NARLF* for this problem is -0.56.

[11] K&D (1982, p. 163) use *M* instead of *S* and define it as the number of projects, but if any projects have equal durations, then $S \leq L$.

amount of contention for) each resource type:

$$AUF_k = \frac{1}{S}\sum_{s=1}^{S}\frac{W_{sk}}{R_k s} \quad (12)$$

where R_k is the (renewable) amount of resource type k available at each interval. Since the *AUF* is essentially a ratio of resources required to resources available, averaged across intervals of problem time, $AUF_k > 1$ indicates that resource type k is, on average, constrained over the course of a problem. To get the *AUF* for a problem involving K types of resources:

$$AUF = \text{Max}(AUF_1, AUF_2, \ldots, AUF_K) \quad (13)$$

We identify two issues with the *AUF* in Observations 5 and 6.

Observation 5: When the projects in a problem have similar durations (which is common when the projects are of similar size), then $S_1 >> S_{s>1}$, and averaging over these disproportionate intervals can obscure the situation.

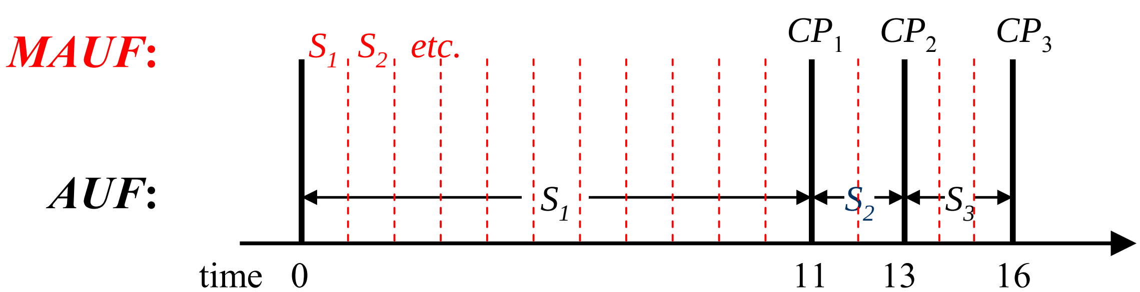


Figure 6: Example of time intervals formed by three projects in a problem

To ameliorate this issue, we propose averaging over equal intervals of problem time, such as the integer units indicated by the dashed, vertical lines in Figure 6.[12] Thus, in Figure 6's example, $S = 16 = CP_{max}$. We call this measure the *modified AUF*, *MAUF*, and equations (11-13) hold, although S is determined differently.

Observation 6: Determining a problem's *AUF* as the maximum of its resources' *AUF*s (equation 13) fails to distinguish between significantly different problems. For example, if a problem with three types of resources has $AUF_1 = 1.6$, $AUF_2 = 1.58$, and $AUF_3 = 1.59$, then $AUF = 1.6$ by equation (13). Since these three *AUF*s are almost equal and all greater than one, all three types of resources are highly constrained, and any activities which are unconstrained by the first type of resource are very likely to be constrained by one or both of the other types. However, if another problem has $AUF_1 = 1.6$, $AUF_2 = 0.6$, and $AUF_3 = 0.6$, then only the first type of resource is highly constrained, but the problem's *AUF* is also 1.6. Hence, two problems can have very different

[12] The actual size of these intervals can be chosen to limit computational intensity if necessary.

amounts of resource contention yet identical *AUF*s (or *MAUF*s).

To provide a clearer picture of resource contention, we propose augmenting the *MAUF* measure with a measure of the variance in the $MAUF_k$s:

$$\sigma^2_{MAUF} = \frac{\sum_{k=1}^{K}\left(MAUF - MAUF_k\right)^2}{K} \tag{14}$$

Note that this is a variance from the maximum, not from the mean. σ^2_{MAUF} will grow as the amount of resource contention in the non-max resource types deviates from the maximum. Therefore, all else being equal, higher σ^2_{MAUF} correlates with reduced problem delay.

5. Scheduling Algorithm

Besides the rule itself, a priority rule-based heuristic requires a *schedule generation scheme* (SGS). Boctor (1990) distinguishes between the "serial" and "parallel" use of priority rules by an SGS. In the serial SGS, each activity's priority is calculated once at beginning of the SGS algorithm, whereas in the parallel SGS an activity's priority is re-determined as necessary at each time step. We adopt the conventional parallel SGS (Bedworth and Bailey 1987; Kolisch 1996a), since studies comparing heuristic performance in both cases (on a single project) concluded that the parallel SGS outperforms the serial SGS in most cases (Kolisch 1996b; Lova and Tormos 2001).[13] Moreover, most MP studies have used the parallel SGS, which proceeds as follows. First, the overall problem duration is broken down into time steps. At each time step, the algorithm separates the activities into four disjoint sets: the *complete set*, ***C*** (finished activities), the *active set*, ***A*** (ongoing, "already scheduled" activities), the *decision set*, ***D*** (unstarted activities that depend only on activities in ***C***), and the ineligible set, ***I*** (activities which depend on activities in ***A*** or ***D***). ***A*** + ***C*** + ***D*** + ***I*** = $\sum_{l=1}^{L} N_l$. Since preemption is not allowed, the SGS automatically assigns resources to activities in ***A***. If the remaining resources are sufficient to perform the activities in ***D***, then the algorithm adds these to ***A***. If not, then it uses a priority rule to rank the activities in ***D***. The highest-ranking activities are added to ***A*** as resources allow. The time step ends when the shortest activity (or activities) in ***A*** finishes. Finished activities are moved to ***C***, and activities in ***I*** are checked for potential transfer to ***D***. The schedule is complete (i.e., the project durations are known) when all activities are in ***C***.

6. Pilot Study - Comparison with Kurtulus and Davis's 1982 Study

Because of K&D's (1982) salient contribution to the RCMPSP literature, we sought to replicate their

[13] Bouleimen and Lecocq (2003) achieved a superior result using a serial SGS and meta-heuristics.

results. Their experiment consisted of seven test problem networks, each with 3-5 projects (with 7-20 activities each), 34-63 total activities, and *ARLF* levels of -3, -2, -1, 0, 1, 2, and 3. They varied each network's *AUF* from 0.6 to 1.6 (11 levels), yielding 77 test problems. They tested the nine priority rules in Table 1 and determined the best performer in terms of R1 (and R2), as shown in Table 2. SASP and MAXTWK outperformed the other priority rules in 52 of the 77 problems. Kurtulus (1985) and Kurtulus and Narula (1985) provided further results.

Table 2: Results from K&D (1982) on best-performing priority rule under each value of *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	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

[a]Tied with MOF and MAXTWK
[b]Tied with SASP
[c]Tied with MAXSLK and SASP
[d]Tied with FCFS
[e]Omitted because no delays produced
[f]Tied with MOF, MAXTWK, FCFS, and SASP

In trying to replicate K&D's results, we began by applying our scheduling algorithm to their test problems. We could find only three of their seven networks.[14] We explored two of these (*ARLF* = -3 and 3) that both have three projects per problem. Besides the discrepancies in scheduling algorithms, which we feel are actually slight, our efforts to replicate K&D's results led to some more interesting observations:

Table 3 compares our results. We were not able to replicate K&D's results exactly, although we came close. We carefully explored several potential reasons for this. First, we scheduled the problems by hand for each priority rule. While tedious, the method is exact. We were able to replicate the results of our scheduling algorithm with this method. Second, we note that results can vary for the SOF, MINSLK, SASP, MINTWK, MAXTWK, and FCFS priority rules because these contain a random tie-breaker. Thus, we scheduled each problem multiple times, noting the min, max, and mean values obtained. While the results would vary slightly, K&D's results did not fall within these bounds. Third, we noted that while K&D found no delay for the {*ARLF* = 3, *AUF* = 0.7} problem, there are actually minor resource constraints (*UF* > 1) at a couple of time steps in this problem. While most of the priority rules provided an equally good choice between the eligible activities in these situations, some delay is nevertheless experienced. Overall, then, we are left to conclude that some minor,

[14] (K&D, 1982) does not provide any of the test problems; (Kurtulus 1978) provides three of the seven. Neither source details the scheduling algorithm.

indiscernible differences exist between our scheduling algorithm and K&D's.

Besides the discrepancies in scheduling algorithms, which we feel are actually slight, our efforts to replicate K&D's results led to some more interesting observations:

Table 3: Comparison of our results with K&D (1982)

	ARLF = -3		*ARLF* = 3	
AUF	**K&D's**	**Ours**	**K&D's**	**Ours**
0.6	Tie: MINSLK, MOF, MAXTWK	Tie: MINSLK, MOF, MAXTWK	No delay	No delay
0.7	MINSLK	MINSLK	No delay	Tie: MINSLK, MOF, MAXTWK, FCFS, SASP
0.8	MINSLK	MINSLK	Tie: MINSLK, MOF, MAXTWK, FCFS, SASP	Tie: All but MINTWK
0.9	SASP	SASP	MINSLK	SASP
1	SASP	SASP, MOF	MINSLK, SASP	SASP
1.1	SASP	SASP	SASP	SASP
1.2	SASP	SASP	SASP	SASP
1.3	MAXSLK	SASP, MAXTWK	SASP	SASP
1.4	SASP	SASP	SASP	MAXTWK
1.5	SASP	SASP	SASP	MAXTWK
1.6	MAXTWK	SASP	SASP	SASP

Observation 7: Each column in Table 2 exhibits a pattern. One priority rule tends to provide the best results for a particular network when its average resource constraints are minor, while another tends to dominate where $AUF \geq 1.0$. Nevertheless, there are many exceptions to these patterns, as particular levels of resource availability cause certain activities to wait, and these holdups play out in different ways through the remainder of the network, allowing a different rule to win and occasionally disrupt the general pattern.[15]

To explore this further, we generated seven random networks of our own, with *ARLF* and *AUF* settings varied as in Table 2, to yield 77 test problems. We observed patterns similar to those noted above, but in most cells the best rule differed from Table 2. Next, we generated 77 random networks, again with *ARLF* and *AUF* settings per Table 2. This again gave us 77 test problems, albeit with each having a unique set of activities, network structure, and resource requirements. Table 4 shows these results, where differences from Table 2 are shaded. Now we lose the patterns in each column. Finally, when we generated 19 more sets of 77 independent networks, the results again differed each time, without any discernable patterns.

These results call into question the conclusion that preferable priority rules can be recommended solely on the basis of a problem's *ARLF* and *AUF*. Apparently, project complexity, number of resource types, and perhaps other characteristics have a significant effect on resource contention, delays, and thus priority rule performance. Moreover, as we described above, the *ARLF* and *AUF* measures themselves have significant shortcomings.

[15] These disruptions may relate to what Herroelen and De Reyck (1999) call "phase transitions."

Finally, the priority rules' tie-breakers are invoked often, making most of the rules essentially two-phase rules. In particular, the random tie-breaker for FCFS (which is itself the tie-breaker for many other priority rules) implies that the overall results may often be largely random. Therefore, we concluded that a more comprehensive study (than K&D (1982) and others conducted since then) would be valuable.

Table 4: Best-performing priority rule(s) with 77 different, random test problem networks

	ARLF						
AUF	**-3**	**-2**	**-1**	**0**	**1**	**2**	**3**
0.6	MINSLK	LALP	Tie: All	MINSLK[b]	Tie: All	MOF[a]	MINSLK
0.7	LALP	MINSLK[b]	MOF[a]	MAXSLK	MINSLK	MINSLK	MAXSLK
0.8	Tie: All	MOF[a]	MINSLK	MINSLK	LALP	MAXSLK	SASP
0.9	MINSLK[b]	MAXSLK	MINSLK	LALP	MOF	SASP	SASP
1	Tie: All	MINSLK	LALP	MOF	LALP	LALP	SASP
1.1	MOF[a]	MINSLK	MAXSLK	SASP	LALP	SASP	LALP
1.2	MINSLK	MAXSLK	SASP	SASP	SASP	LALP	LALP
1.3	MAXSLK	LALP	SASP	SASP	SASP	LALP	LALP
1.4	MINSLK	MAXSLK	MAXTWK	SASP	MAXTWK	MAXTWK	MAXTWK
1.5	MINSLK	MOF	LALP	SASP	LALP	SASP	MOF
1.6	MAXTWK	MINSLK	LALP	LALP	SASP	SASP	MAXTWK

[a]Tied with MAXSLK, LALP, MINTWK [b]Tied with MAXSLK

7. Our Study

Various RCMPSP studies disagree about the best-performing priority rules, and their results are not always easy to replicate. Above, we identified some likely reasons for these discrepancies. To give managers more helpful guidance, we designed a more comprehensive experimental study of the RCMPSP.

7.1 Set-Up

We compiled a set of 20 popular priority rules from the literature (Table 1 and Table 5), some of which were developed specifically for the RCMPSP and others which have been successful in a single-project environment. To increase their comparability, we standardized the tie-breaker for all priority rules to be FCFS.

We designed a full factorial experiment to test the influence of the factors listed in Table 6. To maximize the insights from varying the last four factors, we held the first three constant.[16] The choices for *NARLF* and *MAUF* levels follow K&D (1982). We designated two levels of project complexity, "high" ($C = 0.69$) and "low" ($C = 0.14$).[17] We used these to form four variations in problem complexity: all high-complexity projects ("HHH"), all low-complexity projects ("LLL"), and two intermediate combinations. Furthermore, we wanted some problems where all of the individual resources' *MAUF*s were equal (i.e., where $\sigma^2_{MAUF,des} = 0$) and others where one resource's *MAUF* determined the overall problem's $MAUF_{des}$ while the other three types of resources

[16] No specific relationship has been reported between portfolio size or project size and the solution quality obtained by the various heuristics (Hartmann and Kolisch 2000; Kurtulus and Davis 1982; Lova and Tormos 2001). Meanwhile, since K is used to determine *NARLF* (equation (9)), its variation would be confounded with *NARLF*.
[17] $C = 0.69$ implies 75 non-redundant arcs among 20 activities and $C = 0.14$ implies 30.

had a significantly different *MAUF*. Thus, we needed 7 × 11 × 4 × 2 = 616 test problems for this experiment. Standard problem generators and test sets such as ProGen/PSPLIB (Kolisch *et al.* 1995) cannot create MP prob-

Table 5: Additional priority rules analyzed in this study

Priority Rule (* = multi-project)	Basis	Formula	Comments
10. **RAN**—Random		Activities selected randomly	Best in study by Akpan (2000)[a] but used by others mainly as a benchmark (e.g., Davis and Patterson 1975)[a]
11. **EDDF**—Earliest Due Date First	Activity	Min(LS_{il})	
12. **LCFS**—Last Come First Served	Activity	Max(ES_{il})	
13. **MAXSP**—Maximum Schedule Pressure	Activity	$Max\left(\frac{t - LF_{il}}{d_{il}W_{il}}\right)$, where W_{il} is the percentage of the activity remaining to be done at time t	Also known as "critical ratio" (e.g., Chase *et al.* 2006)[a]
14. **MINLFT**—Minimum Late Finish time	Activity	Min(LF_{il})	Best in study by Mohanty and Siddiq (1989); equivalent to MINSLK in serial scheduling case (Kolisch 1996a)[a]
15. **MINWCS**—Minimum Worst Case Slack	Activity, Resource	Min(LS_i – Max[$E_{(i,j)}$ | $(i,j) \in AP_t$]), where $E_{(i,j)}$ is the earliest time to schedule activity j if activity i is started at time t, and AP_t is the set of all feasible pairs of eligible, un-started activities at time t	Best in study by (Kolisch 1996a)[a]; without resource constraints, reduces to MINSLK
16. **WACRU**—Weighted Activity Criticality & Resource Utilization	Activity, Resource	$Max\left(w\sum_{q=1}^{N_i}\left(1 + SLK_{iq}\right)^{-\alpha} + (1-w)\sum_{k=1}^{K}\frac{r_{ik}}{R_{Max,k}}\right)$, where N_i is the number of immediate successors of the i^{th} activity, w is the weight associated with N_i ($0 \le w \le 1$), SLK_{iq} is the slack in the q^{th} immediate successor of the i^{th} activity, and α is a weight parameter	Best in study by (Thomas and Salhi 1997)[a] We use $w = 0.5$ and $\alpha = 0.5$
17. **TWK-LST***—MAXTWK & earliest Late Start time (2-phase rule)	Activity, Resource	Prioritize first by MAXTWK (without FCFS tie-breaker) and then by Min(LS_{il})	(Lova and Tormos 2001); min. late start time (MINLST) was best in study by (Davis and Patterson 1975)[a]
18. **TWK-EST***—MAXTWK & earliest Early Start time (2-phase rule)	Activity, Resource	Prioritize first by MAXTWK (without FCFS tie-breaker) and then by Min(ES_{il})	(Lova and Tormos 2001)
19. **MS**—Maximum Total Successors	Activity	Max(TS_{il}), where TS_{il} is the total number of successors of the i^{th} activity in the l^{th} project	Best in study by (Kolisch 1996a)[a]
20. **MCS**—Maximum Critical Successors	Activity	Max(CS_{il}), where CS_{il} is the number of critical successors of the i^{th} activity in the l^{th} project; $CS_{il} \in TS_{il}$	

[a]Discusses a *single* project (RCPSP)

Table 6: Experimental design

Constant Factors	Setting	Main Factors	Levels
L	3 projects per problem	$NARLF$	7 levels: -3, -2, -1, 0, 1, 2, 3
N	20 activities per project	$MAUF$	11 levels: 0.6 - 1.6 in increments of 0.1
K	4 types of resources per activity	C	4 levels: HHH, HHL, HLL, and LLL
		σ^2_{MAUF}	2 levels: 0 (no variance) and 0.25 (high variance)[18]

lems to these specifications, so we used a test problem generator recently developed by Browning and Yassine

[18] The basis for choosing $\sigma^2_{MAUF,des}$ = 0.25 for the high variance case is explained in [a separate working paper by the authors].

(2008). To enable the identification of random effects, we used 20 replications for each setting, thus generating 12,320 problems (36,960 networks). We solved each problem with 20 priority rules, thus producing 246,400 experimental outcomes. We specified each outcome in terms of the five objective functions in §4.1, thereby yielding 1,232,000 data points.

7.2 Superiority of the New Measures

We used an analysis of variance (ANOVA) to compare two linear models, the first with the five factors *NARLF*, *MAUF*, *C*, σ^2_{MAUF} , and priority rule, and the second with *ARLF* and *AUF* instead of *NARLF* and *MAUF*. The results are shown in the Appendix. For objective R3, the respective R^2 measures for the two models were 82% and 65%. For objective R5, the R^2 measures were 92% and 65%. We take these results as a tentative confirmation of the superiority of the *NARLF* and *MAUF* measures to *ARLF* and *AUF*, as we supposed in §4. Also, as the third ANOVA model in the Appendix shows, the five original factors and their interactions has an R^2 of 85% for R3 and and 94% for R5. Thus, we also infer that our collection of measures does a reasonable job of explaining performance variances. We discuss the individual factors and interactions in the next sub-sections.

7.3 Results for R3: Average Percent (Project) Delay

Although we analyzed the results for all five objectives, we present the results for R3 and R5 (for the reasons mentioned in §4.1).[19] Starting with R3, Figure 7 shows a one-way analysis of means (ANOM) for all 20 rules (averaging over all other factors), assuming a 95% confidence level. Overall, the "winning" rule (i.e., the one with the smallest average percent delay) is TWK-LST. MAXTWK and MINWCS tied (statistically) for second place, and EDDF, MAXSP, MINLFT and TWK-EST tied for third.[20] Obvious "losers" include MAXSLK as the worst, MS and MCS tied for second-worst, and SOF and MINTWK tied for third-worst. LALP also performed well below average. Note that these rules performed worse than RAN, which itself outperformed the mean.[21] MINSLK and SASP, the two rules recommended by K&D (1982), did not perform especially well.

To investigate the underlying reasons for the success of the TWK-LST priority rule and the relative ineffectiveness of the other rules in minimizing R3, it is necessary to look into how TWK-LST works (Lova &

---

[19] Since there is no reason to believe that the outcomes are normally distributed for any of the five objectives (e.g., Kolisch 1996a; Kurtulus and Davis 1982)—and, indeed, they are not, especially for the low *MAUF* values—the use of parametric statistical techniques, which assume normally distributed outcomes, could be inappropriate. Nonparametric statistics are used in such cases, although they are less powerful than parametric tests. With large samples, however, the difference becomes minor, and parametric tests can be used anyway, since the *p*-value will be nearly correct even if the population is fairly far from normal (Gibbons 1993). Accordingly, other researchers have used parametric statistics to analyze single- and multi-project delays (e.g., Lova and Tormos 2001; Thomas and Salhi 1997). Since we generated a large sample size, and since we are especially interested in the results under high levels of resource constraint (where the distribution is closer to normal), we use parametric techniques.

[20] A statistical tie implies that two or more rules have no statistically significant difference at 95% confidence.

[21] An ANOM for R1 (total delay) and R2 (average delay) shows the same loser rules but no obvious winners.

Tormos 2001). This priority rule first favors activities with MAXTWK (which performed well on its own), which means it considers all (three) projects simultaneously. Meanwhile, the other less effective rules favor either the shortest (or longest) project, or focus on the number of successors within a single project. Rules that consider all three projects (such as MAXTWK, MINWCS, and TWK-EST) generally perform well on R3.

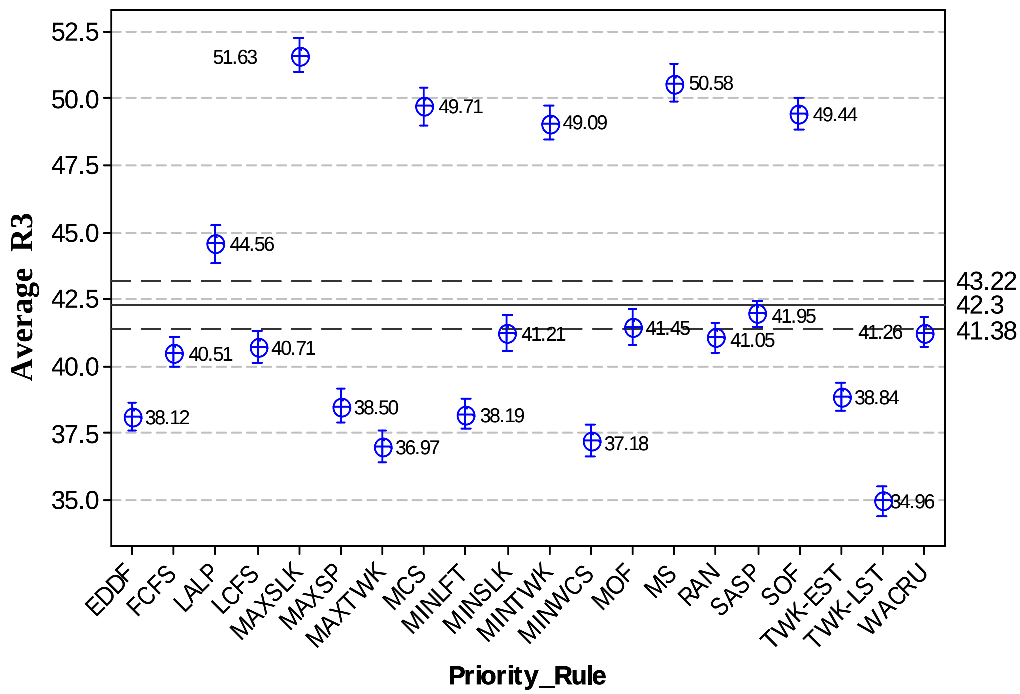


Figure 7: One-way analysis of means (ANOM) for R3 (α = 0.05)

To further explore the influence of each factor on R3, we looked at the main effect plots (Figure 8 and 10). The left side of Figure 8 shows an increase in mean R3 with a decrease in complexity (from HHH to LLL), which is reasonable since high-complexity problems already take longer because of greater precedence constraints.[22] That is, when the precedence constraints play a greater role, there is less concurrency among activities and less additional delay to be caused by the resource constraints, whereas when the precedence relationships are less constraining, resource constraints can have a larger effect.[23] The middle of Figure 8 shows R3 decreasing with an increase in *MAUF* variability, which is also expected since the σ^2_{MAUF} = 0 case implies that the problems are equally constrained by all (four) resource types. However, σ^2_{MAUF} = 0.25 represents cases where the problems are mainly constrained by only one of the resource types. On the right side of Figure 8, we also see the expected increase in percent lateness with higher *MAUF* (i.e., greater resource contention).

[22] Any delay will be a smaller percentage of a "long" problem.
[23] Kolisch (1999) reported improved performance results from commercial software packages on RCPSPs with many precedence constraints.

A fourth main effect plot in Figure 9 shows a decrease in lateness due to an increase in *NARLF* values. That is, negative *NARLF*s cause greater delays than equal, positive *NARLF*s, because the former implies a front-loading of the resource constraints, which has implications for all downstream activities in the network, especially when preemption is not allowed.

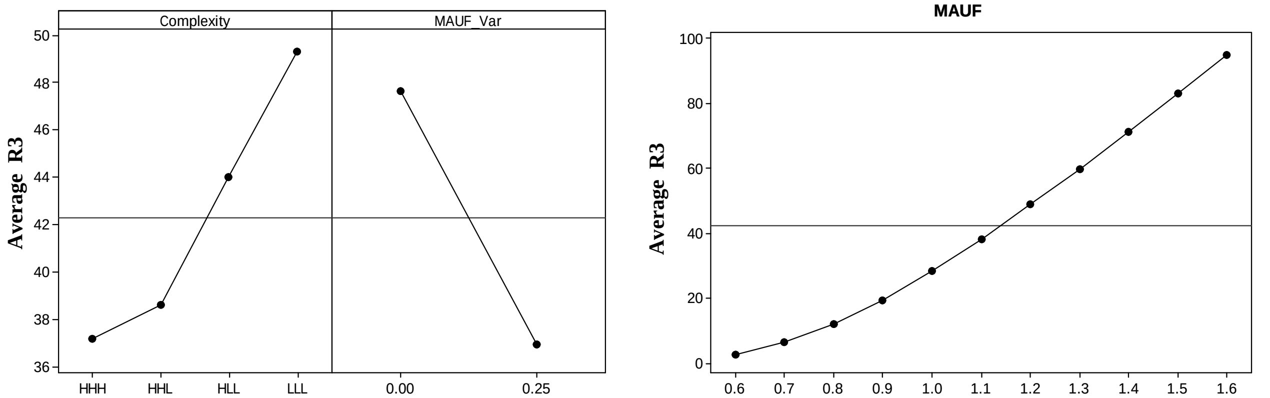


Figure 8: Main effect plots for R3

Observation 8a: Delaying the early (upstream) activities in a project causes a greater average percent delay than holding up the downstream activities.

The ANOVA revealed that all five main factors (the four factors in Figures 9 and 10, plus the priority rule factor) and all two-way interactions were significant at 1% levels.[24] We now turn to a discussion of the two-way interactions between the four main factors and the priority rules. First, the *MAUF* interaction plot in Figure 10 indicates that, as expected, there is little difference in rules when resource constraints are low. However, as *MAUF* increases, the disparity between rules grows. To see these differences more clearly, we performed a *t*-test on the mean of each rule at selected *MAUF* levels, resulting in Table 7. MINWCS and MAXSP are the best rules for low *MAUF* values. At high *MAUF* values, TWK-LST wins.[25]

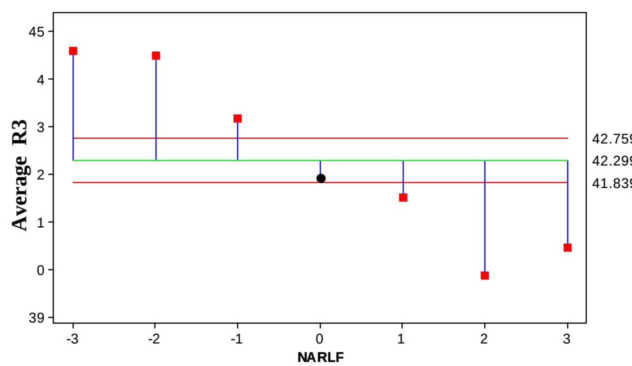


Figure 9: *NARLF* main effect plot for R3

[24] In the ANOVA model, we consider the effect of all five factors and ten two-way interactions. The effects of higher-order interactions were not considered in this model, so their influence is confounded within the error term. However, we address some particular three-way interactions below; we found these to be relatively inconsequential.

[25] The winners at each level form a kind of Pareto front. This front is somewhat fuzzy since it consists of all statistically tied rules at each level, as listed in Table 7.

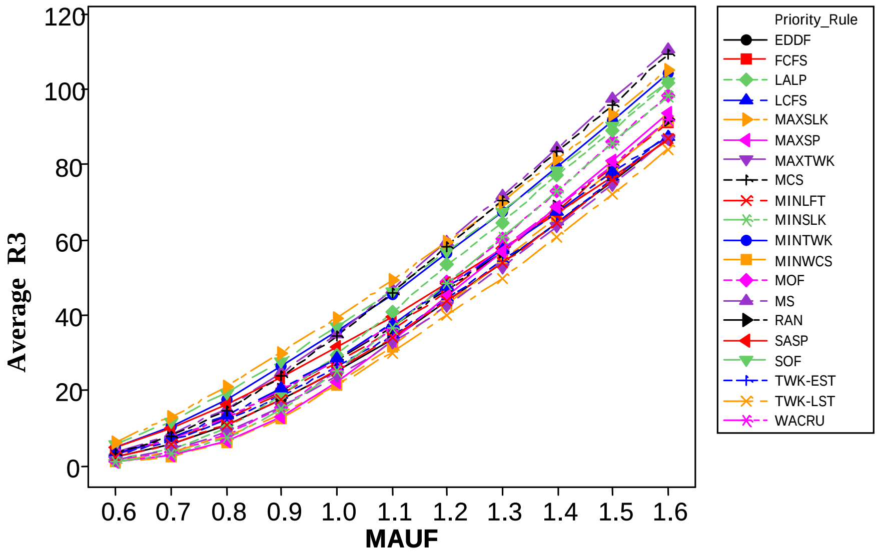


Figure 10: R3 by *MAUF* level and priority rule (two-way interaction plot)

Table 7: Best rules by *MAUF* level at 95% confidence

MAUF:	0.6	0.8	1.0	1.2	1.4	1.6
Best rule(s):	MINWCS MAXSP MINSLK	MINWCS MAXSP	MINWCS TWK-LST MAXSP	TWK-LST	TWK-LST	TWK-LST SASP

Second, the *NARLF* interaction plot shown in Figure 11 exhibits the superiority of TWK-LST at all levels, followed by MINWCS and MAXTWK. Table 8 shows the significant group of best rules at each *NARLF* level. Since we are especially interested in the best rules under conditions of high resource constraints, we also looked at the subset of problems with highly-constrained resources (i.e. with $MAUF \geq 1.4$). However, we found no significant change in the winning rules in the case of this three-way interaction.

Third, we looked at varied complexity levels (Figure 12 and Table 9). TWK-LST outperforms the other rules at high complexity. At low complexity, SASP is best but statistically ties with TWK-LST, MINLFT and EDDF. The behavior of SASP is especially interesting. In contrast to the other rules, SASP seems to be robust to changes in complexity: it is the only rule whose performance improves as C decreases. As complexity decreases, the number of precedence constraints in each project's network decreases. However, SASP does not consider the number of precedence constraints when making the prioritization decision, so its robustness to complexity makes sense. Finally, we again looked at the subset of problems with $MAUF \geq 1.4$ (three-way interactions), but these results were fairly similar.

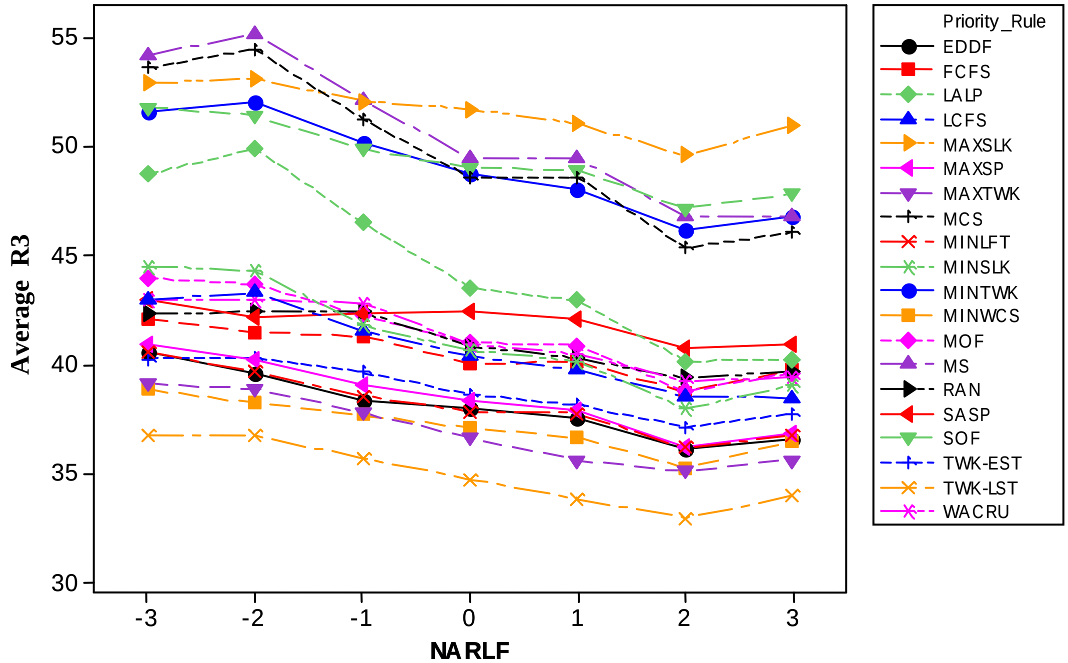


Figure 11: R3 by *NARLF* level and priority rule

Table 8: Best rules by *NARLF* level at 95% confidence

NARLF:	-3	-2	-1	0	1	2	3
Best rule(s):	TWK-LST MINWCS MAXTWK	TWK-LST MINWCS MAXTWK EDDF MINLFT	TWK-LST MINWCS MAXTWK EDDF MINLFT	TWK-LST MAXTWK MINWCS MINLFT	TWK-LST MAXTWK	TWK-LST MAXTWK MINWCS	TWK-LST MAXTWK

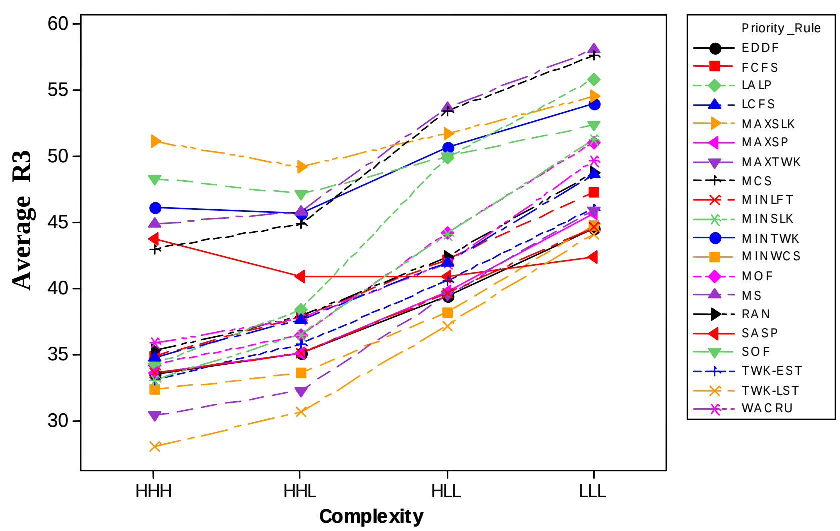
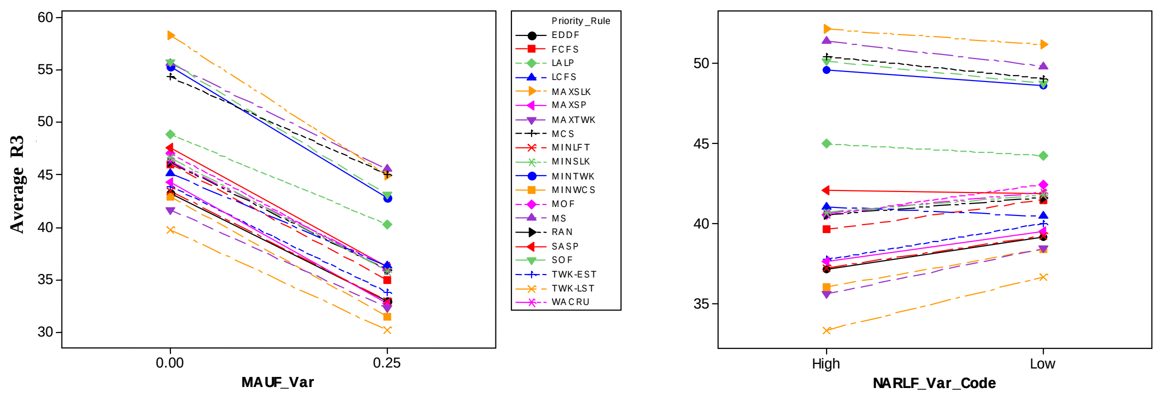


Figure 12: R3 by complexity level and priority rule

Table 9: Best rules by complexity level at 95% confidence

C:	HHH	HHL	HLL	LLL
Best rule(s):	TWK-LST MAXTWK	TWK-LST MAXTWK	TWK-LST MINWCS EDDF MAXTWK	SASP TWK-LST MINLFT EDDF TWK-EST

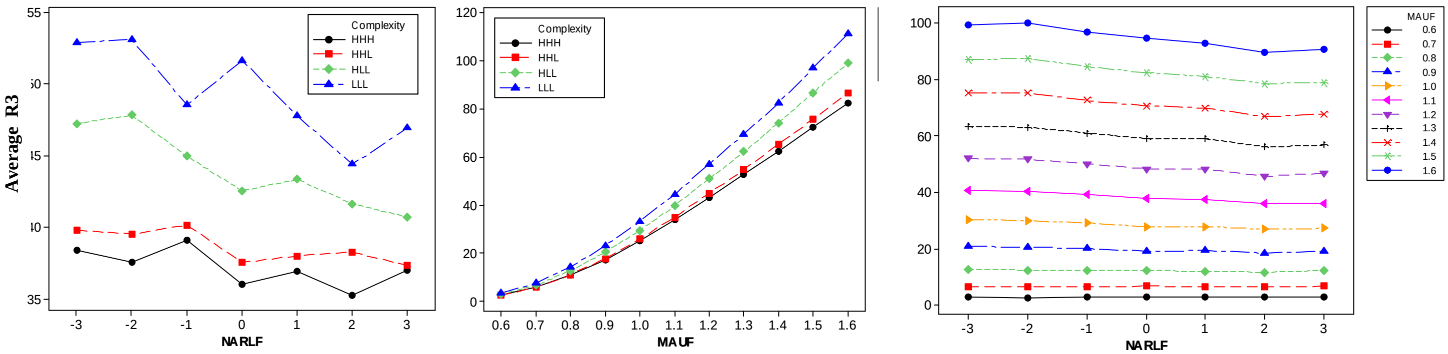
Fourth, regarding σ^2_{MAUF}, TWK-LST was the best rule at zero variance, while it statistically tied with MINWCS at 0.25 variance, as shown in Figure 13(a). As expected, all rules perform better as a problem's resources are constrained by fewer of its resource types. However, this did not change the ranking of most rules. Considering only the problems with highly-constrained resources (i.e. $MAUF \geq 1.4$) did not much alter the picture. We also looked at σ^2_{NARLF} levels (Figure 13(b)), where TWK-LST was the best priority rule regardless. Again, the ranking of the rules did not change much with this factor, although as σ^2_{NARLF} increases, it becomes easier to distinguish the rules' performance.



(a) (b)

Figure 13: R3 by σ^2_{MAUF} & σ^2_{NARLF} and priority rule

Figure 14 shows two-way interaction plots for *NARLF*, *MAUF*, and *C*. In Figure 14(a), we again see the effects of Observation 8a. Furthermore, the effect is heightened as *C* decreases (**Observation 8b**), since the LLL regression line has a steeper negative slope than such a line through the HHH points. Hence, it is worse to delay activities at the beginning of problem than at the end, and this effect is exacerbated as complexity decreases. Observation 8b might be counter-intuitive, since one might think that the "incompressibility" of high-complexity problems would tend to propagate any delays experienced early in the project. While this indeed occurs, its effects are apparently small in terms of a percentage of the greater length of those projects. Next, Figure 14(b) repeats the message of Figure 8(c) while indicating that the increase in percent delay with *MAUF* is regulated by *C*, growing more quickly when *C* is lower. Finally, Figure 14(c) is similar Figure 2 in K&D (1982), although the variance among the points connected by each line is smaller because we have averaged over a much larger set of problems. Note that the slope of the lines becomes increasingly negative as *MAUF* increases. That is, we have a flat line at $MAUF = 0.6$ and the greatest negative slope at $MAUF = 1.6$. Hence, the phenomenon of increasing delays with lower *NARLF* is exacerbated by higher *MAUF* (**Observation 8c**), just as it was by decreased *C*.



(a) (b) (c)

Figure 14: Two-way interaction plots for *NARLF, MAUF,* and *C*

7.4 Results for R5: Percent (Problem) Delay

We repeated the above set of analyses for R5, although we will report only on the key differences from the R3 results. While R3 attends to the effects of delay on the projects individually, R5 only accounts for delays that lengthen the overall portfolio of projects. While individual project managers would care more about R3, program or portfolio managers might have reason to focus on R5. However, since R5 is driven by a single project, it is a less sensitive measure than R3.

Figure 15 shows the ANOM for R5 (cf. Figure 7). While LALP and MS were losers according to R3, their performance improved greatly for R5. MINWCS, TWK-LST, and MAXSP performed well according to both R3 and R5, while MAXSLK, MINTWK and SOF performed poorly by both measures. It is worth noting that SOF has performed well in many single-project situations. While still decent, the relative performance of TWK-LST declined by R5 versus R3. MINWCS emerges as the overall best rule by R5.

Speculating on the reasons behind the MINWCS rule's (Kolisch 1996a) success requires looking more closely at how it works. This rule compares each potential pair of activities in the decision set, ***D***, based on the delay to one if the other is activated in the current time step. An activity's WCS is defined as the difference between its LST and its maximum delay caused by a preference for any other activity. This rule seems to prevent the most damaging delays from occurring. However, it does this at a much greater computational expense than the other rules, since it must "look ahead" for the potential downstream delays in terms of each possible pair of activities in ***D***. Note that this rule also performs well for R3 (except for high *MAUF* cases).

Finally, while SASP performed well by R3 in several situations, it was the worst rule by R5. Taking the shortest activity from the shortest project is good for minimizing the average percent delay to all projects, but the duration of a *problem* is governed by the duration of its longest project, which is given lowest priority by

SASP. Thus, SASP performs very poorly with R5, while LALP performs well.[26]

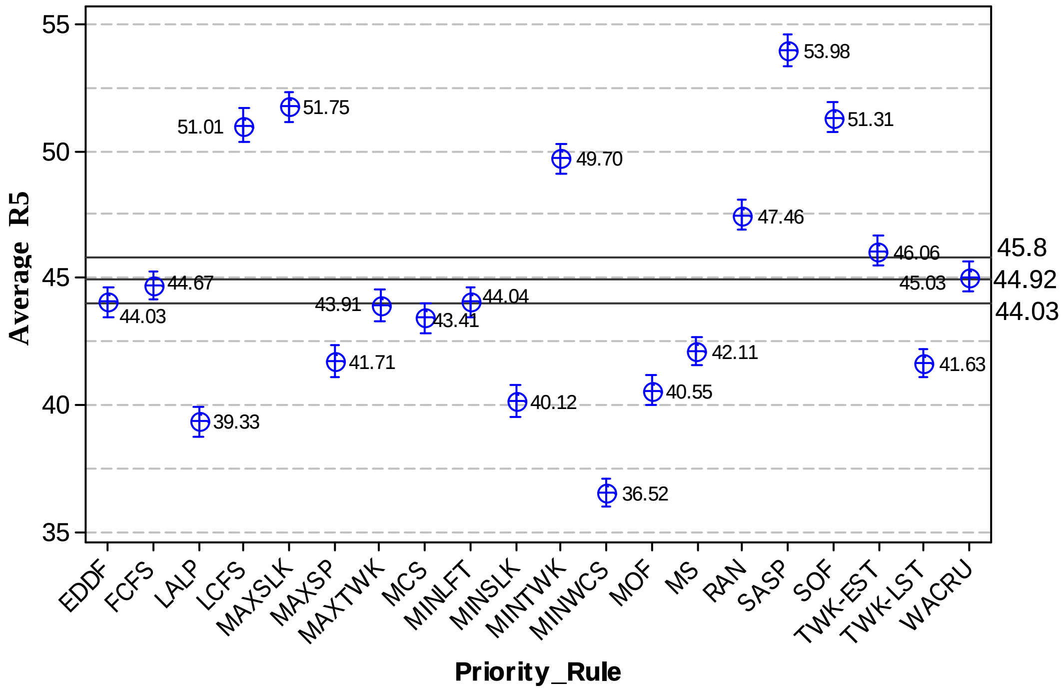


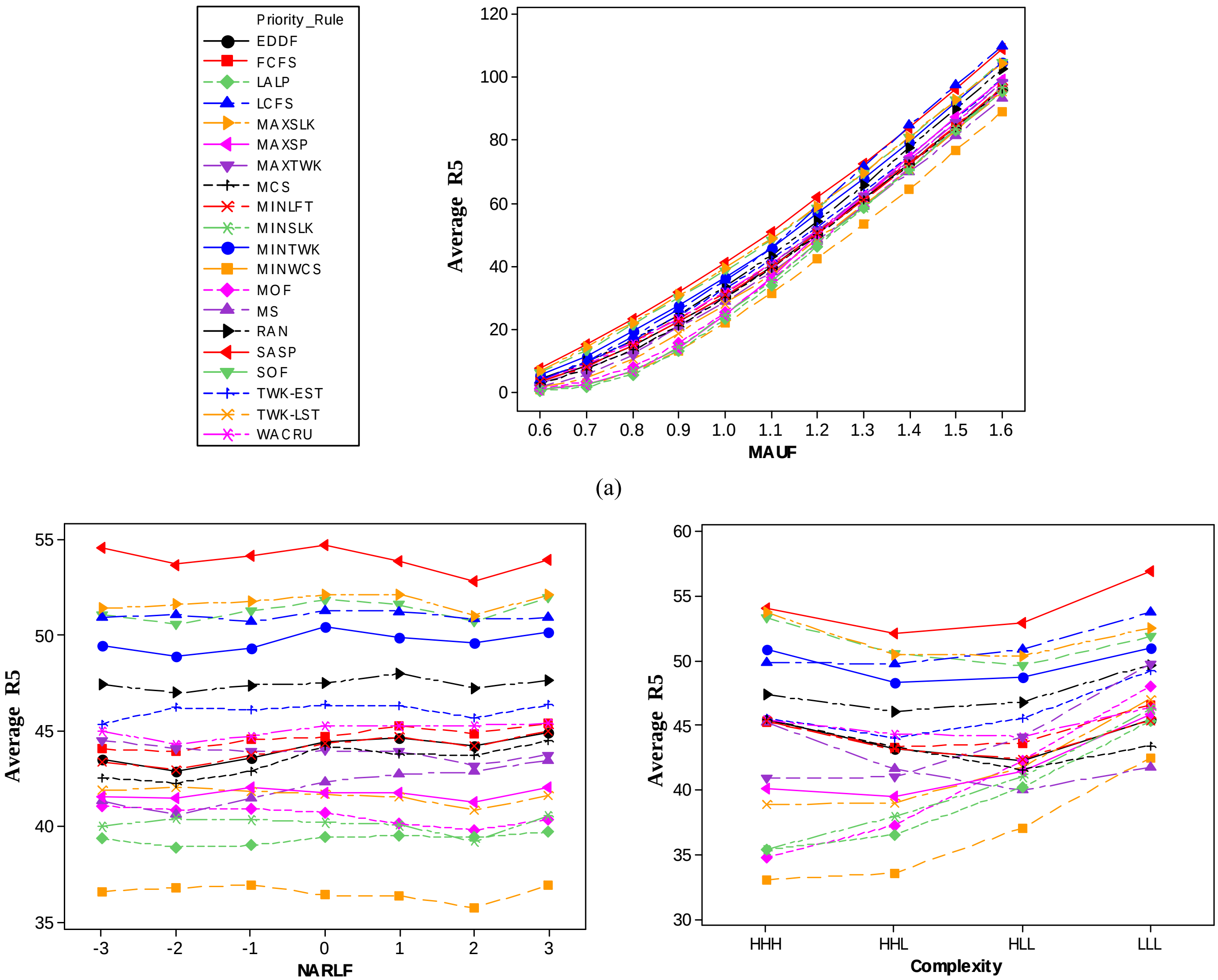
Figure 15: One-way analysis of means (ANOM) for R5 (α = 0.05)

The ANOVA based on R5 also revealed that all five main factors and two-way interactions are significant at the 1% level. The overall trends in the two-way interaction plots are mostly similar to those of R3, although the performance of various rules differs. Figure 16(a) (cf. Figure 10) shows the expected reduction in performance as *MAUF* increases. MINWCS emerges as the best rule for *MAUF* ≥ 1.0. SASP performs poorly at all *MAUF* levels. While also showing the clear superiority of MINWCS and the inferiority of SASP, Figure 16(b) does not show improved performance with higher *NARLF* values, as seen in Figure 11 (for R3). This would seem to be because the delays early in the projects (which occur due to the front-loading of the resource demands) are largely absorbed by the two shorter projects, whose delays do not usually show up in R5. Thus, Observation 8 does not apply to R5 (which we also confirmed by examining other two-way interactions).

Figure 16(c) indicates the trend towards diminished performance with lower complexity levels, as per Figure 12, although SASP no longer bucks the trend. Instead, several other rules now prevail against the general trend by performing poorly under situations of high complexity. In particular, MS and MCS perform better as complexity decreases. Perhaps this is because, as *C* decreases, the number of precedence constraints in each project's network decreases. This dearth of precedence constraints makes problems "harder," because ***D*** is

[26] Although we did not study it, we suspect that a "shortest activity from the longest project" (SALP) rule might perform well for R5.

larger, and thus there are more potential bad decisions that a rule can make. In these situations, rules that happen to focus on the factors that are important for a particular objective (versus on generally important ones) perform better. Rules that do well on R5 tend to prioritize the longest project. The longest project will often have the longest chain of activities, and therefore its activities will tend to have more successors than the shorter projects' activities. As *C* decreases, the number of successors becomes a more discriminating feature of a network. (In high complexity cases, all of the activities have a lot of successors.) The MS and MCS rules favor the longest project in a problem, the minimization of which implies good results for R5, and this becomes more prominent as complexity decreases. Finally, since the σ^2_{MAUF} interaction plot for R5 did not show any unusual trends, we omit it here.



(a) (b) (c)

Figure 16: R5 interaction plots between priority rules and *MAUF, NARLF,* and *C*

8. Implications for Managers

In the previous section, we identified, confirmed, and discussed several important factors that contribute to project and portfolio delay. To distill these results for managers, we developed two decision tables (Tables 10 and 11) to aid in selecting the best priority rule for a particular situation. Here we clearly see the different results for R3 and R5. From an individual project manager's point of view, R3 is a more appropriate objective, whereas R5 might align more with an executive's or portfolio manager's point of view. The different results obtained by these two objectives may relate to the friction that occurs between managers at different organizational levels.

Table 10: Summary of results for R3 (α = 0.05)*

	Resource Distribution					
	Front-loaded ***NARLF*** **= -3 & -2**		**Not front- or back-loaded** ***NARLF*** **= -1 & 0 & 1**		**Back-loaded** ***NARLF*** **= 2 & 3**	
Resource Contention	*C*: **HHH**	*C*: **LLL**	*C*: **HHH**	*C*: **LLL**	*C*: **HHH**	*C*: **LLL**
Low (*MAUF* = 0.6-0.8)	MINWCS MAXSP MINSLK MOF TWK-LST MAXTWK LALP	MINWCS MAXSP TWK-LST MINSLK MINLFT EDDF MAXTWK MOF	MAXSP MINWCS MINSLK MOF LALP TWK-LST MAXTWK	MINWCS MAXSP MINSLK TWK-LST MOF	MINWCS MINSLK MAXSP MOF LALP TWK-LST MAXTWK	MINWCS MAXSP MINSLK TWK-LST MINLFT
Medium (*MAUF* = 1-1.2)	TWK-LST MAXTWK	SASP TWK-LST MINWCS EDDF MINLFT MAXTWK TWK-EST MAXSP FCFS	TWK-LST MAXTWK	TWK-LST SASP EDDF MINLFT MINWCS MAXTWK TWK-EST FCFS LCFS	TWK-LST	TWK-LST EDDF MINLFT SASP MINWCS MAXSP MAXTWK TWK-EST LCFS
High (*MAUF* = 1.4-1.6)	TWK-LST MAXTWK FCFS TWK-EST	SASP MINLFT	TWK-LST	SASP TWK-LST	TWK-LST LCFS MAXTWK TWK-EST	SASP LCFS

*Multiple entries in each cell are listed in order of increasing means (i.e., the best rule is listed first).

We observe several patterns in these tables. First, the number of winning (statistically tied) rules decreases with greater *MAUF* and *C* in both tables. Also, for both R3 and R5, the results seem to be fairly robust to *NARLF*. That is, while *NARLF* affects the amount of delay for R3 (Observation 8), it does not much affect which rule wins. For R3, under tight resource constraints (high *MAUF*), TWK-LST performs well under high *C*, while SASP performs well under low *C*. For R5, under high *MAUF*, MINWCS performs well regardless of *C*.

Thus, if a manager wants to do well with R5, MINWCS is our overall recommendation for a robust rule in a variety of situations where resources are moderately- to highly-constrained. For R3, we recommend TWK-LST, except for cases where *MAUF* is high and *C* is low, where we recommend SASP. These recommendations differ from ones in the previous literature. First, MINSLK is conspicuously absent. Second, the previous studies that have recommended TWK-LST, MINWCS, or SASP have not qualified their recommendations by objective or situation, which we do.

Table 11: Summary of results for R5 (α = 0.05)*

		Resource Distribution					
		Front-loaded ***NARLF* = -3 & -2**		**Not front- or back-loaded** ***NARLF* = -1 & 0 & 1**		**Back-loaded** ***NARLF* = 2 & 3**	
		C		***C***		***C***	
Resource Contention		**HHH**	**LLL**	**HHH**	**LLL**	**HHH**	**LLL**
	Low **(*MAUF* = 0.6-0.8)**	LALP MINSLK MINWCS MAXSP MOF	LALP MS MCS MINSLK MINWCS	LALP MINSLK MINWCS MOF MAXSP	LALP MINWCS MINSLK MAXSP	LALP MOF MINSLK MAXSP MINWCS	LALP MINWCS MINSLK MAXSP
	Medium **(*MAUF* = 1-1.2)**	MINWCS LALP MOF MINSLK	MS MINWCS MCS	MINWCS LALP MINSLK MOF	MINWCS MS MCS	MINWCS MOF LALP	MINWCS MS MCS
	High **(*MAUF* = 1.4-1.6)**	MINWCS LALP MOF MINSLK	MS MCS MINWCS	MINWCS	MS MCS MINWCS	MINWCS MOF	MINWCS MS MCS EDDF MINLFT FCFS

*Multiple entries in each cell are listed in order of increasing means (i.e., the best rule is listed first).

To benefit from our results and recommendations as summarized in Tables 10 and 11, managers must be able to characterize their projects in terms of complexity (*C*), amount of resource contention (*MAUF*), and resource distribution (*NARLF*). Thankfully, *our results remain beneficial even when managers are not able to do this exactly*. First, regarding complexity, managers can qualitatively estimate whether they are dealing with a high-complexity situation or a low-complexity one without having to precisely obtain a numerical estimate. A qualitative measure of portfolio complexity may be obtained by simply asking whether a large portion of the constituent projects are highly sequential or parallel. In a parallel project, many activities can be performed concurrently, while in a sequential project fewer activities can be performed concurrently. Similarly, high-complexity projects contain a much greater number of dependencies (precedence constraints). Thus, several indicators can help a manager rate complexity as qualitatively "high" or "low." Second, the general distribution of resources (front- or back-loading) can be ascertained without too much effort. Third, the rough amount of resource contention can be qualitatively estimated to be "low," "medium," or "high." Hence, these results are readily applicable to practical issues facing project and portfolio managers.

In our estimation, the relative robustness of certain priority rules across appreciable ranges of *NARLF*, *MAUF*, and *C* is a cause for optimism. Since the effort to build a comprehensive activity network model for a new project can be daunting or prohibitive, and since such a model, if built, would include what could be highly questionable assumptions about the project's activity content and precedence relationships,[27] it is difficult to experiment with various priority rules or apply more sophisticated meta-heuristics. However, if a manager can do some rough characterization of a few key project and problem attributes, some helpful guidance on activity prioritization is now available nonetheless.

9. Summary & Conclusion

Multi-project management is becoming ever-more important in contemporary practice. Decisions about which activities to do when (based on resource allocations) have a tremendous effect on project completion times. Yet, many project managers, who often do not have an activity network model to which they might apply more advanced techniques, make resource allocation decisions based on intuition or "rules of thumb" such as MINSLK.

In the context of the static RCMPSP, this paper first contributes five new or modified measures. These measures include modified versions of standard ones such as network complexity (*C*), modified average utilization factor (*MAUF*), and normalized average resource load factor (*NARLF*), plus two new measures, the *MAUF* and *NARLF* variances. Second, this paper notes a number of observations, some of which are relatively intuitive and others which may be less so. Third, the paper comprehensively investigates the performance of 20 popular priority rules in light of the new and modified measures. For the study, we generated 12,320 project portfolios (each consisting of three projects) according to a full factorial experiment that included four factors at various levels. Our analysis confirms that the superiority of certain decision rules depends on the situation and the objective and provides clearer managerial guidance. Fourth, the paper explicitly distinguishes the project and portfolio manager perspectives.

From an individual project manager's perspective (R3), TWK-LST performs well under high network complexity, while SASP performs well under low complexity. From a portfolio manager's perspective (R5), MINWCS performs well regardless of complexity. While exhibiting a significant effect, the *MAUF* and *NARLF* variances do not change the choice of the best rule. Accordingly, we developed a decision table to guide managers in choosing among best priority rules based on *MAUF*, *NARLF*, and *C*, which constitutes a significant

[27] Since a project is doing something new for the first time, and especially in the case of projects such as new product development, there is a large amount of ambiguity in the work to be done and its relationships to other work. Thus, in such a project it can be dangerous to put too much stock in any particular activity network model.

extension to the results reported by K&D (1982) and related studies. These results show how different objectives for individual project managers and portfolio managers can lead to preferences for different decision rules and thus organizational tensions.

Importantly, our analysis shows that previously published results are not generally accurate, since *widely-advocated rules such as MINSLK, SASP, and MAXTWK did not perform well except under limited conditions*. On the other hand, our study confirms the superiority of MINWCS (Kolisch 1996a) and TWK-LST (Lova and Tormos 2001). Since we did not invent new priority rules for this study, it was inevitable that we would confirm certain rules as superior and others as inferior, agreeing with some prior studies while disagreeing with others. However, even the prior studies with which we agree did not caveat their recommendations by problem characteristics, nor did they compare them with many other priority rules. We therefore provide greater generalization and specification to particular results from prior studies.

While project scheduling, priority rules, and related topics have been studied for at least fifty years, resulting in a myriad of papers, it is in some ways astounding that no firmer guidance has appeared for decision makers in a MP context with limited resources, the most realistic situation in contemporary practice. Thus, explaining the conditions under which certain rules perform well (or poorly) is an important contribution that allows managers to sift through the conflicting results in the literature. Distinguishing the project and portfolio manager perspectives is also important in practice. In short, these results should be immediately applicable in practical situations. While we looked at a static (rather than a dynamic) case of identical project start times, our approach could be re-applied on a rolling-horizon basis in a dynamic environment.

Future research could expand our study to include additional priority rules, explore other RCMPSP formulations (such as with preemption, stochastic activity durations, or dynamic project arrivals) in a similarly comprehensive study, or explore the performance gaps between priority rules and more advanced heuristics. The gap to a lower bound or to the optimal solution for the problems would also be interesting: while solving 36,960 project networks optimally would be quite daunting, it would allow a determination of the difficulty of the test problems. The results reported in this study can also be used for the development of improved priority rules that take advantage of the superiority of certain rules under specific conditions. For instance, one might develop an adaptive priority rule that shifts between simpler rules as a project or portfolio progresses and its circumstances change.

References

Akpan, E.O.P. 2000. Priority Rules in Project Scheduling: A Case for Random Activity Selection. *Production Planning & Control*, **11**(2): 165-170.

Ash, R. 2002. Serial and Multi-Project Scheduling with and without Preemption. *Proc. of the Annual Meeting of the Decision Sciences Institute*, San Diego, CA, Nov..

Baker, K. 1974. *Introduction to Sequencing and Scheduling*. Wiley, New York.

Bedworth, D.D. and J.E. Bailey. 1987. *Integrated Production Control Systems*, 2nd Ed. Wiley, New York.

Bock, D. and J. Patterson. 1990. A Comparison of Due Date Setting, Resource Assignment, and Job Preemption Heuristics for the Multi-Project Scheduling Problem. *Decision Sci.*, **21**(3): 387-402.

Boctor, F.F. 1990. Some Efficient Multi-heuristic Procedures for Resource-constrained Project Scheduling. *Eur. J. of Op. Res.*, **49**(1): 3-13.

Bouleimen, K. and H. Lecocq. 2000. Multi-Objective Simulated Annealing for the Resource-Constrained Multi-Project Scheduling Problem. *Proc. of the 7th Int. Workshop on Project Management and Scheduling (PMS 2000)*, Osnabrück, Germany, Apr 17-19.

Bouleimen, K. and H. Lecocq. 2003. A New Efficient Simulated Annealing Algorithm for the Resource-Constrained Project Scheduling Problem and Its Multiple Modes Version. *Eur. J. of Op. Res.*, **149**(2): 268-281.

Browning, T.R. and A.A. Yassine. 2008. A Random Generator of Resource-Constrained Multi-Project Network Problems, TCU Neeley School of Business, Working Paper.

Brucker, P., *et al.* 1999. Resource-Constrained Project Scheduling: Notation, Classification, Models, and Methods. *Eur. J. of Op. Res.*, **112**(1): 3-41.

Chase, R.B., F.R. Jacobs and N.J. Aquilano. 2006. *Operations Management for Competitive Advantage*, 11th Ed. McGraw-Hill/Irwin, New York.

Chen, V.Y.X. 1994. A 0-1 Programming Model for Scheduling Multiple Maintenance Projects at a Copper Mine. *Eur. J. of Op. Res.*, **76**(1): 176-191.

Chiu, H.N. and D.M. Tsai. 1993. A Comparison of Single-project and Multi-project Approaches in Resource Constrained Multi-project Scheduling Problems. *Journal of the Chinese Institute of Industrial Engineers*, **10**: 171-179.

Cohen, I., A. Mandelbaum and A. Shtub. 2004. Multi-Project Scheduling and Control: A Process-Based Comparative Study of the Critical Chain Methodology and Some Alternatives. *Project Mgmt. J.*, **35**(2): 39-50.

Confessore, G., S. Giordani and S. Rismondo. 2007. A Market-based Multi-Agent System Model for Decentralized Multi-Project Scheduling. *Annals of Operations Research*, **150**(1): 115-135.

Cooper, D.F. 1976. Heuristics for Scheduling Resource-Constrained Projects: An Experimental Investigation. *Management Sci.*, **22**(11): 1186-1194.

Davis, E.W. 1975. Project Network Summary Measures and Constrained Resource Scheduling. *IIE Trans.*, **7**(2): 132-142.

Davis, E.W. and J.H. Patterson. 1975. A Comparison of Heuristic and Optimum Solutions in Resource-Constrained Project Scheduling. *Management Sci.*, **21**(8): 944-955.

Deckro, R.F., E.P. Winkofsky, J.E. Hebert and R. Gagnon. 1991. A Decomposition Approach to Multi-project Scheduling. *Eur. J. of Op. Res.*, **51**(1): 110-118.

Demeulemeester, E. and W. Herroelen. 1992. A Branch-and-Bound Procedure for the Multiple Resource-Constrained Project Scheduling Problem. *Management Sci.*, **38**(12): 1803-1818.

Demeulemeester, E. and W. Herroelen. 1997. New Benchmark Results for the Resource-Constrained Project Scheduling Problem. *Management Sci.*, **43**(11): 1485-1492.

Doersch, R.H. and J.H. Patterson. 1977. Scheduling a Project to Maximize Its Present Value: A Zero-One Programming Approach. *Management Sci.*, **23**(8): 882-889.

Dumond, J. and V.A. Mabert. 1988. Evaluating Project Scheduling and Due Date Assignment Procedures: An Experimental Analysis. *Management Sci.*, **34**(1): 101-118.

Elmaghraby, S.E. 1993. Resource Allocation via Dynamic Programming in Activity Networks. *Eur. J. of Op. Res.*, **64**(2): 199-215.

Elmaghraby, S.E. and W.S. Herroelen. 1980. On the Measurement of Complexity in Activity Networks. *Eur. J. of Op. Res.*, **5**(4): 223-234.

Elmaghraby, S.E. and W.S. Herroelen. 1990. The Scheduling of Activities to Maximize the Net Present Value of Projects. *Eur. J. of Op. Res.*, **49**(1): 35-49.

Fendley, L.G. 1968. Towards the Development of a Complete Multiproject Scheduling System. *Journal of Industrial Engineering*, **19**(10): 505-515.

Gibbons, J.D. 1993. *Nonparametric Statistics*. Sage Publications, Newbury Park, CA.

Gonçalves, J.F., J.J.d.M. Mendes and M.G.C. Resende. 2004. A Genetic Algorithm for the Resource Constrained Multi-Project Scheduling Problem, AT&T Labs, Technical Report TD-668LM4.

Hartmann, S. and R. Kolisch. 2000. Experimental Evaluation of State-of-the-Art Heuristics for the Resource-Constrained Project Scheduling Problem. *Eur. J. of Op. Res.*, **127**(2): 394-407.

Herroelen, W. and B.D. Reyck. 1999. Phase Transitions in Project Scheduling. *J. of the Oper. Res. Society*, **50**(2): 148-156.

Herroelen, W.S. 2005. Project Scheduling - Theory and Practice. *Production and Ops. Mgmt.*, **14**(4): 413-432.

Homberger, J. 2007. A Multi-agent System for the Decentralized Resource-constrained Multi-project Scheduling Problem. *International Transactions in Operational Research*, **14**(6): 565-589.

Kim, K.W.*, et al.* 2005. Hybrid Genetic Algorithm with Adaptive Abilities for Resource-constrained Multiple Project Scheduling. *Computers in Industry*, **56**(2): 143-160.

Kim, S.-Y. and R.C. Leachman. 1993. Multi-Project Scheduling with Explicit Lateness Costs. *IIE Trans.*, **25**(2): 34-44.

Kolisch, R. 1996a. Efficient Priority Rules for the Resource-Constrained Project Scheduling Problem. *J. of Operations Mgmt.*, **14**(3): 179-192.

Kolisch, R. 1996b. Serial and Parallel Resource-Constrained Project Scheduling Methods Revisited: Theory and Computation. *Eur. J. of Op. Res.*, **90**(2): 320-333.

Kolisch, R. 1999. Resource Allocation Capabilities of Commercial Project Management Software Packages. *Interfaces*, **29**(4): 19-31.

Kolisch, R. and S. Hartmann. 1999. "Heuristic Algorithms for Solving the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis" in Weglarz, J., Ed. *Project scheduling*, Kluwer Academic Publishers, Boston, 147-178.

Kolisch, R. and S. Hartmann. 2005. Experimental Investigation of Heuristics for Resource-Constrained Project Scheduling: An Update. *Eur. J. of Op. Res.*, **174**(1): 23-37.

Kolisch, R., A. Sprecher and A. Drexl. 1995. Characterization and Generation of a General Class of Resource-Constrained Project Scheduling Problems. *Management Sci.*, **41**(10): 1693-1703.

Kumanan, S., G.J. Jose and K. Raja. 2006. Multi-project Scheduling using an Heuristic and a Genetic Algorithm. *International Journal of Manufacturing Technology*, **31**(3-4): 360-366.

Kurtulus, I. 1978. *An Analysis of Scheduling Rules for Multi-Project Scheduling*, Ph.D. Thesis (Business), University of North Carolina at Chapel Hill, Chapel Hill.

Kurtulus, I. 1985. Multiproject Scheduling: Analysis of Scheduling Strategies under Unequal Delay Penalties. *J. of Operations Mgmt.*, **5**(3): 291-307.

Kurtulus, I. and E.W. Davis. 1982. Multi-Project Scheduling: Categorization of Heuristic Rules Performance. *Management Sci.*, **28**(2): 161-172.

Kurtulus, I.S. and S.C. Narula. 1985. Multi-Project Scheduling: Analysis of Project Performance. *IIE Trans.*, **17**(1): 58-65.

Lawrence, S.R. and T.E. Morton. 1993. Resource-Constrained Multi-Project Scheduling with Tardy Costs: Comparing Myopic, Bottleneck, and Resource Pricing Heuristics. *Eur. J. of Op. Res.*, **64**(2): 168-187.

Lenstra, J.K. and A.H.G.R. Kan. 1978. Complexity of Scheduling under Precedence Constraints. *Ops. Res.*, **26**(1): 22-35.

Liberatore, M.J. and B. Pollack-Johnson. 2003. Factors Influencing the Usage and Selection of Project Management Software. *IEEE Trans. on Eng. Mgmt.*, **50**(2): 164-174.

Linyi, D. and L. Yan. 2007. A Particle Swarm Optimization for Resource-Constrained Multi-Project Scheduling Problem. *Proc. of the International Conference on Computational Intelligence and Security*, Harbin, China, Dec 15-19.

Lova, A. and P. Tormos. 2001. Analysis of Scheduling Schemes and Heuristic Rules Performance in Resource-Constrained Multi-project Scheduling. *Annals of Operations Research*, **102**(1-4): 263-286.

Lova, A. and P. Tormos. 2002. Combining Random Sampling and Backward-Forward Heuristics for Resource-Constrained Multi-Project Scheduling. *Proc. of the 8th Int. Workshop on Project Mgmt. and Scheduling*, Valencia, Spain, Apr. 3-5.

Maroto, C., P. Tormos and A. Lova. 1999. "The Evolution of Software Quality in Project Scheduling" in Weglarz, J., Ed. *Project Scheduling*, Kluwer Academic Publishers, Boston, 239-259.

Meredith, J.R. and S.J. Mantel. 2006. *Project Management*, 6th Ed. Wiley, New York.

Mingozzi, A., V. Maniezzo, S. Ricciardelli and L. Bianco. 1998. An Exact Algorithm for the Resource-Constrained Project Scheduling Problem Based on a New Mathematical Formulation. *Management Sci.*, **44**(5): 714-729.

Mize, J.H. 1964. *A Heuristic Scheduling Model for Multi-Project Organizations*, Ph.D. Thesis Purdue University.

Mohanty, R.P. and M.K. Siddiq. 1989. Multiple Projects - Multiple Resources-Constrained Scheduling: Some Studies. *Int. J. of Production Res.*, **27**(2): 261-280.

Möhring, R.H. 1984. Minimizing Costs of Resource Requirements in Project Networks Subject to a Fixed Completion Time. *Operations Res.*, **32**(1): 89-120.

Neumann, K. and J. Zimmermann. 1999. Resource Levelling for Projects with Schedule-dependent Time Windows. *Eur. J. of Op. Res.*, **117**(3): 591-605.

Özdamar, L. and G. Ulusoy. 1995. A Survey on the Resource-constrained Project Scheduling Problem. *lIE Transactions*, **27**(5): 574-586.

Pascoe, T.L. 1966. Allocation of Resources - CPM. *Revue Française de Recherche Opérationelle*, **38**: 31-38.

Patterson, J.H. 1973. Alternative Methods of Project Scheduling with Limited Resources. *Naval Research Logistics Quarterly*, **20**(4): 767-784.

Patterson, J.H. and G.W. Roth. 1976. Scheduling a Project under Multiple Resource Constraints: A Zero-One Programming Approach. *AIIE Transactions*, **8**: 449-455.

Payne, J.H. 1995. Management of Multiple Simultaneous Projects: A State-of-the-Art Review. *Int. J. of Project Mgmt.*, **13**(3): 163-168.

Pritsker, A.A.B., L.J. Watters and P.M. Wolfe. 1969. Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach. *Management Sci.*, **16**(1): 93-108.

Sprecher, A. 2000. Scheduling Resource-Constrained Projects Competitively at Modest Resource Requirements. *Management Sci.*, **46**(5): 710-723.

Talbot, F.B. 1982. Resource-Constrained Project Scheduling with Time-Resource Tradeoffs: The Nonpreemptive Case. *Management Sci.*, **28**(10): 1197-1210.

Thomas, P.R. and S. Salhi. 1997. An Investigation into the Relationship of Heuristic Performance with Network-Resource Characteristics. *Journal of the Operational Research Society*, **48**(1): 34-43.

Tseng, C.-C. 2004. Multiple Projects Scheduling with Multiple Modes: A Genetic Algorithm. *Proc. of the 1st ORSTW Conference on Technology and Management*, Taipei.

Tsubakitani, S. and R.F. Deckro. 1990. A Heuristic for Multi-Project Scheduling with Limited Resources in the Housing Industry. *Eur. J. of Op. Res.*, **49**(1): 80-91.

Ulusoy, G. and L. Özdamar. 1989. Heuristic Performance and Network/Resource Characteristics in Resource-constrained Project Scheduling. *Journal of the Operational Research Society*, **40**(12): 1145-1152.

Vercellis, C. 1994. Constrained Multi-project Planning Problems: A Lagrangean Decomposition Approach. *Eur. J. of Op. Res.*, **78**(2): 267-275.

Woodworth, B.M. and C.J. Willie. 1975. A Heuristic Algorithm for Resource Leveling in Multi-project, Multi-resource Scheduling. *Decision Sci.*, **6**(3): 525-540.

Yang, K.-K. and C.-C. Sum. 1993. A Comparison of Resource Allocation and Activity Scheduling Rules in a Dynamic Multi-Project Environment. *J. of Operations Mgmt.*, **11**(2): 207-218.

Yang, K.-K. and C.-C. Sum. 1997. An Evaluation of Due Date, Resource Allocation, Project Release, and Activity Scheduling Rules in a Multiproject Environment. *Eur. J. of Op. Res.*, **103**: 139-154.

Appendix: ANOVA Models

ANOVA Model 1: *NARLF, MAUF, C,* σ^2_{MAUF}, and Priority Rule

Factor	Type	Levels	Values
NARLF	fixed	7	-3, -2, -1, 0, 1, 2, 3
MAUF	fixed	11	0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6
C	fixed	4	HHH, HHL, HLL, LLL
MAUF_Var	fixed	2	0.00, 0.25
Priority_Rule	fixed	20	EDDF, FCFS, LALP, LCFS, MAXSLK, MAXSP, MAXTWK, MCS, MINLFT, MINSLK, MINTWK, MINWCS, MOF, MS, RAN, SASP, SOF, TWK-EST, TWK-LST, WACRU

R3:

Source	DF	Seq SS	Adj SS	Adj MS	F	P
C	3	5679615	5679615	1893205	8463.42	0.000
Priority_Rule	19	6065519	6065519	319238	1427.13	0.000
MAUF_Var	1	7001390	7001390	7001390	31299.15	0.000
NARLF	6	730571	730571	121762	544.33	0.000
MAUF	10	226116909	226116909	22611691	101083.73	0.000
Error	246360	55108929	55108929	224		
Total	246399	300702933				

S = 14.9564 **R-Sq = 81.67% R-Sq(adj) = 81.67%**

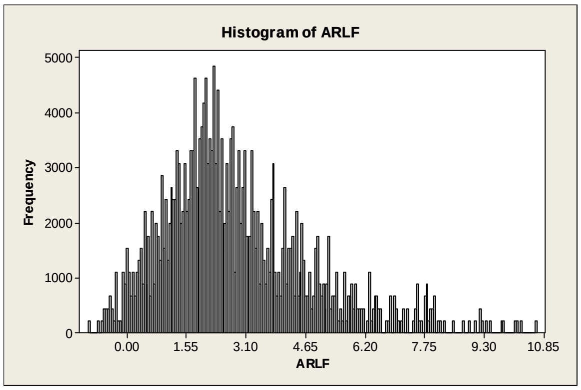
R5:

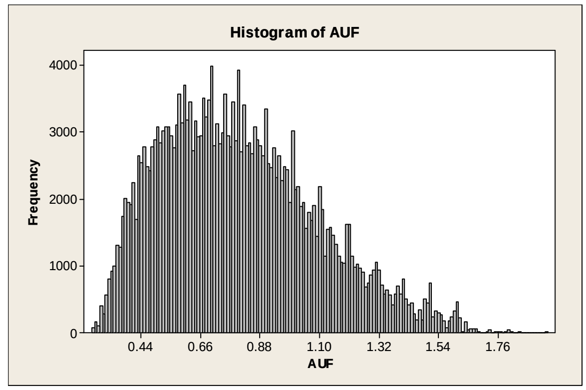
Source	DF	Seq SS	Adj SS	Adj MS	F	P
C	3	774517	774517	258172	2735.35	0.000
Priority_Rule	19	5115549	5115549	269239	2852.61	0.000
MAUF_Var	1	8636518	8636518	8636518	91504.45	0.000
NARLF	6	19741	19741	3290	34.86	0.000
MAUF	10	241235791	241235791	24123579	255590.86	0.000
Error	246360	23252338	23252338	94		
Total	246399	279034453				

S = 9.71512 **R-Sq = 91.67% R-Sq(adj) = 91.67%**

ANOVA Model 2: *ARLF, AUF, C,* σ^2_{MAUF}, and Priority Rule

N.B.: In generating our test problems, we controlled for *NARLF* and *MAUF*, but it is not possible to specify *ARLF* and *AUF* in the same set of problems. While we can calculate the *ARLF* and *AUF* for each test problem, "they are what they are." Because of the biases in the *ARLF* measure, in particular, our test problems with *NARLF*s uniformly distributed over [-3, -2, -1, 0, 1, 2, 3] had *ARLF*s distributed as shown in the following figure. As shown in the second figure, the *AUF*s fell mainly in the same range as the *MAUF*s, although the distribution is not uniform over [0.6, 0.7, 0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6]. Therefore, we assigned each test problem to one of seven groups (levels) according to its *ARLF*, assigning problems with *ARLF* < 0.5 to level 0, problems with 0.5 ≤ *ARLF* < 1.5 to level 1, etc., and *ARLF* ≥ 5.5 to level 6. We assigned each test problem to one of 11 levels according to its *AUF* using a similar approach. Thus, while the number of factors and levels is identical to ANOVA Model 1, the number of data points in each level is not. Nevertheless, this is the best possible comparison of this type that we have found.





Factor	Type	Levels	Values
ARLF	fixed	7	0, 1, 2, 3, 4, 5, 6
AUF	fixed	11	0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6
C	fixed	4	HHH, HHL, HLL, LLL
MAUF_Var	fixed	2	0.00, 0.25
Priority_Rule	fixed	20	EDDF, FCFS, LALP, LCFS, MAXSLK, MAXSP, MAXTWK, MCS, MINLFT, MINSLK, MINTWK, MINWCS, MOF, MS, RAN, SASP, SOF, TWK-EST, TWK-LST, WACRU

R3:

Source	DF	Seq SS	Adj SS	Adj MS	F	P
C	3	5679615	265451	88484	206.28	0.000
Priority_Rule	19	6065519	6065519	319238	744.23	0.000
MAUF_Var	1	7001390	5801041	5801041	13523.74	0.000
ARLF	6	330054	1491887	248648	579.66	0.000
AUF	10	175949658	175949658	17594966	41018.46	0.000
Error	246360	105676696	105676696	429		
Total	246399	300702933				

S = 20.7112   **R-Sq = 64.86%**   R**-Sq(adj) = 64.85%**

R5:

Source	DF	Seq SS	Adj SS	Adj MS	F	P
C	3	774517	2859828	953276	2409.75	0.000
Priority_Rule	19	5115549	5115549	269239	680.60	0.000
MAUF_Var	1	8636518	7337229	7337229	18547.50	0.000
ARLF	6	18126	1194117	199019	503.09	0.000
AUF	10	167031888	167031888	16703189	42223.35	0.000
Error	246360	97457856	97457856	396		
Total	246399	279034453				

S = 19.8895   **R-Sq = 65.07%**   **R-Sq(adj) = 65.07%**

ANOVA Model 3: Full Model with Interactions (no *ARLF* or *AUF*)

Factor	Type	Levels	Values
NARLF	fixed	7	-3, -2, -1, 0, 1, 2, 3
MAUF	fixed	11	0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6
C	fixed	4	HHH, HHL, HLL, LLL
MAUF_Var	fixed	2	0.00, 0.25
Priority_Rule	fixed	20	EDDF, FCFS, LALP, LCFS, MAXSLK, MAXSP, MAXTWK, MCS, MINLFT, MINSLK, MINTWK, MINWCS, MOF, MS, RAN, SASP, SOF, TWK-EST, TWK-LST, WACRU

R3:

Source	DF	SS	MS	F	P
Complexity	3	5676637	1892212	10172.87	0.000
NARLF	6	730977	121829	654.98	0.000
MAUF	10	226106220	22610622	121558.76	0.000
MAUF_Var	1	7001496	7001496	37641.30	0.000
Priority_Rule	19	6065896	319258	1716.39	0.000
Complexity*NARLF	18	405170	22509	121.01	0.000
Complexity*MAUF	30	3076039	102535	551.24	0.000
Complexity*MAUF_Var	3	31672	10557	56.76	0.000
Complexity*Priority_Rule	57	1031467	18096	97.29	0.000
NARLF*MAUF	60	438699	7312	39.31	0.000
NARLF*MAUF_Var	6	35560	5927	31.86	0.000
NARLF*Priority_Rule	114	166361	1459	7.85	0.000
MAUF*MAUF_Var	10	2143448	214345	1152.36	0.000
MAUF*Priority_Rule	190	1960751	10320	55.48	0.000
MAUF_Var*Priority_Rule	19	95391	5021	26.99	0.000
Error	245853	45730060	186		
Total	246399	300695843			

S = 13.6384 **R-Sq = 84.79%** **R-Sq(adj) = 84.76%**

R5:

Source	DF	SS	MS	F	P
Complexity	3	774555	258185	3607.00	0.000
NARLF	6	19749	3291	45.98	0.000
MAUF	10	241218951	24121895	336997.70	0.000
MAUF_Var	1	8634599	8634599	120630.65	0.000
Priority_Rule	19	5115274	269225	3761.24	0.000
Complexity*NARLF	18	103962	5776	80.69	0.000
Complexity*MAUF	30	1159921	38664	540.16	0.000
Complexity*MAUF_Var	3	138260	46087	643.86	0.000
Complexity*Priority_Rule	57	826638	14502	202.61	0.000
NARLF*MAUF	60	39960	666	9.30	0.000
NARLF*MAUF_Var	6	47134	7856	109.75	0.000
NARLF*Priority_Rule	114	41161	361	5.04	0.000
MAUF*MAUF_Var	10	2607946	260795	3643.46	0.000
MAUF*Priority_Rule	190	666202	3506	48.99	0.000
MAUF_Var*Priority_Rule	19	25130	1323	18.48	0.000
Error	245853	17597866	72		
Total	246399	279017306			

S = 8.46043 **R-Sq = 93.69%** **R-Sq(adj) = 93.68%**

Nomenclature *(optional section)*

L	Number of projects (independent networks) in a multi-project problem (portfolio)
l	Project index: $1 \leq l \leq L$
N_l	Number of nodes (activities) in project network l
i	Activity index: $1 \leq i \leq N_l$
C_l	Complexity level of project l
A	Number of arcs (precedence relationships or dependencies) in a network
A'	Number of non-redundant arcs in a network, $A' \leq A$
$A'_{\min}$	Minimum number of non-redundant arcs
$A'_{\max}$	Maximum number of non-redundant arcs
d_{il}	Duration of activity i in project l
K_l	Number of types of resources used by project l
k	Resource type index: $1 \leq k \leq K_l$
K_{il}	Number of types of resources used by activity i in project l
r_{ilk}	Amount of resource type k required by activity i in project l
S	Number of time intervals spanning a problem
s	Interval index: $1 \leq s \leq S$
S_s	Length of interval s
R_k	Renewable amount of resource type k available in each time interval
t	Time period index
X_{ilt}	Boolean variable, true (equal to 1) if activity i of project l is active at time t
CP_l	Non-resource-constrained critical path duration of project l
Z_{ilt}	Equal to -1 if the part of activity i of project l is active at time $t \leq CP_l / 2$; otherwise equal to 1
$ARLF_l$	Average resource loading factor for project l
$NARLF$	Normalized average resource loading factor for problem
$NARLF_{des}$	Desired $NARLF$ setting for a problem
σ^2_{NARLF}	Variance in projects' $ARLF$ values from problem's $NARLF$
AUF_k	Average utilization factor for resource k
$MAUF_k$	Modified average utilization factor for resource k
$MAUF$	$MAUF$ for a problem; $MAUF = \text{Max}(MAUF_1, ..., MAUF_K)$
$MAUF_{des}$	Desired $MAUF$ setting for a problem
$MAUF_{des,k}$	Desired $MAUF$ setting for resource k
σ^2_{MAUF}	Variance in resources' $MAUF$ values from problem's $MAUF$
$\sigma^2_{MAUF,des}$	Desired $MAUF$ variance
P_{il}	Set of all immediate predecessors of activity i in project l
$\hat{i}$	An element in P_{il}