ARTICLE IN PRESS

Int. J. Production Economics ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Int. J. Production Economics

journal homepage: www.elsevier.com/locate/ijpe

Resource-constrained multi-project scheduling: Priority rule performance revisited

Tyson R. Browning [a,*], Ali A. Yassine [b]

[a] Neeley School of Business, Texas Christian University, TCU Box 298530, Fort Worth, TX 76129, USA
[b] Engineering Management Program, American University of Beirut, Beirut, Lebanon

ARTICLE INFO

Article history:
Received 20 May 2009
Accepted 17 March 2010

Keywords:
Project management
Multi-project scheduling
Resource constraints
Priority rules

ABSTRACT

Managers of multiple projects with overly constrained resources face difficult decisions in how to allocate resources to minimize the average delay per project or the time to complete the whole set of projects. We address the static *resource-constrained multi-project scheduling problem* (RCMPSP) with two lateness objectives, project lateness and portfolio lateness. In this context, past research has reported conflicting results on the performance of activity priority rule heuristics and does not provide managers with clear guidance on which rule to use in various situations. Using recently improved measures for RCMPSP characteristics, we conducted a comprehensive analysis of 20 priority rules on 12,320 test problems generated to the specifications of project-, activity-, and resource-related characteristics—including network complexity and resource distribution and contention. We found several situations in which widely advocated priority rules perform poorly. We also confirmed that portfolio managers and project managers will prefer different priority 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.

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. The goal is to prioritize activities so as to optimize an objective function such as minimizing the delay to each project or the overall portfolio. Such is the basic *resource-constrained multi-project scheduling problem* (RCMPSP). According to Payne (1995), up to 90% of the value of all projects accrues in a multi-project context, so the impact of even a small improvement in their management could provide an enormous benefit.

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). In this paper, we take the second approach, because (1) it is more realistic, (2) it has received less attention in past research, (3) it presents a greater opportunity for improvement (Herroelen, 2005), and critically, (4) the existing 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. *Priority rule* (PR) heuristics are crucial for several reasons: (1) meta-heuristics' improved performance comes at greater computational expense, meaning that PRs are necessary for very large problems (Kolisch, 1996a); (2) PRs are a component of other (local search-based and sampling) heuristics (Kolisch, 1996b) and "are indispensible" for constructing initial solutions for meta-heuristics (Hartmann and Kolisch, 2000); and (3) PRs are used extensively by commercial project scheduling software due to their speed and simplicity (Herroelen, 2005). However, perhaps

* Corresponding author. Tel.: +1 817 257 5069.
E-mail addresses: t.browning@tcu.edu (T.R. Browning), ali.yassine@aub.edu.lb (A.A. Yassine).

0925-5273/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.ijpe.2010.03.009

Nomenclature

L	number of projects (independent networks) in a multi-project problem (portfolio)
l	project index: $1 \le l \le L$
N_l	number of nodes (activities) in project network l
i	activity index: $1 \le i \le N_l$
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, $A' \le A$
$A'_{\min}$	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: $1 \le k \le K_l$
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: $1 \le s \le S$
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 $t \le CP_l/2$; 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
$\sigma^2{}_{NARLF}$	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; $MAUF=\mathrm{Max}(MAUF_1,\ldots, MAUF_K)$
$MAUF_{des}$	desired $MAUF$ setting for a problem
$MAUF_{des,k}$	desired $MAUF$ setting for resource k
$\sigma^2{}_{MAUF}$	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
$\widehat{i}$	an element in P_{il}

the most important argument for PRs is that they are very important in practice. For a variety of reasons, most project managers do not (or cannot) actually build the formal activity network models which are prerequisites to the application of meta-heuristics. That is, it does not matter how large an activity network that a modern computer can process optimally or with meta-heuristics if a project cannot or does not invest in the effort to construct such a network! 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 PR to use is discussed in many contemporary project management textbooks (e.g., Meredith and Mantel, 2009), 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 PRs. Moreover, these studies have presented conflicting results on PR performance, because this varies based on portfolio, project, resource, and activity characteristics. Thus, a more comprehensive study of a larger set of PRs 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. Based on the salient characteristics of the RCMPSP in terms of its constituent projects, activities, and resources, we use five recent measures of RCMPSP characteristics to define a multi-dimensional problem space. Then, 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 PRs. We find significant differences in the performance of the PRs—implying that the choice of PR does indeed matter—and that several widely advocated PRs generally do not perform well. Finally, we organize these results for managers, distinguishing the project and portfolio management perspectives.

The rest of the paper is organized as follows. After stating the mathematical formulation of the basic RCMPSP (Section 2) and reviewing related literature (Section 3), in Section 4 we discuss characteristics of the RCMPSP, including objective functions, network characteristics, and resource characteristics. Section 5 presents our study, and Section 6 distills its implications for managers. Section 7 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 $(F_{1l},\ldots, F_{il},\ldots,F_{N_l l})$. Let $\boldsymbol{A}_t$ 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, $\widehat{i} \in P_{il}$. With these definitions, the problem can be formally stated as

$$\text{Optimize}: \text{Performance measure } (\forall i \in N_l,\ l \in L : F_{1l},\ldots,F_{il},\ldots,F_{N_l l}) \tag{1}$$

$$\text{Subject to}: \forall i \in N_l, \quad \widehat{i} \in P_{il}, \quad l \in L: \ F_{\widehat{i}l} \le F_{il} - d_{il} \tag{2}$$

$$\forall i,l \in \boldsymbol{A_t}: \sum_{i,l \in A(t)} r_{ilk} \le R_k k \in K, \quad t \ge 0 \tag{3}$$

$$\forall i \in N_l, \quad l \in L: \ F_{il} \ge 0 \tag{4}$$

Please cite this article as: Browning, T.R., Yassine, A.A., Resource-constrained multi-project scheduling: Priority rule performance revisited. International Journal of Production Economics (2010), doi:10.1016/j.ijpe.2010.03.009

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 (Krüger and Scholl, 2008, 2009), 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.

3. Literature review

This paper addresses the static RCMPSP and maintains the distinction between projects (the MP approach mentioned in Section 1). While an abundant amount of literature addresses the (single-project) RCPSP (several reviews are available: e.g., Brucker et al., 1999; Hartmann and Briskorn, 2009; Herroelen, 2005; Kolisch and Hartmann, 2005), the single-project approach to solving RCMPSPs has several drawbacks (Chiu and Tsai, 1993). First, it is less realistic, 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 the individual project's performance characteristics. Third, aggregating multiple projects yields very large problems.

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: PR-based X-pass heuristics, classical meta-heuristics, non-standard meta-heuristics, and miscellaneous heuristics (Kolisch and Hartmann, 1999, 2005). Classical meta-heuristics include simulated annealing (e.g., Bouleimen and Lecocq, 2000), genetic algorithms (GAs) (e.g., Gonçalves et al., 2008; Kim et al., 2005; Kumanan et al., 2006), and swarm optimization (e.g., Linyi and Yan, 2007). Non-standard meta-heuristics 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).

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

PRs can also be classified by the information they use: (a) activity-related, (b) project-related, or (c) resource-related (Kolisch, 1996a).[2] Activity-related PRs 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 PRs promote activities based on the project they belong to or characteristics of that project (e.g., shortest activity from shortest project first—SASP). Resource-related PRs 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) PR. Some PRs combine elements of information about the activity, the project, and/or the resources (Hartmann and Kolisch, 2000). For each PR addressed in our study, we note its "Basis" according to this classification in Tables 1 and 2.

Davis and Patterson (1975) noted that successful PRs generally incorporate some measure of *time* or *resource usage*, and 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 PR performance (e.g., Pascoe, 1966). Ulusoy and Özdamar (1989) suggested the following measures for distinguishing successful PRs: percentage of critical activities, network complexity and resource measures, obstruction value, and utilization factor.

While many studies have been conducted on the performance of a myriad of PRs 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 PR (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, PRs based on the MP approach perform better when minimizing the average delay in several projects (Kurtulus and Davis, 1982). RCMPSP studies have disagreed about which PR performs best and under which conditions, although MINSLK has generally performed well (Cohen et al., 2004; Davis and Patterson, 1975; Fendley, 1968). Kurtulus and Davis (1982) developed six PRs for the MP environment, and along with three single-project PRs (see the list in Table 1), analyzed these with the objective of minimizing total project delay, finding that SASP was best under most conditions. Kurtulus (1985) and Kurtulus and Narula (1985) provided further results. In summary, while various studies have identified potentially important characteristics of the RCMPSP and proposed various PRs, the variety of results and their disagreements have left project managers lacking clear guidance on which PR to use in a particular situation.

4. RCMPSP characteristics and measures

Four important characteristics of the RCMPSP—objective function, network complexity, resource distribution, and resource

[1] In a static RCMPSP environment (e.g., 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, 1997).

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

Please cite this article as: Browning, T.R., Yassine, A.A., Resource-constrained multi-project scheduling: Priority rule performance revisited. International Journal of Production Economics (2010), doi:10.1016/j.ijpe.2010.03.009

Table 1
Priority rules analyzed by Kurtulus and Davis (1982).

Priority rule (PR) (*=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 ith activity from the lth project	Random	Best in studies by Mize (1964), Dumond and Mabert (1988)[a] and Bock and Patterson (1990)[a]
2. **SOF**—shortest operation first	Activity	Min(d_{il}), where d_{il} is the duration of the ith activity from the lth project	FCFS	Best in study by Patterson (1973)
3. **MOF**—maximum (longest) operation first	Activity	Max(d_{il})	Greatest resource requirements ($GRES$), where $GRES = \text{Max}(\sum_{k=1}^{K} r_{ilk})$	
4. **MINSLK***—minimum slack	Activity	Min(SLK_{il}), where $SLK_{il}=LS_{il}-\text{Max}(ES_{il}, t)$, LS_{il} is the late start time of the ith activity from the lth project, and t is the current time step[c]	FCFS	Best in studies by Fendley (1968), Davis and Patterson (1975)[b], and Boctor (1990)[b]
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 lth project without resource constraints	FCFS	Best in studies by Kurtulus and Davis (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	Min($\sum_{k=1}^{K}\sum_{i\in AS_l} d_{il}r_{ilk}+d_{il}\sum_{k=1}^{K} r_{ilk}$), 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	Max($\sum_{k=1}^{K}\sum_{i\in AS_l} d_{il}r_{ilk}+d_{il}\sum_{k=1}^{K} r_{ilk}$)	FCFS	Best in studies by Kurtulus (1985), Kurtulus and Narula (1985), and Lova and Tormos (2001)

[a] Refers to the *dynamic* RCMPSP.
[b] Discusses a *single* project (RCPSP).
[c] 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 Section 5.1.

contention—have been identified that distinguish problem and project situations. Each requires an appropriate measure for analysis.

4.1. Objective function

A variety of objective functions have been used for the RCMPSP. Minimizing project duration (Baker, 1974) has been used most widely. Other MP objective functions include: minimize total project delay, lateness, or tardiness (Kurtulus and Davis, 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). Several studies have shown that PR performance depends on the chosen objective (e.g., 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 Fig. 1:

$$\text{Total delay} = a+b+c \tag{R1}$$

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

$$\text{Average percent delay} = \frac{(a/A)+(b/B)+(c/C)}{3} \tag{R3}$$

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

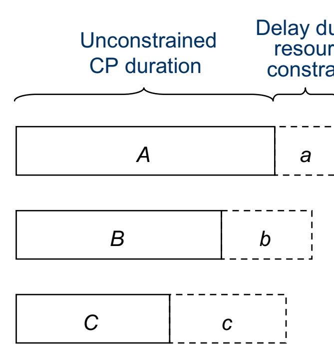

Fig. 1. Example problem composed of three projects.

$$\text{Percentage delay} = \frac{\text{Max}(A+a,\ B+b,\ C+c)-\text{Max}(A,\ B,\ C)}{\text{Max}(A,\ B,\ C)} \tag{R5}$$

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 10-day delay on a 1-day project is probably worse than a 10-day delay on a 100-day project. Furthermore, R3 and R5 represent the individual project manager's and the 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 portfolio 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.

Please cite this article as: Browning, T.R., Yassine, A.A., Resource-constrained multi-project scheduling: Priority rule performance revisited. International Journal of Production Economics (2010), doi:10.1016/j.ijpe.2010.03.009

4.2. Network complexity

Low-complexity networks are less precedence-constrained. A review of complexity measures (Browning and Yassine, 2010) suggested using an adapted form of the number of non-redundant arcs (precedence relationships), A', and nodes (activities), N, that normalizes the complexity measure, C, over [0,1]:

$$C = \frac{4A' - 4N + 4}{(N-2)^2} \tag{5}$$

The level of analysis in the network complexity literature is a single project. In the MP environment, we refrain from using a composite complexity measure such as simple averaging, because the impact of individual project complexity on the MP portfolio remains unclear: a problem with three medium-complexity projects is not necessarily the same as a problem with one high- and two low-complexity projects, even though both average to the same number. Therefore, we maintain a distinction between projects and use a vector of constituent project complexities, $\mathbf{C} = \{C_1, C_2, \ldots, C_L\}$, 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 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, Kurtulus and Davis (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 (resource unconstrained) critical path (CP) duration[3] and the relative size of the disparity. For project l, it is defined as

$$ARLF_l = \frac{1}{CP_l} \sum_{t=1}^{CP_l} \sum_{k=1}^{K_{il}} \sum_{i=1}^{N_l} Z_{ilt} X_{ilt} \left(\frac{r_{ilk}}{K_{il}} \right) \tag{6}$$

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.[4] Projects with $ARLF < 0$ are "front-loaded" in their resource requirements, while projects with $ARLF > 0$ are "back-loaded."

However, the *ARLF* measure has several problems. First, it can fall victim to the "flaw of averages" and fail to distinguish significantly different cases. For example, Fig. 2 illustrates some stylized resource distributions and their *ARLF*s. In the first row,

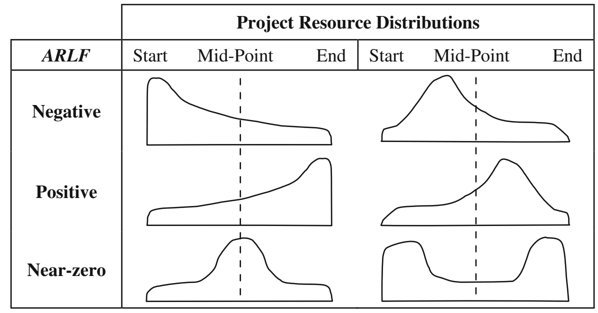

Fig. 2. Example resource distributions over project time.

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 the bottom row have $ARLF \approx 0$.

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

Second, despite its use in the MP context, *ARLF* pertains to a single project. Kurtulus and Davis (1982) define the *ARLF* for a *problem* as the average of its constituent projects' *ARLF*s:

$$ARLF = \frac{1}{L} \sum_{l=1}^{L} ARLF_l \tag{7}$$

Again, averaging sacrifices important information. Consider the following two observations.

Observation 2: As demonstrated in Fig. 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 individual *project* level. Looking only at the *ARLF* of the constituent projects may fail to distinguish between dramatically different types of problems.

Observation 3: Eq. (7) also fails to account for differences in the durations of the constituent projects. For example, consider the three projects in Fig. 4, where the first project is twice as long as the second and third projects. According to Eq. (7), this problem's $ARLF = 1.33$, even though visual analysis confirms a highly front-loaded distribution of resources and a *problem 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), Eq. (7) provides misleading results.

To address these issues, we adopt the *normalized ARLF* (*NARLF*) measure proposed by Browning and Yassine (2010) which normalizes the resource distribution over the *problem's* CP duration:

$$NARLF = \frac{1}{LCP_{Max}} \sum_{l=1}^{L} \sum_{t=1}^{CP_l} \sum_{k=1}^{K_{il}} \sum_{i=1}^{N_l} Z_{ilt} X_{ilt} \left(\frac{r_{ilk}}{K_{il}} \right) \tag{8}$$

where $CP_{max} = \mathrm{Max}(CP_1, \ldots, CP_L)$.[5] We also explored an additional measure proposed by Browning and Yassine (2010), the variance of a problem's constituent *ARLF*s from its *NARLF*:

$$\sigma^2_{ARLF} = \frac{1}{L} \sum_{l=1}^{L} (ARLF_l - NARLF)^2 \tag{9}$$

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

[4] The original definition of Z_{ilt} in Kurtulus and Davis (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.

[5] 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, we use a constant K for all projects in our experiments.

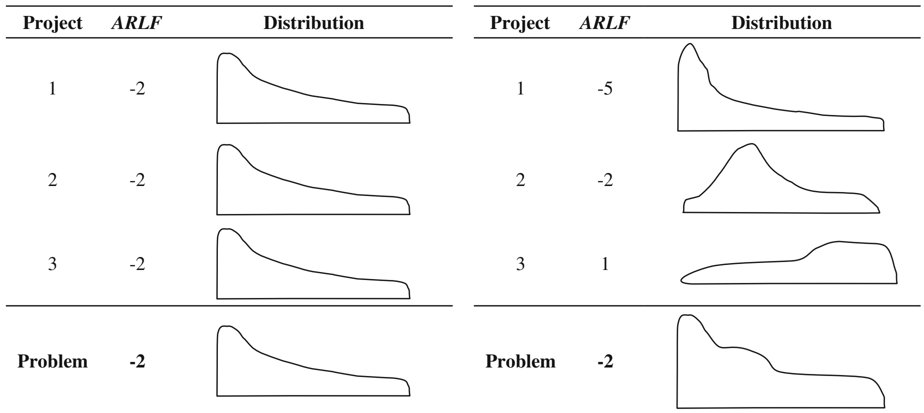

Fig. 3. Two examples of *ARLF* calculation for an overall problem (using Eq. (7)).

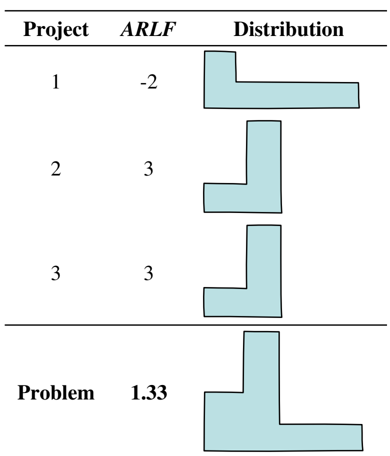

Fig. 4. Example *ARLF* calculation for a problem using Eq. (7).

4.4. Resource contention

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

Observation 4: As $|ARLF|$ and $|NARLF| \to 0$, 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 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 Fig. 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 this important difference in these problems. Interestingly, Kurtulus and Davis (1982) had trouble distinguishing the best priority rule from their experiments when $ARLF \approx 0$.[6] Therefore, we need another measure for resource contention.

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 CP duration. If $UF_k < 1 \forall k$ in each time period, then there is no resource contention. To reduce computational intensity, Kurtulus and Davis (1982) proposed averaging the *UF* over intervals to get an *average utilization factor*, *AUF*. Using Fig. 6 as an example, they proposed using $S = L = 3$ intervals, where L is the number of projects, $S_1 = CP_1$, $S_2 = CP_2 - CP_1$, and $S_3 = CP_3 - CP_2$, once the projects have been sorted from shortest to longest such that $CP_1 \le CP_2 \le CP_3$. The total amount of resource k required over any interval s is given by

$$W_{sk} = \sum_{t=a}^{b} \sum_{l=1}^{L} \sum_{i=1}^{N_l} r_{ilk} X_{ilt} \tag{10}$$

where $a = CP_{s-1} + 1$, $b = CP_s$, r_{ilk} is the amount of resource k required by the ith activity in project l, and X is defined as in Eq. (6). The *AUF* indicates the average tightness of the constraints on (i.e., the average amount of contention for) each resource type:

$$AUF_k = \frac{1}{S} \sum_{s=1}^{S} \frac{W_{sk}}{R_k S} \tag{11}$$

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:

$$AUF = \mathrm{Max}(AUF_1, AUF_2, \ldots, AUF_K) \tag{12}$$

However, Observations 5 and 6 illuminate two problems with the *AUF* measure.

Observation 5: When the projects in a problem have similar CP durations (which is not uncommon when projects are of similar size), then $S_1 \gg S_{s>1}$, and averaging over these disproportionate intervals can obscure the situation.

To ameliorate this issue, Browning and Yassine (2010) proposed averaging over equal intervals of problem time, such as the integer units indicated by the dashed, vertical lines in Fig. 6.[7] Thus, in Fig. 6's example, $S = 16 = CP_{\max}$. Browning and

[6] 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,

(*footnote continued*)
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.

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

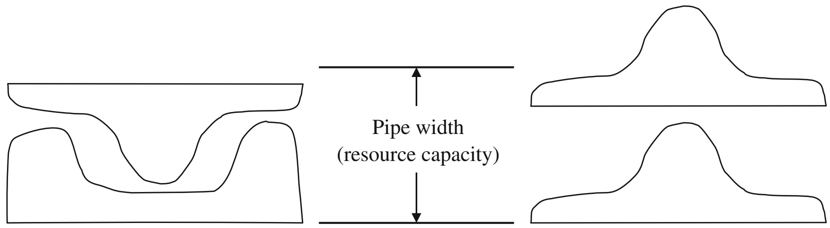

Fig. 5. Two example problems, each containing two projects with $ARLF=NARLF=0$.

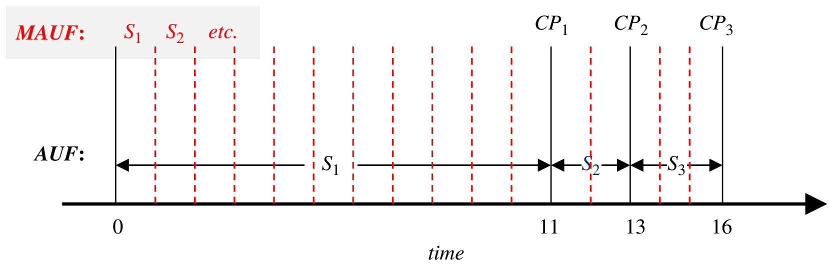

Fig. 6. Example of time intervals formed by three projects in a problem.

Yassine (2010) called this measure the *modified AUF* or *MAUF*, and Eqs. (10)–(12) hold, although S is determined differently.

Observation 6: Determining a problem's *AUF* as the maximum of its resources' *AUF*s (Eq. (12)) 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 Eq. (12). 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 amounts of resource contention yet identical *AUF*s (or *MAUF*s).

To provide a clearer picture of resource contention, we augment the *MAUF* measure with Browning and Yassine's (2010) measure of the variance in the $MAUF_k$s:

$$\sigma^2_{MAUF}=\frac{\sum_{k=1}^{K}(MAUF-MAUF_k)^2}{K} \tag{13}$$

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} should correlate with reduced problem delay.

5. Our study

5.1. Scheduling algorithm

All PR-based heuristics require a *schedule generation scheme* (SGS). Boctor (1990) distinguished between "serial" and "parallel" schemes: 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 dynamically re-determined (as necessary) at each time step. We adopt the parallel SGS, which seems to have been most widely used in MP studies.

The parallel SGS proceeds as follows. First, the overall problem duration is broken down into time steps. At each, the algorithm separates the activities into four disjoint sets: the *completed set*, ***C*** (finished activities), the *active set*, ***A*** (ongoing, "already scheduled" activities), the *decision set*, ***D*** (un-started activities that depend only on activities in ***C***), and the *ineligible* set, ***I*** (activities with predecessors in ***A*** or ***D***). 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 PR to rank the activities in ***D***. The highest-ranking activities are added to ***A*** as resources allow. The time step ends when the first 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***.

5.2. Set-up

We compiled a set of 20 popular PRs from the literature (Tables 1 and 2), 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 PRs to be FCFS.

We designed a full factorial experiment to test the influence of the factors listed in Table 3. To maximize the insights from varying the last four factors, we held the first three constant.[8] The choices for seven *NARLF* and 11 *MAUF* levels follow Kurtulus and Davis (1982). We designated two levels of project complexity, "high" ($C=0.69$) and "low" ($C=0.14$).[9] 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 had a significantly different *MAUF*. Thus, we needed $7\times11\times4\times2=616$ test problems for this experiment. Standard problem generators and test sets such as ProGen/PSPLIB (Kolisch et al., 1995) cannot create MP problems to these

[8] No specific relationship has been reported between portfolio size or project size and the solution quality obtained by the various PRs (Hartmann and Kolisch, 2000; Kurtulus and Davis, 1982; Lova and Tormos, 2001). Meanwhile, since K is used to determine *NARLF* (Eq. (8)), its variation would be confounded with *NARLF*.

[9] By Eq. (5), $C=0.69$ implies 75 non-redundant arcs among 20 activities and $C=0.14$ implies 30.

Table 2
Additional priority rules analyzed in this study.

	Priority rule (PR) (*=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$\left(w\sum_{q=1}^{N_i}(1+SLK_{iq})^{-\alpha}+(1-w)\sum_{k=1}^{K}\frac{r_{ik}}{R_{Max,k}}\right)$, where N_i is the number of immediate successors of the ith activity, w is the weight associated with N_i $(0\leq w\leq 1)$, SLK_{iq} is the slack in the qth immediate successor of the ith 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 ith activity in the lth 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 ith activity in the lth project; $CS_{il}\in TS_{il}$	

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

Table 3
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)[a]

[a] The basis for selecting the "high" variance setting is explained in Browning and Yassine (2010).

specifications, so we used a test problem generator recently developed by Browning and Yassine (2010). 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 PRs, thus producing 246,400 experimental outcomes. We specified each outcome in terms of the five objective functions in Section 4.1, thereby yielding 1,232,000 data points.

5.3. Superiority of the NARLF and MAUF 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 PR, and the second with $ARLF$ and AUF instead of $NARLF$ and $MAUF$. The results are shown in an Online Supplement. 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 over $ARLF$ and AUF, as supposed in Section 4. Also, as the third ANOVA model in the supplement shows, the five original factors and their interactions has an R^2 of 85% for R3 and 94% for R5. Thus, we also infer that the selected measures do a reasonable job of explaining performance variances. We discuss the individual factors and interactions in the next sub-sections.

5.4. 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 Section 4.1).[10] Starting with R3, Fig. 7 shows a one-way analysis of means (ANOM) for all 20 PRs (averaging over all other factors), assuming a 95% confidence level. Overall, the "winning" PR (i.e., the one with the smallest average percent delay) is TWK-LST. MAXTWK and MINWCS tied (statistically, at 95% confidence) for second place, and EDDF, MAXSP, MINLFT and TWK-EST tied for third. Obvious "losers" include MAXSLK as the worst, MS and MCS

[10] Since the outcomes are not necessarily distributed normally for any of the objectives, especially with low $MAUF$ values, we considered using nonparametric statistics. With large samples, however, the differences become minor, and parametric tests can still be used (Gibbons, 1993).

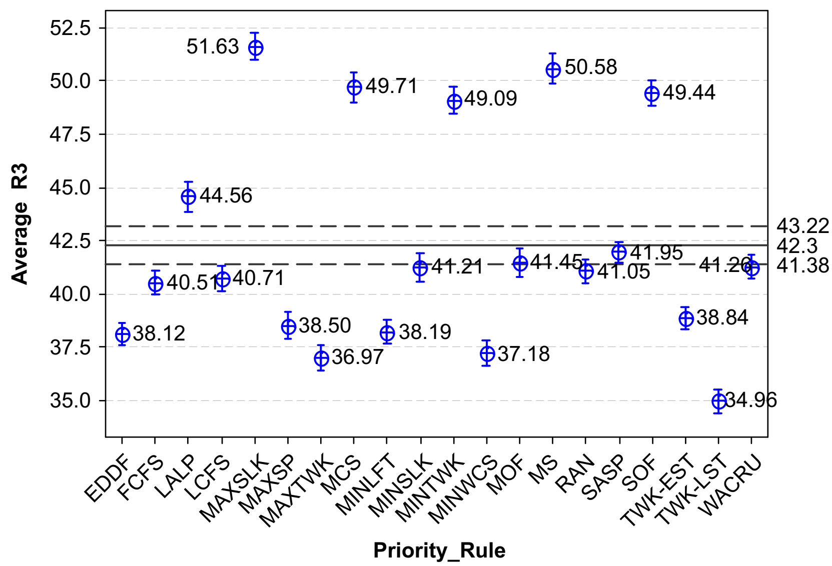

Fig. 7. One-way analysis of means (ANOM) for R3 ($\alpha=0.05$).

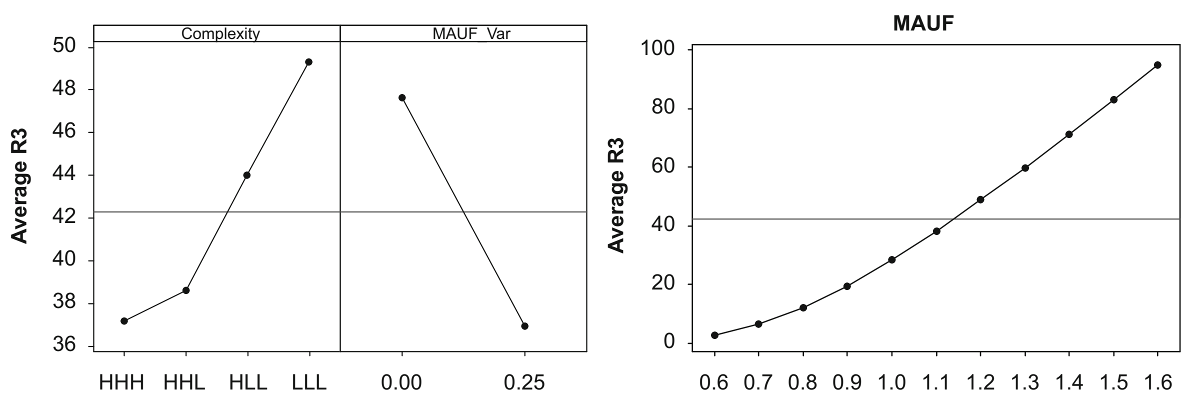

Fig. 8. Main effect plots for R3.

tied for second-worst, and SOF and MINTWK tied for third-worst. LALP also performed well below average. Note that these PRs performed worse than RAN, which itself outperformed the mean.[11] MINSLK and SASP, the two PRs recommended by Kurtulus and Davis (1982), did not perform especially well.

To investigate the underlying reasons for the success of the TWK-LST PR and the relative ineffectiveness of the other PRs in minimizing R3, it is necessary to look into how TWK-LST works (Lova and Tormos, 2001). This PR favors activities with MAXTWK (which performed well on its own), which means it considers all (three) projects simultaneously. Meanwhile, the other less effective PRs favor either the shortest (or longest) project, or focus on the number of successors within a single project. PRs that consider all three projects (such as MAXTWK, MINWCS, and TWK-EST) generally perform well on R3.

To further explore the influence of each factor on R3, we looked at the main effect plots (Fig. 8). The left side of Fig. 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.[12]

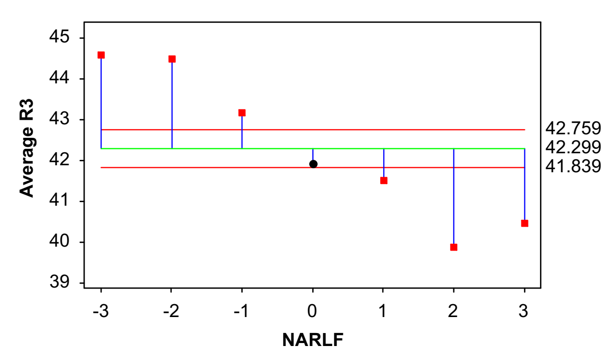

Fig. 9. *NARLF* main effect plot for R3.

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.[13] The middle of Fig. 8 shows R3 decreasing with an increase in *MAUF* variability, which

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

[12] Any delay will be a smaller percentage of a "long" problem.

[13] Kolisch (1999) reported improved performance results from commercial software packages on RCPSPs with many precedence constraints.

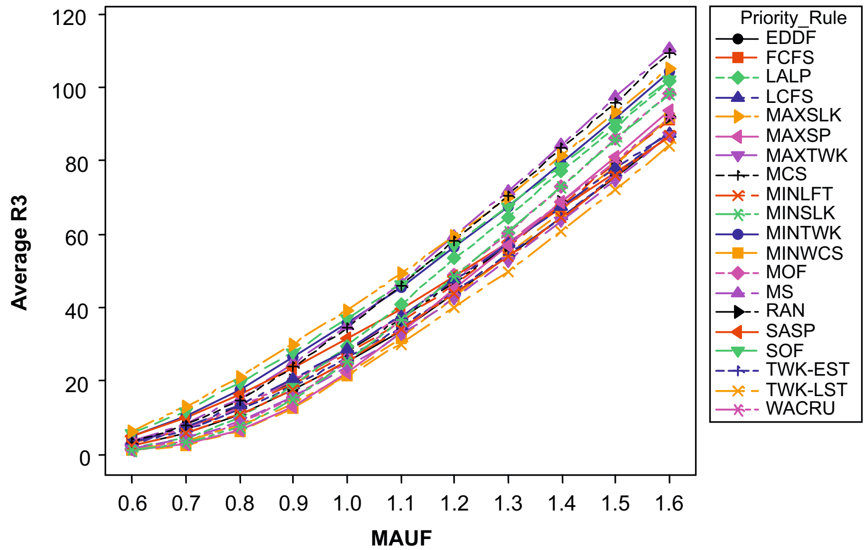

Fig. 10. R3 by *MAUF* level and PR (two-way interaction plot).

Table 4
Best PRs by *MAUF* level at 95% confidence.

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

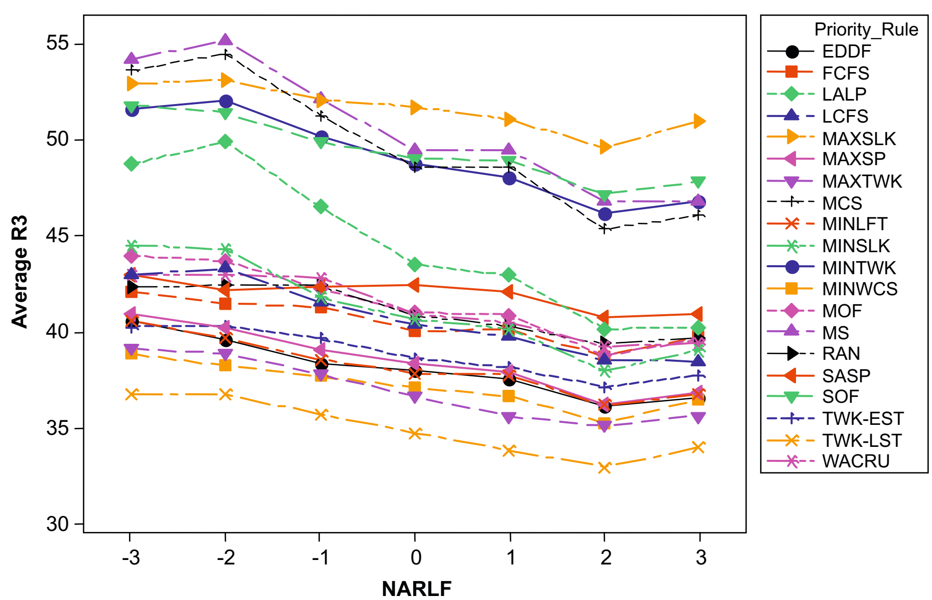

Fig. 11. R3 by *NARLF* level and PR.

Table 5
Best PRs by *NARLF* level at 95% confidence.

NARLF	−3	−2	−1	0	1	2	3
Best PR(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

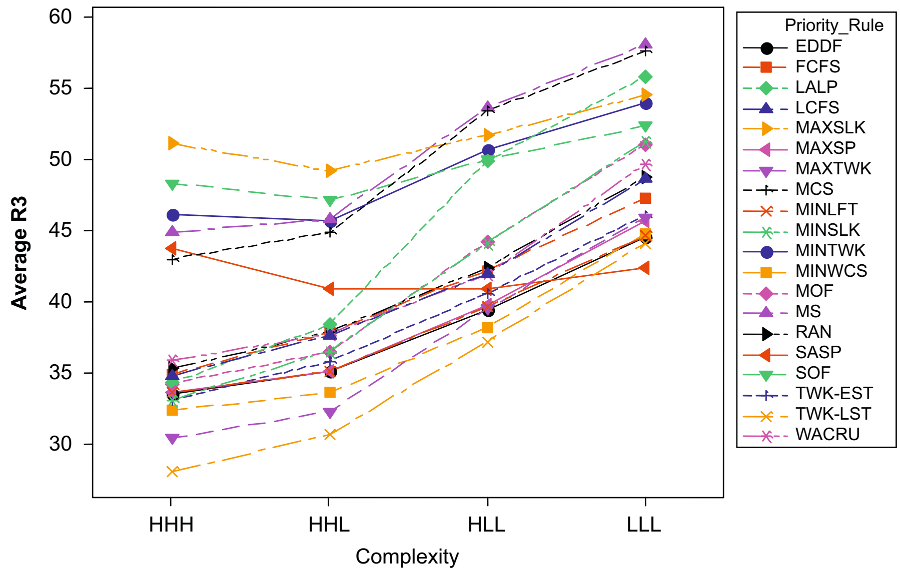

Fig. 12. R3 by complexity level and PR.

is also expected since the $\sigma^2_{MAUF}=0$ case implies that the problems are equally constrained by all (four) resource types. However, $\sigma^2_{MAUF}=0.25$ represents cases where the problems are mainly constrained by only one of the resource types. On the right side of Fig. 8, we also see the expected increase in percent lateness with higher *MAUF* (i.e., greater resource contention).

A fourth main effect plot in Fig. 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.

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 Figs. 9 and 10, plus the PR factor) and all two-way interactions were significant at 1% levels.[14] We now turn to a discussion of the two-way interactions between the four main factors and the PRs. First, the *MAUF* interaction plot in Fig. 10 indicates that, as expected, there is little difference in PRs when resource constraints are low. However, as *MAUF* increases, the performance disparity grows. To see these differences more clearly, we performed a *t*-test on the mean of each PR at selected *MAUF* levels, resulting in Table 4. MINWCS and MAXSP are the best PRs for low *MAUF* values. At high *MAUF* values, TWK-LST wins.[15]

Second, the *NARLF* interaction plot shown in Fig. 11 exhibits the superiority of TWK-LST at all levels, followed by MINWCS and MAXTWK. Table 5 shows the significant group of best PRs at each *NARLF* level. Since we are especially interested in the best PRs under conditions of high resource constraints, we also looked at the subset of problems with highly constrained resources

Table 6
Best PRs by complexity level at 95% confidence.

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

(i.e., with $MAUF \geq 1.4$). However, we found no significant change in Table 5 in the case of this three-way interaction.

Third, we looked at varied complexity levels (Fig. 12 and Table 6). TWK-LST outperforms the other PRs 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 PRs, SASP seems to be robust to changes in complexity: it is the only PR 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); the results were similar to Table 6.

Fourth, regarding σ^2_{MAUF}, TWK-LST was the best PR at zero variance, while it statistically tied with MINWCS at 0.25 variance, as shown in Fig. 13(a). As expected, all PRs perform better as a problem's resources are constrained by fewer of its resource types. However, this did not change the ranking of most PRs. 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 (Fig. 13(b)), where TWK-LST was the best PR regardless. Again, the ranking of the PRs did not change much with this factor, although as σ^2_{NARLF} increases, it becomes easier to distinguish the PRs' performance.

Fig. 14 shows two-way interaction plots for *NARLF*, *MAUF*, and *C*. In Fig. 14(a), we again see the effects of Observation 8a. Furthermore, the effect is heightened as *C* decreases (designated as 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

[14] 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. We address some particular three-way interactions below, although we found these to be relatively inconsequential.

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

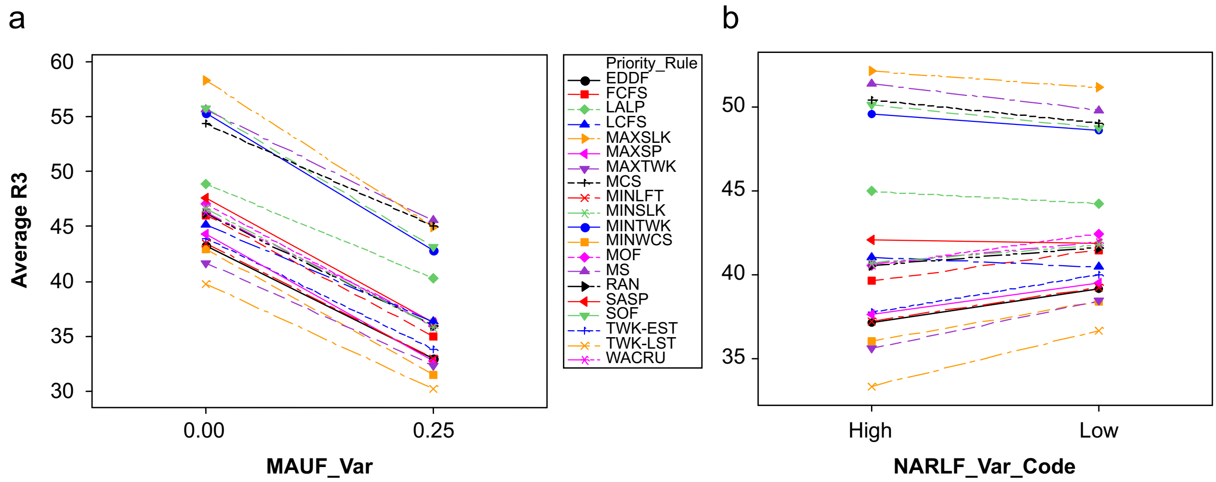

Fig. 13. R3 by σ^2_{MAUF} & σ^2_{NARLF} and PR.

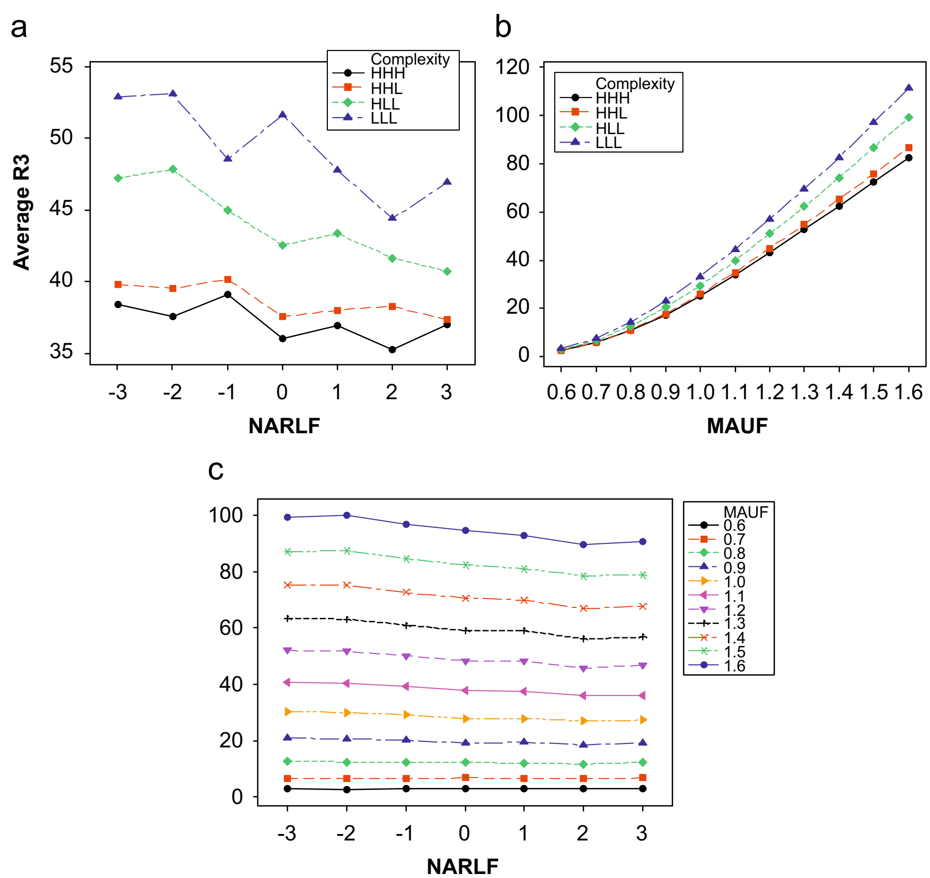

Fig. 14. Two-way interaction plots for *NARLF*, *MAUF*, and *C*.

end, and this effect is exacerbated as *C* 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, Fig. 14(b) repeats the message of Fig. 8(c) while indicating that the increase in percent delay with *MAUF* is regulated by *C*, growing more quickly when *C* is lower. Finally, Fig. 14(c) is similar to Fig. 2 in Kurtulus and Davis (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*.

5.5. 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, portfolio managers would have reason to focus on R5. However, since R5 is driven by the single longest project in a problem, it is a less sensitive measure than R3.

Fig. 15 shows the ANOM for R5 (cf. Fig. 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

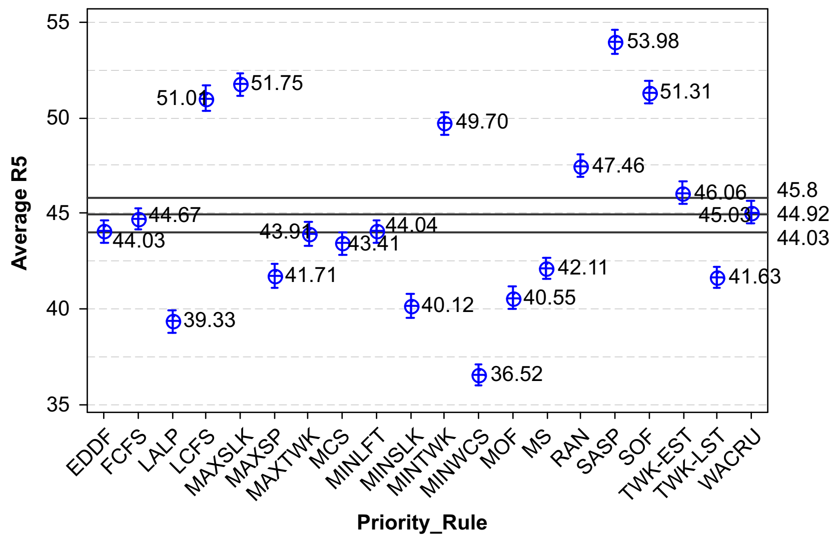

Fig. 15. One-way analysis of means (ANOM) for R5 ($\alpha = 0.05$).

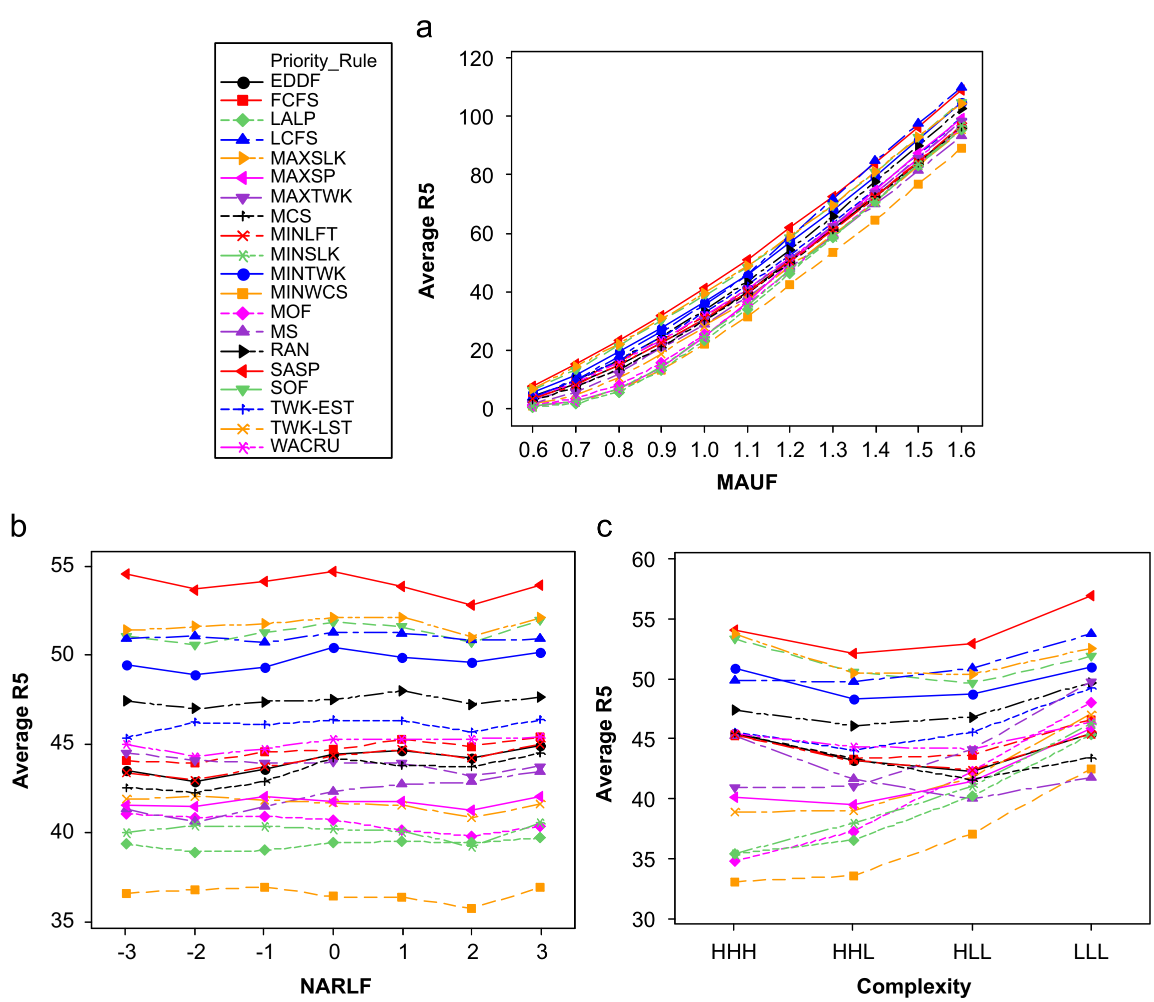

Fig. 16. R5 interaction plots between PRs and *MAUF*, *NARLF*, and *C*.

Please cite this article as: Browning, T.R., Yassine, A.A., Resource-constrained multi-project scheduling: Priority rule performance revisited. International Journal of Production Economics (2010), doi:10.1016/j.ijpe.2010.03.009

Table 7
Summary of results for R3 (α=0.05)[a].

Resource contention	Resource distribution					
	Front-loaded		Not front- or back-loaded		Back-loaded	
	***NARLF*=−3 & −2**		***NARLF*=−1, 0 & 1**		***NARLF*=2 & 3**	
	C		***C***		***C***	
	HHH	**LLL**	**HHH**	**LLL**	**HHH**	**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

[a] Multiple entries in each cell are listed in order of increasing means (i.e., the best PR is listed first).

decent, the relative performance of TWK-LST declined for R5 relative to R3. MINWCS emerges as the overall best PR by R5.

Speculating on the reasons behind the MINWCS PR's success requires looking more closely at how it works. This PR 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 PR seems to prevent the most damaging delays from occurring. However, it does this at a much greater computational expense than the other PRs, since it must "look ahead" for the potential downstream delays in terms of each possible pair of activities in ***D***. Note that this PR also performs well for R3 (except for high *MAUF* cases).

Finally, while SASP performed well by R3 in several situations, it was the worst PR 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.[16]

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 PRs differs. Fig. 16(a) (cf. Fig. 10) shows the expected reduction in performance as *MAUF* increases. MINWCS emerges as the best PR for $MAUF \geq 1.0$. SASP performs poorly at all *MAUF* levels. While also showing the clear superiority of MINWCS and the inferiority of SASP, Fig. 16(b) does not show improved performance with higher *NARLF* values, as seen in Fig. 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).

Fig. 16(c) indicates the trend towards diminished performance with lower complexity levels, as per Fig. 12, although SASP no longer bucks the trend. Instead, several other PRs now prevail against the general trend by performing poorly under situations of high *C*. In particular, MS and MCS perform better as *C* 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 larger, and thus there are more potential bad decisions that a PR can make. In these situations, PRs that happen to focus on the factors that are important for a particular objective (versus on generally important ones) perform better. PRs 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 *C* cases, all of the activities have a lot of successors.) The MS and MCS PRs 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.

6. Implications for managers

In the previous section, we identified, confirmed, and discussed several important factors that contribute to project and

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

Table 8
Summary of results for R5 (α=0.05)[a].

Resource contention	Resource distribution					
	Front-loaded		Not front- or back-loaded		Back-loaded	
	NARLF=−3 & −2		*NARLF*=−1, 0 & 1		*NARLF*=2 & 3	
	C		*C*		*C*	
	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

[a] Multiple entries in each cell are listed in order of increasing means (i.e., the best PR is listed first).

portfolio delay. To distill these results for managers, we developed two decision tables (Tables 7 and 8) to aid in selecting the best PR 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 aligns 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.

We observe several patterns in these tables. First, the number of winning (statistically tied) PRs 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 PR 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.

To benefit from our results and recommendations as summarized in Tables 7 and 8, 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 unable to do this exactly*. First, regarding *C*, managers can qualitatively estimate whether they are dealing with a high-*C* situation or a low-*C* 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-*C* projects contain a much greater number of dependencies (precedence constraints). Thus, several indicators can help a manager rate *C* 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.

The relative robustness of certain PRs 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,[17] it is difficult to experiment with various PRs or apply 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.

7. Summary and 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 "rules of thumb" such as MINSLK.

In the context of the static RCMPSP, this paper uses relatively new measures in the most comprehensive study of PRs to date, making a number of observations, some of which are relatively

[17] 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.

Please cite this article as: Browning, T.R., Yassine, A.A., Resource-constrained multi-project scheduling: Priority rule performance revisited. International Journal of Production Economics (2010), doi:10.1016/j.ijpe.2010.03.009

intuitive and others which are less so. While many additional PRs could also be studied, we cover the most popular ones, and a much larger group than other published studies in this area. 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 provides much-needed guidance on the use of certain PRs in varied project situations and objectives. Finally, the paper explicitly distinguishes the perspectives of project and portfolio managers.

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 PR. Accordingly, we developed a decision table to guide managers in choosing among best PRs based on *MAUF*, *NARLF*, and *C*, which constitutes a significant extension to the results reported by Kurtulus and Davis (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 PRs for this study, it was inevitable that we would confirm certain PRs as superior and others as inferior, agreeing with some prior studies while disagreeing with others. However, even the prior studies with which our results agree did not caveat their recommendations by problem characteristics, nor did they compare them with many other PRs. Therefore, beyond mere confirmation, we provide much greater generalization and specification of some particular results from prior studies.

While project scheduling, PRs, and related topics have been studied for at least 50 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 PRs 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 also be applied on a rolling-horizon basis in a dynamic environment.

Future research could expand our study to include additional PRs, compare results with the serial SGS, 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 PRs and more advanced heuristics. The results reported in this study can also be used for the development of improved PRs that take advantage of the superiority of certain PRs under specific conditions. For instance, one might develop an adaptive PR that shifts between simpler PRs as a project or portfolio progresses and its circumstances change.

Acknowledgements

The authors are grateful to two anonymous reviewers and 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.

Appendix A. Supplementary material

Supplementary data associated with this article can be found in the online version at 10.1016/j.ijpe.2010.03.009.

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. In: Proceedings of the Annual Meeting of the Decision Sciences Institute, San Diego, CA, November.
Baker, K., 1974. Introduction to Sequencing and Scheduling. Wiley, New York.
Bock, D., Patterson, J., 1990. A comparison of due date setting, resource assignment, and job preemption heuristics for the multi-project scheduling problem. Decision Sciences 21 (3), 387–402.
Boctor, F.F., 1990. Some efficient multi-heuristic procedures for resource-constrained project scheduling. European Journal of Operational Research 49 (1), 3–13.
Bouleimen, K., Lecocq, H., 2000. Multi-objective simulated annealing for the resource-constrained multi-project scheduling problem. In: Proceedings of the 7th International Workshop on Project Management and Scheduling (PMS 2000), Osnabrück, Germany, April 17–19.
Browning, T.R., Yassine, A.A., 2010. A random generator of resource-constrained multi-project network problems. Journal of Scheduling 13 (2), 143–161.
Brucker, P., Drexl, A., Möhring, R., Neumann, K., Pesch, E., 1999. Resource-constrained project scheduling: notation, classification, models, and methods. European Journal of Operational Research 112 (1), 3–41.
Chase, R.B., Jacobs, F.R., Aquilano, N.J., 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. European Journal of Operational Research 76 (1), 176–191.
Chiu, H.N., Tsai, D.M., 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., Mandelbaum, A., Shtub, A., 2004. Multi-project scheduling and control: a process-based comparative study of the critical chain methodology and some alternatives. Project Management Journal 35 (2), 39–50.
Confessore, G., Giordani, S., Rismondo, S., 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 Science 22 (11), 1186–1194.
Davis, E.W., 1975. Project network summary measures and constrained resource scheduling. IIE Transactions 7 (2), 132–142.
Davis, E.W., Patterson, J.H., 1975. A comparison of heuristic and optimum solutions in resource-constrained project scheduling. Management Science 21 (8), 944–955.
Deckro, R.F., Winkofsky, E.P., Hebert, J.E., Gagnon, R., 1991. A decomposition approach to multi-project scheduling. European Journal of Operational Research 51 (1), 110–118.
Dumond, J., Mabert, V.A., 1988. Evaluating project scheduling and due date assignment procedures: an experimental analysis. Management Science 34 (1), 101–118.
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, USA.
Gonçalves, J.F., Mendes, J.J.d.M., Resende, M.G.C., 2008. A genetic algorithm for the resource constrained multi-project scheduling problem. European Journal of Operational Research 189 (3), 1171–1190.
Hartmann, S., Briskorn, D., 2009. A survey of variants and extensions of the resource-constrained project scheduling problem. European Journal of Operational Research, forthcoming.
Hartmann, S., Kolisch, R., 2000. Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem. European Journal of Operational Research 127 (2), 394–407.
Herroelen, W.S., 2005. Project scheduling—theory and practice. Production and Operations Management 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., Yun, Y.S., Yoon, J.M., Gen, M., Yamazaki, G., 2005. Hybrid genetic algorithm with adaptive abilities for resource-constrained multiple project scheduling. Computers in Industry 56 (2), 143–160.

Kim, S.-Y., Leachman, R.C., 1993. Multi-project scheduling with explicit lateness costs. IIE Transactions 25 (2), 34–44.

Kolisch, R., 1996a. Efficient priority rules for the resource-constrained project scheduling problem. Journal of Operations Management 14 (3), 179–192.

Kolisch, R., 1996b. Serial and parallel resource-constrained project scheduling methods revisited: theory and computation. European Journal of Operational Research 90 (2), 320–333.

Kolisch, R., 1999. Resource allocation capabilities of commercial project management software packages. Interfaces 29 (4), 19–31.

Kolisch, R., Hartmann, S., 1999. Heuristic algorithms for solving the resource-constrained project scheduling problem: classification and computational analysis. In: Weglarz, J. (Ed.), Project scheduling: Recent Models, Algorithms and Applications. Kluwer Academic Publishers, Boston, pp. 147–178.

Kolisch, R., Hartmann, S., 2005. Experimental investigation of heuristics for resource-constrained project scheduling: an update. European Journal of Operational Research 174 (1), 23–37.

Kolisch, R., Sprecher, A., Drexl, A., 1995. Characterization and generation of a general class of resource-constrained project scheduling problems. Management Science 41 (10), 1693–1703.

Krüger, D., Scholl, A., 2008. Managing and modelling general resource transfers in (multi-)project scheduling. OR Spectrum 32 (4), 369–394.

Krüger, D., Scholl, A., 2009. A heuristic solution framework for the resource constrained (multi-) project scheduling problem with sequence-dependent transfer times. European Journal of Operational Research 197, 492–508.

Kumanan, S., Jose, G.J., Raja, K., 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. Journal of Operations Management 5 (3), 291–307.

Kurtulus, I., Davis, E.W., 1982. Multi-project scheduling: categorization of heuristic rules performance. Management Science 28 (2), 161–172.

Kurtulus, I.S., Narula, S.C., 1985. Multi-project scheduling: analysis of project performance. IIE Transactions 17 (1), 58–65.

Lawrence, S.R., Morton, T.E., 1993. Resource-constrained multi-project scheduling with tardy costs: comparing myopic, bottleneck, and resource pricing heuristics. European Journal of Operational Research 64 (2), 168–187.

Lenstra, J.K., Kan, A.H.G.R., 1978. Complexity of scheduling under precedence constraints. Operations Research 26 (1), 22–35.

Liberatore, M.J., Pollack-Johnson, B., 2003. Factors influencing the usage and selection of project management software. IEEE Transactions on Engineering Management 50 (2), 164–174.

Linyi, D., Yan, L., 2007. A particle swarm optimization for resource-constrained multi-project scheduling problem. In: Proceedings of the International Conference on Computational Intelligence and Security, Harbin, China, December 15–19, pp. 1010–1014.

Lova, A., Tormos, P., 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., Tormos, P., 2002. Combining random sampling and backward–forward heuristics for resource-constrained multi-project scheduling. In: Proceedings of the 8th International Workshop on Project Management and Scheduling, Valencia, Spain, April 3–5, pp. 244–248.

Maroto, C., Tormos, P., Lova, A., 1999. The evolution of software quality in project scheduling. In: Weglarz, J. (Ed.), Project Scheduling. Kluwer Academic Publishers, Boston, pp. 239–259.

Meredith, J.R., Mantel, S.J., 2009. Project Management, 7th ed. Wiley, New York.

Mize, J.H., 1964. A heuristic scheduling model for multi-project organizations. Ph.D. Thesis, Purdue University.

Mohanty, R.P., Siddiq, M.K., 1989. Multiple projects-multiple resources-constrained scheduling: some studies. International Journal of Production Research 27 (2), 261–280.

Özdamar, L., Ulusoy, G., 1995. A survey on the resource-constrained project scheduling problem. IIE 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.

Payne, J.H., 1995. Management of multiple simultaneous projects: a state-of-the-art review. International Journal of Project Management 13 (3), 163–168.

Pritsker, A.A.B., Watters, L.J., Wolfe, P.M., 1969. Multiproject scheduling with limited resources: a zero-one programming approach. Management Science 16 (1), 93–108.

Talbot, F.B., 1982. Resource-constrained project scheduling with time-resource tradeoffs: the nonpreemptive case. Management Science 28 (10), 1197–1210.

Thomas, P.R., Salhi, S., 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. In: Proceedings of the 1st ORSTW Conference on Technology and Management, Taipei, pp. 18–28.

Tsubakitani, S., Deckro, R.F., 1990. A heuristic for multi-project scheduling with limited resources in the housing industry. European Journal of Operational Research 49 (1), 80–91.

Ulusoy, G., Özdamar, L., 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. European Journal of Operational Research 78 (2), 267–275.

Woodworth, B.M., Willie, C.J., 1975. A heuristic algorithm for resource leveling in multi-project, multi-resource scheduling. Decision Sciences 6 (3), 525–540.

Yang, K.-K., Sum, C.-C., 1993. A comparison of resource allocation and activity scheduling rules in a dynamic multi-project environment. Journal of Operations Management 11 (2), 207–218.

Yang, K.-K., Sum, C.-C., 1997. An evaluation of due date, resource allocation, project release, and activity scheduling rules in a multiproject environment. European Journal of Operational Research 103, 139–154.