Managing a Portfolio of Product Development Projects under Resource Constraints

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

Ali A. Yassine
Engineering Management Program
American University of Beirut
Beirut, Lebanon
ali.yassine@aub.edu.lb

This is a preprint of a paper published in *Decision Sciences*.
Citation: Browning, Tyson R. and Ali A. Yassine (2016) "Managing a Portfolio of Product Development Projects under Resource Constraints," *Decision Sciences*, 47(2): 333-372.

The first author is grateful for support from the Neeley Summer Research Award Program from the Neeley School of Business at TCU and a grant from the U.S. Navy, Office of Naval Research (grant no. N00014-11-1-0739). The second author acknowledges the financial support of the University Research Board at the American University of Beirut.

Managing a Portfolio of Product Development Projects under Resource Constraints

Managers of product development (PD) project portfolios face difficult decisions in allocating limited resources to minimize project or portfolio delay. Although PD projects are highly iterative (cyclical), almost all of the vast literature on project scheduling assumes that projects are acyclical. This paper addresses this gap with a comprehensive analysis of 31 priority rules (PRs) on 18,480 portfolios containing 55,440 iterative projects. We find that the best PRs for iterative project portfolios differ significantly from those for acyclical ones, and that the best PRs at the project level differ from those at the portfolio level. The best PR depends on project and portfolio characteristics such as network density, iteration intensity, resource loading profile, and amount of resource contention. In particular, by amplifying the effects of iteration, high-density networks hold dramatically different implications for iterative projects. Moreover, the best PR also differs depending on whether the objective is to minimize the average delay to all projects or to minimize delay to the overall portfolio. Thus, a project or portfolio manager who uses the same PR on all occasions will exhibit unnecessarily poor performance in most cases.

Keywords: *Product Development, Project Management, Portfolio Management, Resource Constraints, Multi-Project Scheduling, Priority Rules*

1. Introduction

Many firms must manage a portfolio of product development (PD) projects that depend upon a common pool of limited resources. Even if the size of this pool is sufficient for the average needs of all of the projects, they will nevertheless contend for specific resources at particular times, causing delays to individual projects and the portfolio. Some of these situations can be foreseen and managed, but the notoriously iterative nature of PD projects (Kline 1985; Smith and Eppinger 1997a) fraught with unplanned and undiscovered rework (Cooper 1993b) exacerbates the situation. Moreover, the novelty and ambiguity of PD projects makes it difficult to specify their activity networks *a priori* (Pich et al. 2002; Anderson and Joglekar 2005; Lévárdy and Browning 2009). In this context, PD project managers must often rely on only the most aggregate projections of resource needs. Hence, resource contentions with other projects in the PD portfolio are especially difficult to predict and manage; they often occur as surprises that derail projects from their plans. Moreover, the managers of individual projects, who want their own projects to be on time, may find themselves at odds with the portfolio manager, who wants to advance the overall portfolio. What is optimal for one project may not be best for the portfolio.

Research in this area is important for several reasons. First, most existing research on project portfolios focuses on project selection (and sometimes initial resource allocations), but it does not address the ongoing adjudication of resource contentions. Practicing managers need guidance on this aspect. Second, almost all of the research on resource-constrained (single and multi-project) scheduling addresses only acyclical networks of activities; it does not account for the iterative nature of PD projects, which calls for different managerial approaches. Third, the existing work does not clarify the differences between the single-project and multi-project perspectives (e.g., Dilts and Pence 2006), which can cause tension between project and portfolio managers. Fourth, the practical reality is that most projects do not actually build detailed activity network models (Liberatore and Titus 1983; Besner and Hobbs 2008), even though such models are necessary for applying the methods developed by researchers to find optimal or near-optimal solutions. In PD projects—noted for their novelty, ambiguity, and complexity—building such networks correctly with foresight is especially problematic. Thus, PD project and portfolio managers are inhibited from applying entire streams of research because of disconnects with practice. Instead, when faced with resource contention situations, many managers arbitrate locally with simple decision rules based on activity or project attributes such as "Which activity has more slack time?" or "Which activity is holding up the most other critical activities?" However, the effectiveness of these approaches, generally called *priority rules* (PRs), can vary greatly depending on the characteristics of the project and the portfolio. It would be extremely beneficial to know which rules to use, which to avoid, and when.

This paper contributes guidance on the efficacy of various PRs in *cyclical activity* networks, and it does so in terms of *project and portfolio characteristics* that can be ascertained *without resorting to fully detailed* *models of the networks*. We address the following research questions. What are the most effective PRs in iterative PD projects? How do these differ from acyclical projects? How do these results compare for two different objectives—minimizing average project delay and overall portfolio delay? How do project and portfolio characteristics—such as front- or back-loading of resources, network density, amount of iteration, and degree of resource contention—affect the results? What are some new PRs based on the characteristics of iterative projects, and how do these fare in comparison to established PRs?

No other study has addressed these questions. To do so, we designed a carefully controlled experiment with 18,480 test portfolios (containing 55,440 projects) with deliberately varied amounts of resource contention, resource front- and back-loading, network density, and activity iteration. We solved each of these portfolios with 31 PRs (20 existing and 11 new), and compared the results in terms of two different objective functions; namely, minimizing average project delay and overall portfolio delay.

The results are significant, interesting, and useful for researchers and managers. The choice of PR matters greatly, and PRs that work well for acyclical projects do not necessarily work well for PD projects or portfolios. Several widely-advocated PRs performed very poorly for PD project portfolios. The choice of best PR depends significantly on project and portfolio characteristics such as network density, iteration intensity, and amount of resource contention. In particular, by amplifying the effects of iteration, high-density networks hold dramatically different implications for iterative projects. Thus, a project or portfolio manager who uses the same PR on all occasions will exhibit unnecessarily poor performance in most cases. Perhaps most importantly for managers, we are able to distill the overall results in a way that provides managerial guidance *without* requiring a detailed model of each project's activity network. These results constitute a significant step beyond other published studies that do not address the cyclical nature of PD project networks.

2. Background and Literature Review

This paper bridges three streams of literature which so far have been fairly disparate: PD project portfolio management, iteration in PD projects, and resource-constrained multi-project scheduling. This section describes the background of this research as motivated by each of these areas.

2.1 PD Project Portfolio Management

As the operationalization of a business strategy, project portfolio management (PPM) is essential for the success and survival of PD organizations. Cooper et al. (2002) noted four goals in PPM: maximizing portfolio value, balancing the portfolio, aligning the portfolio with strategic business objectives, and allocating resources optimally across the portfolio. Various tools have been proposed to facilitate PPM—including the Boston Consulting Group's product portfolio matrix, risk-reward diagrams, Bubble diagrams, etc. (Wysocki and McGarry 2003; Rad and Levin 2006)—that generally help to visualize the portfolio with respect to multiple criteria. Various project rating and scoring techniques have been developed for PPM (Terwiesch and Ulrich

2008), as have sophisticated mathematical programming and optimization techniques such as mixed integer programs (Beaujon et al. 2001; Jiao et al. 2007), non-linear integer programs (Dickinson et al. 2001; Gutjahr et al. 2010), and dynamic programs (Loch and Kavadias 2002; Kavadias and Loch 2003). Cooper et al. (2002) reported that improved product and firm performance can be attributed to certain types of PPM practices and tools—especially an effective stage-gating process that weeds out poor projects and redistributes resources. Shane and Ulrich (2004) found that, out of the technological innovation, PD, and entrepreneurship research published from 1954-2003 in *Management Science*, "product planning and portfolios" comprised the second largest sub-stream—although they noted that this research had "found very little use in practice."

However, the bulk of research on PPM methods has focused at the strategic level—e.g., on project selection—rather than at the operational level—e.g., on project execution (Pennypacker and Dye 2002; Hans et al. 2007). Even stage gate reviews offer only relatively macro-level punctuations of projects. When faced with chronic, day-to-day resource contentions among projects within his or her portfolio, a portfolio manager will find little guidance for a prioritization decision from the PPM literature (Engwall and Jerbrant 2003; Blichfeldt and Eskerod 2008). Thus, it is important to extend the scholarly PPM literature beyond portfolio planning to include portfolio execution and control at the level of the dynamic, operational decisions about resource allocation that are important to practicing managers (Anderson and Joglekar 2005).

2.2 Iteration in PD Projects

Doing something novel, ambiguous, and complex with a multi-disciplinary team requires a degree of experimentation (Thomke 1998) and convergence on a satisfactory solution (Simon 1996). Thus, PD and innovation processes are inherently iterative (cyclical) (Kline 1985; Smith and Eppinger 1997a; Braha and Bar-Yam 2007). In general, iteration represents the rework of an activity due to deficiencies with its prior results, triggered by the feedback of information discovered later in the project (Loch and Terwiesch 2005; Lévárdy and Browning 2009). These deficiencies could be planned (e.g., an activity deliberately defers work, as in "spiral development") or unplanned (e.g., due to failure to meet requirements, detected errors, or changes in the inputs upon which the work was done). The advent of "concurrent engineering" has only increased the challenges of parallel work, iteration, and resource constraints. Iteration and rework have been established empirically as significant drivers of PD time, cost, and quality (Cooper 1993a; Osborne 1993; Sosa et al. 2013). Their effective management is required to plan and control project cost, duration, quality, and risk (Browning and Ramasesh 2007).

However, conventional project management (and scheduling) literature and techniques (e.g., Meredith and Mantel 2012; PMI 2013) assume that projects are acyclical and do not account for iteration. On the other hand, most models of iteration in PD projects do not account for resource constraints at all, or do so only in a general way (e.g., without accounting for different resource types and dynamic requirements from other projects), and deal only with single projects (e.g., Cho and Eppinger 2005; Joglekar and Ford 2005; Lee et al.

2008). It is important to extend the scholarly literature on PD project modeling to explore the very real and practical implications of resource limitations on a portfolio of iterative projects (Froehle and Roth 2007).

2.3 Resource-Constrained Multi-Project Scheduling

Resource-constrained multi-project scheduling (RCMPS) seeks to optimize the performance of a set of projects that draw on a common pool of resources (e.g., Pritsker et al. 1969; Hans et al. 2007; Hartmann and Briskorn 2009). Solutions are found by scheduling the activities in each project and determining the projects' resulting durations. Whenever the resource needs of concurrent activities exceed the amount available, a decision must be made to prioritize some activities and delay others. The RCMPS problem is strongly NP-hard (Lenstra and Kan 1978), so exact solution methods are impractical for large problems (Herroelen 2005). Research has thus focused on heuristic procedures (Kolisch and Hartmann 1999; 2005), including PRs (e.g., Browning and Yassine 2010b; Vázquez et al. 2013), classical meta-heuristics (e.g., Bouleimen and Lecocq 2000; Linyi and Yan 2007; Gonçalves et al. 2008; Okada et al. 2009), non-standard meta-heuristics (e.g., Confessore et al. 2007; Homberger 2007), and other heuristics (e.g., Lova and Tormos 2002). PRs include single- and multi-pass methods (Hartmann and Kolisch 2000), where single-pass PRs draw upon a single value (such as start time or duration) and multi-pass methods either employ more than one PR in succession (e.g., Lova and Tormos 2001) or combine a single PR with some degree of randomness. PRs can also be classified according to whether they use activity-, project-, or resource-related information (Kolisch 1996a), although some multi-pass PRs draw upon more than one of these (Hartmann and Kolisch 2000). Although much of the recent research on RCMPS has emphasized meta-heuristics, PRs remain important for several reasons:

- PRs' lesser computational expense makes them attractive for very large problems (Kolisch 1996a), which are common in RCMPS (as compared to single-project cases).
- PRs are a component of other (local search-based and sampling) heuristics (Kolisch 1996b).
- PRs provide initial solutions for meta-heuristics (Hartmann and Kolisch 2000).
- PRs are used extensively by commercial project scheduling software (Herroelen 2005) and practitioners.
- PRs can be gainfully employed by managers even without the formal activity network models required for the application of meta-heuristics (Browning and Yassine 2010b).

The last reason is especially compelling, because it does not matter how large a problem a modern computer can optimally process, nor how fast a meta-heuristic can reach a near-optimal solution, nor how good that solution is, if a project does not have the requisite input—an accurate activity network model (which is actually quite rare for PD projects). When faced with a resource allocation decision, a practicing project manager will often make a quick call based on intuition or a simple rule of thumb. Therefore, a litany of PRs remain prominent in contemporary project management textbooks (e.g., Meredith and Mantel 2012).

However, almost all RCMPS studies, whether using PRs or other heuristics or meta-heuristics, have addressed only *acyclical* networks, so their guidance could mislead managers of iterative PD projects and portfolios. Only a few RCMPS papers have looked at cyclical projects. Tukel (1996) investigated the performance of five PRs and confirmed that feedback in an activity network (i.e., cycles) affected PR performance. Luh et al. (1999) studied the problem of scheduling design activities with an uncertain number of iterations by using a stochastic dynamic programming approach to determine the optimal start times of activities to minimize expected weighted tardiness. Similarly, Yan et al. (2002) presented a branch-and-bound algorithm to determine an optimal project schedule in the presence of uncertain activity iterations. Zhuang and Yassine (2004) developed a genetic algorithm-based approach for solving the iterative RCPSP that randomly generates a value based on the feedback probability and thus decides whether to incorporate the feedback into the schedule or not. Yet, none of these works address a portfolio of PD projects, nor do they provide guidance on PR performance in diverse situations. Moreover, each requires a complete and accurate model of the activity network, something that is unlikely to exist for many PD projects. Thus, it is important to extend the scholarly literature on RCMPS to account for the realistic context of iterative PD projects.

3. Research Methods

To investigate the performance of popular and high-potential PRs, we designed a suite of test portfolios with a distribution of PD project and portfolio characteristics and solved these for two objective functions with 31 PRs. This section describes the project and portfolio characteristics and their measures, the test portfolios, the PRs, the objective functions, and the solution procedure.

3.1 PD Project and Portfolio Characteristics and Measures

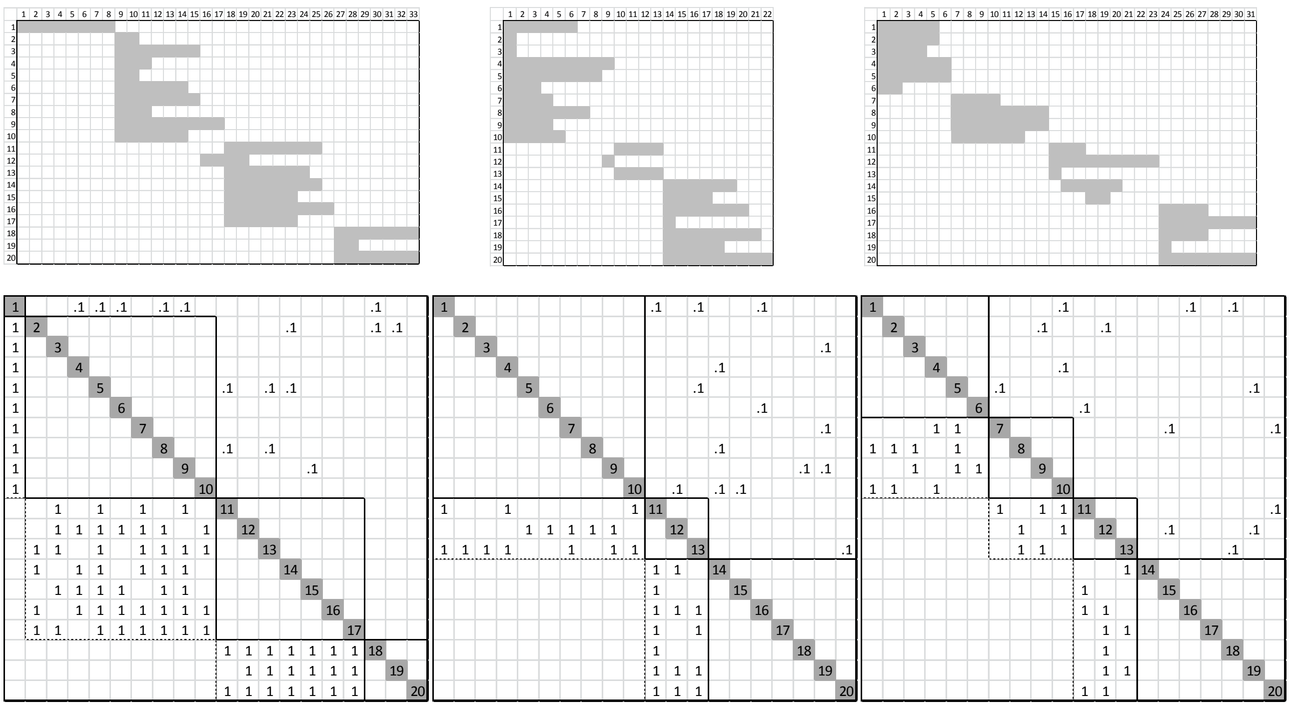
PD projects can differ substantially in terms of their network density, iteration intensity, and resource loading profile. Portfolios can vary in terms of the characteristics of their constituent projects and the overall amount of resource contention among them. These characteristics must be considered because the literature reports that they have a significant effect on PR performance.

3.1.1 Network Density

Several studies (e.g., Davis and Patterson 1975; Ulusoy and Özdamar 1989) in the RCMPS literature have established the significance of network complexity—the density of precedence constraints—to PR performance. Low-density networks are less precedence-constrained, which provides more degrees of freedom in determining a solution and therefore makes them more difficult to solve. High-density networks contain many more precedence relationships among the activities, implying fewer alternatives for starting activities sooner or later without delaying the project. Hence, PR performance differences increase with decreasing network density. Furthermore, not all precedence relationships (often called arcs in the RCMPS literature) are equally important (Tavares et al. 2002; Nassar and Hegab 2006). For example, if three activities are related by two arcs in a serial fashion, then a third arc going from the first to the third adds no further constraint to the network. This latter arc is thus called a redundant arc. The more recent density measures in the literature focus on non-redundant arcs. We adopt Browning and Yassine's (2010a) project network density measure:

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

where A' is the number of non-redundant, feed-forward arcs, N is the number of nodes (activities), and C is normalized over [0,1]. For example, 30 non-redundant, feed-forward arcs among 20 activities implies a fairly sparse network with C = 0.14, whereas 75 such arcs among 20 activities yields C = 0.69. Although the definitions and measures of network density from the RCMPS literature assume an acyclical network and therefore ignore any feedback arcs, C still provides a useful indicator in our context of iterative PD projects.



(a) C = 0.69 (b) C = 0.14 (c) C = 0.14

Figure 1: Example portfolio of three projects, each shown with Gantt chart and DSM representations.

Figure 1 shows an example portfolio with three projects, each with 20 activities shown in both Gantt chart and design structure matrix (DSM) representations (Eppinger and Browning 2012). In each Gantt chart, the activities appear in rows as a bar with width proportional to their duration. The horizontal axis is a time line showing the earliest start and finish time of each activity (ignoring any iteration or resource limitations). In each DSM, the activities are listed along the diagonal, and an arc from activity i to j is noted by an entry in cell (j, i) (row j, column i). For example, in project (a), activity 1 is a predecessor to activities 2-10. Feed-forward arcs appear as sub-diagonal marks in the DSMs; super-diagonal marks represent feedback arcs (to be discussed later). Project (a) has 75 non-redundant, feed-forward arcs, and projects (b) and (c) each have 30. The maximum possible number of *feed-forward* arcs is $(N^2 - N)/2$, which is 190 when $N = 20$, as can be seen by the number of cells available below the diagonal in each DSM. The maximum possible number of *non-redundant*, feed-forward arcs when $N = 20$ is 100, as determined by equation (1), although the actual number possible in any particular project may be less than this upper bound depending on the activity network's tiering structure (Browning and Yassine 2010a), which is shown by the empty blocks along the diagonal in each DSM. For example, in Figure 1 projects (a) and (c) each have four tiers, while project (b) has three. (These tiers are also discernible in the Gantt charts.) Any sub-diagonal marks within the blocks along the diagonal would increase the number of tiers. Any sub-diagonal marks outside the further dashed line blocks would be redundant because they would bypass an intermediate tier. For example, in project (a) a mark in cell (20,1) would be redundant because it would imply a dependency of activity 20 (in tier 4) on activity 1 (in tier 1), even though this dependency already exists indirectly via other activities (e.g., 10 and 17) in tiers 2 and 3.

3.1.2 Iteration Intensity

Iterative PD projects also have feedback arcs that can cause cycles in the network, having a dramatic effect on a project's schedule and stability (Smith and Eppinger 1997b; Browning and Eppinger 2002). These feedbacks may represent rework (e.g., failing to pass a design test or review, discovering an error, etc.), requirement changes, or design improvement (e.g., spiral development, design maturation). Unlike feed-forward arcs, which are usually assumed to be "hard" dependencies that govern the sequence of activities, feedback arcs usually represent the potential but not the certainty of returning to an earlier activity in the network. Thus, feedback arcs are often modeled as having a probability of occurrence. For example, the DSMs in Figure 1 each have 15 feedback arcs, each with 10% likelihood. Increasing the number of feedback arcs (F) and/or their probabilities (P) often increases a project's expected duration. (Note that the activity network's tiering structure constrains the random placement of the feedback arcs so as not to occur *within* a tier, because this would imply that the activities should be resequenced and parsed into separate tiers.)

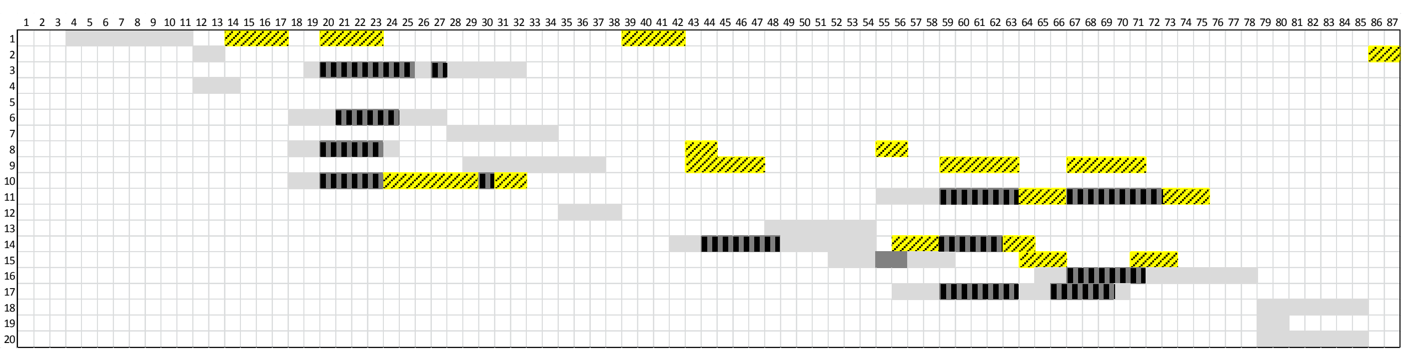


Figure 2: Gantt Chart View of Example Project Outcome with Resource Constraints and Iteration

For example, Figure 2 provides an example of the project shown in Figure 1(a) when iteration and resource constraints are taken into account. In this instance, the project's overall duration changes from 33 to 87 days. The first activity does not begin immediately because activities in other projects (not shown) gain priority. The bars with diagonal texture indicate instances of rework, and bars with bold, vertical texture indicate instances of waiting due to preemption by reworking activities (which are sometimes in other projects, not shown). The effects of iteration and resource constraints can be quite significant, which makes the approach to assigning activity priority especially important.

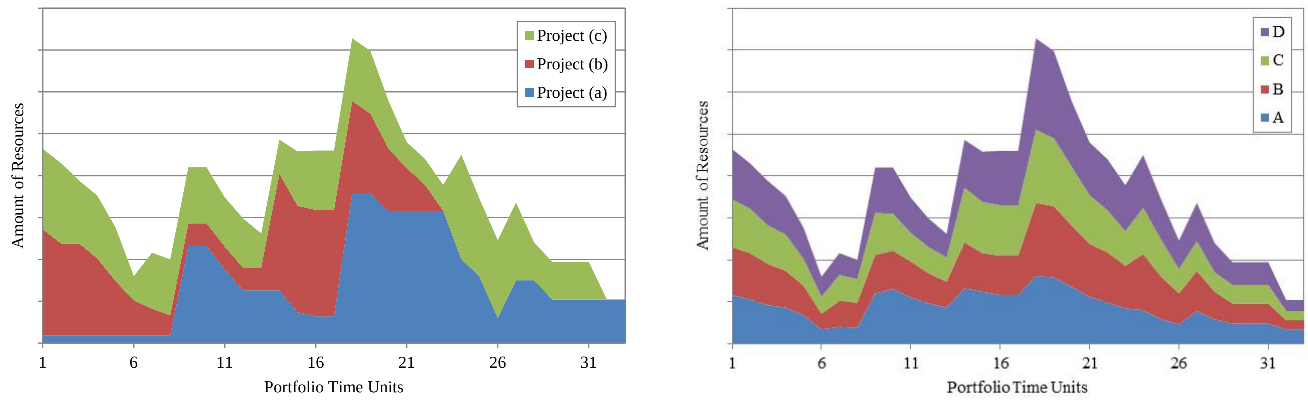
3.1.3 Resource Loading Profile

A resource loading profile characterizes the amount of resource(s) used over project or portfolio time. It identifies situations such as resource front-loading or back-loading. The importance of this characteristic and a metric for it grew from early research that identified measures such as the resource factor, which indicates the average number of resources used by an activity (Pascoe 1966; Cooper 1976). Kurtulus and Davis (1982) proposed the *average resource loading factor* (*ARLF*) to measure the extent to which a project's total resource requirements occur in the front or back half of its (resource unconstrained) critical path (CP) duration. Browning and Yassine (2010a; b) identified several problems with *ARLF* in the multi-project context that led them to propose a *normalized ARLF* (*NARLF*) measure that we adopt. *NARLF* measures the resource loading profile over a *portfolio's* CP duration:

$$NARLF = \frac{1}{L \cdot CP_{\mathrm{Max}}} \sum_{l=1}^{L} \sum_{t=1}^{CP_l} \sum_{k=1}^{K_{il}} \sum_{i=1}^{N_l} Z_{ilt} X_{ilt} \left(\frac{r_{ilk}}{K_{il}} \right) \tag{2}$$

where $Z_{ilt} = \begin{cases} -1 & t \leq CP_l/2 \\ 1 & t > CP_l/2 \end{cases}$, $X_{ilt} = \begin{cases} 1 & \text{if activity } i \text{ of project } l \text{ is active at time } t \\ 0 & \text{otherwise} \end{cases}$, $Z_{ilt}X_{ilt} \in \{-1, 0, 1\}$, 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, r_{ilk} is the amount of resource type k required by task i in project l, and $CP_{\mathrm{Max}} = \mathrm{Max}(CP_1, \ldots, CP_L)$. For example, in the portfolio shown in Figure 1, $CP_{\mathrm{Max}} = \mathrm{Max}(33,22,31) = 33$. Portfolios with $NARLF < 0$ are "front-loaded" in their resource requirements and portfolios with $NARLF > 0$ are "back-loaded." Although the definitions and measures from the RCMPS literature assume an acyclical network and therefore do not account for any feedback arcs, *NARLF* nevertheless provides a useful indicator in our context of iterative PD projects.

Figure 3 shows two versions of resource loading profiles for the portfolio in Figure 1. Profile (a) shows resource usage by project, and profile (b) shows resource usage by each of the four types of resources required by this example portfolio. As this portfolio takes 33 time units to complete with sufficient resources and no iteration, any resources used in the first 16 time units are in the front half and any others are in the back half. Due primarily to the surge in desired resources during time units 18-20, this portfolio has *NARLF* = 2, meaning it is relatively back-loaded. Note that these profiles represent *desired* research usage levels in the absence of resource constraints and iteration. If the desired amount of any resource type exceeds the amount available at any point in time, then some activities will be delayed, thus shifting the profiles to the right. Any activity iterations will also increase the total resource requirements and distort the profiles.



(a) By project

(b) By resource type

Figure 3: Resource loading profiles for the example portfolio in Figure 1.

3.1.4 Resource Contention

The amount of each type of resource available at any given time drives the degree of resource contention among the project activities in the portfolio. Measures proposed in the literature include the utilization factor (Davis 1975), the average utilization factor (Kurtulus and Davis 1982), and the modified average utilization factor (Browning and Yassine 2010a,b) for each type of resource over equal intervals of portfolio time.

$$MAUF_k = \frac{1}{S}\sum_{s=1}^{S}\frac{W_{sk}}{R_k s} \tag{3}$$

where S is the number of intervals (usually, $S = CP_{\mathrm{Max}}$, where the latter is defined as for equation (2), rounded to the nearest integer), R_k is the (renewable) amount of resource type k available at each interval, and W_{sk} specifies the total amount of resource k required over interval s, which is given by:

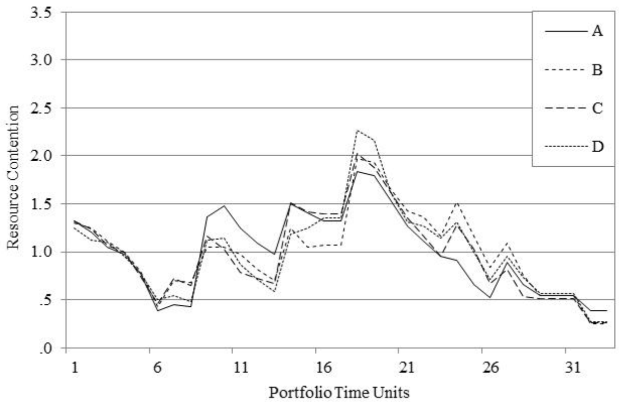
$$W_{sk} = \sum_{t=a}^{b}\sum_{l=1}^{L}\sum_{i=1}^{N_l} r_{ilk} X_{ilt} \tag{4}$$

where a and b respectively represent the beginning and end points of interval 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 (2). Since $MAUF$ is essentially a ratio of resources required to resources available, averaged across equal intervals of portfolio time, $MAUF_k > 1$ indicates that resource type k is, on average, constrained over the course of the portfolio's execution. The $MAUF$ for a portfolio with K types of resources is determined by:

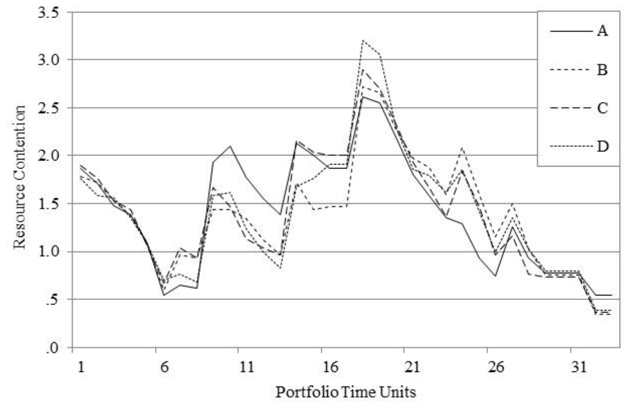
$$MAUF = \mathrm{Max}(MAUF_1, MAUF_2, \ldots, MAUF_K) \tag{5}$$

Figure 4 shows resource contention levels for the portfolio in Figures 1 and 3. In Figure 4(a) $MAUF$ =

1.0, meaning that, *on average*, the desired amount of resources equals the amount available. Nevertheless, the demand for particular resources exceeds supply at numerous points over the course of the portfolio (wherever the peak exceeds 1.0). Figure 4(b) shows the case when the supply of resources is even more limited and *MAUF* = 1.4. Project and portfolio duration increase with increasing *MAUF*, and activity prioritization decisions become increasingly important. Note that the resource contention profiles in Figure 4 do not account for any effects of iteration.



(a) *MAUF* = 1.0



(b) *MAUF* = 1.4

Figure 4: Two examples of resource contention profiles for the portfolio in Figure 1.

3.2 Test Portfolios

We used the generator developed by Browning and Yassine (2010a) to build a rigorous distribution of experimental portfolios with the specifications listed in Table 1. To maximize insights from variations in the main factors, we held four factors constant. First and second, because no specific relationship has been reported between project or portfolio size and PR solution quality (Kurtulus and Davis 1982; Hartmann and Kolisch 2000; Lova and Tormos 2001), it is reasonable to set *L* and *N* at constant levels sufficient to provide leeway for other factors to vary. Third, because *NARLF* assumes fungible resources and would be confounded with *K*, we held *K* constant for all activities. (For example, in a three-project portfolio, if one of the projects uses four types of resources and the other two projects use only one of those types, then *K* = 4, as it also would if all three projects used all four types of resources. We therefore ensure that each activity requires at least a small amount of all four types of resources.) Fourth, we held *F* constant so as not to affect the network density, and because we can readily vary the iteration intensity by changing *P*.

Table 1: Experiment design.

Constant Factors	Setting	Main Factors	Settings
L	3 projects per portfolio	*C*	4 levels: HHH, HHL, HLL, and LLL
N	20 activities per project	*NARLF*	7 levels: -3.0 to 3.0 in increments of 1.0
K	4 types of resources per activity	*MAUF*	11 levels: 60% to 160% in increments of 10%
F	15 feedbacks per project	*P*	3 levels: none (0), low (0.1), and high (0.25)

We designated two levels of project network density, "high" (C = 0.69) and "low" (C = 0.14), and used these to set four types of portfolios: three high-density networks ("HHH"), three low-density networks ("LLL"), and two intermediate combinations. (Figure 1 shows an example "HLL" portfolio.) The *NARLF* and *MAUF* settings follow Kurtulus and Davis (1982): *NARLF* values of -3 and -2 represent highly front-loaded resource loading profiles, whereas values of 2 and 3 represent high back-loading; *MAUF* values from 0.6 to 1.6 represent average levels of resource contention from 60% to 160%. The probability of each feedback arc causing rework for an upstream activity, P, has three levels: none (0), "low" (0.1), and "high" (0.25). P = 0 is an acyclical project, useful for comparison, and we chose the "low" and "high" settings after several precursor trials confirmed their ample distinction. Each realized iteration caused the affected upstream activity to add 50% to its (resource unconstrained) duration. The generator randomly assigns each activity in each project a duration, d_{il} (an integer in [1,9]), and requirements for each of K types of resources, r_{ilk} (an integer in [1,9]). Figures 1-3 show one of the generated portfolios. The full factorial experiment required 4 × 7 × 11 × 3 = 924 portfolios. To distinguish among sources of variance, we used 20 replications for each setting, thus generating 18,480 portfolios (55,440 projects). Although it would be infeasible to conduct a comprehensive study of this type entirely with data from real projects, the test projects we use are realistic in comparison to actual projects, spanning the ranges of variation in several key characteristics representative of real projects and portfolios.

3.3 Priority Rules

We include a set of 20 single- and multi-project PRs (Table 2) compiled from various sources and analyzed by Browning and Yassine (2010b) for acyclical projects; see their paper for a description of each of these PRs. Previous studies have shown many of these PRs to be optimal. To these we added 11 mostly new PRs that account for some basic characteristics of iterative projects, such as the likelihood of rework loops associated with an activity. Table 3 presents these PRs and a theoretical basis for their expected efficacy. For all 31 PRs, we used FCFS as the first tie-breaker (e.g., when using SOF on two activities of equal duration) and RAN as the second tie-breaker.

Table 2: PRs analyzed by Browning and Yassine (2010b) for acyclical projects.

1. **FCFS**—First Come First Served	11. **EDDF**—Earliest Due Date First
2. **SOF**—Shortest Operation First	12. **LCFS**—Last Come First Served
3. **MOF**—Maximum (longest) Operation First	13. **MAXSP**—Maximum Schedule Pressure
4. **MINSLK**—Minimum Slack	14. **MINLFT**—Minimum Late Finish time
5. **MAXSLK**—Maximum Slack	15. **MINWCS**—Minimum Worst Case Slack
6. **SASP**—Shortest Activity from Shortest Project	16. **WACRU**—Weighted Activity Criticality & Resource Utilization
7. **LALP**—Longest Activity from Longest Project	17. **TWK-LST**—MAXTWK & earliest Late Start time (2-pass rule)
8. **MINTWK**—Minimum Total Work content	18. **TWK-EST**—MAXTWK & earliest Early Start time (2-pass rule)
9. **MAXTWK**—Maximum Total Work content	19. **MS**—Maximum Total Successors
10. **RAN**—Random	20. **MCS**—Maximum Critical Successors

Table 3: Additional PRs used in this study.

Priority Rule (PR)	Description and Theoretical Basis
21. **MAXFR**—Maximum number of Feedbacks Received	Prioritizes the activity receiving the most feedbacks, because such activities are more likely to experience first-order rework (when the probability of all feedbacks is equal), under the theory that it is important to front-load iterations and get them out of the way as early as possible (Thomke and Fujimoto 2000; Lévárdy and Browning 2009).
22. **MINFR**—Minimum number of Feedbacks Received	The opposite of MAXFR, included as an alternative hypothesis
23. **MAXWFR**—Maximum Weighted number of Feedbacks Received	Same as MAXFR, but each feedback is weighted by its arc length (calculated as its distance from the diagonal in the DSM), with greater weight given to longer arcs; e.g., a feedback arc from an activity five rows downstream in the DSM is five times larger than a feedback arc from the next activity in the DSM. Longer feedback arcs are more consequential, because they imply a return from an almost completed process towards its beginning, which often spawns a number of second-order rework paths.
24. **MINWFR**—Minimum Weighted number of Feedbacks Received	The opposite of MAXWFR, included as an alternative hypothesis
25. **MAXFGIS**—Maximum number of Feedbacks Generated by Immediate Successors	This PR counts the number of immediate successor activities and multiplies each successor by its number of feedback outputs. Priority is given to the activity with the highest total number of feedbacks caused by immediate successor activities. The theory is that it is important to front-load iterations and get them out of the way as early as possible (Thomke and Fujimoto 2000; Lévárdy and Browning 2009).
26. **MINFGIS**—Minimum number of Feedbacks Generated by Immediate Successors	The opposite of MAXFGIS, included as an alternative hypothesis
27. **MAXRWKP**—Maximum expected Rework Percentage	This PR calculates the joint probability and impact of first-order rework (Browning and Eppinger 2002) due to incoming feedback marks: RWKP = 1 - $\prod_{j=i+1}^{n}\left(1-p_j I_j\right)$. For example, an activity with two incoming feedback marks, each with p = 0.4 and I = 0.5, would have RWKP = 0.36, whereas an activity with three such incoming feedback marks would have a greater RWKP = 0.488. The theory is that it is important to prioritize activities with higher expected amounts of rework to front-load iterations and get them out of the way as early as possible (Thomke and Fujimoto 2000; Lévárdy and Browning 2009).
28. **MINRWKP**—Minimum expected Rework Percentage	The opposite of MAXRWKP, included as an alternative hypothesis
29. **MAXRWK**—Maximum expected Rework	The same as MAXRWKP except that RWKP is multiplied by d_i (the duration of the activity), thus combining MAXRKWP with MOF
30. **MINRWK**—Minimum expected Rework	The opposite of MAXRWKP, included as an alternative hypothesis
31. **MAXSEPTR**—Maximum Shortest Expected Processing Time Ratio	SEPTR = $\frac{d_i}{\sum_{j=1}^{i-1}\left(\frac{p_{ij}}{1-p_{ij}}\right)}$ where d_i is the activity duration. SEPTR was originally developed by Banerjee et al. (2007). We changed the denominator to account only for feedbacks (not feed-forwards), thus accounting only for first-order rework. (If an activity does not cause any feedback, then the denominator is set to 1 to avoid division by zero.) This rule prioritizes longer activities that do not cause rework. In acyclical projects, MAXSEPTR reduces to MOF.

3.4 Objective Functions

We compare results for two objectives. The first (O1) seeks to minimize the individual projects' average percent delay, and the second (O2) seeks to minimize the overall portfolio's percent delay. After defining a due date for each project based on the length of its resource-unconstrained critical path (e.g., A, B, and C for the three projects in Figure 5), we then compared the actual project and portfolio durations ($A + a$, $B + b$, and $C + c$ in Figure 5) against that point, as given by the following equations:

$$\text{Projects' Average Percent Delay} = \frac{\frac{a}{A} + \frac{b}{B} + \frac{c}{C}}{3} \tag{O1}$$

$$\text{Overall Portfolio's Percent Delay} = \frac{\text{Max}(A + a, B + b, C + c) - \text{Max}(A, B, C)}{\text{Max}(A, B, C)} \tag{O2}$$

Using percent delay improves comparisons across projects and portfolios with varied base durations. For example, a five-day delay on a one-day project is usually perceived differently than a five-day delay on a 500-day project. O1 is a more sensitive measure than O2, because the latter is affected only by delays to the longest project in the portfolio. These two objectives provide different perspectives on portfolio management. We solved each of the 18,480 test portfolios for both objectives with each of the 31 PRs, thus producing 1,145,760 portfolio-level and 3,437,280 project-level outcomes.

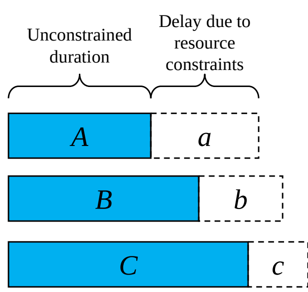


Figure 5: Example portfolio of three projects, each with duration proportional to bar width.

3.5 Solution Procedure

Each problem consists of a portfolio of $l = 2,\ldots, L$ projects with identical start times, where each project consists of $i = 1,\ldots,N_l$ activities with deterministic duration d_{il}. Activity i becomes eligible to start when all of its predecessors (as determined by the feed-forward arcs) have finished. When an activity finishes, any feedback arc from it to an upstream activity is probabilistically checked for occurrence, and if it occurs then the upstream activity is reactivated (at half of its original duration). Each activity requires r_{ik} units of resource type $k \in K$ during each period of its duration. The portfolio has a renewable capacity of R_k for each resource k. If the set of eligible activities requires more than R_k for any k, then non-priority activities must wait for resource availability. Activity preemption is not allowed except when upstream activities must be reworked.

To determine the project and portfolio durations, we use an algorithm that computes the sum of the discrete time steps required to complete all of the constituent activities. At the beginning of each step, the algorithm separates the activities into three disjoint sets: the *completed set*, ***C*** (finished activities), the *decision set*, ***D*** (ongoing and un-started activities with predecessors in ***C***), and the *ineligible* set, ***I*** (activities with predecessors in ***D***). If the available resources are sufficient to perform all of the activities in ***D***, then the algorithm puts these into a fourth set, the *active set*, ***A***. If not, then a PR is used to rank the activities in ***D***. (The priority of each activity in ***D*** is dynamically re-determined at each decision point—a parallel schedule generation scheme (Boctor 1990).) The highest-ranking activities are then added to ***A*** as resources allow. The length of the time step depends on the shortest activity (or activities) in ***A***. When one or more activities in ***A*** finish, these activities are moved to ***C***, and feedback loops are checked to see if any activities must be removed from ***C*** and returned to ***D***. (When a reworked activity finishes, its feed-forward arcs are also checked for potential second-order rework.) Activities in ***I*** are also then checked for potential transfer to ***D***. The procedure is complete when all activities are in ***C***.

Overall, our methodology is rigorous, well-grounded in the literature, and appropriate for a complex, large-scale situation that is intractable to address with analytical approaches. Created with a proven generator, our suite of test problems spans a carefully specified distribution of project and portfolio characteristics that have been shown to matter in previous studies. We test the performance of a mix of novel and proven PRs in terms of two objective functions. We use this established approach to explore, for the first time, PR performance in iterative PD project portfolios.

4. Results

4.1 Results for Projects' Average Percentage Delay (O1)

As a baseline, Figure 6 shows the average performance (O1) and 95% confidence intervals for the 20 older PRs on all 18,480 portfolios (55,440 projects) *without iteration* (P = 0). TWK-LST (Lova and Tormos 2001) provides the best overall results for the acyclical projects. However, several popular and seemingly efficacious PRs such as MINSLK, SASP, SOF, EDDF, and MCS actually perform quite poorly (worse than RAN). (For example, several single- and multi-project studies (Fendley 1968; Davis and Patterson 1975; Allam 1988; Boctor 1990; Cohen et al. 2004) found MINSLK to be optimal, and Kurtulus and Davis (1982) recommended SASP in a multi-project setting.) The less-effective PRs tend to favor either the shortest or longest project or focus on the number of successors within a single project, whereas PRs that consider the entire portfolio (such as TWK-LST, MAXTWK, and MINWCS) generally perform well on O1.

In comparison, Figure 7 shows results for all 31 PRs in projects with *high iteration* (P = .25). These results differ from Figure 6. Cyclicality matters in the choice of PR. Of course, the high iteration increases the overall average percent delay (it almost doubles) and its variance. Here, the best overall PR is MINLFT, although it is statistically tied (at 95% confidence) with SASP, MAXSP, MINWCS, and FCFS. Again, the popular PRs MINSLK, MCS, EDDF, and SOF perform quite poorly (worse than RAN), although SASP improves considerably in comparison with the acyclical results. TWK-LST and MAXTWK, which performed well in the acyclical case, fare much worse under iteration. The salient finding here is that *the best PRs for iterative*

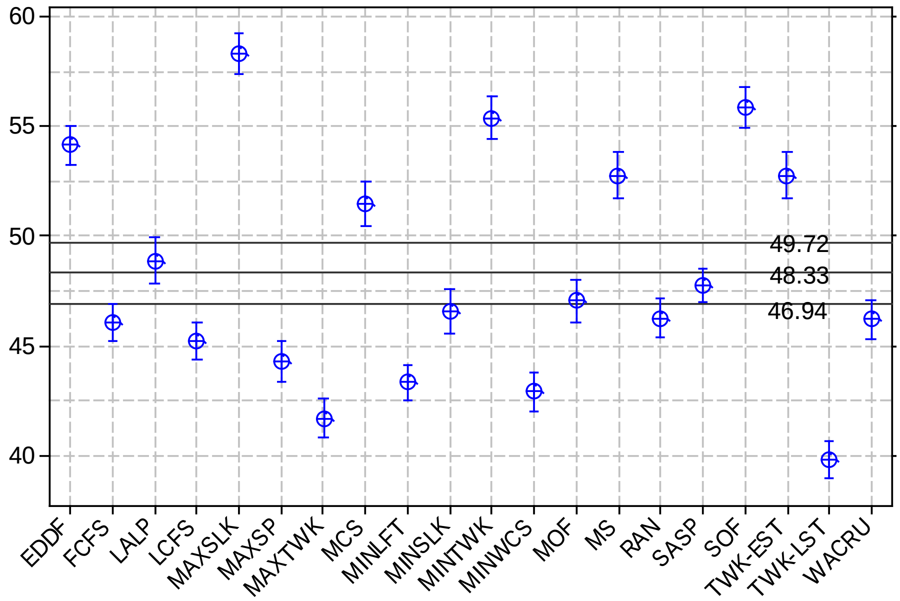


Figure 6: Mean performance (O1) and 95% confidence intervals for 20 PRs on all portfolios *without* iteration.

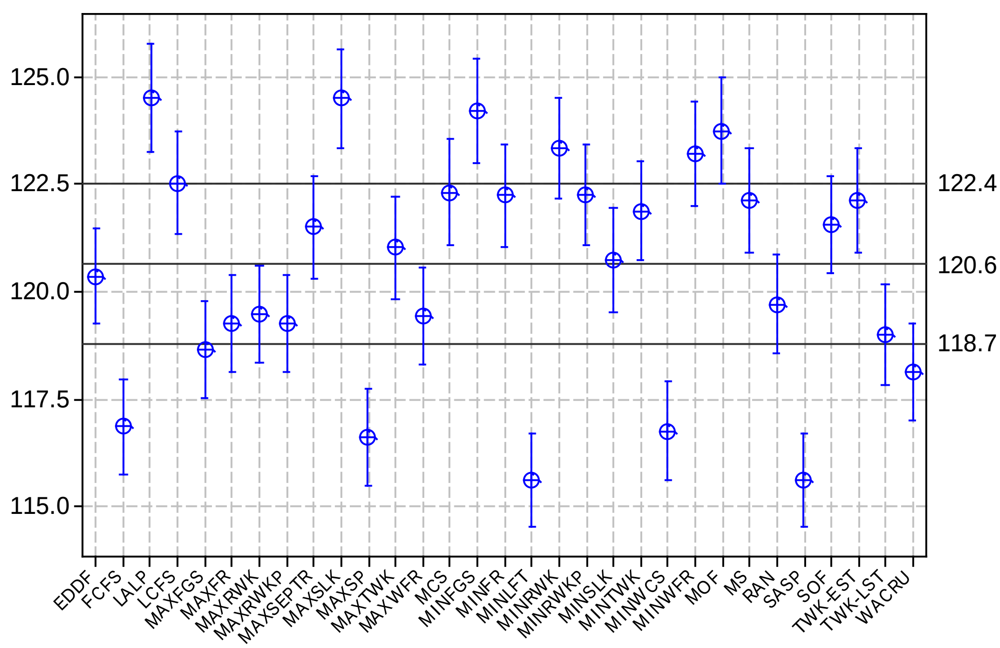
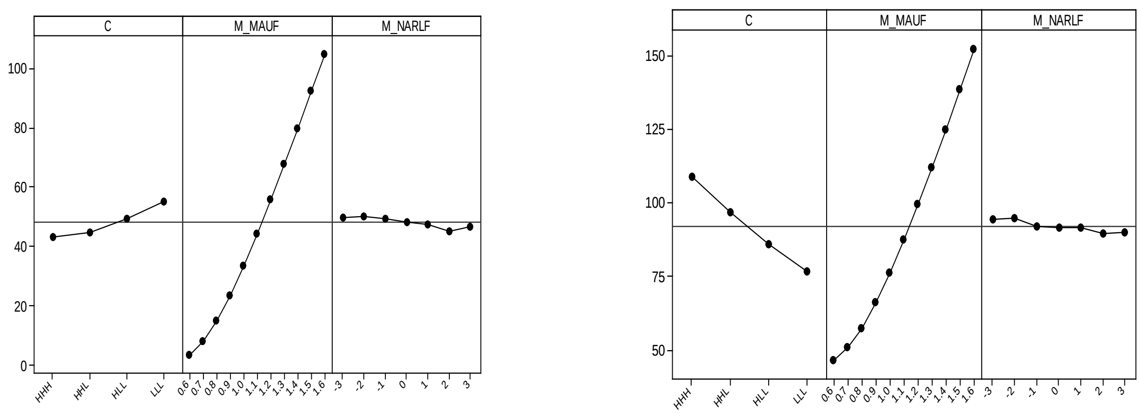


Figure 7: Mean performance (O1) and 95% confidence intervals for 31 PRs on all portfolios with *high* iteration.

projects (such as SASP and MINLFT) are different than the best PR for acyclical projects (TWK-LST). Only MINLFT and MINWCS are fairly robust across both cyclical and acyclical cases. Thus, managers who have a preferred PR for all types of projects could be successful on some projects but get into unnecessary trouble on others. In particular, iterative PD projects should not be managed the same way as acyclical projects. Also, surprisingly, none of the new PRs (Table 2) perform better than the best of the older ones. Therefore, we find no support for the hypotheses that the PRs in Table 2, which are based on relatively simple accounts of feedbacks and expected amounts of rework, should be used.



(a) No Iteration (*P* = 0) **(b) Low and High Iteration (*P* > 0)**

Figure 8: Relative effects of *C*, *MAUF*, and *NARLF* settings on O1, with and without iteration.

The results in Figure 7 are averaged across all combinations of network density (*C*), resource contention (*MAUF*), and resource distribution (*NARLF*), but knowing something about these project and portfolio characteristics can further refine the choice of PR. The main effect plots in Figure 8 provide insight into the implications of *C*, *MAUF*, and *NARLF*. The effects of *MAUF* and *NARLF* are consistent in both the iterative and non-iterative cases: increasing the amount of resource contention in the portfolio (*MAUF*) always increases the average percent delay, and resource contentions early in the portfolio (caused by front-loaded *NARLF*) cause more overall delay than those occurring later. However, iteration has a very interesting interaction with *C*. Without iteration, decreasing the network density of the projects in the portfolio (from HHH to LLL) increases O1, because the increased precedence constraints in a high-density network increase its expected duration (which *decreases* its delays *as a percentage* of this greater expected duration). Also, the more sequential nature of activities in high-density networks decreases their amount of resource contention with other projects in the portfolio. On the other hand, low-density networks have fewer precedence constraints and thus more natural concurrency among activities, which makes them shorter without resource contention but increases the likelihood of resource constraints, both of which increase their effect as a percentage delay. However, *in iterative projects this behavior reverses*! High-density networks increase the likelihood of rework propagation and delay (Chalupnik et al. 2008), even when measured as a percentage of their longer planned duration. The greater interconnectedness of high-density networks greatly increases the number of paths for the propagation of second-order rework. (Recall that *C* does not account for the number of feedbacks, only the number of non-redundant feed-forwards. All of the feed-forward arcs in our test problems are non-redundant.) Figure 9 exhibits the *C*-*P* interaction in greater detail and clearly shows the effect on *C* of increasing *P*. *The effects of iteration are clearly exacerbated by high-density networks*. Thus, once again, iterative PD projects should not be managed the same way as projects without iteration.

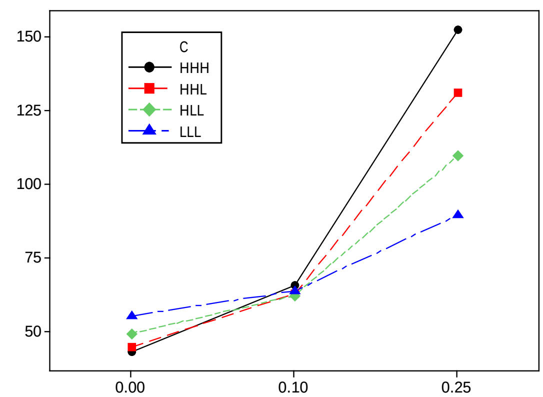
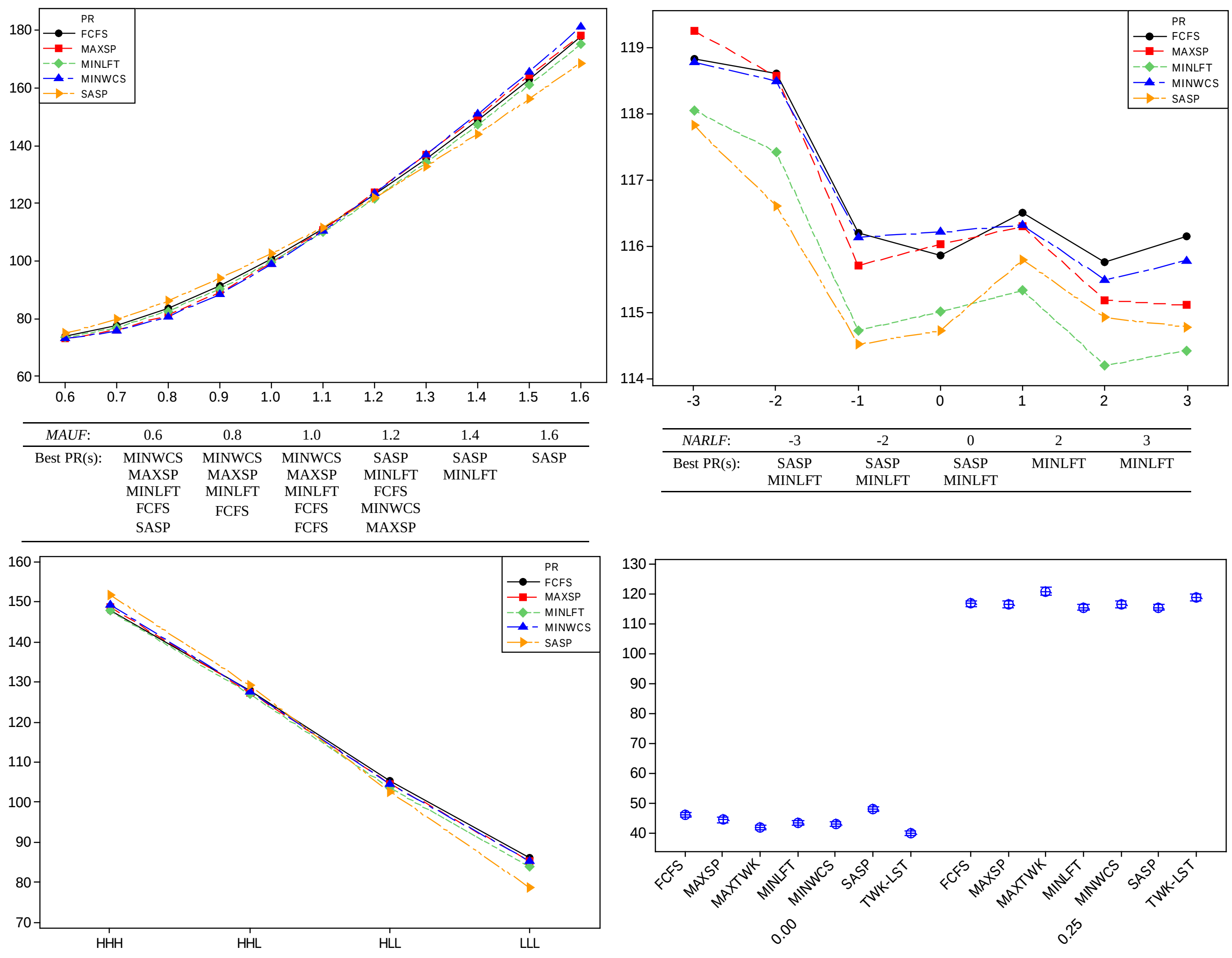


Figure 9: Interactions between *C* and *P* for O1.

An analysis of variance (ANOVA) revealed that the five factors *C*, *P*, *MAUF*, *NARLF*, and the choice of PR—as well as all of their two-way interactions—were highly significant ($p < 0.001$) and collectively accounted for 91.3% of the variation. *MAUF* and *P* explained 44.9% and 37.2% of the variation, respectively. The interesting interaction between *C* and *P* explained an additional 5.7%, *C* accounted for 1.7%, and no other individual source explained more than 0.7%. Obviously, increasing resource constraints and iteration will dramatically increase O1, so the large effects of *MAUF* and *P* were expected. Nevertheless, the other factors provide managers with the capability to have a highly significant effect on the outcome.

How do *C*, *P*, *MAUF*, and *NARLF* affect PR performance in *highly iterative* projects such as PD? Consider the plots in Figure 10 (which, for increased legibility, include only the PRs that perform better than RAN). First, the *MAUF* interaction plot in the upper-left indicates that, as expected, in the choice of PR makes little difference when resource constraints are low. However, as *MAUF* increases, the performance disparity grows, and the choice of PR matters more. We performed a *t*-test on the mean of each PR at selected levels of *MAUF*, resulting in the table shown below the *MAUF* plot in Figure 10, which lists the statistically tied (at 95% confidence) best PRs (from lowest to highest mean result for O1). Whereas several PRs are statistically tied at lower *MAUF* levels, SASP emerges as the best PR at high levels of resource contention in highly iterative projects. Second, the *NARLF* interaction plot shown in the upper-right of Figure 10 (note that the *y*-axis scale differs from the *MAUF* plot) exhibits the robustness of MINLFT across varied resource loading distributions in highly iterative projects, although SASP also performs very well on front-loaded portfolios. Third, the lower-left section of Figure 10 shows the effects of varying network density, where MINLFT is one of the better performers in iterative projects with high-density networks, whereas SASP becomes the clear favorite in low-density cases. Fourth, regarding iteration in general (*P*), MINLFT and SASP perform best for high-iteration projects (as seen in Figure 7).



MAUF:	0.6	0.8	1.0	1.2	1.4	1.6
Best PR(s):	MINWCS MAXSP MINLFT FCFS SASP	MINWCS MAXSP MINLFT FCFS	MINWCS MAXSP MINLFT FCFS FCFS	SASP MINLFT FCFS MINWCS MAXSP	SASP MINLFT	SASP

NARLF:	-3	-2	0	2	3
Best PR(s):	SASP MINLFT	SASP MINLFT	SASP MINLFT	MINLFT	MINLFT

C:	HHH	HHL	HLL	LLL
Best PR(s):	FCFS MINLFT MAXSP MINWCS	MINLFT MAXSP FCFS MINWCS SASP	SASP MINLFT MAXSP MINWCS FCFS MAXSP	SASP

P:	None (0)	High (0.25)
Best PR(s):	TWK-LST	MINLFT SASP MAXSP MINWCS FCFS

Figure 10: Effects of *MAUF*, *NARLF*, and *C* on PR performance (O1) with *high iteration* as well as effects of *P* (showing only PRs better than RAN).

Overall, these results demonstrate that *the choice of best PR depends on project and portfolio characteristics*. A manager who uses a single PR for all occasions will perform unnecessarily poorly most of the time. Table 4 summarizes the PR results for O1 in terms of the project and portfolio characteristics. SASP tends to perform well in project portfolios with medium to high *MAUF* when *C* is low, especially when *P* is high. However, SASP does not perform well when *MAUF* is low or in acyclical projects with high network density. MINLFT performs better than SASP in high-*MAUF* situations with high network density and iteration. For acyclical project portfolios, TWK-LST performs well in most conditions, although SASP is a better choice even for acyclical projects with high *MAUF* and low *C*. For all types of project portfolios with low- to medium-*MAUF*, MINWCS provides fairly good performance. In the next section, we will simplify these results to facilitate managerial applications, but before that we will consider the results for O2.

Table 4: Summary of results for O1 with zero or high iteration*

	Front-loaded Resources *NARLF* = {-3, -2}				**Resources not front- nor back-loaded** *NARLF* = {-1, 0, 1}				**Back-loaded Resources** *NARLF* = {2, 3}			
	***C* = HHH**		***C* = LLL**		***C* = HHH**		***C* = LLL**		***C* = HHH**		***C* = LLL**	
Resource Contention	**Acyclical**	**Iterative**	**Acyclical**	**Iterative**	**Acyclical**	**Iterative**	**Acyclical**	**Iterative**	**Acyclical**	**Iterative**	**Acyclical**	**Iterative**
Low (*MAUF* = 0.6 - 0.8)	All PRs *except* MAXSLK	MINWCS MAXSEPTR MAXSP MINSLK TWK-LST	MINWCS MAXSP TWK-LST MINSLK **MINLFT** MAXSEPTR MAXTWK MOF	MINWCS MAXSP **MINLFT** TWK-LST MINSLK FCFS MAXTWK WACRU	MAXSP MINWCS MINSLK MOF MAXSEPTR LALP TWK-LST	MINWCS MAXSP MAXSEPTR MINSLK TWK-LST **MINLFT**	MINWCS MAXSP MINSLK TWK-LST MOF MAXSEPTR	MINWCS MAXSP MINSLK **MINLFT** TWK-LST	MINWCS MAXSP MINSLK MAXSEPTR MOF LALP	MINWCS MAXSEPTR TWK-LST MAXSP MINSLK	MINWCS MAXSP MINSLK TWK-LST **MINLFT**	MINWCS MAXSP MINSLK **MINLFT**
Medium (*MAUF* = 1.0 - 1.2)	TWK-LST MAXTWK	FCFS **MINLFT** MINWCS MAXFGS MAXSP	TWK-LST **SASP** MINWCS **MINLFT** MAXTWK MAXSP FCFS	**SASP** **MINLFT** MINWCS	TWK-LST MAXTWK	FCFS **MINLFT** MINWCS MAXFGS MAXSP	TWK-LST **SASP** **MINLFT** MINWCS MAXTWK MAXSP FCFS	**SASP** **MINLFT**	TWK-LST MINWCS MINSLK MAXSP	FCFS **MINLFT** MINWCS MAXSP MINSLK	TWK-LST MINWCS MAXSP **MINLFT** MAXTWK MINSLK	**SASP** **MINLFT** MAXSP
High (*MAUF* = 1.4 - 1.6)	TWK-LST MAXTWK LCFS	FCFS **MINLFT** **SASP** MAXFGS WACRU	**SASP**	**SASP**	TWK-LST MAXTWK	**MINLFT** FCFS **SASP**	**SASP**	**SASP**	TWK-LST LCFS MAXTWK	**MINLFT** FCFS WACRU	**SASP** LCFS	**SASP**

*Multiple entries in each cell are statistically tied (at 95% confidence) but listed in order of increasing means (i.e., the best PR is listed first).

4.2 Results for Portfolio Percentage Delay (O2)

We performed similar analyses for O2, although we will focus here only on the key differences from the O1 results. As a baseline, Figure 11 shows the average performance for O2 and 95% confidence intervals for the 20 older PRs on all 18,480 portfolios (55,440 projects) *without iteration* (P = 0). MINWCS, which also performed fairly well for O1 at lower *MAUF* levels, provides the best overall results for acyclical portfolios. LALP and MS perform much better for O2 than for O1. TWK-LST and MAXSP performed fairly well according to both O1 and O2, whereas EDDF, MINTWK, and SOF performed poorly (worse than RAN) with both objectives. The more effective PRs for O2 favor the longest project, because delays to the shorter projects in the portfolio do not affect O2—hence, SASP performs especially poorly.

In comparison, Figure 12 shows the results for all 31 PRs in *highly iterative* projects (P = 0.25). These results differ from Figure 11: cyclicality also matters in the choice of PR for O2. With high iteration, the best overall PR is TWK-EST, although it is statistically tied (at 95% confidence) with MS, WACRU, MCS, and

MAXFGS. The first major finding here is that the best PRs for highly iterative portfolios (TWK-EST and MS) differ from the best PR for acyclical portfolios (MINWCS). Thus, *PD portfolio managers should use different PRs than managers of non-iterative project portfolios*. The second major finding here is that, both with and without iteration, the best PRs for O2 are very different than the best ones for O1. With high iteration, the best PRs for O1 (Figure 7) are SASP, MINLFT, MAXSP, MINWCS, and FCFS, but all of these perform very

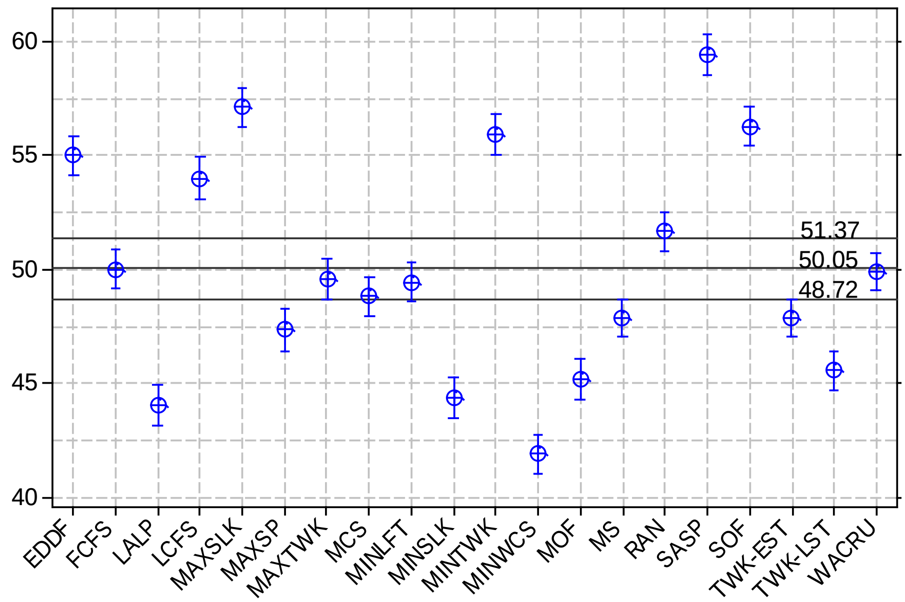


Figure 11: Mean performance (O2) and 95% confidence intervals for 20 PRs on all portfolios *without* iteration.

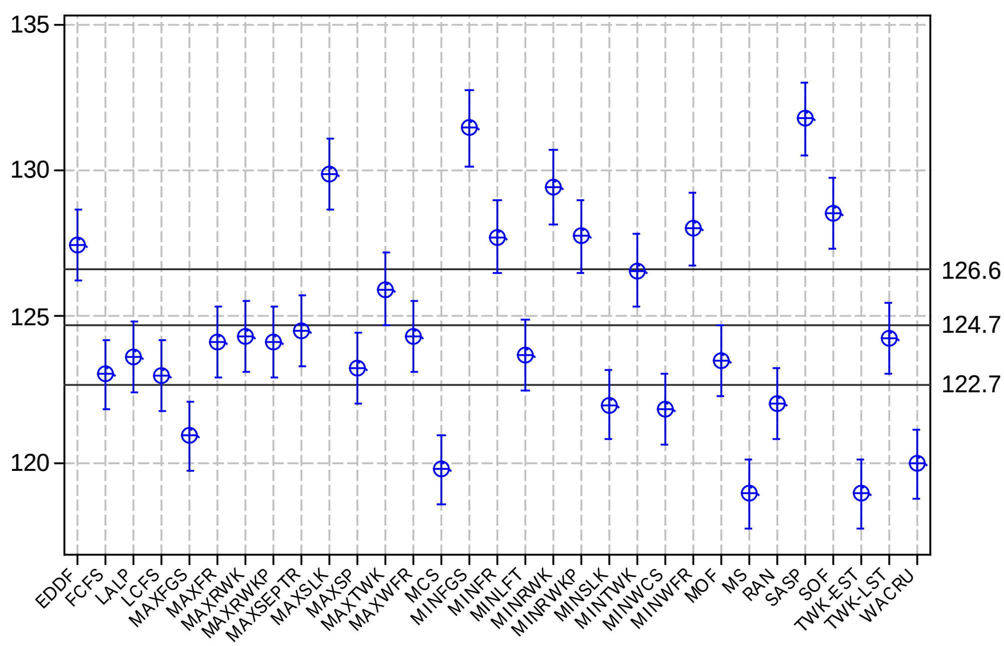
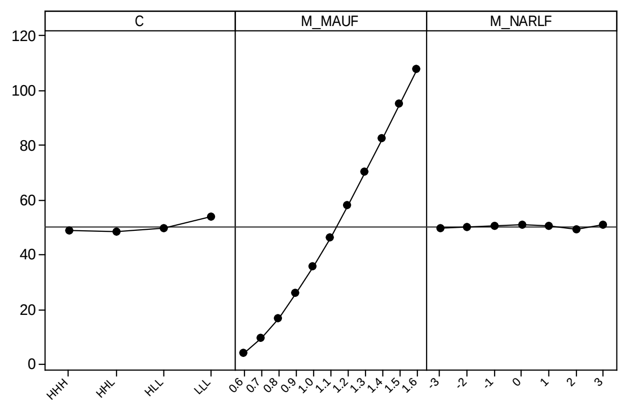


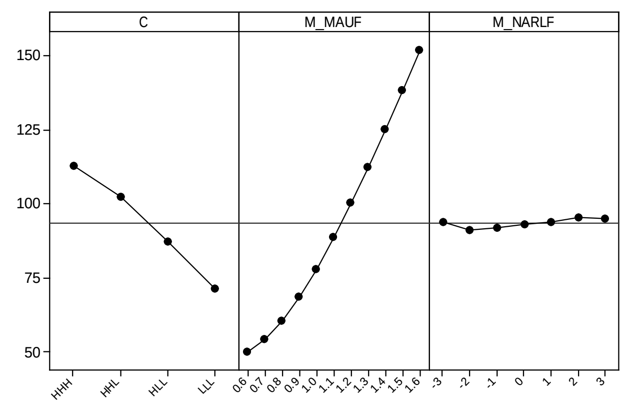
Figure 12: Mean performance (O2) and 95% confidence intervals for 31 PRs on all portfolios with *high* iteration.

poorly for O2. Hence, in both acyclical and iterative portfolios, *the best PR depends on the objective*. For example, in a portfolio of highly iterative PD projects, the SASP rule would minimize the average percent delay to each project, whereas the TWK-EST rule would minimize the average percent delay to the entire portfolio. Of course, pursuit of different objectives and the use of different decision rules can create tension between managers in organizations.

As with O1, the best PR on average for O2 may not always be best under particular combinations of project and portfolio characteristics. The main effect plots for network density (*C*) and resource contention (*MAUF*) for O2 (Figure 13) exhibit the same behaviors as for O1 in both the iterative and acyclical cases, but the effect of resource distribution (*NARLF*) differs for O2, both with and without iteration. From the point of view of O1, greater delays are caused by front-loaded needs for resources, because these create early contentions that propagate through the remainder of the projects. *For O2, however, greater delays are caused by back-loaded resource distributions, because these imply a greater likelihood of delay to the longest project, and thus to the overall portfolio*.



(a) No Iteration ($P = 0$)

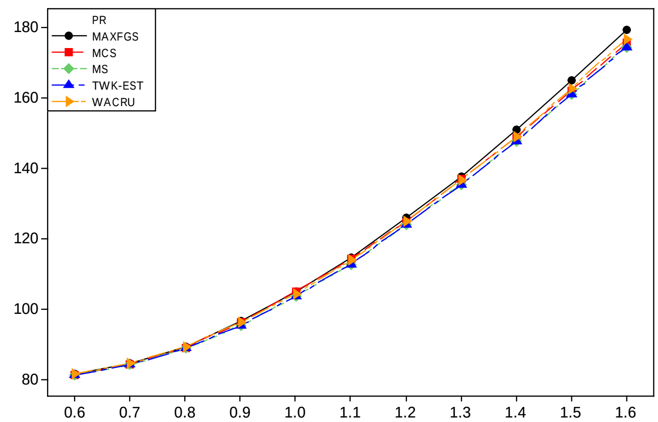


(b) Low and High Iteration ($P > 0$)

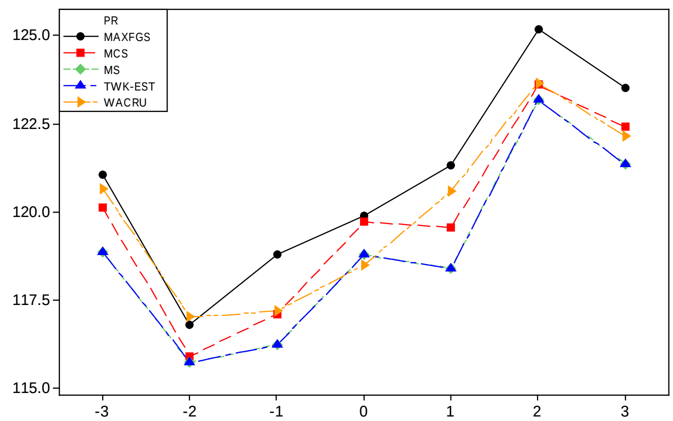
Figure 13: Relative effects of *C, MAUF*, and *NARLF* settings on O2, with and without iteration.

How do *C*, *P*, *MAUF*, and *NARLF* affect PR performance with respect to O2 in iterative portfolios? ANOVA showed that all five main factors and their two-way interactions are highly significant ($p < 0.001$), account for similar amounts of variation in O2 as they do for O1, and exhibit similar overall trends in two-way interactions. Moreover, the plots in Figure 14 show the main factor results for the PRs performing better than RAN in project portfolios with high iteration. As with O1, the choice of PR becomes more consequential as *MAUF* increases. In the case of *NARLF*, we see the detrimental effects of resource back-loading observed in Figure 13, but this does not much alter the PRs' relative performance. WACRU performs best at the intermediate levels of network density, but TWK-EST is best at both extremes (and a close second at the intermediate levels). Without iteration, the minimum slack-related PRs (MINSLK and MINWCS) perform well, but high iteration improves the performance of PRs based on the most successors (MS and MCS) and

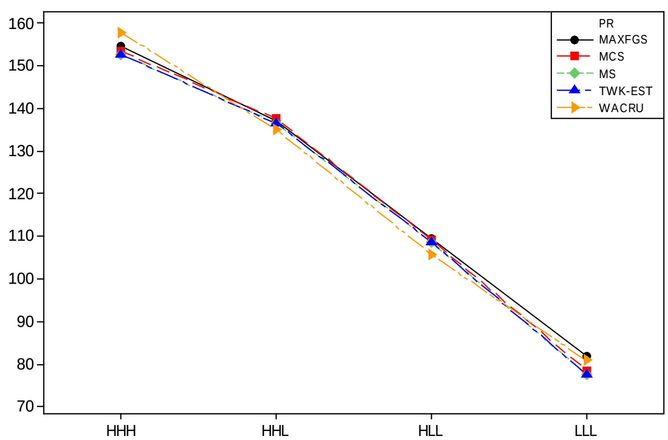
TWK-EST. As with O1, none of the new PRs (Table 2) perform better than the best of the older PRs, so we find no support for the hypotheses that PRs based only on relatively simple accounts of feedbacks and expected amounts of rework should be used for O2. Thus, with O2 as well as with O1, *the choice of PR depends heavily on project and portfolio characteristics such as network density, iteration intensity, and amount of resource contention*. As with O1, a manager who uses a single PR for all occasions will perform unnecessarily poorly most of the time.



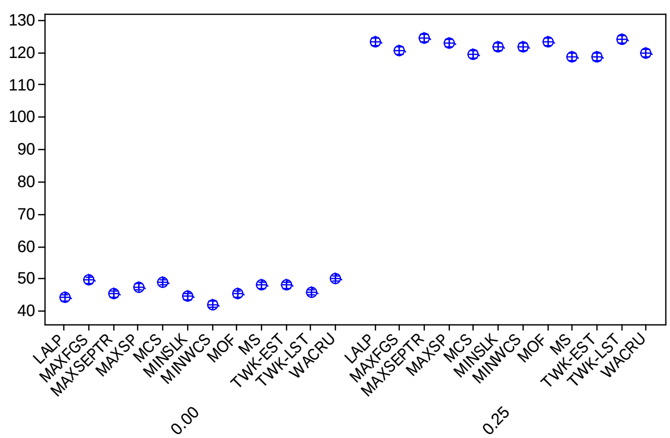
MAUF:	0.6	0.8	1.0	1.2	1.4	1.6
Best PR(s):	MCS TWK-EST MS WACRU MAXFGS	MCS TWK-EST MS WACRU MAXFGS	TWK-EST MS WACRU MCS MAXFGS	TWK-EST MS MCS WACRU MAXFGS	TWK-EST MS MCS WACRU MAXFGS	TWK-EST MS MCS WACRU



NARLF:	-3	-2	0	2	3
Best PR(s):	TWK-EST MS MCS WACRU MAXFGS	TWK-EST MS MCS MAXFGS WACRU	WACRU TWK-EST MS MCS MAXFGS	TWK-EST MS MCS WACRU MAXFGS	TWK-EST MS WACRU MCS MAXFGS



C:	HHH	HHL	HLL	LLL
Best PR(s):	TWK-EST MS MCS MAXFGS WACRU	WACRU TWK-EST MS MAXFGS MCS	WACRU TWK-EST MS MCS MAXFGS	TWK-EST MS MCS WACRU MAXFGS



P:	None (0)	High (0.25)
Best PR(s):	MINWCS LALP MINSLK	MS TWK-EST MCS WACRU MAXFGS

Figure 14: Effects of *MAUF*, *NARLF*, and *C* on PR performance (O2) with *high iteration* as well as effects of *P* (showing only PRs better than RAN).

Table 5 summarizes the PR results for O2 in terms of project and portfolio characteristics. The choice of PR is less consequential at lower levels of resource contention (*MAUF*), although LALP, which prioritizes the longest project in the portfolio, performs best for O2 regardless of other characteristics. Good PR selection becomes more important as *MAUF* increases, because fewer PRs do well and the other project and portfolio characteristics have a greater bearing on the outcome. For portfolios with acyclical projects, MINWCS provides fairly robust performance in most cases, although one can do even better with MS in certain cases of high *MAUF* and low network density (*C*). For portfolios with iterative projects, LALP remains a good choice at moderate levels of *MAUF* and high levels of *C*, and MS works well for moderate *MAUF* and low *C*. At extreme levels of resource contention in portfolios with iterative projects, TWK-EST and MS work best. One of the new PRs, MAXSEPTR, performs best for portfolios with iterative projects with high *MAUF* and *C* and back-loaded resources (high *NARLF*).

Table 5: Summary of results for O2 with zero or high iteration*

	Front-loaded Resources *NARLF* = {-3, -2}				**Resources not front- nor back-loaded** *NARLF* = {-1, 0, 1}				**Back-loaded Resources** *NARLF* = {2, 3}			
	***C* = HHH**		***C* = LLL**		***C* = HHH**		***C* = LLL**		***C* = HHH**		***C* = LLL**	
Resource Contention	**Acyclical**	**Iterative**	**Acyclical**	**Iterative**	**Acyclical**	**Iterative**	**Acyclical**	**Iterative**	**Acyclical**	**Iterative**	**Acyclical**	**Iterative**
Low (*MAUF* = 0.6 - 0.8)	LALP MINSLK **MINWCS** MAXSP MAXSEPTR MOF	LALP MAXSEPTR MINSLK MAXSP **MINWCS** MOF TWK-LST **TWK-EST**	LALP MINSLK **MS** **TWK-EST**	LALP MCS **MS** **TWK-EST** MINSLK	LALP MINSLK **MINWCS** MOF MAXSP	LALP MINSLK **MINWCS** MOF MAXSEPTR MAXSP MCS TWK-LST **TWK-EST** MAXTWK MAXFGS MINLFT MAXFR MAXRWKP FCFS	LALP **MINWCS** MINSLK MAXSP	LALP MCS MINSLK **MS** **TWK-EST** MOF **MINWCS** MAXSEPTR	LALP MOF MAXSEPTR **MINWCS** MINSLK MAXSP	LALP MAXSEPTR MINSLK MAXSP MOF **MINWCS** TWK-LST MCS MAXTWK MAXFGS **MS** **TWK-EST** MINLFT MAXRWKP MAXFR	LALP **MINWCS** MINSLK MAXSP MAXSEPTR	LALP MINSLK **MINWCS** MCS **MS** **TWK-EST** MAXSEPTR MOF TWK-LST WACRU MAXTWK MINLFT LCFS MAXFGS MAXRWK
Medium (*MAUF* = 1.0 - 1.2)	**MINWCS** LALP MAXSEPTR MOF MINSLK	LALP MAXSEPTR **MS** **TWK-EST** MINSLK MCS TWK-LST **MINWCS**	**MS** **TWK-EST** **MINWCS** MCS LALP	**MS** **TWK-EST** MCS	**MINWCS** LALP MINSLK MAXSEPTR MOF	LALP **MS** **TWK-EST** MINSLK MAXSEPTR MCS **MINWCS** MAXSP MOF TWK-LST	**MINWCS** **MS** **TWK-EST** MCS	**MS** **TWK-EST** MCS MOF	**MINWCS** MAXSEPTR MOF MINSLK LALP	LALP MAXSEPTR **MINWCS** MINSLK MOF MAXSP MAXFGS TWK-LST **MS** **TWK-EST**	**MINWCS** **MS** **TWK-EST** LALP MINSLK MAXSP	**MS** **TWK-EST** MINSLK MOF MCS WACRU LCFS MAXSEPTR MAXSP TWK-LST MINLFT MAXTWK LALP
High (*MAUF* = 1.4 - 1.6)	**MINWCS** MOF MAXSEPTR MINSLK TWK-LST LALP	**TWK-EST** **MS** MCS MAXSEPTR MAXFGS LALP	**MS** **TWK-EST** MCS WACRU **MINWCS**	**TWK-EST** **MS** MCS WACRU	**MINWCS** TWK-LST MAXSEPTR MOF MINSLK LALP	**MS** **TWK-EST** MCS	**MS** **TWK-EST** **MINWCS** MCS WACRU MINLFT	**MS** **TWK-EST** MCS LCFS MOF WACRU	**MINWCS** MAXSEPTR MOF TWK-LST MINSLK	MAXSEPTR **TWK-EST** **MS** LALP MINSLK	**MINWCS** **TWK-EST** **MS** WACRU MINLFT MCS EDDF FCFS	**TWK-EST** **MS** LCFS MOF WACRU MCS

*Multiple entries in each cell are statistically tied (at 95% confidence) but listed in order of increasing means (i.e., the best PR is listed first).

5. Managerial Implications

As noted in the Introduction, many practicing managers of PD projects do not have the knowledge or resources to build a complete and accurate model of the entire activity network. This prevents them from practically applying many advanced optimization techniques developed for such networks. However, managers are much more likely to be able to ascertain reasonable estimates of the expected resource loading profile, amount of resource contention, activity network density, and iteration intensity. Although activity network models are necessary to calculate exact measures of these characteristics, they can often be estimated at the level of "high," "medium," or "low" without such models, or with partial models. For example, managers can get an idea of a project's network density simply by asking whether its tasks can easily be done in parallel (low density) or must be done more sequentially due to a high number of dependencies (high density). The general distribution of resources (front- or back-loading) can be ascertained without too much effort, and the rough amount of resource contention can be estimated as "low," "medium," or "high" with reasonable knowledge of resource availability versus portfolio needs. To facilitate usage of the readily obtainable information about project and portfolio characteristics, we distilled our results into simplified decision tables for O1 and O2 (Table 6). Thus, even without complete network models, a manager can still benefit from many PRs and select the best PR for a particular situation.

Table 6: Recommended PRs for O1 and O2 given project and portfolio characteristics.

O1		*Acyclical*		*Iterative*	
		High Density	**Low Density**	**High Density**	**Low Density**
Resource Contention	**Low**	MINWCS	MINWCS	MINWCS	MINWCS
	Medium	TWK-LST	TWK-LST	FCFS	SASP
	High	TWK-LST	SASP	FCFS	SASP

O2		*Acyclical*		*Iterative*	
		High Density	**Low Density**	**High Density**	**Low Density**
Resource Contention	**Low**	LALP	LALP	LALP	LALP
	Medium	MINWCS	MS/MINWCS	LALP	MS
	High	MINWCS	MS/MINWCS	MS	MS

Looking at Table 6, we recommend that a manager pursuing O1 (minimizing the average percentage delay to all projects in the portfolio) use "minimum worst case slack" (MINWCS) under conditions of low resource contention, regardless of cyclicality, resource distribution, or network density. (Resouce distribution is not shown in Table 6 because in most cases it did not alter the results with respect to the other characteristics.) MINWCS is an enhanced version of the commonly used "minimum slack" rule. However, it would be a mistake to use this PR all the time. As resource contention increases for acyclical projects, we recommend "maximum total work with latest start time" (TWK-LST), except when resource contention is very high and the network density is low, where we prefer "shortest activity from the shortest project" (SASP). TWK-LST considers each activity against the entire portfolio, not just against its own project. SASP is the only PR whose performance does not dramatically decrease with decreasing network density. For iterative projects, as resource contention increases, "first-come first-served" (FCFS) is best for high-density networks, and SASP is best for low-density networks. Both FCFS and SASP tend to process jobs quickly and thus "get on" with the realization of any iterations, whereas many of the more complex PRs that evaluate more information from downstream in the network may be getting confounded by the iterations.

For O2 (minimizing the percentage delay to the overall portfolio) we recommend "longest activity from the longest project" (LALP) when resource contention is low, regardless of cyclicality, resource distribution, or network density. Because O2 focuses on the longest project in the portfolio, it is a less sensitive objective than O1, but it also implies that PRs such as LALP that focus on the longest project are expected to do well. Again, however, it would be a mistake to use this PR all the time for O2. As resource contention increases, MINWCS becomes more appropriate for acyclical projects, although MS is better for low-density networks without back-loaded resource distributions. For portfolios with iterative projects, LALP remains best for moderate levels of resource contention in high-density networks, whereas MS is our recommendation for moderate levels of contention in low-density networks and all levels of network density with high contention. LALP and MS also bring the advantage of being relatively easier to determine than many of the more sophisticated PRs.

Although our recommendations in Table 6 stem directly from the full results in Tables 4 and 5, we make some allowances for simplicity. For instance, in a few cases of near-ties in Tables 4 or 5, we recommend a PR that appears elsewhere in Table 6 (to reduce the number of different PRs appearing in the simplified recommendations), or we select a PR (like MS) that is relatively easy for a manager to determine (as compared to more complex PRs such as "maximum total work with earliest start time" [TWK-EST]). Thus, Table 6 represents our attempt to distill a complex set of results into a relatively simple form implementable by practicing managers. Again, managers need not quantify an exact amount of resource contention, resource distribution, network density, or iteration intensity to benefit from these findings. Rather, they need only determine a rough amount of each to locate their project portfolios in the tables. And because the same PR often appears across several cells of Table 6, these recommendations are fairly robust against particular characteristics. Nevertheless, even from these simplified results, it should remain clear that the best PR for a situation can vary, and that a manager using a single PR in all cases would be suboptimal in many projects or portfolios, especially when resources are contentious. Also, we recognize that the determination of some PRs requires more information about the activity network than others. For example, while determining the "most successors" (MS) to an activity requires only a small portion of the network (a portion that is very likely to be available at the time of PR implementation), other PRs such as MINWCS require further information. In the absence of sufficient information to use the recommended PR in Table 6, managers could fall back on the next-best PR from Tables 4 or 5 for which they can acquire adequate estimates.

6. Conclusion

PD portfolio management (PPM) is increasingly important in contemporary practice. Most of the scholarly literature on PPM deals with project selection, but it is important to extend this literature to include portfolio execution and control, including the decisions about resource allocation that present ongoing challenges to practicing managers. Most PD project managers do not have an activity network model to which they might apply the most advanced techniques from the project scheduling literature. Therefore, many make resource allocation decisions on other bases. Some may have a favorite PR, such as "minimum slack" (MINSLK) or "shortest operation first" (SOF), that they always use, and others may essentially use random whim (RAN), yet our study has shown that *all* PRs will perform poorly in many realistic situations. Moreover, managers' objectives may be at odds: minimizing the average delay to all projects requires different PRs than minimizing the delay to the overall portfolio. Without understanding the implications of these differences, managerial tensions could arise.

In addition to contributing to the scholarly literature on PPM, our comprehensive study extends the literature on PD project modeling by exploring the implications of resource limitations on a portfolio of iterative projects. It also extends the scholarly literature on RCMPS to account for iterative PD projects. Moreover, it provides practical guidance to both project and portfolio managers. Out of a large set of 31 PRs, we have shown the most effective PRs in iterative PD projects and how these differ from acyclical projects. Our study has shown that, even if managers employ the best PRs for acyclical projects, they will not achieve the best results in iterative contexts such as PD projects. We have contrasted the results for two different objectives, and we have shown how project and portfolio characteristics—such as front- or back-loading of resources, network density, iteration intensity, and degree of resource contention—affect the results. We have tested 11 new PRs based on the characteristics of iterative projects, finding that these generally fail to surpass 20 established PRs. We have also distilled these results into a practical tool for managers, whereby they may roughly characterize their project(s) in terms of network density, resource contention, resource distribution, and iteration intensity, and then down-select to a PR that is likely to prove efficacious in their situation. Overall, these results constitute a significant step forward that could not be extrapolated from existing studies.

Our study is limited in several ways, most of which point to opportunities for further research. First, our study relies on artificial cases. Although these span the ranges of project and portfolio characteristics typical of real projects, it would be important to confirm our findings on actual portfolios. (However, in so doing, it would be important to include a large enough number of cases so as to achieve statistical significance.) Our study also holds the number of projects, activities, and feedback loops constant. Although we found reasonable justifications for these choices relative to the varying factors, real portfolios would force consideration of variations in these factors as well. Future research could also extend our approach to include additional PRs or to explore other situations such as activity preemption (with switching costs), stochastic activity durations, non-renewable resources, dynamic project arrivals, or alternative objective functions. It would also be interesting to explore the effects of additional project and portfolio characteristics such as the project scatter factor and the resource dedication profile (Hendriks et al. 1999), which might inspire new PRs. Moreover, the goal of developing improved PRs specifically for iterative projects remains to be achieved. Towards this end, it seems likely that it would be helpful to develop a new measure of network density that accounts for both feedback arcs and redundant, feed-forward arcs (since these take on a new significance in cyclical networks). Finally, our results could also inform the development of adaptive PRs that adjust to changes in network and/or resource characteristics. Rather than sticking with the same PR throughout a project, switching PRs at appropriate points in a project or portfolio could be advantageous.

References

Allam, S.I.G. (1988) "Multi-project Scheduling: A New Categorization for Heuristic Scheduling Rules in Construction Scheduling Problems," *Construction Management and Economics*, **6**: 93-115.

Anderson, E.G. and N.R. Joglekar (2005) "A Hierarchical Product Development Planning Framework," *Production and Operations Management*, **14**(3): 344-361.

Banerjee, A., J.E. Carrillo and A. Paul (2007) "Projects with Sequential Iteration: Models and Complexity," *IIE Transactions*, **39**(5): 453-463.

Beaujon, G.J., S.P. Marin and G.C. McDonald (2001) "Balancing and Optimizing a Portfolio of R&D Projects," *Naval Research Logistics*, **48**(1): 18-40.

Besner, C. and B. Hobbs (2008) "Project Management Practice, Generic or Contextual: A Reality Check," *Project Management Journal*, **39**(1): 16-33.

Blichfeldt, B.S. and P. Eskerod (2008) "Project Portfolio Management – There's More to It than What Management Enacts," *International Journal of Project Management*, **26**(4): 357-365.

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

Braha, D. and Y. Bar-Yam (2007) "The Statistical Mechanics of Complex Product Development: Empirical and Analytical Results," *Management Science*, **53**(7): 1127-1145.

Browning, T.R. and S.D. Eppinger (2002) "Modeling Impacts of Process Architecture on Cost and Schedule Risk in Product Development," *IEEE Transactions on Engineering Management*, **49**(4): 428-442.

Browning, T.R. and R.V. Ramasesh (2007) "A Survey of Activity Network-Based Process Models for Managing Product Development Projects," *Production & Operations Management*, **16**(2): 217-240.

Browning, T.R. and A.A. Yassine (2010a) "A Random Generator of Resource-Constrained Multi-Project Network Problems," *Journal of Scheduling*, **13**(2): 143-161.

Browning, T.R. and A.A. Yassine (2010b) "Resource-Constrained Multi-Project Scheduling: Priority Rule Performance Revisited," *International Journal of Production Economics*, **126**(2): 212-228.

Chalupnik, M.J., D.C. Wynn, C.M. Eckert and P.J. Clarkson (2008) "Analysing the Relationship between Design Process Composition and Robustness to Task Delays," *International Design Conference*, Dubrovnik, Croatia, May 19-22.

Cho, S.-H. and S.D. Eppinger (2005) "A Simulation-Based Process Model for Managing Complex Design Projects," *IEEE Transactions on Engineering Management*, **52**(3): 316-328.

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 Management Journal*, **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 Science*, **22**(11): 1186-1194.

Cooper, K.G. (1993a) "The Rework Cycle: Benchmarks for the Project Manager," *Project Management Journal*, **24**(1): 17-21.

Cooper, K.G. (1993b) "The Rework Cycle: Why Projects are Mismanaged," *PMNETwork*(Feb.).

Cooper, R.G., S.J. Edgett and E.J. Kleinschmidt (2002) *Portfolio Management for New Products*, 2^{nd} ed., Basic Books.

Davis, E.W. (1975) "Project Network Summary Measures and Constrained Resource Scheduling," *IIE Transactions*, **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 Science*, **21**(8): 944-955.

Dickinson, M.W., A.C. Thornton and S. Graves (2001) "Technology Portfolio Management: Optimizing Interdependent Projects Over Multiple Time Periods," *IEEE Transactions on Engineering Management*, **48**(4): 518-527.

Dilts, D.M. and K.R. Pence (2006) "Impact of Role in the Decision to Fail: An Exploratory Study of Terminated Projects," *Journal of Operations Management*, **24**(4): 378-396.

Engwall, M. and A. Jerbrant (2003) "The Resource Allocation Syndrome: The Prime Challenge of Multi-Project Management?," *International Journal of Project Management*, **21**(6): 403-409.

Eppinger, S.D. and T.R. Browning (2012) *Design Structure Matrix Methods and Applications*, Cambridge, MA, MIT Press.

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

Froehle, C.M. and A.V. Roth (2007) "A Resource-Process Framework of New Service Development," *Production & Operations Management*, **16**(2): 169-188.

Gonçalves, J.F., J.J.d.M. Mendes and M.G.C. Resende (2008) "A Genetic Algorithm for the Resource Constrained Multi-Project Scheduling Problem," *European Journal of Operational Research*, **189**(3): 1171-1190.

Gutjahr, W.J., S. Katzensteiner, P. Reiter, C. Stummer and M. Denk (2010) "Multi-objective Decision Analysis for Competence-oriented Project Portfolio Selection," *European Journal of Operational Research*, **205**: 670-679.

Hans, E.W., W. Herroelen, R. Leus and G. Wullink (2007) "A Hierarchical Approach to Multi-project Planning under Uncertainty," *Omega*, **35**(5): 563-577.

Hartmann, S. and D. Briskorn (2009) "A Survey of Variants and Extensions of the Resource-constrained Project Scheduling Problem," *European Journal of Operational Research*, **207**(1): 1-14.

Hartmann, S. and R. Kolisch (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.

Hendriks, M.H.A., B. Voeten and L. Kroep (1999) "Human Resource Allocation in a Multi-project R&D Environment," *International Journal of Project Management*, **17**(3): 181-188.

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.

Jiao, J.R., Y. Zhang and Y. Wang (2007) "A Heuristic Genetic Algorithm for Product Portfolio Planning," *Computers & Operations Research*, **34**(6): 1777-1799.

Joglekar, N.R. and D.N. Ford (2005) "Product Development Resource Allocation with Foresight," *European Journal of Operational Research*, **160**(1): 72-87.

Kavadias, S.K. and C.H. Loch (2003) "Optimal Project Sequencing with Recourse at a Scarce Resource," *Production and Operations Management*, **12**(4): 433-444.

Kline, S.J. (1985) "Innovation is not a Linear Process," *Research Management*, **28**(2): 36-45.

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. and S. Hartmann (1999) "Heuristic Algorithms for Solving the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis," in *Project Scheduling*, J. Weglarz, Ed., Boston, Kluwer Academic Publishers, pp. 147-178.
Kolisch, R. and S. Hartmann (2005) "Experimental Investigation of Heuristics for Resource-Constrained Project Scheduling: An Update," *European Journal of Operational Research*, **174**(1): 23-37.
Kurtulus, I. and E.W. Davis (1982) "Multi-Project Scheduling: Categorization of Heuristic Rules Performance," *Management Science*, **28**(2): 161-172.
Lee, Z.W., D.N. Ford and N. Joglekar (2008) "Effects of Resource Allocation Policies for Reducing Project Durations: A Systems Modelling Approach," *Systems Research and Behavioral Science*, **24**(6): 551-566.
Lenstra, J.K. and A.H.G.R. Kan (1978) "Complexity of Scheduling under Precedence Constraints," *Operations Research*, **26**(1): 22-35.
Lévárdy, V. and T.R. Browning (2009) "An Adaptive Process Model to Support Product Development Project Management," *IEEE Transactions on Engineering Management*, **56**(4): 600-620.
Liberatore, M.J. and G.J. Titus (1983) "The Practice of Management Science in R&D Project Management," *Management Science*, **29**(8): 962-974.
Linyi, D. and L. Yan (2007) "A Particle Swarm Optimization for Resource-Constrained Multi-Project Scheduling Problem," *International Conference on Computational Intelligence and Security*, Harbin, China, Dec 15-19.
Loch, C.H. and S. Kavadias (2002) "Dynamic Portfolio Selection of NPD Programs using Marginal Returns," *Management Science*, **48**(10): 1227-1241.
Loch, C.H. and C. Terwiesch (2005) "Rush and Be Wrong or Wait and Be Late? A Model of Information in Collaborative Processes," *Production and Operations Management*, **14**(3): 331-343.
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," *8th International Workshop on Project Management and Scheduling*, Valencia, Spain, Apr. 3-5.
Luh, P.B., F. Liu and B. Moser (1999) "Scheduling of Design Projects with Uncertain Number of Iterations," *European Journal of Operational Research*, **113**(3): 575-592.
Meredith, J.R. and S.J. Mantel (2012) *Project Management*, 8th ed., New York, Wiley.
Nassar, K.M. and M.Y. Hegab (2006) "Developing a Complexity Measure for Project Schedules," *Journal of Construction Engineering and Management*, **132**(6): 554-561.
Okada, I., L. Lin and M. Gen (2009) "Solving Resource Constrained Multiple Project Scheduling Problems by Random Key-Based Genetic Algorithm," *Electronics and Communications in Japan*, **92**(8): 25-35.
Osborne, S.M. (1993) *Product Development Cycle Time Characterization Through Modeling of Process Iteration*, Master's thesis, MIT, Cambridge, MA.
Pascoe, T.L. (1966) "Allocation of Resources - CPM," *Revue Française de Recherche Opérationelle*, **38**: 31-38.
Pennypacker, J.S. and L.D. Dye (2002) "Project Portfolio Management and Managing Multiple Projects: Two Sides of the Same Coin?," in *Managing Multiple Projects*, J. S. Pennypacker and L. D. Dye, Eds., New York, NY, Marcel Dekker, pp. 1-10.
Pich, M.T., C.H. Loch and A.D. Meyer (2002) "On Uncertainty, Ambiguity and Complexity in Project Management," *Management Science*, **48**(8): 1008-1023.
PMI (2013) *A Guide to the Project Management Body of Knowledge*, 5th ed., Newtown Square, PA, Project Management Institute.
Pritsker, A.A.B., L.J. Watters and P.M. Wolfe (1969) "Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach," *Management Science*, **16**(1): 93-108.
Rad, P.F. and G. Levin (2006) *Project Portfolio Management*, IIL Publishing.

Shane, S.A. and K.T. Ulrich (2004) "Technological Innovation, Product Development, and Entrepreneurship in *Management Science*," *Management Science*, **50**(2): 133-144.
Simon, H.A. (1996) *The Sciences of the Artificial*, 3rd ed., Cambridge, MA, MIT Press.
Smith, R.P. and S.D. Eppinger (1997a) "Identifying Controlling Features of Engineering Design Iteration," *Management Science*, **43**(3): 276-293.
Smith, R.P. and S.D. Eppinger (1997b) "A Predictive Model of Sequential Iteration in Engineering Design," *Management Science*, **43**(8): 1104-1120.
Sosa, M.E., J. Mihm and T.R. Browning (2013) "Linking Cyclicality and Product Quality," *Manufacturing & Service Operations Management*, **15**(3): 473-491.
Tavares, V.L., J.A. Ferreira and J.S. Coelho (2002) "A Comparative Morphologic Analysis of Benchmark Sets of Project Networks," *International Journal of Project Management*, **20**(6): 475-485.
Terwiesch, C. and K. Ulrich (2008) "Managing the Opportunity Portfolio," *Research Technology Management*, **51**(5): 27-38.
Thomke, S. and T. Fujimoto (2000) "The Effect of 'Front-Loading' Problem-Solving on Product Development Performance," *Journal of Product Innovation Management*, **17**(2): 128-142.
Thomke, S.H. (1998) "Managing Experimentation in the Design of New Products," *Management Science*, **44**(6): 743-762.
Tukel, O.I. (1996) "Scheduling Resource-Constrained Projects When Non-conformaties Exist," *Project Management Journal*, **27**: 47-55.
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.
Vázquez, E.P., M.P. Calvo and P.M. Ordóñez (2013) "Learning Process on Priority Rules to Solve the RCMPSP," *Journal of Intelligent Manufacturing*(forthcoming).
Wysocki, R.K. and R. McGarry (2003) *Effective Project Management*, New York, Wiley.
Yan, H.-S., Z. Wang and M. Jiang (2002) "A Quantitative Approach to the Process Modeling and Planning in Concurrent Engineering," *Concurrent Engineering: Research and Applications*, **10**(2): 97-111.
Zhuang, M. and A.A. Yassine (2004) "Task Scheduling of Parallel Development Projects Using Genetic Algorithms," *ASME International Design Engineering Technical Conferences (Design Automation Conference)*, Salt Lake City, Sep. 28 - Oct. 2.