Robust Optimization Strategies for Total Cost Control in Project Management

Joel Goh *

Nicholas G. Hall **

Melvyn Sim †‡

* Graduate School of Business, Stanford University

** Department of Management Sciences, The Ohio State University

† NUS Business School and Risk Management Institute, National University of Singapore

‡ Corresponding author: dscsimm@nus.edu.sg

March 9, 2010

Abstract

We describe robust optimization procedures for controlling total completion time penalty plus crashing cost in projects with uncertain activity times. These activity times arise from an unspecified distribution within a family of distributions with known support, mean and covariance. We develop linear and linear-based decision rules for making decisions about activity start times and the crashing of activity times. These rules identify decisions that perform well against all possible scenarios of activity time uncertainty. The resulting crashing strategies are implemented in both static and rolling horizon modes. Whereas the traditional planning methodology PERT does not consider correlation between past and future performance within or between activities, our methodology models both types of correlation. We compare our procedures against PERT, and also against the alternative approach of Monte Carlo simulation which assumes more information. Extensive computational results show that our crashing strategies provide over 99% probability of meeting the overall project budget, compared to less than 45% for the previous approaches that use the same information. Expected budget overruns are similarly improved from over 25% to less than 0.1%. The relative advantages of the static and rolling horizon implementations are also discussed. We identify several managerial insights from our results.

Key words and phrases: project management; time/cost tradeoff under uncertainty; robust optimization; linear decision rule.

OR/MS Index 1989 Subject Classification:
Project management: PERT.
Probability: stochastic model: applications.
Project management: resource constraints.

1 Introduction

The use of project management as a planning methodology has expanded greatly in recent years. As evidence of this, the Project Management Institute saw its membership increase from 50,000 in 1996 to over 270,000 in 2008 (Pells 2008). Moreover, most of this increase is due to modern applications, for example information technology, with substantial further growth potential. Further, the use of shorter life cycles for products and services (Value Based Management.net 2009) is leading to increased application of project management in the development of new products and services. Also, the pace of corporate change is constantly increasing as a result of new technologies, more intense competition, and more demanding and less predictable customers (1000ventures.com 2009), and achieving this organizational change in an effective way requires professional project management.

The two most important quantifiable performance measures in project management are completion time relative to a planned schedule, and cost relative to a budget (Might and Fischer 1985). Indeed, a widely used informal definition of project success is "completion of the project on time and on budget". However, many projects fail to achieve this benchmark. For example, only 35% of software development projects are completed on time (Chaos Report 2006). When late project completion is a possibility, one possible solution is the expediting of activity times, or *crashing*. In practice, crashing is usually accomplished through the commitment of additional resources, such as personnel, equipment, and budget, to the individual activities. This results in an important tradeoff between completion time and cost in making crashing decisions (Klastorin 2004, Kerzner 2009). If activity times are deterministic or highly predictable, then crashing decisions can be made efficiently using optimization models.

However, in most projects, random variations in activity times threaten delays to the overall project schedule. Herroelen and Leus (2005) provide a comprehensive survey of various types of approaches to scheduling under uncertainty. In practice, failure to compensate adequately for activity time variations is a prominent cause of project failure (Hughes 1986). The traditional, and most widely used, approach for planning projects in the presence of uncertainty in activity times is PERT (U.S. Navy 1958). The PERT model requires decision makers to estimate, for each activity, optimistic, pessimistic and most likely durations. Based on an approximation to the beta distribution, these estimates are used in formulas for the expectation and standard deviation of each activity time in the project. In order to build these random activity times into a simple probabilistic model of overall project completion time, PERT requires three important assumptions regarding the project's activities and its precedence network, as follows.

A1: The uncertain activity times are probabilistically independent.

A2: A path with the longest expected length is also a critical path after realization of the activity times.

A3: Based on the Central Limit Theorem, the total length of the critical path approximately follows the normal distribution.

Each of these assumptions may be difficult to justify both in theory and in practice. First, if two activities use the same resources, then their performance is likely to be positively correlated;hence, Assumption A1 is unlikely to hold. Second, in situations where the project network contains many paths of similar expected length, it is frequently the case that a path which is not critical based on expected activity times becomes critical once activity times are realized; hence, Assumption A2 does not hold. Finally, if there are fewer than about 30 activities in series on the critical path, then the approximation of the mean under the Central Limit Theorem in Assumption A3 is likely to be poor (Black 2008). Depending on the precedence structure, there may need to be several hundred activities in the project in order to meet this requirement. Moreover, achieving an accurate approximation to the tail probabilities of the distribution of the critical path length requires still many more activities (Barbour and Jensen 1989).

An alternative to PERT that does not require Assumptions A1–A3 is Monte Carlo simulation. Assuming knowledge of an underlying probability distribution for each activity time, these times can be generated randomly and used to estimate the critical path length. However, whereas PERT is widely used in practice, Monte Carlo simulation is not (Schonberger 1981, Kwak and Ingall 2007). This is apparently because of a lack of management resources allocated to simulation analysis, and also because of inadequate computing capability. Moreover, in the context of project management, both PERT and Monte Carlo simulation are generally used as descriptive rather than prescriptive methods (Bowman 1994). Although stochastic programming has been used (Bowman 1994, Gutjahr et al. 2000) in conjunction with either technique to make prescriptive crashing decisions, the difficulty of estimating the required probability distributions (Williams 2003) limits its practical usefulness.

In most projects, information about the progress of all ongoing activities within a project is updated periodically, for example weekly. A standard methodology for reporting project progress is earned value analysis (U.S. Department of Defense 1962). Earned value analysis compares both the time used and the cost spent so far against the amount of progress that has been achieved in the project (Fleming and Koppelman 2005). The resulting differences are known as time variance and cost variance, respectively. Importantly, these variances may correlate with similar variances during the remainder of the same activity. This is especially likely if the project team, and where applicable the subcontractors, being used are the same as during the earlier part of the activity. Klastorin (2004) describes various methods for the estimation of activity time and cost at completion, based on partial activity progress. In addition, as discussed above, there may be correlation with performance on subsequent activities.

Robust optimization is introduced by Soyster (1973), and popularized by Ben-Tal and Nemirovski (1998), El-Ghaoui et al. (1998), and Bertsimas and Sim (2004). There is apparently only one previous application of robust optimization to the time-cost tradeoff problem in project management. Cohen et al. (2007) consider the problem of minimizing total crash cost plus linear project completion time cost. They assume prespecified interval or ellipsoidal uncertainty sets, and develop linear decision rules for crashing decisions. The size of the uncertainty sets is adjustable and depends on the level of ambiguity averseness of the decision maker. In the objective of their model, they minimize the worst-case total cost over their uncertainty sets. One concern about their proposed model is that it is difficult to calibrate the uncertainty level for the uncertainty sets. Finally, in the objective of their model, they minimize the worst-case total cost over their uncertainty sets, which can provide very conservative results. In our opinion, these concerns limit the practical usefulness of the proposed model.

In this paper, we model the objective of project completion on time and on budget by considering total project cost, consisting of a project completion time penalty plus total crash cost. Our model uses linear and linear-based decision rules (Ben Tal et al. 2004, Goh and Sim 2010) to model these decisions. Since decision rules are functions of the uncertainties in the problem, they are capable of adjusting decisions, based on the realization of uncertainties during the project. Hence, the solutions from our models use information that is received as the project progresses, to obtain activity start time and crashing decisions that perform well against all possible scenarios of activity time uncertainty. Whereas PERT does not recognize correlation between previous and future performance on a single activity, or between performance on previous and subsequent activities, our methodology allows for their modeling, up to the limits of available information. Our crashing strategies are implemented in both static and rolling horizon modes. We show computationally that our crashing strategies provide a much higher probability of meeting the overall project budget than PERT-based crashing strategies. The relative advantages of static and rolling horizon implementations are also discussed. We also identify several managerial insights from our results.

This paper is organized as follows. In Section 2, we provide our notation and a formal description of the problem to be studied. In Section 3, we discuss various formulations of the objective function in the problem. Section 4 describes our development of a robust optimization model for project planning with crashing. Section 5 describes a computational study that compares the performance of our crashing strategies against those of PERT and Monte Carlo simulation. Finally, Section 6 contains a conclusion and some suggestions for future research.

2 Preliminaries

In this section, we provide our definitions and notation, as well as a formal description of the problem studied.

2.1 General notation

We denote a random variable by the tilde sign, for example $\tilde{x}$. Bold lower case letters such as $\boldsymbol{x}$ represent vectors, and bold upper case letters such as $\boldsymbol{A}$ represent matrices. Given a matrix $\boldsymbol{A}$, its transposed ith row is denoted by $\boldsymbol{a}_i$ and its (i, j)th element is denoted by A_{ij}. We denote by $\boldsymbol{e}$ the vector of all ones and by $\boldsymbol{e}^i$ the ith standard basis vector. In addition, $x^+ = \max\{x, 0\}$ and $x^- = \max\{-x, 0\}$. The same notation can be applied to vectors, for example $\boldsymbol{y}^+$ and $\boldsymbol{z}^-$, which denotes that the corresponding operations are performed component-wise. For any set S, we denote by $\mathbb{1}_S$ the indicator function on that set. Also, we denote by $[N]$ the set of positive running indices to N, i.e. $[N] = \{1, \ldots, N\}$, for some positive integer N. For completeness, we assume that $[0] = \emptyset$.

2.2 Problem description

We define a project to be a finite collection of N individual *activities*, with precedence relationships between the activities defined as part of the project specification. The project completes when every activity has completed. The project manager faces the problem of completing the project on budget, where the total project cost includes both completion time penalty cost and total crash cost. In order to achieve this, two sets of decisions are needed. First, the starting time of each activity in the project must be determined, taking into account the restrictions imposed by the precedence relationships. Second, decisions must be made about by how much to crash each activity. The project completion target, τ, and the crashing budget, B, are taken as fixed exogenous parameters. In practice, these parameters are typically set at the project planning stage by senior management, after some input or negotiation from the project manager.

In many projects, activity times are uncertain (Klastorin 2004, Kerzner 2009). We model uncertainty using a probability space $(\Re^N, \mathcal{B}(\Re^N), \mathbb{P})$. We denote by $\tilde{\boldsymbol{z}}$ the elements of the sample space $\Re^N$, which represent the uncertain activity times. The Borel σ-algebra $\mathcal{B}(\Re^N)$ is the set of events that are assigned probabilities by the probability measure $\mathbb{P}$. We denote by $\mathrm{E}_{\mathbb{P}}(\cdot)$ the expectation with respect to $\mathbb{P}$.

The precedence relationships between the activities of a project can be represented by a directed acyclic graph, or project network. Two widely used conventions for this network are the Activity-on-Node (AON) and Activity-on-Arc (AOA) formats (Klastorin 2004). In this paper, we adopt the AOA convention. The project has N activities, and therefore the graph has N arcs. We let M represent the number of nodes in the graph, and define the set of node pairs linked by arcs as

$$\mathcal{E} \triangleq \{(i,j) : \exists \text{ an arc from node } i \text{ to node } j\}.$$

The bijective mapping function $\Phi : \mathcal{E} \to [N]$ is the unique map from a node pair $(i,j) \in \mathcal{E}$ to an activity $k \in [N]$.

2.3 Information flow

The uncertain activity times are revealed over time by two mechanisms. First, whenever an activity is completed, its previously uncertain activity time becomes known. We use this information to form non-anticipative linear decision rules for crashing and start time decisions. Since these decision rules are defined as affine functions of the revealed uncertainties, the actual decisions implicitly adjust themselves to the already revealed activity times. We discuss the development of these decision rules in Section 4.2.

Second, the project manager receives periodic updates by activity managers on the progress of each activity. We denote the length of each time period between updates by T. At the end of each time period, time variance and cost variance are used to monitor the progress of each activity and the overall status of the project with respect to the schedule and budget. Each periodic update provides the manager with updated resource information (budget, schedule, and allowable crashing) and an updated uncertainty description. Using the updated information, the project manager may choose to reallocate crashing resources between activities. We use this periodic information gain, as well as the activity completion time information described above, to develop an iterative solution to the updated model, which we discuss in Section 4.6.

3 Objective Function

As discussed in the Introduction, the project manager faces the two, usually conflicting, objectives of completing the project on schedule and within budget. We consider the objective of minimizing the total project cost, which includes completion time penalty cost plus total crashing cost. Since the individual activity times are subject to uncertainty, the project completion time, $\tilde{T}$, is also uncertain. In many cases, when projects run past their deadline, a contract penalty is incurred. This penalty can be modeled as a continuous function of the project completion time in excess of the deadline, i.e. $p\left(\tilde{T}-\tau\right)$. The penalty function has the following properties:

1. $t \leq 0 \Rightarrow p(t) = 0$.

2. $p(\cdot)$ is nondecreasing and convex.

The first property implies that if the project is completed on schedule, no additional cost is incurred. The second property increasingly penalizes later completion. Both properties are representative of typical business practice (Kerzner 2009).

The activity crashing cost, denoted by $\tilde{C}$, forms the other cost component. Although the crashing cost is a deterministic function of the amount of crashing, decisions about crashing are driven by uncertainties in the activity times. Hence, $\tilde{C}$ is ultimately an uncertain parameter as well. A possible objective is to maximize the probability that the combined cost, $\tilde{V} = \tilde{C} + p\left(\tilde{T} - \tau\right)$ is within the project budget, B. However, maximizing success probability fails to consider the extent of overrun relative to a budget that is missed.

Example 1. Consider two project completion scenarios with an overall project budget of \$10m. Scenario A completes the project at a combined cost of \$9m with probability 79%, and \$11m with probability 21%. Scenario B completes the project at a combined cost of \$9m with probability 80%, and \$1,000m with probability 20%. Under the objective of probability maximization, scenario B is preferred. However, few managers would prefer this alternative, because it ignores tail risk. A similar argument is used in the finance literature to justify the use of Conditional Value-at-Risk instead of Value-at-Risk (Alexander and Baptista 2004).

Brown and Sim (2009) introduce *quasiconcave satisficing measures* (QSMs), a class of functions for evaluating an uncertain prospect against an aspiration level known as a satisficing measure. The probability of success is a member of this class. QSMs favor diversification behavior by the decision maker. We adopt the *CVaR satisficing measure*, which is a QSM defined as follows:

Definition 1 *Given a random variable $\tilde{V}$ representing uncertain expenditure, and a budget B, the* CVaR satisficing measure *is defined by*

$$\rho_B(\tilde{V}) = \begin{cases} \sup\{\beta \in (0,1) : \mu_\beta(\tilde{V}) \le B\} & \textit{if feasible} \\ 0 & \textit{otherwise} \end{cases}$$

where μ_β is the β-Conditional Value-at-Risk (Rockafellar and Uryasev 2000) given by

$$\mu_\beta(\tilde{V}) = \inf_{v \in \Re} \left\{ v + \frac{1}{1-\beta} \mathrm{E}_{\mathbb{P}} \left(\tilde{V} - v\right)^+ \right\}.$$

To understand the satisficing properties of $\rho_B(\tilde{V})$, observe that if $\tilde{V}$ always falls below the budget, B, then $\mu_\beta(\tilde{V}) \le B$ for all $\beta \in (0,1)$ and hence, $\rho_B(\tilde{V}) = 1$. On the other hand, if $\tilde{V}$ never achieves the target, then $\mu_\beta(\tilde{V}) > B$ and hence, $\rho_B(\tilde{V}) = 0$. The CVaR satisficing measure is introduced by Chen and Sim (2009) using the terminology *shortfall aspiration level criterion.* Brown and Sim (2009) show that, among all QSMs, $\rho_B(\tilde{V})$ has the largest lower bound for the success probability, $\mathbb{P}(\tilde{V} \le B)$. Moreover, Brown et al. (2009) show that the measure is able to resolve classical behavioral paradoxes that arise in expected utility theory. Furthermore, the solution obtained by optimizing the satisficing level often provides a high probability of project success. For example, if uncertainties are normally distributed, optimal solutions found under the CVaR satisficing measures are also optimal under the probability measure. We also substantiate this claim with an extensive computational study.

Another justification for using the CVaR satisficing measure in the objective is that the alternative of a probability measure does not consider tail risk. Chen and Sim (2009) show that if $\mathrm{E}_{\mathbb{P}}\left(\tilde{V}\right) < B$, then

$$\rho_B(\tilde{V}) = \sup_{\alpha>0} \mathrm{E}_{\mathbb{P}}\left(\min\{1, \alpha(B - \tilde{V})\}\right) > 0. \tag{1}$$

This representation reflects, first an indifference toward costs that fall within the budget by a margin of at least $\frac{1}{\alpha}$, and second an increasing aversion toward costs that exceed the budget by larger amounts.

4 Robust Optimization Model

In this section, we apply certain assumptions to the structure of the activity time uncertainty in the problem, in order to formulate a model that is computationally tractable. In keeping with practical situations, the project manager rarely has full knowledge of the probability distribution, $\mathbb{P}$. Hence, we assume only certain distributional properties of the uncertain activity time such as its mean, covariance matrix, and support, which characterize a family of distributions $\mathbb{F}$ containing $\mathbb{P}$. We describe $\mathbb{F}$ in Section 4.1 below. Because decision makers are typically ambiguity averse (Ellsberg 1961), we evaluate the worst-case objective over the family of distributions $\mathbb{F}$, that is

$$Z_0 = \sup_{\alpha>0} \inf_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}}\left(\min\left\{1, \alpha B - \alpha\tilde{C} - \alpha p\left(\tilde{T} - \tau\right)\right\}\right), \tag{2}$$

using a similar representation to (1) where $\sup_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}}\left(C + p\left(\tilde{T} - \tau\right)\right) < B$. Somewhat surprisingly, by imposing this additional layer of modeling complexity, and using modest assumptions on the structure of $p(\cdot)$ and $\mathbb{F}$, we can compute a tractable bound on Z_0.

4.1 Model of uncertainty

We assume the family of distributions, $\mathbb{F}$, is described by the following distributional properties.

Support. We denote by $\mathcal{W} \subseteq \Re^N$ the smallest bounded convex set containing the support of $\tilde{\boldsymbol{z}}$. For example, if the actual support of $\tilde{\boldsymbol{z}}$ is nonconvex, we take $\mathcal{W}$ as its convex hull.

We further assume that $\mathcal{W}$ is a full dimensional tractable conic representable set, that is, a set that can be represented, exactly or approximately, by a polynomial number of linear and/or second order conic constraints.

Mean. We denote by $\bar{\boldsymbol{z}}$ the mean of $\tilde{\boldsymbol{z}}$. Instead of modeling the mean as a precisely known quantity, we consider a generalization in which the mean $\bar{\boldsymbol{z}}$ is itself uncertain, with corresponding, possibly unbounded, support contained in a set $\hat{\mathcal{W}}$. We again assume that $\hat{\mathcal{W}}$ is a tractable conic representable set. This includes the case of a known mean, which corresponds to $\hat{\mathcal{W}}$ being a singleton set.

Covariance. We denote by $\boldsymbol{\Sigma}$ the known covariance of $\tilde{\boldsymbol{z}}$.

4.2 Decision rules

Ideally, we would like the crashing and activity start time decisions to be decision rules (DRs), i.e. measurable functions of the underlying uncertainties. As the project progresses, we impose *non-anticipativity* requirements on the DRs, thus requiring that they have functional dependencies only on previously revealed uncertainties. We model the non-anticipativity requirements using an information index set $I \subseteq [N]$. Each DR has functional dependence only on the components of the uncertainty vector with indices within the DR's corresponding information index set. Specifically, we define the parametrized set of functions as

$$\mathcal{Y}(m, N, I) \triangleq \left\{ \boldsymbol{f} : \Re^N \to \Re^m \ : \ \boldsymbol{f}\left(\boldsymbol{z} + \sum_{i \notin I} \lambda_i \boldsymbol{e}^i\right) = \boldsymbol{f}(\boldsymbol{z}), \forall \boldsymbol{\lambda} \in \Re^N \right\}.$$

The first two parameters specify the dimensions of the codomain and domain of the functions, respectively, while the final parameter enforces the non-anticipativity requirement.

For example, if we have $\boldsymbol{w} \in \mathcal{Y}(3, 4, \{2, 4\})$, then the DR $\boldsymbol{w}(\cdot)$ is a function characterized by $\boldsymbol{w}(\tilde{\boldsymbol{z}}) = \boldsymbol{w}(\tilde{z}_2, \tilde{z}_4), \forall \tilde{\boldsymbol{z}} \in \Re^4$. Since 1 and 3 are not included in the information index set of this DR, we have no guarantee that the values of $\tilde{z}_1$ or $\tilde{z}_3$ will be revealed before we make the decision $\boldsymbol{w}(\cdot)$, and correspondingly, we cannot use the values of $\tilde{z}_1$ or $\tilde{z}_3$ in this DR.

Hence, denoting the activity start times as $\boldsymbol{x}(\tilde{\boldsymbol{z}}) \in \Re^M$ and the activity crashing decisions as $\boldsymbol{y}(\tilde{\boldsymbol{z}}) \in \Re^N$, we can write

$$\begin{aligned} x_i &\in \mathcal{Y}(1, N, I_i^x), \quad \forall i \in [M], \\ y_k &\in \mathcal{Y}(1, N, I_k^y), \quad \forall k \in [N], \end{aligned}$$

where the structure of $\{I_i^x\}_{i=1}^M$ and $\{I_k^y\}_{k=1}^N$ depends on the precedence constraints in the project network.

Information index sets and their associated DRs provide sufficient flexibility to model the robust crashing problem, as in Section 4.3. However, the use of general DRs typically makes the optimization model computationally intractable (Ben-Tal et al. 2004). Therefore, instead of considering DRs from the space of all functions $\mathcal{Y}$, we restrict these decisions to be affine functions of the uncertainties. These linear decision rules (LDRs) are approximations that ensure computational tractability. They also allow simplicity of implementation and can easily be extended to more complex bi-deflected LDRs (Goh and Sim 2010) for closer approximation.

Specifically, we define the parametrized set of affine functions as

$$\mathcal{L}(m, N, I) = \left\{ \boldsymbol{f} : \Re^N \to \Re^m : \exists (\boldsymbol{y}^0, \boldsymbol{Y}) \in \Re^m \times \Re^{m \times N} : \begin{array}{c} \boldsymbol{f}(\boldsymbol{z}) = \boldsymbol{y}^0 + \boldsymbol{Y}\boldsymbol{z} \\ \boldsymbol{Y}\boldsymbol{e}^i = \boldsymbol{0}, \forall i \notin I \end{array} \right\}. \tag{3}$$

For LDRs, the non-anticipativity requirements take the explicit form of a set of linear equality constraints. For example, if $\boldsymbol{w} \in \mathcal{L}(3, 4, \{2, 4\})$, then the LDR $\boldsymbol{w}(\tilde{\boldsymbol{z}})$ can be represented as the affine function $\boldsymbol{w}(\tilde{\boldsymbol{z}}) = \boldsymbol{w}^0 + \boldsymbol{W}\tilde{\boldsymbol{z}}$, with the additional constraints that the first and third columns of the 3×4 matrix $\boldsymbol{W}$ must be zero. An equivalent representation is $\boldsymbol{w}(\tilde{\boldsymbol{z}}) = \boldsymbol{w}^0 + \tilde{z}_2 \boldsymbol{w}^2 + \tilde{z}_4 \boldsymbol{w}^4$. If $\boldsymbol{w}(\tilde{\boldsymbol{z}})$ is used in an optimization procedure, then we solve for optimized values of $(\boldsymbol{w}^0, \boldsymbol{w}^2, \boldsymbol{w}^4)$. Observe that the actual numerical value of $\boldsymbol{w}(\tilde{\boldsymbol{z}})$ is known only after the uncertainties $(\tilde{z}_2, \tilde{z}_4)$ have been revealed.

4.3 Project crashing model

We assume without loss of generality that the project commences at node 1 and terminates at the final node M. Hence, the project completion time is expressed as $\tilde{T} = x_M(\tilde{\boldsymbol{z}})$.

The crashing cost for the k^{th} activity can be expressed as a function $c_k : \Re_+ \to \Re_+$ of the amount of crashing for that activity. This function is nondecreasing and weakly convex on its domain, and has $c_k(0) = 0$. In practice, a typical cost function is increasing, piecewise-linear and convex (Kelley and Walker 1959), where each piece of the function represents a different type of resource used to crash the activity, and each resource type has its own linear cost rate. The total crashing cost is then the sum of the individual crashing costs, $\tilde{C} = \sum_{k=1}^N c_k\left(y_k(\tilde{\boldsymbol{z}})\right)$.

Substituting these quantities into the robust objective function (2), we obtain the following static robust optimization model for project crashing:

$$
\begin{array}{rll}
Z_0^* = & \max\limits_{\boldsymbol{x}(\cdot),\boldsymbol{y}(\cdot),\alpha} & \inf\limits_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}}\left(\min\left\{1, \alpha B - \alpha\sum_{k=1}^{N} c_k\left(y_k(\tilde{\boldsymbol{z}})\right) - \alpha p\left(x_M(\tilde{\boldsymbol{z}}) - \tau\right)\right\}\right) \\
& \text{s.t.} & \\
& & x_j(\tilde{\boldsymbol{z}}) - x_i(\tilde{\boldsymbol{z}}) \geq (\tilde{z}_k - y_k(\tilde{\boldsymbol{z}}))^+ \quad \forall (i,j)\in\mathcal{E}, k = \Phi(i,j) \\
& & \mathbf{0} \leq \boldsymbol{y}(\tilde{\boldsymbol{z}}) \leq \boldsymbol{u} \\
& & \boldsymbol{x}(\tilde{\boldsymbol{z}}) \geq \mathbf{0} \\
& & x_i \in \mathcal{Y}(1, N, I_i^x), \quad \forall i \in [M] \\
& & y_k \in \mathcal{Y}(1, N, I_k^y), \quad \forall k \in [N] \\
& & \alpha \geq 0.
\end{array} \tag{4}
$$

In model (4), $\boldsymbol{u}$ is a vector that represents the practical limit on how much each activity can be crashed. This parameter is part of the project specification. The solution $(\boldsymbol{x}(\cdot), \boldsymbol{y}(\cdot))$ to problem (4) is a set of decision rules that prescribes the start time of each node and the amount of crashing for each activity. Finally, the $(\tilde{z}_k - y_k(\tilde{\boldsymbol{z}}))^+$ term in the model ensures that the crashed activity time remains nonnegative.

4.4 Example project

In order to compare our crashing procedure with several PERT-based and simulation-based heuristics, we consider a project with the network shown in Figure 1. Though small, this network suffices to illustrate several of the ideas in the discussion that follows. An extensive computational study using randomly generated networks appears in Section 5.

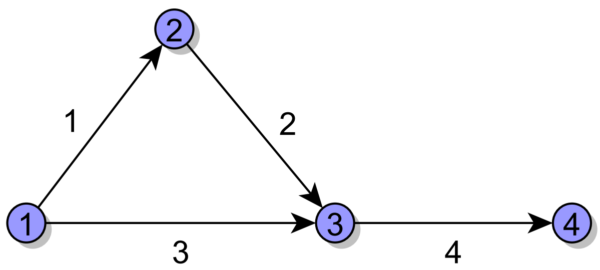

Figure 1: Example Project Network.

In the example in Figure 1, there are four activities and four nodes, i.e. $N = 4$ and $M = 4$. There are two paths from node 1 to node 4. Path A comprises activities 1-2-4, and path B comprises activities 3-4. The cost and schedule parameters are given as follows. The scheduled completion time $\tau = 0.9$ weeks, and the penalty cost function $p(\cdot)$ is a linear cost rate of $p = \$4$ per week in excess of τ. The cost of crashing each activity is described by a linear cost rate of \$5 per week for activities 1 and 3, and \$1 per week for the other activities, i.e. $\boldsymbol{c} = [5, 1, 5, 1]'$. Each activity can be crashed by at most one week. The project budget is $B = \$2$. Each activity has random completion time between 0 and 1, i.e. the support of the uncertainties is the unit hypercube. The mean of each activity is 0.5 weeks, $\boldsymbol{\mu} = 0.5\boldsymbol{e}$, and the covariance matrix is

$$\boldsymbol{\Sigma} = 0.09 \begin{bmatrix} 1 & 9/17 & 0 & 0 \\ 9/17 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

PERT is essentially a descriptive methodology for estimating project completion time. We now describe several heuristic crashing strategies for this project, based on PERT. The simplest strategy is not to crash any activities, and reserve all our budget to pay the penalty cost. This is denoted as Heuristic NOCRASH, with crashing decision vector $\boldsymbol{y} = [0, 0, 0, 0]'$. We use this solution as a baseline strategy for comparison.

An alternative strategy is to use the standard model for PERT, which analyzes the network based on expected activity times. Under this model, only path A is critical, and in order for the project to have an expected completion time within the scheduled deadline, we have to crash activity 4 by at least 0.6 weeks. Following the procedure of Siemens (1971), activity 4 is chosen over activity 2, even though both have the same crash cost rate, since activity 4 lies on both paths. This strategy is denoted as MINCRASH and has crashing decision vector $\boldsymbol{y} = [0, 0, 0, 0.6]'$. This leaves \$1.4 from the budget to pay any penalty costs associated with late project completion.

Finally, we compare against a heuristic based on Monte Carlo simulation, which we denote by MCSIM. Monte Carlo simulation has several advantages over PERT. It uses a richer description of the uncertainties, and does not require the asymptotic approximation of normality that PERT uses. Moreover, Monte-Carlo simulation directly models activity time correlation and overtaking of the critical path, two issues that are ignored in PERT.

Monte Carlo simulation is essentially a descriptive technique for project analysis, rather than a prescriptive one. However, two streams of research apply Monte Carlo simulation results prescriptively. Bowman (1994) develops a procedure for using simulation results prescriptively in a time-cost tradeoff problem with the objective of minimizing the expected project completion time. Two important assumptions of the procedure are the existence of known probability density functions, and independence of the uncertain activity times. Simulation based estimates for derivatives of the expected completion time with respect to activity time distribution parameters are found and incorporated into a greedy crashing heuristic. Gutjahr et al. (2000) develop an enumerative stochastic optimization approach to make binary crashing allocation decisions that minimize the total of crash cost and expected penalty for late project completion. They make the same two assumptions. By contrast, our work explicitly models correlation between the activity times and optimizes over the worst case probability distribution.

In order to obtain prescriptive information at low computational cost, we now develop a benchmark heuristic for obtaining crash times from the simulated sample path data. The objective in our heuristic is the same as that used by Gutjahr et al. (2000). Assuming that we have already simulated L random vectors of activity times from some distribution, which we collect into a matrix $\boldsymbol{Z} \in \Re^{N \times L}$, we solve the following two-stage stochastic optimization problem using sampling average approximation.

$$\begin{array}{lll} \min\limits_{\boldsymbol{X},\boldsymbol{y}} & \displaystyle\sum_{k=1}^{N} c_k y_k + \frac{1}{L}\sum_{l=1}^{L} p\left(X_{Ml} - \tau\right) & \\ \text{s.t.} & & \\ & X_{jl} - X_{il} \geq \left(Z_{kl} - y_k\right)^+, \quad \forall (i,j) \in \mathcal{E}, k = \Phi(i,j), l \in [L] & \\ & \boldsymbol{0} \leq \boldsymbol{y} \leq \boldsymbol{u} & \\ & \boldsymbol{X} \geq \boldsymbol{0}, & \end{array} \tag{5}$$

and use optimal vector $\boldsymbol{y}^*$ as the heuristic crash vector. We note that this is a simplification of the desired crashing decisions, which should be functions of the activity times as in model (4). However, providing such flexibility in crashing decisions would make the stochastic optimization problem highly intractable. Each column of the matrix $\boldsymbol{X} \in \Re^{M \times L}$ represents the node start times in the corresponding sample of activity times.

We summarize these various crashing strategies in Table 1.

Strategy	Crashing Decision Vector, $\boldsymbol{y}$
NOCRASH	$[0, 0, 0, 0]'$
MINCRASH	$[0, 0, 0, 0.6]'$
MCSIM	From solving model (5)

Table 1: PERT-Based Heuristic Crashing Strategies.

4.5 Approximate solution of project crashing model

Due to the intractability of the optimal crashing model (4), we need to construct tractable approximations. We begin by assuming that the project penalty cost and activity crashing costs are piecewise-linear convex functions, represented respectively as

$$p(x) = \max_{m \in [K]} \{p_m x + q_m\} \quad \text{and} \quad c_k(y) = \max_{m \in [L]} \{c_{km} y + d_{km}\}, \forall k \in [N],$$

where $[K]$ and $[L]$ denote the index sets of the linear cost function pieces in each case. Such piecewise-linear costs arise naturally in practice (Kelley and Walker 1959) and they also approximate general convex functions well.

4.5.1 Linear decision rule approximations

We begin with a linear approximation of problem (4), where we restrict all decision rules to be LDRs. This LDR model involves solving the following robust optimization problem. By introducing auxiliary decision rules $\boldsymbol{r}$, $\boldsymbol{s}$, and $\boldsymbol{t}$, model (4) can be equivalently expressed as

$$\begin{array}{rlr}
1 - Z_{LDR} = & & \\
\min & \sup_{\hat{\boldsymbol{z}} \in \hat{\mathcal{W}}} \left\{ r^0 + \boldsymbol{r}'\hat{\boldsymbol{z}} \right\} & \\
\text{variables} & \alpha, t^0, \boldsymbol{t}, r^0, \boldsymbol{r}, \{x_i^0, \boldsymbol{x}_i\}_{i=1}^M, \{y_k^0, \boldsymbol{y}_k, s_k^0, \boldsymbol{s}_k\}_{k=1}^N & \\
\text{s.t.} & 1 - \alpha B + \sum_{k=1}^N \left(s_k^0 + t^0 - r^0 + (\boldsymbol{s}_k + \boldsymbol{t} - \boldsymbol{r})' \boldsymbol{z} \right) \le 0, & \forall \boldsymbol{z} \in \mathcal{W} \\
& c_{km} y_k^0 + \alpha d_{km} - s_k^0 + (c_{km}\boldsymbol{y}_k - \boldsymbol{s}_k)' \boldsymbol{z} \le 0, & \forall \boldsymbol{z} \in \mathcal{W}, \forall m \in [L], \forall k \in [N] \\
& p_m x_M^0 + \alpha q_m - \alpha p_m \tau - t^0 + (p_m \boldsymbol{x}_M - \boldsymbol{t})' \boldsymbol{z} \le 0, & \forall \boldsymbol{z} \in \mathcal{W}, \forall m \in [K] \\
& x_j^0 - x_i^0 + \boldsymbol{x}_j'\boldsymbol{z} - \boldsymbol{x}_i'\boldsymbol{z} \ge \alpha z_k - y_k^0 - \boldsymbol{y}_k'\boldsymbol{z}, & \forall \boldsymbol{z} \in \mathcal{W}, \forall (i,j) \in \mathcal{E}, k = \Phi(i,j) \\
& x_j^0 - x_i^0 + \boldsymbol{x}_j'\boldsymbol{z} - \boldsymbol{x}_i'\boldsymbol{z} \ge 0, & \forall \boldsymbol{z} \in \mathcal{W}, \forall (i,j) \in \mathcal{E} \\
& 0 \le y_k^0 + \boldsymbol{y}_k'\boldsymbol{z} \le \alpha u_k, & \forall \boldsymbol{z} \in \mathcal{W} \\
& x^0 + \boldsymbol{x}_i'\boldsymbol{z} \ge 0, & \forall \boldsymbol{z} \in \mathcal{W}, \forall i \in [M] \\
& r^0 + \boldsymbol{r}'\boldsymbol{z} \ge 0, & \forall \boldsymbol{z} \in \mathcal{W} \\
& \alpha \ge 0 & \\
& x_{ij} = 0, & \forall i \in [M], \forall j \notin I_i^x \\
& y_{kj} = 0, & \forall k \in [N], \forall j \notin I_i^y.
\end{array} \tag{6}$$

This model is a standard robust linear optimization problem, which can be solved using robust counterpart techniques from the literature of robust optimization (Ben-Tal and Nemirovski 1998, Bertsimas and Sim 2004). The tractability of model (6) follows from the structural assumptions on $\mathcal{W}$ and $\hat{\mathcal{W}}$. In particular, if $\mathcal{W}$ and $\hat{\mathcal{W}}$ are polyhedral, model (6) reduces to a simple linear program. We summarize the relationship between this linear approximation and the original problem in the two theorems that follow.

Theorem 1 *If model (6) has a feasible solution where* $\alpha > 0$*, then*

$$\begin{array}{rcl} x_i^*(\tilde{\boldsymbol{z}}) & = & \frac{1}{\alpha}\left(x_i^0 + \boldsymbol{x}_i'\tilde{\boldsymbol{z}}\right), \quad \forall i \in [M] \\ y_k^*(\tilde{\boldsymbol{z}}) & = & \frac{1}{\alpha}\left(y_k^0 + \boldsymbol{y}_k'\tilde{\boldsymbol{z}}\right), \quad \forall k \in [N] \end{array}$$

is a feasible solution to (4).

Proof. By assumption, $\boldsymbol{x}^*(\tilde{\boldsymbol{z}})$ and $\boldsymbol{y}^*(\tilde{\boldsymbol{z}})$ are affine functions of $\tilde{\boldsymbol{z}}$. From the last two constraints of model (6), and recalling the definition (3) of $\mathcal{L}(m, N, I)$, we have

$$\begin{array}{l} x_i^* \in \mathcal{L}(1, N, I_x^i) \subset \mathcal{Y}(1, N, I_x^i), \quad \forall i \in [M]\,, \\ y_k^* \in \mathcal{L}(1, N, I_y^k) \subset \mathcal{Y}(1, N, I_y^k), \quad \forall k \in [N]\,. \end{array}$$

The first through third constraints in model (4) are implied by the fourth through seventh constraints in model (6). The other constraints in model (4) are implied by their corresponding constraints in model (6). ■

Theorem 2 *The following inequality holds:* $Z_0^* \geq Z_{LDR}$.

Proof.

$$\begin{array}{rlll} Z_0^* = & 1 - \min & \sup_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}}\left(r(\tilde{\boldsymbol{z}})\right) & \\ & \text{s.t.} & 1 - \alpha B + \sum_{k=1}^N s_k(\tilde{\boldsymbol{z}}) + t(\tilde{\boldsymbol{z}}) \leq r(\tilde{\boldsymbol{z}}) & \\ & & c_{km}\alpha y_k(\tilde{\boldsymbol{z}}) + \alpha d_{km} \leq s_k(\tilde{\boldsymbol{z}}), & \forall m \in [L]\,, \forall k \in [N] \\ & & p_m \alpha x_M(\tilde{\boldsymbol{z}}) + \alpha q_m - \alpha p_m \tau \leq t(\tilde{\boldsymbol{z}}), & \forall m \in [K] \\ & & x_j(\tilde{\boldsymbol{z}}) - x_i(\tilde{\boldsymbol{z}}) \geq \tilde{z}_k - y_k(\tilde{\boldsymbol{z}}), & \forall (i,j) \in \mathcal{E}, k = \Phi(i,j) \\ & & x_j(\tilde{\boldsymbol{z}}) - x_i(\tilde{\boldsymbol{z}}) \geq 0, & \forall (i,j) \in \mathcal{E} \\ & & \boldsymbol{0} \leq \boldsymbol{y}(\tilde{\boldsymbol{z}}) \leq \boldsymbol{u} & \\ & & \boldsymbol{x}(\tilde{\boldsymbol{z}}) \geq \boldsymbol{0} & \\ & & r(\tilde{\boldsymbol{z}}) \geq 0 & \\ & & \alpha \geq 0 & \\ & & x_i \in \mathcal{Y}(1, N, I_i^x), & \forall i \in [M] \\ & & y_k \in \mathcal{Y}(1, N, I_k^y), & \forall k \in [N] \\ & & \boldsymbol{s} \in \mathcal{Y}(N, N, [N]) & \\ & & r, t \in \mathcal{Y}(1, N, [N]). & \end{array} \tag{7}$$

From any feasible solution of (6), we can form the affine functions

$$\begin{array}{rcll} x_i^*(\tilde{\boldsymbol{z}}) & = & \frac{1}{\alpha}\left(x_i^0 + \boldsymbol{x}_i'\tilde{\boldsymbol{z}}\right), & \forall i \in [M] \\ y_k^*(\tilde{\boldsymbol{z}}) & = & \frac{1}{\alpha}\left(y_k^0 + \boldsymbol{y}_k'\tilde{\boldsymbol{z}}\right), & \forall k \in [N] \\ s_k^*(\tilde{\boldsymbol{z}}) & = & s_k^0 + \boldsymbol{s}_k'\tilde{\boldsymbol{z}}, & \forall k \in [N] \\ r^*(\tilde{\boldsymbol{z}}) & = & r^0 + \boldsymbol{r}'\tilde{\boldsymbol{z}} & \\ t^*(\tilde{\boldsymbol{z}}) & = & t^0 + \boldsymbol{t}'\tilde{\boldsymbol{z}}. & \end{array}$$

By a similar argument to that in the proof of Theorem 1, any feasible solution to (6) results in a feasible solution to (7). Furthermore, since their objective functions coincide, we have $Z_0^* \geq Z_{LDR}$. ■

Theorem 1 implies that solving the robust optimization problem (6) for the LDR approximation results in feasible candidate LDRs for early start times and crashing decisions. Theorem 2 shows that the objective function value to problem (6) can be used as a lower bound on the unknown optimal value Z_0^*.

4.5.2 Piecewise linear decision rule approximation

In this section, we extend the LDR approximation (6) by using piecewise linear decision rules. This enables us to model the covariance matrix Σ. To do so, we solve the distributionally robust optimization problem

$$\begin{array}{rll} 1 - Z^*_{PWLDR} = & \min & \sup_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}} \left((r(\tilde{\boldsymbol{z}}))^+\right) \\ & & + \sup_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}} \left((1 - \alpha B + \boldsymbol{e}'\boldsymbol{s}(\tilde{\boldsymbol{z}}) + t(\tilde{\boldsymbol{z}}) - r(\tilde{\boldsymbol{z}}))^+\right) \\ & & + \sum_{k=1}^{N}\sum_{m=1}^{L} \sup_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}} \left((c_{km} y_k(\tilde{\boldsymbol{z}}) + \alpha d_{km} - s_k(\tilde{\boldsymbol{z}}))^+\right) \\ & & + \sum_{m=1}^{K} \sup_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}} \left((p_m x_M(\tilde{\boldsymbol{z}}) + \alpha q_m - \alpha p_m \tau - t(\tilde{\boldsymbol{z}}))^+\right) \\ & & + p_K \sum_{(i,j)\in\mathcal{E}} \sup_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}} \left(\left(\alpha \tilde{z}_{\Phi(i,j)} - x_j(\tilde{\boldsymbol{z}}) + x_i(\tilde{\boldsymbol{z}}) - y_{\Phi(i,j)}(\tilde{\boldsymbol{z}})\right)^+\right) \\ & & + p_K \sum_{(i,j)\in\mathcal{E}} \sup_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}} \left((-x_j(\tilde{\boldsymbol{z}}) + x_i(\tilde{\boldsymbol{z}}))^+\right) \\ & & + \sup_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}} \left(\boldsymbol{c}_L' \left(\boldsymbol{y}(\tilde{\boldsymbol{z}})\right)^- + \boldsymbol{d}' \left(\boldsymbol{y}(\tilde{\boldsymbol{z}}) - \alpha \boldsymbol{u}\right)^+\right) \\ & & + p_K \boldsymbol{e}' \sup_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}} \left((\boldsymbol{x}(\tilde{\boldsymbol{z}}))^-\right) \\ & \text{s.t.} & x_i \in \mathcal{L}(1, N, I_i^x), \quad \forall i \in [M] \\ & & y_k \in \mathcal{L}(1, N, I_k^y), \quad \forall k \in [N] \\ & & \boldsymbol{s} \in \mathcal{L}(N, N, [N]) \\ & & r, t \in \mathcal{L}(1, N, [N]), \end{array} \tag{8}$$

where we define the parameter $\boldsymbol{d} \in \Re^N$ as $\boldsymbol{d} \triangleq (p_K - \boldsymbol{c}_1)^+$.

Before defining the decision rules for the original problem, we first define the following sets for notational simplicity. For any $j \in [M]$, let $\mathcal{P}(j) \triangleq \{i \in [M] : (i,j) \in \mathcal{E}\}$,

$$\mathcal{N}(j) \triangleq \{j\} \cup \bigcup_{i\in\mathcal{P}(j)} \mathcal{N}(i), \quad \text{and} \quad \mathcal{A}(j) \triangleq \{(i,j) \in \mathcal{E} : i \in \mathcal{P}(j)\} \cup \bigcup_{i\in\mathcal{P}(j)} \mathcal{A}(i).$$

$\mathcal{P}(j)$ represents the set of nodes that are immediate predecessors of j, while the recursively defined $\mathcal{N}(j)$ represents the set of all nodes that are predecessors of j. Similarly, $\mathcal{A}(j)$ represents the set of arcs that enter into all the predecessor nodes of j. We relate problem (8) to the original problem (4) and the LDR approximation (6) using the two theorems that follow.

Theorem 3 *If model (8) has a feasible solution* $(\alpha, \boldsymbol{x}(\tilde{\boldsymbol{z}}), \boldsymbol{y}(\tilde{\boldsymbol{z}}), \boldsymbol{s}(\tilde{\boldsymbol{z}}), r(\tilde{\boldsymbol{z}}), t(\tilde{\boldsymbol{z}}))$ *where* $\alpha > 0$, *then for each* $m \in [M]$,

$$\begin{aligned} x_m^*(\tilde{\boldsymbol{z}}) &= \frac{1}{\alpha}\left(x_m(\tilde{\boldsymbol{z}}) + \sum_{i\in\mathcal{N}(m)} (x_i(\tilde{\boldsymbol{z}}))^- + \sum_{\substack{(i,j)\in\mathcal{A}(m)\\ k=\Phi(i,j)}} (y_k(\tilde{\boldsymbol{z}}) - \alpha u_k)^+ \right) \\ &\quad + \frac{1}{\alpha}\left(\sum_{\substack{(i,j)\in\mathcal{A}(m)\\ k=\Phi(i,j)}} (\alpha \tilde{z}_k - y_k(\tilde{\boldsymbol{z}}) - x_j(\tilde{\boldsymbol{z}}) + x_i(\tilde{\boldsymbol{z}}))^+ + (-x_j(\tilde{\boldsymbol{z}}) + x_i(\tilde{\boldsymbol{z}}))^+ \right), \end{aligned}$$

and

$$\boldsymbol{y}^*(\tilde{\boldsymbol{z}}) = \min\left\{ \max\left\{ \frac{1}{\alpha}\boldsymbol{y}(\tilde{\boldsymbol{z}}), \boldsymbol{0} \right\}, \boldsymbol{u} \right\}$$

is a feasible solution to (4).

Proof. By construction, the decision rules $\boldsymbol{x}^*$ and $\boldsymbol{y}^*$ satisfy the non-anticipativity requirements. We therefore have:

$$\begin{aligned} x_i^* &\in \mathcal{Y}(1, N, I_x^i), \quad \forall i \in [M] \\ y_k^* &\in \mathcal{Y}(1, N, I_y^k), \quad \forall k \in [N]. \end{aligned}$$

The second constraint of model (4) is satisfied directly from the construction of $\boldsymbol{y}^*(\tilde{\boldsymbol{z}})$. The third constraint is satisfied since, for all $m \in [M]$,

$$\begin{aligned} x_m^*(\tilde{\boldsymbol{z}}) &\geq \frac{1}{\alpha} x_m(\tilde{\boldsymbol{z}}) + \frac{1}{\alpha}(x_m(\tilde{\boldsymbol{z}}))^- \qquad \text{(since } m \in \mathcal{N}(m)\text{)} \\ &= \frac{1}{\alpha}(x_m(\tilde{\boldsymbol{z}}))^+ \\ &\geq 0. \end{aligned}$$

To show that the first constraint of model (4) is satisfied, we consider an arbitrary arc on the network, $(i,j) \in \mathcal{E}$. We first observe that $\mathcal{N}(i) \subset \mathcal{N}(j)$, and $\mathcal{A}(i) \subset \mathcal{A}(j)$. Defining $k = \Phi(i,j)$, we have

$$
\begin{aligned}
&\alpha\left(x_j^*(\tilde{\boldsymbol{z}}) - x_i^*(\tilde{\boldsymbol{z}}) + y_k^*(\tilde{\boldsymbol{z}})\right) \\
&\quad \geq x_j(\tilde{\boldsymbol{z}}) - x_i(\tilde{\boldsymbol{z}}) + (y_k(\tilde{\boldsymbol{z}}) - \alpha u_k)^+ + (\alpha\tilde{z}_k - y_k(\tilde{\boldsymbol{z}}) - x_j(\tilde{\boldsymbol{z}}) + x_i(\tilde{\boldsymbol{z}}))^+ + \alpha y_k^*(\tilde{\boldsymbol{z}}) \\
&\quad = x_j(\tilde{\boldsymbol{z}}) - x_i(\tilde{\boldsymbol{z}}) + (y_k(\tilde{\boldsymbol{z}}) - \alpha u_k)^+ + (\alpha\tilde{z}_k - y_k(\tilde{\boldsymbol{z}}) - x_j(\tilde{\boldsymbol{z}}) + x_i(\tilde{\boldsymbol{z}}))^+ \\
&\qquad\quad + y_k(\tilde{\boldsymbol{z}}) + (y_k(\tilde{\boldsymbol{z}}))^- - (y_k(\tilde{\boldsymbol{z}}) - \alpha u_k)^+ \\
&\quad \geq \alpha\tilde{z}_k + (x_j(\tilde{\boldsymbol{z}}) - x_i(\tilde{\boldsymbol{z}}) + y_k(\tilde{\boldsymbol{z}}) - \alpha\tilde{z}_k) + (\alpha\tilde{z}_k - y_k(\tilde{\boldsymbol{z}}) - x_j(\tilde{\boldsymbol{z}}) + x_i(\tilde{\boldsymbol{z}}))^+ \\
&\quad = \alpha\tilde{z}_k + (x_j(\tilde{\boldsymbol{z}}) - x_i(\tilde{\boldsymbol{z}}) + y_k(\tilde{\boldsymbol{z}}) - \alpha\tilde{z}_k)^+ \\
&\quad \geq \alpha\tilde{z}_k.
\end{aligned} \tag{9}
$$

The first inequality results from the inclusions $\mathcal{N}(i) \subset \mathcal{N}(j)$ and $\mathcal{A}(i) \subset \mathcal{A}(j)$, which cancel out many of the common terms. The remainder of the terms are nonnegative and give rise to the inequality. The final equation is a result of applying the identity $x + x^- \equiv x^+$. Similarly, we have

$$
\begin{aligned}
\alpha\left(x_j^*(\tilde{\boldsymbol{z}}) - x_i^*(\tilde{\boldsymbol{z}})\right) &\geq x_j(\tilde{\boldsymbol{z}}) - x_i(\tilde{\boldsymbol{z}}) + (-x_j(\tilde{\boldsymbol{z}}) + x_i(\tilde{\boldsymbol{z}}))^+ \\
&= (x_j(\tilde{\boldsymbol{z}}) - x_i(\tilde{\boldsymbol{z}}))^+ \\
&\geq 0.
\end{aligned} \tag{10}
$$

Together, (9) and (10) imply the first constraint of (4), $x_j^*(\tilde{\boldsymbol{z}}) - x_i^*(\tilde{\boldsymbol{z}}) \geq (\tilde{z} - y_k^*(\tilde{\boldsymbol{z}}))^+$.

Since all the constraints are satisfied, the piecewise-linear decision rules $\boldsymbol{x}^*$ and $\boldsymbol{y}^*$ are feasible in (4). ■

In order to prove the next theorem, we need the following preliminary result.

Lemma 1 *Any convex piecewise-linear function of a scalar-valued decision rule, with any information index set I, i.e. $x \in \mathcal{Y}(1, N, I)$, can be equivalently expressed as*

$$
\max_{k\in[K]} \{a_k x(\tilde{\boldsymbol{z}}) + b_k\} = \min_{t\in\mathcal{Y}(1,N,[N])} t(\tilde{\boldsymbol{z}}) + \sum_{k=1}^{K} (a_k x(\tilde{\boldsymbol{z}}) + b_k - t(\tilde{\boldsymbol{z}}))^+ .
$$

Proof. For any $\boldsymbol{z} \in \mathcal{W}$, define $V(\boldsymbol{z}) \triangleq \max_{k\in[K]} \{a_k x(\boldsymbol{z}) + b_k\}$. Since $\boldsymbol{z}$ is fixed, $x(\boldsymbol{z})$ is a known scalar, and therefore

$$
\max_{k\in[K]} \{a_k x(\boldsymbol{z}) + b_k\} = \min_{v\in\Re} v + \sum_{k=1}^{K} (a_k x(\boldsymbol{z}) + b_k - v)^+ .
$$

Further, define $v^* : \Re^N \to \Re$, by

$$
v^*(\boldsymbol{z}) \triangleq \arg\min_{v\in\Re} v + \sum_{k=1}^{K} (a_k x(\boldsymbol{z}) + b_k - v)^+ . \tag{11}
$$

By definition, $v^* \in \mathcal{Y}(1, N, [N])$. Hence, $V(\boldsymbol{z}) \geq \min_{t\in\mathcal{Y}(1,N,[N])} \{t(\boldsymbol{z}) + \sum_{k=1}^{K} (a_k x(\boldsymbol{z}) + b_k - t(\boldsymbol{z}))^+\}$. Suppose, for purposes of contradiction, that the previous inequality is strict for some $\boldsymbol{z}^* \in \mathcal{W}$

for the optimized decision rule t^*. This implies that

$$t^*(\boldsymbol{z}^*) + \sum_{k=1}^{K} (a_k x(\boldsymbol{z}^*) + b_k - t^*(\boldsymbol{z}^*))^+ < v^*(\boldsymbol{z}^*) + \sum_{k=1}^{K} (a_k x(\boldsymbol{z}^*) + b_k - v^*(\boldsymbol{z}^*))^+ ,$$

which violates the definition of v^* in (11). Hence, we have

$$V(\boldsymbol{z}) = \min_{t \in \mathcal{Y}(1,N,[N])} t(\boldsymbol{z}) + \sum_{k=1}^{K} (a_k x(\boldsymbol{z}) + b_k - t(\boldsymbol{z}))^+ ,$$

for all $\boldsymbol{z} \in \mathcal{W}$, which implies the statement in the lemma. ■

Theorem 4 *The following inequality holds:* $Z_0^* \geq Z_{PWLDR}^* \geq Z_{LDR}$.

Proof. Given a feasible piecewise-linear decision rule $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ obtained from model (8), Theorem 3 shows that it is also feasible in (4). Under this decision rule, the objective becomes

$$
\begin{aligned}
& 1-Z_0^* \\
= \quad & \min \sup_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}} \left(\left(1-\alpha B+\alpha \sum_{k=1}^{N} \max_{m\in[L]} \left\{ c_{km} y_k^*(\tilde{\boldsymbol{z}}) + d_{km} \right\} + \alpha \max_{m\in[K]} \left\{ p_m x_M^*(\tilde{\boldsymbol{z}}) - p_m \tau + q_m \right\} \right)^+ \right) \\
\leq \quad & \min \sup_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}} \left(\begin{pmatrix} 1-\alpha B \\ \quad + \sum_{k=1}^{N} \max_{m\in[L]} \left\{ c_{km} y_k(\tilde{\boldsymbol{z}}) + d_{km} \right\} + \max_{m\in[K]} \left\{ p_m x_m(\tilde{\boldsymbol{z}}) - \alpha p_m \tau + \alpha q_m \right\} \\ \quad + \sum_{\substack{(i,j)\in\mathcal{A}(M) \\ k=\Phi(i,j)}} c_{kL} \left(y_k(\tilde{\boldsymbol{z}}) \right)^- + \sum_{\substack{(i,j)\in\mathcal{A}(M) \\ k=\Phi(i,j)}} (p_K - c_{k1})^+ \left(y_k(\tilde{\boldsymbol{z}}) - \alpha u_k \right)^+ \\ \quad + p_K \sum_{\substack{(i,j)\in\mathcal{A}(M) \\ k=\Phi(i,j)}} \left(\alpha \tilde{z}_k - y_k(\tilde{\boldsymbol{z}}) - x_j(\tilde{\boldsymbol{z}}) + x_i(\tilde{\boldsymbol{z}}) \right)^+ \\ \quad + p_K \sum_{\substack{(i,j)\in\mathcal{A}(M) \\ k=\Phi(i,j)}} \left(-x_j(\tilde{\boldsymbol{z}}) + x_i(\tilde{\boldsymbol{z}}) \right)^+ + p_K \sum_{i\in\mathcal{N}(M)} \left(x_i(\tilde{\boldsymbol{z}}) \right)^- \end{pmatrix}^+ \right) \\
\leq \quad & \min \sup_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}} \left(\begin{pmatrix} 1-\alpha B + \sum_{k=1}^{N} s_k(\tilde{\boldsymbol{z}}) + t(\tilde{\boldsymbol{z}}) \\ \quad + \sum_{k=1}^{N} \sum_{m=1}^{L} \left(c_{km} y_k(\tilde{\boldsymbol{z}}) + d_{km} - s_k(\tilde{\boldsymbol{z}}) \right)^+ \\ \quad + \sum_{m=1}^{L} \left(p_m x_m(\tilde{\boldsymbol{z}}) - \alpha p_m \tau + \alpha q_m - t(\tilde{\boldsymbol{z}}) \right)^+ \\ \quad + \sum_{k=1}^{N} c_{kL} \left(y_k(\tilde{\boldsymbol{z}}) \right)^- + \sum_{k=1}^{N} (p_K - c_{k1})^+ \left(y_k(\tilde{\boldsymbol{z}}) - \alpha u_k \right)^+ \\ \quad + p_K \sum_{(i,j)\in\mathcal{E}} \left(\alpha \tilde{z}_{\Phi(i,j)} - y_{\Phi(i,j)}(\tilde{\boldsymbol{z}}) - x_j(\tilde{\boldsymbol{z}}) + x_i(\tilde{\boldsymbol{z}}) \right)^+ \\ \quad + p_K \sum_{(i,j)\in\mathcal{E}} \left(-x_j(\tilde{\boldsymbol{z}}) + x_i(\tilde{\boldsymbol{z}}) \right)^+ + p_K \sum_{i=1}^{M} \left(x_i(\tilde{\boldsymbol{z}}) \right)^- \end{pmatrix}^+ \right) \\
\leq \quad & 1 - Z_{PWLDR}^*.
\end{aligned}
$$

The first inequality results from applying the definitions of $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ and the assumptions $p_K \geq p_m, \forall m \in [K]$ and $c_{kL} \geq c_{km} \geq c_{k1}, \forall m \in [K], k \in [N]$. We also use the inequality $(p_K - c_{k1})^+ \geq (p_K - c_{k1})$. The second inequality is a consequence of applying Lemma 1, and introducing auxiliary decision rules $\boldsymbol{s} \in \mathcal{Y}(N, N, [N])$ and $t \in \mathcal{Y}(1, N, [N])$. Also, we use the fact that since M is the terminal node, $\mathcal{N}(M) = [M]$ and $\mathcal{A}(M) = \mathcal{E}$, which simplifies the summations. Finally, the last inequality results from adding and subtracting another auxiliary decision rule $r \in \mathcal{Y}(1, N, [N])$ and applying subadditivity of the $(\cdot)^+$ and $\sup_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}}(\cdot)$ operators.

We now show that $Z_{PWLDR}^* \geq Z_{LDR}$. For any feasible solution to (6), we form the following affine functions

$$\begin{array}{rcll} x_i(\tilde{\boldsymbol{z}}) & = & (x_i^0 + \boldsymbol{x}_i'\tilde{\boldsymbol{z}}), & \forall i \in [M] \\ y_k(\tilde{\boldsymbol{z}}) & = & (y_k^0 + \boldsymbol{y}_k'\tilde{\boldsymbol{z}}), & \forall k \in [N] \\ s_k(\tilde{\boldsymbol{z}}) & = & s_k^0 + \boldsymbol{s}_k'\tilde{\boldsymbol{z}}, & \forall k \in [N] \\ r(\tilde{\boldsymbol{z}}) & = & r^0 + \boldsymbol{r}'\tilde{\boldsymbol{z}} & \\ t(\tilde{\boldsymbol{z}}) & = & t^0 + \boldsymbol{t}'\tilde{\boldsymbol{z}}, & \end{array}$$

which are feasible for (8). Moreover, since the inequalities of (6) are satisfied by a feasible solution, the nonlinear terms in the objective of (8) vanish, and the expression for the objective coincides with (6). Hence, we have ${Z_0}^* \geq Z_{PWLDR}^* \geq Z_{LDR}$, as required. ■

In problem (8), the only explicit constraints remaining are the non-anticipativity constraints for the decision rules. The other constraints are implicitly satisfied by our construction of $\boldsymbol{x}^*(\tilde{\boldsymbol{z}})$ and $\boldsymbol{y}^*(\tilde{\boldsymbol{z}})$ in Theorem 3. This property is atypical in distributionally robust optimization models, and follows from the particular structure of the project crashing problem. In particular, the proof of Theorem 3 relies substantially on the properties that $\mathcal{N}(i) \subset \mathcal{N}(j)$ and $\mathcal{A}(i) \subset \mathcal{A}(j)$, for any arc $(i,j) \in \mathcal{E}$ on the project network. Together with the linearity of the constraints, this allows problem (8) to be expressed as shown.

Theorems 3 and 4, respectively, show that the piecewise linear rules are feasible decision rules and lead to a tighter approximation of the optimal objective Z_0^* than LDRs. However, solving problem (8) is not as straightforward as solving (6), due to the piecewise-linear terms within the expectations. Nonetheless, we can solve this problem approximately, by using the following upper bound.

Proposition 1 *(Goh and Sim 2010)*
If $\mathbb{F} = \left\{\mathbb{P} : \mathbb{P}\left(\tilde{\boldsymbol{z}} \in \mathcal{W}\right) = 1, \hat{\boldsymbol{z}} = \mathrm{E}_{\mathbb{P}}\left(\tilde{\boldsymbol{z}}\right) \in \hat{\mathcal{W}}, \mathrm{E}_{\mathbb{P}}\left(\tilde{\boldsymbol{z}}\tilde{\boldsymbol{z}}'\right) = \boldsymbol{\Sigma} + \hat{\boldsymbol{z}}\hat{\boldsymbol{z}}'\right\}$, *then*

$$\pi\left(y^0, \boldsymbol{y}\right) \triangleq \min_{x^0, \boldsymbol{x}} \pi^1\left(x^0, \boldsymbol{x}\right) + \pi^2\left(y^0 - x^0, \boldsymbol{y} - \boldsymbol{x}\right) \geq \sup_{\mathbb{P} \in \mathbb{F}} \mathrm{E}_{\mathbb{P}}\left(\left(y^0 + \boldsymbol{y}\tilde{\boldsymbol{z}}\right)^+\right),$$

where

$$\begin{array}{rcl} \pi^1\left(y^0, \boldsymbol{y}\right) & \triangleq & \inf_{\boldsymbol{s} \in \Re^N} \left(\sup_{\hat{\boldsymbol{z}} \in \hat{\mathcal{W}}} \left\{\boldsymbol{s}'\hat{\boldsymbol{z}}\right\} + \sup_{\boldsymbol{z} \in \mathcal{W}} \left(\max\left\{y^0 + \boldsymbol{y}'\boldsymbol{z} - \boldsymbol{s}'\boldsymbol{z}, -\boldsymbol{s}'\boldsymbol{z}\right\}\right)\right), \\ \pi^2\left(y^0, \boldsymbol{y}\right) & \triangleq & \sup_{\hat{\boldsymbol{z}} \in \hat{\mathcal{W}}} \left\{ \frac{1}{2}\left(y^0 + \boldsymbol{y}'\hat{\boldsymbol{z}}\right) + \frac{1}{2}\sqrt{(y^0 + \boldsymbol{y}'\hat{\boldsymbol{z}})^2 + \boldsymbol{y}'\boldsymbol{\Sigma}\boldsymbol{y}} \right\}. \end{array}$$

Using these bounds, we obtain the following approximate solution to problem (8),

$$\begin{array}{rll} 1 - Z_{PWLDR}{}^{(1)} = & \min & \pi\left(r^0, \boldsymbol{r}\right) + \pi\left(1 - \alpha B + \sum_{k=1}^{N} s_k^0 + t^0 - r^0, \sum_{k=1}^{N} \boldsymbol{s}_k + \boldsymbol{t} - \boldsymbol{r}\right) \\ & & + \displaystyle\sum_{k=1}^{N}\sum_{m=1}^{L} \pi\left(c_{km} y_k^0 + \alpha d_{km} - s_k^0, c_{km}\boldsymbol{y}_k - \boldsymbol{s}_k\right) \\ & & + \displaystyle\sum_{m=1}^{K} \pi\left(p_m x_M^0 + \alpha q_m - \alpha p_m \tau - t^0, p_m \boldsymbol{x}_M - \boldsymbol{t}\right) \\ & & + \displaystyle\sum_{(i,j)\in\mathcal{E}} p_K \pi\left(-x_j^0 + x_i^0 - y_{\Phi(i,j)}^0, \alpha \boldsymbol{e}^{\Phi(i,j)} - \boldsymbol{x}_j + \boldsymbol{x}_i - \boldsymbol{y}_{\Phi(i,j)}\right) \\ & & + \displaystyle\sum_{(i,j)\in\mathcal{E}} p_K \pi\left(-x_j^0 + x_i^0, -\boldsymbol{x}_j + \boldsymbol{x}_i\right) + \sum_{k=1}^{M} p_K \pi\left(-x_k^0, -\boldsymbol{x}_k\right) \\ & & + \displaystyle\sum_{k=1}^{N} c_{kL} \pi\left(-y_k^0, -\boldsymbol{y}_k\right) + \left(p_K - c_{k1}\right)^+ \pi\left(y_k^0 - \alpha u_k, \boldsymbol{y}_k\right) \\ & \text{variables} & \alpha, t^0, \boldsymbol{t}, r^0, \boldsymbol{r}, \left\{x_i^0, \boldsymbol{x}_i\right\}_{i=1}^{M}, \left\{y_k^0, \boldsymbol{y}_k, s_k^0, \boldsymbol{s}_k\right\}_{k=1}^{N} \\ & \text{s.t.} & x_{ij} = 0, \qquad \forall i \in [M], \forall j \notin I_i^x \\ & & y_{kj} = 0, \qquad \forall k \in [N], \forall j \notin I_i^y. \end{array} \tag{12}$$

From the structure of $\pi\left(y^0, \boldsymbol{y}\right)$, for some decision variable w, the constraint $\pi\left(y^0, \boldsymbol{y}\right) \leq w$ can be expressed in terms of second order conic constraints. Such problems are both theoretically and computationally tractable (Nesterov and Nemirovski 1994).

In order to establish the relative strength of the various bounds on Z_0^* described above, we need the following two preliminary results.

Lemma 2 *Consider a scalar-valued LDR with an arbitrary information index set,* $y \in \mathcal{L}(1, N, I)$*.*

1. *If* $y(\tilde{\boldsymbol{z}}) = y^0 + \boldsymbol{y}'\tilde{\boldsymbol{z}} \leq 0$*, then* $\pi^1\left(y^0, \boldsymbol{y}\right) \leq 0$*.*

2. *If* $y(\tilde{\boldsymbol{z}}) = y^0 + \boldsymbol{y}'\tilde{\boldsymbol{z}} \geq 0$*, then* $\pi^1\left(y^0, \boldsymbol{y}\right) \leq \sup_{\hat{\boldsymbol{z}} \in \hat{\mathcal{W}}} \left\{y^0 + \boldsymbol{y}'\hat{\boldsymbol{z}}\right\}$*.*

Proof. Applying the definition of $\pi^1\left(y^0, \boldsymbol{y}\right)$, the assumption $y(\tilde{\boldsymbol{z}}) \leq 0$ implies that $\pi^1\left(y^0, \boldsymbol{y}\right) = \inf_{\boldsymbol{s} \in \Re^N} \left(\sup_{\hat{\boldsymbol{z}} \in \hat{\mathcal{W}}} \left\{\boldsymbol{s}'\hat{\boldsymbol{z}}\right\} + \sup_{\boldsymbol{z} \in \mathcal{W}} \left\{-\boldsymbol{s}'\boldsymbol{z}\right\} \right)$. Since $\boldsymbol{s} = \boldsymbol{0}$ is feasible, we have $\pi^1\left(y^0, \boldsymbol{y}\right) \leq 0$. Similarly, the assumption $y(\tilde{\boldsymbol{z}}) \geq 0$ implies that $\pi^1\left(y^0, \boldsymbol{y}\right) = \inf_{\boldsymbol{s} \in \Re^N} \left(\sup_{\hat{\boldsymbol{z}} \in \hat{\mathcal{W}}} \left\{\boldsymbol{s}'\hat{\boldsymbol{z}}\right\} + \sup_{\boldsymbol{z} \in \mathcal{W}} \left\{y^0 + \boldsymbol{y}'\boldsymbol{z} - \boldsymbol{s}'\boldsymbol{z}\right\} \right)$. Choosing $\boldsymbol{s} = \boldsymbol{y}$ yields $\pi^1\left(y^0, \boldsymbol{y}\right) \leq \sup_{\hat{\boldsymbol{z}} \in \hat{\mathcal{W}}} \left\{y^0 + \boldsymbol{y}'\hat{\boldsymbol{z}}\right\}$. ■

Lemma 3 *Consider a scalar-valued LDR with an arbitrary information index set,* $y \in \mathcal{L}(1, N, I)$*. If* $y(\tilde{\boldsymbol{z}}) = y^0 + \boldsymbol{y}'\tilde{\boldsymbol{z}} \leq 0$*, then* $\pi\left(y^0, \boldsymbol{y}\right) = 0$*.*

Proof. $y(\tilde{\boldsymbol{z}}) \leq 0$ implies $\sup_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}}\left(\left(y^0+\boldsymbol{y}'\tilde{\boldsymbol{z}}\right)^+\right)=0$. Using Proposition 1, we have $\pi\left(y^0,\boldsymbol{y}\right)\geq 0$. However,

$$
\begin{array}{rcll}
\pi\left(y^0,\boldsymbol{y}\right) & = & \min_{x^0,\boldsymbol{x}} \pi^1\left(x^0,\boldsymbol{x}\right)+\pi^2\left(y^0-x^0,\boldsymbol{y}-\boldsymbol{x}\right) & \\
& \leq & \pi^1\left(y^0,\boldsymbol{y}\right)+\pi^2\left(0,\boldsymbol{0}\right) & \text{(choosing } (x^0,\boldsymbol{x})=(y^0,\boldsymbol{y})\text{)} \\
& = & \pi^1\left(y^0,\boldsymbol{y}\right) & \text{(by positive homogeneity of } \pi^2(\cdot)\text{)} \\
& \leq & 0. & \text{(by Lemma 2, part 1)}
\end{array}
$$

Hence, we have $\pi\left(y^0,\boldsymbol{y}\right)=0$. ■

We are now ready to prove our main result about the relative strength of the various bounds.

Theorem 5 *The following inequality holds:* $Z_0^* \geq Z^*_{PWLDR} \geq Z^{(1)}_{PWLDR} \geq Z_{LDR}$.

Proof. Consider a feasible solution to (6). Observe that this solution is also feasible in (12). The objective of (12) is simplified by iteratively applying Lemma 3 to each constraint of model (6), which causes all the $\pi(\cdot)$ terms except the first to vanish. Thus, we have

$$
\begin{array}{rcll}
1-Z^{(1)}_{PWLDR} & \leq & \pi(r^0,\boldsymbol{r}) & \\
& \leq & \pi^1(r^0,\boldsymbol{r}) & \text{(from the proof of Lemma 3)} \\
& \leq & \sup_{\hat{\boldsymbol{z}}\in\hat{\mathcal{W}}}\left\{r^0+\boldsymbol{r}'\hat{\boldsymbol{z}}\right\} & \text{(by Lemma 2, part 2)} \\
& = & 1-Z_{LDR}. &
\end{array}
$$

Therefore, we have $Z^{(1)}_{PWLDR} \geq Z_{LDR}$.

The other inequality $Z^*_{PWLDR} \geq Z^{(1)}_{PWLDR}$ follows, first from applying subadditivity of the $\sup_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}}\left(\cdot\right)$ operator to obtain

$$
\begin{array}{l}
\sup_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}}\left((\boldsymbol{y}(\tilde{\boldsymbol{z}}))^-+(p_K-\boldsymbol{c}_1')^+\left(\boldsymbol{y}(\tilde{\boldsymbol{z}})-\alpha\boldsymbol{u}\right)^+\right) \\
\quad \leq \sup_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}}\left((\boldsymbol{y}(\tilde{\boldsymbol{z}}))^-\right)+\sup_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}}\left((p_K-\boldsymbol{c}_1')^+\left(\boldsymbol{y}(\tilde{\boldsymbol{z}})-\alpha\boldsymbol{u}\right)^+\right)
\end{array}
$$

in the objective of (8), and second from iteratively applying Proposition 1 to bound the respective terms in the objective of (8) from above. ■

This implies that we can obtain decision rules which perform better than the simpler LDRs, at the cost of additional computational complexity in solving (12). This improvement arises from two sources. First, the piecewise-linear rules explore a larger space of possible decisions. Second, by using the piecewise-linear rules and bounds developed, we can incorporate correlation information between different activities, through the covariance matrix $\boldsymbol{\Sigma}$ which appears in $\pi^2(\cdot)$.

4.5.3 Crashing decisions for the example project

For the example project described in Section 4.4, after solving model (6) and applying Theorem 1, we obtain the LDR for crashing amounts as $\boldsymbol{y}_L(\tilde{\boldsymbol{z}}) = [0, \tilde{z}_1, 0, 1]'$. Similarly, solving model (12) and applying Theorem 3, the PWLDR solution can be simplified to $\boldsymbol{y}_P(\tilde{\boldsymbol{z}}) = [0, 0.215 + 0.785\tilde{z}_1, 0, 1]'$.

We compare the performance of the LDR solution with the other heuristics shown in Table 1. We simulate the uncertain activity times as beta random variables (Kerzner 2009) with the given statistics. In particular, we use the mixture approach of Michael and Schucany (2002) to simulate the beta-distributed correlated activity times. In our simulation, we first generate the activity time, and then subtract the crash amount from the activity time. If the reduced activity time is negative, then we assign it to zero; however, we still include the full crash amount in our cost computation. This convention models situations where outsourced work components are billable even if they are not fully utilized (Greaver 1999).

We perform 10,000 independent simulation runs using each strategy to compute the project success probability, that is, the probability that the sum of the crashing and penalty costs is within the overall project budget. These probabilities are summarized in Table 2.

Strategy	Probability
NOCRASH	38.76%
MINCRASH	57.38%
MCSIM	78.70%
LDR	86.10%
PWLDR	86.92%

Table 2: Probability of Meeting Overall in the Example.

The results illustrate that, for this simple example, the solutions obtained from our LDR and PWLDR strategies yield higher success probabilities than all the previous methods. More extensive computational tests in Section 5 further demonstrate that the benefits in project performance which result from the use of our decision rules are typically substantial.

4.5.4 An alternative bound

Model (12) is a natural approximation of (8), but not a unique one. As an alternative, we can apply a different bound on some terms to construct a new approximation. Specifically, one of the worst-case expectation terms in the objective of (8) has an argument that can be decomposed into the sum of piecewise-linear convex terms, as follows.

$$\begin{aligned}
&\sup_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}}\left(\boldsymbol{c}_L'\left(\boldsymbol{y}(\tilde{\boldsymbol{z}})\right)^{-} + \boldsymbol{d}'\left(\boldsymbol{y}(\tilde{\boldsymbol{z}}) - \alpha\boldsymbol{u}\right)^{+}\right)\\
&\quad = \sup_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}}\left(\sum_{k=1}^{N} \max\left\{-c_{kL}y_k(\tilde{\boldsymbol{z}}), 0, d_k y_k(\tilde{\boldsymbol{z}}) - \alpha u_k d_k\right\}\right)\\
&\quad \le \sum_{k=1}^{N} \sup_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}}\left(\max\left\{-c_{kL}y_k(\tilde{\boldsymbol{z}}), 0, d_k y_k(\tilde{\boldsymbol{z}}) - \alpha u_k d_k\right\}\right).
\end{aligned}$$

We make use of the following result.

Proposition 2 *(Natarajan et al. 2010)*
If $\mathbb{F} = \left\{\mathbb{P} : \mathbb{P}\left(\tilde{\boldsymbol{z}} \in \mathcal{W}\right) = 1, \hat{\boldsymbol{z}} = \mathrm{E}_{\mathbb{P}}\left(\tilde{\boldsymbol{z}}\right) \in \hat{\mathcal{W}}, \mathrm{E}_{\mathbb{P}}\left(\tilde{\boldsymbol{z}}\tilde{\boldsymbol{z}}'\right) = \boldsymbol{\Sigma} + \hat{\boldsymbol{z}}\hat{\boldsymbol{z}}'\right\}$, *and given parameters* $\boldsymbol{a}, \boldsymbol{b} \in \Re^K$ *for some positive integer* $K \ge 2$, *then*

$$\eta(y^0, \boldsymbol{y}; \boldsymbol{a}, \boldsymbol{b}) \triangleq \min_{\boldsymbol{w},\boldsymbol{x}} \eta^1\left(\boldsymbol{w}, \boldsymbol{x}\right) + \eta^2\left(\boldsymbol{a}y^0 + \boldsymbol{b} - \boldsymbol{w}, \boldsymbol{y} - \boldsymbol{x}\right) \ge \sup_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}}\left(\max_{i\in[K]}\left\{a_i\left(y^0 + \boldsymbol{y}'\tilde{\boldsymbol{z}}\right) + b_i\right\}\right),$$

where

$$\begin{aligned}
\eta^1\left(\boldsymbol{w}, \boldsymbol{y}\right) &\triangleq \inf_{\boldsymbol{s}\in\Re^N} \left(\sup_{\hat{\boldsymbol{z}}\in\hat{\mathcal{W}}}\left\{\boldsymbol{s}'\hat{\boldsymbol{z}}\right\} + \sup_{\boldsymbol{z}\in\mathcal{W}}\left(\max_{i\in[K]}\left\{w_i + \left(a_i\boldsymbol{y} - \boldsymbol{s}\right)'\boldsymbol{z}\right\}\right)\right),\\
\eta^2\left(\boldsymbol{w}, \boldsymbol{y}\right) &\triangleq \inf_{s,t,v\in\Re} \left(s + \max_{i\in[K]}\left\{a_i^2 v + a_i t + w_i + \sup_{\hat{\boldsymbol{z}}\in\hat{\mathcal{W}}} \boldsymbol{y}'\hat{\boldsymbol{z}}\right\}\right)\\
&\qquad \text{s.t.} \quad \boldsymbol{y}'\boldsymbol{\Sigma}\boldsymbol{y} + t^2 \le 4sv\\
&\qquad\qquad\quad v \ge 0.
\end{aligned}$$

This bound was shown by Natarajan et al. (2010) to be exact if $\mathbb{F}$ is defined by mean and support information, or if $\mathbb{F}$ is defined by mean and covariance information.

Using this bound, and defining the set of parameters $\forall k \in [N]$ as

$$\begin{aligned}
\boldsymbol{a}_k &\triangleq [0, -c_{kL}, d_k]'\\
\boldsymbol{b}_k &\triangleq [0, 0, \alpha u_k d_k]',
\end{aligned} \tag{13}$$

we obtain the following alternative approximation of (8),

$$
\begin{aligned}
1 - Z^{(2)}_{PWLDR} = \quad \min \quad & \pi\left(r^0, \boldsymbol{r}\right) + \pi\left(1 - \alpha B + \textstyle\sum_{k=1}^N s_k^0 + t^0 - r^0, \sum_{k=1}^N \boldsymbol{s}_k + \boldsymbol{t} - \boldsymbol{r}\right) \\
& + \sum_{k=1}^N \sum_{m=1}^L \pi\left(c_{km} y_k^0 + \alpha d_{km} - s_k^0, c_{km} \boldsymbol{y}_k - \boldsymbol{s}_k\right) \\
& + \sum_{m=1}^K \pi\left(p_m x_M^0 + \alpha q_m - \alpha p_m \tau - t^0, p_m \boldsymbol{x}_M - \boldsymbol{t}\right) \\
& + \sum_{(i,j)\in\mathcal{E}} p_K \pi\left(-x_j^0 + x_i^0 - y^0_{\Phi(i,j)}, \alpha \boldsymbol{e}^{\Phi(i,j)} - \boldsymbol{x}_j + \boldsymbol{x}_i - \boldsymbol{y}_{\Phi(i,j)}\right) \\
& + \sum_{(i,j)\in\mathcal{E}} p_K \pi\left(-x_j^0 + x_i^0, -\boldsymbol{x}_j + \boldsymbol{x}_i\right) \\
& + \sum_{k=1}^N \eta\left(y_k^0, \boldsymbol{y}_k; \boldsymbol{a}_k, \boldsymbol{b}_k\right) + \sum_{k=1}^M p_K \pi\left(-x_k^0, -\boldsymbol{x}_k\right) \\
\text{variables} \quad & \alpha, t^0, \boldsymbol{t}, r^0, \boldsymbol{r}, \left\{x_i^0, \boldsymbol{x}_i\right\}_{i=1}^M, \left\{y_k^0, \boldsymbol{y}_k, s_k^0, \boldsymbol{s}_k\right\}_{k=1}^N \\
\text{s.t.} \quad & x_{ij} = 0 \quad \forall i \in [M], \forall j \notin I_i^x \\
& y_{kj} = 0 \quad \forall k \in [N], \forall j \notin I_i^y.
\end{aligned}
\tag{14}
$$

The new bound $Z^{(2)}_{PWLDR}$ can be compared with the earlier bounds, as follows.

Theorem 6 *The following inequality holds:* $Z_0^* \geq Z^*_{PWLDR} \geq Z^{(2)}_{PWLDR} \geq Z_{LDR}$.

Proof. First, for any $k \in [N]$, if $\exists y_k^0, \boldsymbol{y}_k$ such that $\forall \boldsymbol{z} \in \mathcal{W}, 0 \leq y_k^0 + \boldsymbol{y}_k' \boldsymbol{z} \leq \alpha u_k$ then, for $(\boldsymbol{a}_k, \boldsymbol{b}_k)$ as defined in (13), we have

$$
\begin{aligned}
\eta\left(y_k^0, \boldsymbol{y}_k; \boldsymbol{a}_k, \boldsymbol{b}_k\right) & \leq \eta^1\left(\boldsymbol{a}_k y_k^0 + \boldsymbol{b}_k, \boldsymbol{y}_k; \boldsymbol{a}_k\right) \\
& = \inf_{\boldsymbol{s}\in\Re^N} \left(\sup_{\hat{\boldsymbol{z}}\in\hat{\mathcal{W}}} \left\{\boldsymbol{s}'\hat{\boldsymbol{z}}\right\} + \sup_{\boldsymbol{z}\in\mathcal{W}} \left(\max_{i\in[K]} \left\{ a_{ki}\left(y_k^0 + \boldsymbol{y}'\boldsymbol{z}\right) + b_{ki} - \boldsymbol{s}'\boldsymbol{z} \right\} \right) \right) \\
& = \inf_{\boldsymbol{s}\in\Re^N} \left(\sup_{\hat{\boldsymbol{z}}\in\hat{\mathcal{W}}} \left\{\boldsymbol{s}'\hat{\boldsymbol{z}}\right\} + \sup_{\boldsymbol{z}\in\mathcal{W}} \left(\max\left\{0, -c_{kL}\left(y_k^0 + \boldsymbol{y}'\boldsymbol{z}\right), d_k\left(y_k^0 + \boldsymbol{y}'\boldsymbol{z} - \alpha u_k\right)\right\} - \boldsymbol{s}'\boldsymbol{z} \right) \right) \\
& = \inf_{\boldsymbol{s}\in\Re^N} \left(\sup_{\hat{\boldsymbol{z}}\in\hat{\mathcal{W}}} \left\{\boldsymbol{s}'\hat{\boldsymbol{z}}\right\} + \sup_{\boldsymbol{z}\in\mathcal{W}} \left\{\boldsymbol{s}'\boldsymbol{z}\right\} \right) \\
& \leq 0.
\end{aligned}
$$

Since Proposition 2 implies that $\eta\left(y_k^0, \boldsymbol{y}_k; \boldsymbol{a}_k, \boldsymbol{b}_k\right) \geq 0$, we conclude that under these conditions, $\eta\left(y_k^0, \boldsymbol{y}_k; \boldsymbol{a}_k, \boldsymbol{b}_k\right) = 0$. By an identical argument to that in Theorem 5, any feasible solution to (6) is also feasible in (14) with matching objective, and hence $Z^{(2)}_{PWLDR} \geq Z_{LDR}$.

Similarly, applying Propositions 1 and 2 to (8) yields $Z^*_{PWLDR} \geq Z^{(2)}_{PWLDR}$. ■

4.6 Rolling horizon crashing model

To make the most effective use of periodic information updates as well as activity completion time information, we develop a rolling horizon crashing model. As described in Section 2.3, an updated project report is received in each period. Then the project manager updates the model parameters based on the realized uncertainties, and solves problem (4).

The solution follows an iterative procedure. The iteration number is denoted by $t \geq 0$. For notational simplicity, in the tth period, we define the vector of unfinished fraction of each activity as $\boldsymbol{f}^{(t)} \in [0,1]^N$. For example, if activity k has been completed (respectively, has not started) prior to the tth report, then $f_k^{(t)} = 0$ (resp., $f_k^{(t)} = 1$). Also, we denote by $\boldsymbol{g}^{(t)} \in [0,1]^N$ the vector of fractional work completed for each activity in period t. Accordingly, we have

$$\boldsymbol{g}^{(t)} = \boldsymbol{f}^{(t-1)} - \boldsymbol{f}^{(t)}. \tag{15}$$

Furthermore, we define by $\boldsymbol{D}_{\boldsymbol{f}}^{(t)}$ the $N \times N$ diagonal matrix with $\boldsymbol{f}^{(t)}$ on the main diagonal, and $\boldsymbol{D}_{\boldsymbol{g}}^{(t)}$ the $N \times N$ diagonal matrix with $\boldsymbol{g}^{(t)}$ on the main diagonal. The solution procedure uses the following algorithm.

Algorithm CrashPlan

Step 1. Initialize $t = 0, \mathbb{F}^{(0)} = \mathbb{F}, \boldsymbol{f}^{(0)} = \boldsymbol{e}, \boldsymbol{u}^{(0)} = \boldsymbol{u}, \tau^{(0)} = \tau, B^{(0)} = B$.

Step 2. Solve the following problem:

$$\begin{array}{rll} Z_0(t)^* = & \max\limits_{\boldsymbol{x}^{(t)}(\cdot),\boldsymbol{y}^{(t)}(\cdot),\alpha} & \inf\limits_{\mathbb{P}\in\mathbb{F}^{(t)}} \mathrm{E}_{\mathbb{P}}\left(\min\left\{1, \alpha B^{(t)} - \alpha\sum_{k=1}^{N} c_k\left(y_k^{(t)}(\tilde{\boldsymbol{z}})\right) - \alpha p\left(x_M^{(t)}(\tilde{\boldsymbol{z}}) - \tau^{(t)}\right)\right\}\right) \\ & \text{s.t.} & \\ & & x_j^{(t)}(\tilde{\boldsymbol{z}}) - x_i^{(t)}(\tilde{\boldsymbol{z}}) \geq \left(\tilde{z}_k - y_k^{(t)}(\tilde{\boldsymbol{z}})\right)^+ \quad \forall (i,j) \in \mathcal{E}, k = \Phi(i,j) \\ & & \boldsymbol{0} \leq \boldsymbol{y}^{(t)}(\tilde{\boldsymbol{z}}) \leq \boldsymbol{u}^{(t)} \\ & & \boldsymbol{x}^{(t)}(\tilde{\boldsymbol{z}}) \geq \boldsymbol{0} \\ & & x_i^{(t)} \in \mathcal{Y}(1, N, I_i^x) \quad \forall i \in [M] \\ & & y_k^{(t)} \in \mathcal{Y}(1, N, I_k^y) \quad \forall k \in [N] \\ & & \alpha \geq 0 \end{array} \tag{16}$$

to obtain new decision rules $\left(\boldsymbol{x}^{(t)}(\cdot), \boldsymbol{y}^{(t)}(\cdot)\right)$ for start times and crashing amounts, which are implemented in period $t+1$.

Step 3. Wait to receive the tth periodic project report.

Step 4. Obtain $\boldsymbol{f}^{(t)}$ from the report. Compute $\boldsymbol{g}^{(t)}$ from (15). Also, compute the matrices $\boldsymbol{D}_{\boldsymbol{f}}^{(t)}$ and $\boldsymbol{D}_{\boldsymbol{g}}^{(t)}$. If $\boldsymbol{f}^{(t)} = \mathbf{0}$, then stop. Otherwise, continue with Step 5.

Step 5. Update the family of uncertainty distributions using

$$\mathbb{F}^{(t+1)} = \left\{\mathbb{P} : \mathrm{E}_{\mathbb{P}}\left(\tilde{\boldsymbol{z}}\right) = \boldsymbol{D}_{\boldsymbol{f}}^{(t)}\boldsymbol{\mu}, \mathrm{E}_{\mathbb{P}}\left(\tilde{\boldsymbol{z}}\tilde{\boldsymbol{z}}'\right) = \boldsymbol{D}_{\boldsymbol{f}}^{(t)}\left(\boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}'\right)\boldsymbol{D}_{\boldsymbol{f}}^{(t)}, \mathbb{P}\left(\tilde{\boldsymbol{z}} \in \mathcal{W}^{(t+1)}\right) = 1\right\},$$

where

$$\mathcal{W}^{(t+1)} = \left\{\boldsymbol{\zeta} \in \Re^N : \zeta_i = \frac{z_i}{f_i}\mathbb{1}_{\{f_i>0\}}\ \forall i \in [N], \boldsymbol{z} \in \mathcal{W}\right\}. \tag{17}$$

Step 6. Update the vector of remaining crash limits using

$$\boldsymbol{u}^{(t+1)} = \boldsymbol{u}^{(t)} - \boldsymbol{D}_{\boldsymbol{g}}^{(t)}\boldsymbol{y}^{(t)}(\tilde{\boldsymbol{z}}). \tag{18}$$

Step 7. Update the remaining time and budget, respectively, using

$$\begin{aligned} \tau^{(t+1)} &= \tau - tT, \qquad \text{and} \\ B^{(t+1)} &= B - \sum_{k=1}^{N} c_k\left(y_k^{(t)}(\tilde{\boldsymbol{z}}) \cdot \left(1 - f_k^{(t)}\right)\right). \end{aligned} \tag{19}$$

Step 8. If $\boldsymbol{f}^{(\boldsymbol{t})} = 0$, then stop. Otherwise, set $t = t + 1$, and go to Step 2.

In Step 5, the family of uncertainty distributions $\mathbb{F}^{(t+1)}$ is updated using a heuristic that proportionately scales the distributional properties. In particular, when $\mathcal{W}$ has the simple but common structure of a box, $\mathcal{W} = \left\{\boldsymbol{z} \in \Re^N : \boldsymbol{z}_l \leq \boldsymbol{z}_u\right\}$ for some vectors of lower and upper limits $\boldsymbol{z}_l$ and $\boldsymbol{z}_u$, the support update simplifies to

$$\mathcal{W}^{(t+1)} = \left\{\boldsymbol{\zeta} \in \Re^N : \boldsymbol{D_f}^{(t)}\boldsymbol{z}_l \leq \boldsymbol{\zeta} \leq \boldsymbol{D_f}^{(t)}\boldsymbol{z}_u\right\}.$$

Notice that this is effectively an assumption that the uncertainty associated with an activity is uniform across the activity, which parallels a standard assumption in earned value analysis where resources for each activity are used at a uniform rate (Klastorin 2004).

Regarding consumption of the budget by crashing cost, we assume that crashing activity and the accrual of crashing cost occur at a constant rate within each period. This assumption is consistent with the situation where crashing involves assigning additional resources to an activity, with resources having constant usage and cost rates throughout the period (Harrison and Lock

2004). Hence, the amount of crashing used in the tth period can be written as $\boldsymbol{D_g}^{(t)}\boldsymbol{y}^{(t)}(\tilde{\boldsymbol{z}})$, which explains the formula used for updating $\boldsymbol{u}^{(t)}$ in Step 6.

We note two points regarding our rolling horizon model. First, the crashing decisions are, by construction, only functionally dependent on fully completed activities. Consequently, in each update period, the indices k where $g_k > 0$ are exactly those for which $y_k^{(t)}(\tilde{\boldsymbol{z}})$ is measurable, with respect to the filtration generated by the activity times that are known by period t. As a result, in (18) and (19), the crash limits and budget are always updated with deterministic quantities. Second, in the later update periods, if an activity k is fully complete, (17) gives us $\tilde{z}_k = 0$ almost surely. In this case, the crashing decision in period t for activity k is redundant, since the $\left(\tilde{z}_k - y_k^{(t)}(\tilde{\boldsymbol{z}})\right)^+$ term in (16) evaluates to identically zero. The redundant decision rules are retained in the model for ease of exposition.

For any given sample path comprising the realization of uncertain activities times, the rolling horizon approach leads to better project performance with respect to the probability of meeting the total project budget than model (4). Intuitively, the rolling horizon approach uses periodic information updates to improve the crashing decisions iteratively. In Section 5, we use a computational study to estimate the extent of this improvement.

5 Computational Study

In this section, we summarize the results of our computational study on the crashing strategies described above. A discussion of our data generation procedure and experimental design is followed by a summary of our computational results.

Our data set consists of 1,000 sets of activity times for each of 500 networks, which are randomly generated as follows. Using the guidelines of Hall and Posner (2001), (a) we generate a wide range of parameter specifications, (b) the data generated is representative of real world scenarios, and (c) the experimental design varies only the parameters that may affect the analysis. Also, we use the procedure described by Hall and Posner (2001) within each project network to generate precedence constraints that have an expected density, or order strength (OS), which meets a prespecified target and is also uniform across the network.

We use the random network generator *RanGen1* of Demeulemeester et al. (2003) to generate networks with a wide range of network topologies. The authors show that the networks generated span the full range of problem complexity with a prespecified OS. In our experiment, we generate 500 project instances that contain 10 activities each, with an OS of 0.5.

We then apply a series of transformations to each network to fit it into our modeling framework. First, we convert the generated networks from the AON convention to the AOA convention, using the technique of Cohen and Sadeh (2006). Second, to introduce stochasticity into the project networks, we use the activity times generated by *RanGen1* as mean activity times, and assume that each activity has support on the interval $[0.2t, 1.8t]$, where t represents the mean activity time of a given activity. We conduct two sets of tests, first where the marginal distribution of each activity is uniform, and second where the marginal distribution of each activity is $\beta(5,5)$ distributed (Kerzner 2009). The diagonal entries of the covariance matrix are therefore implicitly defined by the choice of distribution. For each marginal distribution, we test three subcases, (a) where the activity times are independent (INDEP), (b) where the activity times are randomly correlated with a diagonal covariance matrix on average (SYMCORR), and (c) where the activity times are randomly correlated with a nondiagonal covariance matrix on average (ASYMCORR).

Next, we choose problem parameters to ensure that our project instances are relevant to the project crashing problem. The crash cost rate for each activity is generated from a uniform distribution between 0 and 2, and the penalty cost rate of each project is set to the number of activities in the network. The justification for this is that we expect that, as the project network grows in size, the penalty cost rate should increase approximately linearly, to reflect the impact of a delay. To choose the project completion time τ, we make the PERT Assumptions A1–A3, and choose τ so that the project is delayed with probability 95%. We then choose B such that, according to the PERT assumptions, we have a 70% probability of project failure if no crashing is used. For simplicity, we assume that each activity can be completely crashed if desired.

For each project instance, we randomly generate 1,000 sets of activity times drawn from our assumed distribution. In the experiment with correlated activities, we use the standard Gaussian copula technique (Nelsen 1999) to generate the correlated activities. We compare various crashing strategies. First, we consider the NOCRASH strategy, which performs no crashing and saves all of the project budget to pay the penalty costs in the event of late project completion. Second, we consider the MINCRASH strategy that is described in Section 4.4. Third, we consider a benchmark strategy MCSIM, which computes the crashing decisions using model (5), based on the Utopian assumption of complete knowledge of the distribution of activity times. Next, we use our LDR and PWLDR decision rules to compute the crashing decisions. We record the results for the PWLDR decision rules under the bounds $Z^{(1)}_{PWLDR}$ and $Z^{(2)}_{PWLDR}$ separately as PWLDR$^{(1)}$ and PWLDR$^{(2)}$, respectively.

Finally, we compare the crashing decisions with the rolling horizon method using the LDR decision rule to solve model (16) with $T = 0.7\tau$. The choice of a relatively large period τ was made to keep the number of information update periods per network after crashing to be between two and three, for computational reasons. Unlike in the other methods, here we solve for individual crashing decisions on individual networks, hence having a large number of periods per network is computationally expensive.

Models for each strategy are implemented using ROME (Goh and Sim 2009), which is a MATLAB-based modeling language for robust optimization problems, and solved using CPLEX on a 64-bit workstation running a Linux operating system. The average time taken for computing the crashing decisions in a network instance is approximately 0.5 seconds for MINCRASH, 3.5 seconds for MCSIM, and 5.0 seconds for both the LDR and PWLDR strategies. Table 3 summarizes the average probability of meeting the overall project budget over the 500 × 1,000 = 500,000 randomly generated instances within each test, as well as the 95% confidence intervals based on a normal approximation, for the various crashing strategies. In the table, p represents the average project success probability, and $p \pm \epsilon$ represents the 95% confidence interval for the probability of project success. Also, for a given project network, we compute the expected budget overrun as a fraction of the project budget, using $L = \frac{1}{B}\mathrm{E}\left(\tilde{\Gamma} - B\right)^{+}$, where $\tilde{\Gamma}$ represents the total crashing and penalty cost associated with each strategy. We record $L \pm \epsilon$, the 95% confidence interval for each strategy, in Table 4.

The results for uniform marginal activity time distributions show that crashing by any strategy increases the average probability of project success and reduces the expected budget overrun in each of the six subcases. Even the benchmark PERT-based MINCRASH strategy substantially improves the success probability over NOCRASH. The MCSIM results are provided as a Utopian benchmark, since they assume knowledge of the activity time distribution. In practice, however, this assumption is unrealistic (Williams 2003). By contrast, our LDR and PWLDR strategies only use general, descriptive statistics of the activity times. Both these strategies greatly outperform the NOCRASH and MINCRASH strategies. Moreover, the PWLDR strategy also performs

Distribution	Strategy	INDEP		SYMCORR		ASYMCORR	
		$p\%$	$\epsilon\%$	$p\%$	$\epsilon\%$	$p\%$	$\epsilon\%$
Uniform	NOCRASH	13.27	2.97	12.70	2.92	39.00	4.28
	MINCRASH	50.55	4.38	50.57	4.38	63.03	4.23
	MCSIM	94.45	2.01	93.97	2.09	90.45	2.58
	LDR	92.26	2.34	90.75	2.54	91.94	2.39
	PWLDR$^{(1)}$	92.84	2.26	93.94	2.09	97.48	1.37
	PWLDR$^{(2)}$	96.22	1.67	97.62	1.34	97.77	1.30
	RH(LDR)	86.97	2.95	86.07	3.03	81.83	3.38
$\beta(5,5)$	NOCRASH	18.50	3.40	17.92	3.36	36.42	4.22
	MINCRASH	67.63	4.10	68.40	4.08	67.68	4.10
	MCSIM	96.53	1.61	96.60	1.59	93.32	2.19
	LDR	76.56	3.71	74.31	3.83	73.55	3.87
	PWLDR$^{(1)}$	96.62	1.58	98.89	0.92	98.64	1.01
	PWLDR$^{(2)}$	99.07	0.84	99.43	0.66	98.45	1.08
	RH(LDR)	86.41	3.00	85.01	3.13	82.78	3.31

Table 3: Probability of Meeting Overall Project Budget in 10-Activity Networks.

Distribution	Strategy	INDEP		SYMCORR		ASYMCORR	
		$L\%$	$\epsilon\%$	$L\%$	$\epsilon\%$	$L\%$	$\epsilon\%$
Uniform	NOCRASH	103.87	6.72	104.03	6.70	110.93	10.66
	MINCRASH	26.48	3.34	27.32	3.46	39.99	5.74
	MCSIM	1.20	0.59	1.31	0.61	3.57	1.27
	LDR	1.06	0.42	1.24	0.47	1.55	0.57
	PWLDR$^{(1)}$	0.98	0.42	0.76	0.38	0.38	0.28
	PWLDR$^{(2)}$	0.46	0.28	0.29	0.23	0.30	0.25
	RH(LDR)	4.38	1.31	4.71	1.37	8.21	1.96
$\beta(5,5)$	NOCRASH	93.06	6.88	92.56	6.89	107.63	10.79
	MINCRASH	17.91	3.16	18.09	3.30	35.84	6.18
	MCSIM	1.11	0.69	1.14	0.73	4.03	1.75
	LDR	6.69	1.42	7.24	1.49	7.19	1.44
	PWLDR$^{(1)}$	1.94	1.27	0.30	0.37	0.25	0.30
	PWLDR$^{(2)}$	0.18	0.25	0.13	0.24	0.38	0.40
	RH(LDR)	3.21	1.00	3.53	1.03	4.06	1.10

Table 4: Normalized Expected Budget Overrun in 10-Activity Networks.

competitively with the Utopian benchmark MCSIM in general, and outperforms it when the activity time covariance matrix is nondiagonal. The PWLDR$^{(2)}$ strategy has the best performance among the linear-based decision rules, exceeding even the Utopian MCSIM benchmark in all three subcases.

In the presence of marginally β-distributed activity times, the LDR strategy shows the sharpest drop in performance. Since the LDR strategy ignores variance information, it provides over-conservative results when applied to activity times that have a less flat distribution. Chen et al. (2008) report similar results. Nonetheless, the LDR strategy still greatly outperforms the NOCRASH and MINCRASH strategies. Both the PWLDR strategies outperform all others when activity times are correlated, and the PWLDR$^{(2)}$ strategy provides over 98% probability of meeting the overall project budget, and less than 0.5% expected budget overrun. Moreover, the PWLDR strategies typically improve in performance in the presence of either form of correlation, since they explicitly incorporate covariance information.

From Tables 3 and 4, we observe that the rolling horizon strategy improves on the LDR strategy for $\beta(5,5)$-distributed activity times, while it performs less well for uniformly-distributed activity times. Since the rolling horizon strategy uses more information about the uncertainties than the LDR strategy, we would naturally expect it to improve on the LDR strategy in all instances. However, since we use approximations in both cases to solve models (4) and (16), the RH(LDR) strategy does not dominate the LDR strategy. This is especially true where the static LDR strategy already performs relatively well, as with the marginally uniform activity times.

We repeat our computational tests for a set of larger networks with 20 activities each. Due to the longer computation times required to solve each network, we use a smaller test set of 50 randomly generated networks, and randomly generate 1,000 sets of activity times for each network, with the exception of the RH(LDR) strategy, where due to longer computation times, we use 50 sets of activity times. We summarize our results in Tables 5 and 6.

Tables 5 and 6 show that our results for 20-activity networks are similar to those for 10-activity networks, but even stronger. The LDR and PWLDR strategies typically outperform even the Utopian MCSIM strategy, and greatly outperform the PERT-based strategies MINCRASH and NOCRASH. Moreover, a comparison of results for the 10-activity and 20-activity networks shows that project success probabilities and expected budget overruns for our LDR and PWLDR strategies improve substantially in the larger networks. The most significant im-

Distribution	Strategy	INDEP		SYMCORR		ASYMCORR	
		$p\%$	$\epsilon\%$	$p\%$	$\epsilon\%$	$p\%$	$\epsilon\%$
Uniform	NOCRASH	8.40	7.69	8.19	7.60	40.23	13.59
	MINCRASH	44.48	13.77	43.79	13.75	61.33	13.50
	MCSIM	99.36	2.21	99.38	2.18	96.91	4.80
	LDR	100.00	0.00	100.00	0.00	100.00	0.00
	PWLDR$^{(1)}$	100.00	0.00	100.00	0.00	100.00	0.00
	PWLDR$^{(2)}$	100.00	0.00	100.00	0.00	100.00	0.00
	RH(LDR)	98.72	3.12	96.68	4.97	97.60	4.24
$\beta(5,5)$	NOCRASH	15.59	10.05	16.22	10.22	38.62	13.50
	MINCRASH	63.22	13.37	64.44	13.27	66.01	13.13
	MCSIM	99.08	2.64	99.08	2.65	96.70	4.95
	LDR	97.20	4.58	95.65	5.65	96.11	5.36
	PWLDR$^{(1)}$	99.53	1.90	99.97	0.46	99.81	1.19
	PWLDR$^{(2)}$	99.89	0.90	99.97	0.46	99.68	1.56
	RH(LDR)	97.76	4.10	99.88	0.96	99.56	1.83

Table 5: Probability of Meeting Overall Project Budget in 20-Activity Networks.

Distribution	Strategy	INDEP		SYMCORR		ASYMCORR	
		$L\%$	$\epsilon\%$	$L\%$	$\epsilon\%$	$L\%$	$\epsilon\%$
Uniform	NOCRASH	116.16	20.98	116.32	20.86	125.08	38.71
	MINCRASH	29.95	10.99	32.33	11.72	50.34	22.25
	MCSIM	0.07	0.32	0.07	0.36	0.58	1.19
	LDR	0.00	0.00	0.00	0.00	0.00	0.00
	PWLDR$^{(1)}$	0.00	0.00	0.00	0.00	0.00	0.00
	PWLDR$^{(2)}$	0.00	0.00	0.00	0.00	0.00	0.00
	RH(LDR)	0.20	0.74	1.06	2.02	0.68	1.51
$\beta(5,5)$	NOCRASH	98.86	21.77	97.83	22.24	120.23	39.50
	MINCRASH	20.61	10.53	20.11	10.75	45.89	24.20
	MCSIM	0.22	0.86	0.26	1.02	1.85	3.65
	LDR	0.14	0.29	0.38	0.55	0.57	0.88
	PWLDR$^{(1)}$	0.01	0.07	0.00	0.00	0.02	0.12
	PWLDR$^{(2)}$	0.00	0.02	0.00	0.00	0.03	0.16
	RH(LDR)	0.65	1.26	0.01	0.15	0.00	0.02

Table 6: Normalized Expected Budget Overrun in 20-Activity Networks.

provement occurs for the LDR strategy where the activity times are $\beta(5,5)$-distributed. Since the LDR and PWLDR strategies explicitly take advantage of complex precedence structures, we expect that their performance improves as a function of network size.

In summary, our results show that both the LDR and PWLDR crashing strategies improve substantially over benchmark PERT-based strategies. Our strongest recommendation is for the use of the PWLDR strategy with the PWLDR$^{(2)}$ bound, which provides very close to 100% probability of overall project success and very close to zero expected budget overrun across a wide variety of data sets. Where activity time variances are unavailable or estimates are unreliable, in which case we cannot use the PWLDR crashing strategy, the RH(LDR) strategy is a valuable alternative.

6 Concluding Remarks

In this paper, we study the problem of project planning with crashing, under uncertainty. We consider a satisficing objective that models penalties for completing a project after a deadline as well as the total cost of crashing activities, relative to an overall project budget. Two solution procedures are proposed. The first procedure develops a linear decision rule and two piecewise-linear decision rules that are computed at the project planning stage and that use updated information about activity completion times during project execution. In addition, we describe a rolling horizon procedure that takes advantage of periodic information updates during the execution of activities, to determine new decision rules.

Our results provide several insights that project managers will find useful. Based on our computational results, our PWLDR strategies recommend crashing decisions that deliver 20-activity projects on time and on budget with over 99% probability, compared to less than 45% probability using traditional PERT-based strategies for the same benchmark problems. Moreover, our strategies provide expected budget overruns that average less than 0.1%, compared with over 25% using PERT-based strategies. These results are achieved without knowledge of the specific activity time distribution. The simpler decision strategy, LDR, also provides results that substantially outperform traditional strategies. Additionally, the LDR strategy can be implemented in a rolling horizon format for further improvement, which is especially advantageous when activity time variances are small. Importantly, the performance of our crashing strategies appears to improve with project size. The issue of activity time correlation that severely complicates project scheduling using traditional strategies is easily handled, which provides an incentive to project managers to document correlation where possible. Finally, while PERT is widely used in practice because of its ease of implementation, the availability of ROME software (and Sim 2009b) makes the implementation of our methodology similarly easy.

Two important special cases of the model are relevant. First, the crash cost budget may be viewed deterministically if there is no flexibility in the budget allocated to a project. In this case, the project management problem is to make crashing decisions that maximize the probability of completing the project by the given time τ, subject to the deterministic budget constraint. Second, the project completion time may be viewed deterministically. This is relevant where the client imposes a firm deadline on the project delivery time, beyond which the project contract is void. In this case, the project management problem is to make crashing decisions that maximize the probability of completing the project within the given budget B, subject to the deterministic completion time constraint. Both these applications can be modeled and solved using the methodology described above in our paper. However, the special structure of these models may enable more efficient solution procedures and more accurate approximations.

Several extensions of the general problem considered in this paper are also relevant for future research. The first extension is the modeling of discrete crashing resources, for example the outsourcing of an entire activity, within a robust optimization framework. Related to this is the modeling of discrete crash cost functions for activities, for example when adding an extra production shift in order to meet a deadline on project completion time. Another practical generalization is the modeling of discrete penalty functions for project completion time, such as sometimes arise in large public construction projects. Unfortunately, many projects suffer from a loss of resources during their execution stage, due to competition from other projects. Hence, a further useful generalization is the modeling of uncertainty about the availability of resources for crashing. Still more generally, the study of multiple projects that share crashing resources would be a valuable contribution. In conclusion, we hope that our work will encourage the development of effective robust optimization approaches to these important project management problems.

Acknowledgments

The work of the first author is supported by the Kaneko/Lainovic International Fellowship. The work of the second author is supported by the Summer Fellowship Program and the Center for Operational Excellence of the Fisher College of Business, The Ohio State University. The work of the third author is supported by the Singapore - MIT Alliance, and by National University of Singapore academic research grant R-314-000-068-122. Helpful comments on an earlier draft were provided by Zhuoyu Long and Jin Qi (National University of Singapore).

References

Alexander, G.J., A.M. Baptista. 2004. A comparison of VaR and CVaR constraints on portfolio selection with the mean-variance model. *Management Science* **50** 1261–1273.

Barbour, A.D., J.L. Jensen. 1989. Local and tail approximations near the Poisson limit. *Scandinavian Journal of Statistics* **16** 75–87.

Ben-Tal, A., A. Goryashko, E. Guslitzer, A. Nemirovski. 2004. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming* **99** 351–376.

Ben-Tal, A., A. Nemirovski. 1998. Robust convex optimization. *Mathematics of Operations Research* **23** 769–805.

Bertsimas, D., M. Sim. 2004. The price of robustness. *Operations Research* **52** 35–53.

Black, K. 2008. *Business Statistics: Contemporary Decision Making*, 5th edition. Wiley, Hoboken, NJ.

Bowman, R.A. 1994. Stochastic gradient-based time-cost tradeoffs in PERT networks using simulation. *Annals of Operations Research* **53** 533-551.

Brown, D.B., M. Sim. 2009. Satisficing measures for analysis of risky positions. *Management Science* **55** 71–84.

Brown, D.B., E. De Giorgi, M. Sim. 2009. A satisficing alternative to prospect theory. *Working Paper*, Fuqua School of Business, Duke University, Durham, NC, November.

Chaos Report. 2006. The Standish Group, Boston, MA.

Chen, W., M. Sim. 2009. Goal driven optimization. *Operations Research* **57** 342–357.

Chen, X., M. Sim, P. Sun, J. Zhang. 2008. A linear decision-based approximation approach to stochastic programming. *Operations Research* **56** 344–357.

Cohen, I., B. Golany, A. Shtub. 2007. The stochastic time-cost tradeoff problem: A robust optimization approach. *Networks* **49** 175–188.

Cohen Y., A. Sadeh. 2006. A new approach for constructing and generating AOA networks. *Journal of Engineering, Computing and Architecture* **1** 1–15.

Demeulemeester, E., M. Vanhoucke, W. Herroelen. 2003. A random network generator for activity-on-the-node networks. *Journal of Scheduling* **6** 13–34.

El-Ghaoui, L., F. Oustry, H. Lebret. 1998. Robust solutions to uncertain semidefinite programs. *SIAM Journal on Optimization* **9** 33–52.

Ellsberg, D. 1961. Risk, ambiguity, and the Savage axioms. *Quarterly Journal of Economics* **75** 643–669.

Fleming, Q., J. Koppelman. 2005. *Earned Value Project Management*, 3rd edition. Project Management Institute, Newtown Square, PA.

Goh, J., M. Sim. 2009. Robust optimization made easy with ROME. *Working Paper*, NUS Business School, National University of Singapore.

Goh, J., M. Sim. 2010. Distributionally robust optimization and its tractable approximations. *Operations Research*, to appear.

Greaver, M.F., II. 1999. *Strategic Outsourcing: A Structured Approach to Outsourcing Decisions and Initiatives*, 1st edition. AMACOM, New York City, NY.

Gutjahr, W.J., C. Strauss, E. Wagner. 2000. A stochastic branch-and-bound approach to activity crashing in project management. *INFORMS Journal on Computing* **12** 125–135.

Hall, N.G., M.E. Posner. 2001. Generating experimental data for computational testing with machine scheduling applications. *Operations Research* **49** 854–865.

Harrison, F.L., D. Lock. 2004. *Advanced Project Management: A Structured Approach*, 4th edition. Gower Publishing, Aldershot, U.K.

Herroelen, W., R. Leus. 2005. Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research* **165** 289–306.

Hughes, M.W. 1986. Why projects fail: The effects of ignoring the obvious. *Industrial Engineering*, April, 14–18.

Kelley, J.E., Jr., M.R. Walker. 1959. Critical-path planning and scheduling. *Proceedings, AFIPS Joint Conferences*, 160-173.

Kerzner, H. 2009. *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*, 10th edition. Wiley, Hoboken, NJ.

Klastorin, T.D. 2004. *Project Management: Tools and Trade-Offs*, 1st edition. Wiley, Hoboken, NJ.

Kwak, H.K., L. Ingall. 2007. Exploring Monte Carlo simulation applications for project management. *Risk Management* **9** 44–57.

Michael, J.R., W.R. Schucany. 2002. The mixture approach for simulating bivariate distributions with specified correlations. *The American Statistician* **56** 48–54.

Might, R.J., W.A. Fischer. 1985. The role of structural factors in determining project management success. *IEEE Transactions on Engineering Management* **32** 71–77.

Natarajan, K., M. Sim, J. Uichanco. 2010. Tractable robust expected utility and risk models for portfolio optimization. *Mathematical Finance*, to appear.

Nelsen, R.B. 1999. *An Introduction to Copulas*. Springer, New York City, NY.

Nesterov, Y., A. Nemirovski. 1994. *Interior-point polynomial algorithms in convex programming.* SIAM Studies in Applied Mathematics **13**. SIAM, Philadelphia, PA.

1000ventures.com. 2009. Change management. Available at: http://1000ventures.com/business_guide/crosscuttings/change_management.html

Pells, D.L. 2008. Seven good reasons for the rapid growth of project management in IT and why that trend will continue. *PM World Today* **X**, June, 1–14.

Rockafellar, R.T., S. Uryasev. 2000. Optimization of conditional value-at-risk. *The Journal of Risk* **2** 21–41.

Schonberger, R.J. 1981. Why projects are "always" late: A rationale based on manual simulation of a PERT/CPM network. *Interfaces* **11** 66–70.

Siemens, N. 1971. A simple CPM time-cost tradeoff algorithm. *Management Science* **17** 354–363.

Soyster, A.L. 1973. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research* **21** 1154–1157.

U.S. Department of Defense. 1962. PERT cost systems design. DoD and NASA Guide.

U.S. Navy. 1958. PERT summary report, phase I. Defense Technical Information Center, AD-735 902, Special Projects Office, Bureau of Naval Weapons.

Value Based Management.net. 2009. Product life cycle - Industry maturity stages. Available at: http://www.valuebasedmanagement.net/methods_product_life_cycle.html

Williams, T. 2003. The contribution of mathematical modeling to the practice of project management. *IMA Journal of Management Mathematics* **14** 3–30.