Submitted to
manuscript (Please, provide the manuscript number!)

Robust Capacity Planning for Project Management

Antonio J. Conejo
Department of Electrical & Computer Engineering, The Ohio State University, Columbus, OH 43210, conejonavarro.1@osu.edu

Nicholas G. Hall
Department of Management Sciences, The Ohio State University, Columbus, OH 43210, hall.33@osu.edu

Daniel Zhuoyu Long
Department of Systems Engineering & Engineering Management, The Chinese University of Hong Kong, Shatin, Hong Kong, corresponding author: zylong@se.cuhk.edu.hk

Runhao Zhang
Department of Systems Engineering & Engineering Management, The Chinese University of Hong Kong, Shatin, Hong Kong, rhzhang@se.cuhk.edu.hk

We consider a significant problem that arises in the planning of many projects. Project companies often use outsourced providers which require capacity reservation that must be contracted before task durations are realized. We model these decisions for a company which, given partially characterized distributional information, assumes the worst case distribution for task durations. Once task durations are realized, the project company makes decisions about fast tracking and outsourced crashing, to minimize the total capacity reservation, fast tracking, crashing, and makespan penalty costs. We model the company's objective using the target-based measure of minimizing an underperformance riskiness index. We allow for correlation in task performance, and for piecewise linear costs of crashing and makespan penalties. Computationally efficient optimal solution of the discrete, nonlinear model is possible for practical size projects. We compare the performance of our model against the best available benchmarks from the robust optimization literature, and show that it provides lower risk and greater robustness to distributional information. Our work thus enables more effective risk minimization in projects, and provides insights about how to make more robust capacity reservation decisions.

Key words: project management, capacity reservation, adjustable distributionally robust optimization, optimal algorithm

1. Introduction

The global economic value of projects exceeds $12 trillion annually, and represents over 20% of the world's economic activity (Project Management Institute 2008). Moreover, in recent years, the range of applications that are managed as projects has expanded greatly, to include for example IT, change management, new product and service development, pharmaceutical development, and research and development. In response to the different characteristics of modern project management applications, new planning methodologies have been developed for project management. Wysocki (2014) contrasts and compares traditional and modern project planning methods. Hall (2012, 2015, 2016) describes a variety of high level research questions for project management.

This work studies a financially significant problem that arises at the planning stage of many projects. In order to respond to delays in task execution, crashing i.e. expediting, of tasks is needed. However, resources needed for crashing are outsourced, and their capacity must be reserved in advance. This is especially typical for high technology, pharmaceutcial, and other specialized projects, due to long lead times in product certification and quality assurance. Once task durations are realized, the project company uses the previously reserved capacity to respond to any delays. For this problem, we develop and solve an adjustable distributionally robust optimization (ADRO) model. This two-stage model considers capacity reservation decisions, the possibility of delays to the tasks, and the use of previously reserved capacity to crash them. Decisions that are recommended by optimal solution of our model provide robust performance against task time uncertainty even under distributional ambiguity for task time distributions.

Uncertainty in task time durations arises from three main sources in projects. The first is task time estimation errors due to, for example, limited data, poor estimation techniques, or deliberate overestimation for self-protection (Okoro 2015). The second is variation in task performance due to, for example, miscommunication, unavailability of resources, unanticipated technical difficulty, scope changes by project owners, or quality problems leading to rework (Soni and Acharya 2015). The third is behavioral issues, such as Parkinson's Law (Parkinson 1955, 1958), that affect task time performance.

Outsourcing is the procurement of products, services or capacity from external subcontractors. Organizations which run projects increasingly use outsourcing (Prahalad and Hamel 1990, Quinn and Hilmer 1994, Tully 1994), for several reasons. First, those organizations do not have the resources needed to support all the tasks in their projects, and it would be too expensive to maintain those resources in-house. Second, in many projects, it is essential that specialized tasks are performed by outside subcontractors. An example is creative or highly technical tasks to support new product development (Pisano and Verganti 2008). The focus of our work is on outsourced capacity for crashing tasks. The use of outsourced capacity requires reservation, which has generated a substantial literature. See, for example, Cachon and Lariviere (2001), Serel et al. (2001), Jin and Wu (2007), Boulaksil et al. (2011), and Silver and Jain (1994).

Further examples exist within options contracts. As discussed in Cachon and Terwiesch (2006, p. 359), options contracts are widely used in situations of uncertainty. Under an options contract, a buyer (in the present case a project company) pays one price to purchase an option for capacity, and another price to exercise the purchased option. The purchase of an option for capacity reservation is thus used as a hedge against unexpected delays in executing a project.

To illustrate the practicality of capacity reservation contracts that support projects, we reference two examples from the business literature. In Securities and Exchange Commission (2007), p.109, an unspecified capacity reservation fee is used to require Cobra Biologics Limited to "reserve capacity and resources at its Facility". In Securities and Exchange Commission (2014), p.20, Variation Biotechnologies, Inc. agrees to pay Paragon Bioservices, Inc.$180,000, "in order to ensure sufficient development and manufacturing capacity / personnel resources ...". These two contracts are taken from pharmaceutical development projects. In both cases, the pharmaceutical development company reserves capacity with an outsourced provider of product development and manufacturing services. These examples match the characteristics of our model, in that payment is made for capacity that eventually may or may not be needed. Also, reservation of capacity provides the pharmaceutical development company with additional options.

Classical project planning methodologies such as PERT (DOD and NASA 1962) approximate a probability distribution for task time distributions, based on optimistic, pessimistic and most likely estimates. As a result, much of the project scheduling literature with uncertain task times assumes full knowledge of the probability distributions of task durations. However, consistent industry and academic evidence (for example, Adler et al. 1995, Leach 1999, van Dorp and Duffey 1999, Pender 2001, Williams 2003, Herroelen and Leus 2005) questions this assumption. Therefore, we assume only partially characterized task time distributions. Distributional ambiguity of this type suggests the use of *robust optimization*, which enables optimization under worst-case uncertainties (e.g., Soyster 1973, Ben-Tal and Nemirovski 1998, El Ghaoui et al. 1998, Bertsimas and Sim 2004). However, simultaneous realization of all worst-case uncertainties is highly unlikely and results in overly conservative decision making, which motivates the use of *distributionally robust optimization*, as developed by Delage and Ye (2010), Goh and Sim (2010), Xu and Mannor (2012), and Wiesemann et al. (2014) for related applications.

Robust optimization has been applied to project management, to model crashing decisions during project execution (Goh and Hall 2013) and project portfolio selection (Hall et al. 2015). However, the problem studied here involves decision making both before and after uncertainties have been realized. Robust optimization problems within such multi-stage settings are called *adjustable robust optimization* (ARO) and are computationally intractable in general (Shapiro and Nemirovski 2005). Ben-Tal et al. (2004) propose the use of affine decision rules. For some special cases, this approach provides either optimal decisions (Bertsimas et al. 2010, Bertsimas and Goyal 2012, Iancu et al. 2013), or provably close to optimal decisions (Bertsimas and Goyal 2012).

The capacity reservation decision conisdered here is part of a planning, rather than an execution, process. Consequently, achieving fast computation time is not the highest priority. A similar perspective is adopted by several related works. These works form aother stream of literature on ARO, which designs algorithm for problems with reasonable size. In particular, Bertsimas et al. (2013) describe an ARO model for energy unit commitment, using cuts generated from an inner minimization problem and a budget of uncertainty. Ruiz and Conejo (2015) study an energy transmission problem using ARO. Zeng and Zhao (2013) develop a column-and-constraint generation algorithm for two-stage robust optimization problems; this algorithm dynamically generates constraints in the primal space of the problem. These works use only scenario information about uncertain factors. However, we also incorporate frequency information, which enables a more general characterization of uncertainties and more extensive use of historical data. Our uncertainty characterization generalizes typical information, such as mean, variance, and absolute deviation.

The sequence of decisions and events in the project planning problem which we consider is as follows. First, the project company makes capacity reservation decisions of two types. For routine task work, such as standard business processes, providers of outsourcing services offer blocks of capacity with announced prices, and the project company can reserve them. For technical components or specialized requirements of tasks, a request for a fixed amount of customized capacity is initiated by the project company. A task may require outsourced capacity of both types. Following capacity reservation decisions, we estimate the downside risk associated with task delays, which may cause penalties due to late delivery of the project. The distribution of uncertain delays is controlled by a set of constraints that model distributional robustness. The project company can respond to delays in two ways. A typically less expensive response, but limited in range, is fast tracking (Passionate Project Management 2011), i.e. concurrent performance of tasks that are formally required to be performed in sequence. A typically more expensive option is crashing the tasks (Passionate Project Management 2011), however this requires the use of previously reserved capacity. Therefore, the costs which we consider include capacity reservation, fast tracking and crashing of tasks, and penalties for later makespan completion.

Risk aversion is important in many projects, since dysfunctional projects tend to draw resources away from others which compromises their performance (Cooper et al. 2000). There may also be a loss of future business as a result of an unsuccessful project. Performance measures for evaluating downside risk include expected utility (von Neumann and Morgenstern 1944), and coherent risk measures including CVaR (Artzner et al. 1999, Rockafellar and Uryasev 2000, 2002). However, both these approaches require the specification of a risk aversion parameter. In general, this parameter is abstract and difficult for a decision maker to calibrate. This recommends the use of a target-oriented approach (Simon 1955). A target-oriented decision criterion, based on a set of axioms, is proposed by Brown and Sim (2009) and applied to project management (Goh and Hall 2013, Hall et al. 2015). However, two additional considerations complicate our work. The first is that we are considering a two-stage optimization problems that requires binary decisions, which imposes considerable computational challenges. The second is that the costs of project delay are frequently nonlinear due to penalty clauses in contracts, for example in local government construction projects. For these reasons, we use a target-oriented decision criterion based on general piecewise linear utility, which balances the practical aspects of project costs against the issue of computational solvability. With this approach, our work addresses the problem of achieving robust performance from high uncertainty projects under limited availability of information about the distributions of task durations.

The main contributions of our work are as follows:

1. We model the capacity reservation problem, including the practical features of fast tracking, crashing, and nonlinear completion time cost.

2. Our information set for task durations is general enough to model the practical issues of correlation and disperse information.

3. We develop a target-based decision making framework where the optimal solution provides minimum risk to the predetermined benchmark cost.

4. We develop a column and constraint generation algorithm, where each iteration requires solution of a bilinear problem, which we reformulate as a mixed integer program with special structure and solve efficiently.

5. Our computational study shows that our algorithm can solve practical size projects to optimality.

6. Our results outperform those found by the best available benchmark procedures from robust optimization.

7. We provide project managers with an improved planning tool for effective risk minimization in projects with high uncertainty, and guidance towards achieving more effective use of outsourced capacity.

This paper is organized as follows. Section 2 provides our notation, formally describes the problem studied, and discusses how we model uncertainty and risk. In Section 3, we develop an algorithm for finding optimal solutions to the problem. Section 4 demonstrates the computational efficiency of our algorithm, and compares its performance against several benchmark approaches. Finally, Section 5 describes insights for project managers, and identifies suggestions for future research. All proofs appear in appendices.

2. Model

In Section 2.1, we provide our notation and formally describe the problem we consider. In Section 2.2, we describe our modeling of the uncertainties in the problem and how to evaluate the riskiness of the uncertain cost.

2.1. Problem description and motivation

We use bold characters, e.g., $\boldsymbol{x}$ and $\boldsymbol{A}$, to represent vectors and matrices, respectively. Given any vector $\boldsymbol{x}$, we let x_i denote its ith element. Random variables are denoted using the tilde symbol, for example $\tilde{z}$, where z is the corresponding realization. The inequality between two random variables is statewise, i.e., $\tilde{w}_i \geq \tilde{w}_j$ implies that the probability that $(\tilde{w}_i \geq \tilde{w}_j)$ is 1. We use brackets to represent a set of running indices, e.g. $[N] = \{1, 2, \dots, N\}$.

We consider a project that consists of a collection of N individual tasks with precedence relationships between them. We let $n_i \rightarrow n_j$ denote that task n_i must be completed before task n_j starts. All decisions about capacity reservation and about how to use fast tracking and crashing to respond to uncertainties in task durations are made at the planning stage, where necessary based on assumptions about future realizations of those uncertainties. The sequence of decisions and events is as follows.

1. The project manager needs to decide how much of various types of capacity to reserve.

2. Uncertain task durations are assumed to be realized.

3. The project manager responds by fast tracking and/or utilizing the previously reserved capacity to crash the project, in order to minimize the total cost of capacity reservation, fast tracking, crashing, and makespan penalty.

Two types of capacities are considered. The first type is a general resource, representing capacity to perform standard tasks, including general administrative support. This type of capacity is offered in a given quantity-price pair list $(\boldsymbol{a}, \boldsymbol{u}_x) \in \Re^{2\times O}$, where O is the number of quantity-price pair on the list. If the jth available capacity package is chosen, a_j units of the general resource are reserved and the reservation cost u_{xj} is paid, for $j \in [O]$. We let $\boldsymbol{x} \in \{0,1\}^O$ denote the capacity decision for the general resource, where $x_j = 1$ if the jth package is reserved, and $x_j = 0$ otherwise. Hence, the total cost of general capacity reserved is $\boldsymbol{u}_x^T \boldsymbol{x}$.

The second type of capacity represents a customized resource for each individual task that is reserved following a solicitation by the project company. Examples of such resources include contract manufacturing and specialized engineering or chemical services. Since the details of the task are known, the solicitation is for a specific amount of capacity. Hence, the reservation decision is modeled as binary. Let $\boldsymbol{y} \in \{0,1\}^N$ and $\boldsymbol{u}_y \in \Re^N$ represent the reservation decision and reservation cost, respectively, for the customized capacity. We let $y_n = 1$ mean that a decision has been made to reserve customized capacity for task n, and $y_n = 0$ otherwise. Hence, the total cost of customized capacity reserved is $\boldsymbol{u}_y^T \boldsymbol{y}$.

We denote the uncertain delays for the tasks by $\tilde{\boldsymbol{z}}$, which realizes to $\boldsymbol{z}$ after the capacity reservation decision $(\boldsymbol{x}, \boldsymbol{y})$. Then, the project manager needs to make fast tracking and crashing decisions, with related costs. The first cost component is the cost incurred by fast tracking, which is the concurrent processing of tasks that have a formal precedence relationship. In principle, fast tracking incurs no increase in direct project costs; however, there is risk associated with doing more tasks concurrently, which implies greater management supervision and cost (Mochal 2006). Since such costs are not defined contractually, we model them as linear in the amount of fast tracking.

The second cost component is the crashing cost. Crashing costs are typically considered to be convex, because cheaper crashing options are exhausted before more expensive ones are initiated; a reasonable approximation is given by a piecewise linear time-cost tradeoff function (Vrat and Kriengkrairut 1986). Similar comments apply to penalties, or equivalently lost incentives, for delays in project makespan. These can also be modeled as piecewise linear and convex (Pinedo, 2009, p. 70). The objective is to minimize the total of capacity reservation cost and the three operational costs.

We assume that fast tracking is only constrained by an exogenous upper limit for any pair of tasks linked by a precedence constraint. However, the crashing decision of each task is constrained by both an exogenous upper limit and the earlier capacity reservation decisions. We assume that, for any task n, $n \in [N]$, crashing of w_n units requires $\eta_n w_n$ units of the task's customized capacity and also $\theta_n w_n$ units of general capacity. We let $\eta_n = 0$ if task n does not require any customized capacity, and $\theta_n = 0$ if task n does not require any general capacity.

We list the decision variables and parameters of the operational problem as follows.

Decision variables:

c_n = completion time of task n, $n \in [N]$

$w^F_{n_1 n_2}$ = concurrent processing of tasks n_1 and n_2 due to fast tracking, $n_1, n_2 \in [N]$, $n_1 \to n_2$

w^C_{jn} = usage of the jth linear cost interval of crashing for task n, $j \in [Q_n], n \in [N]$

v_j = usage of the jth linear cost interval of project makespan, $j \in [Q]$.

Here Q_n, $n \in [N]$, is the number of linear segments in the piecewise linear mapping from the crashing of task n to the associated cost. Similarly, Q is the number of linear segments of the cost function for makespan.

Parameters:

b_n = standard duration of task n without random delay or crashing, for $n \in [N]$

$\tilde{z}_n$ = uncertain duration of task n without fast tracking and crashing, for $n \in [N]$

z_n = realization of $\tilde{z}_n$, $n \in [N]$

$g^F_{n_1 n_2}$ = linear cost of fast tracking of task n_2 with task n_1, for $n_1, n_2 \in [N],\ n_1 \to n_2$

$\gamma_{n_1 n_2}$ = upper limit on concurrent processing of tasks n_1 and n_2, for $n_1, n_2 \in [N],\ n_1 \to n_2$

g^C_{jn} = linear cost of crashing task n in the jth linear cost interval, for $j \in [Q_n], n \in [N]$, where $g^C_{1n} < \cdots < g^C_{Q_n n}$ for $n \in [N]$

e^C_{jn} = upper limit on the jth linear cost interval for crash cost of task n, for $j \in [Q_n], n \in [N]$

η_n = a binary indicator of whether a customized capacity is needed to crash task n, for $n \in [N]$

θ_n = a usage rate such that general capacity $\theta_n w_n$ is needed to crash task n by w_n, for $n \in [N]$

d_j = linear cost of project completion time penalty in the jth linear cost interval, for $j \in [Q]$, where $d_1 < \cdots < d_Q$

l_j = upper limit on the jth linear cost interval for completion time penalty, for $j \in [Q]$.

Let $\boldsymbol{w}_F = (\boldsymbol{w}^F_1, \ldots, \boldsymbol{w}^F_N)$ and $\boldsymbol{w}^F_{n_2} = [w^F_{n_1 n_2}]_{n_1 \to n_2}, n_2 \in [N]$. We define $\boldsymbol{g}_F$ in the same way. Similarly, we define $\boldsymbol{w}_C = (\boldsymbol{w}^C_1, \ldots, \boldsymbol{w}^C_N)$, and $\boldsymbol{w}^C_n = (w^C_{1n}, \ldots, w^C_{Q_n n}), n \in [N]$. Also, $\boldsymbol{g}_C$ is defined similarly.

Given capacity reservation decisions $\boldsymbol{x}$ and $\boldsymbol{y}$, and uncertainty realizations $\boldsymbol{z}$, the project management planning problem we consider is

$$\begin{aligned} \min \quad & \boldsymbol{g}_F^T \boldsymbol{w}_F + \boldsymbol{g}_C^T \boldsymbol{w}_C + \boldsymbol{d}^T \boldsymbol{v} \\ \text{s.t.} \quad & (\boldsymbol{w}_F, \boldsymbol{w}_C, \boldsymbol{v}, \boldsymbol{c}) \in \mathcal{S}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}). \end{aligned} . \tag{1}$$

Here, $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ is the set of feasible second stage decisions for $(\boldsymbol{w}_F, \boldsymbol{w}_C, \boldsymbol{v}, \boldsymbol{c})$, where the first stage decision is $(\boldsymbol{x}, \boldsymbol{y})$ and the uncertainties realize as $\boldsymbol{z}$, as defined by the following constraints.

$$c_{n_2} - c_{n_1} + w^F_{n_1 n_2} + \sum_{j=1}^{Q_{n_2}} w^C_{j n_2} \geq b_{n_2} + z_{n_2}, \quad n_1, n_2 \in [N], \quad n_1 \to n_2 \tag{2}$$

$$\sum_{j=1}^{Q} v_j - c_n \geq 0, \quad n \in [N] \tag{3}$$

$$-\sum_{n=1}^{N} \theta_n \sum_{j=1}^{Q_n} w^C_{jn} \geq -\boldsymbol{a}^T \boldsymbol{x} \tag{4}$$

$$-\sum_{j=1}^{Q_n} w^C_{jn} \geq -e^C_{Q_n n} y_n, \quad n \in [N] \text{ if } \eta_n \neq 0 \tag{5}$$

$$-w^F_{n_1 n_2} \geq -\gamma_{n_1 n_2}, \quad n_1, n_2 \in [N],\ n_1 \to n_2 \tag{6}$$

$$-w_{jn}^C \geq -e_{jn}^C, \quad j \in [Q_n], \quad n \in [N] \tag{7}$$

$$-v_j \geq -l_j, \quad j \in [Q] \tag{8}$$

$$c_n \geq 0, \quad n \in [N]$$

$$w_{n_1 n_2}^F \geq 0, \quad n_1, n_2 \in [N], \quad n_1 \to n_2$$

$$w_{jn}^C \geq 0, \quad j \in [Q_n], \quad n \in [N]$$

$$v_j \geq 0, \quad j \in [Q].$$

The objective in (1) minimizes the total of fast tracking cost, crashing cost and project makespan penalty. Constraints (2) in $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ enforce precedence requirements between tasks, allowing for random durations, fast tracking and crashing. Constraints (3) ensure that all tasks finish by the project completion time, $\sum_{j=1}^{Q} v_j$. Constraint (4) ensures that the total general capacity used for crashing is no greater than that reserved. Constraints (5) require that any customized capacity used to crash task n has been reserved, as required if $\eta_n \neq 0$. Constraints (6) ensure that the fast tracking between any pair of jobs does not exceed the given limit. Constraints (7) and (8) define piecewise linear cost functions for crashing task n and the project makespan, respectively.

Apart from the cost of fast tracking, crashing, and penalty for makespan delay, the total cost also includes the cost of reserving capacity. Hence, the problem of minimizing the total operational cost is

$$\hat{h}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \boldsymbol{u}_x^T \boldsymbol{x} + \boldsymbol{u}_y^T \boldsymbol{y} + \min_{(\boldsymbol{w}_F, \boldsymbol{w}_C, \boldsymbol{v}, \boldsymbol{c}) \in \mathcal{S}(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{y})} \left\{ \boldsymbol{g}_F^T \boldsymbol{w}_F + \boldsymbol{g}_C^T \boldsymbol{w}_C + \boldsymbol{d}^T \boldsymbol{v} \right\}, \tag{9}$$

where $\boldsymbol{u}_x$ and $\boldsymbol{u}_y$ denote the capacity reservation costs corresponding to $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively.

Then, the overall cost minimization problem of the project company is

$$\min_{\boldsymbol{x}, \boldsymbol{y}} \hat{h}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}). \tag{10}$$

2.2. Uncertainties and risk

We model the uncertain duration $\tilde{\boldsymbol{z}}$ of each task as an affine function of bounded uncertain factors $\tilde{\boldsymbol{\delta}} = (\tilde{\delta}_1, \dots, \tilde{\delta}_K)$, i.e.,

$$\tilde{z}_n = z_n^0 + \sum_{k=1}^{K} z_n^k \tilde{\delta}_k,$$

where the factor coefficients $z_n^0, \ldots, z_n^K$ are known. We describe correlation between uncertain delays from different tasks using a factor-based model rather than the typical covariance matrix. As Hall et al. (2015) discuss, the factor-based model has particular value in project management applications, since (a) estimating the covariance matrix is difficult for new or unique projects due to a lack of historical data, (b) uncertain durations of different tasks often arise from common factors such as resources, and (c) the project management problem itself is computationally challenging and the linear-factor based model preserves linear structure, which limits complexity.

We denote the probability space as $(\mathbb{P}, \Omega, \mathcal{F})$, where Ω is the set of all possible outcomes, $\mathcal{F}$ is the set of events that describe task durations, and $\mathbb{P}$ is a function that assigns probabilities to events. For several reasons, for example limited historical data about similar projects and estimation error (Ward and Chapman 2003), we assume that full distributional knowledge, i.e. $\mathbb{P}$, is not available. Instead, the distributional information $\mathbb{P}$ is only partially characterized. In particular, we describe the probability by $\mathbb{P} \in \mathcal{P}$, where

$$\mathcal{P} = \left\{ \mathbb{P} \ : \ \begin{array}{l} \mathbb{P}\left(\tilde{\delta}_k \in \{\delta_{k1}, \ldots, \delta_{kL_k}\}\right) = 1, \ k = 1, \ldots, K \\ \mathbb{E}_{\mathbb{P}}\left[\chi_j^{S,k}\left(\tilde{\delta}_k\right)\right] \le \sigma_j^{S,k}, \ k = 1, \ldots, K, \ j = 1, \ldots, J_{S,k} \\ \mathbb{E}_{\mathbb{P}}\left[\chi_j^{C}\left(\boldsymbol{\xi}_j{}^T \tilde{\boldsymbol{\delta}} + \epsilon_j\right)\right] \le \sigma_j^{C}, \ j = 1, \ldots, J_C \end{array} \right\}. \tag{11}$$

In (11), $\delta_{k1}, \ldots, \delta_{kL_k}$ represent all potential realizations of $\tilde{\delta}_k$ from a discrete distribution. The use of a discrete distribution of task times is consistent with resource planning in projects using discrete time units (Herroelen et al. 1998). We incorporate the information about any estimator of an individual uncertain factor in the second constraint in (11). Specifically, $\chi_j^{S,k} : \Re \to \Re$ can be any general function to estimate the uncertain factor $\tilde{\delta}_k$, and $\sigma_j^{S,k}$ is an upper bound on the estimation. The third constraint in (11) takes into account correlation information among any subset of uncertain factors. In particular, the function $\chi_j^C : \Re \to \Re$ is used to estimate the linear combination $(\boldsymbol{\xi}_j^T \tilde{\boldsymbol{\delta}} + \epsilon_j)$, where $\boldsymbol{\xi}_j$ and ϵ_j are given parameters. Also, σ_j^C is a given upper bound on the estimation. For tractability, we assume χ_j^C to be convex. We remark that even with the convexity requirement on χ_j^C, the information set defined in (11) is general enough to include a broad class of practical information. We now show three relevant examples which can be represented as special cases of $\mathcal{P}$ as defined in (11).

Example 1 *The range of mean for $\tilde{\delta}_k$. We let $\chi_j^{S,k}(x) = x \ \forall x \in \Re$ for certain j, and $\sigma_j^{S,k} = \bar{\mu}_k$. Then the second constraint in (11) for this k and j becomes*

$$\mathbb{E}_{\mathbb{P}}\left[\tilde{\delta}_k\right] \leq \bar{\mu}_k,$$

which defines the upper bound for the mean of $\tilde{\delta}_k$. Similarly, if we let $\chi_j^{S,k}(x) = -x \ \forall x \in \Re$ and $\sigma_j^{S,k} = -\underline{\mu}_k$, the constraint becomes

$$\mathbb{E}_{\mathbb{P}}\left[\tilde{\delta}_k\right] \geq \underline{\mu}_k,$$

which defines the lower bound for the mean of $\tilde{\delta}_k$.

Example 2 *The range of any estimator for $\tilde{\delta}_k$, which generalizes the previous example. We let $\chi_j^{S,k}$ denote the underlying function of any estimator, such as $\chi_j^{S,k}(x) = x^2$, for $x \in \Re$. We then incorporate the information about upper and lower bounds, for the estimation of uncertain factor $\tilde{\delta}_k$. Indeed, since there is no restrictions on the function $\chi_j^{S,k}$; we can also incorporate information for some nonconvex estimators, such as $\chi_j^{S,k}(x) = x^3$ for the third moment.*

Example 3 *Absolute deviation. For a given j, we let the function $\chi_j^C(x) = |x|$, and let $\boldsymbol{\xi}_j$ and ϵ_j be such that $\boldsymbol{\xi}_j^T \tilde{\boldsymbol{\delta}} + \epsilon_j = \sum_{k \in \mathcal{K}_j} \frac{\tilde{\delta}_k - \mu_k}{\sigma_k}$. Here $\mathcal{K}_j \subseteq [K]$ is a given subset which contains the indices of the related uncertain factors. The third constraint in (11) for this j is then*

$$\mathbb{E}_{\mathbb{P}}\left[\left|\sum_{k \in \mathcal{K}_j} \frac{\tilde{\delta}_k - \mu_k}{\sigma_k}\right|\right] \leq \sigma_j^C.$$

This design emulates the budget of uncertainty in Bertsimas and Sim (2004), where uncertain factors are unlikely to realize in the extreme case simultaneously. The special case where $\mathcal{K}_j$ is a singleton is recently studied by Postek et al. (2015).

Remark 1 *Many alternative descriptions of the set of available distributions $\mathcal{P}$ are possible, including information on moments (Popescu 2007, Delage and Ye 2010, Zymler et al. 2013), ϕ-divergence (Ben-Tal et al. 2013, Wang et al. 2016, Jiang and Guan 2016), Wasserstein distance (Esfahani and Kuhn 2017, Gao and Kleywegt 2018), and a general framework (Wiesemann et al. 2014). However, the requirement for optimal solvability of the two-stage problem with binary decisions disqualifies several of these. We use discrete support for the task durations to be consistent with project management practice and obtain a finite-dimensional problem; the second and third constraints in (11) model individual and joint dispersion while maintaining a linear structure. In considering information on joint dispersion, even verifying the emptiness of the information set $\mathcal{P}$ is* NP*-hard due to the exponential number of elements in the sample space. Nevertheless, we show below that the problem can be solved efficiently for practical size projects.*

We demonstrate the dependence of the total cost on uncertain factors. We define the function

$$h\left(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\delta}\right)=\hat{h}\left(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}^0+\boldsymbol{Z}\boldsymbol{\delta}\right) \tag{12}$$

to represent the total cost corresponding to capacity reservation decisions $(\boldsymbol{x},\boldsymbol{y})$, and uncertain factor realizations $\boldsymbol{\delta}$. We define the vector $\boldsymbol{z}^0=(z_1^0,\ldots,z_N^0)$, and the matrix $\boldsymbol{Z}\in\Re^{N\times K}$ such that the element in the nth row and kth column is z_n^k, $n\in[N]$, $k\in[K]$.

Given any capacity reservation decisions $(\boldsymbol{x},\boldsymbol{y})$, the corresponding total cost is $h\left(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{\delta}}\right)$, which is a random variable depending on the elementary uncertainties $\tilde{\boldsymbol{\delta}}$. As discussed in Section 1, we assume that the decision maker has a benchmark cost, τ, and her objective is to minimize the risk that the cost exceeds τ. In order to incorporate the effect of the target τ, we evaluate a random cost by its loss relative to this target. We define $\mathcal{W}=\{\tilde{w}:\tilde{w}=\tilde{c}-\tau,\tilde{v}\in\mathcal{V}\}$ as the set of all possible losses with respect to the target, where $\mathcal{V}$ is the set of all possible costs. Therefore, a loss $\tilde{w}$ with $\mathbb{P}(\tilde{w}\le 0)=1$ indicates that the cost is never higher than the target τ.

Following Brown and Sim (2009), and Hall et al. (2015), the target-oriented objective is defined as follows.

Definition 1 *A function $\rho: \mathcal{W} \to [0,\infty]$ is an Underperformance Riskiness Index (URI) if, for all $\tilde{w}, \tilde{w}^o \in \mathcal{W}$, it satisfies the following properties:*

1. *Monotonicity: if $\tilde{w} \le \tilde{w}^o$, then $\rho(\tilde{w}) \le \rho(\tilde{w}^o)$.*
2. *Satisficing:*
   (a) *Attainment Content: if $\tilde{w} \le 0$, then $\rho(\tilde{w}) = 0$;*
   (b) *Starvation Aversion: if $\tilde{w} > 0$, then $\rho(\tilde{w}) = \infty$.*
3. *Convexity: $\rho(\lambda\tilde{w} + (1-\lambda)\tilde{w}^o) \le \lambda\rho(\tilde{w}) + (1-\lambda)\rho(\tilde{w}^o)$, for all $\lambda \in [0,1]$.*
4. *Positive Homogeneity: $\rho(\lambda\tilde{w}) = \lambda\rho(\tilde{w})$, for all $\lambda > 0$.*

Monotonicity indicates that if the cost $\tilde{w}^o$ is always higher than $\tilde{w}$, then it is associated with higher risk. Satisficing implies that a random cost has no risk if it is always lower than the target; by contrast, it has infinite risk if the cost is always strictly greater than the target. Convexity is motivated by a preference for diversification. Positive Homogeneity establishes the cardinal nature of risk, e.g. $2\tilde{w}$ is as twice as risky as $\tilde{w}$.

Apart from its justification from the above properties, the URI can be dually represented by classical convex risk measures, as we now show.

Theorem 1 *(Hall et al. 2015) A function $\rho: \mathcal{W} \to [0,\infty]$ is a URI if and only if it has the representation*

$$\rho(\tilde{w}) = \inf\left\{\alpha \ : \ \psi\left(\frac{\tilde{w}}{\alpha}\right) \le 0, \ \alpha > 0\right\}, \tag{13}$$

where we define $\inf\emptyset = \infty$ by convention and ψ is a normalized convex risk measure. That is, it satisfies the properties of Monotonicity, Cash Invariance, Convexity, and has $\psi(0) = 0$. Conversely, given a URI ρ, the underlying normalized convex risk measure is given by

$$\psi(\tilde{w}) = \min\{a \ : \ \rho(\tilde{w} + a) \le 1\}. \tag{14}$$

To establish a connection with classical expected utility theory, we focus on a special class of convex risk measure, which is a *shortfall risk measure* (Föllmer and Schied 2002). We redefine it under distributional ambiguity.

Definition 2 *A shortfall risk measure is any function $\psi_U : \mathcal{W} \to \Re$ defined by*

$$\psi_U(\tilde{w}) = \inf_{\eta \in \Re} \left\{ \eta \ : \ \sup_{\mathbb{P} \in \mathcal{P}} \mathbb{E}_{\mathbb{P}} \left[u\left(\tilde{w} - \eta\right)\right] \leq 0 \right\},$$

where $u : \Re \to \Re$ is an increasing convex differentiable function normalized by $u(0) = 0$ and $u'(0) = 1$. Correspondingly, using the dual representation in Theorem 1, we define the utility-based URI, $\rho_U : \mathcal{W} \to [0, \infty]$, as one based on a shortfall risk measure as follows:

$$\rho_U(\tilde{w}) = \inf \left\{ \alpha \ : \ \sup_{\mathbb{P} \in \mathcal{P}} \mathbb{E}_{\mathbb{P}} \left[u\left(\frac{\tilde{w}}{\alpha}\right)\right] \leq 0, \alpha > 0 \right\}. \tag{15}$$

To preserve linearity of the problem, we assume

$$u(w) = \max_{m \in [M]} \left\{ a_m w + b_m \right\} \tag{16}$$

with $[M] = \{1, 2, \ldots, M\}$, and given parameters $0 \leq a_1 \leq a_2 \leq \cdots \leq a_M$, $b_1, \ldots, b_M \in \Re$. We observe that this piecewise-linear function can be used to approximate any general utility function.

Then, from (15) and (16), the problem of minimizing the URI of overall uncertain cost is

$$\begin{aligned} \alpha^* \ = \ & \inf \ \alpha \\ & \text{s.t.} \ \sup_{\mathbb{P} \in \mathcal{P}} \mathbb{E}_{\mathbb{P}} \left[\max_{m \in [M]} \left\{ a_m \frac{h\left(\boldsymbol{x}, \boldsymbol{y}, \tilde{\boldsymbol{\delta}}\right) - \tau}{\alpha} + b_m \right\} \right] \leq 0 \\ & \qquad \alpha > 0 \\ & \qquad (\boldsymbol{x}, \boldsymbol{y}) \in \Lambda, \end{aligned} \tag{17}$$

where $(\boldsymbol{x}, \boldsymbol{y}) \in \Lambda$ represents the feasible set for the capacity reservation decision $(\boldsymbol{x}, \boldsymbol{y})$. We assume that all constraints in $(\boldsymbol{x}, \boldsymbol{y}) \in \Lambda$ are linear, except for $\boldsymbol{x} \in \{0, 1\}^O$ and $\boldsymbol{y} \in \{0, 1\}^N$.

The following two results discuss the formal tractability of problem (10) in the deterministic case, i.e. where all probability distributions are single-point.

Theorem 2 *The recognition version of problem (10) with no uncertainty is unary* NP*-complete.*

Theorem 2 establishes that, even without uncertainty, the existence of an efficient algorithm for problem (10) is highly unlikely (Garey and Johnson 1979). The solvability of another closely related problem is clarified by the following remark.

Remark 2 *Problem (10) with no uncertainty is solvable in pseudopolynomal time if the precedence constraint network is a chain. Further, the recognition version of this problem is binary* NP-*complete.*

3. Solution Procedure

It is difficult to solve problem (17) directly for several reasons. First, the problem involves several levels of maximization and minimization; hence, given any feasible reservation decision $(\boldsymbol{x}, \boldsymbol{y})$, even evaluating the objective value is problematic. Second, the evaluation of $\sup_{\mathbb{P}\in\mathcal{P}}$ requires a high-dimensional optimization. Third, as shown in Theorem 2, the underlying deterministic problem is highly intractable.

We develop a procedure to solve problem (17). The first step is to ensure the closedness of the feasible set of the optimization problem, which is not guaranteed in problem (17) since there is a constraint of $\alpha > 0$. Hence, we analyze an ϵ-closure of problem (17) instead.

Proposition 1 *For any $\epsilon > 0$, we have $\alpha^* \leq \alpha^*_\epsilon \leq \alpha^* + \epsilon$, where*

$$\begin{aligned} \alpha^*_\epsilon = \min\ & \alpha \\ s.t.\ & \sup_{\mathbb{P}\in\mathcal{P}} \mathbb{E}_{\mathbb{P}}\left[\max_{m\in[M]}\left\{a_m h\left(\boldsymbol{x}, \boldsymbol{y}, \tilde{\boldsymbol{\delta}}\right) - a_m\tau + b_m\alpha\right\}\right] \leq 0 \\ & \alpha \geq \epsilon \\ & (\boldsymbol{x}, \boldsymbol{y}) \in \Lambda \end{aligned} \tag{18}$$

From Proposition 1, the optimal solution of problem (18) is arbitrarily close to that of problem (17). We therefore focus on solving problem (18).

In solving problem (18), a necessary subproblem is the evaluation of $\sup_{\mathbb{P}\in\mathcal{P}} \mathbb{E}_{\mathbb{P}}\left[\max_{m\in[M]}\left\{a_m h\left(\boldsymbol{x}, \boldsymbol{y}, \tilde{\boldsymbol{\delta}}\right) - a_m\tau + b_m\alpha\right\}\right]$. That is, given capacity reservation decisions $(\boldsymbol{x}, \boldsymbol{y})$, we need to evaluate performance under the worst case distribution. We consider this problem as follows.

Lemma 1

$$\begin{aligned}
&\sup_{\mathbb{P}\in\mathcal{P}} \mathbb{E}_{\mathbb{P}}\left[\max_{m\in[M]}\left\{a_m h\left(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{\delta}}\right) - a_m\tau + b_m\alpha\right\}\right] \\
=\ &\min\ \tau_0 + \sum_{k=1}^{K} {\boldsymbol{\sigma}^{S,k}}^T \boldsymbol{\tau}^{S,k} + {\boldsymbol{\sigma}^{C}}^T \boldsymbol{\tau}^{C} \\
&\quad \text{s.t. } a_m\boldsymbol{u}_x^T\boldsymbol{x} + a_m\boldsymbol{u}_y^T\boldsymbol{y} + a_m\boldsymbol{g}_F^T\boldsymbol{w}_F^i + a_m\boldsymbol{g}_C^T\boldsymbol{w}_C^i + a_m\boldsymbol{d}^T\boldsymbol{v}^i - a_m\tau + b_m\alpha \\
&\qquad\qquad \le \tau_0 + \sum_{k=1}^{K}\sum_{j=1}^{J_{S,k}} \omega_j^{S,k,i}\tau_j^{S,k} + \sum_{j=1}^{J_C}\omega_j^{C,i}\tau_j^{C},\ i\in[I],\ m\in[M] \\
&\qquad (\boldsymbol{w}_F^i,\boldsymbol{w}_C^i,\boldsymbol{v}^i,\boldsymbol{c}^i)\in\mathcal{S}(\boldsymbol{z}^0+\boldsymbol{Z}\boldsymbol{\delta}^i,\boldsymbol{x},\boldsymbol{y}),\ i\in[I] \\
&\qquad \boldsymbol{\tau}^C,\ \boldsymbol{\tau}^{S,k}\ge\boldsymbol{0},\ k\in[K],
\end{aligned} \tag{19}$$

where we let $\boldsymbol{\delta}^i$, $i=1,\ldots,I$ *represent all arbitrarily indexed elements of the set* $\mathcal{Q}$ *defined as*

$$\mathcal{Q} = \left\{\boldsymbol{\delta}\ :\ \delta_k\in\{\delta_{k1},\ldots,\delta_{kL_k}\},\ k\in[K]\right\}, \tag{20}$$

and correspondingly,

$$\omega_j^{S,k,i} = \chi_j^{S,k}\left(\delta_k^i\right),$$

$$\omega_j^{C,i} = \chi_j^{C}\left({\boldsymbol{\xi}_j}^T\boldsymbol{\delta}^i + \epsilon_j\right),$$

are also given parameters. In addition, $\tau_0,\ \boldsymbol{\tau}^{S,k},\ \boldsymbol{\tau}^C,\ \boldsymbol{w}_F^i,\ \boldsymbol{w}_C^i,\ \boldsymbol{v}^i,\ \boldsymbol{c}^i,\ k\in[K],\ i\in[I]$, *are decision variables.*

Lemma 1 enables the following reformulation of the overall problem.

Theorem 3 *The overall problem (18) is equivalent to the following mixed-integer programming problem*

$$\min\ \ \alpha \tag{21}$$

$$\text{s.t. } \tau_0 + \sum_{k=1}^{K} {\boldsymbol{\sigma}^{S,k}}^T \boldsymbol{\tau}^{S,k} + {\boldsymbol{\sigma}^{C}}^T \boldsymbol{\tau}^{C} \le 0 \tag{22}$$

$$\begin{aligned}
&a_m\boldsymbol{u}_x^T\boldsymbol{x} + a_m\boldsymbol{u}_y^T\boldsymbol{y} + a_m\boldsymbol{g}_F^T\boldsymbol{w}_F^i + a_m\boldsymbol{g}_C^T\boldsymbol{w}_C^i + a_m\boldsymbol{d}^T\boldsymbol{v}^i - a_m\tau + b_m\alpha \\
&\qquad \le \tau_0 + \sum_{k=1}^{K}\sum_{j=1}^{J_{S,k}} \omega_j^{S,k,i}\tau_j^{S,k} + \sum_{j=1}^{J_C}\omega_j^{C,i}\tau_j^{C},\ i\in[I],\ m\in[M]
\end{aligned} \tag{23}$$

$$(\boldsymbol{w}_F^i,\boldsymbol{w}_C^i,\boldsymbol{v}^i,\boldsymbol{c}^i)\in\mathcal{S}(\boldsymbol{z}^0+\boldsymbol{Z}\boldsymbol{\delta}^i,\boldsymbol{x},\boldsymbol{y}),\ i\in[I] \tag{24}$$

$$\boldsymbol{\tau}^C,\ \boldsymbol{\tau}^{S,k} \geq \mathbf{0},\ k \in [K] \tag{25}$$

$$\alpha \geq \epsilon \tag{26}$$

$$(\boldsymbol{x}, \boldsymbol{y}) \in \Lambda. \tag{27}$$

Since I is the number of elements in $\mathcal{Q}$ which is exponentially large, problem (21)–(27) has many decision variables and constraints. Hence, we apply a column and constraint generation method.

Definition 3 *Let problem (R) denote problem (21)–(27) where the constraints defined over the set* $[I]$ *are replaced by a subset of constraints defined by* $[I^o]$ *with* $I^o \leq I$.

If the solution of problem (R) is feasible for the original problem (21)–(27), then we stop. Otherwise, we increase the value of I^o, and solve the relaxed problem again.

We consider the case where the function χ_j^C in the distributional information set defined in (11) is piecewise linear and convex, for all j. Specifically, let

$$\chi_j^C(x) = \max_{m \in [M_j]} \left\{ r_j^m x + \vartheta_j^m \right\}, \tag{28}$$

where r_j^m and ϑ_j^m are given parameters.

Now, we demonstrate how to check the feasibility of constraints (23) and (24). We define a bilinear programming problem which is used for this purpose. Observe that $\mathcal{S}(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{y})$ defines a polyhedron of $(\boldsymbol{w}_F, \boldsymbol{w}_C, \boldsymbol{v}, \boldsymbol{c})$, where the constraints are also linear in $(\boldsymbol{x}, \boldsymbol{y})$. Following the definition of $\mathcal{S}(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{y})$, we find matrices $\boldsymbol{A}_F$, $\boldsymbol{A}_C$, $\boldsymbol{A}_v$, $\boldsymbol{A}_c$, $\boldsymbol{A}_x$, $\boldsymbol{A}_y$, $\boldsymbol{A}_z$, and vector $\boldsymbol{b}^o$ such that

$$\mathcal{S}(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{y}) = \left\{ (\boldsymbol{w}_F, \boldsymbol{w}_C, \boldsymbol{v}, \boldsymbol{c}) \ : \ \begin{array}{l} \boldsymbol{A}_F \boldsymbol{w}_F + \boldsymbol{A}_C \boldsymbol{w}_C + \boldsymbol{A}_v \boldsymbol{v} + \boldsymbol{A}_c \boldsymbol{c} \geq \boldsymbol{A}_x \boldsymbol{x} + \boldsymbol{A}_y \boldsymbol{y} + \boldsymbol{A}_z \boldsymbol{z} + \boldsymbol{b}^o \\ \boldsymbol{w}_F \geq \mathbf{0}, \boldsymbol{w}_C \geq \mathbf{0}, \boldsymbol{v} \geq \mathbf{0}, \boldsymbol{c} \geq \mathbf{0} \end{array} \right\}.$$

Based on these matrices, we let P denote the polyhedron defined by

$$P = \left\{ \boldsymbol{p} \geq \mathbf{0} \ : \ \boldsymbol{p}^T \left(\boldsymbol{A}_F \ \boldsymbol{A}_C \ \boldsymbol{A}_v \ \boldsymbol{A}_c \right) \leq \left(\boldsymbol{g}_F^T \ \boldsymbol{g}_C^T \ \boldsymbol{d}^T \ \mathbf{0}^T \right) \right\}.$$

Further, for any $m \in [M]$, we define a function $q_m^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\tau}^C, \boldsymbol{\tau}^{S,1}, \ldots, \boldsymbol{\tau}^{S,K})$ as the optimal value of an optimization problem parameterized by $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\tau}^C, \boldsymbol{\tau}^{S,1}, \ldots, \boldsymbol{\tau}^{S,K})$ as follows:

$$\begin{aligned} & q_m^*\left(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\tau}^C, \boldsymbol{\tau}^{S,1}, \ldots, \boldsymbol{\tau}^{S,K}\right) \\ = \max \quad & q_m\left(\boldsymbol{p}, \boldsymbol{\delta}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\tau}^C, \boldsymbol{\tau}^{S,1}, \ldots, \boldsymbol{\tau}^{S,K}\right) \\ \text{s.t.} \quad & \boldsymbol{p} \in P, \quad \boldsymbol{\delta} \in \mathcal{Q}, \end{aligned} \tag{29}$$

where

$$\begin{aligned} & q_m\left(\boldsymbol{p}, \boldsymbol{\delta}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\tau}^C, \boldsymbol{\tau}^{S,1}, \ldots, \boldsymbol{\tau}^{S,K}\right) \\ = & -\sum_{k=1}^{K}\sum_{j=1}^{J_{S,k}} \tau_j^{S,k} \chi_j^{S,k}\left(\delta_k\right) - \sum_{j=1}^{J_C} \tau_j^C \chi_j^C\left(\boldsymbol{\xi}_j^T \boldsymbol{\delta} + \epsilon_j\right) + a_m \boldsymbol{\delta}^T (\boldsymbol{A}_z \boldsymbol{Z})^T \boldsymbol{p} + a_m\left(\boldsymbol{A}_x \boldsymbol{x} + \boldsymbol{A}_y \boldsymbol{y} + \boldsymbol{A}_z \boldsymbol{z}^0 + \boldsymbol{b}^o\right)^T \boldsymbol{p} \end{aligned}$$

has a bilinear term in the decision variables $\boldsymbol{p}$ and $\boldsymbol{\delta}$.

The next result establishes necessary and sufficient conditions for a solution to the relaxed problem (R) to be feasible for constraints (23) and (24).

Lemma 2 *Consider any feasible solution $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\tau}^{S,1}, \ldots, \boldsymbol{\tau}^{S,K}, \boldsymbol{\tau}^C, \tau_0, \alpha$ for the relaxed problem (R). If $\forall m \in [M]$,*

$$q_m^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\tau}^C, \boldsymbol{\tau}^{S,1}, \ldots, \boldsymbol{\tau}^{S,K}) \le \tau_0 + a_m \tau - b_m \alpha - a_m \boldsymbol{u}_x^T \boldsymbol{x} - a_m \boldsymbol{u}_y^T \boldsymbol{y}, \tag{30}$$

then this solution is feasible for constraints (23) and (24). Otherwise, this solution is not feasible for constraints (23) and (24), since for any $m_* \in [M]$ and $i^* \in [I]$ with

$$\max_{\boldsymbol{p} \in P} q_{m_*}\left(\boldsymbol{p}, \boldsymbol{\delta}^{i^*}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\tau}^C, \boldsymbol{\tau}^{S,1}, \ldots, \boldsymbol{\tau}^{S,k}\right) > \tau_0 + a_{m_*} \tau - b_{m_*} \alpha - a_{m_*} \boldsymbol{u}_x^T \boldsymbol{x} - a_{m_*} \boldsymbol{u}_y^T \boldsymbol{y}, \tag{31}$$

there does not exist $(\boldsymbol{w}_F^{i^*}, \boldsymbol{w}_C^{i^*}, \boldsymbol{v}^{i^*}, \boldsymbol{c}^{i^*}) \in \mathcal{S}(\boldsymbol{z}^0 + \boldsymbol{Z}\boldsymbol{\delta}^{i^*}, \boldsymbol{x}, \boldsymbol{y})$ with

$$\begin{aligned} & a_{m_*} \boldsymbol{u}_x^T \boldsymbol{x} + a_{m_*} \boldsymbol{u}_y^T \boldsymbol{y} + a_{m_*} \boldsymbol{g}_F^T \boldsymbol{w}_F^{i^*} + a_{m_*} \boldsymbol{g}_C^T \boldsymbol{w}_C^{i^*} + a_{m_*} \boldsymbol{d}^T \boldsymbol{v}^{i^*} - a_{m_*} \tau + b_{m_*} \alpha \\ & \le \tau_0 + \textstyle\sum_{k=1}^{K} \sum_{j=1}^{J_{S,k}} \omega_j^{S,k,i^*} \tau_j^{S,k} + \sum_{j=1}^{J_C} \omega_j^{C,i^*} \tau_j^C, \end{aligned}$$

i.e., constraints (23) and (24) are not satisfied for $m = m_*$ and $i = i^*$.

Lemma 2 provides both a stopping rule, and an iterative improvement direction, for our Column and Constraint Generation (CCG) algorithm described below. Specifically, every time after solving the relaxed problem, we check whether $q_m^* \leq \tau_0 + a_m\tau - b_m\alpha - a_m\boldsymbol{u}_x^T\boldsymbol{x} - a_m\boldsymbol{u}_y^T\boldsymbol{y}$ for all $m \in [M]$. If so, then we stop the iterative process. Otherwise, we need to find $m \in [M]$ such that any solution of the corresponding bilinear problem (29) with objective value greater than $(\tau_0 + a_m\tau - b_m\alpha - a_m\boldsymbol{u}_x^T\boldsymbol{x} - a_m\boldsymbol{u}_y^T\boldsymbol{y})$ and add it to the constraints. It now remains to solve the bilinear problem (29). While bilinear problems are in general *NP*-hard, problem (29) can be solved using a computationally efficient approach, as we discuss after presenting the following result.

Lemma 3 *The bilinear programming problem (29) is equivalent to the following mixed-integer programming problem,*

$$
\begin{aligned}
\max \quad & a_m\boldsymbol{c}^T\boldsymbol{p} - \sum_{k=1}^{K}\sum_{j=1}^{J_{S,k}}\tau_j^{S,k}\sum_{l=1}^{L_k}\chi_j^{S,k}(\delta_{kl})\phi_{kl} - \sum_{j=1}^{J_C}\tau_j^C\Omega_j + a_m\sum_{k=1}^{K}\sum_{l=1}^{L_k}\zeta_{kl} \\
\text{s.t.} \quad & \boldsymbol{p} \in P \\
& \sum_{l=1}^{L_k}\phi_{kl} = 1, \quad k \in [K] \\
& \phi_{kl} \in \{0,1\}, \quad k \in [K],\ l \in [L_k] \\
& \zeta_{kl} \leq \delta_{kl}\boldsymbol{b}_k\boldsymbol{p} + M(1-\phi_{kl}), \quad k \in [K],\ l \in [L_k] \\
& \zeta_{kl} \leq M\phi_{kl}, \quad k \in [K],\ l \in [L_k], \\
& \Omega_j \geq r_j^m\left(\sum_{k=1}^{K}\xi_{jk}\sum_{l=1}^{L_k}\phi_{kl}\delta_{kl} + \epsilon_j\right) + \vartheta_j^m, \quad m \in [M_j],\ j \in [J_C],
\end{aligned}
\tag{32}
$$

where we let $\boldsymbol{B} = (\boldsymbol{A}_z\boldsymbol{Z})^T$*, let* $\boldsymbol{b}_k$ *denote its* k*th row vector,* $\boldsymbol{c} = \boldsymbol{A}_x\boldsymbol{x} + \boldsymbol{A}_y\boldsymbol{y} + \boldsymbol{A}_z\boldsymbol{z}^0 + \boldsymbol{b}^o$*, and let* M *be a sufficiently large constant.*

Remark 3 *We recall that all bilinear optimization problems have an optimal solution that is an extreme point of the feasible region, hence they can be equivalently formulated as a mixed integer programming problem using the approach in Lemma 3. In general, this approach requires one auxiliary binary decision variable for each extreme point, the number of which is exponentially large. Hence, the corresponding mixed integer programming problem is in general still computationally intractable and the reformulation does not greatly help to solve the bilinear optimization problem. However, in problem (29), since the constraints on* $\boldsymbol{\delta}$ *can be decomposed into one constraint* *for each uncertain factor δ_k, $k \in [K]$, the number of auxiliary binary decision variable reduces to $L_1 + \ldots + L_K$. Consequently, the problem can be solved efficiently. When the distributional information $\mathcal{P}$ contains only bound information, a similar approach is proposed to formulate the bilinear problem into a mixed integer programming problem; see, for example, Jiang et al. (2012), and Zeng and Zhao (2013). However, in Lemma 3, we extend this approach to the case where $\mathcal{P}$ contains frequency information as well as information about different scenarios.*

We are now ready to present our algorithm, CCG, for solving the overall problem.

Algorithm CCG

---

1. Initialize $I^o = 1$, and $\boldsymbol{\delta}^1 = (\delta_{11}, \ldots, \delta_{K1})$.

2. Solve the relaxed problem (R), and denote the solution by $\boldsymbol{x}, \boldsymbol{y}$, $\boldsymbol{\tau}^{S,1}, \ldots, \boldsymbol{\tau}^{S,K}$, $\boldsymbol{\tau}^C$, τ_0, and α. In addition, set $m = 1$.

3. Define an instance of the bilinear problem (29) using the solution found in Step 2 and the current value of m. Solve problem (29) by solving the equivalent mixed integer programming problem (32); output the objective value O_b and an optimal solution of $\boldsymbol{\delta}$ as $\boldsymbol{\delta}^*$.

4. If $O_b > \tau_0 + a_m \tau - b_m \alpha - a_m \boldsymbol{u}_x^T \boldsymbol{x} - a_m \boldsymbol{u}_y^T \boldsymbol{y}$, then set $\boldsymbol{\delta}^{I^o+1} = \boldsymbol{\delta}^*$, $I^o = I^o + 1$, and go to Step 2.

5. If $m < M$, then set $m = m + 1$, and go to Step 3.

6. Output $\boldsymbol{x}, \boldsymbol{y}$ as optimal decisions, and stop.

Theorem 4 *Algorithm CCG outputs an optimal solution for the overall problem in a finite number of steps.*

Remark 4 *When the function χ_j^C is not piecewise linear and convex but an arbitrary convex function, the problem can still be solved using a procedure similar to Algorithm CCG. The only difference is that, given a solution to the relaxed problem (R), we would need a more complex procedure to verify its feasibility for the original problem (21)-(27). In particular, when the function*

χ_j^C *is not piecewise linear, we can no longer use the MILP (32) since the last constraint in (32) is replaced by* $\Omega_j \geq \chi_j^C(\sum_{k=1}^{K} \xi_{jk} \sum_{l=1}^{L_k} \phi_{kl}\delta_{kl} + \epsilon_j)$*, which has nonlinear structure since* χ_j^C *is not piece-wise linear. However, as* $\boldsymbol{\delta}$ *can take only a finite number of values due to the use of a discrete distribution, we can still linearize this constraint by enumerating all possible* $\boldsymbol{\delta}$ *values; hence, we still obtain a MILP. This MILP can also be solved using a column and constraint generation method.*

4. Computational Study

In this section, we describe a computational study of Algorithm CCG. The objectives of this study are as follows. In Section 4.1, we test the computational efficiency of Algorithm CCG, to establish the size of projects for which it can provide optimal solutions. In Section 4.2, we compare the performance of decisions recommended by our algorithm against those from four alternative procedures that serve as benchmarks. Section 4.3 includes an analysis of sensitivity to the most important factors. Section 4.4 studies the robustness of the performance of those decisions against incorrect assumptions about the distribution of task time durations. Section 4.5 demonstrates the computational advantage of Algorithm CCG over direct solution.

4.1. Computational efficiency

We study the computational efficiency of Algorithm CCG for projects with various characteristics. As recommended by Hall and Posner (2001), our experiment includes a number of instances with different parameters that may affect performance. Specifically, we use the 1250 project management instances from the webpage of Operations Research & Scheduling Research Group at http://www.projectmanagement.ugent.be. We study how computational efficiency changes as a function of the number of tasks, N, and the depth indicator of the network, $\Gamma = (\varrho - 1)/(N - 1)$, where ϱ is the length of the longest chain. In the dataset of 1250 project instances, the number of tasks, N, in a project takes values in $\{12, 22, 32, 42, 52\}$. Also, Γ takes a wide range of values. We divide the instances with the same N value into two groups, where one group contains instances with $\Gamma < \Gamma_0$ and the other contains instances with $\Gamma \geq \Gamma_0$. We choose Γ_0 such that there is a similar number of instances in each group. All instances are solved using CPLEX 12 on a computer with an Intel Xeon E5 processor and 64GB RAM.

Table 1 reports the mean and maximum computation time in seconds for the instances corresponding to each group. Our result shows that within a few hours of computation, Algorithm CCG solves project management instances with up to 52 tasks. When N takes a relatively large value such as 52, the computation time increases with the depth indicator. But this effect is less obvious for small values of N. Since Algorithm CCG is designed to be used at the planning stage of projects, rather than in real time, these computation times are reasonable enough to be useful in practice.

	Average time (in seconds)		Maximum time (in seconds)	
N	$\Gamma < \Gamma_0$	$\Gamma \geq \Gamma_0$	$\Gamma < \Gamma_0$	$\Gamma \geq \Gamma_0$
12	58	37	194	104
22	86	83	163	145
32	80	47	169	122
42	89	83	190	3373
52	192	349	1700	6758

Table 1 Computation time for various projects.

4.2. Comparison with benchmarks

We first describe four approaches to risk minimization that can be used as alternatives to Algorithm CCG. These approaches are apparently the best available alternatives, and are used as benchmarks in our computational study. The first benchmark is to minimize expected cost. The second is to use the mean value of task durations deterministically, and make capacity reservation decisions using that information. The third and fourth approaches are both to optimize the cost in the worst scenario of the uncertainty realization. Under this criterion, the third approach of Zeng & Zhao (2013) solves the exact optimal solution, whereas the fourth approach of Bertsimas et. al. (2013) uses a heuristic to obtain a sub-optimal solution with higher computational efficiency.

In order to provide an objective comparison between our approach and the benchmark approaches, we use the first 240 project management instances from the webpage of Operations

Research & Scheduling Research Group. The instances are generated as described there. In each project instance, the number of tasks is either 27 or 52, and the number of immediate precedence relationships varies from 56 to 257. The underlining task duration, $\boldsymbol{b}$, is specified by the instance data as well. The other parameters are specified as follows.

To highlight the influence of capacity reservation, in our computational study we do not incorporate fast tracking decisions, since they require no capacity. To implement this, we set the upper limit of fast tracking, $\boldsymbol{\gamma}$, to 0. For similar reasons, we let Λ, the feasible set for the capacity reservation decisions, be the set of all binary vectors.

For crashing, let $\boldsymbol{g}_c = \mathbf{0}$ and $\boldsymbol{e}_c$ be large values that make their respective constraints redundant. For the general capacity, we define a price list containing five prices with quantity discounts, such that when the purchasing quantity is $a_i = i \times \mathbf{1}'\boldsymbol{b}$, the per-unit price is $0.01 \times (1 - 0.05 * (i-1))$ and hence $u_{xi} = a_i \times 0.01 \times (1 - 0.05 * (i-1))$, $i = 1, \ldots, 5$. For the customized capacity reservation, we let the cost be $\boldsymbol{u}_y = 10 \times \mathbf{1}$, and let $\boldsymbol{\eta} = \mathbf{1}$ such that all crashing requires customized capacity.

We assume there are three uncertain factors, $\tilde{\delta}_i, i = 1, 2, 3$, all of which follow an independent and identical beta-binomial distribution with $n = 6, \alpha = 1, \beta = 3$. We let the delay of each task be $\tilde{z}_j = b_j\tilde{\delta}_1 + b_j\tilde{\delta}_2 + b_j\tilde{\delta}_3$, where b_j is the underlying task duration specified from the instance data. Therefore, in total we have three uncertain underlying uncertain factors, and each of them can take seven different values.

Finally, we define the cost for delay in makespan to have a two-piece linear structure, i.e., $Q = 2$. Specifically, we let the breakpoint l_1 be four times the makespan when there is no delay; the cost rate for makespan lower than l_1 is $d_1 = 1$, and beyond l_1 it is $d_2 = 3$.

For each instance, we perform the following steps:

a. We estimate the information set for uncertain factors. For individual information, we use scenario, mean, and variance. For the joint information, we use the mean dispersion. Specifically,

$$\mathcal{P} = \left\{ \mathbb{P} \ : \ \begin{array}{l} \mathbb{P}\left(\tilde{\delta}_k \in \{\delta_{k1}, \ldots, \delta_{kL_k}\}\right) = 1, \ \mathbb{E}_{\mathbb{P}}\left[\tilde{\delta}_k\right] = \mu_k, \ \mathbb{E}_{\mathbb{P}}\left[\tilde{\delta}_k^2\right] \le \sigma_k^2, \ \forall k \in [K] \\ \mathbb{E}_{\mathbb{P}}\left[\left|\sum_{k \in [K]} \dfrac{\tilde{\delta}_k - \mu_k}{\sigma_k}\right|\right] \le \sigma^C \end{array} \right\}.$$

Based on this information set, we solve the first stage solutions using five solution approaches: ADRO, minimization of expected cost, use of expected parameter values as deterministic, the two worst-scenario robust optimization approaches (Zeng & Zhao 2013, Bertsimas et al. 2013). For the ADRO approach, the target is pre-determined as 120% of the minimal expected cost. These first stage solutions define the level of capacity reservation.

b. To compare the resulting costs, we simulate the realization of the uncertainty based on the underlying beta-binomial distribution. We solve the operational stage problem over 10,000 scenarios to compute the performance of each solution.

c. For each approach, among the cost for all scenarios, we calculate their best (minimal) value, worst (maximal) value, average, variance, VaR and CVaR at 5%, probability that the cost exceeds the target, and expected loss with respect to the target. Across all the criteria, a low value is preferable. To simplify the comparison, we normalize the values by the corresponding performance from the ADRO approach.

Approach	Best	Worst	Average	Variance	VaR5%	CVaR5%	EP	EL
ADRO	1	1	1	1	1	1	1	1
Robust-Z	1.2108	0.9369	1.1110	0.8734	1.0269	1.0056	3.1113	2.8687
Robust-B	1.2108	0.9369	1.1112	0.8787	1.0288	1.0063	3.1430	2.7878
RN	0.6269	1.3723	0.8685	5.3727	1.1599	1.2432	1.6510	4.3490
D	0.5636	1.7165	0.8998	11.1462	1.3953	1.5617	2.3266	9.7579

Robust-Z=Robust approach by Zeng & Zhao (2013); Robust-B=Robust approach by Bertsimas et al. (2013); RN=Risk Neutral; D=Deterministic; EP = Exceeding probability; EL = expected loss.

Table 2 Comparison with benchmark approaches.

Table 2 presents the average performance from the 240 instances for each approach. Due to normalization, each performance measure for the ADRO approach in Table 2 is 1. For the other approaches, a number larger than 1 indicates that the ADRO approach outperforms them. From the table, ADRO is not as good as the risk neutral and deterministic approaches in the best scenario and average, since the ADRO approach is risk averse. In terms of worst scenario and variance, ADRO is worse than the two robust benchmark approaches. This is not surprising, since the two robust benchmark approaches are extremely risk averse and indeed optimize the worst scenario performance. Even so, these two robust approaches only slightly outperform the ADRO approach. When using all other performance measures, the results from ADRO are always better, and typically much better, than those from both the benchmark approaches. The ADRO approach requires more computation time than the benchmark approaches since more information is exploited. While the ADRO approach needs 28 iterations to solve the problem on average, the approach by Zeng & Zhao (2013) only needs two iterations. Similarly, while our overall solution time is 62 seconds, that of Zeng & Zhao (2013) is only 0.6 second. However, the computational efficiency of the approach by Zeng & Zhao (2013) results in part from finding the worst case scenario, which would be trivial here since the worst case scenario is one where all uncertain factors realize at maximum value. From Table 2, the ADRO approach results in better performance. In addition, the result in Section 4.1 indicates that for instances with reasonable size, the ADRO approach is efficient enough, especially since the problem being solved is an offline one at the planning stage of a project.

To illustrate the performance over the all 240 instances, we also provide, in Figure 1, a histogram of the performances from the 240 instances. As in Table 2, the results are normalized against a value of 1 for the ADRO approach.

4.3. Sensitivity analysis

We study the sensitivity of the results from our ADRO approach to the size and depth of projects. Our initial intuition is as follows. When we have fewer tasks in the project, we have less flexibility to adjust operational decisions. Hence, accurate capacity planning is more important, and the advantage of using the ADRO approach should be more significant. Similarly, when the value of the depth indicator Γ is high, the longest chain is relatively long and hence the uncertain delay is potentially large. In that case, the ADRO approach should provide a more significant advantage, since it incorporates risk, whereas the risk neutral and deterministic benchmark approaches do

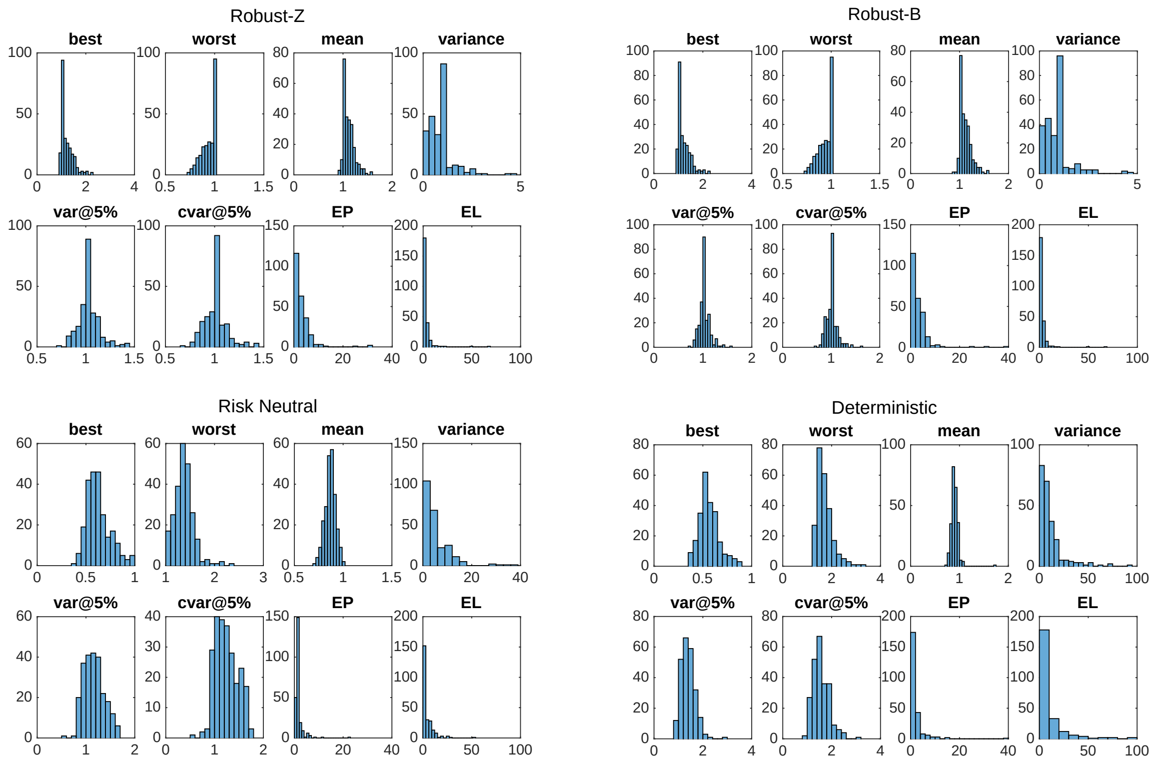


Figure 1 **Comparison with benchmark approaches**

not. The above insights are supported by the comparison between our ADRO approach and those two benchmark approaches. However, similar patterns are not observed for the robust approaches, hence we do not report their results in detail.

We divide the 240 instances into two groups based on the number of tasks, N. We let the instances with $N = 27$ form one group, and those with $N = 52$ form another. We then use ANOVA to study whether the average performance from these two groups is different. The results are summarized in Table 3. They show that the advantage of the ADRO approach over both the benchmark approaches is significant in both groups, but more significant in the group with small N. Similarly, we provide sensitivity analysis to the depth indicator Γ, which varies from 0.0784 to 0.4231 in the 240 instances. We also compare the group of the 120 instances with $\Gamma \leq 0.1538$ and the group of the 120 instances with $\Gamma > 0.1538$. The results in Table 4 indicate that in the group with large Γ, the advantage from our ADRO approach is more significant, although again there is a big advantage from the use of ADRO in both groups of data.

	RN			D		
Performance criteria	Average from small N	Average from large N	p-value	Average from small N	Average from large N	p-value
Best	0.6609	0.5928	8.19E-06($\star$)	0.5793	0.5478	0.01629($\star$)
Worst	1.4004	1.3441	0.02950($\star$)	1.8486	1.5843	2.42E-10($\star$)
Mean	0.8607	0.8762	0.0237($\star$)	0.8859	0.9137	0.0066($\star$)
Variance	6.2193	4.5261	0.0177($\star$)	14.5446	7.7479	3.96E-05($\star$)
VaR@5%	3.8849	3.1914	0.8485	5.7217	7.2654	0.7511
CVaR@5%	3.5642	2.2200	0.1452	7.6943	3.2624	0.0596
EP	1.9487	1.3534	0.0175($\star$)	3.0520	1.6012	0.0004($\star$)
EL	5.7004	2.9975	0.0003($\star$)	14.2533	5.2626	1.77E-06($\star$)

($\star$) : significant difference between the group with small N and the group with large N at 0.05 level.

Table 3 Sensitivity analysis to the number of tasks, N

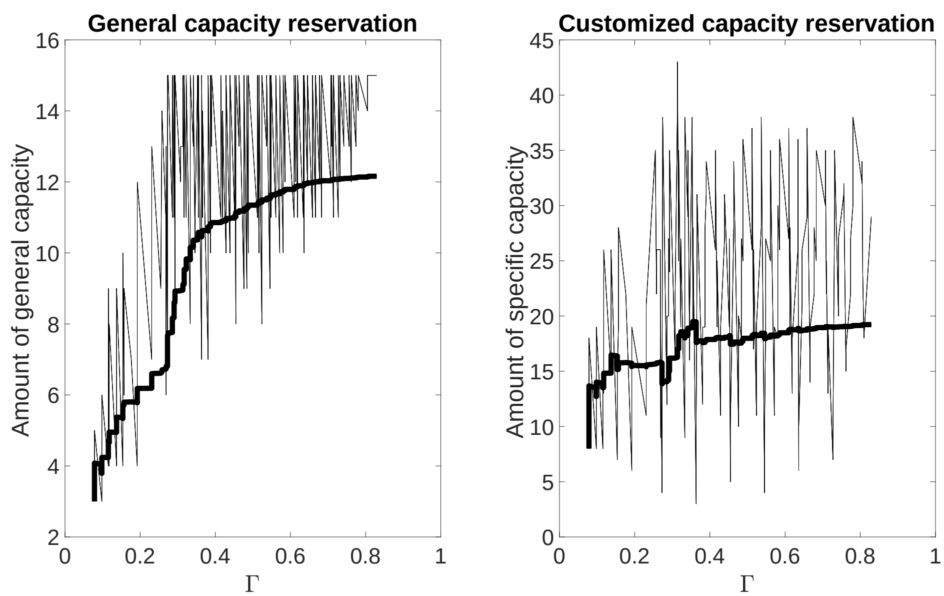
	RN			D		
Performance criteria	Average from small Γ	Average from large Γ	p-value	Average from small Γ	Average from large Γ	p-value
Best	0.6034	0.6503	0.0024($\star$)	0.5573	0.5699	0.3387
Worst	1.3247	1.4199	0.0002($\star$)	1.5610	1.8720	3.32E-14($\star$)
Mean	0.8788	0.8582	0.0027($\star$)	0.9077	0.8919	0.1265
Variance	4.1715	6.5738	0.0007($\star$)	6.4934	15.7991	1.07E-08($\star$)
VaR@5%	3.3967	3.6796	0.9379	5.0527	7.9344	0.5536
CVaR@5%	2.0485	3.7357	0.0671	2.8768	8.0799	0.0268($\star$)
EP	1.3985	1.9036	0.0442($\star$)	1.5738	3.0793	0.0003($\star$)
EL	2.7936	5.9044	2.42E-05($\star$)	4.1502	15.3657	1.50E-09($\star$)

($\star$) : significant difference between the group with small Γ and the group with large Γ at 0.05 level.

Table 4 Sensitivity analysis to the depth indicator, Γ**.**

We also study the possibility of an interaction between the impact of the number of tasks and the impact from the depth indicator. However, a two-way ANOVA for N and Γ shows no significant evidence of such an interaction.

Next, we study how the amount of capacity reservation varies with different project characteristics. We identify no clear relationship between the number of tasks and the percentage of available general or customized capacity that is reserved in an optimal solution. However, as shown in Figure 2, both types of capacity show a strong positive relationship with the network depth indicator, Γ. In networks with greater depth, there are typically more tasks in series, and therefore delays in completing tasks have more opportunity to accumulate and delay the overall project completion time significantly. For the same reason, as network depth increases, there is a growing preference for the use of general over customized capacity, because outsourcing is likely to be needed for more tasks. This explains the stronger and clearer relationship with Γ for general capacity reservation in Figure 2(a) than for customized capacity reservation in Figure 2(b).



—— : capacity reservation for instance with depth indicator Γ.
▬ : average capacity reservation for instances with depth indicator no greater than Γ.

Figure 2 **Sensitivity of capacity reservation**

4.4. Robustness

We study the robustness of the solutions delivered by Algorithm CCG to incorrect assumptions about task time durations, for example different parameter settings in the beta distribution. To do so, we compare the solutions delivered by Algorithm CCG with those obtained by assuming an exact distribution. Specifically, for any given project management instance, we assume that we have certain sampling data for the uncertain factors. On one hand, based on the sampling data, we extract the scenario, mean, variance, and joint dispersion information, and solve the solution from Algorithm CCG; we call this solution the *robust solution*. On the other hand, we use the sampling data as the exact sampling distribution and solve another solution to minimize the URI criterion under that sampling distribution; we call this solution the *sampling solution*. For both solutions, conditioning on the scenarios, mean, variance, and joint dispersion information, we solve the worst case distribution under which the worst case URI is achieved. We compare the two solutions under both worst case distributions, by analyzing their uncertain cost under the eight criteria described in Section 4.2. We conduct this comparison for the same 240 instances as in Section 4.2. The average comparison between the *robust solution* and the *sampling solution* is presented in Table 5.

	Solution	Best	Worst	Average	Variance	VaR5%	CVaR5%	EP	EL
Distribution 1	*Robust solution*	176	634	231	14907	634	634	8%	30
	Sampling solution	209	764	264	22503	764	764	8%	40
Distribution 2	*Robust solution*	181	593	220	7526	404	484	10%	16
	Sampling solution	211	709	250	11515	458	565	20%	25

Distribution 1: worst case distribution for the robust solution;
Distribution 2: worst case distribution for the sampling solution.

Table 5 Comparison with benchmarks for various projects.

From Table 5, the *robust solution* outperforms the *sampling solution* in all eight criteria. Hence, the *robust solution* is more robust in the sense that when the worst case distribution is realized, the *robust solution* has a cost that is lower than that of the *sampling solution*.

4.5. Advantage of the CCG Algorithm

In this subsection, we demonstrate the computational advantage from using the CCG algorithm. The overall problem (21)-(27) can be solved directly. However, due to the exponentially large values of I, the problem is computationally difficult to solve, whereas it can be solved using the CCG algorithm. We vary the number of uncertain factors and the number of potential scenarios for each uncertain factor, such that I is changed. We test 20 out of the 240 instances considered in Section 4.2, and report the average solution time in Table 6.

	CCG	direct solving
5 factors, each of them have 5 scenarios, $I = 5^5 = 3125$	80	3,384
7 factors, each of them have 7 scenarios, $I = 7^7 = 823,543$	1425	$> 48,000$

Table 6 **Comparison between CCG and direct solving computation time in seconds.**

Table 6 shows that Algorithm CCG can solve problems with large values of I, and indeed solution times do not increase significantly with I. However, direct solution of the problem becomes much more time consuming. For example, when $I = 823,543$, the problem still cannot be solved directly after 48,000 seconds.

5. Concluding Remarks

This work presents an ADRO model to achieve robust performance in highly uncertain projects, where there is limited availability of information about the distributions of task durations. Improved performance is achieved through more robust capacity reservation decisions at the planning stage of projects. Our model is general enough to incorporate various practical realities of project management, including both general and customized outsourced capacity, correlation between task durations, worst case distribution of delays to tasks, fast tracking of tasks, and piecewise linear crashing costs and project makespan penalties. A column and constraint generation algorithm is developed to minimize the URI of total cost, which includes capacity reservation, fast tracking, crashing, and makespan penalty costs. In part due to the decomposition property described in Remark 3, this algorithm is computationally efficient and finds optimal solutions for practical size projects. The decisions recommended by our model deliver results that provide significantly improved performance, relative to the best available benchmarks, and are also robust against incorrect distributional assumptions. In summary, our work provides project managers with a planning tool for effective risk minimization in highly uncertain projects, and provides insights about how to make better capacity reservation decisions.

Project managers who work with outsourced resources on highly uncertain projects should benefit from several insights that arise from our work. First, the optimization of capacity reservation decisions should enable a reduction in unnecessary capacity reservations, which will reduce costs for project companies. This will also help providers of outsourcing services to improve their capacity utilization and profitability. Second, our sensitivity analysis results indicate for which types of projects the ADRO planning approach is most likely to deliver greater benefits. Third, our sensitivity analysis results also indicate how network depth modulates the correct amount of capacity reservation. Fourth, our robustness study reveals the value of obtaining a more precise definition of the probability distributions of task time durations. Finally, it is possible to adjust the information set constraints parametrically. This provides an understanding of the effect of different levels of uncertainty on capacity reservations decisions, which can inform companies about the value of reducing uncertainty.

Our work also suggests several directions for future research. First, whereas we assume a piecewise linear time cost tradeoff function, there are various project management applications where this tradeoff is discrete. However, the modeling of this alternative is complicated by the intractability of project scheduling with discrete crashing options (De et al. 1997). Second, our work can be generalized to consider scope changes, for example the introduction of a new task, requested by a project owner. Third, it would be valuable to study specific project management applications where unique characteristics such as high correlation between the task durations due to substantial resource sharing may provide opportunities for delivering greater value from our work. Finally, our work considers capacity reservation in the planning process for a single project. However, for companies with multiple projects, it should be more economical to make one set of capacity reservation decisions that applies across multiple projects, and such decisions can also be modeled using ADRO. In conclusion, we hope that our work will encourage research on these topics of substantial practical value to project companies.

Acknowledgments

This work is supported by National Science Foundation grant 1548015 to the first author; and by the Summer Fellowship Program of the Fisher College of Business, The Ohio State University, to the second author.

References

Adler, P.S., A. Mandelbaum, V. Nguyen, E. Schwerer. 1995. From project to process management: An empirically-based framework for analyzing product development time. *Management Science* **41**(3) 458–484.

Artzner, P., F. Delbaen, J.M. Eber, D. Heath. 1999. Coherent measures of risk. *Mathematical Finance* **9**(3) 203–228.

Ben-Tal, A., A. Goryashko, E. Guslitzer. 2004. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming* **99**(2) 351–376.

Ben-Tal, A., D. den Hertog, A. De Waegenaere, B. Melenberg, G. Rennen. 2013. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science* **59**(2) 341–357.

Ben-Tal, A., A. Nemirovski. 1998. Robust convex optimization. *Mathematics of Operations Research* **23**(4) 769–805.

Bertsimas D, V. Goyal. 2012. On the power and limitations of affine policies in two-stage adaptive optimization. *Mathematical programming* **134**(2) 491–531.

Bertsimas, D., D.A. Iancu, P.A. Parrilo. 2010. Optimality of affine policies in multistage robust optimization. *Mathematics of Operations Research* **35**(2) 363–394.

Bertsimas, D., E. Litvinov, X.A. Sun, J, Zhao, T. Zheng. 2013. Adaptive robust optimization for the security constrained unit commitment problem. *IEEE Transactions on Power Systems* **28**(1) 52–63.

Bertsimas, D., M. Sim. 2004. The price of robustness. *Operations Research* **52**(1) 35–53.

Brown, D. B.,M. Sim. 2009. Satisficing measures for analysis of risky positions. *Management Science* **55**(1) 71–84.

Boulaksil, Y., M. Grunow, J.C. Fransoo. 2011. Capacity flexibility allocation in an outsourced supply chain with reservation. *International Journal of Production Economics* **129**(1) 111–118.

Cachon, G.P., M.A. Lariviere. 2001. Contracting to assure supply: How to share demand forecasts in a supply chain. *Management Science* **47**(5) 629–646.

Cachon, G.P., C. Terwiesch. 2006. *Matching Supply with Demand*. McGraw-Hill Irwin, New York, NY.

Cooper, R.G., S.J. Edgett, E.J. Kleinschmidt. 2000. Portfolio management for new product development: Results of an industry practices study. Retrieved January 27, 2017, `http://www.stage-gate.net/downloads/wp/wp_13.pdf`.

De, P., E.J. Dunne, J.B. Ghosh, C.E. Wells. 1997. Complexity of the discrete time-cost tradeoff problem for project networks. *Operations Research* **45**(2) 302–306.

Delage, E., Y. Ye. 2010. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research* **58**(3) 595–612.

DOD and NASA. 1962. *DOD and NASA Guide: PERT COST Systems Design*. United States Department of Defense, Washington, DC.

El Ghaoui, L., F. Oustry, H. Lebret. 1998. Robust solutions to uncertain semidefinite programs. *SIAM Journal on Optimization* **9**(1) 33–52.

Esfahani, P.M, D. Kuhn. 2017. Data-driven distributionally robust optimization using the wasserstein metric: performance guarantees and tractable reformulations. *Mathematical Programming* 1–52.

Föllmer, H., A. Schied. 2002. Convex measures of risk and trading constraints. *Finance and stochastics* **6**(4) 429–447.

Gao, R., J.K. Anton. 2018. Distributionally robust stochastic optimization with wasserstein distance. *Working Paper*.

Garey, M.R., D.S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA.

Goh, J., N.G. Hall. 2013. Total cost control in project management via satisficing. *Management Science* **59** (2013) 1354–1372.

Goh, J., M. Sim. 2010. Distributionally robust optimization and its tractable approximations. *Operations Research* **58**(4) 902–917.

Hall, N.G. 2012. Project management: Recent developments and research opportunities. *Journal of Systems Science and Systems Engineering* **21**(2) 129–143.

Hall, N.G. 2015. Further research opportunities in project management. In *Handbook on Project Management and Scheduling*, ed. C. Schwindt, J. Zimmermann. Springer, Cham, Switzerland, 945–970.

Hall, N.G. 2016. Research and teaching opportunities in project management. In *INFORMS Tutorials in Operations Research*. Published online November 4, 329–388. Retrieved January 27, 2017, `http://dx.doi.org/10.1287/educ.2016.0146`.

Hall, N.G., Z. Long, J. Qi, M. Sim. 2015. Managing underperformance risk in project portfolio selection. *Operations Research* **63**(3) 660–675.

Hall, N.G., M.E. Posner. 2001. Generating experimental data for computational testing with machine scheduling applications. *Operations Research* **49**(6) 854–865.

Herroelen, W., B. De Reyck, E. Demeulemeester. 1998. Resource-constrained project scheduling: A survey of recent developments. *Computers & Operations Research* **25**(4) 279–302.

Herroelen, W., R. Leus. 2005. Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research* **165**(2) 289–306.

Iancu, D.A., M. Sharma, M. Sviridenko. 2013. Supermodularity and affine policies in dynamic robust optimization. *Operations Research* **61**(4) 941–956.

Jiang R., Y. Guan. 2016. Data-driven chance constrained stochastic program. *Mathematical Programming* **158** 291–327.

Jiang R., J. Wang, Y. Guan. 2012. Robust unit commitment with wind power and pumped storage hydro. *IEEE Transaction on Power Systems* **27**(2) 800–810.

Jin, M., S.D. Wu. 2007. Capacity reservation contracts for high-tech industry. *European Journal of Operational Research* **176**(3) 1659–1677.

Leach, L.P. 1999. Critical chain project management improves project performance. *Project Management Journal* **30**(2) 39–51.

Mochal, T. 2006. Fast-tracking and crashing can get your project back on schedule. Retrieved January 27, 2017, http://www.techrepublic.com/article/fast-tracking-and-crashing-can-get-your-\project-back-o

Okoro, S. 2015. Why over 90 percent of all projects finish late. Retrieved January 27, 2017, http://www.projectsmart.co.uk/why-over-90-percent-of-all-projects-finish-late.php

Parkinson, C.N. 1955. *Parkinson's Law*, Economist, November 19.

Parkinson, C.N. 1958. *Parkinson's Law: The Pursuit of Progress.* John Murray, London, U.K.

Passionate Project Management. 2011. Fast-tracking versus crashing (PMP concept 19). Retrieved January 27, 2017, https://www.passionatepm.com/blog/fast-tracking-versus-crashing-pmp-concept-19

Pender, S. 2001. Managing incomplete knowledge: Why risk management is not sufficient. *International Journal of Project Management* **19**(2) 79–87.

Pinedo, M.L. 2009. *Planning and Scheduling in Manufacturing and Services.* Springer.

Pisano, G.P., R. Verganti. 2008. Which kind of collaboration is right for you? *Harvard Business Review* **86** 78–86.

Popescu I. 2007. Robust mean-covariance solutions for stochastic optimization. *Operations Research* **55**(1) 98–112.

Postek, K., A. Ben-Tal, D. Den Hertog, B. Melenberg. 2015. Exact robust counterparts of ambiguous stochastic constraints under mean and dispersion information. *Working paper.*

Prahalad, C.K., G. Hamel. 1990. The core competence of the corporation. *Harvard Business Review* **68** 79–91.

Project Management Institute. 2008. *Should You Be Teaching Project Management.* Project Management Institute, Newtown Square, PA. Retrieved January 27, 2017, http://www.mosaicprojects.com.au/PDF/Why_teach_PM.pdf.

Quinn, J.B., F.G. Hilmer. 1994. Strategic outsourcing. *Sloan Management Review* **35** 43–55.

Rockafellar, R.T., S. Uryasev. 2000. Conditional value-at-risk for general loss distributions. *Journal of Banking and Finance* **26**(7) 1443–1471.

Rockafellar, R.T., S. Uryasev. 2002. Optimization of conditional value-at-risk. *Journal of Risk* **2** 21–42.

Ruiz, C., A.J. Conejo. 2015. Robust transmission expansion planning. *European Journal of Operational Research* **242**(2) 390–401.

Securities and Exchange Commission. 2007. Form 10-K, Genvec, Inc. http://www.annualreports.com/HostedData/AnnualReportArchive/g/NASDAQ_GNVC_2007.pdf. Retrieved March 13, 2018.

Securities and Exchange Commission. 2014. GMP - Manufacturing Services Agreement by and between Variation Biotechnologies, Inc. and Paragon Bioservices, Inc. Available at: https://www.sec.gov/Archives/edgar/data/764195/000149315216007225/ex10-44.htm Retrieved March 13, 2018.

Serel, D.A., M. Dada, H. Moskowitz. 2001. Sourcing decisions with capacity reservation contracts. *European Journal of Operational Research* **131** 635–648.

Shapiro, A., A. Nemirovski. 2005. On complexity of stochastic programming problems. *Continuous optimization* Springer, 111-146.

Silver, E.A., K. Jain. 1994. Some ideas regarding reserving supplier capacity and selecting replenishment quantities in a project context. *International Journal of Production Economics* **35** 177–182.

Simon, H. A. 1955. A behavioral model of rational choice. *The Quarterly Journal of Economics* **69**(1) 99–118.

Soni, M., M. Acharya. 2015. Schedule slippage: Root causes. Retrieved January 27, 2017, http://www.projectperfect.com.au/info_schedule_slippage_root_cause_reviewed.php

Soyster, A.L. 1973. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research* **21**(5) 1154–1157.

Tully, S. 1994. You'll never guess who really makes … *Fortune* **130** 124–129.

van Dorp, J.R., M.R. Duffey. 1999. Modeling statistical dependence in risk analysis for project networks. *International Journal of Production Economics* **58**(1) 17–29.

von Neumann, J., O. Morgenstern. 1944. *Theory of games and economic behavior.* Princeton, NJ., Princeton University Press.

Vrat, P., C. Kriengkrairut. 1986. A goal programming model for project crashing with piecewise linear time-cost trade-off. *Engineering Costs and Production Economics* **10**(2) 161–172.

Wang Z., P.W. Glynn, Y. Ye. 2016. Likelihood robust optimization for data-driven problems. *Computational Management Science* **13**(2) 241–261.

Ward, S., C. Chapman. 2003. Transforming project risk management into project uncertainty management. *International Journal of Project Management* **21**(2) 97–105.

Wiesemann, W., D. Kuhn, M. Sim. 2014. Distributionally robust convex optimization. *Operations Research* **62**(6) 1358–1376.

Williams, T. 2003. The contribution of mathematical modeling to the practice of project management. *IMA Journal of Management Mathematics* **14**(1) 3–30.

Wysocki, R.K. 2014. *Effective Project Management: Traditional, Agile, Extreme*, 7th edition. Wiley, Indianapolis, IN.

Xu, H., S. Mannor. 2012. Distributionally robust markov decision processes. *Mathematics of Operations Research* **37**(2) 288–300.

Zeng, B., L. Zhao. 2013. Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operations Research Letters* **41**(5) 457–461.

Zymler, S., D. Kuhn, B. Rustem. 2013. Distributionally robust joint chance constraints with second-order moment information. *Mathematical Programming* **137** 167–198.

Proofs of Statements

EC.1. Proof of Theorem 2.

Proof. We prove that this problem is formally intractable. To do so, we first definer the following problem:

Discrete Time-Cost Tradeoff (DTCT): given a project network where each task n has at most two discrete (time, cost) processing alternatives (t_{1n}, c_{1n}) and if there is a second alternative (t_{2n}, c_{2n}), where $t_{1n} < t_{2n}$ and $c_{1n} > c_{2n}$, and a given deadline D, does there exist a feasible project schedule that meets the deadline D and has total cost less than or equal to a given value C?

De et al. (1997) prove that the recognition version of problem DTCT is unary *NP*-complete. We now establish a similar result for problem (10).

By reduction from problem DTCT. Given an arbitrary instance of problem DTCT with N tasks, we construct an instance of problem (10) as follows. Define $\gamma_{ij} = 0$, for $i, j = 1, \ldots, i \to j$; hence, no fast tracking is possible. We let $\boldsymbol{\theta} = \mathbf{0}$ such that there is no need for any general capacity, hence without loss of generality the optimal value of $\boldsymbol{x}$ in problem (10) is $\mathbf{0}$. Define $\boldsymbol{\eta} = \mathbf{1}$; hence, customized capacity is needed for crashing any task. We also define $\boldsymbol{g}_C = \mathbf{0}$; hence, the crashing of any task is free, once capacity has been reserved for it. For any task j with only one processing alternative, we let b_j denote its corresponding processing time in DTCT, and let $u_{yj} = \infty$; hence, we do not reserve customized capacity to crash task j since doing so is too expensive. Observe that the tasks with two processing alternatives in problem DTCT have a "faster" alternative 1, and a "cheaper" alternative 2. For any task j with two processing alternatives, we let $b_j = t_{2j}$, i.e., the processing time of the "cheaper" alternative; let $u_{yj} = c_{1j} - c_{2j}$, i.e., the cost difference between the two alternatives in problem DTCT; in addition, we define $q_j = 1$ and $e^C_{1j} = t_{2j} - t_{1j}$, i.e., the largest possible crashing of task j reduces its processing time to $b_j - e^C_{1j} = t_{1j}$. Therefore, since $\boldsymbol{g}_C = \mathbf{0}$, any task j with two processing alternatives in DTCT without loss of generality also has two (time, cost) pairs in problem (10): $(t_{1j}, c_{1j} - c_{2j})$ if customized capacity is reserved, and $(t_{2j}, 0)$ otherwise. Finally, we define $l_1 = D$, $l_2 = D + \sigma$, $d_1 = 0$, and $d_2 = +\infty$, where $\sigma > 0$ is an arbitrarily small constant; hence, the completion time penalty is zero up to time D, and infinite thereafter. We let $C' = C - \sum_{j=1}^{N} c_{2j}$ denote a threshold cost for the constructed instance of problem (10).

We prove that there exists a feasible schedule for the constructed instance of problem (10) with cost less than or equal to C', if and only if there exists a "yes" answer to problem DTCT.

($\Rightarrow$) Consider any feasible schedule that provides a "yes" answer to problem DTCT. We construct a solution to problem (10) as follows. For any task with only one processing alternative, we do not reserve the customized capacity. For any task with two processing alternatives, we reserve (respectively, do not reserve) capacity in problem (10) if the faster (resp., cheaper) alternative is used for the corresponding task in problem DTCT. By definition of the instance of problem (10), this solution to problem (10) completes the project at the same time as the feasible schedule to problem DTCT, i.e., by time D, and thus has zero completion time penalty. Now, each task j with reserved capacity in problem (10) has capacity reservation cost of $(c_{1j} - c_{2j})$ and linear crashing cost of $g_{1j}^C = 0$. Hence, task j has total cost $(c_{1j} - c_{2j})$ in problem (10), compared to a cost of c_{1j} in problem DTCT. Further, each task j without reserved capacity in problem (10) has zero capacity reservation and linear crashing cost, hence task j has total cost 0, compared to total cost c_{2j} in problem DTCT. Combining these two observations, each task j costs exactly c_{2j} less in problem (10) than in problem DCTC, for $j = 1, \ldots, N$. Therefore, the total cost of the schedule in problem (10) is $C - \sum_{j=1}^{N} c_{2j} = C'$.

($\Leftarrow$) Given a feasible schedule for problem (10) with cost less than or equal to C', we prove that the answer to problem DTCT is "yes". Since $l_2 = D + \sigma$ and $d_2 = +\infty$, any schedule for problem (10) with cost less than or equal to C' must complete by time D. From the discussion in the previous paragraph, the corresponding schedule for problem DTCT also completes by time D, and hence is feasible. By assumption, the total cost in problem (10) is less than or equal to C'. Moreover, from the discussion in the previous paragraph, each task has a total cost in problem DTCT that is greater by exactly t_{2j} than it is in problem (10). Hence, the cost of the solution in problem DTCT is less than or equal to $C' + \sum_{j=1}^{N} c_{2j} = C$, and the schedule provides a "yes" answer. Q.E.D.

EC.2. Proof of Proposition 1

Proof. Note that $\alpha^* \leq \alpha^*_\epsilon$ follows from the feasible set of Problem (18) being a subset of that from Problem (17). To prove $\alpha^*_\epsilon \leq \alpha^* + \epsilon$, consider any feasible $(\boldsymbol{x}, \boldsymbol{y}, \alpha)$ in Problem (17), we show the feasibility of $(\boldsymbol{x}, \boldsymbol{y}, \alpha + \epsilon)$ in Problem (18). For any $\mathbb{P} \in \mathcal{P}$,

$$\begin{aligned}
&\mathbb{E}_{\mathbb{P}}\left[\max_{m\in[M]}\left\{a_m h\left(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{\delta}}\right) - a_m\tau + b_m(\alpha+\epsilon)\right\}\right] \\
&= (\alpha+\epsilon)\mathbb{E}_{\mathbb{P}}\left[\max_{m\in[M]}\left\{a_m \frac{h\left(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{\delta}}\right)-\tau}{\alpha+\epsilon} + b_m\right\}\right] \\
&= (\alpha+\epsilon)\mathbb{E}_{\mathbb{P}}\left[\max_{m\in[M]}\left\{a_m \left(\frac{\alpha}{\alpha+\epsilon}\frac{h\left(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{\delta}}\right)-\tau}{\alpha} + \frac{\epsilon}{\alpha+\epsilon}\cdot 0\right) + b_m\right\}\right] \\
&\leq (\alpha+\epsilon)\mathbb{E}_{\mathbb{P}}\left[\frac{\alpha}{\alpha+\epsilon}\max_{m\in[M]}\left\{a_m\left(\frac{h\left(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{\delta}}\right)-\tau}{\alpha}\right) + b_m\right\} + \frac{\epsilon}{\alpha+\epsilon}\max_{m\in[M]}\left\{a_m\cdot 0 + b_m\right\}\right] \qquad \text{(EC.1)} \\
&= \alpha\mathbb{E}_{\mathbb{P}}\left[\max_{m\in[M]}\left\{a_m\left(\frac{h\left(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{\delta}}\right)-\tau}{\alpha}\right) + b_m\right\}\right] \\
&\leq \alpha\sup_{\mathbb{P}\in\mathcal{P}}\mathbb{E}_{\mathbb{P}}\left[\max_{m\in[M]}\left\{a_m\left(\frac{h\left(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{\delta}}\right)-\tau}{\alpha}\right) + b_m\right\}\right] \\
&\leq 0,
\end{aligned}$$

where the first inequality is due to the convexity of the piecewise-linear utility, the last equality holds since the utility function is normalized by $u(0) = 0$, and the last inequality follows from the feasibility of $(\boldsymbol{x}, \boldsymbol{y}, \alpha)$ in Problem (17). Taking the supremum over all $\mathbb{P} \in \mathcal{P}$ for the left hand side of Inequality (EC.1), we conclude that $(\boldsymbol{x}, \boldsymbol{y}, \alpha + \epsilon)$ is feasible in Problem (18). Q.E.D.

EC.3. Proof of Lemma 1.

Proof. Consider $\sup_{\mathbb{P}\in\mathcal{P}} \mathbb{E}_{\mathbb{P}}\left[\max_{m\in[M]}\left\{a_m h\left(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{\delta}}\right) - a_m\tau + b_m\alpha\right\}\right]$ as a primal optimization problem where the decision variables are $\mathbb{P}$. Observe that since we assume $\tilde{\boldsymbol{\delta}}$ follows a discrete distribution, the decision variables $\mathbb{P}$ is actually a finite dimensional vector. Hence, we have a finite dimensional linear programming problem. Moreover, let $\boldsymbol{\delta}^i$, $\omega_j^{S,k,i}$, $\omega_j^{C,i}$ be defined as in the lemma statement. Then

$$
\begin{aligned}
&\sup_{\mathbb{P}\in\mathcal{P}} \mathbb{E}_{\mathbb{P}}\left[\max_{m\in[M]}\left\{a_m h\left(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{\delta}}\right)-a_m\tau+b_m\alpha\right\}\right]\\
= \ \sup \quad & \mathbb{E}_{\mathbb{P}}\left[\max_{m\in[M]}\left\{a_m h\left(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{\delta}}\right)-a_m\tau+b_m\alpha\right\}\right]\\
\text{s.t.} \quad & \mathbb{P}\left(\tilde{\boldsymbol{\delta}}\in\mathcal{Q}\right)=1\\
& \mathbb{E}_{\mathbb{P}}\left[\chi_j^{S,k}\left(\tilde{\delta}_k\right)\right]\le\sigma_j^{S,k},\ k\in[K],\ j\in[J_{S,k}]\\
& \mathbb{E}_{\mathbb{P}}\left[\chi_j^{C}\left({\boldsymbol{\xi}_j}^T\tilde{\boldsymbol{\delta}}+\epsilon_j\right)\right]\le\sigma_j^{C},\ j\in[J_C]\\
= \min \quad & \tau_0+\sum_{k=1}^{K}{\boldsymbol{\sigma}^{S,k}}^T\boldsymbol{\tau}^{S,k}+{\boldsymbol{\sigma}^{C}}^T\boldsymbol{\tau}^{C}\\
\text{s.t.} \quad & \tau_0+\sum_{k=1}^{K}\sum_{j=1}^{J_{S,k}}\omega_j^{S,k,i}\tau_j^{S,k}+\sum_{j=1}^{J_C}\omega_j^{C,i}\tau_j^{C}\ge\max_{m\in[M]}\left\{a_m h\left(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\delta}^i\right)-a_m\tau+b_m\alpha\right\},\quad\forall i\in[I]\\
& \boldsymbol{\tau}^{S,k},\ \boldsymbol{\tau}^{C}\ge\boldsymbol{0},\ k\in[K]\\
= \min \quad & \tau_0+\sum_{k=1}^{K}{\boldsymbol{\sigma}^{S,k}}^T\boldsymbol{\tau}^{S,k}+{\boldsymbol{\sigma}^{C}}^T\boldsymbol{\tau}^{C}\\
\text{s.t.} \quad & \tau_0+\sum_{k=1}^{K}\sum_{j=1}^{J_{S,k}}\omega_j^{S,k,i}\tau_j^{S,k}+\sum_{j=1}^{J_C}\omega_j^{C,i}\tau_j^{C}\ge a_m h\left(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\delta}^i\right)-a_m\tau+b_m\alpha,\quad\forall i\in[I],\ m\in[M]\\
& \boldsymbol{\tau}^{C},\ \boldsymbol{\tau}^{S,k}\ge\boldsymbol{0},\ k\in[K].
\end{aligned}
\tag{EC.2}
$$

The second equality follows from strong duality, which is due to the finite dimension of the primal problem.

Following the definition of the function h in (12), given any $i\in[I]$, $m\in[M]$, the constraint

$$
\tau_0+\sum_{k=1}^{K}\sum_{j=1}^{J_{S,k}}\omega_j^{S,k,i}\tau_j^{S,k}+\sum_{j=1}^{J_C}\omega_j^{C,i}\tau_j^{C}\ge a_m h\left(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\delta}^i\right)-a_m\tau+b_m\alpha
$$

can be rewritten as

$$
\tau_0+\sum_{k=1}^{K}\sum_{j=1}^{J_{S,k}}\omega_j^{S,k,i}\tau_j^{S,k}+\sum_{j=1}^{J_C}\omega_j^{C,i}\tau_j^{C}\ge a_m\left(\boldsymbol{u}_x^T\boldsymbol{x}+\boldsymbol{u}_y^T\boldsymbol{y}+\boldsymbol{g}_F^T\boldsymbol{w}_F^i+\boldsymbol{g}_C^T\boldsymbol{w}_C^i+\boldsymbol{d}^T\boldsymbol{v}^i\right)-a_m\tau+b_m\alpha,
$$

$$
(\boldsymbol{w}_F^i,\boldsymbol{w}_C^i,\boldsymbol{v}^i,\boldsymbol{c}^i)\in\mathcal{S}(\boldsymbol{z}^0+\boldsymbol{Z}\boldsymbol{\delta}^i,\boldsymbol{x},\boldsymbol{y}).
$$

Q.E.D.

EC.4. Proof of Theorem 3.

Proof. The equivalence between problems (18) and (21)–(27) follows immediately from Lemma 1. Moreover, given any $\boldsymbol{\delta}$, the set $(\boldsymbol{w}_F^i, \boldsymbol{w}_C^i, \boldsymbol{v}^i, \boldsymbol{c}^i) \in \mathcal{S}(\boldsymbol{z}^0 + \boldsymbol{Z}\boldsymbol{\delta}^i, \boldsymbol{x}, \boldsymbol{y})$ defines a polyhedron for $(\boldsymbol{w}_F^i, \boldsymbol{w}_C^i, \boldsymbol{v}^i, \boldsymbol{c}^i, \boldsymbol{x}, \boldsymbol{y})$, and hence problem (21)–(27) is a mixed-integer linear programming problem. Q.E.D.

EC.5. Proof of Lemma 2.

Proof. Given $m \in [M]$, the constraints are feasible if and only if

$$\max_{\boldsymbol{\delta}\in\mathcal{Q}} \left\{ -\sum_{k=1}^{K}\sum_{j=1}^{J_{S,k}} \tau_j^{S,k}\chi_j^{S,k}(\delta_k) - \sum_{j=1}^{J_C} \tau_j^C \chi_j^C\left(\boldsymbol{\xi}_j^T\boldsymbol{\delta} + \epsilon_j\right) + a_m c_2^*(\boldsymbol{\delta}, \boldsymbol{x}, \boldsymbol{y}) \right\} \le \tau_0 + a_m\tau - b_m\alpha - a_m\boldsymbol{u}_x^T\boldsymbol{x} - a_m\boldsymbol{u}_y^T\boldsymbol{y}, \tag{EC.3}$$

where $c_2^*(\cdot,\cdot,\cdot)$ is a mapping from the uncertain realization and first stage decisions to the lowest cost at the operational stage. In particular, c_2^* is defined and can be reformulated as follows.

$$\begin{aligned}
c_2^*(\boldsymbol{\delta}, \boldsymbol{x}, \boldsymbol{y}) = \ & \min \ \boldsymbol{g}_F^T\boldsymbol{w}_F + \boldsymbol{g}_C^T\boldsymbol{w}_C + \boldsymbol{d}^T\boldsymbol{v} \\
& \ \text{s.t.} \ (\boldsymbol{w}_F, \boldsymbol{w}_C, \boldsymbol{v}, \boldsymbol{c}) \in \mathcal{S}(\boldsymbol{z}^0 + \boldsymbol{Z}\boldsymbol{\delta}, \boldsymbol{x}, \boldsymbol{y}) \\
= \ & \min \ \boldsymbol{g}_F^T\boldsymbol{w}_F + \boldsymbol{g}_C^T\boldsymbol{w}_C + \boldsymbol{d}^T\boldsymbol{v} \\
& \ \text{s.t.} \ \boldsymbol{A}_F\boldsymbol{w}_F + \boldsymbol{A}_C\boldsymbol{w}_C + \boldsymbol{A}_v\boldsymbol{v} + \boldsymbol{A}_c\boldsymbol{c} \ge \boldsymbol{A}_x\boldsymbol{x} + \boldsymbol{A}_y\boldsymbol{y} + \boldsymbol{A}_z(\boldsymbol{z}^0 + \boldsymbol{Z}\boldsymbol{\delta}) + \boldsymbol{b}^o \\
& \qquad \boldsymbol{w}_F \ge \boldsymbol{0}, \boldsymbol{w}_C \ge \boldsymbol{0}, \boldsymbol{v} \ge \boldsymbol{0}, \boldsymbol{c} \ge \boldsymbol{0} \\
= \ & \max \ \left(\boldsymbol{A}_x\boldsymbol{x} + \boldsymbol{A}_y\boldsymbol{y} + \boldsymbol{A}_z(\boldsymbol{z}^0 + \boldsymbol{Z}\boldsymbol{\delta}) + \boldsymbol{b}^o\right)^T\boldsymbol{p} \\
& \ \text{s.t.} \ \boldsymbol{p}^T\left(\boldsymbol{A}_F \ \boldsymbol{A}_C \ \boldsymbol{A}_v \ \boldsymbol{A}_c\right) \le \left(\boldsymbol{g}_F^T \ \boldsymbol{g}_C^T \ \boldsymbol{d}^T \ \boldsymbol{0}^T\right) \\
& \qquad \boldsymbol{p} \ge \boldsymbol{0} \\
= \ & \max_{\boldsymbol{p}\in P} \ \left(\boldsymbol{A}_x\boldsymbol{x} + \boldsymbol{A}_y\boldsymbol{y} + \boldsymbol{A}_z(\boldsymbol{z}^0 + \boldsymbol{Z}\boldsymbol{\delta}) + \boldsymbol{b}^o\right)^T\boldsymbol{p},
\end{aligned} \tag{EC.4}$$

where the third equality follows from strong duality. Hence, we can reformulate the LHS of constraint (EC.3) to $q_m^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\tau}^C, \boldsymbol{\tau}^{S,1}, \ldots, \boldsymbol{\tau}^{S,K})$, as defined in problem (29). Further, constraint (EC.3) is feasible if and only if $q_m^*(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\tau}^C, \boldsymbol{\tau}^{S,1}, \ldots, \boldsymbol{\tau}^{S,K}) \le \tau_0 + a_m\tau - b_m\alpha - a_m\boldsymbol{u}_x^T\boldsymbol{x} - a_m\boldsymbol{u}_y^T\boldsymbol{y}$.

Consider the case where $q_m^*(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\tau}^C,\boldsymbol{\tau}^{S,1},\ldots,\boldsymbol{\tau}^{S,K}) > \tau_0 + a_m\tau - b_m\alpha - a_m\boldsymbol{u}_x^T\boldsymbol{x} - a_m\boldsymbol{u}_y^T\boldsymbol{y}$ for some $m\in[M]$. There must exist $m_*\in[M], i^*\in[I]$ such that (31) holds. Consider any such m_* and i^*. Observe that

$$
\begin{aligned}
&\tau_0 + a_{m_*}\tau - b_{m_*}\alpha - a_{m_*}\boldsymbol{u}_x^T\boldsymbol{x} - a_{m_*}\boldsymbol{u}_y^T\boldsymbol{y}\\
&< \max_{\boldsymbol{p}\in P} q_{m_*}\left(\boldsymbol{p},\boldsymbol{\delta}^{i^*},\boldsymbol{x},\boldsymbol{y},\boldsymbol{\tau}^C,\boldsymbol{\tau}^{S,1},\ldots,\boldsymbol{\tau}^{S,K}\right)\\
&= -\sum_{k=1}^{K}\sum_{j=1}^{J_{S,k}}\tau_j^{S,k}\chi_j^{S,k}\left(\delta_k^{i^*}\right) - \sum_{j=1}^{J_C}\tau_j^C\chi_j^C\left(\boldsymbol{\xi}_j^T\boldsymbol{\delta}^{i^*}+\epsilon_j\right)\\
&\qquad + a_{m_*}\max_{\boldsymbol{p}\in P}\left\{\left(\boldsymbol{A}_x\boldsymbol{x}+\boldsymbol{A}_y\boldsymbol{y}+\boldsymbol{A}_z\left(\boldsymbol{z}^0+\boldsymbol{Z}\boldsymbol{\delta}^{i^*}\right)+\boldsymbol{b}^o\right)^T\boldsymbol{p}\right\}\\
&= -\sum_{k=1}^{K}\sum_{j=1}^{J_{S,k}}\tau_j^{S,k}\chi_j^{S,k}\left(\delta_k^{i^*}\right) - \sum_{j=1}^{J_C}\tau_j^C\chi_j^C\left(\boldsymbol{\xi}_j^T\boldsymbol{\delta}^{i^*}+\epsilon_j\right) + a_{m_*}c_2^*(\boldsymbol{\delta}^{i^*},\boldsymbol{x},\boldsymbol{y})\\
&= -\sum_{k=1}^{K}\sum_{j=1}^{J_{S,k}}\tau_j^{S,k}\chi_j^{S,k}\left(\delta_k^{i^*}\right) - \sum_{j=1}^{J_C}\tau_j^C\chi_j^C\left(\boldsymbol{\xi}_j^T\boldsymbol{\delta}^{i^*}+\epsilon_j\right)\\
&\qquad + a_{m_*}\min_{(\boldsymbol{w}_F^{i^*},\boldsymbol{w}_C^{i^*},\boldsymbol{v}^{i^*},\boldsymbol{c}^{i^*})\in\mathcal{S}(\boldsymbol{z}^0+\boldsymbol{Z}\boldsymbol{\delta}^{i^*},\boldsymbol{x},\boldsymbol{y})}\left\{\boldsymbol{g}_F^T\boldsymbol{w}_F^{i^*}+\boldsymbol{g}_C^T\boldsymbol{w}_C^{i^*}+\boldsymbol{d}^T\boldsymbol{v}^{i^*}\right\},
\end{aligned}
$$

where the last two equalities follow from (EC.4). Hence, we cannot find $(\boldsymbol{w}_F^{i^*},\boldsymbol{w}_C^{i^*},\boldsymbol{v}^{i^*},\boldsymbol{c}^{i^*})\in\mathcal{S}(\boldsymbol{z}^0+\boldsymbol{Z}\boldsymbol{\delta}^{i^*},\boldsymbol{x},\boldsymbol{y})$ with $\tau_0 + a_{m_*}\tau - b_{m_*}\alpha - a_{m_*}\boldsymbol{u}_x^T\boldsymbol{x} - a_{m_*}\boldsymbol{u}_y^T\boldsymbol{y} \geq a_{m_*}\boldsymbol{g}_F^T\boldsymbol{w}_F^{i^*} + a_{m_*}\boldsymbol{g}_C^T\boldsymbol{w}_C^{i^*} + a_{m_*}\boldsymbol{d}^T\boldsymbol{v}^{i^*} - \sum_{k=1}^K\sum_{j=1}^{J_{S,k}}\tau_j^{S,k}\chi_j^{S,K}\left(\delta_k^{i^*}\right) - \sum_{j=1}^{J_C}\tau_j^C\chi_j^C\left(\boldsymbol{\xi}_j^T\boldsymbol{\delta}^{i^*}+\epsilon_j\right)$. Q.E.D.

EC.6. Proof of Lemma 3.

Proof. Given any feasible solution $(\boldsymbol{p},\boldsymbol{\delta})$ to problem (29), we construct a feasible solution to problem (32) with the same $\boldsymbol{p}$, $\phi_{kl}=1$ if $\delta_k=\delta_{kl}$ and 0 otherwise, $\zeta_{kl}=\delta_{kl}\boldsymbol{b}_k\boldsymbol{p}$ if $\phi_{kl}=1$ and 0 otherwise, $\Omega_j=\chi_j^C(\boldsymbol{\xi}_j^T\boldsymbol{\delta}+\epsilon_j)$. In addition, the objective value of problem (29) for $(\boldsymbol{p},\boldsymbol{\delta})$ is identical to the objective value of problem (32) for the corresponding $(\boldsymbol{p},\boldsymbol{\phi},\boldsymbol{\zeta},\boldsymbol{\Omega})$.

On the other hand, given any feasible solution $(\boldsymbol{p},\boldsymbol{\phi},\boldsymbol{\zeta},\boldsymbol{\Omega})$ for problem (32), it follows that $(\boldsymbol{p},\boldsymbol{\delta})$ with $\delta_k=\sum_{l=1}^{L_k}\phi_{kl}\delta_{kl}$ is also a feasible solution for problem (29). In addition, the objective value of problem (32) for $(\boldsymbol{p},\boldsymbol{\phi},\boldsymbol{\zeta},\boldsymbol{\Omega})$ is no greater than the objective value of problem (29) for $(\boldsymbol{p},\boldsymbol{\delta})$. Q.E.D.

EC.7. Proof of Theorem 4.

Proof. The result follows directly from Lemmas 2 and 3. Q.E.D.