Distributionally Robust Project Crashing with Partial or No Correlation Information

Selin Damla Ahipasaoglu* Karthik Natarajan† Dongjian Shi‡

January 27, 2019

Abstract

Crashing is shortening the project makespan by reducing activity times in a project network by allocating resources to them. Activity durations are often uncertain and an exact probability distribution itself might be ambiguous. We study a class of distributionally robust project crashing problems where the objective is to optimize the first two marginal moments (means and standard deviations) of the activity durations to minimize the worst-case expected makespan. Under partial correlation information and no correlation information, the problem is solvable in polynomial time as a semidefinite program and a second-order cone program, respectively. However solving semidefinite programs is challenging for large project networks. We exploit the structure of the distributionally robust formulation to reformulate a convex-concave saddle point problem over the first two marginal moment variables and the arc criticality index variables. We then use a projection and contraction algorithm for monotone variational inequalities in conjunction with a gradient method to solve the saddle point problem enabling us to tackle large instances. Numerical results indicate that a manager who is faced with ambiguity in the distribution of activity durations has a greater incentive to invest resources in decreasing the variations rather than the means of the activity durations.

Keywords: distributionally robust optimization, project networks, makespan, moments, saddle point, projection and contraction algorithm

1 Introduction

Projects are ubiquitous, be it in th e construction industry or the software development industry. Formally, a project is defined by a set of activities that must be completed with given precedence constraints. In a project, an

*Engineering Systems and Design, Singapore University of Technology and Design, 8 Somapah Road, Singapore 487372. Email: ahipasaoglu@sutd.edu.sg

†Engineering Systems and Design, Singapore University of Technology and Design, 8 Somapah Road, Singapore 487372. Email: karthik_natarajan@sutd.edu.sg

‡Engineering Systems and Design, Singapore University of Technology and Design, 8 Somapah Road, Singapore 487372. Email: dongjian_shi@163.com

activity is a task that must be performed and an event is a milestone marking the start of one or more activities. Before an activity begins, all of its predecessor activities must be completed. Such a project is represented by an activity-on-arc network $\mathcal{G}(\mathcal{V}, \mathcal{A})$, where $\mathcal{V} = \{1, 2, \ldots, n\}$ is the set of nodes denoting the events, and $\mathcal{A} \subseteq \{(i, j) : i, j \in \mathcal{V}\}$ is the set of arcs denoting the activities (see [20]). We let $m = |\mathcal{A}|$ denote the number of arcs in the network. The network $\mathcal{G}(\mathcal{V}, \mathcal{A})$ is directed and acyclic where we use node 1 and node n to represent the start and the end of the project respectively. Let the arc length t_{ij} denote the duration of activity (i, j). The completion time or the makespan of the project is equal to the length of the critical or the longest path of the project network from node 1 to node n. This problem is formulated as:

$$Z(\boldsymbol{t}) = \max_{\boldsymbol{x} \in \mathcal{X} \cap \{0,1\}^m} \sum_{(i,j) \in \mathcal{A}} t_{ij} x_{ij}, \tag{1.1}$$

where

$$\mathcal{X} = \left\{ \boldsymbol{x} : \sum_{j:(i,j) \in \mathcal{A}} x_{ij} - \sum_{j:(j,i) \in \mathcal{A}} x_{ji} = \begin{cases} 1, & i = 1 \\ 0, & i = 2, 3, \ldots, n-1 \\ -1, & i = n \end{cases}, \quad x_{ij} \in [0, 1], \forall (i, j) \in \mathcal{A} \right\}. \tag{1.2}$$

Project crashing is a method for shortening the project makespan by reducing the time of one or more of the project activities to less than its normal activity time. To reduce the duration of an activity, the project manager might assign more resources to it which implies additional costs. Such a resource allocation may include using more efficient equipment or hiring more workers. In such a situation it is important to model the tradeoff between the makespan and the crashing cost so as to identify the specific activities to crash and the corresponding amounts by which to crash them. Early work on the deterministic project crashing problem (PCP) dates back to the 1960s (see [35, 24]) where parametric network flow methods were developed to solve the problem. One intuitive method is to find the critical path of the project, and then crash one or more activities on the critical path. However this approach is known to be sub-optimal in general, since the original critical path may no longer be critical after a while. The problem of minimizing the project makespan with a given cost budget is formulated as the following optimization problem (see [35, 24]):

$$\begin{aligned} \min_{\boldsymbol{t}} \quad & Z(\boldsymbol{t}) \\ \text{s.t.} \quad & \sum_{(i,j) \in \mathcal{A}} c_{ij}(t_{ij}) \le B, \\ & \underline{t}_{ij} \le t_{ij} \le \bar{t}_{ij}, \qquad \forall (i, j) \in \mathcal{A}, \end{aligned} \tag{1.3}$$

where B is the cost budget, $\bar{t}_{ij}$ is the original duration of activity (i, j), $\underline{t}_{ij}$ is the minimal value of the duration of activity (i, j) that can be achieved by crashing, and $c_{ij}(t_{ij})$ is the cost of crashing which is a decreasing function of the activity duration t_{ij}. Since the longest path problem on a directed acyclic graph is solvable as a linear program by optimizing over the set $\mathcal{X}$ directly, we can use linear programming duality to reformulate the project crashing problem (1.3) as:

$$
\begin{array}{lll}
\min\limits_{\boldsymbol{t},\boldsymbol{y}} & y_n - y_1, & \\
\text{s.t.} & y_j - y_i \geq t_{ij}, & \forall (i,j) \in \mathcal{A}, \\
& \sum\limits_{(i,j)\in\mathcal{A}} c_{ij}(t_{ij}) \leq B, & \\
& \underline{t}_{ij} \leq t_{ij} \leq \bar{t}_{ij}, & \forall (i,j) \in \mathcal{A}.
\end{array}
\tag{1.4}
$$

When the cost functions are linear or piecewise linear convex, the deterministic PCP (1.4) can be solved as a linear program. For nonlinear convex differentiable cost functions, an algorithm based on piecewise linear approximations was proposed by [39] to solve the project crashing problem. When the cost function is nonlinear and concave, [21] proposed a globally convergent branch and bound algorithm to solve the problem of minimizing the total cost and when the cost function is discrete, the project crashing problem has shown to be NP-hard (see [16]). While the formulations (1.3)-(1.4) are the typical time-cost tradeoff formulations that have been studied in the literature, it is possible to extend the model to include other resource constraints that might jointly affect the activity durations. For example, when the cost functions are linear or piecewise linear convex, additional linear constraints on the decision vector $\boldsymbol{t}$ can be easily incorporated while still solving a linear program. In the methodology proposed in this paper, our only key requirement is that it is possible to efficiently compute projections on to the set of feasible crashing decision variables. In the next section, we provide a literature review on project crashing problems with uncertain activity durations.

Notations

Throughout the paper, we use bold letters to denote vectors and matrices (such as $\boldsymbol{x}$, $\boldsymbol{W}$, $\boldsymbol{\rho}$), and standard letters to denote scalars (such as x, W, ρ). A vector of dimension n with all entries equal to 1 is denoted by $\mathbf{1}_n$. The unit simplex is denoted by $\boldsymbol{\Delta}_{n-1} = \{\boldsymbol{x} : \mathbf{1}_n^T\boldsymbol{x} = 1, \boldsymbol{x} \geq 0\}$. We suppress the subscript when the dimension is clear. We use the tilde notation to denote a random variable or random vector (such as $\tilde{r}$, $\tilde{\boldsymbol{r}}$). The number of elements in the set $\mathcal{A}$ is denoted by $|\mathcal{A}|$. The set of nonnegative integers is denoted by $\mathbb{Z}^+$ and the sets of n dimensional vectors whose entries are all nonnegative and strictly positive are denoted by $\Re_n^+$ and $\Re_n^{++}$ respectively. The sets of all symmetric real $n \times n$ positive semidefinite matrices and positive definite matrices are denoted by $\mathcal{S}_n^+$ and $\mathcal{S}_n^{++}$ respectively. For a positive semidefinite matrix $\boldsymbol{X}$, we use $\boldsymbol{X}^{1/2}$ to denote the unique positive semidefinite square root of the matrix such that $\boldsymbol{X}^{1/2}\boldsymbol{X}^{1/2} = \boldsymbol{X}$. For a square matrix $\boldsymbol{X}$, $\boldsymbol{X}^\dagger$ denotes the its unique Moore-Penrose pseudoinverse. We use $\text{diag}(\boldsymbol{X})$ to denote a vector formed by the diagonal elements of a matrix $\boldsymbol{X}$ and $\text{Diag}(\boldsymbol{x})$ to denote a diagonal matrix whose diagonal elements are the entries of $\boldsymbol{x}$. We denote a finite set of alternatives as $[k] = \{1, 2, ..., k\}$.

2 Literature Review

Our literature review in this section focuses on the project crashing problem with uncertain activity durations. Our goal is to highlight some of the main approaches to solve the project crashing problem before discussing the contributions of this paper.

2.1 Stochastic Project Crashing

In stochastic projects, the activity durations are typically modeled as random variables with a specific probability distribution such as the normal, uniform, exponential or beta distribution. When the probability distribution of the activity durations is known, a popular performance measure is the expected project makespan which is defined as follows:

$$\mathbb{E}_{\theta(\boldsymbol{\lambda})}(Z(\tilde{\boldsymbol{t}})). \tag{2.1}$$

In (2.1), the activity duration vector $\tilde{\boldsymbol{t}}$ is random with a probability distribution denoted by $\theta(\boldsymbol{\lambda})$ where $\boldsymbol{\lambda}$ is assumed to be a finite dimensional vector that parameterizes the distribution and can be modified by the project manager. For example in the multivariate normal case, $\boldsymbol{\lambda}$ will incorporate information on the mean and the covariance matrix of the activity durations. The stochastic PCP that minimizes the expected makespan is formulated as:

$$\min_{\boldsymbol{\lambda}\in\boldsymbol{\Lambda}} \mathbb{E}\left(Z(\tilde{\boldsymbol{t}})\right), \tag{2.2}$$

where $\boldsymbol{\Lambda}$ is the possible set of values from which $\boldsymbol{\lambda}$ can be chosen. For a fixed $\boldsymbol{\lambda}$, computing the expected project makespan unlike the deterministic makespan is a hard problem. [30] showed that computing the expected project makespan is NP-hard when the activity durations are independent discrete random variables. Even with simple distributions such as the multivariate normal distribution, the expected makespan does not have a simple expression and the standard approach is to use Monte Carlo simulation methods to estimate the expected makespan (see [57, 9]). Bounds ([25, 45]) and approximations ([41]) have been proposed for the expected makespan. For example, by replacing the activity times with the mean durations and computing the deterministic longest path, we obtain a lower bound on the expected makespan due to the convexity of the makespan objective. Equality holds if and only if there is a path that is the longest with probability 1, but this condition is rarely satisfied in applications.

To solve the stochastic project crashing problem in (2.2), heuristics and simulation-based optimization methods have been proposed. [37] developed a heuristic approach to minimize the quantile of the makespan by using a surrogate deterministic objective function. Other heuristics for the project crashing problem have also been developed by [44]. Stochastic gradient methods for minimizing the expected makespan have been developed in this context (see [8, 23]). Another popular approach is to use the sample average approximation (SAA) method to minimize the expected makespan (see [50, 54, 36]).

2.2 Distributionally Robust Project Crashing

Distributionally robust optimization is a more recent approach that has been used to address the project crashing problem. Under this model, the uncertain activity durations are assumed to be random variables but the probability distribution of the random variables is itself ambiguous and lies in a set of distributions. Assume that $\Theta(\boldsymbol{\omega})$ is a set of probability distributions where $\boldsymbol{\omega}$ is a finite dimensional parameter vector that parameterizes this set and can be modified by the project manager. The performance measure in this case is the worst-case expected project makespan which is defined as follows:

$$\max_{\theta \in \Theta(\boldsymbol{\omega})} \mathbb{E}_{\theta}(Z(\tilde{\boldsymbol{t}})). \tag{2.3}$$

The distributionally robust PCP that minimizes the expected makespan is then formulated as:

$$\min_{\boldsymbol{\omega} \in \boldsymbol{\Omega}} \max_{\theta \in \Theta(\boldsymbol{\omega})} \mathbb{E}_{\theta}(Z(\tilde{\boldsymbol{t}})), \tag{2.4}$$

where $\boldsymbol{\Omega}$ is the possible set of values from which $\boldsymbol{\omega}$ can be chosen. [43] studied the problem of computing the worst-case expected makespan under the assumption that the marginal distributions of the random activity durations are known but the joint distribution of the activity durations is unknown. Under this assumption, they showed that the worst-case expected makespan can be computed by solving a convex optimization problem. [7] extended this bound to the case where the support for each activity duration is known and up to the first two marginal moments (mean and standard deviation) of the random activity duration are provided. [4, 5] extended this result to the case where general higher order univariate moment information is known and developed a semidefinite program to compute the worst-case expected makespan. [42] applied the dual of the formulation to solve an appointment scheduling problem where the objective is to choose service times to minimize the worst-case expected waiting time and overtime costs as a second-order cone program. Under the assumption that the mean, standard deviation and correlation matrix of the activity durations are known, [48] developed a completely positive programming reformulation for the worst-case expected makespan. While this problem is NP-hard, semidefinite relaxations can be used to find weaker upper bounds on the worst-case expected makespan. [38] developed a dual copositive formulation for the appointment scheduling problem where the objective is to choose service times to minimize the worst-case expected waiting time and overtime costs given correlation information. Since this problem is NP-hard, they developed a tractable semidefinite relaxation for this problem. [47] recently showed that the complexity of computing such bounds is closely related to the complexity of characterizing the convex hull of quadratic forms of the underlying feasible region. In a related stream of literature, [26] developed approximations for the distributionally robust project crashing problem using information on the support, mean and correlation matrix of the activity durations. Using linear and piecewise linear decision rules, they developed computationally tractable second-order conic programming formulations to find resource allocations to minimize an upper bound on the worst-case expected makespan under both static and adaptive policies. While their numerical results demonstrate the promise of the distributionally robust approach, it is not clear as to how far their solution is from the true optimal solution. Recently, [31] studied a distributionally robust chance constrained version of the project crashing problem and developed a conic program to solve the problem under the assumption of the knowledge of a conic support, the mean and an upper bound on a positive homogeneous dispersion measure of the random activity durations. While their formulation is exact for the distributionally robust chance constrained project crashing problem, the size of the formulation grows in the number of paths in the network. An example where the worst-case expected makespan is computable in polynomial time was developed in [18] who assumed that a discrete distribution is provided for the activity durations for the set of arcs coming out of each node. The dependency structure among the activities for arcs coming out of two different nodes is however unspecified. [40] extended this result to propose a bound on the worst-case expected makespan with information on the mean and covariance matrix. A related stream of research uses robust optimization methods to solve project crashing problems where uncertainty sets are used to model the activity durations (see [59, 14, 34, 11, 12]).

In this paper, we build on these models to solve a class of distributionally robust project crashing problems in polynomial time. Furthermore, unlike the typical use of semidefinite programming solvers to directly solve the problem, we exploit the structure of the objective function to illustrate that the problem can be reformulated as a convex-concave saddle point problem over the first two marginal moment variables and the arc criticality index variables. This simplification provides the opportunity to make use of the saddle point formulation directly to solve the distributionally robust project crashing problem. To the best of our knowledge, this is one of the few attempts to solve the distributionally robust project crashing problem using the saddle point formulation directly. We use a projection and contraction algorithm for monotone variational inequalities in conjunction with a gradient method to solve the saddle point problem. As we demonstrate, this helps us solve larger problems than those currently possible with the semidefinite formulations. Lastly, we provide insights into the nature of the crashing solution from distributionally robust models that we believe are useful. Our results show that in comparison to the sample average approximation method for a multivariate normal distribution of activity durations, the distributionally robust models deploy more resources in crashing the standard deviations rather than the means. This implies that the project manager who is facing ambiguity in activity durations has more incentive to invest resources in reducing the variations rather than the means of the activity durations in comparison to a project manager who does not face any ambiguity.

3 SOCP, SDP and Saddle Point Formulations

In this section, we propose a model of the distributionally robust project crashing problem with moment information and identify instances where the problem is solvable in polynomial time in the size of the network.

3.1 Model

Let $\boldsymbol{\mu} = (\mu_{ij} : (i,j) \in \mathcal{A})$ and $\boldsymbol{\sigma} = (\sigma_{ij} : (i,j) \in \mathcal{A})$ denote the vector of means and standard deviations of the activity durations with $\boldsymbol{\rho}$ denoting additional (possibly limited) information on the correlation matrix that is available. The random duration of the activity (i,j) denoted by $\tilde{t}_{ij}$ is expressed as:

$$\tilde{t}_{ij} = \mu_{ij} + \sigma_{ij}\tilde{\xi}_{ij}, \quad \forall (i,j) \in \mathcal{A}, \tag{3.1}$$

where the random variables $\tilde{\xi}_{ij}$ have zero mean and unit variance for all activities with the correlation information among these random variables captured in $\boldsymbol{\rho}$. A simplifying assumption that is commonly made in project management is to assume that the activity durations are mutually independent. In our formulation, we can model this situation approximately by setting the correlation matrix to be the identity matrix, namely assuming that the activity durations are uncorrelated. However, we allow for the possibility that the activity durations might be correlated. This often arises in projects when activities are affected by common factors such as weather in a construction project (see [2]). A typical approach to decrease the average time of an activity duration is by allocating more workers to do the job. Several factors are known to affect the activity duration variability in projects in the construction industry. In particular, [58] identified the top nine causes for activity duration variation from over fifty causes by surveying laborers, foremen and project managers from civilian construction contractors that work with government agencies in the United States. The top nine causes that were identified in this survey which affect the variability in the activity durations were the following: (1) waiting to get answers to questions about a design or drawing, (2) turnaround time from engineers when there are questions about a drawing, (3) completion of previous activities, (4) socializing with fellow workers, (5) weather impacts such as excessive heat or rain, (6) rework that is needed due to the quality of previous work, (7) lack of skills and experience of the worker to perform a task, (8) people arriving late due to illness or personal reasons and (9) needing guidance and instructions from a supervisor. While some of these causes of variation are controllable by allocating resources to standardize the process and workflow, some of the variability in the activity durations due to factors such as weather are simply unavoidable. [17] outline several other examples of managing variability in projects. For example, in a case study analyzed there on notebook development, the computer maker Acer reduced variation in durations by (1) creating buffers in the form of slack capacities, (2) better documenting operating procedures so that it could train young engineers, (3) better quality management methods and (4) focussing the responsibilities of product specifications to within one group. At the same time it is known from the project management literature, that there is no straightforward dependence between the impact of variability in an activity duration at a particular stage of the project and the expected project duration (see [29]). While heuristics have been proposed to deal with managing the means and the variability of the activity durations, in this paper, we propose a mathematical optimization framework that captures the tradeoff between reducing the means and variances of individual activity durations in minimizing the expected project completion time, while accounting for ambiguity in distributions (see Figure 1). On the other hand, we assume that the correlation among the activity durations is not controllable by the project manager. This corresponds to the practical setting, where resources need to be deployed to individual activities to reduce the mean durations or better variance control techniques are applied to activities to reduce the standard deviations while the joint dependency among the activity durations is not under the project manager's control. This helps us develop a computationally tractable method for distributionally robust project crashing, without having to solve large semidefinite programs, while providing managers prescriptions on where resources need to be deployed.

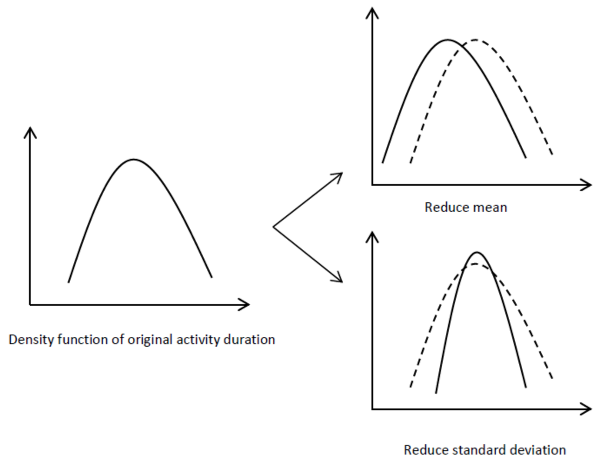


Figure 1: Reducing the mean and the standard deviation of the activity duration

These resources might jointly affect the mean and the correlation or might affect only one of the two. When the joint distribution of $\tilde{\boldsymbol{t}}$ is known only to lie in a set of distributions $\Theta_{\boldsymbol{\rho}}(\boldsymbol{\mu}, \boldsymbol{\sigma})$ with the given mean, standard deviation and correlation information, the worst-case expected makespan is defined as:

$$\max_{\theta \in \Theta_{\boldsymbol{\rho}}(\boldsymbol{\mu}, \boldsymbol{\sigma})} \mathbb{E}_\theta \left(\max_{\boldsymbol{x} \in \mathcal{X} \cap \{0,1\}^m} \tilde{\boldsymbol{t}}^T \boldsymbol{x} \right), \tag{3.2}$$

where the outer maximization is over the set of distributions with the given moment information on the random $\tilde{\boldsymbol{t}}$ and the inner maximization is over the intersection of the set $\mathcal{X}$ defined in (1.2) and $\{0,1\}^m$. When the correlation matrix $\boldsymbol{\rho}$ is completely specified, computing just the worst-case expected makespan for a given $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ is known to be a hard problem (see [3, 48, 59]). The distributionally robust project crashing problem to minimize the worst-case expected makespan is formulated as:

$$\min_{(\boldsymbol{\mu}, \boldsymbol{\sigma}) \in \boldsymbol{\Omega}} \max_{\theta \in \Theta_{\boldsymbol{\rho}}(\boldsymbol{\mu}, \boldsymbol{\sigma})} \mathbb{E}_\theta \left(\max_{\boldsymbol{x} \in \mathcal{X} \cap \{0,1\}^m} \tilde{\boldsymbol{t}}^T \boldsymbol{x} \right), \tag{3.3}$$

where $\boldsymbol{\Omega}$ defines a convex set of feasible allocations for $\boldsymbol{\omega} = (\boldsymbol{\mu}, \boldsymbol{\sigma})$. A simple example of the set $\boldsymbol{\Omega}$ is as follows:

$$\boldsymbol{\Omega} = \left\{ (\boldsymbol{\mu}, \boldsymbol{\sigma}) : \sum_{(i,j) \in \mathcal{A}} c_{ij}^{(1)}(\mu_{ij}) + c_{ij}^{(2)}(\sigma_{ij}) \leq B,\ \underline{\mu}_{ij} \leq \mu_{ij} \leq \overline{\mu}_{ij},\ \underline{\sigma}_{ij} \leq \sigma_{ij} \leq \overline{\sigma}_{ij},\ \ \forall (i,j) \in \mathcal{A} \right\}, \tag{3.4}$$

where $\overline{\mu}_{ij}$ and $\overline{\sigma}_{ij}$ are the mean and standard deviation of the original random duration of activity(i,j), and $\underline{\mu}_{ij}$ and $\underline{\sigma}_{ij}$ are the minimal mean and standard deviation of the duration of activity (i,j) that can be achieved by allocating resources to it. Further B is the amount of total cost budget, and $c_{ij}^{(1)}(\mu_{ij})$ and $c_{ij}^{(2)}(\sigma_{ij})$ are the cost functions which have the following properties: (a) $c_{ij}^{(1)}(\overline{\mu}_{ij}) = c_{ij}^{(2)}(\overline{\sigma}_{ij}) = 0$ which means the extra cost of activity (i,j) is 0 under the original mean duration and standard deviation; and (b) $c_{ij}^{(1)}(\mu_{ij})$ is a decreasing function of

μ_{ij} and likewise $c_{ij}^{(2)}(\sigma_{ij})$ is a decreasing function of σ_{ij}. Note that it is possible in this formulation to just crash the means by forcing the standard deviation to be fixed by setting $\overline{\sigma}_{ij} = \underline{\sigma}_{ij}$ in the outer optimization problem. Another simple example of a set to which our results can be directly applied is:

$$\boldsymbol{\Omega} = \{(\boldsymbol{\mu}, \boldsymbol{\sigma}) : \boldsymbol{\mu} = \boldsymbol{a} + \boldsymbol{A}\boldsymbol{y}, \boldsymbol{\sigma} = \boldsymbol{d} + \boldsymbol{D}\boldsymbol{y}, \boldsymbol{y} \in \mathcal{Y}\}, \tag{3.5}$$

where the mean vector $\boldsymbol{\mu}$ and the standard deviation vector $\boldsymbol{\sigma}$ are defined through affine transformations of a resource vector $\boldsymbol{y}$ that is assumed to lie in a convex set $\mathcal{Y}$. In this case through appropriate specification of the matrices $\boldsymbol{A}$ and $\boldsymbol{D}$ it is possible to allow the resource to jointly affect both the mean and the standard deviation.

3.2 Existing Formulations

In this section, we consider two formulations for the distributionally robust project crashing problem with no correlation and full correlation information on a parallel network.

3.2.1 No Correlation Information

We consider the marginal moment model (MMM) where information on the means and the standard deviations of the activity durations is assumed but no information on the correlations is assumed. The set of probability distributions of the activity durations with the given first moments is defined as:

$$\Theta_{\text{mmm}}(\boldsymbol{\mu}, \boldsymbol{\sigma}) = \left\{\theta \in \mathbb{M}(\Re_m) : \mathbb{E}_\theta(\tilde{t}_{ij}) = \mu_{ij}, \mathbb{E}_\theta(\tilde{t}_{ij}^2) = \mu_{ij}^2 + \sigma_{ij}^2, \forall (i,j) \in \mathcal{A}\right\}, \tag{3.6}$$

where $\mathbb{M}(\Re_m)$ is the set of finite positive Borel measures supported on $\Re_m$. In the definition of this set, we allow for the activity durations to be positively correlated, negatively correlated or even possibly uncorrelated. Furthermore, we do not make an explicit assumption on the nonnegativity of activity durations. There are three main reasons for this. Firstly, since we allow for the possibility of any valid correlation matrix with the given means and standard deviations, the most easy way to fit multivariate probability distribution to the activity durations is the normal distribution. As a result, this distribution has been used extensively in the literature on project networks (see [13, 2, 29]), particularly when the activity durations are correlated for the ease with which it can be fit to a given mean-covariance matrix. Secondly in practice, such an assumption is reasonable to justify when the mean of the activity duration is comparatively larger than the standard deviation in which case the probability of having a negative realization is small. In our numerical results, we focus on such examples. Lastly, it is known that even testing the feasibility of a nonnegative random vector with a fixed mean and a covariance matrix is already NP-hard (see [6]). In contrast, by dropping the assumption of nonnegativity, we can verify feasibility in a straightforward manner with a positive semidefinite condition. Assuming no correlation information, the worst-case expected makespan in (3.2) is equivalent to the optimal objective value of the following concave maximization problem over the convex hull of the set $\mathcal{X} \cap \{0,1\}^m$ which is exactly $\mathcal{X}$ (see Lemma 2 on page 458 in [46]):

$$\max_{\boldsymbol{x} \in \mathcal{X}} f_{\mathrm{mmm}}(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{x}), \tag{3.7}$$

where

$$f_{\mathrm{mmm}}(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{x}) = \sum_{(i,j) \in \mathcal{A}} \left(\mu_{ij} x_{ij} + \sigma_{ij} \sqrt{x_{ij}(1 - x_{ij})} \right). \tag{3.8}$$

In the formulation, the optimal x^*_{ij} variables is an estimate of the arc criticality index of activity (i, j) under the worst-case distribution. The worst-case expected makespan in (3.7) is computed using the following second-order cone program (SOCP):

$$\begin{aligned} \max_{\boldsymbol{x}, \boldsymbol{t}} \quad & \sum_{(i,j) \in \mathcal{A}} (\mu_{ij} x_{ij} + \sigma_{ij} t_{ij}) \\ \text{s.t.} \quad & \boldsymbol{x} \in \mathcal{X}, \\ & \sqrt{t_{ij}^2 + \left(x_{ij} - \frac{1}{2} \right)^2} \leq \frac{1}{2}, \quad \forall (i, j) \in \mathcal{A}. \end{aligned} \tag{3.9}$$

The distributionally robust PCP (3.3) under the marginal moment model is then formulated as a saddle point over the moment variables and arc criticality index variables as follows:

$$\min_{(\boldsymbol{\mu}, \boldsymbol{\sigma}) \in \boldsymbol{\Omega}} \max_{\boldsymbol{x} \in \mathcal{X}} f_{\mathrm{mmm}}(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{x}). \tag{3.10}$$

One approach to solve the problem is to take the dual of the maximization problem in (3.9) in which case the distributionally robust PCP (3.10) is formulated as the following second-order cone program:

$$\begin{aligned} \min_{\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{y}, \boldsymbol{\alpha}, \boldsymbol{\beta}} \quad & y_n - y_1 + \frac{1}{2} \sum_{(i,j) \in \mathcal{A}} (\alpha_{ij} - \beta_{ij}) \\ \text{s.t.} \quad & y_j - y_i - \beta_{ij} \geq \mu_{ij}, && \forall (i, j) \in \mathcal{A}, \\ & \sqrt{\sigma_{ij}^2 + \beta_{ij}^2} \leq \alpha_{ij}, && \forall (i, j) \in \mathcal{A} \\ & (\boldsymbol{\mu}, \boldsymbol{\sigma}) \in \Omega. \end{aligned} \tag{3.11}$$

Several points regarding the formulation in (3.11) are important to take note of. Firstly, the formulation is tight, namely it is an exact reformulation of the distributionally robust project crashing problem. Secondly, such a dual formulation has been recently applied by [42] to the appointment scheduling problem where the appointment times are chosen for patients (activities) while the actual service times for the patients are random. Their problem is equivalent to simply crashing the means of the activity durations. From formulation (3.11), we see that it is also possible to crash the standard deviation of the activity durations in a tractable manner using second-order cone programming with the marginal moment model. Lastly, though we focus on the project crashing problem, one of the nice features of this model is that it easily extends to all sets $\mathcal{X} \subseteq \{0, 1\}^n$ with a compact convex hull representation. In the next section, we discuss a partial correlation information structure that makes use of the project network structure in identifying a polynomial time solvable project crashing instance with semidefinite programming.

3.2.2 Parallel Network with Full Covariance Information

In the marginal moment model, we do not make any assumptions on the correlation information between the activity durations. Hence it is possible that in the worst-case, the correlations might be unrealistic particularly if some information on the dependence between activity durations is available. In this section, we consider alternative formulations where partial correlation information on the activity durations is known. Since the general version of this problem is hard, we focus on partial correlation information structures where the problem is solvable in polynomial time. Towards this, we first consider a parallel network. Consider a project network with k parallel paths (see Figure 2). In this network, the total number of nodes $n = k + 2$ and the number of arcs $m = 2k$. The activity durations of arcs $(1, j)$ for $j = 2, \ldots, k + 1$ are random while the activity durations of arcs $(j, k + 2)$ for $j = 2, \ldots, k + 1$ are deterministic with duration 0. We assume that the correlation among the random activity durations is known. In this case, without loss of generality, we can restrict our attention to the

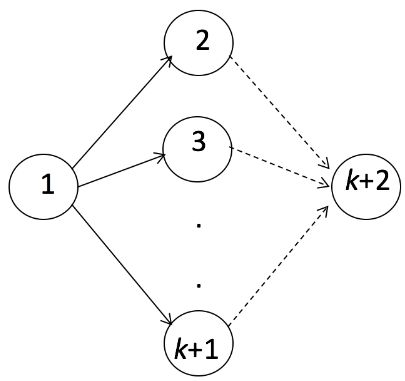


Figure 2: Parallel Project Network

set of probability distributions of the activity durations for the arcs $(1, j)$ for $j = 2, \ldots, k + 1$ which is defined as:

$$\Theta_{\text{cmm},\boldsymbol{\rho}}(\boldsymbol{\mu}, \boldsymbol{\sigma}) = \left\{ \theta \in \mathbb{M}(\Re_k) : \mathbb{E}_\theta(\tilde{\boldsymbol{t}}) = \boldsymbol{\mu}, \mathbb{E}_\theta(\tilde{\boldsymbol{t}}\tilde{\boldsymbol{t}}^T) = \boldsymbol{\mu}\boldsymbol{\mu}^T + \text{Diag}(\boldsymbol{\sigma})\boldsymbol{\rho}\text{Diag}(\boldsymbol{\sigma}) \right\}, \tag{3.12}$$

where $\boldsymbol{\rho} \in \mathcal{S}_k^{++}$ denotes the correlation matrix among the k random activity durations. The covariance matrix is defined as:

$$\boldsymbol{\Sigma} = \text{Diag}(\boldsymbol{\sigma})\boldsymbol{\rho}\text{Diag}(\boldsymbol{\sigma}). \tag{3.13}$$

We refer to this model as the cross moment model (CMM). The distributionally robust project crashing problem for a parallel network is then formulated as:

$$\min_{(\boldsymbol{\mu},\boldsymbol{\sigma}) \in \boldsymbol{\Omega}} \quad \max_{\theta \in \Theta_{\text{cmm},\boldsymbol{\rho}}(\boldsymbol{\mu},\boldsymbol{\sigma})} \quad \mathbb{E}_\theta \left(\max_{\boldsymbol{x} \in \boldsymbol{\Delta}_{k-1}} \tilde{\boldsymbol{t}}^T \boldsymbol{x} \right), \tag{3.14}$$

where the inner maximization is over the simplex since the network is parallel. The inner worst-case expected makespan problem in (3.14) is equivalent to the moment problem over the probability measure of the the random vector $\tilde{\boldsymbol{\xi}} = \text{Diag}(\boldsymbol{\sigma})^{-1}(\tilde{\boldsymbol{t}} - \boldsymbol{\mu})$ denoted by γ as follows:

$$\begin{aligned} \max \quad & \mathbb{E}_\gamma \left(\max_{\boldsymbol{x} \in \boldsymbol{\Delta}_{k-1}} \left(\boldsymbol{\mu} + \text{Diag}(\boldsymbol{\sigma}) \tilde{\boldsymbol{\xi}} \right)^T \boldsymbol{x} \right) \\ \text{s.t.} \quad & \mathbb{P}_\gamma(\tilde{\boldsymbol{\xi}} \in \Re_k) = 1, \\ & \mathbb{E}_\gamma(\tilde{\boldsymbol{\xi}}) = 0, \\ & \mathbb{E}_\gamma(\tilde{\boldsymbol{\xi}} \tilde{\boldsymbol{\xi}}^T) = \boldsymbol{\rho}. \end{aligned} \tag{3.15}$$

A direct application of standard moment duality for this problem by associating the dual variables λ_0, $\boldsymbol{\lambda}$ and $\boldsymbol{\Lambda}$ with the constraints and disaggregating the maximum over the extreme points of the simplex implies that problem (3.15) can be solved as a semidefinite program:

$$\begin{aligned} \min_{\lambda_0, \boldsymbol{\lambda}, \boldsymbol{\Lambda}} \quad & \lambda_0 + \langle \boldsymbol{\rho}, \boldsymbol{\Lambda} \rangle \\ \text{s.t.} \quad & \begin{pmatrix} \lambda_0 - \mu_{1j} & \frac{1}{2}(\boldsymbol{\lambda} - \sigma_{1j} \boldsymbol{e}_j)^T \\ \frac{1}{2}(\boldsymbol{\lambda} - \sigma_{1j} \boldsymbol{e}_j) & \boldsymbol{\Lambda} \end{pmatrix} \succeq 0, \quad \forall j = 2, \ldots, k+1, \end{aligned} \tag{3.16}$$

where $\boldsymbol{e}_j$ is a vector of dimension k with 1 in the entry corresponding to node j and 0 otherwise. Strong duality holds in this case under the assumption that the correlation matrix is positive definite. Plugging it back into (3.14), we obtain the semidefinite program for distributionally robust project crashing problem over a parallel network as follows:

$$\begin{aligned} \min_{\boldsymbol{\mu}, \boldsymbol{\sigma}, \lambda_0, \boldsymbol{\lambda}, \boldsymbol{\Lambda}} \quad & \lambda_0 + \langle \boldsymbol{\rho}, \boldsymbol{\Lambda} \rangle \\ \text{s.t.} \quad & \begin{pmatrix} \lambda_0 - \mu_{1j} & \frac{1}{2}(\boldsymbol{\lambda} - \sigma_{1j} \boldsymbol{e}_j)^T \\ \frac{1}{2}(\boldsymbol{\lambda} - \sigma_{1j} \boldsymbol{e}_j) & \boldsymbol{\Lambda} \end{pmatrix} \succeq 0, \quad \forall j = 2, \ldots, k+1, \\ & (\boldsymbol{\mu}, \boldsymbol{\sigma}) \in \boldsymbol{\Omega}. \end{aligned} \tag{3.17}$$

3.3 New Formulations

In this section, we consider new formulations for the distributionally robust project crashing problem for the parallel network case and then a generalization to the full network case with partial correlation information.

3.3.1 Parallel Network

We first provide an alternative reformulation of the project crashing problem in the spirit of (3.10) as a convex-concave saddle point problem where the number of variables in the formulation grow linearly in the number of arcs. Unlike the original minimax formulation in (3.14) where the outer maximization problem is over infinite dimensional probability measures, we transform the outer maximization problem to optimization over finite dimensional variables (specifically the arc criticality indices).

Proposition 1. *Define $S(\boldsymbol{x}) = \text{Diag}(\boldsymbol{x}) - \boldsymbol{x}\boldsymbol{x}^T$. Under the cross moment model with a parallel network, the distributionally robust PCP (3.14) is solvable as a convex-concave saddle point problem:*

$$\min_{(\boldsymbol{\mu}, \boldsymbol{\sigma}) \in \boldsymbol{\Omega}} \max_{\boldsymbol{x} \in \boldsymbol{\Delta}_{k-1}} f_{cmm}(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{x}), \tag{3.18}$$

where

$$f_{cmm}(\boldsymbol{\mu},\boldsymbol{\sigma},\boldsymbol{x}) = \boldsymbol{\mu}^T\boldsymbol{x} + \text{trace}\left(\left(\boldsymbol{\Sigma}^{1/2}S(\boldsymbol{x})\boldsymbol{\Sigma}^{1/2}\right)^{1/2}\right). \tag{3.19}$$

The objective function is convex with respect to the moments $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}^{1/2}$ (and hence $\boldsymbol{\sigma}$ for a fixed $\boldsymbol{\rho}$) and strictly concave with respect to the criticality index variables $\boldsymbol{x}$.

Proof. First, note that the matrix $\boldsymbol{S}(\boldsymbol{x})$ is positive semidefinite for all $\boldsymbol{x} \in \boldsymbol{\Delta}_{k-1}$ since:

$$\boldsymbol{v}^T\boldsymbol{S}(\boldsymbol{x})\boldsymbol{v} = \sum_{i\in[k]} v_i^2 x_i - \left(\sum_{i\in[k]} v_i x_i\right)^2 \geq 0, \forall \boldsymbol{v} \in \Re_k,$$

where the last inequality comes from $\mathbb{E}(\tilde{v}^2) \geq \mathbb{E}(\tilde{v})^2$ assuming the random variable $\tilde{v}$ is defined to take value v_i with probability x_i for $i \in [k]$. The worst-case expected makespan under the cross moment model for a parallel network was studied in [1] (see Theorem 1) who showed that it is equivalent to the optimal objective value of the following nonlinear concave maximization problem over the unit simplex:

$$\max_{\theta\in\Theta_{\text{cmm},\boldsymbol{\rho}}(\boldsymbol{\mu},\boldsymbol{\sigma})} \mathbb{E}_\theta\left(\max_{\boldsymbol{x}\in\boldsymbol{\Delta}_{k-1}} \tilde{\boldsymbol{t}}^T\boldsymbol{x}\right) = \max_{\boldsymbol{x}\in\boldsymbol{\Delta}_{k-1}}\left(\boldsymbol{\mu}^T\boldsymbol{x} + \text{trace}\left(\left(\boldsymbol{\Sigma}^{1/2}S(\boldsymbol{x})\boldsymbol{\Sigma}^{1/2}\right)^{1/2}\right)\right), \tag{3.20}$$

where the optimal x_{ij}^* variables is an estimate of the arc criticality index of activity (i, j) under the worst-case distribution. This results in the equivalent saddle point formulation (3.18) for the distributionally robust project crashing problem under the cross moment model with a parallel network. The function $f_{\text{cmm}}(\boldsymbol{\mu},\boldsymbol{\sigma},\boldsymbol{x})$ has shown to be strongly concave in the $\boldsymbol{x}$ variable (see Theorem 3 in [1]). The function $f_{\text{cmm}}(\boldsymbol{\mu},\boldsymbol{\sigma},\boldsymbol{x})$ is linear and hence convex in the $\boldsymbol{\mu}$ variable. Furthermore this function is convex with respect to $\boldsymbol{\Sigma}^{1/2} \in \mathcal{S}_k^{++}$. To see this, we apply Theorem 7.2 in [10] which shows that the function $g(\boldsymbol{A}) = \text{trace}((\boldsymbol{B}^T\boldsymbol{A}^2\boldsymbol{B})^{1/2})$ is convex in $\boldsymbol{A} \in \mathcal{S}_k^{++}$ for a fixed $\boldsymbol{B} \in \Re_{k\times k}$. Clearly, the function $\text{trace}((\boldsymbol{\Sigma}^{1/2}S(\boldsymbol{x})\boldsymbol{\Sigma}^{1/2})^{1/2}) = \text{trace}((S(\boldsymbol{x})^{1/2}\boldsymbol{\Sigma}S(\boldsymbol{x})^{1/2})^{1/2})$ since for any square matrix $\boldsymbol{X}$, the eigenvalues of $\boldsymbol{X}\boldsymbol{X}^T$ are the same as $\boldsymbol{X}^T\boldsymbol{X}$, which implies $\text{trace}((\boldsymbol{X}\boldsymbol{X}^T)^{1/2}) = \text{trace}((\boldsymbol{X}^T\boldsymbol{X})^{1/2})$. Setting $\boldsymbol{A} = \boldsymbol{\Sigma}^{1/2}$ and $\boldsymbol{B} = S(\boldsymbol{x})^{1/2}$, implies that the objective function is convex with respect to $\boldsymbol{\Sigma}^{1/2} \in \mathcal{S}_k^{++}$. $\square$

3.3.2 General Network

To model the partial correlation information, we assume that for the subset of arcs that leave a node, information on the correlation matrix is available. Let $\mathcal{A}_i$ denote the set of arcs originating from node i for $i = 1, \ldots, n-1$. Note that the sets $\mathcal{A}_i$ are non-overlapping. Then, the set of arcs $\mathcal{A} = \bigcup_{i=1}^{n-1} \mathcal{A}_i$. We let $\tilde{\boldsymbol{t}}_i$ denote the sub-vector of durations $\tilde{t}_{ij}$ for arcs $(i, j) \in \mathcal{A}_i$. In the non-overlapping marginal moment model (NMM), we define the set of distributions of the random vector $\tilde{\boldsymbol{t}}$ as follows:

$$\Theta_{\text{nmm},\boldsymbol{\rho}_i,\forall i}(\boldsymbol{\mu},\boldsymbol{\sigma}) = \left\{\theta \in \mathbb{M}(\Re_m) : \mathbb{E}_\theta(\tilde{\boldsymbol{t}}_i) = \boldsymbol{\mu}_i, \mathbb{E}_\theta(\tilde{\boldsymbol{t}}_i\tilde{\boldsymbol{t}}_i^T) = \boldsymbol{\mu}_i\boldsymbol{\mu}_i^T + \text{Diag}(\boldsymbol{\sigma}_i)\boldsymbol{\rho}_i\text{Diag}(\boldsymbol{\sigma}_i), \forall i = 1, \ldots, n-1\right\}, \tag{3.21}$$

where $\boldsymbol{\mu}_i$ denotes the mean vector for $\tilde{\boldsymbol{t}}_i$, $\boldsymbol{\rho}_i \in \mathcal{S}_{|\mathcal{A}_i|}^{++}$ denotes the correlation matrix of $\tilde{\boldsymbol{t}}_i$ and $\boldsymbol{\Sigma}_i = \text{Diag}(\boldsymbol{\sigma}_i)\boldsymbol{\rho}_i\text{Diag}(\boldsymbol{\sigma}_i)$ denotes the covariance matrix of $\tilde{\boldsymbol{t}}_i$. However, note that in the definition of (3.21), we assume that the correlation between activity durations of the arcs that originate from different nodes is unknown. This is often a reasonable assumption in project networks since the local information of activity durations that originate from a node will typically be better understood by the project manager who might subcontract those activities to a group that is responsible for that part of the project while the global dependency information is much more complicated to model. A typical simplifying assumption is to then let the activity durations be independent for arcs leaving different nodes. The expected project completion time is hard to compute in this case and bounds have been proposed under the assumption of independence (see [25]). On the other hand, in the model discussed in this paper, we allow for these activity durations to be arbitrarily dependent for arcs exiting different nodes and thus take a worst-case perspective. Under partial correlation information, the distributionally robust project crashing problem for a general network is formulated as:

$$\min_{(\boldsymbol{\mu},\boldsymbol{\sigma})\in\boldsymbol{\Omega}} \max_{\theta\in\Theta_{\mathrm{nmm},\boldsymbol{\rho}_i,\forall i}(\boldsymbol{\mu},\boldsymbol{\sigma})} \mathbb{E}_\theta\left(\max_{\boldsymbol{x}\in\mathcal{X}} \tilde{\boldsymbol{t}}^T\boldsymbol{x}\right). \tag{3.22}$$

Under the nonoverlapping marginal moment model, the worst-case expected makespan in (3.22) is equivalent to the optimal objective value of the following semidefinite maximization problem over the convex hull of the set $\mathcal{X}$ (see Theorem 15 on page 467 in [40]):

$$\begin{aligned}
\max_{x_{ij},\boldsymbol{w}_{ij},\boldsymbol{W}_{ij}} \quad & \sum_{(i,j)\in\mathcal{A}} \left(\mu_{ij}x_{ij} + \sigma_{ij}\boldsymbol{e}_{ij}^T\boldsymbol{w}_{ij}\right) \\
\text{s.t.} \quad & \boldsymbol{x}\in\mathcal{X}, \\
& \begin{pmatrix} 1 & 0 \\ 0 & \boldsymbol{\rho}_i \end{pmatrix} - \sum_{(i,j)\in\mathcal{A}_i} \begin{pmatrix} x_{ij} & \boldsymbol{w}_{ij}^T \\ \boldsymbol{w}_{ij} & \boldsymbol{W}_{ij} \end{pmatrix} \succeq 0, \quad \forall i = 1,\ldots,n-1, \\
& \begin{pmatrix} x_{ij} & \boldsymbol{w}_{ij}^T \\ \boldsymbol{w}_{ij} & \boldsymbol{W}_{ij} \end{pmatrix} \succeq 0, \qquad \forall (i,j)\in\mathcal{A},
\end{aligned}$$

where $\boldsymbol{e}_{ij}$ is a vector of dimension $|\mathcal{A}_i|$ with 1 in the element corresponding to node j and 0 otherwise. Here, as usual in stochastic project crashing formulations, x_{ij} correspond to the probability of arc (i,j) being on the critical (longest) path. The rest of the decision variables correspond to conditional moments:

$$\begin{pmatrix} x_{ij} & \boldsymbol{w}_{ij}^T \\ \boldsymbol{w}_{ij} & \boldsymbol{W}_{ij} \end{pmatrix} = \mathbb{P}(\text{(i,j) is critical}) \begin{pmatrix} 1 & \mathbb{E}(\tilde{\boldsymbol{t}}_i^T|\text{(i,j) is critical}) \\ \mathbb{E}(\tilde{\boldsymbol{t}}_i|\text{(i,j) is critical}) & \mathbb{E}(\tilde{\boldsymbol{t}}_i\tilde{\boldsymbol{t}}_i^T|\text{(i,j) is critical}) \end{pmatrix}. \tag{3.23}$$

By taking the dual of the problem where strong duality holds under the assumption $\boldsymbol{\rho}_i \in \mathcal{S}_{|\mathcal{A}_i|}^{++}$ for all i, the distributionally robust PCP (3.22) is solvable as the semidefinite program:

$$\begin{aligned}
\min_{\boldsymbol{\mu},\boldsymbol{\sigma},\boldsymbol{y},\boldsymbol{d},\lambda_{0i},\boldsymbol{\lambda}_i,\boldsymbol{\Lambda}_i} \quad & y_n - y_1 + \sum_{i=1}^{n-1}\left(\lambda_{0i} + \langle\boldsymbol{\rho}_i,\boldsymbol{\Lambda}_i\rangle\right) \\
\text{s.t.} \quad & y_j - y_i \geq d_{ij}, \qquad \forall (i,j)\in\mathcal{A}, \\
& \begin{pmatrix} \lambda_{0i} + d_{ij} - \mu_{ij} & \frac{1}{2}(\boldsymbol{\lambda}_i - \sigma_{ij}\boldsymbol{e}_{ij})^T \\ \frac{1}{2}(\boldsymbol{\lambda}_i - \sigma_{ij}\boldsymbol{e}_{ij}) & \boldsymbol{\Lambda}_i \end{pmatrix} \succeq 0, \quad \forall (i,j)\in\mathcal{A}, \\
& \begin{pmatrix} \lambda_{0i} & \frac{1}{2}\boldsymbol{\lambda}_i^T \\ \frac{1}{2}\boldsymbol{\lambda}_i & \boldsymbol{\Lambda}_i \end{pmatrix} \succeq 0, \quad \forall i = 1,\ldots,n-1, \\
& (\boldsymbol{\mu},\boldsymbol{\sigma})\in\boldsymbol{\Omega}.
\end{aligned}$$

We next provide an alternative reformulation of the project crashing problem with partial correlation information as a convex-concave saddle point problem using the result from the previous section for parallel networks.

Proposition 2. *Let* $\boldsymbol{x}_i \in \Re_{|\mathcal{A}_i|}$ *and define* $S(\boldsymbol{x}_i) = \text{Diag}(\boldsymbol{x}_i) - \boldsymbol{x}_i\boldsymbol{x}_i^T$ *for all* $i = 1, \ldots, n-1$. *Under the nonoverlapping multivariate marginal moment model for a general network, the distributionally robust PCP (3.22) is solvable as a convex-concave saddle point problem:*

$$\min_{(\boldsymbol{\mu},\boldsymbol{\sigma})\in\boldsymbol{\Omega}} \max_{\boldsymbol{x}\in\mathcal{X}} f_{nmm}(\boldsymbol{\mu},\boldsymbol{\sigma},\boldsymbol{x}), \tag{3.24}$$

where

$$f_{nmm}(\boldsymbol{\mu},\boldsymbol{\sigma},\boldsymbol{x}) = \sum_{i=1}^{n-1}\left(\boldsymbol{\mu}_i^T\boldsymbol{x}_i + \text{trace}\left(\boldsymbol{\Sigma}_i^{1/2}S(\boldsymbol{x}_i)\boldsymbol{\Sigma}_i^{1/2}\right)^{1/2}\right). \tag{3.25}$$

The objective function is convex with respect to the moment variables $\boldsymbol{\mu}_i$ *and* $\boldsymbol{\Sigma}_i^{1/2}$ *(and hence* $\boldsymbol{\sigma}_i$ *for a fixed* $\boldsymbol{\rho}_i$*) and strictly concave with respect to the arc criticality index variables* $\boldsymbol{x}$.

Proof. See Appendix. □

4 Saddle Point Methods for Project Crashing

In this section, we illustrate the possibility of using first order saddle point methods to solve distributionally robust project crashing problems.

4.1 Gradient Characterization and Optimality Condition

We first characterize the gradient of the objective function for the parallel network and the general network before characterizing the optimality condition.

Proposition 3. *Define* $T(\boldsymbol{x}) = \boldsymbol{\Sigma}^{1/2}S(\boldsymbol{x})\boldsymbol{\Sigma}^{1/2}$. *Under the cross moment model with a parallel network, the gradient of* f_{cmm} *in (3.19) with respect to* $\boldsymbol{x}$ *is given as:*

$$\nabla_{\boldsymbol{x}} f_{cmm}(\boldsymbol{\mu},\boldsymbol{\sigma},\boldsymbol{x}) = \boldsymbol{\mu} + \frac{1}{2}\left(\text{diag}(\boldsymbol{\Sigma}^{1/2}(T^{1/2}(\boldsymbol{x}))^{\dagger}\boldsymbol{\Sigma}^{1/2}) - 2\boldsymbol{\Sigma}^{1/2}(T^{1/2}(\boldsymbol{x}))^{\dagger}\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\right). \tag{4.1}$$

The gradient of f_{cmm} *in (3.19) with respect to* $(\boldsymbol{\mu},\boldsymbol{\sigma})$ *is given as:*

$$\nabla_{\boldsymbol{\mu},\boldsymbol{\sigma}} f_{cmm}(\boldsymbol{\mu},\boldsymbol{\sigma},\boldsymbol{x}) = \left(\boldsymbol{x}, \text{diag}\left(\boldsymbol{\Sigma}^{-1/2}T^{1/2}(\boldsymbol{x})\boldsymbol{\Sigma}^{-1/2}\text{Diag}(\boldsymbol{\sigma})\boldsymbol{\rho}\right)\right). \tag{4.2}$$

Proof. See Appendix. □

Note that, the gradients given in (4.1) and (4.2) can be calculated in $O(n^3)$ time as the computation requires calculating the square root of an $n \times n$ positive semidefinite matrix and a few number of multiplications of $n \times n$ matrices.

The optimality condition for (3.18) is then given as:

$$\begin{aligned} \boldsymbol{x} =& P_{\Delta}\Big(\boldsymbol{x} + \nabla_{\boldsymbol{x}} f_{\text{cmm}}(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{x})\Big) \\ (\boldsymbol{\mu}, \boldsymbol{\sigma}) =& P_{\boldsymbol{\Omega}}\Big((\boldsymbol{\mu}, \boldsymbol{\sigma}) - \nabla_{\boldsymbol{\mu},\boldsymbol{\sigma}} f_{\text{cmm}}(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{x})\Big), \end{aligned} \tag{4.3}$$

where $P_S(\cdot)$ denotes the projection onto a set S and ∇ denotes the partial derivative. Note that, the projection of a vector on the unit simplex is traditionally done by sorting the elements of the vector. Therefore, the complexity of the projection operation is $O(n^2)$ or $O(n \log n)$ depending on the choice of the sorting algorithm. This can be slightly improved with recent methods as discussed in detail by [15].

Similarly for the general network with partial correlations, we can extend the gradient characterization from Proposition 3 to the general network. Define $T(\boldsymbol{x}_i) = \boldsymbol{\Sigma}_i^{1/2} S(\boldsymbol{x}_i) \boldsymbol{\Sigma}_i^{1/2}$, $\forall i = 1, \dots, n-1$. The gradients of f_{nmm} with respect to $\boldsymbol{x}$ and $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ are

$$\nabla_{\boldsymbol{x}} f_{\text{nmm}}(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{x}) = \begin{pmatrix} \boldsymbol{\mu}_1 + g_{\boldsymbol{x}}(\boldsymbol{\sigma}_1, \boldsymbol{x}_1) \\ \boldsymbol{\mu}_2 + g_{\boldsymbol{x}}(\boldsymbol{\sigma}_2, \boldsymbol{x}_2) \\ \vdots \\ \boldsymbol{\mu}_{n-1} + g_{\boldsymbol{x}}(\boldsymbol{\sigma}_{n-1}, \boldsymbol{x}_{n-1}) \end{pmatrix}, \ \nabla_{\boldsymbol{\mu},\boldsymbol{\sigma}} f_{\text{nmm}}(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{x}) = \left(\begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_{n-1} \end{pmatrix}, \begin{pmatrix} g_{\boldsymbol{\sigma}}(\boldsymbol{\sigma}_1, \boldsymbol{x}_1) \\ g_{\boldsymbol{\sigma}}(\boldsymbol{\sigma}_2, \boldsymbol{x}_2) \\ \vdots \\ g_{\boldsymbol{\sigma}}(\boldsymbol{\sigma}_{n-1}, \boldsymbol{x}_{n-1}) \end{pmatrix} \right), \tag{4.4}$$

where

$$\begin{aligned} g_{\boldsymbol{x}}(\boldsymbol{\sigma}_i, \boldsymbol{x}_i) =& \frac{1}{2}\Big(\text{diag}(\boldsymbol{\Sigma}_i^{1/2}(T^{1/2}(\boldsymbol{x}_i))^{\dagger}\boldsymbol{\Sigma}_i^{1/2}) - 2\boldsymbol{\Sigma}_i^{1/2}(T^{1/2}(\boldsymbol{x}_i))^{\dagger}\boldsymbol{\Sigma}_i^{1/2}\boldsymbol{x}_i\Big), \forall i = 1, \dots, n-1, \\ g_{\boldsymbol{\sigma}}(\boldsymbol{\sigma}_i, \boldsymbol{x}_i) =& \text{diag}\Big(\boldsymbol{\Sigma}_i^{-1/2} T^{1/2}(\boldsymbol{x}_i) \boldsymbol{\Sigma}_i^{-1/2} \text{Diag}(\boldsymbol{\sigma}_i) \boldsymbol{\rho}_i\Big), \forall i = 1, \dots, n-1. \end{aligned} \tag{4.5}$$

The optimality condition for (3.24) is then given as:

$$\begin{aligned} \boldsymbol{x} =& P_{\mathcal{X}}\Big(\boldsymbol{x} + \nabla_{\boldsymbol{x}} f_{\text{nmm}}(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{x})\Big) \\ (\boldsymbol{\mu}, \boldsymbol{\sigma}) =& P_{\boldsymbol{\Omega}}\Big((\boldsymbol{\mu}, \boldsymbol{\sigma}) - \nabla_{\boldsymbol{\mu},\boldsymbol{\sigma}} f_{\text{nmm}}(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{x})\Big). \end{aligned} \tag{4.6}$$

A key step in the algorithm is to compute the projections given in (4.6) onto the sets $\mathcal{X}$ and $\boldsymbol{\Omega}$, respectively. In our setting, $\mathcal{X}$ is a network flow polytope and the projection can be computed by solving a least square problem that is efficiently solvable with standard convex quadratic programming solvers. Similarly, if we use a budgeted uncertainty set for $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ as given in Equation (3.4) where $c_{ij}^{(1)}(\cdot)$ and $c_{ij}^{(2)}(\cdot)$ are univariate piecewise linear or quadratic convex functions, then the projection on the set $\boldsymbol{\Omega}$ is computed by solving a convex quadratic program. For more general closed convex sets $\boldsymbol{\Omega}$, the complexity of the projection operation would depend on the representation of the set. In our numerical examples, we focus on simple sets $\boldsymbol{\Omega}$ such that the projection operator is efficiently computable as convex quadratic programs for tractability purposes.

In the next section, we discuss saddle point methods that can be used to solve the problem.

4.2 Algorithm

In this section, we discuss the possibility of the use of saddle point algorithms to solve the distributionally robust project crashing problem. Define the inner maximization problem $\phi(\boldsymbol{\mu}, \boldsymbol{\sigma}) = \max_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{x})$ which requires solving a maximization problem of a strictly concave function over a polyhedral set $\mathcal{X}$. One possible method is to use a projected gradient method possibly with an Armijo line search method to compute the value of $\phi(\boldsymbol{\mu}, \boldsymbol{\sigma})$ and the corresponding optimal $\boldsymbol{x}^*(\boldsymbol{\mu}, \boldsymbol{\sigma})$. Such an algorithm is described in Algorithm 1 and has been used in [1] to solve the inner maximization problem in a discrete choice problem setting. It is also shown that under mild assumptions, the algorithm converges in linear time after an iterate in a local neighborhood of the optimal solution has been reached. In addition, its practical advantages are demonstrated via computational experiments for large n.

Algorithm 1: Projected gradient algorithm with Armijo search

Input: $\boldsymbol{\mu}, \boldsymbol{\sigma}$, $\mathcal{X}$, starting point $\boldsymbol{x}_0$, initial step size α, tolerance ϵ.

Output: Optimal solution $\boldsymbol{x}$.

Initialize stopping criteria: $criteria \leftarrow \epsilon + 1$;

while $criteria > \epsilon$ **do**

- $\boldsymbol{z} \leftarrow P_{\mathcal{X}}(\boldsymbol{x}_0 + \nabla_{\boldsymbol{x}} f(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{x}_0))$,
- $criteria \leftarrow \|\boldsymbol{z} - \boldsymbol{x}_0\|$,
- $\boldsymbol{x} \leftarrow \boldsymbol{x}_0 + \gamma(\boldsymbol{z} - \boldsymbol{x}_0)$, where γ is determined with an Armijo rule, i.e. $\gamma = \alpha \cdot 2^{-l}$ with
  $l = \min\{j \in \mathbb{Z}_+ : f(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{x}_0 + \alpha \cdot 2^{-j}(\boldsymbol{z} - \boldsymbol{x}_0)) \geq f(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{x}_0) + \tau\alpha 2^{-j}\langle \nabla_{\boldsymbol{x}} f(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{x}_0), \boldsymbol{z} - \boldsymbol{x}_0\rangle\}$
  for some $\tau \in (0, 1)$.
- $\boldsymbol{x}_0 \leftarrow \boldsymbol{x}$.

end

The optimality condition (4.6) in this case is reduced to:

$$(\boldsymbol{\mu}, \boldsymbol{\sigma}) = P_{\boldsymbol{\Omega}}\Big((\boldsymbol{\mu}, \boldsymbol{\sigma}) - F(\boldsymbol{\mu}, \boldsymbol{\sigma})\Big), \tag{4.7}$$

where

$$F(\boldsymbol{\mu}, \boldsymbol{\sigma}) = \nabla_{\boldsymbol{\mu}, \boldsymbol{\sigma}} f(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{x}^*(\boldsymbol{\mu}, \boldsymbol{\sigma})). \tag{4.8}$$

Proposition 4. *The operator F as defined in (4.8) is continuous and monotone.*

Proof. First, the optimal solution $\boldsymbol{x}^*(\boldsymbol{\mu}, \boldsymbol{\sigma})$ to $\max_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{x})$ is unique because of the strict concavity of $f(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{x})$ with respect $\boldsymbol{x}$. Moreover, $\boldsymbol{x}^*(\boldsymbol{\mu}, \boldsymbol{\sigma})$ is continuous with respect to $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ since f is strictly concave with respect to $\boldsymbol{x}$ and $\mathcal{X}$ is convex and bounded (see [22]). In addition, the function $\nabla_{\boldsymbol{\mu}, \boldsymbol{\sigma}} f(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{x})$ is continuous with respect to $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ and $\boldsymbol{x}$. Therefore, $F(\boldsymbol{\mu}, \boldsymbol{\sigma})$ is continuous with respect to $(\boldsymbol{\mu}, \boldsymbol{\sigma})$. Notice that $F(\boldsymbol{\mu}, \boldsymbol{\sigma})$ is a subgradient of the convex function $\phi(\boldsymbol{\mu}, \boldsymbol{\sigma}) = \max_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{x})$ (see [52]). Hence F is monotone:

$$\langle F(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\sigma}}) - F(\boldsymbol{\mu}, \boldsymbol{\sigma}), (\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\sigma}}) - (\boldsymbol{\mu}, \boldsymbol{\sigma})\rangle \geq 0, \quad \forall (\boldsymbol{\mu}, \boldsymbol{\sigma}), (\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\sigma}}) \in \boldsymbol{\Omega}. \tag{4.9}$$

□

The optimality condition is then equivalent to the following variational inequality ([19]) :

$$\text{find } (\boldsymbol{\mu}^*, \boldsymbol{\sigma}^*) \in \boldsymbol{\Omega} : \ \langle F(\boldsymbol{\mu}^*, \boldsymbol{\sigma}^*), \ \ (\boldsymbol{\mu}, \boldsymbol{\sigma}) - (\boldsymbol{\mu}^*, \boldsymbol{\sigma}^*)\rangle \geq 0, \ \forall (\boldsymbol{\mu}, \boldsymbol{\sigma}) \in \boldsymbol{\Omega}. \tag{4.10}$$

Under the condition that the operator F is continuous monotone, one method to find a solution to such a variational inequality is the projection and contraction method ([33]). The algorithm is as follows:

Algorithm 2: Projection and contraction algorithm for monotone variational inequalities

---

Input: Parameters for set $\boldsymbol{\Omega}$, the starting point $(\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0)$, initial step size α, tolerance ϵ.

Output: Optimal solution $(\boldsymbol{\mu}, \boldsymbol{\sigma})$.

Initialize stopping criteria: $criteria \leftarrow \epsilon + 1$, set a value of $\delta \in (0, 1)$.

while $criteria > \epsilon$ **do**

  $\beta \leftarrow \alpha$

  $\boldsymbol{res} \leftarrow (\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0) - P_{\boldsymbol{\Omega}}((\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0) - \beta F(\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0))$

  $\boldsymbol{d} \leftarrow \boldsymbol{res} - \beta[F(\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0) - F(P_{\boldsymbol{\Omega}}((\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0) - \beta F(\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0)))]$

  $criteria \leftarrow \|\boldsymbol{res}\|$

  while $\langle \boldsymbol{res}, \boldsymbol{d} \rangle < \delta \|\boldsymbol{res}\|^2$ **do**

    $\beta \leftarrow \beta/2$

    $\boldsymbol{res} \leftarrow (\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0) - P_{\boldsymbol{\Omega}}((\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0) - \beta F(\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0))$

    $\boldsymbol{d} \leftarrow \boldsymbol{res} - \beta[F(\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0) - F(P_{\boldsymbol{\Omega}}((\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0) - \beta F(\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0)))]$

  end

  $(\boldsymbol{\mu}, \boldsymbol{\sigma}) \leftarrow (\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0) - \frac{\langle \boldsymbol{res}, \boldsymbol{d} \rangle}{\langle \boldsymbol{d}, \boldsymbol{d} \rangle} \cdot \boldsymbol{d}, \quad (\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0) \leftarrow (\boldsymbol{\mu}, \boldsymbol{\sigma}).$

end

We have chosen to solve the saddle point formulation of the robust PCP using a projection contraction method because it is relatively easy to implement, uses little storage, and therefore it is an attractive alternative for solving large-scale problems in general. (See for example [60] and the references therein for an extensive discussion on the properties and advantages of projection type algorithms for solving variational inequalities.) Most of the recent papers in robust and distributionally robust optimization solve formulations with min-max objective functions by first taking the dual of the inner problem as we have also discussed in formulations (3.9), (3.17) and (3.24). In this paper instead, we solve the saddle point formulations directly by making use of the projection type algorithms which seems to be reasonable, especially for large scale problems. Our experiments in the next section illustrate this concept and shows that more such algorithms can be used to solve problems with similar structure. (See [32]) for a survey on successful applications of such algorithms to solve variational inequalities arising from a broad range of of applications.)

5 Numerical Experiments

In this section, we report the results from numerical tests in which we solve the project crashing problem under various models. We also demonstrate that for large instances of the problem solving the saddle point reformulations of the robust project crashing problem under CMM and NMM models using Algorithm 2 is more efficient than solving their SDP formulations using a standard solver. For the numerical tests, we assume that the feasible set of $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ is defined as

$$\boldsymbol{\Omega} = \Big\{ (\boldsymbol{\mu}, \boldsymbol{\sigma}) : \sum_{(i,j)\in\mathcal{A}} c_{ij}^{(1)}(\mu_{ij}) + c_{ij}^{(2)}(\sigma_{ij}) \le B, \ \underline{\mu}_{ij} \le \mu_{ij} \le \overline{\mu}_{ij}, \ \underline{\sigma}_{ij} \le \sigma_{ij} \le \overline{\sigma}_{ij}, \quad \forall (i,j) \in \mathcal{A} \Big\}. \tag{5.1}$$

The cost functions are assumed to be convex and quadratic of the form:

$$c_{ij}^{(1)}(\mu_{ij}) = a_{ij}^{(1)}(\overline{\mu}_{ij} - \mu_{ij}) + b_{ij}^{(1)}(\overline{\mu}_{ij} - \mu_{ij})^2, \quad \forall (i,j) \in \mathcal{A}, \tag{5.2}$$

and

$$c_{ij}^{(2)}(\sigma_{ij}) = a_{ij}^{(2)}(\overline{\sigma}_{ij} - \sigma_{ij}) + b_{ij}^{(2)}(\overline{\sigma}_{ij} - \sigma_{ij})^2, \quad \forall (i,j) \in \mathcal{A}, \tag{5.3}$$

where $a_{ij}^{(1)}, a_{ij}^{(2)}, b_{ij}^{(1)}$ and $b_{ij}^{(2)}$ are given nonnegative real numbers. These cost functions are chosen so that: (a) the costs are 0 under the normal mean and standard deviation denoted by $\overline{\mu}_{ij}$ and $\overline{\sigma}_{ij}$; and (b) the costs are convex decreasing functions of μ_{ij} and $\sigma_{i,j}$ respectively. In our tests, we compare the distributionally robust project crashing solution with the following models:

1. Deterministic PCP: In this model, we simply use the mean value of the random activity durations as the deterministic activity durations and ignore the variability. The crashing solution in this case is given as:

$$\min_{(\boldsymbol{\mu}, \boldsymbol{\sigma}) \in \boldsymbol{\Omega} : \boldsymbol{\sigma} = \bar{\boldsymbol{\sigma}}} \ \max_{\boldsymbol{x} \in \mathcal{X} \cap \{0,1\}^m} \sum_{(i,j)\in\mathcal{A}} \mu_{ij} x_{ij}. \tag{5.4}$$

2. Heuristic PCP: [37] developed a heuristic approach to minimize the quantile of the makespan by using a surrogate deterministic objective function with activity durations defined as:

$$d_{ij} = \mu_{ij} + \kappa \cdot \sigma_{ij},$$

with fixed margin coefficients $\kappa \ge 0$. They then solved the following deterministic model to develop a heuristic solution for project crashing:

$$\min_{(\boldsymbol{\mu}, \boldsymbol{\sigma}) \in \boldsymbol{\Omega}} \ \max_{\boldsymbol{x} \in \mathcal{X} \cap \{0,1\}^m} \sum_{(i,j)\in\mathcal{A}} (\mu_{ij} + \kappa \cdot \sigma_{ij}) x_{ij}. \tag{5.5}$$

In the numerical tests, we set $\kappa = 3$.

3. Sample Average Approximation (SAA): Consider the case where the activity duration vector $\tilde{\boldsymbol{t}}$ is a multivariate normal random vector $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and the covariance matrix of the random vector $\tilde{\boldsymbol{t}}$, respectively. Then $\tilde{\boldsymbol{t}} = \boldsymbol{\mu} + \mathrm{Diag}(\boldsymbol{\sigma})\tilde{\boldsymbol{\xi}}$, where $\boldsymbol{\sigma}$ is a vector of standard deviations of $\tilde{t}_{ij}, (i,j) \in \mathcal{A}$ and $\tilde{\boldsymbol{\xi}} \sim N(\boldsymbol{0}, \boldsymbol{\rho})$ is a normally distributed random vector where $\boldsymbol{\rho}$ is the correlation matrix of $\tilde{\boldsymbol{\xi}}$ and hence $\tilde{\boldsymbol{t}}$. Let $\boldsymbol{\xi}^{(k)}, k = 1, 2, \ldots, N$, denote a set of independent and identical samples of the random vector $\tilde{\boldsymbol{\xi}}$. The SAA formulation is then given as:

$$\min_{(\boldsymbol{\mu}, \boldsymbol{\sigma}) \in \boldsymbol{\Omega}} \frac{1}{N} \sum_{k=1}^{N} \left(\max_{\boldsymbol{x} \in \mathcal{X}} \left(\boldsymbol{\mu} + \mathrm{Diag}(\boldsymbol{\sigma}) \boldsymbol{\xi}^{(k)} \right)^T \boldsymbol{x} \right), \tag{5.6}$$

which is equivalent to the optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\mu},\boldsymbol{\sigma},\boldsymbol{y}^{(k)}} \quad & \frac{1}{N}\sum_{k=1}^{N}\left(y_n^{(k)} - y_1^{(k)}\right) \\ \text{s.t.} \quad & y_j^{(k)} - y_i^{(k)} \geq \mu_{ij} + \sigma_{ij}\xi_{ij}^{(k)}, \quad \forall k = 1,\ldots,N, \\ & (\boldsymbol{\mu},\boldsymbol{\sigma}) \in \boldsymbol{\Omega}. \end{aligned} \tag{5.7}$$

In our experiments, we use a sample size of $N = 5000$.

Since the feasible region $\boldsymbol{\Omega}$ in (5.1) is quadratic and convex, all the three formulations are solvable as convex quadratic programming problems.

Example 1

In the first example, we consider the small project network in Figure 3 by [57]. This example with eight activities (arcs) and six paths is previously used by [5] to illustrate the importance of identifying the critical arcs correctly. Using simulated data, the authors show that the deterministic approach does not necessarily identify the right set of critical arcs. We will continue this discussion by illustrating the difference in the crashing solutions from the different models. We will observe that as a result of not being able to identify correct arcs to crash, the deterministic approach to PCP fails to reduce the expected makespan as much as the robust models that we propose.

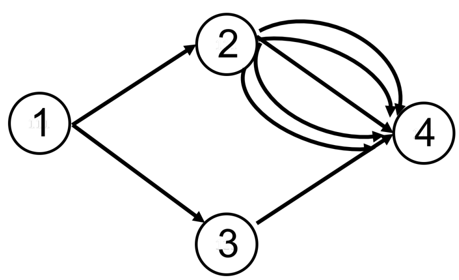


Figure 3: Small Project Network by [57]

The means for the original activity durations are set to 10 for all arcs, except that of arc (1,3) which is instead set to 10.2. The standard deviations of all arcs are set to 2. We also assume that the minimum value of the mean and standard deviation that can be achieved by crashing is half of the mean and standard deviation of the original activity durations. We set the correlation between arcs (1,2) and (1,3) to be 0.5, while the correlation matrix for the arcs emanating from node 2 are generated randomly. To apply the CMM approach for this small network, we convert the network to a parallel network by path enumeration. The makespan in this case is equal to $\max(\tilde{t}_{12}+\tilde{t}^1_{24}, \tilde{t}_{12}+\tilde{t}^2_{24}, \tilde{t}_{12}+\tilde{t}^3_{24}, \tilde{t}_{12}+\tilde{t}^4_{24}, \tilde{t}_{12}+\tilde{t}^5_{24}, \tilde{t}_{13}+\tilde{t}_{34})$[1]. For the SAA approach, the correlations between

[1]Here, we slightly abuse the notation as denote the activity time of the k^{th} arc between nodes 2 and 4 as $\tilde{t}^k_{24}$.

the activity durations originating from different nodes is set to 0. As the cost budget B varied, the time cost trade-off of the six models: Deterministic PCP (5.4), Heuristic PCP (5.5), SAA (5.7), and the Distributionally Robust PCP under MMM, CMM and NMM is shown in Figure 4 for both the linear (setting $b_{ij}^{(1)} = b_{ij}^{(2)} = 0$) and quadratic cost function case.

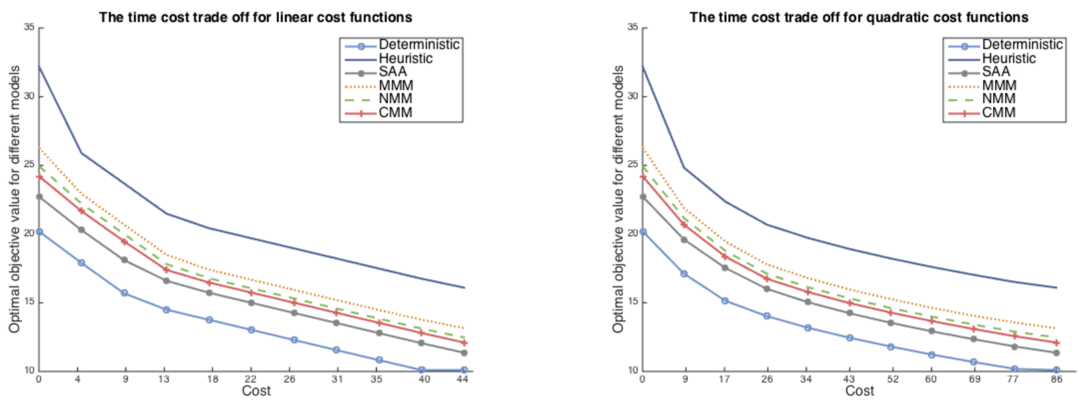


Figure 4: Optimal objective value as cost budget increases

From Figure 4, we see that all the makespans decrease as the cost budget increases as should be expected. The deterministic PCP always has the smallest objective, since its objective is the lower bound for the expected makespan by Jensen's inequality. The objective value of the distributionally robust PCP is a tight upper bound for the expected makespan under MMM, CMM and NMM. We also find that the objective value of the distributionally robust PCP is fairly close to the expected makespan under normal distribution, and by using more correlation information the objective value is closer to the expected makespan under SAA. However, the objective value of the heuristic PCP is much larger than the objective values of other models implying it is a poor approximation of the expected makespan.

Next, we compare the expected makespan under the assumption of a normal distribution using the crashed solution of these models. Using the optimal $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ obtained by the tested models, and assuming that the activity duration vector follows a multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma} = \text{Diag}(\boldsymbol{\sigma})\boldsymbol{\rho}\text{Diag}(\boldsymbol{\sigma})$, we compute the expected makespan by a Monte Carlo simulation with 10000 samples. The results are shown in Figure 5. As expected, using the optimal solution obtained by SAA, we get the smallest expected makespan. When the cost budget is small, the deterministic PCP also provides a reasonable decision with small expected makespan, but this model is not robust. When the budget is large, we observe that the expected makespan of the deterministic model is much larger than the expected makespans of other models. The solutions obtained by the distributionally robust models are very close to the expected makespan of SAA.

We now provide a comparison of the optimal solutions from the various models. Assume the cost function is quadratic and the cost budget $B = 17.18$. In this case the expected makespans are 17.64 (MMM), 17.63 (NMM), 17.61 (CMM) and 17.53 (SAA).The optimal solutions of the four models are shown in Figure 6. The optimal

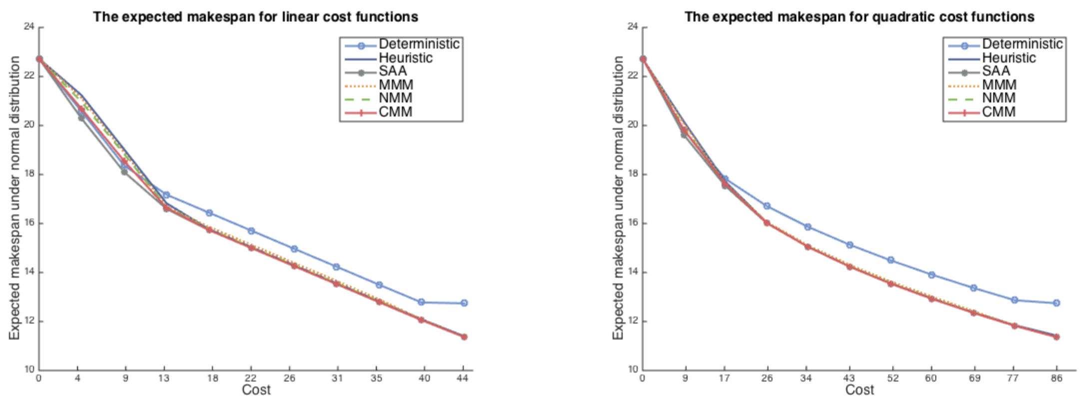
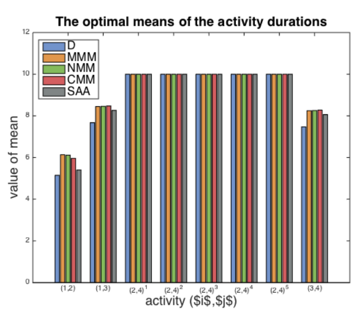
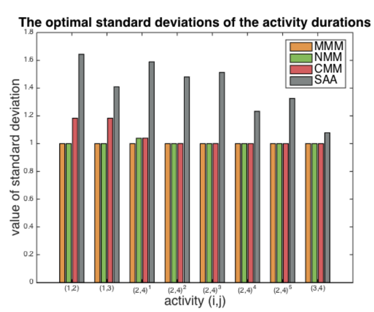


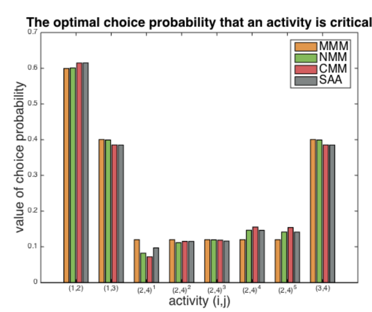
Figure 5: The expected makespan under the multivariate normal distribution



(a) Optimal mean



(b) Optimal standard deviation

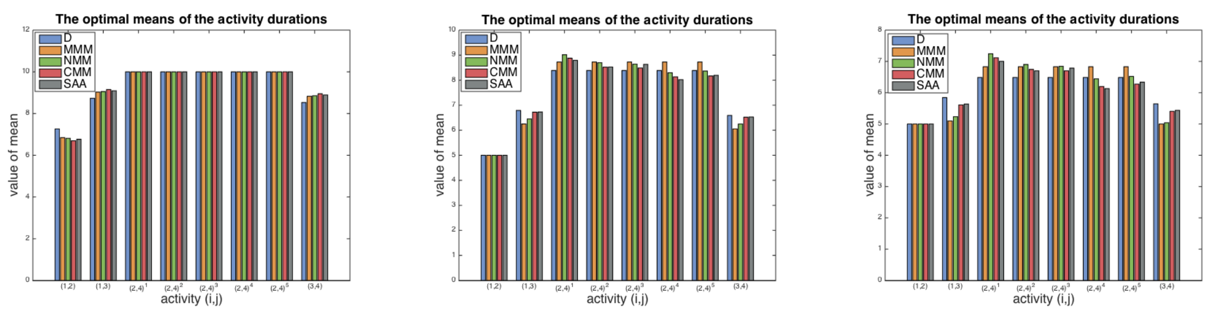


(c) Optimal choice probability

Figure 6: Optimal solutions obtained by MMM, NMM, CMM and SAA

solutions from the four models are fairly close. We observe that SAA tends to crash the mean more while the robust models tend to crash the standard deviation more. We validate this observation with a larger numerical example later. We also observe that the deterministic solution crashes the mean of arc (1,3), while the robust models invest in crashing the standard deviations of the multiple arcs between nodes 2 and 4. The robust models are also observed to be successful in identifying the critically of the arcs.

Finally, we present the optimal mean values obtained by the deterministic PCP model and the three distributionally robust PCP models in Figure 7 where the budget is set to $B = 8$ (left), $B = 32$ (center), $B = 56$ (right); and we allow crashing only mean activity times, i.e., standard deviations are fixed at 2 for all arcs. We observe that when the budget is low, the deterministic model decreases the means of arcs (1,3) and (3,4) since they have been mistakenly identified as the critical arcs. On the other hand, the robust models recognize the importance of arc (1,2) and crash its mean. When the budget is abundant, we observe that the crashing decisions obtained by the CMM model is very close to those from SAA, while the deterministic solution does not have the flexibility to crash the arcs between nodes 2 and 4 appropriately.



(a) Low budget

(b) Medium budget

(c) High budget

Figure 7: Optimal solutions obtained by MMM, NMM, CMM and SAA

Example 2 (Parallel Network)

In this example, we consider a project with k parallel activities. The data is randomly generated as follows:

1. For every activity, the means and the standard deviations of the original activity duration are generated by uniform distributions $\overline{\mu}_{ij} \sim U(10, 20)$, $\overline{\sigma}_{ij} \sim U(6, 10)$, and the minimal values of mean and standard deviation that can be obtained by crashing are $\underline{\mu}_{ij} \sim U(5, 10)$, $\underline{\sigma}_{ij} \sim U(2, 6)$.

2. The coefficients in the cost function (5.2) are chosen as follows $a_{ij}^{(1)} \sim U(1, 2)$, $b_{ij}^{(1)} \sim U(0, 1)$, $a_{ij}^{(1)} \sim U(1, 2)$ and $b_{ij}^{(2)} \sim U(0, 1)$. The amount of the cost budget is chosen as $\frac{1}{4}\sum_{(i,j)}(c_{ij}^{(1)}(\underline{\mu}_{ij}) + c_{ij}^{(2)}(\underline{\sigma}_{ij}))$.

We first consider a simple case with two parallel activities. In Figure 8, we plot the optimal values of f_{cmm} as the correlation between the two activity durations increases from -1 to 1, and compare these values with the optimal value of f_{mmm} for one such random instance. The worst-case expected project makespan f_{cmm} is a decreasing

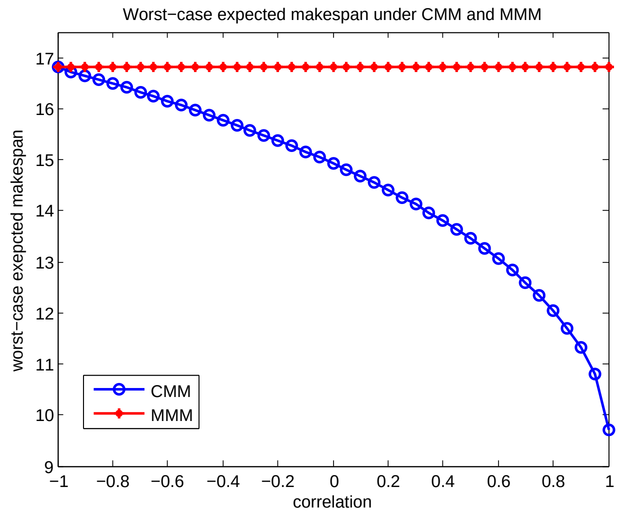


Figure 8: Optimal value of f_{cmm} and f_{mmm}

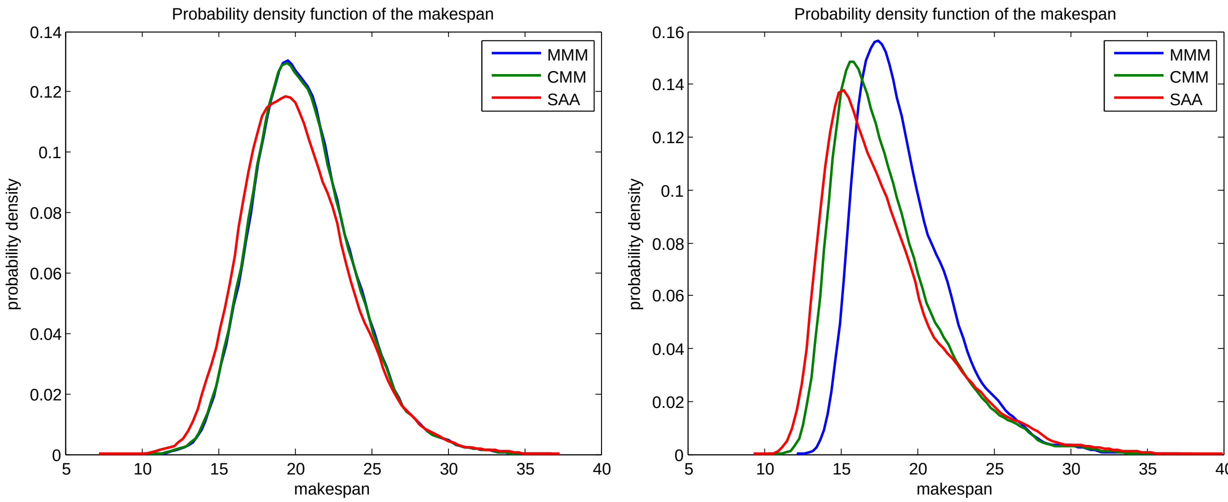
function of the correlation ρ, and when $\rho = -1$ (perfectly negatively correlated), the worst-case expected makespan under CMM and MMM are the same. Clearly if the activity durations are positively correlated, then the bound from capturing correlation information is much tighter.

Next, we consider a parallel network with 10 activities. We compute the optimal values of the crashed moments $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ under the MMM, CMM and SAA models and compute the expected makespans under these moments by assuming that the activity durations follows a multivariate normal distribution $\mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$. We consider two types of instances - one with uncorrelated activity durations and the other with highly correlated activity durations. The distribution of the makespan and the statistics are provided in Figure 9 and Table 1. When the activities are uncorrelated, with the optimal solution obtained from MMM and CMM, the distribution of the makespan is very close. However when the activities are highly correlated, the distribution is farther apart. As should be expected, SAA provides the smallest expected makespan under the normal distribution. However, the maximum value and standard deviation of the makespan obtained by SAA are the largest in comparison to the distributionally robust models indicating that the robust models provide a reduction in the variability of the makespan.

Table 1: Statistics of the makespan

Makespan Statistics	Uncorrelated activities			Highly correlated activities		
	MMM	CMM	SAA	MMM	CMM	SAA
Min	11.9012	11.7435	9.1955	13.744	12.0311	11.2626
Max	34.9514	35.0389	35.3919	35.5185	36.1969	38.0104
Mean	20.5953	20.5539	20.1615	19.167	18.056	17.885
Median	20.312	20.2567	19.832	18.5274	17.2832	17.0001
Std deviation	3.1807	3.1939	3.4344	3.0475	3.4391	3.8575

Finally, we compare the CPU time of solving the SDP reformulation (3.17) and the saddle point reformulation (3.18) for the distributionally robust PCP under CMM. To solve the SDP (3.17), we used CVX, a package for



(a) Uncorrelated activity durations

(b) Highly correlated activity durations

Figure 9: Probability density function of the makespan

specifying and solving convex programs ([28, 27]) with the solver SDPT3 ([55, 56]). We set the accuracy of the SDP solver with "cvx_precision low". To solve the saddle point problem (3.18), we use Algorithm 2 with tolerance $\epsilon = 10^{-3}$. In our implementation, we used the means and standard deviations obtained from MMM by solving a SOCP as a warm start for Algorithm 2. The average value of the CPU time and the objective function for 10 randomly generated instances is provided in Table 2. For large instances, it is clear that using the saddle point algorithm to solve the problem to reasonable accuracy for all practical purposes is much faster than solving the semidefinite program formulation of the problem with interior point method solvers.

Table 2: CPU time of the SDP solver and Algorithm 2

k	CPU time in seconds		Objective value	
	SDP formulation	Saddle point formulation	SDP formulation	Saddle point formulation
20	0.81	4.47	31.642	31.643
40	3.36	9.06	39.273	39.274
60	22.73	37.57	45.909	45.910
80	77.53	71.42	50.400	50.401
100	255.54	169.65	54.261	54.261
120	685.56	297.57	58.541	58.542
140	1749.25	458.60	62.577	62.579
160	**	568.91	**	65.025
180	**	810.99	**	68.581
200	**	1255.38	**	70.919

** means the instances cannot be solved in 2 hours by the SDP solver.

Example 3 (Grid Network)

In this third example, we consider a grid network (see Figure 10). The size of the problem is determined by its width and height. Let width=w and height=h in which case there are $n = (w+1)(h+1)$ nodes, $m = w(h+1) + h(w+1)$ activities and $\binom{w+h}{w}$ possible critical paths in the project. For example, in Figure 10, $w = 6$ and $h = 4$, then there are 35 nodes, 58 activities and 210 possible critical paths.

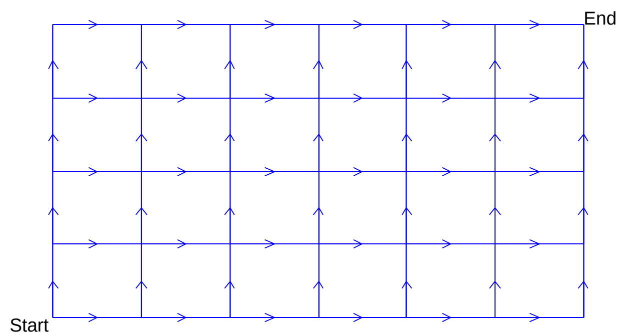


Figure 10: Grid project network with width = 6, height=4

We test the distributionally robust project crashing models with randomly generated data. The data is chosen as follows:

1. For every activity $(i,j) \in \mathcal{A}$, the mean and the standard deviation of the original activity duration are generated by uniform distributions $\overline{\mu}_{ij} \sim U(5,10)$, $\overline{\sigma}_{ij} \sim U(4,8)$, and the minimal values of mean and standard deviation that can be obtained by crashing are chosen as $\underline{\mu}_{ij} \sim U(2,\overline{\mu}_{ij})$, $\underline{\sigma}_{ij} \sim U(1,\overline{\sigma}_{ij})$.

2. For the coefficients in the cost function (5.2), we choose $a_{ij}^{(1)} \sim U(2,4)$, $b_{ij}^{(1)} \sim U(0,1)$, $a_{ij}^{(2)} \sim U(1,2)$ and $b_{ij}^{(2)} \sim U(0,1)$ for all $(i,j) \in \mathcal{A}$.

3. The amount of the cost budget is chosen as $\sum_{(i,j)} a_{ij}^{(1)}(\overline{\mu}_{ij} - \underline{\mu}_{ij}) + a_{ij}^{(2)}(\overline{\mu}_{ij} - \underline{\mu}_{ij})^2$. In the deterministic model (5.4), an optimal strategy is to reduce the means of every activity duration to its lower bound without the change of variance.

4. For simulations, the activities are assumed to be independent which implies that the correlation matrix for the activity durations is an identity matrix.

Both the deterministic PCP (5.4) and the heuristic PCP (5.5) can be formulated as convex quadratic programs which can be quickly solved. Solving the distributionally robust PCP and SAA are more computationally expensive. We compare the expected makespan with the optimal solutions obtained by the Deterministic PCP (5.4), Heuristic PCP (5.5), SAA and the distributionally robust PCP under MMM and NMM. Let $\mathbb{E}(T_0)$ denote the expected makespan without crashing, and $\mathbb{E}(T_1)$ denote the expected makespan with crashing by the deterministic model (5.4). We define the "reduction" as the percentage of the extra expected makespan achieved by

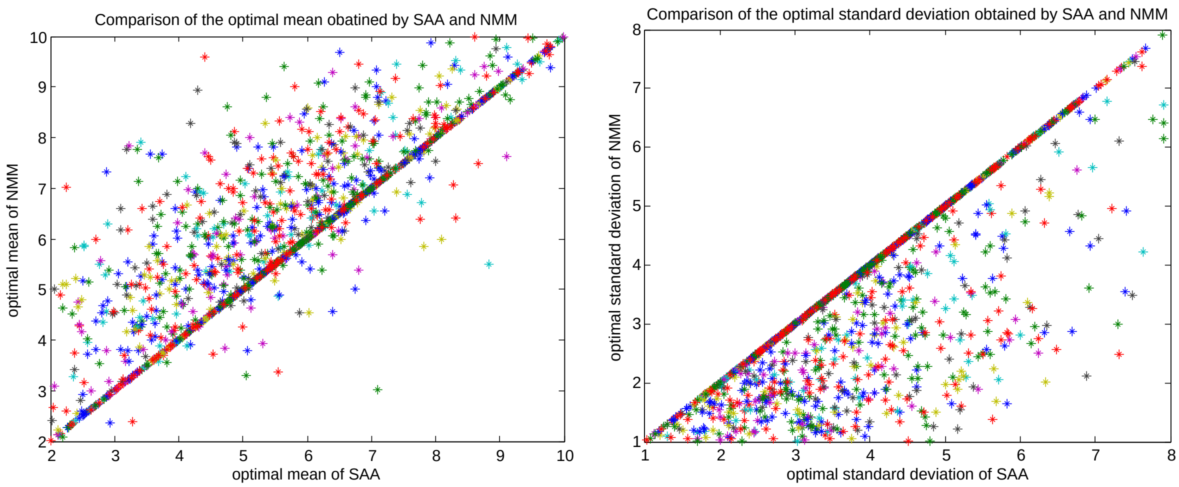


Figure 11: Optimal solution comparison of SAA and NMM

the other models, that is

$$100 \cdot \left(\frac{\mathbb{E}(T_0) - \mathbb{E}(T_{new})}{\mathbb{E}(T_0) - \mathbb{E}(T_1)} - 1 \right),$$

where $\mathbb{E}(T_{new})$ is the expected makespan with crashed activity durations obtained from a project crashing model (Heuristic, SAA, MMM or NMM).

The numerical results presented in Table 3 are the average of 10 randomly generated instances. With the crashed activity durations, we compare the expected makespan under four different distributions including normal, uniform, gamma and the worst-case distribution in NMM. The expected makespans achieved by the distributionally robust PCP are always smaller than the deterministic and heuristic PCP models. The improvement of the distributionally robust PCP is much larger than the improvement of the heuristic PCP under all the distributions. In comparison with SAA, we see that the expected makespans obtained by MMM and NMM are bigger than the expected makespans obtained by SAA under normal, uniform and gamma distributions. However, the gaps are quite small and under the worst-case distribution the expected makespan of NMM and MMM are always smaller than the expected makespan of SAA. Moreover, we find that the computational time of solving MMM and NMM is smaller than solving SAA. Between MMM and NMM, the additional information in this graph is the correlation between each pair of activities originated from a node. Due to the grid network structure, we find that the optimal solutions between MMM and NMM are much closer in comparison to the parallel graph in Example 2. We also compare the optimal crashing decisions obtained by SAA and the distributionally robust PCP. The results for all 10 instances for the 8×8 gird network are plotted in Figure 11. From the figure, we see that the SAA model tends to crash the means more while NMM tends to crash the standard deviations more.

Table 3: Expected makespan and reduction

size	models	objective | time	expected makespan				reduction (%)			
			normal	uniform	gamma	worst-case	normal	uniform	gamma	worst-case
2 × 1 grid 7 arcs	Deterministic	14.30 | 0.44	20.12	20.22	19.56	30.07	-	-	-	-
	Heuristic	45.92 | 0.52	21.41	21.46	21.41	26.76	-14.44	-13.89	-19.85	38.92
	MMM	26.04 | 0.72	19.47	19.55	19.36	25.10	8.28	8.55	3.01	59.36
	NMM	25.08 | 0.82	19.40	19.48	19.27	25.08	9.10	9.35	3.84	59.69
	SAA	19.00 | 42.34	19.00	19.08	18.80	25.84	13.75	14.04	8.93	50.54
2 × 2 grid 12 arcs	Deterministic	18.37 | 0.45	28.90	29.03	28.46	45.91	-	-	-	-
	Heuristic	59.95 | 0.61	29.06	29.12	29.15	38.43	-1.21	-0.50	-5.56	64.44
	MMM	37.71 | 0.97	27.32	27.39	27.32	36.73	14.86	15.45	10.51	80.37
	NMM	36.71 | 1.14	27.28	27.35	27.28	36.71	15.14	15.74	10.74	80.50
	SAA	26.63 | 66.27	26.63	26.70	26.63	38.10	20.49	21.07	15.87	68.86
3 × 3 grid 24 arcs	Deterministic	27.72 | 0.54	46.14	46.11	46.09	79.22	-	-	-	-
	Heuristic	90.20 | 0.83	44.48	44.48	44.85	63.39	9.96	9.91	7.41	92.88
	MMM	61.74 | 1.43	42.52	42.51	42.76	60.74	23.06	22.98	20.92	109.99
	NMM	60.73 | 2.50	42.52	42.51	42.74	60.73	23.07	22.99	21.02	110.04
	SAA	41.39 | 128.06	41.39	41.39	41.90	63.15	29.46	29.37	25.64	96.37
4 × 3 grid 31 arcs	Deterministic	33.00 | 0.62	54.83	54.89	55.67	97.10	-	-	-	-
	Heuristic	106.59 | 0.96	52.67	52.67	53.29	77.17	11.37	11.66	12.51	102.54
	MMM	74.66 | 1.68	50.30	50.34	50.77	73.65	25.31	25.38	27.18	122.39
	NMM	73.64 | 2.98	50.31	50.35	50.79	73.64	25.25	25.33	27.10	122.42
	SAA	48.89 | 163.02	48.89	48.91	49.77	76.83	32.44	32.59	32.15	106.45
6 × 4 grid 58 arcs	Deterministic	47.63 | 0.85	81.31	81.17	83.64	156.23	-	-	-	-
	Heuristic	153.50 | 1.49	76.91	76.82	78.37	121.34	16.87	16.57	19.93	129.10
	MMM	116.44 | 2.83	73.60	73.52	74.66	115.41	31.11	30.80	35.74	152.91
	NMM	115.41 | 4.58	73.62	73.55	74.65	115.41	31.04	30.72	35.80	152.92
	SAA	71.45 | 388.74	71.45	71.26	73.14	120.94	39.16	39.27	41.29	132.79
6 × 6 grid 84 arcs	Deterministic	55.77 | 1.07	99.36	99.19	103.30	202.86	-	-	-	-
	Heuristic	181.39 | 1.97	92.47	92.37	94.55	153.48	21.42	21.17	27.97	150.56
	MMM	147.81 | 3.90	89.12	89.04	90.86	146.78	33.88	33.67	41.66	172.96
	NMM	146.77 | 6.28	89.15	89.07	90.84	146.78	33.80	33.58	41.78	172.99
	SAA	86.21 | 594.79	86.21	86.05	88.81	154.36	42.87	42.89	48.08	150.41
8 × 6 grid 110 arcs	Deterministic	66.33 | 1.31	117.60	117.56	123.52	249.05	-	-	-	-
	Heuristic	214.57 | 2.67	109.63	109.51	112.37	187.98	22.04	22.32	31.54	163.40
	MMM	180.23 | 5.13	105.34	105.30	107.57	179.20	35.77	35.88	47.10	189.01
	NMM	179.19 | 8.08	105.37	105.33	107.58	179.19	35.69	35.80	47.09	189.03
	SAA	101.82 | 880.33	101.82	101.65	105.34	188.92	45.51	46.01	53.39	163.36
8 × 8 grid 144 arcs	Deterministic	74.47 | 1.62	135.59	135.47	143.48	301.01	-	-	-	-
	Heuristic	242.32 | 3.50	125.20	125.03	128.78	223.20	24.65	24.79	36.44	180.68
	MMM	214.44 | 6.77	120.64	120.50	123.74	213.40	37.73	37.89	51.21	205.70
	NMM	213.40 | 10.43	120.67	120.54	123.78	213.40	37.66	37.81	51.16	205.73
	SAA	116.41 |1385.43	116.41	116.11	120.96	225.59	47.87	48.44	58.12	177.91
10 × 10 grid 220 arcs	Deterministic	93.18 | 2.50	172.33	171.67	183.83	409.65	-	-	-	-
	Heuristic	303.30 | 5.95	158.44	157.90	163.43	299.44	26.64	26.39	41.51	207.00
	MMM	287.21 | 10.82	152.56	152.06	156.98	286.18	40.43	40.17	57.00	234.56
	NMM	286.17 | 16.88	152.59	152.09	157.06	286.17	40.38	40.11	56.87	234.60
	SAA	146.78 |2657.14	146.78	146.14	153.16	303.58	51.79	51.83	64.90	202.32

Example 4 (Random Networks)

Our final test aims to demonstrate that using Algorithm 2 to solve the saddle point reformulation (3.24) is significantly faster than solving the SDP reformulation for the distributionally robust PCP under NMM model. We have generated random project networks with a given number of nodes ($n = 50, 100, 200$) and maximum outdegree ($d = 3, .., 10$). The means and standard deviations of the activity durations, their respective minimal values, and the crashing cost functions are randomly generated as in Example 2. For each node, a random correlation matrix is generated using the 'randcorr' function given in [51]. For each problem size, i.e., (n, d), we have solved 10 different instances of the problem with 11 different crashing budgets varying from 0 to the maximum budget (we define the maximum budget as the minimum amount needed to crash all activities to their minimum values). We present a summary of the results in Table 4, where each row reports the number of nodes (n), maximum outdegree (d) , average number of arcs (m), i.e., number of activities to be crashed, the average CPU time to solve 110 instances corresponding to a given (n, d) pair with the SeDuMi solver in CVX package, and the average CPU time to solve the same instances with Algorithm 2 implemented in Matlab. We observe that the saddle point reformulations are preferable to the SDP formulations especially for networks with large number of arcs. The objective function values obtained by the two formulations are very close to each other and are omitted from the table. To our surprise, the well-known SDP solver SDPT3 has failed in a large number of instances, while SeDuMi was more reliable. Even with SeDuMi solver, we have encountered some problem instances which didn't return a solution, these were all solved by the saddle-point algorithm without any issues. We have eliminated these instances from the computational study. This provides further evidence that the saddle-point algorithms are superior to the SDP reformulations.

Table 4: CPU time of the SDP solver and Algorithm 2

n	d	m	CPU time in seconds	
			SDP formulation	Saddle point formulation
50	3	87.0	4.60	1.22
	4	105.9	6.50	1.20
	5	124.8	7.81	2.12
	6	144.6	10.17	3.35
	7	163.3	11.88	4.91
	8	182.8	13.81	5.30
	9	203.1	15.83	9.20
	10	221.6	21.63	10.87
100	3	173.6	12.25	1.74
	4	212.6	18.35	1.95
	5	253.6	26.35	5.90
	6	293.1	30.33	13.57
	7	334.6	46.53	14.83
	8	371.6	53.69	25.96
	9	411.6	61.57	30.23
	10	459.3	84.20	42.67
200	3	347.6	177.19	10.79
	4	427.7	55.87	9.67
	5	507.9	87.95	22.47
	6	586.6	177.19	45.46

6 Conclusions

In this paper, we proposed a class of distributionally robust project crashing problems that is solvable in polynomial time where the objective is to choose the first two marginal moments to minimize the worst-case expected project makespan. While semidefinite programming is the typical approach to tackle such problems, we provide an alternative saddle point reformulation over the moment and arc criticality index variables which helps us to use alternative methods to solve the problem. Numerical experiments show that this can help us solve larger instances of such problems. Furthermore, in terms of insights the robust models tend to crash the standard deviations more in comparison with the sample average approximation for standard distributions such as the mulivariate normal distribution.

We believe there are several ways to build on this work. Given several developments that have occurred in first order methods for saddle point formulations in the recent years, we believe more can be done to apply these methods to solve distributionally robust optimization problems. To the best of knowledge, little has been done in this area thus far. Another research direction is to identify new instances where the distributionally robust project crashing problem is solvable in polynomial time. Lastly it would be interesting if these results can be used to find approximation guarantees for the general distributionally robust project crashing problem with arbitrary correlations.

Acknowledgement

The research of the second author was partially supported by the MOE Academic Research Fund Tier 2 grant T2MOE1706, "On the Interplay of Choice, Robustness and Optimization in Transportation" and the SUTD-MIT International Design Center grant IDG21700101 on "Design of the Last Mile Transportation System: What Does the Customer Really Want?".

Appendix

Proof of Proposition 2

We consider the inner maximization problem of (3.22), which is to compute the worst-case expected duration of the project with given mean, standard deviation and partial correlation information of the activity durations under the nonoverlapping structure. We denote it by

$$\phi_{\text{nmm}}(\boldsymbol{\mu}, \boldsymbol{\sigma}) = \max_{\theta \in \Theta_{\text{nmm},\boldsymbol{\rho}_i,\forall i}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i, \forall i)} \mathbb{E}_\theta \left(\max_{\boldsymbol{x} \in \mathcal{X}} \sum_{i=1}^{n-1} \tilde{t}_i^T \boldsymbol{x}_i \right). \tag{6.1}$$

Applying Theorem 15 on page 467 in [40], the worst-case expected makespan in (6.1) is formulated as the following SDP:

$$\begin{aligned} \phi_{\text{nmm}}(\boldsymbol{\mu}, \boldsymbol{\sigma}) = \quad & \max_{x_{ij}, \boldsymbol{w}_{ij}, \boldsymbol{W}_{ij}} \quad \sum_{(i,j) \in \mathcal{A}} \boldsymbol{e}_{ij}^T \boldsymbol{w}_{ij} \\ & \text{s.t.} \quad \boldsymbol{x} \in \mathcal{X}, \\ & \qquad \begin{pmatrix} 1 & \boldsymbol{\mu}_i^T \\ \boldsymbol{\mu}_i & \boldsymbol{\Sigma}_i + \boldsymbol{\mu}_i \boldsymbol{\mu}_i^T \end{pmatrix} - \sum_{(i,j) \in \mathcal{A}_i} \begin{pmatrix} x_{ij} & \boldsymbol{w}_{ij}^T \\ \boldsymbol{w}_{ij} & \boldsymbol{W}_{ij} \end{pmatrix} \succeq 0, \quad \forall i = 1, \ldots, n-1, \\ & \qquad \begin{pmatrix} x_{ij} & \boldsymbol{w}_{ij}^T \\ \boldsymbol{w}_{ij} & \boldsymbol{W}_{ij} \end{pmatrix} \succeq 0, \qquad \forall (i,j) \in \mathcal{A}. \end{aligned} \tag{6.2}$$

To show the result of Proposition 2, we need the following two lemmas:

Lemma 1. *The SDP problem (6.2) can be simplified as*

$$\begin{aligned} \max_{\boldsymbol{x}, \boldsymbol{Y}_i} \quad & \sum_{i=1}^{n-1} \text{trace}(\boldsymbol{Y}_i) \\ s.t. \quad & \boldsymbol{x} \in \mathcal{X}, \\ & \begin{pmatrix} \boldsymbol{\Sigma}_i + \boldsymbol{\mu}_i \boldsymbol{\mu}_i^T & \boldsymbol{Y}_i^T & \boldsymbol{\mu}_i \\ \boldsymbol{Y}_i & \text{Diag}(\boldsymbol{x}_i) & \boldsymbol{x}_i \\ \boldsymbol{\mu}_i^T & \boldsymbol{x}_i^T & 1 \end{pmatrix} \succeq 0, \quad \forall i = 1, \ldots, n-1. \end{aligned} \tag{6.3}$$

Proof. First, we show the optimal value of (6.2) $\leq$ the optimal value of (6.3). Consider an optimal solution to the SDP (6.2) denoted by $(x_{ij}^*, \boldsymbol{w}_{ij}^*, \boldsymbol{W}_{ij}^*)$ for $(i,j) \in \mathcal{A}$. Let $\boldsymbol{x} = \boldsymbol{x}^*$ and $\boldsymbol{Y}_i^T \boldsymbol{e}_{ij} = \boldsymbol{w}_{ij}^*$ for all $(i,j) \in \mathcal{A}$. Then

$\text{trace}(\boldsymbol{Y}_i) = \sum_{(i,j)\in\mathcal{A}_i} \boldsymbol{e}_{ij}^T \boldsymbol{w}_{ij}^*$, which implies

$$\sum_{i=1}^{n-1} \text{trace}(\boldsymbol{Y}_i) = \sum_{(i,j)\in\mathcal{A}} \boldsymbol{e}_{ij}^T \boldsymbol{w}_{ij}^*.$$

Next we verify that $\boldsymbol{x}_i, \boldsymbol{Y}_i, i = 1, \ldots, n-1$ is feasible for (6.3). For an $i = 1, \ldots, n-1$, we consider the case with all the x_{ij} values being strictly positive first. In this case

$$\begin{aligned}
&\begin{pmatrix} \boldsymbol{\Sigma}_i + \boldsymbol{\mu}_i\boldsymbol{\mu}_i^T & \boldsymbol{\mu}_i \\ \boldsymbol{\mu}_i^T & 1 \end{pmatrix} - \begin{pmatrix} \boldsymbol{Y}_i^T \\ \boldsymbol{x}_i^T \end{pmatrix} \text{Diag}(\boldsymbol{x}_i)^{-1} \begin{pmatrix} \boldsymbol{Y}_i & \boldsymbol{x}_i \end{pmatrix} \\
= &\begin{pmatrix} \boldsymbol{\Sigma}_i + \boldsymbol{\mu}_i\boldsymbol{\mu}_i^T - \boldsymbol{Y}_i^T \text{Diag}(\boldsymbol{x}_i)^{-1}\boldsymbol{Y}_i & \boldsymbol{\mu}_i - \boldsymbol{Y}_i^T\boldsymbol{1} \\ \boldsymbol{\mu}_i^T - \boldsymbol{1}^T\boldsymbol{Y}_i & 1 - \boldsymbol{1}^T\boldsymbol{x}_i \end{pmatrix} \\
= &\begin{pmatrix} \boldsymbol{\Sigma}_i + \boldsymbol{\mu}_i\boldsymbol{\mu}_i^T - \sum_{(i,j)\in\mathcal{A}_i} \frac{\boldsymbol{w}_{ij}^*\boldsymbol{w}_{ij}^{*\,T}}{x_{ij}} & \boldsymbol{\mu}_i - \sum_{(i,j)\in\mathcal{A}_i} \boldsymbol{w}_{ij}^* \\ (\boldsymbol{\mu}_i - \sum_{(i,j)\in\mathcal{A}_i} \boldsymbol{w}_{ij}^*)^T & 1 - \boldsymbol{1}^T\boldsymbol{x}_i \end{pmatrix} \\
\succeq &\begin{pmatrix} \boldsymbol{\Sigma}_i + \boldsymbol{\mu}_i\boldsymbol{\mu}_i^T - \boldsymbol{W}_{ij}^* & \boldsymbol{\mu}_i - \sum_{(i,j)\in\mathcal{A}_i} \boldsymbol{w}_{ij}^* \\ (\boldsymbol{\mu}_i - \sum_{(i,j)\in\mathcal{A}_i} \boldsymbol{w}_{ij}^*)^T & 1 - \sum_{(i,j)\in\mathcal{A}_i} x_{ij} \end{pmatrix} \\
\succeq &\ 0.
\end{aligned}$$

The last two matrix inequalities come from the feasibility condition of (6.2). The case with some of the variables $x_{ij} = 0$ is handled similarly by dropping the rows and columns corresponding to the zero entries. Thus the solution $(\boldsymbol{Y}_i, \boldsymbol{x}_i), i = 1, \ldots, n-1$ is feasible to the semidefinite program (6.3) by the Schur complement condition for positive semidefiniteness . Therefore, the optimal value of (6.2) is less than or equal to the optimal value of (6.3).

Next, we show the optimal value of (6.2) $\geq$ the optimal value of (6.3). Consider an optimal solution to (6.3) denoted by $(\boldsymbol{Y}_i^*, \boldsymbol{x}_i^*), i = 1, \ldots, n-1$. For an $i = 1, \ldots, n-1$ we consider the case x_{ij}^* are all positive for $(i,j) \in \mathcal{A}_i$. From Schur's complement, the positive semidefiniteness constraint in (6.3) is equivalent to:

$$\begin{pmatrix} \boldsymbol{\Sigma}_i + \boldsymbol{\mu}_i\boldsymbol{\mu}_i^T - \boldsymbol{Y}_i^{*T} \text{Diag}(\boldsymbol{x}_i^*)^{-1}\boldsymbol{Y}_i^* & \boldsymbol{\mu}_i - \boldsymbol{Y}_i^{*T}\boldsymbol{1} \\ \boldsymbol{\mu}_i^T - \boldsymbol{1}^T\boldsymbol{Y}_i^* & 1 - \boldsymbol{1}^T\boldsymbol{x}_i^* \end{pmatrix} \succeq 0,$$

Define:

$$\begin{pmatrix} \boldsymbol{W}_{ij} & \boldsymbol{w}_{ij} \\ \boldsymbol{w}_{ij}^T & x_{ij} \end{pmatrix} = \begin{pmatrix} \boldsymbol{Y}_i^{*T}\boldsymbol{e}_{ij}\boldsymbol{e}_{ij}^T\boldsymbol{Y}_i^*/x_{ij}^* & \boldsymbol{Y}_i^{*T}\boldsymbol{e}_{ij} \\ \boldsymbol{e}_{ij}^T\boldsymbol{Y}_i^* & x_{ij}^* \end{pmatrix}, \quad (i,j) \in \mathcal{A}_i.$$

Then $(\boldsymbol{W}_{ij}, \boldsymbol{w}_{ij}, \boldsymbol{x}_{ij}), (i,j) \in A$ is a feasible solution to the SDP (6.2), the objective function has the same value as the optimal objective function value of (6.3). As before, the case with some of the $x_{ij}^* = 0$ can be handled by dropping the rows and columns corresponding to the zeros. Therefore, the optimal value of (6.2) is greater than to equal to the optimal value of (6.3). □

The second lemma is a generalization of Theorem 1 from [1] and a related Theorem 4 from [49].

Lemma 2. *For a fixed $\boldsymbol{x} \geq 0$ such that $\mathbf{1}^T\boldsymbol{x} \leq 1$, consider the formulation:*

$$\begin{aligned} Z^*_{cmm}(\boldsymbol{x}) = max \quad & trace(\boldsymbol{Y}) \\ s.t. \quad & \begin{pmatrix} \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T & \boldsymbol{Y}^T & \boldsymbol{\mu} \\ \boldsymbol{Y} & Diag(\boldsymbol{x}) & \boldsymbol{x} \\ \boldsymbol{\mu}^T & \boldsymbol{x}^T & 1 \end{pmatrix} \succeq 0. \end{aligned} \tag{6.4}$$

Then,

$$Z^*_{cmm}(\boldsymbol{x}) = \quad \boldsymbol{\mu}^T\boldsymbol{x} + trace\left(\left(\boldsymbol{\Sigma}^{1/2}\ S(\boldsymbol{x})\ \boldsymbol{\Sigma}^{1/2}\right)^{1/2}\right), \tag{6.5}$$

where $S(\boldsymbol{x}) = Diag(\boldsymbol{x}) - \boldsymbol{x}\boldsymbol{x}^T \succeq 0$.

Proof. Applying Schur's lemma to the positive semidefinite matrix in (6.4), we obtain the equivalent formulation:

$$\begin{aligned} Z^*_{\mathrm{cmm}}(\boldsymbol{x}) = \max \quad & \mathrm{trace}(\boldsymbol{Y}) \\ \mathrm{s.t.} \quad & \begin{pmatrix} \boldsymbol{\Sigma} & \boldsymbol{Y}^T - \boldsymbol{\mu}\boldsymbol{x}^T \\ \boldsymbol{Y} - \boldsymbol{x}\boldsymbol{\mu}^T & \mathrm{Diag}(\boldsymbol{x}) - \boldsymbol{x}\boldsymbol{x}^T \end{pmatrix} \succeq 0. \end{aligned}$$

Note that $S(\boldsymbol{x}) = \mathrm{Diag}(\boldsymbol{x}) - \boldsymbol{x}\boldsymbol{x}^T$ is positive semidefinite under the condition that $\boldsymbol{x} \geq 0$ and $\mathbf{1}_n^T\boldsymbol{x} \leq 1$. This implies from Theorem 4 in [49] that the optimal $\boldsymbol{Y}^T - \boldsymbol{\mu}\boldsymbol{x}^T = \boldsymbol{\Sigma}\left(S(\boldsymbol{x})\right)^{1/2}(((S(\boldsymbol{x}))^{1/2}\boldsymbol{\Sigma}(S(\boldsymbol{x}))^{1/2})^{1/2})^{\dagger}\left(S(\boldsymbol{x})\right)^{1/2}$, which yields the desired result. □

Lemma 1 shows that formulation (6.2) is equal to formulation (6.3). Next, observe that for each node i, we have $\mathbf{1}^T\boldsymbol{x}_i \leq 1$ and $\boldsymbol{x}_i \geq 0$. Hence for a fixed $\boldsymbol{x} \in \mathcal{X}$, the SDP (6.3) is separable across the nodes i, implying that we can apply Lemma 2 to each node, to show that SDP (6.3) is equivalent to:

$$\max_{\boldsymbol{x}\in\mathcal{X}} \sum_{i=1}^{n-1} \left(\boldsymbol{\mu}_i^T\boldsymbol{x}_i + \mathrm{trace}\left(\left(\boldsymbol{\Sigma}_i^{1/2}S(\boldsymbol{x}_i)\boldsymbol{\Sigma}_i^{1/2}\right)^{1/2}\right)\right). \tag{6.6}$$

Therefore, the project crashing problem is equivalent to

$$\min_{(\boldsymbol{\mu},\boldsymbol{\sigma})\in\boldsymbol{\Omega}} \max_{\boldsymbol{x}\in\mathcal{X}} \sum_{i=1}^{n} \left(\boldsymbol{\mu}_i^T\boldsymbol{x}_i + \mathrm{trace}\left(\left(\boldsymbol{\Sigma}_i^{1/2}S(\boldsymbol{x}_i)\boldsymbol{\Sigma}_i^{1/2}\right)^{1/2}\right)\right), \tag{6.7}$$

where $\boldsymbol{\Sigma}_i = \mathrm{Diag}(\boldsymbol{\sigma}_i)\boldsymbol{\rho}_i\mathrm{Diag}(\boldsymbol{\sigma}_i)$ is a matrix function of $\boldsymbol{\sigma}_i$, $S(\boldsymbol{x}_i) = \mathrm{Diag}(\boldsymbol{x}_i) - \boldsymbol{x}_i\boldsymbol{x}_i^T$. The convexity of the objective function with respect to $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i^{1/2}$ and concavity with respect to the $\boldsymbol{x}_i$ variables follows naturally from Proposition 1.

Proof of Proposition 3

The gradient of the function with respect to $\boldsymbol{x}$ is derived in Theorem 4 in [1]. The gradient with respect to $\boldsymbol{\mu}$ is straightforward. We derive the expression for the gradient of f_{cmm} with respect to $\boldsymbol{\sigma}$ next. Towards this, we first characterize the gradient of the trace function $f(\boldsymbol{A}) = \mathrm{trace}((\boldsymbol{A}\boldsymbol{S}\boldsymbol{A})^{1/2})$ with $\boldsymbol{A}$ defined on the set of positive definite matrices.

Proposition 5. *Function $f : S^n_{++} \to \Re$ is defined as $f(\boldsymbol{A}) = \mathrm{trace}((\boldsymbol{ASA})^{1/2})$ where $\boldsymbol{S} \in \mathcal{S}_n^{++}$. When the matrix $\boldsymbol{S}$ is positive definite, then the gradient of f at the point $\boldsymbol{A}$ is*

$$g(\boldsymbol{A}) = \frac{1}{2}[\boldsymbol{A}^{-1}(\boldsymbol{ASA})^{1/2} + (\boldsymbol{ASA})^{1/2}\boldsymbol{A}^{-1}]. \tag{6.8}$$

Proof. Let $F(\boldsymbol{A}) = (\boldsymbol{ASA})^{1/2}$, then $f(\boldsymbol{A}) = \mathrm{trace}(F(\boldsymbol{A}))$. For a given symmetric matrix $\boldsymbol{D}$,

$$F(\boldsymbol{A} + t\boldsymbol{D}) - F(\boldsymbol{A}) = (\boldsymbol{ASA} + \boldsymbol{E_D}(t, \boldsymbol{A}))^{1/2} - (\boldsymbol{ASA})^{1/2},$$

where $\boldsymbol{E_D}(t, \boldsymbol{A}) = t(\boldsymbol{DSA} + \boldsymbol{ASD}) + t^2\boldsymbol{DSD}$. Since both $\boldsymbol{A}$ and $\boldsymbol{S}$ are positive definite, the matrix $\boldsymbol{ASA}$ is positive definite. Let $L_{1/2}(\boldsymbol{ASA}, \boldsymbol{E_D}(t, \boldsymbol{A}))$ (or $L_{1/2}$ in short format) denote the Fréchet derivative for the matrix square root which is the unique solution to the Sylvester equation:

$$(\boldsymbol{ASA})^{1/2}L_{1/2} + L_{1/2}(\boldsymbol{ASA})^{1/2} = \boldsymbol{E_D}(t, \boldsymbol{A}). \tag{6.9}$$

By the definition of Fréchet derivative, we have

$$\|F(\boldsymbol{A} + t\boldsymbol{D}) - F(\boldsymbol{A}) - L_{1/2}(\boldsymbol{ASA}, \boldsymbol{E_D}(t, \boldsymbol{A}))\| = o(\|\boldsymbol{E_D}(t, \boldsymbol{A})\|) = o(t).$$

Then

$$\begin{aligned}
f(\boldsymbol{A} + t\boldsymbol{D}) - f(\boldsymbol{A}) &= \mathrm{trace}(F(\boldsymbol{A} + t\boldsymbol{D}) - F(\boldsymbol{A})) \\
&= \mathrm{trace}((\boldsymbol{ASA})^{-1/2}(\boldsymbol{ASA})^{1/2}[F(\boldsymbol{A} + t\boldsymbol{D}) - F(\boldsymbol{A})]) \\
&= \mathrm{trace}((\boldsymbol{ASA})^{-1/2}(\boldsymbol{ASA})^{1/2}L_{1/2}) + o(t) \\
&= \frac{1}{2}\mathrm{trace}((\boldsymbol{ASA})^{-1/2}\boldsymbol{E_D}(t, \boldsymbol{A})) + o(t) \\
&= \frac{1}{2}\mathrm{trace}((\boldsymbol{ASA})^{-1/2}[t(\boldsymbol{DSA} + \boldsymbol{ASD}) + t^2\boldsymbol{DSD}]) + o(t) \\
&= \frac{1}{2}t \cdot \mathrm{trace}(\boldsymbol{SA}(\boldsymbol{ASA})^{-1/2}\boldsymbol{D} + (\boldsymbol{ASA})^{-1/2}\boldsymbol{ASD}) + o(t).
\end{aligned}$$

Hence the directional derivative of f in the direction $\boldsymbol{D} \in S^n$ is

$$\begin{aligned}
\boldsymbol{\nabla_D} f(\boldsymbol{A}) &= \lim_{t \to 0} \frac{1}{t}(f(\boldsymbol{A} + t\boldsymbol{D}) - f(\boldsymbol{A})) \\
&= \langle \frac{1}{2}[\boldsymbol{SA}(\boldsymbol{ASA})^{-1/2} + (\boldsymbol{ASA})^{-1/2}\boldsymbol{AS}], \boldsymbol{D} \rangle.
\end{aligned}$$

Therefore, the gradient of f at point $\boldsymbol{A}$ is

$$\begin{aligned}
g(\boldsymbol{A}) &= \frac{1}{2}[\boldsymbol{SA}(\boldsymbol{ASA})^{-1/2} + (\boldsymbol{ASA})^{-1/2}\boldsymbol{AS}] \\
&= \frac{1}{2}[\boldsymbol{A}^{-1}(\boldsymbol{ASA})^{1/2} + (\boldsymbol{ASA})^{1/2}\boldsymbol{A}^{-1}].
\end{aligned}$$

□

We next extend the result of Proposition 5 to a more general case in which the matrix $\boldsymbol{S}$ is positive semidefinite but not necessarily positive definite.

Proposition 6. *Function $f : S^n_{++} \to \Re$ is defined as $f(\boldsymbol{A}) = \mathrm{trace}((\boldsymbol{ASA})^{1/2})$ where $\boldsymbol{S} \in \mathcal{S}^+_n$. Then the gradient of f at the point $\boldsymbol{A}$ is*

$$g(\boldsymbol{A}) = \frac{1}{2}[\boldsymbol{A}^{-1}(\boldsymbol{ASA})^{1/2} + (\boldsymbol{ASA})^{1/2}\boldsymbol{A}^{-1}]. \tag{6.10}$$

Proof. Let $f(\epsilon, \boldsymbol{A}) = \mathrm{trace}((\boldsymbol{A}(\boldsymbol{S}+\epsilon \boldsymbol{I})\boldsymbol{A})^{1/2})$, $\epsilon \in (0, 1]$, then $f(\boldsymbol{A}) = \lim_{\epsilon \downarrow 0} f(\epsilon, \boldsymbol{A})$. From Proposition 5 we know that the gradient of $f(\epsilon, \boldsymbol{A})$ is

$$g(\epsilon, \boldsymbol{A}) = \frac{1}{2}[\boldsymbol{A}^{-1}(\boldsymbol{A}(\boldsymbol{S}+\epsilon \boldsymbol{I})\boldsymbol{A})^{1/2} + (\boldsymbol{A}(\boldsymbol{S}+\epsilon \boldsymbol{I})\boldsymbol{A})^{1/2}\boldsymbol{A}^{-1}].$$

For a given symmetric matrix $\boldsymbol{D}$, there exists $\delta > 0$ such that $\boldsymbol{A} + t\boldsymbol{D} \succ 0$ when $t \in [-\delta, \delta]$. The directional derivative of f on the direction $\boldsymbol{D}$ is

$$\begin{aligned}
\lim_{t\to 0} \frac{1}{t}[f(\boldsymbol{A}+t\boldsymbol{D}) - f(\boldsymbol{A})] &= \lim_{t\to 0}\lim_{\epsilon\to 0} \frac{1}{t}[f(\epsilon, \boldsymbol{A}+t\boldsymbol{D}) - f(\epsilon, \boldsymbol{A})] \\
&= \lim_{\epsilon\to 0}\lim_{t\to 0} \frac{1}{t}[f(\epsilon, \boldsymbol{A}+t\boldsymbol{D}) - f(\epsilon, \boldsymbol{A})] \\
&= \lim_{\epsilon\to 0}\langle g(\epsilon, \boldsymbol{A}), \boldsymbol{D}\rangle \\
&= \langle g(\boldsymbol{A}), \boldsymbol{D}\rangle.
\end{aligned}$$

In the second equality, we change limits which we justify next. For given matrices $\boldsymbol{A}$ and $\boldsymbol{D}$, we define

$$G(\epsilon, t) = \begin{cases} \frac{1}{t}[f(\epsilon, \boldsymbol{A}+t\boldsymbol{D}) - f(\epsilon, \boldsymbol{A})] & \text{if } t \neq 0, \\ \langle g(\epsilon, \boldsymbol{A}), \boldsymbol{D}\rangle & \text{if } t = 0 \end{cases}$$

as a function of $\epsilon \in (0, 1]$ and $t \in [-\delta, \delta]$. To show that

$$\lim_{t\to 0}\lim_{\epsilon\to 0} G(\epsilon, t) = \lim_{\epsilon\to 0}\lim_{t\to 0} G(\epsilon, t),$$

a sufficient condition is (see Theorem 7.11 in [53]):

(a) For every $\epsilon \in (0, 1]$ the finite limit $\lim_{t\to 0} G(\epsilon, t)$ exists.

(b) For every $t \in [-\delta, \delta]$, the finite limit $\lim_{\epsilon\to 0} G(\epsilon, t)$ exists.

(c) As $t \to 0$, $G(\epsilon, t)$ uniformly converges to a limit function for $\epsilon \in (0, 1]$.

It is obvious that conditions (a) and (b) are true. A sufficient and necessary condition for (c) is (see Theorem 7.9 in [53]):

$$\lim_{t\to 0} \sup_{\epsilon\in(0,1]} |G(\epsilon, t) - G(\epsilon, 0)| = 0. \tag{6.11}$$

Next, we prove the result of (6.11). By the mean value theorem and the proof of Proposition 5, there exists a t_1 between 0 and t, such that $G(\epsilon, t) = \langle (g(\epsilon, \boldsymbol{A} + t_1\boldsymbol{D}), \boldsymbol{D}\rangle$. Then

$$G(\epsilon, t) - G(\epsilon, 0) = \langle g(\epsilon, \boldsymbol{A} + t_1\boldsymbol{D}), \boldsymbol{D})\rangle - \langle g(\epsilon, \boldsymbol{A}), \boldsymbol{D})\rangle. \tag{6.12}$$

For $t \in (-\delta, \delta)$, define $h(\epsilon, t) = \langle g(\epsilon, \boldsymbol{A} + t\boldsymbol{D}), \boldsymbol{D}) \rangle$. Let $\boldsymbol{A}_t = (\boldsymbol{A} + t\boldsymbol{D})$, $\boldsymbol{S}_\epsilon = \boldsymbol{S} + \epsilon \boldsymbol{I}$, then

$$\begin{aligned}\frac{\partial h(\epsilon,t)}{\partial t} &= \lim_{\Delta t \to 0} \frac{1}{\Delta t} \mathrm{trace}\left(\boldsymbol{D}(\boldsymbol{A}_t + \Delta t\boldsymbol{D})^{-1}[(\boldsymbol{A}_t + \Delta t\boldsymbol{D})\boldsymbol{S}_\epsilon(\boldsymbol{A}_t + \Delta t\boldsymbol{D})]^{1/2} - \boldsymbol{D}\boldsymbol{A}_t^{-1}[\boldsymbol{A}_t\boldsymbol{S}_\epsilon\boldsymbol{A}_t]^{1/2}\right) \\ &= \lim_{\Delta t \to 0} \frac{1}{\Delta t} \mathrm{trace}\Big(\boldsymbol{D}\boldsymbol{A}_t^{-1}(\boldsymbol{I} - \Delta t\boldsymbol{D}\boldsymbol{A}_t^{-1} + o(\Delta t))\big[(\boldsymbol{A}_t\boldsymbol{S}_\epsilon\boldsymbol{A}_t)^{1/2} + L_{1/2}(\boldsymbol{A}_t\boldsymbol{S}_\epsilon\boldsymbol{A}_t, \Delta t(\boldsymbol{A}_t\boldsymbol{S}_\epsilon\boldsymbol{D} + \boldsymbol{D}\boldsymbol{S}_\epsilon\boldsymbol{A}_t)) + o(t)\big] \\ &\quad - \boldsymbol{D}\boldsymbol{A}_t^{-1}(\boldsymbol{A}_t\boldsymbol{S}_\epsilon\boldsymbol{A}_t)^{1/2}\Big),\end{aligned}$$

where $L_{1/2}(\boldsymbol{A}_t\boldsymbol{S}_\epsilon\boldsymbol{A}_t, t(\boldsymbol{A}_t\boldsymbol{S}_\epsilon\boldsymbol{D} + \boldsymbol{D}\boldsymbol{S}_\epsilon\boldsymbol{A}_t))$ is the Fréchet derivative of the matrix square root. For simplicity we denote it by $L_{1/2}$, and it satisfies

$$(\boldsymbol{A}_t\boldsymbol{S}_\epsilon\boldsymbol{A}_t)^{1/2}L_{1/2} + L_{1/2}(\boldsymbol{A}_t\boldsymbol{S}_\epsilon\boldsymbol{A}_t)^{1/2} = \Delta t(\boldsymbol{A}_t\boldsymbol{S}_\epsilon\boldsymbol{D} + \boldsymbol{D}\boldsymbol{S}_\epsilon\boldsymbol{A}_t).$$

Let $L = L_{1/2}/\Delta t$, then

$$(\boldsymbol{A}_t\boldsymbol{S}_\epsilon\boldsymbol{A}_t)^{1/2}L + L(\boldsymbol{A}_t\boldsymbol{S}_\epsilon\boldsymbol{A}_t)^{1/2} = (\boldsymbol{A}_t\boldsymbol{S}_\epsilon\boldsymbol{D} + \boldsymbol{D}\boldsymbol{S}_\epsilon\boldsymbol{A}_t) \tag{6.13}$$

and

$$\begin{aligned}\frac{\partial h(\epsilon,t)}{\partial t} &= \lim_{\Delta t \to 0} \frac{1}{\Delta t} \mathrm{trace}\left(\boldsymbol{D}\boldsymbol{A}_t^{-1}(\Delta t L - \Delta t\boldsymbol{D}\boldsymbol{A}^{-1}(\boldsymbol{A}_t\boldsymbol{S}_\epsilon\boldsymbol{A}_t)^{1/2}\right) + o(\Delta t) \\ &= \mathrm{trace}\left(\boldsymbol{D}\boldsymbol{A}_t^{-1}L\right) - \mathrm{trace}\left((\boldsymbol{D}\boldsymbol{A}_t^{-1})^2(\boldsymbol{A}_t\boldsymbol{S}_\epsilon\boldsymbol{A}_t)^{1/2}\right). \end{aligned} \tag{6.14}$$

L is the solution of the Sylvester equation (6.13) which is unique, hence the partial derivative of $h(\epsilon, t)$ with respect t exists. Next, we show that $\frac{\partial h(\epsilon,t)}{\partial t}$ is bounded for $(\epsilon, t) \in (0, 1] \times (-\delta, \delta)$. We can find that the second item of (6.14) is well defined and continuous on a compact set $[0, 1] \times [-\delta, \delta]$, hence it is bounded on $(0, 1] \times (-\delta, \delta)$. For the first item of (6.14), we know that

$$|\mathrm{trace}\left(\boldsymbol{D}\boldsymbol{A}_t^{-1}L\right)| \leq \frac{1}{2}\|\boldsymbol{D}\boldsymbol{A}_t^{-1}\|_F^2 + \frac{1}{2}\|L\|_F^2,$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. $\|\boldsymbol{D}\boldsymbol{A}_t^{-1}\|_F^2$ is continuous in t on the set $[-\delta, \delta]$, hence it is bounded. We only need to show that $\|L\|_F$ is bounded. Actually, we can obtain the closed form of L by solving the Sylvester equation (6.13).

Let $\boldsymbol{P}^T\boldsymbol{\Lambda}\boldsymbol{P}$ be the eigenvalue decomposition of $\boldsymbol{A}_t\boldsymbol{S}_\epsilon\boldsymbol{A}_t$. Then (6.13) can be written as

$$\begin{aligned}\boldsymbol{P}^T\boldsymbol{\Lambda}^{1/2}\boldsymbol{P}L + L\boldsymbol{P}^T\boldsymbol{\Lambda}^{1/2}\boldsymbol{P} &= (\boldsymbol{A}_t\boldsymbol{S}_\epsilon\boldsymbol{A}_t)\boldsymbol{A}_t^{-1}\boldsymbol{D} + \boldsymbol{D}\boldsymbol{A}_t^{-1}(\boldsymbol{A}_t\boldsymbol{S}_\epsilon\boldsymbol{A}_t) \\ &= \boldsymbol{P}^T\boldsymbol{\Lambda}\boldsymbol{P}\boldsymbol{A}_t^{-1}\boldsymbol{D} + \boldsymbol{D}\boldsymbol{A}_t^{-1}\boldsymbol{P}^T\boldsymbol{\Lambda}\boldsymbol{P}\end{aligned}$$

Let $\bar{L} = \boldsymbol{P}L\boldsymbol{P}^T$, we have

$$\begin{aligned}\boldsymbol{\Lambda}^{1/2}\bar{L} + \bar{L}\boldsymbol{\Lambda}^{1/2} &= \boldsymbol{\Lambda}\boldsymbol{P}\boldsymbol{A}_t^{-1}\boldsymbol{D}\boldsymbol{P}^T + \boldsymbol{P}\boldsymbol{D}\boldsymbol{A}_t^{-1}\boldsymbol{P}^T\boldsymbol{\Lambda} \\ &= \boldsymbol{\Lambda}\boldsymbol{E} + \boldsymbol{E}^T\boldsymbol{\Lambda},\end{aligned} \tag{6.15}$$

where $\boldsymbol{E} = \boldsymbol{P}\boldsymbol{A}_t^{-1}\boldsymbol{D}\boldsymbol{P}^T$. The solution for equation (6.15) is

$$\bar{L}_{ij} = \frac{\lambda_i E_{ij} + \lambda_j E_{ji}}{\lambda_i^{1/2} + \lambda_j^{1/2}}.$$

Therefore $|\bar{L}_{ij}| \leq (\lambda_i^{1/2}+\lambda_j^{1/2})\max_{i,j}[|E_{ij}|]$. Notice that $\|\boldsymbol{E}\| \leq \|\boldsymbol{P}\|\cdot\|\boldsymbol{A}_t^{-1}\boldsymbol{D}\|\cdot\|\boldsymbol{P}^T\|$ is bounded on $(0,1]\times[-\delta,\delta]$. The eigenvalues of $\boldsymbol{A}_t(\boldsymbol{S}+\epsilon\boldsymbol{I})\boldsymbol{A}_t$ are no larger than the eigenvalues of $\boldsymbol{A}_t(\boldsymbol{S}+\boldsymbol{I})\boldsymbol{A}_t$ when $\epsilon \in (0,1]$. Since $\|\boldsymbol{A}_t(\boldsymbol{S}+\boldsymbol{I})\boldsymbol{A}_t\|_F$ is continuous on $[-\delta,\delta]$, it is bounded. Hence the eigenvalues of $\boldsymbol{A}_t(\boldsymbol{S}+\epsilon\boldsymbol{I})\boldsymbol{A}_t$ are bounded on $(0,\epsilon]\times[-\delta,\delta]$. Therefore $\|\bar{L}\|_F$ is bound which implies that $\|L\|_F$ is also bounded.

By the above discussion, we know that for all $\epsilon \in (0,1]$, and $t \in (-\delta,\delta)$, there exists a constant M such that $|\frac{\partial h(\epsilon,t)}{\partial t}| \leq M$. Therefore, for all $\epsilon \in (0,1]$, and $t \in (-\delta,\delta)$, $|h(\epsilon,t)-h(\epsilon,0)| \leq M|t|$. Then by (6.12) and the definition of $h(\epsilon,t)$,

$$\begin{aligned}\lim_{t\to 0}\sup_{\epsilon\in(0,1]}|G(\epsilon,t)-G(\epsilon,0)| &= \lim_{t\to 0}\sup_{\epsilon\in(0,1]}|h(\epsilon,t_1)-h(\epsilon,0)| \\ &\leq \lim_{t\to 0} M|t| \\ &= 0.\end{aligned}$$

□

We now provide the proof of the gradient of the function which helps complete the proof.

Proposition 7. *Function $V : \Re_n^{++} \to \Re$ is defined as*

$$V(\boldsymbol{\sigma}) = \operatorname{trace}\Big(\big([\boldsymbol{A}\operatorname{Diag}(\boldsymbol{\sigma})\boldsymbol{C}\operatorname{Diag}(\boldsymbol{\sigma})\boldsymbol{A}^T]^{1/2}\boldsymbol{S}[\boldsymbol{A}\operatorname{Diag}(\boldsymbol{\sigma})\boldsymbol{C}\operatorname{Diag}(\boldsymbol{\sigma})\boldsymbol{A}^T]^{1/2}\big)^{1/2}\Big), \tag{6.16}$$

where $\boldsymbol{A} \in \Re^{m\times n}$ is a matrix with full row rank, $\boldsymbol{C} \in \mathcal{S}_n^{++}$ is a given positive definite matrix, and $\boldsymbol{S} \in \mathcal{S}_m^{+}$ is a given positive semidefinite matrix. Then the gradient of V is

$$grad(\boldsymbol{\sigma}) = \operatorname{diag}\Big(\boldsymbol{A}^T h(\boldsymbol{\sigma})^{-1}\big(h(\boldsymbol{\sigma})\boldsymbol{S}h(\boldsymbol{\sigma})\big)^{1/2}h(\boldsymbol{\sigma})^{-1}\boldsymbol{A}\operatorname{Diag}(\boldsymbol{\sigma})\boldsymbol{C}\Big), \tag{6.17}$$

where $h(\boldsymbol{\sigma}) = (\boldsymbol{A}\operatorname{Diag}(\boldsymbol{\sigma})\boldsymbol{C}\operatorname{Diag}(\boldsymbol{\sigma})\boldsymbol{A}^T)^{1/2}$.

Proof. Let $L(\boldsymbol{\sigma},\cdot)$ be the Fréchet derivative of h. Then for all unit vector $\boldsymbol{v} \in \Re_n$ and $t \in \Re$,

$$\|h(\boldsymbol{\sigma}+t\boldsymbol{v})-h(\boldsymbol{\sigma})-L(\boldsymbol{\sigma},t\boldsymbol{v})\| = o(t). \tag{6.18}$$

By simple calculation, we have

$$h(\boldsymbol{\sigma}+t\boldsymbol{v})-h(\boldsymbol{\sigma}) = (h(\boldsymbol{\sigma})+\boldsymbol{E}_{\boldsymbol{v}}(t,\boldsymbol{\sigma}))^{1/2}-h(\boldsymbol{\sigma}), \tag{6.19}$$

where $\boldsymbol{E}_{\boldsymbol{v}}(t,\boldsymbol{\sigma}) = t\boldsymbol{A}[\operatorname{Diag}(\boldsymbol{\sigma})\boldsymbol{C}\operatorname{Diag}(\boldsymbol{v})+\operatorname{Diag}(\boldsymbol{v})\boldsymbol{C}\operatorname{Diag}(\boldsymbol{\sigma})]\boldsymbol{A}^T+t^2\boldsymbol{A}\operatorname{Diag}(\boldsymbol{v})\boldsymbol{C}\operatorname{Diag}(\boldsymbol{v})\boldsymbol{A}^T$. Let $L_{1/2}$ denote the Fréchet derivative of the matrix square root, then

$$\|h(\boldsymbol{\sigma}+t\boldsymbol{v})-h(\boldsymbol{\sigma})-L_{1/2}(h(\boldsymbol{\sigma})^2,\boldsymbol{E}_{\boldsymbol{v}}(t,\boldsymbol{\sigma}))\| = o(\|\boldsymbol{E}_{\boldsymbol{v}}(t,\boldsymbol{\sigma})\|) = o(t). \tag{6.20}$$

By (6.18) and (6.20), we have

$$\|L(\boldsymbol{\sigma},t\boldsymbol{v})-L_{1/2}(h(\boldsymbol{\sigma})^2,\boldsymbol{E}_{\boldsymbol{v}}(t,\boldsymbol{\sigma}))\| = o(t). \tag{6.21}$$

From the Sylvester equation for the Fréchet derivative of matrix square root, we obtain

$$h(\boldsymbol{\sigma})L(\boldsymbol{\sigma},t\boldsymbol{v})+L(\boldsymbol{\sigma},t\boldsymbol{v})h(\boldsymbol{\sigma}) = t\boldsymbol{A}[\operatorname{Diag}(\boldsymbol{\sigma})\boldsymbol{C}\operatorname{Diag}(\boldsymbol{v})+\operatorname{Diag}(\boldsymbol{v})\boldsymbol{C}\operatorname{Diag}(\boldsymbol{\sigma})]\boldsymbol{A}^T+o(t). \tag{6.22}$$

By the above equation we have $\|L(\boldsymbol{\sigma}, t\boldsymbol{v})\| = O(t)$.

Let f be the trace function $f(\boldsymbol{A}) = \mathrm{trace}((\boldsymbol{ASA})^{1/2})$, and g be its gradient as in Proposition 6. Then $V(\boldsymbol{\sigma}) = f(h(\boldsymbol{\sigma}))$. By the mean value theorem, there exist an $\alpha \in [0,1]$ such that

$$\begin{aligned} V(\boldsymbol{\sigma}+t\boldsymbol{v}) - V(\boldsymbol{\sigma}) &= f(h(\boldsymbol{\sigma}+t\boldsymbol{v}) - h(\boldsymbol{\sigma})) \\ &= \langle g(\alpha h(\boldsymbol{\sigma}+t\boldsymbol{v}) + (1-\alpha)h(\boldsymbol{\sigma})), h(\boldsymbol{\sigma}+t\boldsymbol{v}) - h(\boldsymbol{\sigma})\rangle \\ &= \langle g(\alpha h(\boldsymbol{\sigma}+t\boldsymbol{v}) + (1-\alpha)h(\boldsymbol{\sigma})), L(\boldsymbol{\sigma}, t\boldsymbol{v})\rangle + o(t) \end{aligned}$$

Then the directional derivative of V at the direction $\boldsymbol{v}$ is

$$\begin{aligned} \lim_{t\to 0}\frac{1}{t}(V(\boldsymbol{\sigma}+t\boldsymbol{v}) - V(\boldsymbol{\sigma})) &= \lim_{t\to 0}\langle g(h(\boldsymbol{\sigma})), L(\boldsymbol{\sigma}, t\boldsymbol{v})\rangle \\ &= \lim_{t\to 0}\mathrm{trace}\left(\frac{1}{2}[h(\boldsymbol{\sigma})^{-1}(h(\boldsymbol{\sigma})\boldsymbol{S}h(\boldsymbol{\sigma}))^{1/2} + (h(\boldsymbol{\sigma})\boldsymbol{S}h(\boldsymbol{\sigma}))^{1/2}h(\boldsymbol{\sigma})^{-1}]L(\boldsymbol{\sigma}, t\boldsymbol{v})\right) \\ &= \lim_{t\to 0}\mathrm{trace}\left(h(\boldsymbol{\sigma})^{-1}(h(\boldsymbol{\sigma})\boldsymbol{S}h(\boldsymbol{\sigma}))^{1/2}L(\boldsymbol{\sigma}, t\boldsymbol{v})\right) \\ &= \lim_{t\to 0}\mathrm{trace}\left(\boldsymbol{B}(\boldsymbol{\sigma})h(\boldsymbol{\sigma})L(\boldsymbol{\sigma}, t\boldsymbol{v})\right), \end{aligned}$$

where $\boldsymbol{B}(\boldsymbol{\sigma}) = h(\boldsymbol{\sigma})^{-1}(h(\boldsymbol{\sigma})\boldsymbol{S}h(\boldsymbol{\sigma}))^{1/2}h(\boldsymbol{\sigma})^{-1}$ is a symmetric matrix. By (6.22), we know that

$$\begin{aligned} \mathrm{trace}\left(\boldsymbol{B}(\boldsymbol{\sigma})h(\boldsymbol{\sigma})L(\boldsymbol{\sigma}, t\boldsymbol{v})\right) &= \frac{t}{2}\mathrm{trace}(\boldsymbol{B}(\boldsymbol{\sigma})\boldsymbol{A}[\mathrm{Diag}(\boldsymbol{\sigma})\boldsymbol{C}\mathrm{Diag}(\boldsymbol{v}) + \mathrm{Diag}(\boldsymbol{v})\boldsymbol{C}\mathrm{Diag}(\boldsymbol{\sigma})]\boldsymbol{A}^T) + o(t) \\ &= t\cdot\mathrm{trace}(\boldsymbol{B}(\boldsymbol{\sigma})\boldsymbol{A}\mathrm{Diag}(\boldsymbol{\sigma})\boldsymbol{C}\mathrm{Diag}(\boldsymbol{v})\boldsymbol{A}^T) + o(t). \end{aligned}$$

Then

$$\begin{aligned} \lim_{t\to 0}\frac{1}{t}(V(\boldsymbol{\sigma}+t\boldsymbol{v}) - V(\boldsymbol{\sigma})) &= \mathrm{trace}(\boldsymbol{B}(\boldsymbol{\sigma})\boldsymbol{A}\mathrm{Diag}(\boldsymbol{\sigma})\boldsymbol{C}\mathrm{Diag}(\boldsymbol{v})\boldsymbol{A}^T) \\ &= \langle \mathrm{diag}(\boldsymbol{A}^T\boldsymbol{B}(\boldsymbol{\sigma})\boldsymbol{A}\mathrm{Diag}(\boldsymbol{\sigma})\boldsymbol{C}), \boldsymbol{v}\rangle. \end{aligned}$$

Hence, the gradient of V at $\boldsymbol{\sigma}$ is $\mathrm{diag}\big(\boldsymbol{A}^T\boldsymbol{B}(\boldsymbol{\sigma})\boldsymbol{A}\mathrm{Diag}(\boldsymbol{\sigma})\boldsymbol{C}\big)$ and the result is proved. $\square$

Using the result of Proposition 7, the closed form (4.2) for the gradient of f_{cmm} with respect to $\boldsymbol{\sigma}$ can be easily obtained.

References

[1] S.D. Ahipasaoglu, X. Li, and K. Natarajan, *A convex optimization approach for computing correlated choice probabilities with many alternatives*, IEEE Trans. Automatic Control **64** (2019), 190–205.

[2] A. Banerjee and A. Paul, *On path correlation and PERT bias*, Eur. J. Oper. Res. **189** (2008), 1208–1216.

[3] D. Bertsimas, X.V. Doan, K. Natarajan, and C.P. Teo, *Models for minimax stochastic linear optimization problems with risk aversion*, Math. Oper. Res. **35** (2010), 580–602.

[4] D. Bertsimas, K. Natarajan, and C.P. Teo, *Probabilistic combinatorial optimization: Moments, semidefinite programming, and asymptotic bounds*, SIAM J. Optim. **15** (2004), 185–209.

[5] D. Bertsimas, K. Natarajan, and C.P. Teo, *Persistence in discrete optimization under data uncertainty*, Math. Program. **108** (2006), 251–274.

[6] D. Bertsimas and I. Popescu, *Optimal inequalities in probability theory: A convex optimization approach*, SIAM J. Optim. **15** (2005), 780–804.

[7] J.R. Birge and M.J. Maddox, *Bounds on expected project tardiness*, Oper. Res. **43** (1995), 838–850.

[8] R.A. Bowman, *Stochastic gradient-based time-cost tradeoffs in pert networks using simulation*, Ann. Oper. Res. **53** (1994), 533–551.

[9] J.M. Burt Jr and M.B. Garman, *Conditional Monte Carlo: A simulation technique for stochastic network analysis*, Manage. Sci. **18** (1971), 207–217.

[10] E. Carlen, *Trace inequalities and quantum entropy: An introductory course*, Entropy Quantum **529** (2010), 73–140.

[11] X. Chen, M. Sim, and P. Sun, *A robust optimization perspective on stochastic programming*, Oper. Res. **55** (2007), 1058–1071.

[12] X. Chen, M. Sim, P. Sun, and J. Zhang, *A linear decision-based approximation approach to stochastic programming*, Oper. Res. **56** (2008), 344–357.

[13] C.E. Clark, *The greatest of a finite set of random variables*, Oper. Res. **9** (1961), 145–162.

[14] I. Cohen, B. Golany, and A. Shtub, *The stochastic time–cost tradeoff problem: A robust optimization approach*, Networks **49** (2007), 175–188.

[15] L. Condat, *Fast projection onto the simplex and the l_1 ball*, Math. Program., Ser. A **158** (2016), 575–585.

[16] P. De, E.J. Dunne, J.B. Ghosh, and C.E. Wells, *Complexity of the discrete time-cost tradeoff problems for project networks*, Oper. Res. **45** (2007), 302–306.

[17] A. De Meyer, C.H. Loch, and M.T. Pich, *Managing project uncertainty*, MIT Sloan Manage. Review **43** (2002), 60–67.

[18] X.V. Doan and K. Natarajan, *On the complexity of non-overlapping multivariate marginal bounds for probabilistic combinatorial optimization*, Oper. Res. **60** (2012), 138–149.

[19] B. Eaves, *On the basic theorem of complementarity*, Math. Program. **1** (1971), 68–75.

[20] S.E. Elmaghraby, *Activity Networks: Project Planning and Control by Network Models*, John Wiley & Sons: New York, 1978.

[21] J.E. Falk and J.L. Horowitz, *Critical path problems with concave cost-time curves*, Manage. Sci. **19** (1972), 446–455.

[22] A.V. Fiacco and Y. Ishizuka, *Sensitivity and stability analysis for nonlinear programming*, Ann. Oper. Res. **27** (1990), 215–235.

[23] M. Fu, *Handbook of Simulation Optimization*, Springer, 2015.

[24] D. Fulkerson, *A network flow computation for project cost curves*, Manage. Sci. **7** (1961), 167–178.

[25] D. Fulkerson, *Expected critical path lengths in pert networks*, Oper. Res. **10** (1962), 808–817.

[26] J. Goh and N.G. Hall, *Total cost control in project management via satisficing*, Manage. Sci. **59** (2013), 1354–1372.

[27] M. Grant and S. Boyd, "*Graph implementations for nonsmooth convex programs*," *Recent Advances in Learning and Control*, V. Blondel, S. Boyd, and H. Kimura (eds.), Springer-Verlag Limited, 2008, pp. 95–110.

[28] M. Grant and S. Boyd, 2014. CVX: Matlab software for disciplined convex programming, version 2.1.

[29] G.J. Gutierrez and A. Paul, *Robustness to variability in project networks*, IIE Trans. **33** (2001), 649–660.

[30] J.N. Hagstrom, *Computational complexity of PERT problems*, Networks **18** (1988), 139–147.

[31] G.A. Hanasusanto, V. Roitch, D. Kuhn, and W. Wiesemann, *Ambiguous joint chance constraints under mean and dispersion information*, Oper. Res. **65** (2017), 751–767.

[32] P.T. Harker and J.S. Pang, *Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications*, Math. Program. **48** (1990), 161–220.

[33] B. He, *A class of projection and contraction methods for monotone variational inequalities*, Appl. Math. Optim. **35** (1997), 69–76.

[34] D.A. Iancu and N. Trichakis, *Pareto efficiency in robust optimization*, Manage. Sci. **60** (2014), 130–147.

[35] J. Kelley Jr, *Critical-path planning and scheduling: Mathematical basis*, Oper. Res. **9** (1961), 296–320.

[36] S. Kim, R. Pasupathy, and S.G. Henderson, "*A guide to sample average approximation*," *Handbook of Simulation Optimization*, Springer, 2015, pp. 207–243.

[37] S.J. Kim, S.P. Boyd, S. Yun, D. Patil, and M.A. Horowitz, *A heuristic for optimizing stochastic activity networks with applications to statistical digital circuit sizing*, Optim. Eng. **8** (2007), 397–430.

[38] Q. Kong, C.Y. Lee, C.P. Teo, and Z. Zheng, *Scheduling arrivals to a stochastic delivery system using copositive cones*, Oper. Res. **61** (2013), 711–726.

[39] L.R. Lamberson and R.R. Hocking, *Optimum time compression in project scheduling*, Manage. Sci. **16** (1970), B597–B606.

[40] X. Li, K. Natarajan, C.P. Teo, and Z. Zheng, *Distributionally robust mixed integer linear programs: Persistency models with applications*, Eur. J. Oper. Res. **233** (2014), 459–473.

[41] J.H. Lindsey, *An estimate of expected critical-path length in pert networks*, Oper. Res. **20** (1972), 800–812.

[42] H.Y. Mak, Y. Rong, and J. Zhang, *Appointment scheduling with limited distributional information*, Manage. Sci. **61** (2015), 316–334.

[43] I. Meilijson and A. Nadas, *Convex majorization with an application to the length of critical paths*, J. Appl. Prob. **16** (1979), 671–677.

[44] G. Mitchell and T. Klastorin, *An effective methodology for the stochastic project compression problem*, IIE Trans. **39** (2007), 957–969.

[45] R.H. Möhring, *Scheduling under uncertainty: Bounding the makespan distribution*, Comput. Discr. Math. (2001), 79–97.

[46] K. Natarajan, M. Song, and C.P. Teo, *Persistency model and its applications in choice modeling*, Manage. Sci. **55** (2009), 453–469.

[47] K. Natarajan and C.P. Teo, *On reduced semidefinite programs for second order moment bounds with applications*, Math. Program. **161** (2017), 487–518.

[48] K. Natarajan, C.P. Teo, and Z. Zheng, *Mixed 0-1 linear programs under objective uncertainty: A completely positive representation*, Oper. Res. **59** (2011), 713–728.

[49] I. Olkin and F. Pukelsheim, *The distance between two random vectors with given dispersion matrices*, Linear Algebra Its Appl. **48** (1982), 257–263.

[50] E.L. Plambeck, B.R. Fu, S.M. Robinson, and R. Suri, *Sample-path optimization of convex stochastic performance functions*, Math. Program. **75** (1996), 137–176.

[51] M. Pourahmadi and X. Wang, *Distribution of random correlation matrices: Hyperspherical parameterization of the Cholesky factor*, Stat. Prob. Lett. **106** (2015), 5–12.

[52] R.T. Rockafellar, *Convex Analysis*, Number 28, Princeton University Press, 1997.

[53] W. Rudin, *Principles of Mathematical Analysis* Vol. 3, McGraw-Hill New York, 1964.

[54] A. Shapiro, *Monte Carlo sampling methods*, Handbooks operations research management science **10** (2003), 353–425.

[55] K.C. Toh, M.J. Todd, and R.H. Tütüncü, *Sdpt3—a MATLAB software package for semidefinite programming, version 1.3*, Optim. Methods Software **11** (1999), 545–581.

[56] R.H. Tütüncü, K.C. Toh, and M.J. Todd, *Solving semidefinite-quadratic-linear programs using sdpt3*, Math. Program. **95** (2003), 189–217.

[57] R.M. Van Slyke, *Letter to the editor - Monte Carlo methods and the PERT problem*, Oper. Res. **11** (1963), 839–860.

[58] B.W. Wambeke, S.M. Hsiang, and M. Liu, *Causes of variation in construction project task starting times and duration*, J. Construction Eng. Manage. **137** (2011), 663–677.

[59] W. Wiesemann, D. Kuhn, and B. Rustem, *Robust resource allocations in temporal networks*, Math. Program. **135** (2012), 437–471.

[60] N. Xiu, C. Wang, and J. Zhang, *Convergence properties of projection and contraction methods for variational inequality problems*, Appl. Math. Optim. **43** (2001), 147–168.