Distributionally Robust Project Crashing with Partial or No Correlation Information

Selin Damla Ahipasaoglu* Karthik Natarajan† Dongjian Shi‡

November 8, 2016

Abstract

Crashing is a method for optimally shortening the project makespan by reducing the time of one or more activities in a project network by allocating resources to it. Activity durations are however uncertain and techniques in stochastic optimization, robust optimization and distributionally robust optimization have been developed to tackle this problem. In this paper, we study a class of distributionally robust project crashing problems where the objective is to choose the first two moments of the activity durations to minimize the worst-case expected makespan. Under a partial correlation information structure or no correlation information, the problem is shown to be solvable in polynomial time as a semidefinite program or a second order cone program respectively. However in practice, solving the semidefinite program is challenging for large project networks. We exploit the structure of the problem to reformulate it as a convex-concave saddle point problem over the first two moment variables and the arc criticality index variables. This provides the opportunity to use first order saddle point methods to solve larger sized distributionally robust project crashing problems. Numerical results also provide an useful insight that as compared to the crashing solution for the multivariate normal distribution, the distributionally robust project crashing solution tends to deploy more resources in reducing the standard deviation rather than the mean of the activity durations.

1 Introduction

A project is defined by a set of activities with given precedence constraints. In a project, an 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

*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

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. The corresponding 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 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 (longest) path of the project network from node 1 to node n where arc lengths denote activity durations. The 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

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

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 can assign more resources to it which typically implies additional costs. This may include using more efficient equipment or hiring more workers. Hence it is important to find 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 1960s (see Kelley Jr (1961), Fulkerson (1961)) 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, when the activities on the critical path have been crashed, the original critical path may no longer be critical. The problem of minimizing the project makespan with a given cost budget is formulated as follows:

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

where M 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 function 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, using duality, the project crashing problem (1.3) is formulated as:

$$\begin{aligned}\min_{\boldsymbol{t},\boldsymbol{y}}\quad & y_n-y_1,\\ \text{s.t.}\quad & y_j-y_i\ge t_{ij}, \qquad \forall(i,j)\in\mathcal{A},\\ & \sum_{(i,j)\in\mathcal{A}}c_{ij}(t_{ij})\le M,\\ & \underline{t}_{ij}\le t_{ij}\le\bar{t}_{ij}, \qquad \forall(i,j)\in\mathcal{A}.\end{aligned} \tag{1.3}$$

When the cost functions are linear or piecewise linear convex, the deterministic PCP (1.3) is formulated as a linear program. For nonlinear convex differentiable cost functions, an algorithm based on piecewise linear approximations was proposed by Lamberson and Hocking (1970) to solve the project crashing problem. When the cost function is nonlinear and concave, Falk and Horowitz (1972) 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 De et al. (2007)). 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. $\mathbf{1}_n$ is a vector of dimension n with all entries equal to 1. $\boldsymbol{\Delta}_{n-1} = \{\boldsymbol{x} : \mathbf{1}_n^T \boldsymbol{x} = 1, \boldsymbol{x} \geq 0\}$ denotes a unit simplex. We suppress the subscript when the dimension is clear. We use the tilde notation to denote a random variable or random vector, such as $\tilde{r}$, $\tilde{\boldsymbol{r}}$. $|\mathcal{A}|$ denote the number of elements in the set $\mathcal{A}$. $\mathbb{Z}_+$ denotes the set of nonnegative integers. $\Re_n^+$ and $\Re_n^{++}$ are the sets of n dimensional vectors whose entries are all nonnegative and strictly positive respectively. $\mathcal{S}_n^+$ and $\mathcal{S}_n^{++}$ are the sets of all symmetric $n \times n$ positive semidefinite matrices and positive definite matrices, 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. $\mathrm{diag}(\boldsymbol{X})$ denotes a vector formed by the diagonal elements of a matrix $\boldsymbol{X}$, and $\mathrm{Diag}(\boldsymbol{x})$ denotes a diagonal matrix whose diagonal elements are the entries of $\boldsymbol{x}$.

2 Literature Review

In this section, we provide a literature review of the project crashing problem with uncertain activity durations. Since this problem has been well-studied, our literature review while extensive is not exhaustive. We highlight some of the current state of art methods to solve the project crashing problem before discussing our proposed method in later sections.

2.1 Stochastic Project Crashing

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

$$\mathbb{E}(Z(\tilde{\boldsymbol{t}})) = \int Z(\boldsymbol{t}) f_{\boldsymbol{\lambda}}(\boldsymbol{t}) d\boldsymbol{t}, \tag{2.1}$$

where $f_{\boldsymbol{\lambda}}(\boldsymbol{t})$ is the probability density function of the random vector $\tilde{\boldsymbol{t}}$ with the parameter vector $\boldsymbol{\lambda}$. The stochastic PCP that minimizes the expected makespan with a given cost budget is formulated as:

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

where $\Omega_{\boldsymbol{\lambda}}$ 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. Hagstrom (1988) showed that computing the expected project makespan is NP-hard when the activity durations are independent discrete random variables. Even in simple cases 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 Van Slyke (1963), Burt Jr and Garman (1971)). Simple bounds (Fulkerson (1962), Möhring (2001)) and approximations (Lindsey (1972)) have also 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, heuristics and simulation-based optimization methods have been proposed. Kim et al. (2007) developed a heuristic approach to minimize the quantile of the makespan by using a surrogate deterministic objective function with activity durations defined as:

$$d_{ij}(\boldsymbol{\lambda}) = \mu_{ij}(\boldsymbol{\lambda}) + k_{ij}\sigma_{ij}(\boldsymbol{\lambda}),$$

with fixed margin coefficients $k_{ij} \geq 0$, where $\mu_{ij}(\boldsymbol{\lambda})$ and $\sigma_{ij}(\boldsymbol{\lambda})$ denote the mean and standard deviation of the activity duration $\tilde{t}_{ij}$ that depends on the decision vector $\boldsymbol{\lambda}$. They then solve the deterministic PCP:

$$\min_{\boldsymbol{\lambda}\in\Omega_{\boldsymbol{\lambda}}} Z(\boldsymbol{d}(\boldsymbol{\lambda})), \tag{2.3}$$

to find a heuristic solution for project crashing. Other heuristics for the project crashing problem have also been developed (see Mitchell and Klastorin (2007)). Simulation based optimization techniques have been used to solve the project crashing problem. Stochastic gradient methods for minimizing the expected makespan have been developed in this context (see Bowman (1994), Fu (2015)). Another approach is to use the sample average approximation (SAA) method to minimize the expected makespan (see Plambeck et al. (1996), Shapiro (2003), Kim et al. (2015)). For example, 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 $\tilde{\boldsymbol{t}}$. Suppose the decision variables are the means and standard deviations of the random activity durations where $\Omega_{\boldsymbol{\mu},\boldsymbol{\sigma}}$ represents a convex set of feasible parameters from which it can be chosen. Let $\boldsymbol{\xi}^{(k)}, k = 1, 2, \ldots, N$, denote a set of i.i.d samples of the random vector $\tilde{\boldsymbol{\xi}}$. The SAA formulation is given as:

$$\min_{(\boldsymbol{\mu},\boldsymbol{\sigma})\in\Omega_{\boldsymbol{\mu},\boldsymbol{\sigma}}} \frac{1}{N}\sum_{k=1}^{N}\left(\max_{\boldsymbol{x}\in\mathcal{X}}\left(\boldsymbol{\mu} + \mathrm{Diag}(\boldsymbol{\sigma})\boldsymbol{\xi}^{(k)}\right)^T \boldsymbol{x}\right), \tag{2.4}$$

which is equivalent to the linear program:

$$\begin{aligned} \min_{\boldsymbol{\mu},\boldsymbol{\sigma},\boldsymbol{y}^{(k)}} \quad & \frac{1}{N}\sum_{k=1}^{N}\left(y_n^{(k)} - y_1^{(k)}\right) \\ \text{s.t.} \quad & y_j^{(k)} - y_i^{(k)} \geq \mu_{ij} + \sigma_{ij}\xi_{ij}^{(k)}, \quad \forall k = 1,\ldots,N, \\ & (\boldsymbol{\mu},\boldsymbol{\sigma}) \in \Omega_{\boldsymbol{\mu},\boldsymbol{\sigma}}. \end{aligned} \tag{2.5}$$

Convergence results for the objective value and the optimal solution as $N \uparrow \infty$ have been derived in Shapiro (2003).

2.2 Robust Project Crashing

There are two main challenges in the stochastic project crashing model. Firstly, solving the problem is computationally challenging as discussed. Secondly, the distribution of the activity durations is assumed to be given but in reality it might be often difficult to estimate the full distributional information. To overcome these challenges, robust optimization methods (see Ben-Tal et al. (2009)) have been developed to solve the project crashing problem. In this technique, uncertainty sets are used to model the activity durations and the objective is to minimize the worst-case makespan. Cohen et al. (2007) adopted the affinely adjustable robust formulation of Ben-Tal et al. (2004) to tackle the problem of minimizing the worst-case project makespan under interval and ellipsoidal uncertainty sets. The corresponding problems are formulated as linear and second order cone programs. Iancu and Trichakis (2014) extended the affinely adjustable robust counterpart to develop Pareto robust optimal solutions for project crashing. However as discussed in Wiesemann et al. (2012) while computationally tractable, the linear decision rule is sub-optimal for the robust project crashing problem. Chen et al. (2007) proposed new uncertainty sets for the robust project crashing problem to capture asymmetry information in the activity durations. Using linear decision rules they developed second order cone programs for the robust project crashing problem. Chen et al. (2008) considered more general decision rules beyond the linear decision rule for the crashing problem. While their results clearly demonstrate that the linear and piecewise linear decision rules are computationally tractable, it only solves a relaxation of the robust problem. Wiesemann et al. (2012) showed that for a given resource allocation, the problem of computing the worst-case makespan with some of the popular uncertainty sets is NP-hard. Examples of uncertainty sets for which worst-case makespan is known to be NP-hard to compute include the ellipsoidal uncertainty set and polyhedral uncertainty sets such as the intersection of a hypercube and a halfspace. Only for simple uncertainty sets such as the hypercube this problem is easy since the worst-case makespan is obtained simply by making all the activity durations take the maximum value. Wiesemann et al. (2012) proposed alternative methods to solve the robust project crashing problem by developing convergent bounds using path enumeration and path generation methods.

2.3 Distributionally Robust Project Crashing

Distributionally robust optimization is a more recent approach that has been used to tackle 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 chosen by nature from a set of distributions. Meilijson and Nadas (1979) studied 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. Birge and Maddox (1995) extended this bound to the case where the support for each activity duration is known and up to the first two moments (mean and standard deviations) of the random activity duration is provided. Bertsimas et al. (2004, 2006) extended this result to general higher order univariate moment information and developed a semidefinite program to compute the worst-case expected makespan. Mak et al. (2015) applied the dual of the formulation with two moments 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 is known, Natarajan et al. (2011) 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. Kong et al. (2013) 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. Natarajan and Teo (2016) recently showed that the complexity of computing this bound is closely related to characterizing the convex hull of quadratic forms of directed paths from the start node to the end node in the network. In a related stream of literature, Goh and Hall (2013) 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 that minimizes the worst-case expected makespan. Recently, Hanasusanto et al. (2016) 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 Doan and Natarajan (2012) 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. Li et al. (2014) extended this result to propose a bound on the worst-case expected makespan with information on the mean and covariance matrix.

In this paper, we build on these models to solve a class of distributionally robust project crashing problems in polynomial time. To the best of our knowledge, these are the only type of information structures for the moment models for which the problem is known to be solvable in polynomial time currently. 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 recent developments in first order saddle point methods can be used to solve the problem. As we demonstrate, this helps us solve much larger problems in practice. Lastly, we provide numerical insights into the nature of the crashing solution from distributionally robust models that we believe is insightful. Specifically, 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.

3 SOCP, SDP and Saddle Point Formulations

In this section, we study instances of the distributionally robust project crashing problem with moment information that is solvable in polynomial time in the size of the network. The random duration of activity (i, j) is denoted by $\tilde{t}_{ij}$. We assume that it is possible to vary the mean and the standard deviation of $\tilde{t}_{ij}$. Specifically 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 information on the correlation matrix that is available. We allow for the activity durations to be correlated. This often arises in projects when activities use the same set of resources such as equipment and manpower or are affected by common factors such as weather in a construction project (see Banerjee and Paul (2008)). When the joint distribution of $\tilde{\boldsymbol{t}}$ is known only to lie in a set of distributions $\Theta(\boldsymbol{\mu}, \boldsymbol{\sigma}; \boldsymbol{\rho})$ with the given mean, standard deviation and correlation information, the worst-case expected makespan is:

$$\max_{\theta \in \Theta(\boldsymbol{\mu}, \boldsymbol{\sigma}; \boldsymbol{\rho})} \mathbb{E}_\theta \left(\max_{\boldsymbol{x} \in \mathcal{X}} \tilde{\boldsymbol{t}}^T \boldsymbol{x} \right), \tag{3.1}$$

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 set $\mathcal{X}$ defined in (1.2). 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 Bertsimas et al. (2010), Natarajan et al. (2011), Wiesemann et al. (2012)). The distributionally robust project crashing problem of selecting the means and standard deviations to minimize the worst-case expected makespan for the project network is formulated as:

$$\min_{(\boldsymbol{\mu}, \boldsymbol{\sigma}) \in \Omega_{\boldsymbol{\mu}, \boldsymbol{\sigma}}} \quad \max_{\theta \in \Theta(\boldsymbol{\mu}, \boldsymbol{\sigma}; \boldsymbol{\rho})} \quad \mathbb{E}_\theta \left(\max_{\boldsymbol{x} \in \mathcal{X}} \tilde{\boldsymbol{t}}^T \boldsymbol{x} \right), \tag{3.2}$$

where $\Omega_{\boldsymbol{\mu},\boldsymbol{\sigma}}$ defines a convex set of feasible allocations for $(\boldsymbol{\mu},\boldsymbol{\sigma})$. The set $\Omega_{\boldsymbol{\mu},\boldsymbol{\sigma}}$ for example can be defined as:

$$\Omega_{\boldsymbol{\mu},\boldsymbol{\sigma}} = \left\{ (\boldsymbol{\mu},\boldsymbol{\sigma}) : \sum_{(i,j)\in\mathcal{A}} c_{ij}(\mu_{ij},\sigma_{ij}) \leq M,\ \underline{\mu}_{ij} \leq \mu_{ij} \leq \overline{\mu}_{ij},\ \underline{\sigma}_{ij} \leq \sigma_{ij} \leq \overline{\sigma}_{ij},\ \ \forall (i,j) \in \mathcal{A} \right\}, \tag{3.3}$$

where $\overline{\mu}_{ij}$ and $\overline{\sigma}_{ij}$ are the mean and standard deviation of the original 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 crashing. Further M is the amount of total cost budget, and $c_{ij}(\mu_{ij},\sigma_{ij})$ is the cost function which has the following properties: (a) $c_{ij}(\overline{\mu}_{ij},\overline{\sigma}_{ij}) = 0$, that means the extra cost of activity (i,j) is 0 under the original activity duration; and (b) $c_{ij}(\mu_{ij},\sigma_{ij})$ is a decreasing function of μ_{ij} and $\sigma_{i,j}$. In this formulation, it is possible to 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.

3.1 No Correlation Information

We first consider the marginal moment model where information on the mean and the standard deviation 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 two moments is defined as:

$$\Theta_{\text{mmm}}(\boldsymbol{\mu},\boldsymbol{\sigma}) = \left\{ \theta \in \mathbb{M}(\Re_m) : \mathbb{E}_\theta(\tilde{t}_{ij}) = \mu_{ij}, \mathbb{E}_\theta(\tilde{t}_{ij}^2) = \mu_{ij}^2 + \sigma_{ij}^2, \forall (i,j) \in \mathcal{A} \right\}, \tag{3.4}$$

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 two 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 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 Clark (1961), Banerjee and Paul (2008)), particularly when the activity durations are correlated for this very reason. 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. Assuming no correlation information, the worst-case expected makespan in (3.1) is equivalent to the optimal objective value of the following concave maximization problem over the convex hull of the set $\mathcal{X}$ (see Lemma 2 on page 458 in Natarajan et al. (2009)):

$$\max_{\boldsymbol{x}\in\mathcal{X}} f_{\text{mmm}}(\boldsymbol{\mu},\boldsymbol{\sigma},\boldsymbol{x}), \tag{3.5}$$

where

$$f_{\text{mmm}}(\boldsymbol{\mu},\boldsymbol{\sigma},\boldsymbol{x}) = \sum_{(i,j)\in\mathcal{A}} \left(\mu_{ij}x_{ij} + \sigma_{ij}\sqrt{x_{ij}(1-x_{ij})} \right). \tag{3.6}$$

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 (3.5) is computed using the following second order

cone program (SOCP):

$$\begin{aligned} \max_{\boldsymbol{x},\boldsymbol{t}} \quad & \sum_{(i,j)\in\mathcal{A}} (\mu_{ij} x_{ij} + \sigma_{ij} t_{ij}) \\ \text{s.t.} \quad & \boldsymbol{x} \in \mathcal{X}, \\ & \sqrt{t_{ij}^2 + \left(x_{ij} - \frac{1}{2}\right)^2} \leq \frac{1}{2}, \quad \forall (i,j) \in \mathcal{A}. \end{aligned} \tag{3.7}$$

The distributionally robust PCP (3.2) under the marginal moment model is hence formulated as a saddle point over the moment variables and arc criticality index variables as follows:

$$\min_{(\boldsymbol{\mu},\boldsymbol{\sigma})\in\Omega_{\boldsymbol{\mu},\boldsymbol{\sigma}}} \max_{\boldsymbol{x}\in\mathcal{X}} f_{\text{mmm}}(\boldsymbol{\mu},\boldsymbol{\sigma},\boldsymbol{x}). \tag{3.8}$$

One approach to solve the problem is to take the dual of the maximization problem in (3.7) in which case the distributionally robust PCP (3.8) is formulated as the following second order cone program:

$$\begin{aligned} \min_{\boldsymbol{\mu},\boldsymbol{\sigma},\boldsymbol{y},\boldsymbol{\alpha},\boldsymbol{\beta}} \quad & y_n - y_1 + \frac{1}{2} \sum_{(i,j)\in\mathcal{A}} (\alpha_{ij} - \beta_{ij}) \\ \text{s.t.} \quad & y_j - y_i - \beta_{ij} \geq \mu_{ij}, && \forall (i,j) \in \mathcal{A}, \\ & \sqrt{\sigma_{ij}^2 + \beta_{ij}^2} \leq \alpha_{ij}, && \forall (i,j) \in \mathcal{A} \\ & (\boldsymbol{\mu},\boldsymbol{\sigma}) \in \Omega_{\boldsymbol{\mu},\boldsymbol{\sigma}}. \end{aligned} \tag{3.9}$$

Several points regarding the formulation in (3.9) are important to take note of. Firstly, the formulation is tight, namely it solves the distributionally robust project crashing problem exactly. Secondly, such a dual formulation has been recently applied by Mak et al. (2015) 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 (3.9), 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 under 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 in identifying a polynomial time solvable project crashing instance.

3.2 Partial Correlation 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, before considering a general network.

3.2.1 Parallel Network

Consider a parallel project network with m activities where the correlation among the activities is known. In this case the set of probability distributions of the activity durations is defined as:

$$\Theta_{\text{cmm}}(\boldsymbol{\mu}, \boldsymbol{\sigma}; \boldsymbol{\rho}) = \left\{ \theta \in \mathbb{M}(\Re_m) : \mathbb{E}_\theta(\tilde{\boldsymbol{t}}) = \boldsymbol{\mu}, \mathbb{E}_\theta(\tilde{\boldsymbol{t}}\tilde{\boldsymbol{t}}^T) = \boldsymbol{\mu}\boldsymbol{\mu}^T + \text{Diag}(\boldsymbol{\sigma})\boldsymbol{\rho}\text{Diag}(\boldsymbol{\sigma}) \right\}, \tag{3.10}$$

where $\boldsymbol{\rho} \in \mathcal{S}_m^{++}$ denotes the correlation matrix and $\boldsymbol{\Sigma} = \text{Diag}(\boldsymbol{\sigma})\boldsymbol{\rho}\text{Diag}(\boldsymbol{\sigma})$ denotes the covariance matrix. We refer to this model as the cross moment model. The distributionally robust project crashing problem for a parallel network is then formulated as:

$$\min_{(\boldsymbol{\mu},\boldsymbol{\sigma})\in\Omega_{\boldsymbol{\mu},\boldsymbol{\sigma}}} \quad \max_{\theta\in\Theta_{\text{cmm}}(\boldsymbol{\mu},\boldsymbol{\sigma};\boldsymbol{\rho})} \quad \mathbb{E}_\theta \left(\max_{\boldsymbol{x}\in\boldsymbol{\Delta}} \tilde{\boldsymbol{t}}^T \boldsymbol{x} \right), \tag{3.11}$$

where the inner maximization is over the simplex since the network is parallel. The worst-case expected makespan in (3.11) is equivalent to the moment problem over the random vector $\tilde{\boldsymbol{\xi}}$ with the given first two moments:

$$\begin{array}{ll} \max & \mathbb{E}_\gamma \left(\max\limits_{\boldsymbol{x}\in\boldsymbol{\Delta}} \left(\boldsymbol{\mu} + \text{Diag}(\boldsymbol{\sigma})\tilde{\boldsymbol{\xi}} \right)^T \boldsymbol{x} \right) \\ \text{s.t.} & \mathbb{P}_\gamma(\tilde{\boldsymbol{\xi}} \in \Re_m) = 1, \\ & \mathbb{E}_\gamma(\tilde{\boldsymbol{\xi}}) = 0, \\ & \mathbb{E}_\gamma(\tilde{\boldsymbol{\xi}}\tilde{\boldsymbol{\xi}}^T) = \boldsymbol{\rho}. \end{array} \tag{3.12}$$

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 (3.12) can be solved as a semidefinite program:

$$\begin{array}{ll} \min\limits_{\lambda_0,\boldsymbol{\lambda},\boldsymbol{\Lambda}} & \lambda_0 + \langle \boldsymbol{\rho}, \boldsymbol{\Lambda} \rangle \\ \text{s.t.} & \begin{pmatrix} \lambda_0 - \mu_{ij} & \frac{1}{2}(\boldsymbol{\lambda} - \sigma_{ij}\boldsymbol{e}_{ij})^T \\ \frac{1}{2}(\boldsymbol{\lambda} - \sigma_{ij}\boldsymbol{e}_{ij}) & \boldsymbol{\Lambda} \end{pmatrix} \succeq 0, \quad \forall (i,j) \in \mathcal{A}. \end{array} \tag{3.13}$$

Strong duality holds in this case under the assumption that the correlation matrix is positive definite. Plugging it back into (3.11), we obtain the semidefinite program for distributionally robust project crashing problem over a parallel network as follows:

$$\begin{array}{ll} \min\limits_{\boldsymbol{\mu},\boldsymbol{\sigma},\lambda_0,\boldsymbol{\lambda},\boldsymbol{\Lambda}} & \lambda_0 + \langle \boldsymbol{\rho}, \boldsymbol{\Lambda} \rangle \\ \text{s.t.} & \begin{pmatrix} \lambda_0 - \mu_{ij} & \frac{1}{2}(\boldsymbol{\lambda} - \sigma_{ij}\boldsymbol{e}_{ij})^T \\ \frac{1}{2}(\boldsymbol{\lambda} - \sigma_{ij}\boldsymbol{e}_{ij}) & \boldsymbol{\Lambda} \end{pmatrix} \succeq 0, \quad \forall (i,j) \in \mathcal{A}, \\ & (\boldsymbol{\mu}, \boldsymbol{\sigma}) \in \Omega_{\boldsymbol{\mu},\boldsymbol{\sigma}}. \end{array} \tag{3.14}$$

We next provide an alternative reformulation of the project crashing problem in the spirit of (3.8) as a convex-concave saddle point problem where the number of variables in the formulation grow linearly in the number of arcs m. Unlike the original minimax formulation in (3.11) 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 (3.11) is solvable as a convex-concave saddle point problem:*

$$\min_{(\boldsymbol{\mu},\boldsymbol{\sigma})\in\Omega_{\boldsymbol{\mu},\boldsymbol{\sigma}}} \max_{\boldsymbol{x}\in\boldsymbol{\Delta}} f_{cmm}(\boldsymbol{\mu},\boldsymbol{\sigma},\boldsymbol{x}), \tag{3.15}$$

where

$$f_{cmm}(\boldsymbol{\mu},\boldsymbol{\sigma},\boldsymbol{x}) = \boldsymbol{\mu}^T\boldsymbol{x} + \text{trace}\left(\left(\boldsymbol{\Sigma}^{1/2}S(\boldsymbol{x})\boldsymbol{\Sigma}^{1/2}\right)^{1/2}\right). \tag{3.16}$$

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. The worst-case expected makespan under the cross moment model for a parallel network was studied in Ahipasaoglu et al. (2016) (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:

$$\max_{\theta\in\Theta_{\text{cmm}}(\boldsymbol{\mu},\boldsymbol{\sigma};\boldsymbol{\rho})} \mathbb{E}_\theta\left(\max_{\boldsymbol{x}\in\boldsymbol{\Delta}} \tilde{\boldsymbol{t}}^T\boldsymbol{x}\right) = \max_{\boldsymbol{x}\in\boldsymbol{\Delta}}\left(\boldsymbol{\mu}^T\boldsymbol{x} + \text{trace}\left(\left(\boldsymbol{\Sigma}^{1/2}S(\boldsymbol{x})\boldsymbol{\Sigma}^{1/2}\right)^{1/2}\right)\right), \tag{3.17}$$

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 (3.15) 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 Ahipasaoglu et al. (2016)). 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 $\boldsymbol{\Sigma}^{1/2} \in \mathcal{S}_m^{++}$. To see this, we apply Theorem 7.2 in Carlen (2010) which proves that the function $g(\boldsymbol{A}) = \text{trace}((\boldsymbol{B}^T\boldsymbol{A}^2\boldsymbol{B})^{1/2})$ is convex in $\boldsymbol{A} \in \mathcal{S}_m^{++}$ for a fixed $\boldsymbol{B} \in \Re_{m\times m}$. 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 $\boldsymbol{\Sigma}^{1/2} \in \mathcal{S}_m^{++}$. □

3.2.2 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 $[n-1]$ denote the subset of nodes $\{1,2,\ldots,n-1\}$ and $\mathcal{A}_i$ denote the set of arcs originating from node i for $i \in [n-1]$. Note that the sets $\mathcal{A}_i, i \in [n-1]$ 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, we define the set of distributions of the random vector $\tilde{\boldsymbol{t}}$ as follows:

$$\Theta_{\text{nmm}}(\boldsymbol{\mu}_i,\boldsymbol{\sigma}_i;\boldsymbol{\rho}_i\forall i) = \left\{\theta \in \mathbb{M}(\Re_n) : \mathbb{E}_\theta(\tilde{\boldsymbol{t}}_i) = \boldsymbol{\mu}_i, \mathbb{E}_\theta(\tilde{\boldsymbol{t}}_i\tilde{\boldsymbol{t}}_i^T) = \boldsymbol{\mu}_i\boldsymbol{\mu}_i^T + \text{Diag}(\boldsymbol{\sigma}_i)\boldsymbol{\rho}_i\text{Diag}(\boldsymbol{\sigma}_i), \forall i \in [n-1]\right\}, \tag{3.18}$$

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 (3.18), we assume that the correlation between activity durations of the arcs that originate from different nodes is unknown. This is 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 often 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 Fulkerson (1962)). On the other hand, in the model discussed in this paper, the activity durations are arbitrarily dependent for arcs exiting different nodes. Under partial correlation information, the distributionally robust project crashing problem for a general network is formulated as:

$$\min_{(\boldsymbol{\mu},\boldsymbol{\sigma})\in\Omega_{\boldsymbol{\mu},\boldsymbol{\sigma}}} \quad \max_{\theta\in\Theta_{\mathrm{nmm}}(\boldsymbol{\mu}_i,\boldsymbol{\sigma}_i;\boldsymbol{\rho}_i\forall i)} \quad \mathbb{E}_\theta\left(\max_{\boldsymbol{x}\in\mathcal{X}} \tilde{\boldsymbol{t}}^T\boldsymbol{x}\right). \tag{3.19}$$

Under the nonoverlapping marginal moment model, the worst-case expected makespan in (3.19) 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 Li et al. (2014)):

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

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 (3.19) is solvable as the semidefinite program:

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

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 $\boldsymbol{x}_i \in \Re^{|\mathcal{A}_i|}$ have x_{ij} as its j^{th} coordinate and define $S(\boldsymbol{x}_i) = \mathrm{Diag}(\boldsymbol{x}_i) - \boldsymbol{x}_i\boldsymbol{x}_i^T$ for all $i\in[n-1]$. Under the nonoverlapping multivariate marginal moment model for a general network, the distributionally robust*

PCP (3.19) is solvable as a convex-concave saddle point problem:

$$\min_{(\boldsymbol{\mu},\boldsymbol{\sigma})\in\Omega_{(\boldsymbol{\mu},\boldsymbol{\sigma})}} \max_{\boldsymbol{x}\in\mathcal{X}} f_{nmm}(\boldsymbol{\mu},\boldsymbol{\sigma},\boldsymbol{x}), \tag{3.20}$$

where

$$f_{nmm}(\boldsymbol{\mu},\boldsymbol{\sigma},\boldsymbol{x}) = \sum_{i=1}^{n-1} \left(\boldsymbol{\mu}_i^T \boldsymbol{x}_i + \text{trace}\left(\boldsymbol{\Sigma}_i^{1/2} S(\boldsymbol{x}_i) \boldsymbol{\Sigma}_i^{1/2}\right)\right). \tag{3.21}$$

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 $\boldsymbol{\Sigma} = \text{Diag}(\boldsymbol{\sigma})\boldsymbol{\rho}\text{Diag}(\boldsymbol{\sigma})$ and $T(\boldsymbol{x}) = \boldsymbol{\Sigma}^{1/2} S(\boldsymbol{x}) \boldsymbol{\Sigma}^{1/2}$. Under the cross moment model with a parallel network, the gradient of f_{cmm} in (3.16) 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 (3.16) with respect to $(\boldsymbol{\mu},\boldsymbol{\sigma})$ is given as:

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

Proof. See Appendix. □

The optimality condition for (3.15) is then given as:

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

where $P_S(\cdot)$ denotes the projection onto a set S and ∇ denotes the partial derivative.

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

respect to $\boldsymbol{x}$ and $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ are

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

where

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

The optimality condition for (3.20) is then given as:

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

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 Ahipasaoglu et al. (2016) to solve the inner maximization problem in a discrete choice problem setting.

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: $\textit{criteria} \leftarrow \epsilon + 1$;

while $\textit{criteria} > \epsilon$ **do**

- $\boldsymbol{z} \leftarrow P_{\mathcal{X}}(\boldsymbol{x}_0 + \nabla_{\boldsymbol{x}} f(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{x}_0))$,
- $\textit{criteria} \leftarrow \|\boldsymbol{z} - \boldsymbol{x}_0\|$,
- $\boldsymbol{x} \leftarrow \boldsymbol{x}_0 + \gamma(\boldsymbol{z} - \boldsymbol{x}_0)$, 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:

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

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 Fiacco and Ishizuka (1990)). 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 Rockafellar (1997)). Hence F is monotone:

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

□

The optimality condition is then equivalent to the following variational inequality (Eaves (1971)) :

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

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 (He (1997)). The algorithm is as follows:

Algorithm 2: Projection and contraction algorithm for monotone variational inequalities

Input: Parameters for set $\Omega_{\boldsymbol{\mu},\boldsymbol{\sigma}}$, 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**

- $\beta \leftarrow \alpha$
- $\boldsymbol{res} \leftarrow (\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0) - P_{\Omega_{\boldsymbol{\mu},\boldsymbol{\sigma}}}((\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0) - \beta F(\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0))$
- $\boldsymbol{d} \leftarrow \boldsymbol{res} - \beta[F(\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0) - F(P_{\Omega_{\boldsymbol{\mu},\boldsymbol{\sigma}}}((\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0) - \beta F(\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0)))]$
- $criteria \leftarrow \|\boldsymbol{res}\|$
- **while** $\langle \boldsymbol{res}, \boldsymbol{d}\rangle < \delta \|\boldsymbol{res}\|^2$ **do**
  - $\beta \leftarrow \beta/2$
  - $\boldsymbol{res} \leftarrow (\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0) - P_{\Omega_{\boldsymbol{\mu},\boldsymbol{\sigma}}}((\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0) - \beta F(\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0))$
  - $\boldsymbol{d} \leftarrow \boldsymbol{res} - \beta[F(\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0) - F(P_{\Omega_{\boldsymbol{\mu},\boldsymbol{\sigma}}}((\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0) - \beta F(\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0)))]$
- **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

5 Numerical Experiments

In this section, we report the results from numerical tests for the distributionally robust project crashing problem. In the numerical tests, the feasible set of $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ is defined as

$$\Omega_{\boldsymbol{\mu},\boldsymbol{\sigma}} = \Big\{(\boldsymbol{\mu}, \boldsymbol{\sigma}) : \sum_{(i,j)\in\mathcal{A}} c_{ij}(\mu_{ij}, \sigma_{ij}) \le M,\ \underline{\mu}_{ij} \le \mu_{ij} \le \overline{\mu}_{ij},\ \underline{\sigma}_{ij} \le \sigma_{ij} \le \overline{\sigma}_{ij},\ \ \forall (i,j) \in \mathcal{A}\Big\}. \tag{5.1}$$

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

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

where $a_{ij}^{(1)}, a_{ij}^{(2)}, b_{ij}^{(1)}$ and $b_{ij}^{(2)}$ are given nonnegative real numbers. These cost functions are chosen such that: (a) $c_{ij}(\overline{\mu}_{ij}, \overline{\sigma}_{ij}) = 0$, namely the cost of activity (i,j) is 0 under normal duration; and (b) $c_{ij}(\mu_{ij}, \sigma_{ij})$ is a convex decreasing function of μ_{ij} and $\sigma_{i,j}$.

We compare the distributionally robust project crashing solution with the following models:

1. Deterministic PCP: 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:

$$\min_{(\boldsymbol{\mu},\boldsymbol{\sigma})\in\Omega_{\boldsymbol{\mu},\boldsymbol{\sigma}}:\boldsymbol{\sigma}=\bar{\boldsymbol{\sigma}}} \max_{\boldsymbol{x}\in\mathcal{X}} \sum_{(i,j)\in\mathcal{A}} \mu_{ij} x_{ij}. \tag{5.3}$$

2. Heuristic model (Kim et al. (2007)): The crashing solution in this case is given as:

$$\min_{(\boldsymbol{\mu},\boldsymbol{\sigma})\in\Omega_{\boldsymbol{\mu},\boldsymbol{\sigma}}} \max_{\boldsymbol{x}\in\mathcal{X}} \sum_{(i,j)\in\mathcal{A}} (\mu_{ij} + \kappa \cdot \sigma_{ij}) x_{ij}. \tag{5.4}$$

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

3. Sample Average Approximation (SAA): Assume that the activity duration vector follows a multivariate normal distribution in which case the SAA solution is given by (2.5). In our experiment, we use a sample size of $N = 5000$.

Example 1

In the first example, we consider the small project network in Figure 1. The mean and standard deviation for the original activity durations are given as

$$\overline{\boldsymbol{\mu}}_1 = \begin{pmatrix} \bar{\mu}_{12} \\ \bar{\mu}_{13} \\ \bar{\mu}_{14} \end{pmatrix} = \begin{pmatrix} 2 \\ 2.5 \\ 4 \end{pmatrix}, \quad \overline{\boldsymbol{\sigma}}_1 = \begin{pmatrix} \bar{\sigma}_{12} \\ \bar{\sigma}_{13} \\ \bar{\sigma}_{14} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}, \quad \boldsymbol{\rho}_1 = \begin{pmatrix} 1 & 0.5 & 0 \\ 0.5 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\bar{\boldsymbol{\mu}}_2 = \begin{pmatrix} \bar{\mu}_{23} \\ \bar{\mu}_{24} \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \quad \bar{\boldsymbol{\sigma}}_2 = \begin{pmatrix} \bar{\sigma}_{23} \\ \bar{\sigma}_{24} \end{pmatrix} = \begin{pmatrix} 1.5 \\ 2 \end{pmatrix}, \quad \boldsymbol{\rho}_2 = \begin{pmatrix} 1 & -0.2 \\ -0.2 & 1 \end{pmatrix}$$

$$\bar{\boldsymbol{\mu}}_3 = \bar{\mu}_{34} = 4, \qquad \bar{\boldsymbol{\sigma}}_3 = \bar{\sigma}_{34} = 3.$$

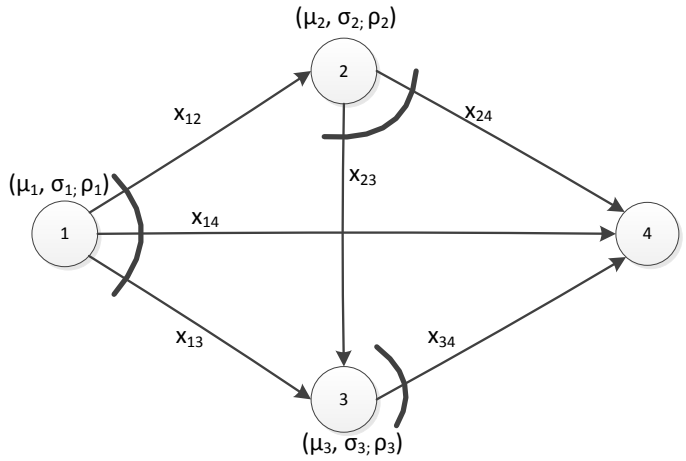


Figure 1: Project Network

For the SAA approach, the correlations between the activity durations originating from different nodes is set to 0. Similarly for the CMM approach, we convert the network to a parallel network by path enumeration. The minimum value of the mean and standard deviation that can be achieved by crashing is set to half of the mean and standard deviation of the original activity durations. As the cost budget M is varied from 0 to $\sum_{(i,j)} c_{ij}(\underline{\mu}_{ij}, \underline{\sigma}_{ij})$, the time cost trade-off of the six models: Deterministic PCP (5.3), Heuristic PCP (5.4), SAA (2.5), and the Distributionally Robust PCP under MMM, CMM and NMM is shown in Figure 2.

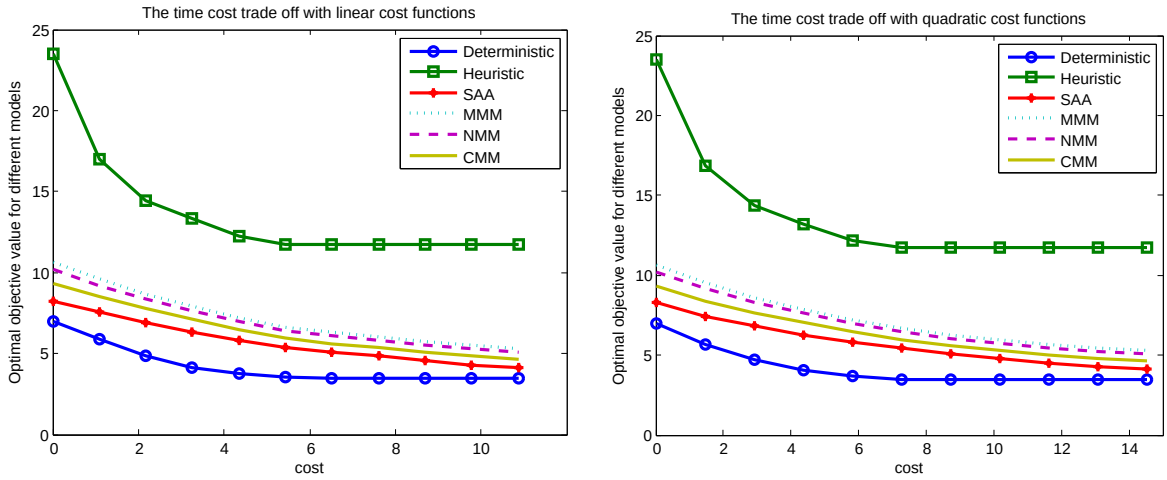


Figure 2: Optimal objective value as cost budget increases

From Figure 2, we see that all the objective functions decreases 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 distributionally robust PCP are tight upper bounds for the expected makespan under MMM, CMM and NMM. We also find the objective values 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 value of other models implying it is a poor approximation of the expected makespan. We also see that the objective value of deterministic PCP and heuristic PCP does not change when the cost budget exceeds a certain value (close to half of the budget upper bound $\sum_{(i,j)} c_{ij}(\underline{\mu}_{ij}, \underline{\sigma}_{ij})$). This implies that under these models the critical paths of the two models do not change beyond this budget and the mean and standard deviation of the activity durations on the critical paths has been crashed to minimum. In this case, reducing the mean and variance of other activity durations will not change the objective value of the deterministic PCP and heuristic PCP. In comparison, under uncertainty the objective values of SAA and the distributionally robust PCP always decreases as the cost increases.

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 3. As expected, using the optimal solution obtained by SAA, we get the smallest expected makespan.

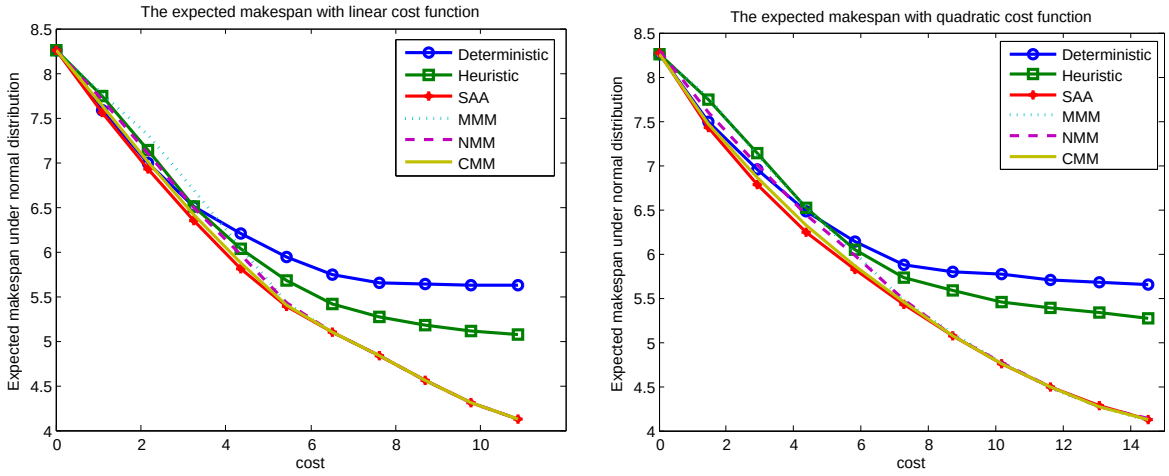
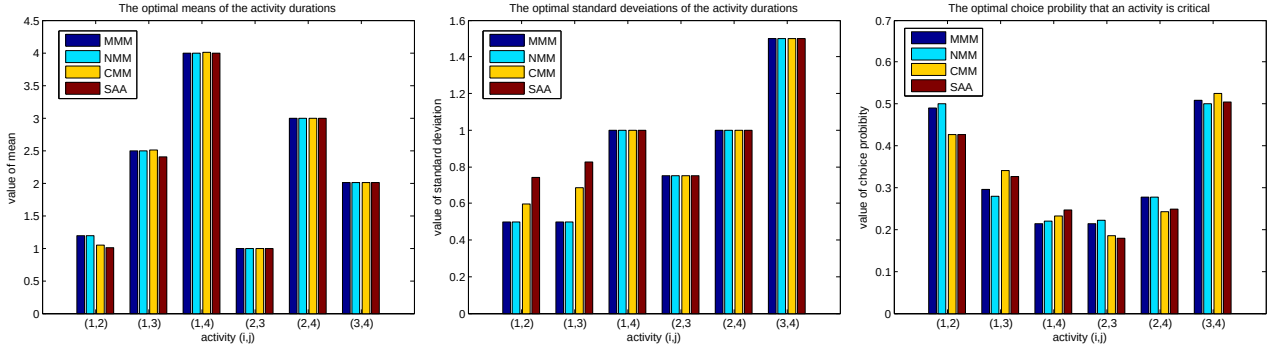


Figure 3: The expected makespan under the multivariate normal distribution

When the cost budget is small, the deterministic PCP also provides 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 heuristic PCP has better performance that the deterministic PCP, but still the gap of the expected makespan between this model and SAA is large. 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 linear and the cost budget $M = \frac{1}{2}\sum_{(i,j)} c_{ij}(\underline{\mu}_{ij}, \underline{\sigma}_{ij}) = 5.4375$. In this case the expected makespans are 5.43 (MMM), 5.43 (NMM), 5.40 (CMM) and 5.39 (SAA), and the standard deviation of the makespans are 1.10

(MMM), 1.10 (NMM), 1.14 (CMM) and 1.18 (SAA). The optimal solutions of the four models are shown in Figure 4. The optimal solutions from the four models are fairly close. For activities (1,2) and (1,3), SAA tends to crash



(a) Optimal mean (b) Optimal standard deviation (c) Optimal choice probability

Figure 4: Optimal solutions obtained by MMM, NMM, CMM and SAA

the mean more while the robust models tend to crash the standard deviation more. We validate this observation with more numerical examples later.

Example 2

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

1. For every activity, the mean and the standard deviation 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 $a_{ij}^{(1)} \sim U(1, 2)$, $a_{ij}^{(2)} \sim U(0, 1)$, $b_{ij}^{(1)} \sim U(1, 2)$ and $b_{ij}^{(2)} \sim U(0, 1)$. The amount of the cost budget is chosen as $\frac{1}{4}\sum_{(i,j)} c_{ij}(\underline{\mu}_{ij}, \underline{\sigma}_{ij})$.

We first consider a simple case with two parallel activities. In Figure 5, 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 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 $\text{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$. 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 6 and Table 1. When the activities are uncorrelated, with the optimal solution obtained from MMM and CMM, the distribution of the makespan is very

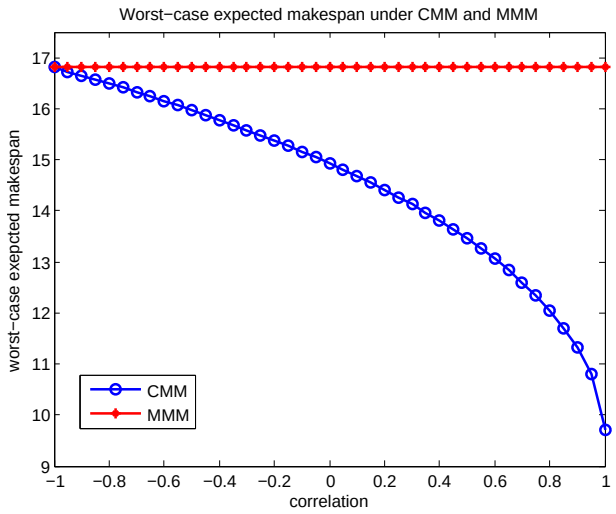
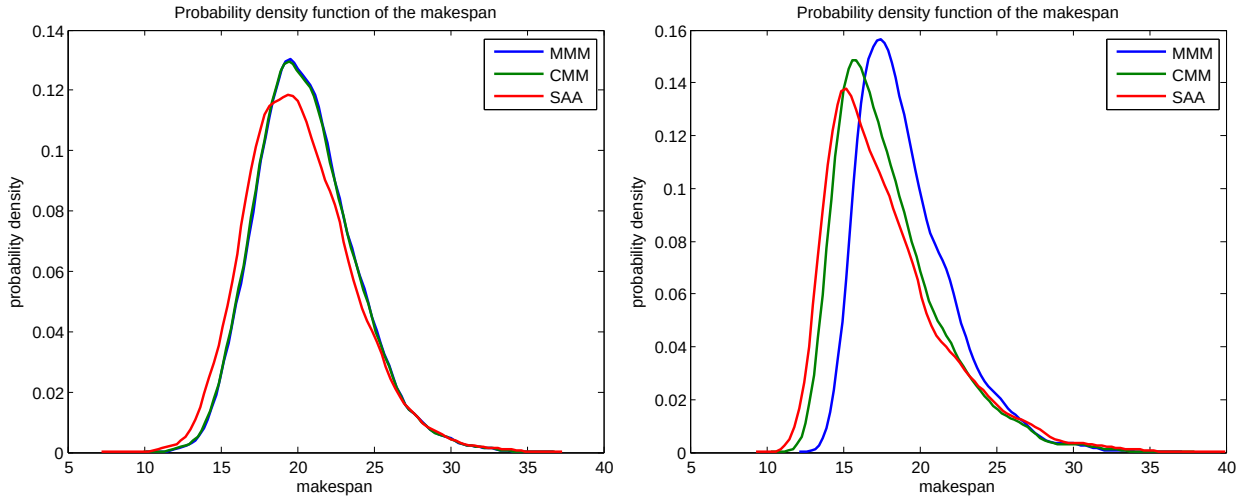


Figure 5: Optimal value of f_{cmm} and f_{mmm}

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 is the largest in comparison to the distributionally robust models indicating that the robust models provide a reduction in the variability of the makespan.



(a) Uncorrelated activity durations

(b) Highly correlated activity durations

Figure 6: Probability density function 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 (3.14) and the saddle point reformulation (3.15) for the distributionally robust PCP under CMM. To solve the SDP (3.14), we used CVX, a package for specifying and solving convex programs (Grant and Boyd (2014, 2008)) with the solver SDPT3 (Toh et al. (1999), Tütüncü et al. (2003)). We set the accuracy of the SDP solver with "cvx_precision low". To solve the saddle point problem (3.15), we use Algorithm 2 with tolerance $\epsilon = 10^{-3}$. In our implementation, we used the mean and standard deviation 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 the saddle point algorithm is able to solve problems to reasonable accuracy for all practical purposes much faster than interior point method solvers.

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

Arcs	CPU time in seconds		Objective value	
	SDP solver	Saddle point algorithm	SDP solver	Saddle point algorithm
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

In this third example, we consider a grid network (see Figure 7). The size of the problem is determined by its width and height. Let width=m and height=n in which case there are $(m+1)(n+1)$ nodes, $m(n+1)+n(m+1)$ activities and $\binom{m+n}{m}$ possible critical paths in the project. For example, in Figure 7, $m = 6$ and $n = 4$, then there are 35 nodes, 58 activities and 210 possible critical paths.

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

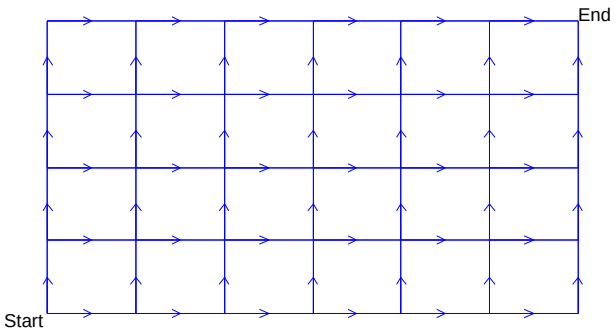


Figure 7: Grid project network with width = 6, height=4

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 $a_{ij}^{(1)} \sim U(2,4)$, $a_{ij}^{(2)} \sim U(0,1)$, $b_{ij}^{(1)} \sim U(1,2)$ 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 (5.3), an optimal strategy is to reduce the mean 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 (5.3) and the heuristic PCP (5.4) 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 (5.3), Heuristic PCP (5.4), 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 (5.3). We define the "reduction" as the percentage of the extra expected makespan achieved by 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 reduction improvement

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

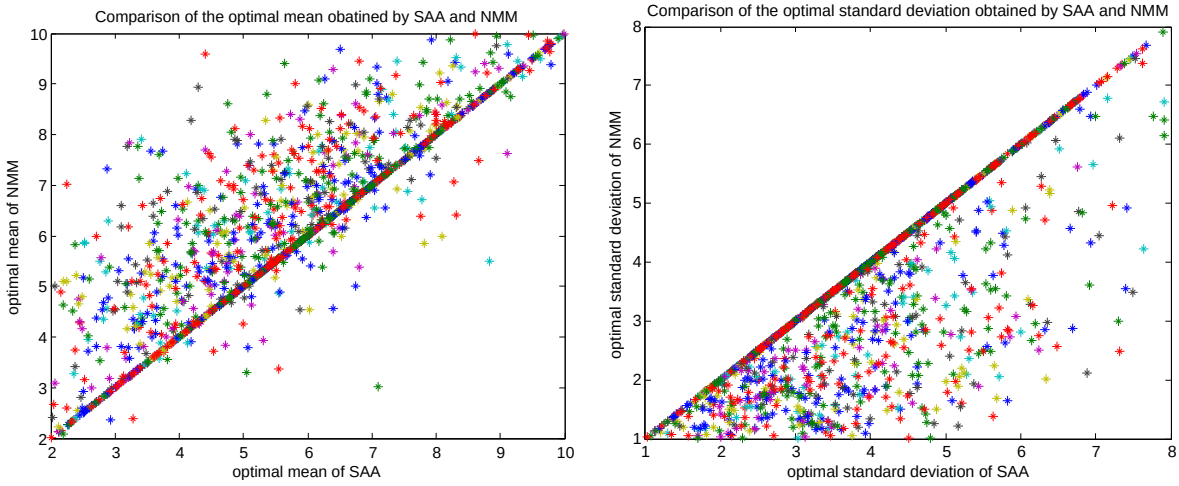


Figure 8: Optimal solution comparison of SAA and NMM

of the distributionally robust PCP is much larger than the reduction 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 is 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 8. From the figure, we see that the SAA model tends to crash the means more while NMM tends to crash the standard deviations more.

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 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 use saddle point 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 problems 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 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.

Appendix

Proof of Proposition 2

We consider the inner maximization problem of (3.19), 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

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

Applying Theorem 15 on page 467 in Li et al. (2014), the worst-case expected makespan in (6.1) is formulated as the following SDP:

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

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

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

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

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

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

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 $(\boldsymbol{Y}_i, \boldsymbol{x}_i), i \in [n-1]$ 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 $(\boldsymbol{Y}^*_i, \boldsymbol{x}^*_i), i \in [n-1]$. For an $i \in [n-1]$ 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:

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

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). $\square$

Given the formulation (6.2) and using Theorem 2 from Ahipasaoglu et al. (2016) for each node i, it is easy to verify that the SDP (6.3) is equivalent to:

$$\max_{\boldsymbol{x}\in\mathcal{X}} \sum_{i=1}^{n-1} \left(\boldsymbol{\mu}_i^T \boldsymbol{x}_i + \text{trace}\left(\left(\boldsymbol{\Sigma}_i^{1/2} S(\boldsymbol{x}_i) \boldsymbol{\Sigma}_i^{1/2} \right)^{1/2} \right) \right). \tag{6.4}$$

Therefore, the project crashing problem is equivalent to

$$\min_{(\boldsymbol{\mu},\boldsymbol{\sigma})\in\Omega_{\boldsymbol{\mu},\boldsymbol{\sigma}}} \max_{\boldsymbol{x}\in\mathcal{X}} \sum_{i=1}^{n} \left(\boldsymbol{\mu}_i^T \boldsymbol{x}_i + \text{trace}\left(\left(\boldsymbol{\Sigma}_i^{1/2} S(\boldsymbol{x}_i) \boldsymbol{\Sigma}_i^{1/2} \right)^{1/2} \right) \right), \tag{6.5}$$

where $\boldsymbol{\Sigma}_i = \text{Diag}(\boldsymbol{\sigma}_i)\boldsymbol{\rho}_i\text{Diag}(\boldsymbol{\sigma}_i)$ is a matrix function of $\boldsymbol{\sigma}_i$, $S(\boldsymbol{x}_i) = \text{Diag}(\boldsymbol{x}_i) - \boldsymbol{x}_i\boldsymbol{x}_i^T$. 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 Ahipasaoglu et al. (2016). 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 $f(\boldsymbol{A}) = \text{trace}((\boldsymbol{ASA})^{1/2})$ with $\boldsymbol{A}$ defined on the set of positive definite matrices.

Proposition 5. *Function $f : S_{++}^n \to \Re$ is defined as $f(\boldsymbol{A}) = \text{trace}((\boldsymbol{ASA})^{1/2})$ 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*

$$g(\boldsymbol{A}) = \frac{1}{2}[\boldsymbol{A}^{-1}(\boldsymbol{ASA})^{1/2} + (\boldsymbol{ASA})^{1/2}\boldsymbol{A}^{-1}]. \tag{6.6}$$

Proof. Let $F(\boldsymbol{A}) = (\boldsymbol{ASA})^{1/2}$, then $f(\boldsymbol{A}) = \text{trace}(F(\boldsymbol{A}))$. 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:

$$(\boldsymbol{ASA})^{1/2}L_{1/2} + L_{1/2}(\boldsymbol{ASA})^{1/2} = \boldsymbol{E_D}(t, \boldsymbol{A}). \tag{6.7}$$

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

$$\begin{aligned} f(\boldsymbol{A} + t\boldsymbol{D}) - f(\boldsymbol{A}) &= \text{trace}(F(\boldsymbol{A} + t\boldsymbol{D}) - F(\boldsymbol{A})) \\ &= \text{trace}((\boldsymbol{ASA})^{-1/2}(\boldsymbol{ASA})^{1/2}[F(\boldsymbol{A} + t\boldsymbol{D}) - F(\boldsymbol{A})]) \end{aligned}$$

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

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 $\boldsymbol{S}$ might be singular.

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

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

Proof. Let $f(\epsilon, \boldsymbol{A}) = \operatorname{trace}((\boldsymbol{A}(\boldsymbol{S}+\epsilon\boldsymbol{I})\boldsymbol{A})^{1/2})$, $\epsilon \in (0,1]$, then $f(\boldsymbol{A}) = \lim_{\epsilon\downarrow 0} f(\epsilon, \boldsymbol{A})$. From Theorem 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 Rudin (1964)):

(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 Rudin (1964)):

$$\lim_{t\to 0}\sup_{\epsilon\in(0,1]} |G(\epsilon, t) - G(\epsilon, 0)| = 0. \tag{6.9}$$

Next, we prove the result of (6.9). By the mean value theorem and the proof of Theorem 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

$$G(\epsilon, t) - G(\epsilon, 0) = \langle g(\epsilon, \boldsymbol{A} + t_1\boldsymbol{D}), \boldsymbol{D})\rangle - \langle g(\epsilon, \boldsymbol{A}), \boldsymbol{D})\rangle. \tag{6.10}$$

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

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

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

$$(\boldsymbol{A}_t\boldsymbol{S}_\epsilon\boldsymbol{A}_t)^{1/2}L + L(\boldsymbol{A}_t\boldsymbol{S}_\epsilon\boldsymbol{A}_t)^{1/2} = (\boldsymbol{A}_t\boldsymbol{S}_\epsilon\boldsymbol{D} + \boldsymbol{D}\boldsymbol{S}_\epsilon\boldsymbol{A}_t) \tag{6.11}$$

and

$$\begin{aligned}
\frac{\partial h(\epsilon, t)}{\partial t} &= \lim_{\Delta t\to 0}\frac{1}{\Delta t}\text{trace}\left(\boldsymbol{D}\boldsymbol{A}_t^{-1}(\Delta t L - \Delta t\boldsymbol{D}\boldsymbol{A}^{-1}(\boldsymbol{A}_t\boldsymbol{S}_\epsilon\boldsymbol{A}_t)^{1/2}\right) + o(\Delta t) \\
&= \text{trace}\left(\boldsymbol{D}\boldsymbol{A}_t^{-1}L\right) - \text{trace}\left((\boldsymbol{D}\boldsymbol{A}_t^{-1})^2(\boldsymbol{A}_t\boldsymbol{S}_\epsilon\boldsymbol{A}_t)^{1/2}\right).
\end{aligned} \tag{6.12}$$

L is the solution of the Sylvester equation (6.11) 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 (6.12) 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 (6.12), we know that

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

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 (6.11).

Let $\boldsymbol{P}^T\boldsymbol{\Lambda}\boldsymbol{P}$ be the eigenvalue decomposition of $\boldsymbol{A}_t\boldsymbol{S}_\epsilon\boldsymbol{A}_t$. Then (6.11) 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

$$\begin{aligned}\boldsymbol{\Lambda}^{1/2}\bar{L} + \bar{L}\boldsymbol{\Lambda}^{1/2} &= \boldsymbol{\Lambda}\boldsymbol{P}\boldsymbol{A}_t^{-1}\boldsymbol{D}\boldsymbol{P}^T + \boldsymbol{P}\boldsymbol{D}\boldsymbol{A}_t^{-1}\boldsymbol{P}^T\boldsymbol{\Lambda}\\ &= \boldsymbol{\Lambda}\boldsymbol{E} + \boldsymbol{E}^T\boldsymbol{\Lambda},\end{aligned} \tag{6.13}$$

where $\boldsymbol{E} = \boldsymbol{P}\boldsymbol{A}_t^{-1}\boldsymbol{D}\boldsymbol{P}^T$. The solution for equation (6.13) 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 (6.10) 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*

$$V(\boldsymbol{\sigma}) = \text{trace}\Big(\big([\boldsymbol{A}\text{Diag}(\boldsymbol{\sigma})\boldsymbol{C}\text{Diag}(\boldsymbol{\sigma})\boldsymbol{A}^T]^{1/2}\boldsymbol{S}[\boldsymbol{A}\text{Diag}(\boldsymbol{\sigma})\boldsymbol{C}\text{Diag}(\boldsymbol{\sigma})\boldsymbol{A}^T]^{1/2}\big)^{1/2}\Big), \tag{6.14}$$

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

$$grad(\boldsymbol{\sigma}) = \mathrm{diag}\Big(\boldsymbol{A}^T h(\boldsymbol{\sigma})^{-1}\big(h(\boldsymbol{\sigma})\boldsymbol{S}h(\boldsymbol{\sigma})\big)^{1/2} h(\boldsymbol{\sigma})^{-1}\boldsymbol{A}\mathrm{Diag}(\boldsymbol{\sigma})\boldsymbol{C}\Big), \tag{6.15}$$

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

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

$$\|h(\boldsymbol{\sigma} + t\boldsymbol{v}) - h(\boldsymbol{\sigma}) - L(\boldsymbol{\sigma}, t\boldsymbol{v})\| = o(t). \tag{6.16}$$

By simple calculation, we have

$$h(\boldsymbol{\sigma} + t\boldsymbol{v}) - h(\boldsymbol{\sigma}) = (h(\boldsymbol{\sigma}) + \boldsymbol{E}_{\boldsymbol{v}}(t, \boldsymbol{\sigma}))^{1/2} - h(\boldsymbol{\sigma}), \tag{6.17}$$

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

$$\|h(\boldsymbol{\sigma} + t\boldsymbol{v}) - h(\boldsymbol{\sigma}) - L_{1/2}(h(\boldsymbol{\sigma})^2, \boldsymbol{E}_{\boldsymbol{v}}(t, \boldsymbol{\sigma}))\| = o(\|\boldsymbol{E}_{\boldsymbol{v}}(t, \boldsymbol{\sigma})\|) = o(t). \tag{6.18}$$

By (6.16) and (6.18), we have

$$\|L(\boldsymbol{\sigma}, t\boldsymbol{v}) - L_{1/2}(h(\boldsymbol{\sigma})^2, \boldsymbol{E}_{\boldsymbol{v}}(t, \boldsymbol{\sigma}))\| = o(t). \tag{6.19}$$

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

$$h(\boldsymbol{\sigma})L(\boldsymbol{\sigma}, t\boldsymbol{v}) + L(\boldsymbol{\sigma}, t\boldsymbol{v})h(\boldsymbol{\sigma}) = t\boldsymbol{A}[\mathrm{Diag}(\boldsymbol{\sigma})\boldsymbol{C}\mathrm{Diag}(\boldsymbol{v}) + \mathrm{Diag}(\boldsymbol{v})\boldsymbol{C}\mathrm{Diag}(\boldsymbol{\sigma})]\boldsymbol{A}^T + o(t). \tag{6.20}$$

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 Theorem 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 (6.20), we know that

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

Then

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

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

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

Ahipasaoglu, S. D., X. Li, K. Natarajan. 2016. A convex optimization approach for computing correlated choice probabilities with many alternatives. *Available at http://www.optimization-online.org/DB_HTML/2013/09/4034.html* .

Banerjee, A., A. Paul. 2008. On path correlation and pert bias. *European Journal of Operational Research* **189** 1208–1216.

Ben-Tal, A., L. El Ghaoui, A. Nemirovski. 2009. Robust optimization. *Princeton Series in Applied Mathematics* .

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

Bertsimas, D., X. V. Doan, K. Natarajan, C-P. Teo. 2010. Models for minimax stochastic linear optimization problems with risk aversion. *Mathematics of Operations Research* **35**(3) 580–602.

Bertsimas, D., K. Natarajan, C-P. Teo. 2004. Probabilistic combinatorial optimization: Moments, semidefinite programming, and asymptotic bounds. *SIAM Journal on Optimization* **15**(1) 185–209.

Bertsimas, D., K. Natarajan, C-P. Teo. 2006. Persistence in discrete optimization under data uncertainty. *Mathematical Programming* **108**(2-3) 251–274.

Birge, J. R., M. J. Maddox. 1995. Bounds on expected project tardiness. *Operations Research* **43**(5) 838–850.

Bowman, R. A. 1994. Stochastic gradient-based time-cost tradeoffs in pert networks using simulation. *Annals of Operations Research* **53**(1) 533–551.

Burt Jr, J. M., M. B. Garman. 1971. Conditional monte carlo: A simulation technique for stochastic network analysis. *Management Science* **18**(3) 207–217.

Carlen, E. 2010. Trace inequalities and quantum entropy: an introductory course. *Entropy and the Quantum* **529** 73–140.

Chen, X., M. Sim, P. Sun. 2007. A robust optimization perspective on stochastic programming. *Operations Research* **55**(6) 1058–1071.

Chen, X., M. Sim, P. Sun, J. Zhang. 2008. A linear decision-based approximation approach to stochastic programming. *Operations Research* **56**(2) 344–357.

Clark, E., C. 1961. The greatest of a finite set of random variables. *Operations Research* **9**(2) 145–162.

Cohen, I., B. Golany, A. Shtub. 2007. The stochastic time–cost tradeoff problem: a robust optimization approach. *Networks* **49**(2) 175–188.

De, P, E. J. Dunne, J. B. Ghosh, C. E. Wells. 2007. Complexity of the discrete time-cost tradeoff problems for project networks. *Operations Research* **45**(2) 302–306.

Doan, X. V., K. Natarajan. 2012. On the complexity of non-overlapping multivariate marginal bounds for probabilistic combinatorial optimization. *Operations Research* **60**(1) 138–149.

Eaves, B.C. 1971. On the basic theorem of complementarity. *Mathematical Programming* **1**(1) 68–75.

Falk, J. E., J. L. Horowitz. 1972. Critical path problems with concave cost-time curves. *Management Science* **19**(4) 446–455.

Fiacco, A. V., Y. Ishizuka. 1990. Sensitivity and stability analysis for nonlinear programming. *Annals of Operations Research* **27**(1) 215–235.

Fu, M. 2015. *Handbook of Simulation Optimization*. Springer.

Fulkerson, D.R. 1961. A network flow computation for project cost curves. *Management Science* **7**(2) 167–178.

Fulkerson, D.R. 1962. Expected critical path lengths in pert networks. *Operations Research* **10**(6) 808–817.

Goh, J., N. G. Hall. 2013. Total cost control in project management via satisficing. *Management Science* **59**(6) 1354–1372.

Grant, M., S. Boyd. 2008. Graph implementations for nonsmooth convex programs. V. Blondel, S. Boyd, H. Kimura, eds., *Recent Advances in Learning and Control*. Lecture Notes in Control and Information Sciences, Springer-Verlag Limited, 95–110.

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

Hagstrom, J. N. 1988. Computational complexity of PERT problems. *Networks* **18**(2) 139–147.

Hanasusanto, G. A., V. Roitch, D. Kuhn, W. Wiesemann. 2016. Ambiguous joint chance constraints under mean and dispersion information. *To appear in Operations Research* .

He, B. 1997. A class of projection and contraction methods for monotone variational inequalities. *Applied Mathematics and Optimization* **35**(1) 69–76.

Iancu, D. A., N. Trichakis. 2014. Pareto efficiency in robust optimization. *Management Science* **60**(1) 130–147.

Kelley Jr, J.E. 1961. Critical-path planning and scheduling: Mathematical basis. *Operations Research* **9**(3) 296–320.

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

Kim, S-J., S. P. Boyd, S. Yun, D. Patil, M. A. Horowitz. 2007. A heuristic for optimizing stochastic activity networks with applications to statistical digital circuit sizing. *Optimization and Engineering* **8**(4) 397–430.

Kong, Q., C-Y. Lee, C-P. Teo, Z. Zheng. 2013. Scheduling arrivals to a stochastic delivery system using copositive cones. *Operations Research* **61**(3) 711–726.

Lamberson, L. R., R. R. Hocking. 1970. Optimum time compression in project scheduling. *Management Science* **16**(10) B597–B606.

Li, X., K. Natarajan, C-P. Teo, Z. Zheng. 2014. Distributionally robust mixed integer linear programs: Persistency models with applications. *European Journal of Operational Research* **233**(3) 459–473.

Lindsey, J. H. 1972. An estimate of expected critical-path length in pert networks. *Operations Research* **20**(4) 800–812.

Mak, H-Y., Y. Rong, J. Zhang. 2015. Appointment scheduling with limited distributional information. *Management Science* **61**(2) 316–334.

Meilijson, I., A. Nadas. 1979. Convex majorization with an application to the length of critical paths. *Journal of Applied Probability* **16**(3) 671–677.

Mitchell, G., T. Klastorin. 2007. An effective methodology for the stochastic project compression problem. *IIE Transactions* **39**(10) 957–969.

Möhring, R. H. 2001. Scheduling under uncertainty: Bounding the makespan distribution. *Computational Discrete Mathematics* 79–97.

Natarajan, K., M. Song, C-P. Teo. 2009. Persistency model and its applications in choice modeling. *Management Science* **55**(3) 453–469.

Natarajan, K., C-P. Teo. 2016. On reduced semidefinite programs for second order moment bounds with applications. *To appear in Mathematical Programming* .

Natarajan, K., C-P. Teo, Z. Zheng. 2011. Mixed 0-1 linear programs under objective uncertainty: A completely positive representation. *Operations Research* **59**(3) 713–728.

Plambeck, E. L., B-R. Fu, S. M. Robinson, R. Suri. 1996. Sample-path optimization of convex stochastic performance functions. *Mathematical Programming* **75**(2) 137–176.

Rockafellar, R Tyrrell. 1997. *Convex analysis*. 28, Princeton university press.

Rudin, Walter. 1964. *Principles of mathematical analysis*, vol. 3. McGraw-Hill New York.

Shapiro, A. 2003. Monte carlo sampling methods. *Handbooks in operations research and management science* **10** 353–425.

Toh, K-C., M. J Todd, R. H. Tütüncü. 1999. Sdpt3a matlab software package for semidefinite programming, version 1.3. *Optimization methods and software* **11**(1-4) 545–581.

Tütüncü, R. H., K-C. Toh, M. J. Todd. 2003. Solving semidefinite-quadratic-linear programs using sdpt3. *Mathematical Programming* **95**(2) 189–217.

Van Slyke, R. M. 1963. Letter to the editor - Monte carlo methods and the PERT problem. *Operations Research* **11**(5) 839–860.

Wiesemann, W., D. Kuhn, B. Rustem. 2012. Robust resource allocations in temporal networks. *Mathematical Programming* **135**(1-2) 437–471.