Ambiguous Chance-Constrained Binary Programs under Mean-Covariance Information

Yiling Zhang[1], Ruiwei Jiang[1], Siqian Shen[1]

Abstract

We consider chance-constrained binary programs, where each row of the inequalities that involve uncertainty needs to be satisfied probabilistically. Only the information of the mean and covariance matrix is available, and we solve distributionally robust chance-constrained binary programs (DCBP). Using two different ambiguity sets, we equivalently reformulate the DCBPs as 0-1 second-order cone (SOC) programs. We further exploit the submodularity of 0-1 SOC constraints under special and general covariance matrices, and utilize the submodularity as well as lifting to derive extended polymatroid inequalities to strengthen the 0-1 SOC formulations. We incorporate the valid inequalities in a branch-and-cut algorithm for efficiently solving DCBPs. We demonstrate the computational efficacy and solution performance using diverse instances of a chance-constrained bin packing problem.

Key words: Chance-constrained binary program; distributionally robust optimization; conic integer program; submodularity; extended polymatroid; bin packing

1 Introduction

We consider chance-constrained binary programs that involve a set of individual chance constraints. More specifically, we consider I individual chance constraints and, for each $i \in [I] := \{1, \ldots, I\}$, let $y_i \in \{0,1\}^J$ be a binary decision vector such that $y_i := [y_{i1}, \ldots, y_{iJ}]^\top$ and $\tilde{t}_i$ be the corresponding random coefficients such that $\tilde{t}_i := [\tilde{t}_{i1}, \ldots, \tilde{t}_{iJ}]^\top$. Then, we consider the following individual chance constraints:

$$\mathbb{P}\left\{\tilde{t}_i^\top y_i \le T_i\right\} \;\ge\; 1 - \alpha_i \quad \forall i \in [I], \tag{1}$$

where $T_i \in \mathbb{R}$, $\mathbb{P}$ represents the joint probability distribution of $\{\tilde{t}_{ij} : i \in [I],\ j \in [J]\}$, and each α_i represents an allowed risk tolerance of constraint violation that often takes a small value (e.g.,

[1]Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI, USA; E-mail: {zyiling, ruiwei, siqian}@umich.edu

$\alpha_i = 0.05$).

The individual chance constraints have wide applications in service and operations management, providing an effective and convenient way of controlling capacity violation and ensuring high quality of service. For example, in surgery allocation, y_i represents yes-no decisions of allocating J surgeries in operating room (OR) i, for all $i \in [I]$. The operational time limit of each OR (i.e., T_i) is usually deterministic, but the processing time of each surgery (i.e., $\tilde{t}_{ij}$) is usually random due to the variety of patients, surgical teams, and surgery characteristics. Then, chance constraints (1) make sure that each OR will not go overtime with a large probability, offering an appropriate "end-of-the-day" guarantee.

Chance-constrained programs are difficult to solve, mainly because the feasible region described by constraints (1) is non-convex in general Prékopa (2003). Nonetheless, promising special cases have been identified to recapture the convexity of chance-constrained models. In particular, if $\{\tilde{t}_{ij} : j \in [J]\}$ are assumed to follow a Gaussian distribution with a known mean μ_i and covariance matrix Σ_i, then the chance constraints (1) are equivalent to the second-order cone (SOC) constraints

$$\mu_i^\top y_i + \Phi^{-1}(1-\alpha_i)\sqrt{y_i^\top \Sigma_i y_i} \ \le \ T_i \quad \forall i \in [I], \tag{2}$$

where $\Phi(\cdot)$ represents the cumulative distribution function of the standard Gaussian distribution. In this case, feasible binary solutions y_i to constraints (2) can be quickly found by off-the-shelf optimization solvers. In another promising research stream, the probability distribution $\mathbb{P}$ of $\tilde{t}_{ij}$ is replaced by a finite-sample approximation, leading to a sample average approximation (SAA) of the chance-constrained model Luedtke and Ahmed (2008); Pagnoncelli et al. (2009). The SAA model is then recast as a mixed-integer linear program (MILP), on which many strong valid inequalities can be derived to accelerate the branch-and-cut algorithm (see, e.g., Luedtke et al. (2010); Küçükyavuz (2012); Luedtke (2014); Song et al. (2014); Liu et al. (2016)).

However, a basic challenge of the chance-constrained approach is that the perfect knowledge of probability distribution $\mathbb{P}$ may not be accessible. Under many circumstances, we only have a series of historical data that can be considered as samples taken from the true (while ambiguous) distribution. As a consequence, the solution obtained from a chance-constrained model can be sensitive to the choice of distribution $\mathbb{P}$ we employ in (1) and hence perform poorly in out-of-sample tests. This phenomenon is often observed when solving stochastic programs and is called the *optimizer's curse* Smith and Winkler (2006). A natural way of addressing this curse is that, instead of a single estimate of $\mathbb{P}$, we employ a set of plausible probability distributions, termed the ambiguity set and denoted $\mathcal{D}$. Then, from a robust perspective, we ensure that chance constraints (1) hold valid with regard to all probability distributions belonging to $\mathcal{D}$, i.e.,

$$\inf_{\mathbb{P}\in\mathcal{D}} \mathbb{P}\left\{\tilde{t}_i^\top y_i \leq T_i\right\} \geq 1-\alpha_i \quad \forall i \in [I], \tag{3}$$

and accordingly we call (3) *distributionally robust chance constraints* (DRCCs).

In this paper, we consider distributionally robust chance-constrained binary programs (DCBP) that involve binary y_i-variables and DRCCs (3). Without making the Gaussian assumption on $\mathbb{P}$, we show that a DCBP is equivalent to a 0-1 SOC program when $\mathcal{D}$ is characterized by the first two moments of $\tilde{t}_i$. Furthermore, building upon existing work on valid inequalities for submodular/supermodular functions, we exploit the submodularity of the 0-1 SOC program to derive valid inequalities. As demonstrated in extensive computational experiments of bin packing instances, these valid inequalities significantly accelerate the branch-and-cut algorithm for solving the related DCBPs. Notably, the proposed submodular approximations and the resulting valid inequalities apply to general 0-1 SOC programs than DCBPs.

The remainder of the paper is organized as follows. Section 2 reviews the prior work related to optimization techniques used in this paper and stochastic bin packing problems. Section 3 presents two 0-1 SOC representations, respectively, for DRCCs under two moment-based ambiguity sets. Section 4 utilizes submodularity and lifting to derive valid inequalities to strengthen the 0-1 SOC formulations. Section 5 demonstrates the computational efficacy of our approaches for solving different 0-1 SOC reformulations of a DCBP for chance-constrained bin packing, with diverse problem sizes and parameter settings. Section 6 summarizes the paper and discusses future research directions.

2 Prior Work

Chance-constrained binary programs with uncertain technology matrix and/or right-hand side are computationally challenging, largely because (i) the non-convexity of chance constraints and (ii) the discrete variables. The majority of existing literature requires full distributional knowledge of the random coefficients and applies the SAA approach to approximate the models as MILPs. For example, Song et al. (2014) considered a generic chance-constrained binary packing problem using finite samples of the random item weights, and derived lifted cover inequalities to accelerate the computation. For generic chance-constrained programs, the SAA approach and valid inequalities for the related MILPs have been well studied in the literature (see, e.g., Luedtke and Ahmed (2008); Luedtke et al. (2010); Küçükyavuz (2012); Luedtke (2014)). We also refer to Shylo et al. (2012); Deng et al. (2016); Shen and Wang (2014) for wide applications of chance-constrained binary programs, mainly in service systems and operations. As compared to the existing work, this paper waives the assumption of full distributional information and only relies on the first two moments of the uncertainty.

Distributionally robust optimization has received growing attention, mainly because it provides effective modeling and computational approaches for handling ambiguous distributions of random variables in stochastic programming by using available distributional information. Moment information has been widely used for building ambiguity sets in various distributionally robust optimization models (see, e.g., Bertsimas et al. (2010); Delage and Ye (2010); Wiesemann et al. (2014)). Using moment-based ambiguity sets, El Ghaoui et al. (2003); Calafiore and El Ghaoui (2006); Wagner (2008); Chen et al. (2010); Zymler et al. (2013); Cheng et al. (2014); Jiang and Guan (2016) derived exact reformulations and/or approximations for DRCCs, often in the form of SDPs. In special cases, e.g., when the first two moments are *exactly* matched in the ambiguity set, the SDPs can further be simplified as SOC programs. While many existing ambiguity sets exactly match the first two moments of uncertainty (see, e.g., El Ghaoui et al. (2003); Wagner (2008); Zymler et al. (2013)), Delage and Ye (2010) proposed a data-driven approach to construct an ambiguity set that can model moment estimation errors. In this paper, we consider both types of moment-based ambiguity sets. To the best of our knowledge, for the first time, we provide an SOC representation of DRCCs using the general ambiguity set proposed by Delage and Ye (2010).

Meanwhile, distributionally robust optimization has received much less attention in discrete optimization problems, possibly due to the difficulty of solving 0-1 nonlinear programs. For example, most off-the-shelf solvers cannot directly handle 0-1 SDPs, which often arise from discrete optimization problems with DRCCs. To the best of our knowledge, our results on chance constraints are most related to Cheng et al. (2014) that studied DRCCs in the binary knapsack problem and derived 0-1 SDP reformulations. As compared to Cheng et al. (2014), we investigate a different ambiguity set and derive a 0-1 *SOC* representation. Additionally, we solve the 0-1 SOC reformulation to *global* optimality instead of considering an SDP relaxation as in Cheng et al. (2014).

In the seminal work Nemhauser et al. (1978), the authors identified submodularity in combinatorial and discrete optimization problems and proved a sufficient and necessary condition for 0-1 quadratic functions being submodular. We use this condition to exploit the submodularity of our 0-1 SOC reformulations. Indeed, submodular and supermodular knapsack sets (the discrete lower level set of a submodular function and discrete upper level set of a supermodular function, respectively) often arise when modeling utility, risk, and chance constraints on discrete variables. Extended polymatroid inequalities (see Edmonds (1970); Nemhauser and Wolsey (1999)) can be efficiently obtained through greedy-based separation procedures to optimize submodular/supermodular functions. Recently, Atamtürk and Narayanan (2009); Atamtürk and Bhardwaj (2015); Bhardwaj (2015) proposed cover and packing inequalities for efficiently solving submodular and supermodular knapsack sets with 0-1 variables. Our results on valid inequalities are most related to Atamtürk and Bhardwaj (2018); Bhardwaj (2015) that identified a sufficient condition for the submodularity of 0-1 SOC constraints and strengthened their formulations by using the corresponding extended polymatroid inequalities (see Section 2.2 of Atamtürk and Bhardwaj (2018)). In contrast, we derive a different way to exploit the submodularity of general 0-1 SOC constraints. In particular, we apply the sufficient and necessary condition derived by Nemhauser et al. (1978) to search for "optimal" submodular approximations of the 0-1 SOC constraints (see Section 4.1).

The main contributions of the paper are three-fold. First, using the general moment-based ambiguity set proposed by Delage and Ye (2010), we equivalently reformulate DRCCs as 0-1 SOC constraints that can readily be solved by solvers. Second, we exploit the (hidden) submodularity of the 0-1 SOC constraints and derive extended polymatroid valid inequalities to accelerate solving DCBP. In particular, we provide an efficient way of finding "optimal" submodular approximations of the 0-1 SOC constraints in the original variable space, and furthermore show that any 0-1 SOC constraint possesses submodularity in a lifted space. The valid inequalities in original and lifted spaces can both be efficiently separated via the well-known greedy algorithm (see, e.g., Atamtürk and Gómez (2017); Atamtürk and Narayanan (2008); Edmonds (1970)). Third, we conduct extensive numerical studies to demonstrate the computational efficacy of our solution approaches.

3 DCBP Models and Reformulations

We study DRCCs under two alternatives of ambiguity set $\mathcal{D}$ based on the first two moments of $\tilde{t}_i,\ i \in [I]$. The first ambiguity set, denoted $\mathcal{D}_1$, exactly matches the mean and covariance matrix of each $\tilde{t}_i$. In contrast, the second ambiguity set, denoted $\mathcal{D}_2$, considers the estimation errors of sample mean and sample covariance matrix (see Delage and Ye (2010)). In Section 3.1, we introduce these two ambiguity sets and their calibration based on historical data. In Section 3.2, we derive SOC representations of DRCC (3) under $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively. While the former case (i.e., (3) under $\mathcal{D}_1$) has been well studied (see, e.g., El Ghaoui et al. (2003); Calafiore and El Ghaoui (2006); Zymler et al. (2013)), to the best of our knowledge, this is the first work to show the SOC representation of the latter case based on general covariance matrices (i.e., (3) under $\mathcal{D}_2$).

3.1 Ambiguity Sets

Suppose that a series of independent historical data samples $\{\tilde{t}_i^n\}_{n=1}^N$ are drawn from the true probability distribution $\mathbb{P}$ of $\tilde{t}_{ij}$. Then, the first two moments of $\tilde{t}_i$ can be estimated by the sample mean and sample covariance matrix

$$\mu_i \;=\; \frac{1}{N}\sum_{n=1}^{N}\tilde{t}_i^n, \qquad \Sigma_i \;=\; \frac{1}{N}\sum_{n=1}^{N}(\tilde{t}_i^n-\mu_i)(\tilde{t}_i^n-\mu_i)^\top.$$

Throughout this paper, we assume that both Σ_i and the true covariance matrix of $\tilde{t}_i$ are symmetric and positive definite. As promised by the law of large numbers, as the data size N grows, μ_i and Σ_i converge to the true mean and true covariance matrix of $\tilde{t}_i$, respectively. Hence, when N takes a large value, a natural choice of the ambiguity set consists of all probability distributions that match the sample moments μ_i and Σ_i, i.e.,

$$\mathcal{D}_1 \;=\; \left\{ \mathbb{P} \in \mathcal{P}(\mathbb{R}^J) : \quad \begin{array}{l} \mathbb{E}_{\mathbb{P}}[\tilde{t}_i] = \mu_i, \\ \\ \mathbb{E}_{\mathbb{P}}[(\tilde{t}_i-\mu_i)(\tilde{t}_i-\mu_i)^\top] = \Sigma_i,\ \forall i \in [I] \end{array} \right\},$$

where $\mathcal{P}(\mathbb{R}^J)$ represents the set of all probability distributions on $\mathbb{R}^J$.

Under many circumstances, however, the historical data can be inadequate. With a small N, there may exist considerable estimation errors in μ_i and Σ_i, which brings a layer of "moment ambiguity" into $\mathcal{D}_1$ and adds to the existing distributional ambiguity of $\mathbb{P}$. To address the moment ambiguity and take into account the estimation errors, Delage and Ye (2010) proposed an alternative ambiguity set

$$\mathcal{D}_2 \;=\; \left\{ \mathbb{P} \in \mathcal{P}(\mathbb{R}^J) : \begin{array}{rcl} (\mathbb{E}_{\mathbb{P}}[\tilde{t}_i] - \mu_i)^\top \Sigma_i^{-1} (\mathbb{E}_{\mathbb{P}}[\tilde{t}_i] - \mu_i) & \leq & \gamma_1, \\ \mathbb{E}_{\mathbb{P}}\left[(\tilde{t}_i - \mu_i)(\tilde{t}_i - \mu_i)^\top\right] & \preceq & \gamma_2 \Sigma_i, \quad \forall i \in [I] \end{array} \right\},$$

where $\gamma_1 > 0$ and $\gamma_2 > \max\{\gamma_1, 1\}$ represent two given parameters. Set $\mathcal{D}_2$ designates that (i) the true mean of $\tilde{t}_i$ is within an ellipsoid centered at μ_i, and (ii) the true covariance matrix of $\tilde{t}_i$ is bounded from above by $\gamma_2\Sigma_i - (\mathbb{E}_{\mathbb{P}}[\tilde{t}_i] - \mu_i)(\mathbb{E}_{\mathbb{P}}[\tilde{t}_i] - \mu_i)^\top$ (note that $\mathbb{E}_{\mathbb{P}}[(\tilde{t}_i - \mu_i)(\tilde{t}_i - \mu_i)^\top] = \mathbb{E}_{\mathbb{P}}[(\tilde{t}_i - \mathbb{E}_{\mathbb{P}}[\tilde{t}_i])(\tilde{t}_i - \mathbb{E}_{\mathbb{P}}[\tilde{t}_i])^\top] + (\mathbb{E}_{\mathbb{P}}[\tilde{t}_i] - \mu_i)(\mathbb{E}_{\mathbb{P}}[\tilde{t}_i] - \mu_i)^\top$).

Delage and Ye (2010) offered a rigorous guideline for selecting γ_1 and γ_2 values (see Theorem 2 in Delage and Ye (2010)) so that $\mathcal{D}_2$ includes the true distribution of $\tilde{t}_i$ with a high confidence level. In practice, we can select the values of γ_1 and γ_2 via cross validation. For example, we can divide the N data points into two halves. We estimate sample moments (μ_i^1, Σ_i^1) based on the first half of the data and (μ_i^2, Σ_i^2) based on the second half. Then, we characterize $\mathcal{D}_2$ based on (μ_i^1, Σ_i^1), and select γ_1 and γ_2 such that probability distributions with moments (μ_i^2, Σ_i^2) belong to $\mathcal{D}_2$.

3.2 SOC Representations of the DRCC

Now we derive SOC representations of DRCC (3) for all $i \in [I]$. For notation brevity, we define vector $y := [y_{i1}, \ldots, y_{iJ}]$ and omit the subscript i throughout this section. First, we review the celebrated SOC representation of DRCC (3) under ambiguity set $\mathcal{D}_1$ in the following theorem.

Theorem 1. (Adapted from El Ghaoui et al. (2003), also see Wagner (2008)) The DRCC (3) with $\mathcal{D} = \mathcal{D}_1$ is equivalent to the following SOC constraint:

$$\mu^\top y + \sqrt{\frac{1-\alpha}{\alpha}} \sqrt{y^\top \Sigma y} \;\leq\; T. \tag{4}$$

Theorem 1 shows that we can recapture the convexity of DRCC (3) by employing ambiguity set $\mathcal{D}_1$ to model the $\tilde{t}$ uncertainty. Perhaps more surprisingly, in this case, the convex feasible region characterized by DRCC (3) is SOC representable. It follows that the continuous relaxation of the DCBP model is an SOC program, which can be solved very efficiently by standard nonlinear optimization solvers.

Next, we show that DRCC (3) under the ambiguity set $\mathcal{D}_2$ is also SOC representable. This implies that the computational complexity of the DCBP remains the same even if we take the moment ambiguity into account. We present the main result of this section in the following theorem.

Theorem 2. DRCC (3) with $\mathcal{D} = \mathcal{D}_2$ is equivalent to

$$\mu^\top y + \left(\sqrt{\gamma_1} + \sqrt{\left(\frac{1-\alpha}{\alpha}\right)(\gamma_2 - \gamma_1)}\right)\sqrt{y^\top \Sigma y} \ \le \ T \tag{5a}$$

if $\gamma_1/\gamma_2 \le \alpha$, and is equivalent to

$$\mu^\top y + \sqrt{\frac{\gamma_2}{\alpha}}\sqrt{y^\top \Sigma y} \ \le \ T \tag{5b}$$

if $\gamma_1/\gamma_2 > \alpha$.

Rujeerapaiboon et al. (2015) considered an ambiguity set similar to $\mathcal{D}_2$ and derived an SOC representation of DRCCs in portfolio optimization under an assumption of weak sense white noise, i.e., the uncertainty is stationary and mutually uncorrelated over time (see Definition 4 and Theorem 5 in Rujeerapaiboon et al. (2015)). In contrast, the SOC representation in Theorem 2 holds for general covariance matrices. We prove Theorem 2 in two steps. In the first step, we project the random vector $\tilde{t}$ and its ambiguity set $\mathcal{D}_2$ from $\mathbb{R}^J$ to the real line, i.e., $\mathbb{R}$. This simplifies DRCC (3) as involving a one-dimensional random variable. In the second step, we derive optimal (i.e., worst-case) mean and covariance matrix in $\mathcal{D}_2$ that attain the worst-case probability bound in (3). We then apply Cantelli's inequality to finish the representation. We present the first step of the proof in the following lemma.

Lemma 1. Let $\tilde{s}$ be a random vector in $\mathbb{R}^J$ and $\tilde{\xi}$ be a random variable in $\mathbb{R}$. For a given $y \in \mathbb{R}^J$, define ambiguity sets $\mathcal{D}_{\tilde{s}}$ and $\mathcal{D}_{\tilde{\xi}}$ as

$$\mathcal{D}_{\tilde{s}} \ = \ \left\{\mathbb{P} \in \mathcal{P}(\mathbb{R}^J) : \ \mathbb{E}_{\mathbb{P}}[\tilde{s}]^\top \Sigma^{-1} \mathbb{E}_{\mathbb{P}}[\tilde{s}] \le \gamma_1, \quad \mathbb{E}_{\mathbb{P}}[\tilde{s}\tilde{s}^\top] \preceq \gamma_2 \Sigma\right\} \tag{6a}$$

and

$$\mathcal{D}_{\tilde{\xi}} \ = \ \left\{\mathbb{P} \in \mathcal{P}(\mathbb{R}) : \ |\mathbb{E}_{\mathbb{P}}[\tilde{\xi}]| \le \sqrt{\gamma_1}\sqrt{y^\top \Sigma y}, \quad \mathbb{E}_{\mathbb{P}}[\tilde{\xi}^2] \le \gamma_2 (y^\top \Sigma y)\right\}. \tag{6b}$$

Then, for any Borel measurable function $f : \mathbb{R}^J \to \mathbb{R}$, we have

$$\inf_{\mathbb{P} \in \mathcal{D}_{\tilde{s}}} \mathbb{P}\{f(y^\top \tilde{s}) \le 0\} \ = \ \inf_{\mathbb{P} \in \mathcal{D}_{\tilde{\xi}}} \mathbb{P}\{f(\tilde{\xi}) \le 0\}.$$

Proof. We first show that $\inf_{\mathbb{P}\in\mathcal{D}_{\tilde{s}}} \mathbb{P}\{f(y^\top \tilde{s}) \le 0\} \ge \inf_{\mathbb{P}\in\mathcal{D}_{\tilde{\xi}}} \mathbb{P}\{f(\tilde{\xi}) \le 0\}$. Pick a $\mathbb{P} \in \mathcal{D}_{\tilde{s}}$, and let $\tilde{s}$ denote the corresponding random vector and $\tilde{\xi} = y^\top \tilde{s}$. It follows that

$$
\begin{aligned}
\mathbb{E}_{\mathbb{P}}[\tilde{\xi}] &= y^\top \mathbb{E}_{\mathbb{P}}[\tilde{s}] \\
&\le \max_{s:\ s^\top \Sigma^{-1} s\ \le\ \gamma_1} \ y^\top s \qquad (7a) \\
&= \max_{z:\ \|z\|_2\ \le \sqrt{\gamma_1}} (\Sigma^{1/2} y)^\top z \ = \ \sqrt{\gamma_1}\sqrt{y^\top \Sigma y},
\end{aligned}
$$

where inequality (7a) is because $\mathbb{E}_{\mathbb{P}}[\tilde{s}]^\top \Sigma^{-1} \mathbb{E}_{\mathbb{P}}[\tilde{s}] \le \gamma_1$. Similarly, we have $\mathbb{E}_{\mathbb{P}}[\tilde{\xi}] \ge -\sqrt{\gamma_1}\sqrt{y^\top \Sigma y}$. Meanwhile, note that

$$
\begin{aligned}
\mathbb{E}_{\mathbb{P}}[\tilde{\xi}^2] &= y^\top \mathbb{E}_{\mathbb{P}}[\tilde{s}\tilde{s}^\top] y \\
&\le y^\top (\gamma_2 \Sigma) y \ = \ \gamma_2 (y^\top \Sigma y), \qquad (7b)
\end{aligned}
$$

where inequality (7b) is because $\mathbb{E}_{\mathbb{P}}[\tilde{s}\tilde{s}^\top] \preceq \gamma_2 \Sigma$. Hence, the probability distribution of $\tilde{\xi}$ belongs to $\mathcal{D}_{\tilde{\xi}}$. It follows that $\inf_{\mathbb{P}\in\mathcal{D}_{\tilde{s}}} \mathbb{P}\{f(y^\top \tilde{s}) \le 0\} \ge \inf_{\mathbb{P}\in\mathcal{D}_{\tilde{\xi}}} \mathbb{P}\{f(\tilde{\xi}) \le 0\}$.

Second, we show that $\inf_{\mathbb{P}\in\mathcal{D}_{\tilde{s}}} \mathbb{P}\{f(y^\top \tilde{s}) \le 0\} \le \inf_{\mathbb{P}\in\mathcal{D}_{\tilde{\xi}}} \mathbb{P}\{f(\tilde{\xi}) \le 0\}$. Pick a $\mathbb{P} \in \mathcal{D}_{\tilde{\xi}}$, and let $\tilde{\xi}$ denote the corresponding random variable and $\tilde{s} = \big[\tilde{\xi}/(y^\top \Sigma y)\big] \Sigma y$. It follows that

$$
\begin{aligned}
\mathbb{E}_{\mathbb{P}}[\tilde{s}]^\top \Sigma^{-1} \mathbb{E}_{\mathbb{P}}[\tilde{s}] &= \mathbb{E}_{\mathbb{P}}[\tilde{\xi}]^2 \frac{y^\top \Sigma}{y^\top \Sigma y} \Sigma^{-1} \frac{\Sigma y}{y^\top \Sigma y} \ = \ \frac{\mathbb{E}_{\mathbb{P}}[\tilde{\xi}]^2}{y^\top \Sigma y} \\
&\le \frac{\gamma_1 y^\top \Sigma y}{y^\top \Sigma y} \ = \ \gamma_1, \qquad (7c)
\end{aligned}
$$

where inequality (7c) is because $|\mathbb{E}_{\mathbb{P}}[\tilde{\xi}]| \le \sqrt{\gamma_1}\sqrt{y^\top \Sigma y}$. Meanwhile, note that

$$
\begin{aligned}
\mathbb{E}_{\mathbb{P}}[\tilde{s}\tilde{s}^\top] &= \mathbb{E}_{\mathbb{P}}\left[\tilde{\xi}^2 \frac{\Sigma y}{y^\top \Sigma y} \frac{y^\top \Sigma}{y^\top \Sigma y}\right] \\
&= \mathbb{E}_{\mathbb{P}}[\tilde{\xi}^2] \frac{(\Sigma y)(\Sigma y)^\top}{(y^\top \Sigma y)^2} \\
&\preceq \gamma_2 (y^\top \Sigma y) \frac{(\Sigma y)(\Sigma y)^\top}{(y^\top \Sigma y)^2} \qquad (7d) \\
&\preceq \gamma_2 (y^\top \Sigma y) \frac{(y^\top \Sigma y)\Sigma}{(y^\top \Sigma y)^2} \ = \ \gamma_2 \Sigma, \qquad (7e)
\end{aligned}
$$

where inequality (7d) is because $\mathbb{E}_{\mathbb{P}}[\tilde{\xi}^2] \le \gamma_2 (y^\top \Sigma y)$ and inequality (7e) is because $(\Sigma y)(\Sigma y)^\top \preceq (y^\top \Sigma y)\Sigma$, which holds because for all $z \in \mathbb{R}^J$,

$$
z^\top (\Sigma y)(\Sigma y)^\top z \ = \ \Big[(\Sigma^{1/2} z)^\top (\Sigma^{1/2} y)\Big]^2
$$

$$\leq ||\Sigma^{1/2}z||^2 \; ||\Sigma^{1/2}y||^2 \qquad \text{(Cauchy-Schwarz inequality)}$$
$$= (y^\top \Sigma y)(z^\top \Sigma z)$$
$$= z^\top [(y^\top \Sigma y)\Sigma] z.$$

Hence, the probability distribution of $\tilde{s}$ belongs to $\mathcal{D}_{\tilde{s}}$. It follows that $\inf_{\mathbb{P}\in\mathcal{D}_{\tilde{s}}} \mathbb{P}\{\, f(y^\top \tilde{s}) \leq 0\} \leq \inf_{\mathbb{P}\in\mathcal{D}_{\tilde{\xi}}} \mathbb{P}\{f(\tilde{\xi}) \leq 0\}$ because $\tilde{\xi} = y^\top \tilde{s}$, and the proof is completed. □

Remark 1. Popescu (2007) and Yu et al. (2009) showed a similar projection property for $\mathcal{D}_1$, i.e., when the first two moments of $\tilde{s}$ are exactly known. Lemma 1 employs a different transformation approach to show the projection property for $\mathcal{D}_2$ when these moments are ambiguous in the sense of Delage and Ye (2010).

We are now ready to finish the proof of Theorem 2.

Proof of Theorem 2. First, we define random vector $\tilde{s} = \tilde{t} - \mu$, random variable $\tilde{\xi} = y^\top \tilde{s}$, constant $b = T - \mu^\top y$, and set S such that

$$S \;=\; \{(\mu_1, \sigma_1) \in \mathbb{R} \times \mathbb{R}_+ : \; |\mu_1| \leq \sqrt{\gamma_1}\sqrt{y^\top \Sigma y}, \; \mu_1^2 + \sigma_1^2 \leq \gamma_2 y^\top \Sigma y\}.$$

It follows that

$$\begin{aligned} \inf_{\mathbb{P}\in\mathcal{D}_2} \mathbb{P}\{\tilde{t}^\top y \leq T\} \;&=\; \inf_{\mathbb{P}\in\mathcal{D}_{\tilde{s}}} \mathbb{P}\{y^\top \tilde{s} \leq b\} \\ &= \inf_{\mathbb{P}\in\mathcal{D}_{\tilde{\xi}}} \mathbb{P}\{\tilde{\xi} \leq b\} \qquad (8a) \\ &= \inf_{(\mu_1,\sigma_1)\in S} \; \inf_{\mathbb{P}\in\mathcal{D}_1(\mu_1,\sigma_1^2)} \mathbb{P}\{\tilde{\xi} \leq b\}, \qquad (8b) \end{aligned}$$

where $\mathcal{D}_{\tilde{s}}$ and $\mathcal{D}_{\tilde{\xi}}$ are defined in (6a) and (6b), respectively, equality (8a) follows from Lemma 1, and equality (8b) decomposes the optimization problem in (8a) into two layers: the outer layer searches for the optimal (i.e., worst-case) mean and covariance, while the inner layer computes the worst-case probability bound under the given mean and covariance. For the inner layer, based on Cantelli's inequality, we have

$$\inf_{\mathbb{P}\in\mathcal{D}_1(\mu_1,\sigma_1^2)} \mathbb{P}\{\tilde{\xi} \leq b\} = \begin{cases} \frac{(b-\mu_1)^2}{\sigma_1^2 + (b-\mu_1)^2}, & \text{if } b \geq \mu_1, \\ 0, & \text{o.w.} \end{cases}$$

As DRCC (3) states that $\inf_{\mathbb{P}\in\mathcal{D}_2}\mathbb{P}\{\tilde{t}^\top y \le T\} \ge 1-\alpha > 0$, we can assume $b \ge \mu_1$ for all $(\mu_1, \sigma_1) \in S$ without loss of generality. That is,

$$b \;\ge\; \max_{(\mu_1,\sigma_1)\in S} \mu_1 \;=\; \sqrt{\gamma_1}\sqrt{y^\top \Sigma y}.$$

It follows that

$$\begin{aligned}
\inf_{\mathbb{P}\in\mathcal{D}_2} \mathbb{P}\{\tilde{t}^\top y \le T\} &= \inf_{(\mu_1,\sigma_1)\in S} \frac{(b-\mu_1)^2}{\sigma_1^2 + (b-\mu_1)^2} \\
&= \inf_{(\mu_1,\sigma_1)\in S} \frac{1}{\left(\frac{\sigma_1}{b-\mu_1}\right)^2 + 1}.
\end{aligned} \tag{8c}$$

Note that the objective function value in (8c) decreases as $\sigma_1/(b-\mu_1)$ increases. Hence, (8c) shares optimal solutions with the following optimization problem:

$$\inf_{(\mu_1,\sigma_1)\in S} -\left(\frac{\sigma_1}{b-\mu_1}\right). \tag{8d}$$

The feasible region of problem (8d) is depicted in the shaded area of Figure 1. Furthermore, we

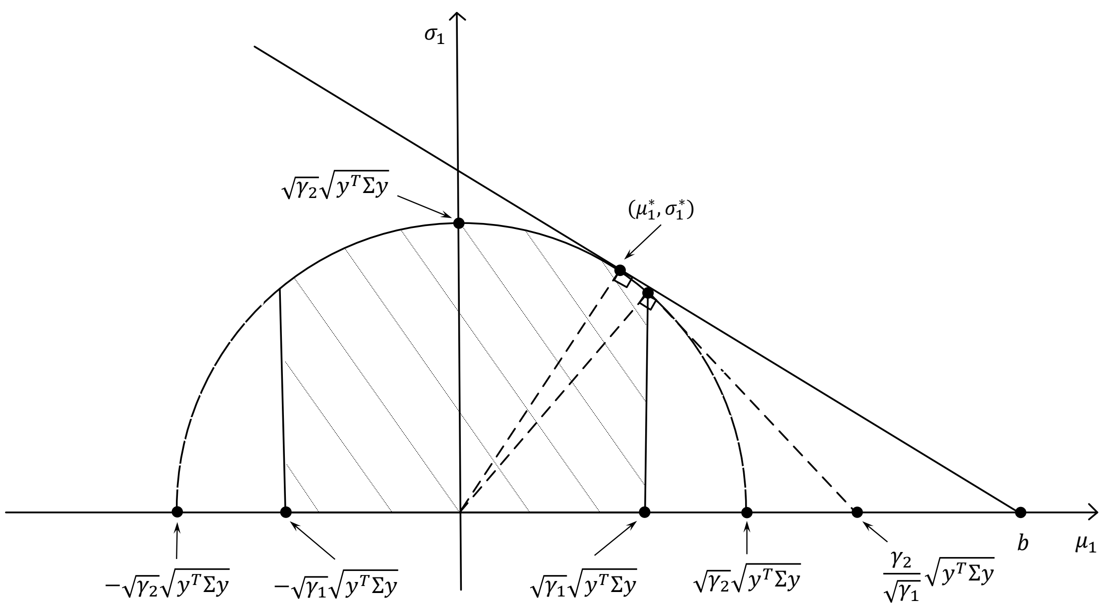

Figure 1: Graphical Solution of Problem (8d)

note that the objective function of (8d) equals to the slope of the straight line connecting points $(b, 0)$ and (μ_1, σ_1) (see Figure 1 for an example). It follows that an optimal solution (μ_1^*, σ_1^*) to problem (8d), and so to problem (8c), lies in one of the following two cases:

Case 1. If $\sqrt{\gamma_1}\sqrt{y^\top \Sigma y} \le b \le (\gamma_2/\sqrt{\gamma_1})\sqrt{y^\top \Sigma y}$, then $\mu_1^* = \sqrt{\gamma_1}\sqrt{y^\top \Sigma y}$ and $\sigma_1^* = \sqrt{\gamma_2 - \gamma_1}\sqrt{y^\top \Sigma y}$.

Case 2. If $b > (\gamma_2/\sqrt{\gamma_1})\sqrt{y^\top \Sigma y}$, then $\mu_1^* = (\gamma_2 y^\top \Sigma y)/b$ and
$\sigma_1^* = \sqrt{\gamma_2 y^\top \Sigma y - (\gamma_2 y^\top \Sigma y)^2/b^2}$.

Denoting $\kappa(b, y) = \frac{b}{\sqrt{y^\top \Sigma y}}$, we have

$$\inf_{\mathbb{P}\in\mathcal{D}_2} \mathbb{P}\{\tilde{t}^\top y \le T\} \;=\; \begin{cases} \dfrac{1}{\left(\frac{\sqrt{\gamma_2-\gamma_1}}{\kappa(b,y)-\sqrt{\gamma_1}}\right)^2+1}, & \text{if } \sqrt{\gamma_1} \le \kappa(b,y) \le \frac{\gamma_2}{\sqrt{\gamma_1}}, \\[2ex] \frac{\kappa(b,y)^2-\gamma_2}{\kappa(b,y)^2}, & \text{if } \kappa(b,y) > \frac{\gamma_2}{\sqrt{\gamma_1}}. \end{cases} \tag{8e}$$

Second, based on (8e), the DRCC $\inf_{\mathbb{P}\in\mathcal{D}_2} \mathbb{P}\{\tilde{t}^\top y \le T\} \ge 1-\alpha$ has the following representations:

$$\text{DRCC} \;\Leftrightarrow\; \begin{cases} \frac{b}{\sqrt{y^\top \Sigma y}} \;\ge\; \sqrt{\gamma_1} + \sqrt{\left(\frac{1-\alpha}{\alpha}\right)(\gamma_2-\gamma_1)}, & \text{if } \sqrt{\gamma_1} \le \frac{b}{\sqrt{y^\top \Sigma y}} \le \frac{\gamma_2}{\sqrt{\gamma_1}}, \\ \frac{b}{\sqrt{y^\top \Sigma y}} \;\ge\; \sqrt{\frac{\gamma_2}{\alpha}}, & \text{if } \frac{b}{\sqrt{y^\top \Sigma y}} > \frac{\gamma_2}{\sqrt{\gamma_1}}. \end{cases}$$

It follows that DRCC (3) is equivalent to an SOC constraint by discussing the following two cases:

Case 1. If $\gamma_1/\gamma_2 \le \alpha$, then $\gamma_2/\sqrt{\gamma_1} \ge \sqrt{\gamma_1} + \sqrt{[(1-\alpha)/\alpha](\gamma_2-\gamma_1)}$ and $\gamma_2/\sqrt{\gamma_1} \ge \sqrt{\gamma_2/\alpha}$. It follows that (i) if $\sqrt{\gamma_1} \le b/\sqrt{y^\top \Sigma y} \le \gamma_2/\sqrt{\gamma_1}$, then DRCC is equivalent to $b/\sqrt{y^\top \Sigma y} \ge \sqrt{\gamma_1} + \sqrt{[(1-\alpha)/\alpha](\gamma_2-\gamma_1)}$ and (ii) if $b/\sqrt{y^\top \Sigma y} > \gamma_2/\sqrt{\gamma_1}$, then DRCC always holds. Combining sub-cases (i) and (ii) yields that DRCC (3) is equivalent to $b/\sqrt{y^\top \Sigma y} \ge \sqrt{\gamma_1} + \sqrt{[(1-\alpha)/\alpha](\gamma_2-\gamma_1)}$.

Case 2. If $\gamma_1/\gamma_2 > \alpha$, then $\gamma_2/\sqrt{\gamma_1} < \sqrt{\gamma_1} + \sqrt{[(1-\alpha)/\alpha](\gamma_2-\gamma_1)}$ and $\gamma_2/\sqrt{\gamma_1} < \sqrt{\gamma_2/\alpha}$. It follows that (i) if $\sqrt{\gamma_1} \le b/\sqrt{y^\top \Sigma y} \le \gamma_2/\sqrt{\gamma_1}$, then DRCC always fails and (ii) if $b/\sqrt{y^\top \Sigma y} > \gamma_2/\sqrt{\gamma_1}$, then DRCC is equivalent to $b/\sqrt{y^\top \Sigma y} \ge \sqrt{\gamma_2/\alpha}$. Combining sub-cases (i) and (ii) yields that DRCC (3) is equivalent to $b/\sqrt{y^\top \Sigma y} \ge \sqrt{\gamma_2/\alpha}$.

The proofs of the above two cases are both completed by the definition of b. □

To sum up, we have two exact 0-1 SOC constraint reformulations of DRCC (3) under ambiguity sets $\mathcal{D}_1$ and $\mathcal{D}_2$, being constraints (4), (5a)/(5b), respectively.

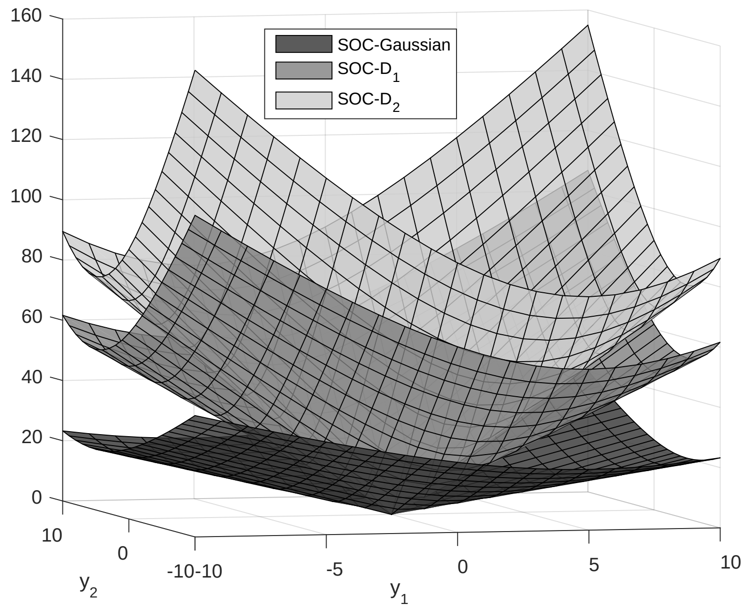

Figure 2: Illustration of the Three SOC Reformulations (2), (4), and (5b) of DRCC (3)

Example 1. We consider a DRCC (3) with $I = 1$ (the subscript i is hence omitted), $J = 2$, $1 - \alpha = 95\%$, mean vector $\mu = [0\ 0]^\top$ and covariance matrix $\Sigma = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$. For the ambiguity set $\mathcal{D}_2$, we set $\gamma_1 = 1$ and $\gamma_2 = 2$. We note that $\Phi^{-1}(1 - \alpha) = 1.6449$ in the SOC reformulation (2), $\sqrt{(1 - \alpha)/\alpha} = 4.3589$ in the SOC reformulation (4), and $\sqrt{\gamma_2/\alpha} = 6.3246$ in the SOC reformulation (5b) (since $\gamma_1/\gamma_2 > \alpha$). Without specifying the value of T, we depict the boundaries of the second-order cones associated with the three SOC reformulations in Figure 2. From this figure, we observe that the Gaussian approximation leads to the widest cone and so the largest SOC feasible region corresponding to DRCC (3).

4 Valid Inequalities for DCBP

Although 0-1 SOC constraint reformulations can be directly handled by the off-the-shelf solvers, as we report in Section 5, the resultant 0-1 SOC programs are often time-consuming to solve, primarily because of the binary restrictions on variables. In this section, we derive valid inequalities for DRCC (3), with the objective of accelerating the branch-and-cut algorithm for solving DCBP with individual DRCCs and also general 0-1 SOC programs in commercial solvers. Specifically, we exploit the submodularity of 0-1 SOC constraints. In Section 4.1, we derive a sufficient condition for submodularity and two approximations of the 0-1 SOC constraints that satisfy this condition. Using the submodular approximations, we derive extended polymatroid inequalities. In Section 4.2, we show the submodularity of the 0-1 SOC constraints in a lifted (i.e., higher-dimensional) space. While the extended polymatroid inequalities are well-known (see, e.g., Atamtürk and Gómez (2017); Atamtürk and Narayanan (2008); Edmonds (1970)), to the best of our knowledge, the two submodular approximations and the submodularity of 0-1 SOC constraints in the lifted space appear to be new and have not been studied in any literature.

4.1 Submodularity of the 0-1 SOC Constraints

We consider SOC constraints of the form

$$\mu^\top y + \sqrt{y^\top \Lambda y} \ \leq \ T, \tag{9}$$

where Λ represents a $J \times J$ positive definite matrix. Note that all SOC reformulations (4), (5a), and (5b) derived in Section 3, as well as the SOC reformulation (2) of a chance constraint with Gaussian uncertainty, possess the form of (9). Before investigating the submodularity of (9), we review the definitions of submodular functions and extended polymatroids.

Definition 1. Define the collection of set $[J]$'s subsets $\mathcal{C} := \{R : R \subseteq [J]\}$. A set function h: $\mathcal{C} \to \mathbb{R}$ is called submodular if and only if $h(R \cup \{j\}) - h(R) \geq h(S \cup \{j\}) - h(S)$ for all $R \subseteq S \subseteq [J]$ and all $j \in [J] \setminus S$.

Throughout this section, we refer to a set function as $h(R)$ and $h(y)$ interchangeably, where $y \in \{0,1\}^J$ represents the indicating vector for subset $R \subseteq [J]$, i.e., $y_j = 1$ if $j \in R$ and $y_j = 0$ otherwise.

Definition 2. For a submodular function $h(y)$, the polyhedron

$$\mathrm{EP}_h = \{\pi \in \mathbb{R}^J : \ \pi(R) \leq h(R), \quad \forall R \subseteq [J]\}$$

is called an extended polymatroid associated with $h(y)$, where $\pi(R) = \sum_{j \in R} \pi_j$.

For a submodular function $h(y)$ with $h(\emptyset) = 0$, inequality

$$\pi^\top y \leq t, \tag{10}$$

termed an *extended polymatroid inequality*, is valid for the epigraph of h, i.e.,

$$\{(y,t) \in \{0,1\}^J \times \mathbb{R} : t \geq h(y)\}, \text{ if and only if } \pi \in \mathrm{EP}_h \text{ (see Edmonds (1970))} .$$

Furthermore, the separation of (10) can be efficiently done by a greedy algorithm Edmonds (1970), which we briefly describe in Algorithm 1.

Algorithm 1 A greedy algorithm for separating extended polymatroid inequalities (10).

Input: A point $(\hat{y}, \hat{t})$ with $\hat{y} \in [0,1]^J$, $\hat{t} \in \mathbb{R}$, and a function h that is submodular on $\{0,1\}^J$.

Sort the entries in $\hat{y}$ such that $y_{(1)} \geq \cdots \geq y_{(J)}$. Obtain the permutation $\{(1), \ldots, (J)\}$ of $[J]$.
Letting $R_{(j)} := \{(1), \ldots, (j)\}$, $\forall j \in \mathcal{J}$, compute $\hat{\pi}_{(1)} = h(R_{(1)})$, and $\hat{\pi}_{(j)} = h(R_{(j)}) - h(R_{(j-1)})$ for $j = 2, \ldots, J$.
if $\hat{\pi}^\top \hat{y} > \hat{t}$ **then**
    We generate a valid extended polymatroid inequality $\hat{\pi}^\top y \leq t$.
else
    The current solution $(\hat{y}, \hat{t})$ satisfies $h(\hat{y}) \leq \hat{t}$.
end if
return either $(\hat{y}, \hat{t})$ is feasible, or a violated extended polymatroid inequality $\hat{\pi}^\top y \leq t$.

The strength and efficient separation of the extended polymatroid inequality motivate us to explore the submodularity of function $g(y) := \mu^\top y + \sqrt{y^\top \Lambda y}$. In a special case, Λ is assumed to be a diagonal matrix and so the random coefficients $\tilde{t}_{ij}, j \in [J]$ for the same i are uncorrelated. In this case, Atamtürk and Narayanan (2008) successfully show that $g(y)$ is submodular. As a result, we can strengthen a DCBP by incorporating extended polymatroid inequalities in the form $\pi^\top y \leq T$, where $\pi \in \mathrm{EP}_g$. Unfortunately, the submodularity of $g(y)$ quickly fades when the off-diagonal entries of Λ become non-zero, e.g., when Λ is associated with a general covariance matrix. We present an example as follows.

Example 2. Suppose that $[J] = \{1,2,3\}$, $\mu = [0,0,0]^\top$, and

$$\Lambda = \begin{bmatrix} 0.6 & -0.2 & 0.2 \\ -0.2 & 0.7 & 0.1 \\ 0.2 & 0.1 & 0.6 \end{bmatrix}.$$

The three eigenvalues of Λ are 0.2881, 0.7432, and 0.8687, and so $\Lambda \succ 0$. However, function $g(y) = \mu^\top y + \sqrt{y^\top \Lambda y}$ is not submodular because $g(R \cup \{j\}) - g(R) < g(S \cup \{j\}) - g(S)$, where $R = \{1\}$, $S = \{1,2\}$, and $j = 3$.

In this section, we provide a sufficient condition for function $g(y)$ being submodular for general

Λ. To this end, we apply a necessary and sufficient condition derived in Nemhauser et al. (1978) for quadratic function $y^\top \Lambda y$ being submodular (see the second paragraph on Page 276 of Nemhauser et al. (1978), following Proposition 3.5). We summarize this condition in the following theorem.

Theorem 3. (Nemhauser et al. (1978)) Define function $h : \{0,1\}^J \to \mathbb{R}$ such that $h(y) := y^\top \Lambda y$, where $\Lambda \in \mathbb{R}^{J\times J}$ represents a symmetric matrix. Then, $h(y)$ is submodular if and only if $\Lambda_{rs} \le 0$ for all $r, s \in [J]$ and $r \neq s$.

Note that Theorem 3 does not assume $\Lambda \succeq 0$ and so can be applied to general (convex or non-convex) quadratic functions. Theorem 3 leads to a sufficient condition for function $g(y)$ being submodular. We summarize the main result of this section in the following theorem and present a proof.

Theorem 4. Let $\Lambda \in \mathbb{R}^{J\times J}$ represent a symmetric and positive semidefinite matrix that satisfies (i) $2\sum_{s=1}^{J} \Lambda_{rs} \ge \Lambda_{rr}$ for all $r \in [J]$ and (ii) $\Lambda_{rs} \le 0$ for all $r, s \in [J]$ and $r \neq s$. Then, function $g(y) = \mu^\top y + \sqrt{y^\top \Lambda y}$ is submodular.

Proof. As $\mu^\top y$ is submodular in y, it suffices to prove that $\sqrt{y^\top \Lambda y}$ is submodular. Hence, we can assume $\mu = 0$ without loss of generality. We let $f(x) = \sqrt{x}$ and $h(y) = y^\top \Lambda y$. Then, $g(y) = f(h(y))$.

First, we note that $h(y \vee e_r) \ge h(y)$ for all $y \in \{0,1\}^J$ and $r \in [J]$, where $a \vee b = [\max\{a_1, b_1\}, \ldots, \max\{a_J, b_J\}]^\top$ for $a, b \in \mathbb{R}^J$. Indeed, if $y_r = 1$ then $y \vee e_r = y$ and so $h(y \vee e_r) = h(y)$. Otherwise, if $y_r = 0$, then $y \vee e_r = y + e_r$ and so

$$
\begin{aligned}
h(y \vee e_r) &= y^\top \Lambda y + 2e_r^\top \Lambda y + e_r^\top \Lambda e_r \\
&= y^\top \Lambda y + 2 \sum_{s:\ y_s = 1} \Lambda_{rs} + \Lambda_{rr} \\
&\ge y^\top \Lambda y + 2 \sum_{\substack{s=1 \\ s\neq r}}^{J} \Lambda_{rs} + \Lambda_{rr} \\
&\ge y^\top \Lambda y \ = \ h(y),
\end{aligned}
$$

where the first inequality is due to $y_r = 0$ and condition (ii), and the second inequality is due to condition (i). It follows that $h(y') \ge h(y)$ for all $y, y' \in \{0,1\}^J$ such that $y' \ge y$. Hence, $h(y)$ is increasing.

Second, based on Theorem 3, $h(y)$ is submodular due to condition (ii). It follows that $g(y) = f(h(y))$ is submodular because function f is concave and nondecreasing and function h is submodular and increasing (see, e.g., Proposition 2.2.5 in Simchi-Levi et al. (2014))[1]. □

Theorem 4 generalizes the sufficient condition in Atamtürk and Narayanan (2008) because conditions (i)–(ii) are automatically satisfied if Λ is diagonal and positive definite.

For general $\Lambda \succ 0$ that does not satisfy sufficient conditions (i)–(ii), we can approximate SOC constraint (9) by replacing Λ with a matrix Δ that satisfies these conditions. We derive relaxed and conservative submodular approximations of constraint (9) in the following theorem.

Theorem 5. Constraint (9) implies the SOC constraint

$$\mu^\top y + \sqrt{y^\top \Delta^{\mathrm{L}} y} \ \le \ T, \tag{11}$$

where function $g^{\mathrm{L}}(y) := \mu^\top y + \sqrt{y^\top \Delta^{\mathrm{L}} y}$ is submodular and Δ^{L} is an optimal solution of SDP

$$\begin{aligned}
\min_{\Delta} \quad & ||\Delta - \Lambda||_2 && \text{(12a)}\\
\text{s.t.} \quad & 0 \ \preceq \ \Delta \ \preceq \ \Lambda, && \text{(12b)}\\
& 2\sum_{s=1}^{J} \Delta_{rs} \ge \Delta_{rr}, \quad \forall r \in [J], && \text{(12c)}\\
& \Delta_{rs} \le 0, \quad \forall r, s \in [J] \text{ and } r \neq s. && \text{(12d)}
\end{aligned}$$

Additionally, constraint (9) is implied by the SOC constraint

$$\mu^\top y + \sqrt{y^\top \Delta^{\mathrm{U}} y} \ \le \ T, \tag{13}$$

where function $g^{\mathrm{U}}(y) := \mu^\top y + \sqrt{y^\top \Delta^{\mathrm{U}} y}$ is submodular and Δ^{U} is an optimal solution of SDP

$$\begin{aligned}
\min_{\Delta} \quad & ||\Delta - \Lambda||_2 && \text{(14a)}\\
\text{s.t.} \quad & \Delta \ \succeq \ \Lambda, \quad \text{(12c)–(12d)}. && \text{(14b)}
\end{aligned}$$

Proof. By construction, $g^{\mathrm{L}}(y)$ is submodular because Δ^{L} satisfies constraints (12c)–(12d) and so conditions (i)–(ii). Additionally, constraint (9) implies (11) because Δ^{L} satisfies constraint (12b)

[1] Proposition 2.2.5 in Simchi-Levi et al. (2014) assumes that $y \in \mathbb{R}^n$ and provides a sufficient condition for g being supermodular. It can be shown that a similar proof of this proposition applies to our case.

and so $\Delta^{\mathrm{L}} \preceq \Lambda$. Similarly, we obtain that $g^{\mathrm{U}}(y)$ is submodular and constraint (9) is implied by (13). $\square$

Note that there always exist matrices Δ^{L} and Δ^{U} that are feasible to SDPs (12a)–(12d) and (14a)–(14b), respectively. For example, $\mathrm{diag}(\lambda_{\min}, \ldots, \lambda_{\min}) \in \mathbb{R}^{J\times J}$ satisfy constraints (12b)–(12d), where $\lambda_{\min}$ represents the smallest eigenvalue of matrix Λ. Additionally, $\mathrm{diag}(\lambda_{\max}, \ldots, \lambda_{\max}) \in \mathbb{R}^{J\times J}$ satisfy constraints (14b), where $\lambda_{\max}$ represents the largest eigenvalue of matrix Λ.

By minimizing the ℓ^2 distance between Δ and Λ in objective functions (12a) and (14a), we find "optimal" approximations of Λ that satisfies sufficient conditions (i)–(ii) in Theorem 4. Accordingly, we obtain "optimal" submodular approximations of the 0-1 SOC constraint (9). There are many possible alternatives of the ℓ^2 norm in (12a) and (14a). For example, formulations (12a)–(12d) and (14a)–(14b) remain SDPs if the ℓ^2 norm is replaced by the ℓ^1 norm or the ℓ^∞ norm. We have empirically tested ℓ^1, ℓ^2, and ℓ^∞ norms based on a server allocation problem (see Section 5.2 for a brief description) and the ℓ^2 norm leads to the largest improvement on CPU time. In computation, we only need to solve these two SDPs once to obtain Δ^{L} and Δ^{U}. Then, extended polymatroid inequalities can be obtained from the relaxed approximation (11). Additionally, the conservative approximation (13) leads to an upper bound of the optimal objective value of the related DCBP.

Example 3. Recall the 3×3 matrix Λ in Example 2 and the corresponding function $g(y) = \mu^\top y + \sqrt{0.6y_1^2 + 0.7y_2^2 + 0.6y_3^2 - 0.4y_1y_2 + 0.4y_1y_3 + 0.2y_2y_3}$ being not submodular. We set $\mu = [0, 0, 0]^\top$ and apply Theorem 5 to optimize the two SDPs (12) and (14), yielding the following two positive semidefinite matrices:

$$\Delta^{\mathrm{L}} = \begin{bmatrix} 0.35 & -0.15 & 0 \\ -0.15 & 0.37 & 0 \\ 0 & 0 & 0.38 \end{bmatrix} \text{ and}$$

$$\Delta^{\mathrm{U}} = \begin{bmatrix} 0.83 & -0.22 & 0 \\ -0.22 & 0.95 & 0 \\ 0 & 0 & 0.82 \end{bmatrix}.$$

Due to Theorem 4, $g^{\mathrm{L}}(y) := \mu^\top y + \sqrt{y^\top \Delta^{\mathrm{L}} y} = \sqrt{0.35y_1^2 + 0.37y_2^2 + 0.38y_3^2 - 0.3y_1y_2}$ and $g^{\mathrm{U}}(y) := \mu^\top y + \sqrt{y^\top \Delta^{\mathrm{U}} y} = \sqrt{0.83y_1^2 + 0.95y_2^2 + 0.82y_3^2 - 0.44y_1y_2}$ are submodular.

Now suppose that $T = 0.8$ and we are given $\hat{y} = [\hat{y}_1, \hat{y}_2, \hat{y}_3]^\top = [1, 0.5, 0.9]^\top$ with $g(\hat{y}) =$

$\mu^\top \hat{y} + \sqrt{\hat{y}^\top \Lambda \hat{y}} = 1.23 > 0.8 = T$. First, with respect to constraint $g^{\mathrm{L}}(y) \le T$, we follow Algorithm 1 to find an extended polymatroid inequality and note that this inequality is also valid for constraint $g(y) \le T$. Specifically, we sort the entries of $\hat{y}$ to obtain $\hat{y}_1 \ge \hat{y}_3 \ge \hat{y}_2$ and $\{1, 3, 2\}$, the corresponding permutation. It follows that $R_{(1)} = \{1\}$, $R_{(2)} = \{1, 3\}$, and $R_{(3)} = \{1, 3, 2\}$. Hence, $\hat{\pi}_{(1)} = g^{\mathrm{L}}([1, 0, 0]^\top) = 0.59$, $\hat{\pi}_{(2)} = g^{\mathrm{L}}([1, 0, 1]^\top) - g^{\mathrm{L}}([1, 0, 0]^\top) = 0.26$, and $\hat{\pi}_{(3)} = g^{\mathrm{L}}([1, 1, 1]^\top) - g^{\mathrm{L}}([1, 0, 1]^\top) = 0.04$. This generates the extended polymatroid inequality $0.59y_1 + 0.26y_3 + 0.04y_2 \le 0.8$ that cuts off $\hat{y}$. Second, we can replace constraint $g(y) \le T$ with $g^{\mathrm{U}}(y) \le T$ in a DCBP to obtain a conservative approximation.

4.2 Valid Inequalities in a Lifted Space

In Section 4.1, we derive extended polymatroid inequalities based on a relaxed approximation of SOC constraint (9). These valid inequalities might not be facet-defining if matrix Λ does not satisfy sufficient conditions (i)–(ii) in Theorem 4. In this section, we show that the submodularity of (9) holds for general Λ in a lifted (i.e., higher-dimensional) space. Accordingly, we derive extended polymatroid inequalities in the lifted space.

To this end, we reformulate constraint (9) as $\mu^\top y \le T$ and $y^\top \Lambda y \le (T - \mu^\top y)^2$, i.e., $y^\top(\Lambda - \mu\mu^\top)y + 2T\mu^\top y \le T^2$. Note that $\mu^\top y \le T$ is because $T - \mu^\top y \ge \sqrt{y^\top \Lambda y} \ge 0$ by (9). Then, we define $w_{jk} = y_j y_k$ for all $j, k \in [J]$ and augment vector y to vector $v = [y_1, \ldots, y_J, w_{11}, \ldots, w_{1J}, w_{21}, \ldots, w_{JJ}]^\top$. We can incorporate the following McCormick inequalities to define each w_{ij}:

$$w_{jk} \le y_j, \quad w_{jk} \le y_k, \quad w_{jk} \ge y_j + y_k - 1, \quad w_{ij} \ge 0. \tag{15}$$

Accordingly, we rewrite (9) as $[\mu^\top, 0^\top]v \le T$ and

$$a^\top v + v^\top B_{\mathrm{N}} v \le T^2, \tag{16}$$

where we decompose $(\Lambda - \mu\mu^\top)$ to be the sum of two matrices, one containing all positive entries and the other containing all nonpositive entries. Accordingly, we define vector $a := [2T\mu; B_{\mathrm{P}}]^\top$ with $B_{\mathrm{P}} \in \mathbb{R}_+^{J^2}$ representing all the positive entries after vectorization, and matrix $B_{\mathrm{N}} \in \mathbb{R}_-^{(J+J^2)\times(J+J^2)}$ collects all nonpositive entries, i.e., $B_{\mathrm{N}} = \begin{bmatrix} -(\Lambda - \mu\mu^\top)_- & 0 \\ 0 & 0 \end{bmatrix}$, where $(x)_- = -\min\{0, x\}$ for $x \in \mathbb{R}$.

As $a^\top v + v^\top B_{\mathrm{N}} v$ is a submodular function of v by Theorem 3, we can incorporate extended polymatroid inequalities to strengthen the lifted SOC constraints (9). We summarize this result in the following theorem.

Theorem 6. Define function $h : \{0,1\}^{J+J^2} \to \mathbb{R}$ such that $h(v) := a^\top v + v^\top B_{\mathrm{N}} v$. Then, h is submodular. Furthermore, inequality $\pi^\top v \le T^2$ is valid for set $\{v \in \{0,1\}^{J+J^2} : h(v) \le T^2\}$ for all $\pi \in \mathrm{EP}_h$ and the separation of this inequality can be done by Algorithm 1.

Note that this lifting procedure introduces J^2 additional variables w_{ijk} for each i. However, w_{ijk} can be treated as continuous variables when solving DCBP in view of the McCormick inequalities (15), and the number of w_{ijk} variables can be reduced by half because $w_{ijk} = w_{ikj}$. In our numerical studies later, we derive more valid inequalities to strengthen the formulation in the lifted space for distributionally robust chance-constrained bin packing problems that involve DRCCs, using the bin packing structure.

Example 4. Recall Example 2 and the corresponding function $g(y)$ being not submodular. We set $\mu = [0,0,0]^\top$ and $T = 0.8$, and rewrite the constraint $g(y) \le T$ in the form of (16) as

$$0.6v_4 + 0.2v_6 + 0.7v_8 + 0.1v_9 + 0.2v_{10} + 0.1v_{11} + 0.6v_{12} - 0.4v_1v_2 \le 0.64,$$

where $v := [y_1, y_2, y_3, w_{11}, w_{12}, \ldots, w_{33}]^\top = [v_1, \ldots, v_{12}]^\top$. Now suppose that we are given $\hat{y} = [1, 0.5, 0.9]^\top$. The corresponding $\hat{v} = [1, 0.5, 0.9, 1, 0.5, 0.9, 0.5, 0.25, 0.45,\ 0.9, 0.45, 0.81]^\top$ violates the above inequality. We follow Algorithm 1 to find an extended polymatroid inequality in the lifted space of v. Specifically, we sort the entries of $\hat{v}$ to obtain $\hat{v}_1 \ge \hat{v}_4 \ge \hat{v}_3 \ge \hat{v}_6 \ge \hat{v}_{10} \ge \hat{v}_{12} \ge \hat{v}_2 \ge \hat{v}_5 \ge \hat{v}_7 \ge \hat{v}_9 \ge \hat{v}_{11} \ge \hat{v}_8$ and the corresponding permutation $\{1, 4, 3, 6, 10, 12, 2, 5, 7, 9, 11, 8\}$. It follows that $\hat{\pi}_{(2)} = 0.6$, $\hat{\pi}_{(4)} = 0.2$, $\hat{\pi}_{(5)} = 0.2$, $\hat{\pi}_{(6)} = 0.6$, $\hat{\pi}_{(7)} = -0.4$, $\hat{\pi}_{(10)} = 0.1$, $\hat{\pi}_{(11)} = 0.1$, $\hat{\pi}_{(12)} = 0.7$, and all other $\hat{\pi}_{(i)}$'s equal zero. This generates the following extended polymatroid inequality

$$0.6v_4 + 0.2v_6 + 0.2v_{10} + 0.6v_{12} - 0.4v_2 + 0.1v_9 + 0.1v_{11} + 0.7v_8 \le 0.64$$

that cuts off $\hat{v}$.

5 Numerical Studies

We numerically evaluate the performance of our proposed models and solution approaches. In Section 5.1, we present the formulation of a chance-constrained bin-packing problem with DRCCs and the related 0-1 SOC reformulations. We describe the solution methods and more valid inequalities based on the bin packing structure. In Section 5.2, we describe the experimental setup of the stochastic bin packing instances. Our results consist of two parts, which report the CPU time (Section 5.3) and the out-of-sample performance of solutions given by different models (Appendix C), respectively. More specifically, Section 5.3 demonstrates the computational efficacy of the valid inequalities we derived in Section 4 for the original or lifted SOC constraints. Appendix C demonstrates that DCBP solutions can well protect against the distributional ambiguity as opposed to the solutions obtained by following the Gaussian distribution assumption or by the SAA method.

5.1 Formulation of Ambiguous Chance-Constrained Bin Packing

For the classical bin packing problem, $[I]$ is the set of bins and $[J]$ is the set of items, where each bin i has a weight capacity T_i and each item j, if assigned to bin i, has a random weight $\tilde{t}_{ij}$. The deterministic bin packing problem aims to assign all J items to a minimum number of bins, while respecting the capacity of each bin. If we consider a slightly more general setting by introducing a cost for each assignment, then the DCBP of bin packing with DRCCs is presented as:

$$
\begin{aligned}
\min_{\mathbf{z},\mathbf{y}} \quad & \sum_{i=1}^{I} c_i^{\mathrm{z}} z_i + \sum_{i=1}^{I}\sum_{j=1}^{J} c_{ij}^{\mathrm{y}} y_{ij} && \text{(17a)}\\
\text{s.t.} \quad & y_{ij} \le \rho_{ij} z_i \quad \forall i \in [I],\ j \in [J] && \text{(17b)}\\
& \sum_{i=1}^{I} y_{ij} = 1 \quad \forall j \in [J] && \text{(17c)}\\
& y_{ij}, z_i \in \{0,1\} \quad \forall i \in [I],\ j \in [J], && \text{(17d)}\\
& \inf_{\mathbb{P}\in\mathcal{D}} \mathbb{P}\left\{ \sum_{j=1}^{J} \tilde{t}_{ij} y_{ij} \le T_i \right\} \ \ge\ 1 - \alpha_i \quad \forall i \in [I], && \text{(17e)}
\end{aligned}
$$

where c_i^{z} represents the cost of opening bin i and c_{ij}^{y} represents the cost of assigning item j to bin i. For each $i \in [I]$ and $j \in [J]$, we let binary variables z_i represent if bin i is open (i.e., $z_i = 1$ if open and $z_i = 0$ otherwise), binary variables y_{ij} represent if item j is assigned to bin i (i.e., $y_{ij} = 1$ if assigned and $y_{ij} = 0$ otherwise), and parameters ρ_{ij} represent if we can assign item j to bin i (i.e., $\rho_{ij} = 1$ if we can and $\rho_{ij} = 0$ otherwise). The objective function (17a) minimizes the total cost of opening bins and assigning items to bins. Constraints (17b) ensure that all items are assigned to open bins, constraints (17c) ensure that each item is assigned to one and only one bin, and constraints (17e) are the DRCCs.

In our computational studies, we follow Section 3 to derive 0-1 SOC reformulations of model (17) and then follow Section 4 to derive valid inequalities in the original and lifted space for the 0-1 SOC reformulations. We strengthen the extended polymatroid inequalities, as well as derive valid inequalities in the lifted space containing variables z_i, y_{ij}, and w_{ijk} to further strengthen the formulation as follows. We refer to Appendices A and B for the detailed proofs of the valid inequalities below, and will test their effectiveness later.

Theorem 7. For all extended polymatroid inequalities $\pi^\top y_i \le T$ with regard to bin i, $\forall i \in [I]$, inequality

$$\pi^\top y_i \le T z_i \tag{18a}$$

is valid for the DCBP formulation. Similarly, for all extended polymatroid inequalities $\pi^\top v_i \le T^2$ with regard to bin i, $\forall i \in [I]$, inequality

$$\pi^\top v_i \le T^2 z_i \tag{18b}$$

is valid for the DCBP formulation.

Theorem 8. Consider set

$$L = \Big\{(z, y, w) \in \{0,1\}^{I\times(IJ)\times(IJ^2)} : \ \text{(17b)–(17d)}, \ w_{ijk} = y_{ij}y_{ik}, \ \forall j,k \in [J]\Big\}.$$

Without loss of optimality, the following inequalities are valid for L:

$$w_{ijk} \ \ge y_{ij} + y_{ik} + \sum_{\substack{\ell=1 \\ \ell\ne i}}^{I} w_{\ell jk} - 1 \quad \forall j,k \in [J] \tag{19a}$$

$$w_{ijk} \ \ge y_{ij} + y_{ik} - z_i \quad \forall i \in [I], \ \forall j,k \in [J] \tag{19b}$$

$$\sum_{\substack{j=1 \\ j\ne k}}^{J} w_{ijk} \ \le \ \sum_{j=1}^{J} y_{ij} - z_i \quad \forall i \in [I], \ \forall k \in [J] \tag{19c}$$

$$\sum_{j=1}^{J}\sum_{k=j+1}^{J} w_{ijk} \ \geq \ \sum_{j=1}^{J} y_{ij} - z_i \quad \forall i \in [I]. \tag{19d}$$

Remark 2. We note that valid inequalities (19a)–(19d) are *polynomially many* and all coefficients are in *closed-form*. Hence, we do not need any separation processes for these inequalities, and we can incorporate them in the DCBP formulation without dramatically increasing its size.

5.2 Computational Setup

We first consider $I = 6$ servers (i.e., bins) and $J = 32$ appointments (i.e., items) to test the DCBP model under various distributional assumptions and ambiguity sets. The daily operating time limit (i.e., capacity) T_i of each server i varies in between [420, 540] minutes (i.e., 7–9 hours). We let the opening cost c_i^{z} of each server i be an increasing function of T_i such that $c_i^{\mathrm{z}} = T_i^2/3600 + 3T_i/60$, and let all assignment costs $c_{ij}^{\mathrm{y}}, \ \forall i \in [I], \ j \in [J]$ vary in between $[0, 18]$, so that the total opening cost and the total assignment cost have similar magnitudes. The above problem size and parameter settings follow the literature of surgery block allocation (see, e.g., Denton et al. (2010); Shylo et al. (2012); Deng et al. (2016)).

To generate samples of random service time (i.e., random item weight), we consider "high mean (hM)" and "low mean (ℓM)" being 25 minutes and 12.5 minutes, respectively. We set the coefficient of variation (i.e., the ratio of the standard deviation to the mean) as 1.0 for the "high variance (hV)" case and as 0.3 for the "low variance (ℓV)" case. We equally mix all four types of appointments with "hMhV", "hMℓV", "ℓMhV", "ℓMℓV", and thus have eight appointments of each type. We sample 10,000 data points as the random service time of each appointment on each server, following a Gaussian distribution with the above settings of mean and standard deviation. We will hereafter call them the in-sample data. To formulate the 0-1 SOC models with diagonal covariance matrices, we use the empirical mean and standard deviation of each $\tilde{t}_{ij}$ obtained from the in-sample data and set $\alpha_i = 0.05, \ \forall i \in [I]$. Using the same α_i-values, we formulate the 0-1 SOC models under general covariance matrices, for which we use the empirical mean and covariance matrix obtained from the in-sample data. The empirical covariance matrices we obtain have most of their off-diagonal entries being non-zero, and some being quite significant.

All the computation is performed on a Windows 7 machine with Intel(R) Core(TM) i7-2600

CPU 3.40 GHz and 8GB memory. We implement the optimization models and the branch-and-cut algorithm using commercial solver GUROBI 5.6.3 via Python 2.7.10. The GUROBI default settings are used for optimizing all the 0-1 programs, and we set the number of threads as one. When implementing the branch-and-cut algorithm, we add the violated extended polymatroid inequalities using GUROBI `callback` class by `Model.cbCut()` for both integer and fractional temporary solutions. For all the nodes in the branch-and-bound tree, we generate violated cuts at each node as long as any exists. The optimality gap tolerance is set as 0.01%. We also set the threshold for identifying violated cuts as 10^{-4}, and set the time limit for computing each instance as 3600 seconds.

5.3 CPU Time Comparison

We solve 0-1 SOC reformulations or approximations of DCBP, and use either a diagonal or a general covariance matrix in each model. Our valid inequalities significantly reduce the solution time of directly solving the 0-1 SOC models in GUROBI, while the extended polymatroid inequalities generated based on the approximate and lifted SOC constraints perform differently depending on the problem size. The details are presented as follows.

5.3.1 Computing 0-1 SOC models with diagonal matrices

We first optimize 0-1 SOC models with a diagonal matrix in constraint (9), of which the left-hand side function $g(y)$ is submodular, and thus we use extended polymatroid inequalities (18a) with $\pi \in EP_g$ in a branch-and-cut algorithm. Table 1 presents the CPU time (in seconds), optimal objective values, and other solution details (including "**Server**" as the number of open servers, "**Node**" as the total number of branching nodes, and "**Cut**" as the total number of cuts added) for solving the three 0-1 SOC models **DCBP1** (i.e., using ambiguity set $\mathcal{D}_1$), **DCBP2** (i.e., using ambiguity set $\mathcal{D}_2$ with $\gamma_1 = 1,\ \gamma_2 = 2$), and **Gaussian** (assuming Gaussian distributed service time). We also implement the SAA approach (i.e., row "**SAA**") by optimizing the MILP reformulation of the chance-constrained bin packing model based on the 10,000 in-sample data points. We compare the branch-and-cut algorithm using our extended polymatroid inequalities (in rows "**B&C**") with directly solving the 0-1 SOC models in GUROBI (in rows "**w/o Cuts**").

In Table 1, the branch-and-cut algorithm quickly optimizes DCBP1 and DCBP2. Especially, if

Table 1: CPU time and solution details for solving instances with diagonal matrices

Approach	Model	Time (s)	Opt. Obj.	Server	Opt. Gap	Node	Cut
B&C	DCBP1	0.73	328.99	3	0.00%	83	82
	DCBP2	27.50	366.54	3	0.00%	2146	2624
	Gaussian	0.13	297.94	2	0.00%	0	0
w/o Cuts	DCBP1	95.73	328.99	3	0.01%	76237	N/A
	DCBP2	LIMIT	380.09	2	9.15%	409422	N/A
	Gaussian	0.02	297.94	2	0.00%	16	N/A
SAA	MILP	21.20	297.94	2	0.00%	89	N/A

being directly solved by GUROBI, DCBP2 cannot be solved within the 3600-second time limit and ends with a 9.15% optimality gap. Solving DCBP1 by using the branch-and-cut algorithm is much faster than solving the large-scale SAA-based MILP model, while the solution time of DCBP2 is similar to the latter. The two DCBP models also yield higher objective values, since they both provide more conservative solutions that open one more server than either the Gaussian or the SAA-based approach.

5.3.2 Computing 0-1 SOC models with general covariance matrices

In this section, we focus on testing DCBP2 yielded by the ambiguity set $\mathcal{D}_2$ with parameters $\gamma_1 = 1$, $\gamma_2 = 2$, and $\alpha_i = 0.05,\ \forall i \in [I]$. We use empirical covariance matrices of the in-sample data. Note that these covariance matrices are general and non-diagonal. We compare the time of solving the 0-1 SOC reformulations of DCBP2 on ten independently generated instances. We examine two implementations of the branch-and-cut algorithm: one uses extended polymatroid inequalities (18a) with $\pi \in EP_{g^L}$ based on the relaxed 0-1 SOC constraint (11), and the other uses the extended polymatroid inequalities (18b) based on the lifted SOC constraint.

Table 2 reports the CPU time (in seconds) and number of branching nodes (in column "**Node**") for various methods. First, we directly solve the 0-1 SOC models of DCPB2 in GUROBI, without or with the linear valid inequalities (19a)–(19d), and report their results in columns "**w/o Cuts**" and "**Ineq.**", respectively. We then implement the branch-and-cut algorithm, and examine the results of using extended polymatroid inequalities (18a) with $\pi \in EP_{g^L}$ (reported in columns "**B&C-Relax**") and cuts (18b) based on lifted SOC constraints (reported in columns "**B&C-Lifted**"). (The latter does not involve valid inequalities (18a) used in the former.) The time reported under

B&C-Relax also involves the time of solving SDPs for obtaining Δ^{L} and the relaxed 0-1 SOC constraint (11). The time of solving related SDPs are quite small (varying from 1 to 2 seconds for instances of different sizes) and negligible as compared to the total B&C time. For both B&C methods, we also present the number of extended polymatroid inequalities (see column "**Cut**") added.

Table 2: CPU time of DCBP2 solved by different methods with general covariance matrices

Instance	w/o Cuts		Ineq.		B&C-Relax			B&C-Lifted		
	Time (s)	Node	Time (s)	Node	Time (s)	Node	Cut	Time (s)	Node	Cut
1	286.29	10409	156.50	795	51.99	9095	702	**35.03**	618	823
2	433.32	10336	167.91	687	26.63	6524	698	**12.34**	405	235
3	284.17	10434	206.82	971	70.43	17420	621	**29.84**	595	729
4	310.11	10302	139.06	656	**15.37**	2467	723	25.31	419	617
5	329.32	10453	181.83	777	56.53	12349	737	**35.09**	678	921
6	365.28	10300	168.26	652	**23.89**	4807	695	26.73	555	595
7	296.55	10759	198.87	873	45.21	11585	738	**21.08**	440	626
8	278.62	10490	211.05	900	53.84	14540	721	**47.78**	1064	1686
9	139.24	7771	177.41	632	19.90	3918	645	**19.37**	216	360
10	297.72	10330	159.52	822	30.36	6877	649	**29.43**	400	727

In Table 2, we highlight the solution time of the method that runs the fastest for each instance. Note that without the extended polymatroid inequalities or the valid inequalities, the GUROBI solver takes the longest time for solving all the instances except instance #9. Adding the valid inequalities (19a)–(19d) to the solver reduces the solution time by 40% or more in almost all the instances, and drastically reduces the number of branching nodes. The extended polymatroid inequalities (18a)–(18b) further reduce the CPU time significantly (see columns "**B&C-Relax**" and "**B&C-Lifted**"). Moreover, for all the instances having 6 servers and 32 appointments, the algorithm using the extended polymatroid inequalities (18b) runs faster in eight out of ten instances than the algorithm using cuts (18a) based on relaxed SOC constraints without lifting. It indicates that the extended polymatroid inequalities generated by the lifted SOC constraints are more effective than those generated by the relaxed SOC constraints. This observation is overturned when we later increase the problem size.

In the following, we continue reporting the CPU time of solving DCBP2 with general covariance matrix. We vary the problem sizes (i.e., values of I and J) in Section 5.3.3, and vary the values of Λ in the SOC constraint (9) in Section 5.3.4.

5.3.3 Solving 0-1 SOC models under different problem sizes

We use the same problem settings as in Section 5.2, and vary $I = 6, 8, 10$ and $J = 32, 40$ to test DCBP2 instances with different sizes. We still keep an equal mixture of all the four appointment types in each instance. Table 3 presents the computational time (in seconds), the total number of branching nodes ("**Node**"), and the total number of extended polymatroid inequalities generated ("**Cuts**"; if applicable) for solving the 0-1 SOC reformulation of DCBP2 by directly using GUROBI ("**w/o Cuts**") and by using the two implementations of the extended polymatroid inequalities ("**B&C-Relax**" and "**B&C-Lifted**").

Table 3: CPU time of DCBP2 with general covariance matrices for different problem sizes

	Method		$J = 32$					$J = 40$				
		Inst.	1	2	3	4	5	6	7	8	9	10
$I = 6$	B&C-Relax	Time (s)	51.99	26.63	70.43	**15.37**	56.53	**6.87**	**12.76**	**1.59**	**2.36**	**12.73**
		Node	9095	6524	17420	2467	12349	1009	1322	176	285	1270
		Cut	702	698	621	723	737	174	604	171	179	602
	B&C-Lifted	Time (s)	**35.03**	**12.34**	**29.84**	25.31	**35.09**	64.58	98.18	91.12	60.11	59.50
		Node	618	405	595	419	678	274	484	447	289	234
		Cut	823	235	729	617	921	470	690	688	462	394
	w/o Cuts	Time (s)	286.29	433.32	284.17	310.11	329.32	1654.31	208.12	1182.46	1580.41	1266.27
		Node	10409	10336	10434	10302	10453	10525	1272	10732	10658	10642
$I = 8$	B&C-Relax	Time (s)	**41.57**	139.41	**55.22**	261.24	305.72	**23.91**	**9.73**	**17.76**	**27.16**	**12.98**
		Node	8342	29042	12267	49820	61334	2130	1240	1561	2607	1024
		Cut	737	770	742	803	790	714	199	728	702	690
	B&C-Lifted	Time (s)	106.03	**28.55**	84.64	**97.05**	**13.56**	331.29	273.14	307.06	178.41	161.39
		Node	678	502	647	634	125	1177	836	1397	457	529
		Cut	114	691	128	143	216	1781	1175	2066	703	719
	w/o Cuts	Time (s)	866.12	597.43	649.72	683.18	497.15	2265.53	2428.60	2294.62	1781.95	851.99
		Node	10338	10305	10309	10306	14386	11441	11219	11708	11128	5241
$I = 10$	B&C-Relax	Time (s)	**3.75**	**9.28**	**6.56**	**3.23**	**16.71**	**29.94**	**80.34**	**22.58**	**24.48**	**339.93**
		Node	637	972	659	549	2274	2336	7315	1870	1959	34306
		Cut	241	552	390	230	741	767	714	736	729	715
	B&C-Lifted	Time (s)	108.43	117.44	120.60	22.10	111.37	186.72	714.45	197.42	549.90	661.13
		Node	668	785	828	291	779	766	1108	811	896	1209
		Cut	108	191	314	281	188	1196	850	1106	568	808
	w/o Cuts	Time (s)	987.92	1140.23	183.06	1113.09	1425.83	2382.97	2917.03	LIMIT	2052.42	2451.62
		Node	10353	10357	4992	10307	10401	11015	11197	12101	10812	11001

In Table 3, we again highlight the solution time of the method that runs the fastest in each instance. We keep the first five instances we reported in Table 2 for instances with $I = 6,\ J = 32$, and report five instances for other (I, J) combinations. From Table 3, we observe that both implementations of the extended polymatroid inequalities run significantly faster than directly using GUROBI, especially when we increase the problem sizes (i.e., I increased from 6 to 10, and J increased from 32 to 40). In particular, the CPU time of directly using GUROBI is consistently 1 or 2 orders of magnitude larger than that of our approaches. For smaller (I, J)-values (e.g., $(I, J) = (6, 32)$ or $(I, J) = (8, 32)$), we see that **B&C-Lifted** sometimes runs faster than **B&C-**

Relax, but for all the other (I, J) combinations, the latter completely dominates the former. This is expected because the cuts (18b) are generated in a lifted space with J^2 additional variables for each server $i \in [I]$. Therefore, it makes sense that the scalability of **B&C-Lifted** is worse than that of **B&C-Relax**, which uses cuts (18a) without lifting.

5.3.4 Solving 0-1 SOC models with different Λ-values in (9)

We again focus on instances with $I = 6$ and $J = 32$ under the same general covariance matrix Σ obtained from the in-sample data points. We let $\Lambda := \Omega^2\Sigma$ in the 0-1 SOC constraint (9) and adjust the scalar Ω to obtain different Λ. We want to show how the computational time of directly using GUROBI increases as we increase Ω, as compared to using the branch-and-cut algorithm with extended polymatroid inequalities. (The results of **B&C-Lifted** are used here and similar observations can be made if the results of **B&C-Relax** are used.)

Considering specific cases of the SOC constraint (9) for modeling DCBP, we have $\Omega = \Phi^{-1}(1 - \alpha) = 1.64$ for the Gaussian approximation model when $\alpha = 0.05$, and $\Omega = \sqrt{\gamma_2/\alpha} = 6.32$ for DCBP2 when $\gamma_2 = 2$ and $\alpha = 0.05$. We test four values of Ω equally distributed in between $[1.64, 6.32]$ including the two end points.

Figure 3 depicts the average CPU time and the average number of branching nodes of solving five independent DCBP2 instances for each Ω-setting. Specifically, the four values 1.64, 3.20, 4.76, and 6.32 of Ω lead to 0.98, 97.68, 299.29, and 363.56 CPU seconds when directly using GUROBI, respectively, together with the significantly growing number of branching nodes 0, 909.8, 6662, and 10418, respectively. On the other hand, the branch-and-cut algorithm with the extended polymatroid inequalities respectively takes 1.03, 35.19, 16.09, and 26.63 CPU seconds on average for solving the same instances, and branches on average 0, 188.6, 247.2, and 374.6 nodes, respectively. This indicates that our approach is more scalable than directly using the off-the-shelf solvers.

6 Conclusions

In this paper, we considered distributionally robust individual chance constraints, where the true distributional information of the constraint coefficients is ambiguous and only the empirical first and second moments are given. The goal is to restrict the worst-case probability of violating a linear

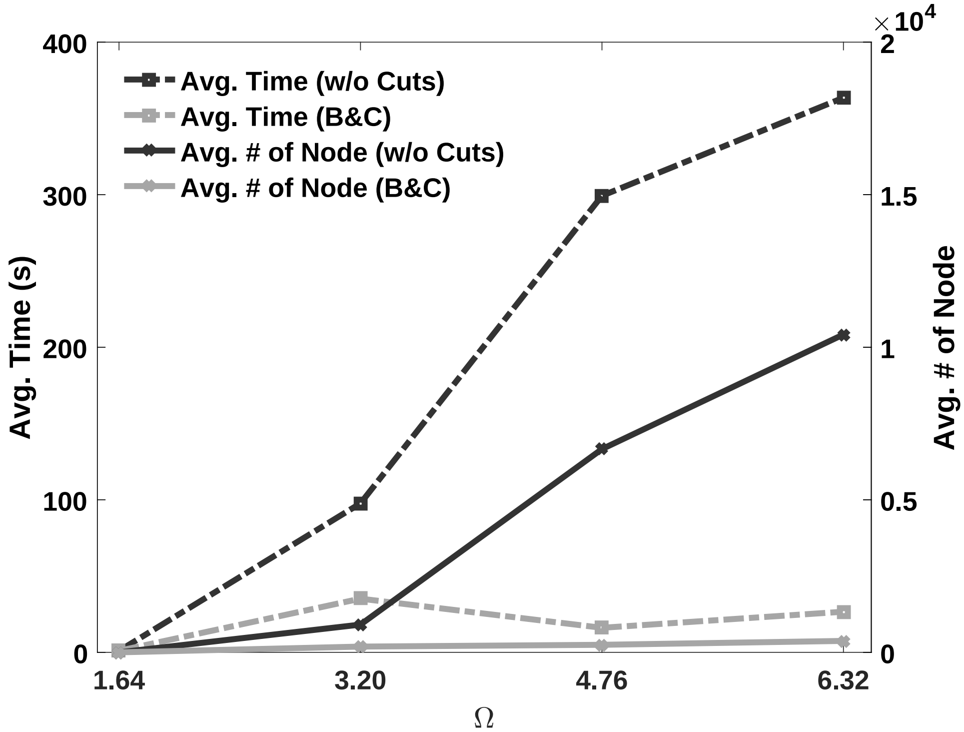

Figure 3: CPU time and number of branching nodes under different Ω-values in constraint (9)

constraint under a given threshold. We provided 0-1 SOC representations of DRCCs under two types of ambiguity sets. In addition, we derived an efficient way of obtaining extended polymatroid inequalities for the 0-1 SOC constraints in both original and lifted spaces. Via extensive numerical studies, we demonstrated that our solution approaches significantly accelerate solving the DCBP model as compared to the state-of-the-art commercial solvers. In particular, a branch-and-cut algorithm with extended polymatroid inequalities in the original space scales very well as the problem size grows.

For future research, we plan to investigate DRCCs under other types of ambiguity sets, which could take into account not only the moment information but also density or structural information of their probability distribution. The connections between SOC program, SDP, and submodular optimization are also interesting to study.

Acknowledgments

The authors are grateful to Profs. Alper Atamtürk and Andrés Gómez for their comments on an earlier version of this paper and for bringing to our attention the works Atamtürk and Bhardwaj (2018); Atamtürk and Gómez (2017); Bhardwaj (2015). The authors are grateful to the two referees and the Associate Editor for their constructive comments and helpful suggestions.

Appendix A Proof of Theorem 7

Proof. When $z_i = 1$, inequality (18a) reduces to the extended polymatroid inequality. When $z_i = 0$, we have $y_{ij} = 0$ for all $j \in [J]$ due to constraints (17b). It follows that inequality (18a) holds valid.

When $z_i = 1$, inequality (18b) reduces to the extended polymatroid inequality. When $z_i = 0$, we have $y_{ij} = 0$ for all $j \in [J]$ due to constraints (17b) and so $w_{ijk} = 0$ for all $j, k \in [J]$. It follows that $v_i = 0$ by definition. Hence, inequality (18b) holds valid. □

Appendix B Proof of Theorem 8

Proof. (**Validity of inequality** (19a)) If $j = k$, then $w_{ijk} = y_{ij}^2 = y_{ij}$. In this case, inequality (19a) reduces to $y_{ij} \geq 2y_{ij} + \sum_{\substack{\ell=1 \\ \ell \neq i}}^{I} y_{\ell j} - 1$, which clearly holds because $y_{ij} + \sum_{\substack{\ell=1 \\ \ell \neq i}}^{I} y_{\ell j} = \sum_{i=1}^{I} y_{ij} \leq 1$. If $j \neq k$, then we discuss the following two cases:

1. If $\max\{y_{ij}, y_{ik}\} = 1$, then we assume $y_{ij} = 1$ without loss of generality. It follows that $y_{\ell j} = 0$ due to constraints (17c) and so $w_{\ell jk} = y_{\ell j} y_{\ell k} = 0$ for all $\ell \neq i$. Hence, $\sum_{\substack{\ell=1 \\ \ell \neq i}}^{I} w_{\ell jk} = 0$ and inequality (19a) reduces to $w_{ijk} \geq y_{ij} + y_{ik} - 1$, which holds valid.

2. If $\max\{y_{ij}, y_{ik}\} = 0$, then $y_{ij} = y_{ik} = 0$ and $w_{ijk} = 0$. It remains to show $\sum_{\substack{\ell=1 \\ \ell \neq i}}^{I} w_{\ell jk} \leq 1$. Indeed, since $w_{\ell jk} \leq y_{\ell j}$, we have $\sum_{\substack{\ell=1 \\ \ell \neq i}}^{I} w_{\ell jk} \leq \sum_{\substack{\ell=1 \\ \ell \neq i}}^{I} y_{\ell j} \leq \sum_{\ell=1}^{I} y_{\ell j} = 1$, where the last equality is due to constraints (17c).

(**Validity of inequality** (19b)) This inequality clearly holds valid when $z_i = 1$. When $z_i = 0$, we have $y_{ij} = y_{ik} = 0$ due to constraints (17b). It follows that $w_{ijk} = y_{ij} y_{ik} = 0$ and so the inequality holds valid.

(**Validity of inequality** (19c)) This inequality holds valid when $z_i = 0$. When $z_i = 1$, this inequality is equivalent to $y_{ik} \sum_{\substack{j=1 \\ j \neq k}}^{J} y_{ij} \leq \sum_{j=1}^{J} y_{ij} - 1$. We discuss the following two cases:

1. If $y_{ik} = 0$, then $\sum_{j=1}^{J} y_{ij} \geq 1$ without loss of optimality because $z_i = 1$. Inequality (19c) holds valid.

2. If $y_{ik} = 1$, then $y_{ik} \sum_{\substack{j=1 \\ j \neq k}}^{J} y_{ij} = \sum_{\substack{j=1 \\ j \neq k}}^{J} y_{ij}$. Meanwhile, $\sum_{j=1}^{J} y_{ij} - 1 = \sum_{\substack{j=1 \\ j \neq k}}^{J} y_{ij} + y_{ik} - 1 = \sum_{\substack{j=1 \\ j \neq k}}^{J} y_{ij}$. Inequality (19c) holds valid.

(**Validity of inequality** (19d)) This inequality holds valid when $z_i = 0$. When $z_i = 1$, we have $\sum_{j=1}^{J} y_{ij} \geq 1$ without loss of optimality. It follows that $\sum_{j=1}^{J} y_{ij} = 1$ or $\sum_{j=1}^{J} y_{ij} \geq 2$, and so $(\sum_{j=1}^{J} y_{ij} - 1)(\sum_{j=1}^{J} y_{ij} - 2) \geq 0$. Hence,

$$
\begin{aligned}
\Big(\sum_{j=1}^{J} y_{ij} - 1\Big)\Big(\sum_{j=1}^{J} y_{ij} - 2\Big) &= \sum_{j=1}^{J} y_{ij}^2 + 2\sum_{j=1}^{J} \sum_{k=j+1}^{J} y_{ij} y_{ik} - 3\Big(\sum_{j=1}^{J} y_{ij}\Big) + 2 \\
&= \sum_{j=1}^{J} y_{ij} + 2\sum_{j=1}^{J} \sum_{k=j+1}^{J} w_{ijk} - 3\Big(\sum_{j=1}^{J} y_{ij}\Big) + 2 \\
&= 2\sum_{j=1}^{J} \sum_{k=j+1}^{J} w_{ijk} - 2\Big[\Big(\sum_{j=1}^{J} y_{ij}\Big) - 1\Big] \;\geq\; 0.
\end{aligned}
$$

Inequality (19d) follows. $\square$

Appendix C Out-of-Sample Performance of DCBP

Through testing instances of chance-constrained bin packing, we show that DCBP solutions have very low probabilities of violating capacities in all the out-of-sample tests, even when the distributional information is misspecified. Specifically, we evaluate the out-of-sample performance of the optimal solutions to the DCBP1, DCBP2, Gaussian-based 0-1 SOC models, and the SAA-based MILP model. To generate the out-of-sample reference scenarios, we consider either misspecified distribution type or misspecified moment information as follows.

- **Misspecified distribution type:** We sample 10,000 out-of-sample data points from a two-point distribution having the same mean and standard deviation of each random variable $\tilde{t}_{ij}$ for $i \in [I]$ and $j \in [J]$ as the in-sample data. The service time is realized as $\mu_{ij} + \frac{(1-p)}{\sqrt{p(1-p)}}\sigma_{ij}$ with probability p $(0 < p < 1)$ and as $\mu_{ij} - \frac{\sqrt{p(1-p)}}{(1-p)}\sigma_{ij}$ with probability $1 - p$, where μ_{ij} and

σ_{ij} are the sample mean and standard deviation of $\tilde{t}_{ij}$ obtained from the in-sample data. We set $p = 0.3$ so that we have smaller probability of having larger service time realizations.

- **Misspecified moments:** Alternatively, we sample 10,000 data points from the Gaussian distribution, but only consider the hMℓV type of appointments, instead of an equal mixture of all the four types. In each sample and for each $i \in [I]$, we draw a standard-Gaussian random number ρ_i, and for each $j \in [J]$, generate a service time realization as $\mu_{ij} + \rho_i \sigma_{ij}$.

Performance of solutions under diagonal matrices Under diagonal matrices, both of the two DCBP models open three servers (i.e., Servers 4, 5, 6 by DCBP1 and Servers 2, 4, 5 by DCBP2), while the Gaussian and SAA approaches only open Servers 4 and 6. We first use the $10,000$ out-of-sample data points given by misspecified distribution type, namely, the two-point distribution. Table 4 reports each solution's probability of having the total time of assigned appointments not exceeding the capacity of the server to which they are assigned.

Table 4: Solution reliability in out-of-sample data following a misspecified distribution type

Model	Server 2	Server 4	Server 5	Server 6
DCBP1	N/A	1.00	1.00	1.00
DCBP2	1.00	1.00	1.00	N/A
Gaussian	N/A	0.69	N/A	0.91
SAA	N/A	0.69	N/A	0.91

"N/A": the server is not opened by using the corresponding method.

Recall that $\alpha_i = 0.05$ for all i used in all four approaches. The reliability results of the Gaussian and SAA approaches are significantly lower than the desired probability threshold $1 - \alpha_i = 0.95$ on Server 4, and slightly lower than 0.95 on Server 6. On the other hand, the optimal solutions of DCBP1 and DCBP2 do not exceed the capacity of any open servers.

Next, we use the $10,000$ out-of-sample data points given by misspecified moments. Table 5 reports the reliability performance of each optimal solution. The DCBP2 solution still outperforms

Table 5: Solution reliability in out-of-sample scenarios with misspecified moments

Model	Server 2	Server 4	Server 5	Server 6
DCBP1	N/A	0.94	1.00	1.00
DCBP2	0.98	1.00	0.99	N/A
Gaussian	N/A	0.59	N/A	0.89
SAA	N/A	0.59	N/A	0.89

"N/A": the server is not opened by using the corresponding method.

solutions given by all the other approaches and achieves the desired reliability in all the three open servers. The Gaussian and SAA solutions perform poorly when the moment information is different from the empirical inputs. The DCBP1 solution respects the capacities of Servers 5 and 6 with sufficiently high probability (i.e., > 0.95), but yields a slightly lower reliability (0.94) than the threshold on Server 4.

Performance of solutions under general matrices We optimize all the models under general matrices by using the empirical covariance matrices of the in-sample data, and report their corresponding solutions in Table 6. Each entry illustrates the number of appointments assigned to an open server. Note that the Gaussian and SAA approaches yield the same solution of opening servers and assigning appointments.

Table 6: Optimal open servers and appointment-to-server assignments under general matrices

Model	Server 3	Server 4	Server 5	Server 6
DCBP1	12	N/A	13	7
DCBP2	12	11	9	N/A
Gaussian	15	N/A	17	N/A
SAA	15	N/A	17	N/A

"N/A": the server is not opened by using the corresponding method.

We test the solutions shown in Table 6 in the out-of-sample scenarios under misspecified distribution type, and present their reliability performance in Table 7. We again show that under general covariance matrices, the DCBP2 model yields the most conservative solution that does not exceed any open server's capacity, while DCBP1 only ensures the desired reliability on Servers 3 and 6, but not on Server 5. The Gaussian and SAA approaches cannot produce solutions that can achieve the desired reliability threshold on any of their open servers.

Table 7: Solution reliability in out-of-sample scenarios with misspecified moments

Model	Server 3	Server 4	Server 5	Server 6
DCBP1	1.00	N/A	0.91	1.00
DCBP2	1.00	1.00	1.00	N/A
Gaussian	0.91	N/A	0.91	N/A
SAA	0.91	N/A	0.91	N/A

"N/A": the server is not opened by using the corresponding method.

References

Atamtürk, A. and Bhardwaj, A. (2015). Supermodular covering knapsack polytope. *Discrete Optimization*, 18:74–86.

Atamtürk, A. and Bhardwaj, A. (2018). Network design with probabilistic capacities. *Networks*, 71(1):16–30.

Atamtürk, A. and Gómez, A. (2017). Polymatroid inequalities for p-order conic mixed 0-1 optimization. Available at arXiv e-prints `https://arxiv.org/abs/1705.05918`.

Atamtürk, A. and Narayanan, V. (2008). Polymatroids and mean-risk minimization in discrete optimization. *Operations Research Letters*, 36(5):618–622.

Atamtürk, A. and Narayanan, V. (2009). The submodular knapsack polytope. *Discrete Optimization*, 6(4):333–344.

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

Bhardwaj, A. (2015). *Binary conic quadratic knapsacks*. PhD thesis, University of California, Berkeley.

Calafiore, G. C. and El Ghaoui, L. (2006). On distributionally robust chance-constrained linear programs. *Journal of Optimization Theory and Applications*, 130(1):1–22.

Chen, W., Sim, M., Sun, J., and Teo, C. P. (2010). From CVaR to uncertainty set: Implications in joint chance constrained optimization. *Operations Research*, 58(2):470–485.

Cheng, J., Delage, E., and Lisser, A. (2014). Distributionally robust stochastic knapsack problem. *SIAM Journal on Optimization*, 24(3):1485–1506.

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

Deng, Y., Shen, S., and Denton, B. T. (2016). Chance-constrained surgery planning under conditions of limited and ambiguous data. Available at SSRN: `http://dx.doi.org/10.2139/ssrn.2432375`.

Denton, B. T., Miller, A. J., Balasubramanian, H. J., and Huschka, T. R. (2010). Optimal allocation of surgery blocks to operating rooms under uncertainty. *Operations Research*, 58(4):802–816.

Edmonds, J. (1970). Submodular functions, matroids, and certain polyhedra. *Combinatorial Structures and Their Applications*, pages 69–87.

El Ghaoui, L., Oks, M., and Oustry, F. (2003). Worst-case Value-at-Risk and robust portfolio optimization: A conic programming approach. *Operations Research*, 51(4):543–556.

Jiang, R. and Guan, Y. (2016). Data-driven chance constrained stochastic program. *Mathematical Programming, Series A*, 158(1):291–327.

Küçükyavuz, S. (2012). On mixing sets arising in chance-constrained programming. *Mathematical Programming*, 132(1–2):31–56.

Liu, X., Küçükyavuz, S., and Luedtke, J. (2016). Decomposition algorithms for two-stage chance-constrained programs. *Mathematical Programming*, 157(1):219–243.

Luedtke, J. (2014). A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support. *Mathematical Programming*, 146(1-2):219–244.

Luedtke, J. and Ahmed, S. (2008). A sample approximation approach for optimization with probabilistic constraints. *SIAM Journal on Optimization*, 19(2):674–699.

Luedtke, J., Ahmed, S., and Nemhauser, G. (2010). An integer programming approach for linear programs with probabilistic constraints. *Mathematical Programming*, 122(2):247–272.

Nemhauser, G. and Wolsey, L. (1999). *Integer and Combinatorial Optimization*. Wiley.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. (1978). An analysis of approximations for maximizing submodular set functions–I. *Mathematical Programming*, 14(1):265–294.

Pagnoncelli, B., Ahmed, S., and Shapiro, A. (2009). Sample average approximation method for chance constrained programming: Theory and applications. *Journal of Optimization Theory and Applications*, 142(2):399–416.

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

Prékopa, A. (2003). Probabilistic programming. In *Stochastic Programming: Handbooks in Operations Research and Management Science*, volume 10, chapter 5, pages 267–351. Elsevier.

Rujeerapaiboon, N., Kuhn, D., and Wiesemann, W. (2015). Robust growth-optimal portfolios. *Management Science*, 62(7):2090–2109.

Shen, S. and Wang, J. (2014). Stochastic modeling and approaches for managing energy footprints in cloud computing services. *Service Science*, 6(1):15–33.

Shylo, O. V., Prokopyev, O. A., and Schaefer, A. J. (2012). Stochastic operating room scheduling for high-volume specialties under block booking. *INFORMS Journal on Computing*, 25(4):682–692.

Simchi-Levi, D., Chen, X., and Bramel, J. (2014). *The Logic of Logistics: Theory, Algorithms, and Applications for Logistics Management*. Springer Verlag.

Smith, J. E. and Winkler, R. L. (2006). The optimizer's curse: Skepticism and postdecision surprise in decision analysis. *Management Science*, 52(3):311–322.

Song, Y., Luedtke, J. R., and Küçükyavuz, S. (2014). Chance-constrained binary packing problems. *INFORMS Journal on Computing*, 26(4):735–747.

Wagner, M. (2008). Stochastic 0–1 linear programming under limited distributional information. *Operations Research Letters*, 36(2):150–156.

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

Yu, Y.-L., Li, Y., Schuurmans, D., and Szepesvári, C. (2009). A general projection property for distribution families. In *Advances in Neural Information Processing Systems*, pages 2232–2240.

Zymler, S., Kuhn, D., and Rustem, B. (2013). Distributionally robust joint chance constraints with second-order moment information. *Mathematical Programming*, 137(1-2):167–198.