Ambiguous Chance-Constrained Bin Packing under Mean-Covariance Information

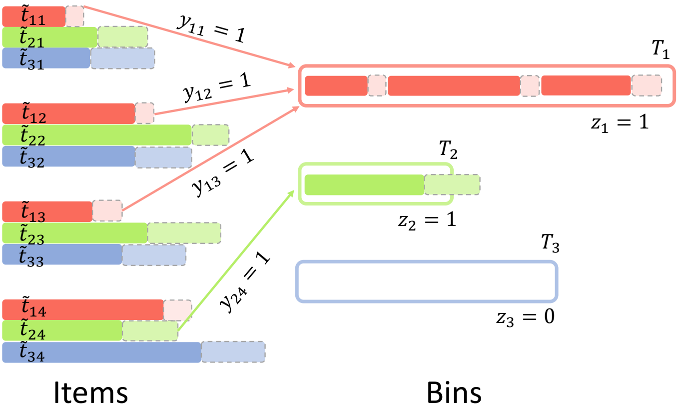
Siqian Shen
Dept. of Industrial and Operations Engineering
University of Michigan at Ann Arbor

Joint work with
Yiling Zhang and Ruiwei Jiang (UMich)
(supported by NSF grants #1662774)

Distributionally Robust Optimization Workshop
BIRS, Canada

Outline

Stochastic Bin Packing



Applications:

- healthcare, cloud computing, airline scheduling...

Problem Setup

Parameter:

- J: set of items
- I: set of bins
- c_i^z: the cost of opening bin i, $\forall i \in I$
- c_{ij}^y: the cost of assigning item j to bin i, $\forall i \in I,\ j \in J$
- T_i: capacity of bin i, $\forall i \in I$
- $\rho_{ij} = 1$ if item j can be assigned to bin i; $= 0$ o.w.
- $\tilde{t}_{ij}$: item j's **random** weight in bin i

Decision Variables:

- $z_i \in \{0, 1\}$: $z_i = 1$ if we open bin i, and $= 0$ if not
- $y_{ij} \in \{0, 1\}$: $y_{ij} = 1$ if item j is assigned to bin i

A Chance-Constrained Formulation

$$\min_{\mathbf{z},\mathbf{y}} \quad \sum_{i \in I} c_i^z z_i + \sum_{i \in I} \sum_{j \in J} c_{ij}^y y_{ij} \tag{1a}$$

$$\text{s.t.} \quad y_{ij} \leq \rho_{ij} z_i \quad \forall i \in I,\ j \in J \tag{1b}$$

$$\sum_{i \in I} y_{ij} = 1 \quad \forall j \in J \tag{1c}$$

$$y_{ij}, z_i \in \{0, 1\} \quad \forall i \in I,\ j \in J \tag{1d}$$

$$\mathbb{P}\left\{ \sum_{j \in J} \tilde{t}_{ij} y_{ij} \leq T_i \right\} \geq 1 - \alpha_i \quad \forall i \in I \tag{1e}$$

- Objective (1a): Minimize the total cost.
- (1b)–(1d): Feasible assignment of items $\rightarrow$ open bins
- (1e): "total weight $\leq$ bin i capacity" at $1 - \alpha_i$ probability

Gaussian Approximation ⇒ 0-1 SOC Reformulation

If $\tilde{t}_i = [\tilde{t}_{ij},\ j \in J]^\mathsf{T}$ follows a Gaussian with known mean μ_i and covariance Σ_i, the chance constraints (1e) are equivalent to (see, Prékopa (2003)):

$$\Phi^{-1}(1-\alpha_i)\sqrt{y_i^\mathsf{T}\Sigma_i y_i} \leq T_i - \mu_i^\mathsf{T} y_i,\ \forall i \in I, \tag{2}$$

where $\Phi(\cdot)$ represents the CDF of the standard normal distribution.

If $\tilde{t}_i$ follows a general distribution, model (1) can be approximated by a second-order cone (SOC) program by replacing (1e) with (2).

Outline

DR Chance-Constrained Bin Packing Model

DCBP:

$$\begin{aligned} \min_{\mathbf{z},\mathbf{y}} \quad & \sum_{i\in I} c_i^z z_i + \sum_{i\in I}\sum_{j\in J} c_{ij}^y y_{ij} \\ \text{s.t.} \quad & \text{(1b)–(1d)}, \\ & \inf_{\mathbb{P}\in\mathcal{D}} \mathbb{P}\left\{\sum_{j\in J} \tilde{t}_{ij} y_{ij} \le T_i\right\} \ge 1-\alpha_i \quad \forall i\in I \qquad (3) \end{aligned}$$

- An accurate and complete estimation of $\mathbb{P}$ is rarely accessible.
- Alternative: a set of plausible candidates of $\mathbb{P}$ (ambiguity set $\mathcal{D}$).
- (3): The **worst-case** probability for any $\mathbb{P}\in\mathcal{D}$ is guaranteed at least $1-\alpha_i$ (an ambiguous chance constraint).

Literature Review

Distributionally robust optimization

- Scarf et al. (1958); Delage and Ye (2010); Bertsimas et al. (2010); Wiesemann et al. (2014); Esfahani and Kuhn (2016)...

Distributionally robust chance-constrained programming

- El Ghaoui et al. (2003); Calafiore and El Ghaoui (2006); Wanger (2008); Zymler et al. (2013); Jiang and Guan (2015)...

Stochastic (chance-constrained) bin packing/knapsack

- Kleinberg, et al. (2000); Goyal and Ravi (2010); Han, et al. (2016); Kosuch and Lisser (2010); Song, et al. (2014)...

Literature Review

Distributionally robust optimization

- Scarf et al. (1958); Delage and Ye (2010); Bertsimas et al. (2010); Wiesemann et al. (2014); Esfahani and Kuhn (2016)...

Distributionally robust chance-constrained programming

- El Ghaoui et al. (2003); Calafiore and El Ghaoui (2006); Wanger (2008); Zymler et al. (2013); Jiang and Guan (2015)...

Stochastic (chance-constrained) bin packing/knapsack

- Kleinberg, et al. (2000); Goyal and Ravi (2010); Han, et al. (2016); Kosuch and Lisser (2010); Song, et al. (2014)...

DR chance-constrained knapsack/bin packing

- most papers use “mean + variance” for constructing $\mathcal{D}$, e.g., Zhang-Denton-Xie (2015)
- Cheng-Delage-Lisser (2014): DR chance-constrained knapsack, mean + covariance, SDP reformulation

Literature Review

Distributionally robust optimization

- Scarf et al. (1958); Delage and Ye (2010); Bertsimas et al. (2010); Wiesemann et al. (2014); Esfahani and Kuhn (2016)...

Distributionally robust chance-constrained programming

- El Ghaoui et al. (2003); Calafiore and El Ghaoui (2006); Wanger (2008); Zymler et al. (2013); Jiang and Guan (2015)...

Stochastic (chance-constrained) bin packing/knapsack

- Kleinberg, et al. (2000); Goyal and Ravi (2010); Han, et al. (2016); Kosuch and Lisser (2010); Song, et al. (2014)...

DR chance-constrained knapsack/bin packing

- most papers use "mean + variance" for constructing $\mathcal{D}$, e.g., Zhang-Denton-Xie (2015)
- Cheng-Delage-Lisser (2014): DR chance-constrained knapsack, mean + covariance, SDP reformulation

Submodularity in (conic) quadratic program

- Atamtürk and Narayanan (2008); Atamtürk-Berenguer-Shen (2012); Atamtürk and Bhardwaj (2016); Atamtürk and Jeon (2017); Nemhauser et al. (1978)

Outline

Two Moment Ambiguity Sets

$\mathcal{M}_+^J$: the set of all probability distributions on $\mathbb{R}^J$.

Case 1: Exactly match the empirical mean and covariance:

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

Two Moment Ambiguity Sets

$\mathcal{M}_+^J$: the set of all probability distributions on $\mathbb{R}^J$.

Case 1: Exactly match the empirical mean and covariance:

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

However, due to estimation error in μ_i and Σ_i, we also consider:
Case 2: A more general ambiguity set (Delage and Ye, 2010):

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

- $\gamma_1 > 0$ and $\gamma_2 > \max\{\gamma_1, 1\}$ are for controlling $\mathcal{D}_2$; can be chosen based on the amount of data and desired confidence level or via cross validation.

Outline

0-1 SOC Representation with $\mathcal{D} = \mathcal{D}_1$

Following Chebyshev's inequality (see, El Ghaoui et al. (2003); Wagner (2008)), the DR chance constraint (3) is equivalent to

$$\sqrt{y_i^\mathsf{T} \Sigma_i y_i} \leq \sqrt{\frac{\alpha_i}{1-\alpha_i}} \left(T_i - \mu_i^\mathsf{T} y_i \right), \ \forall i \in I \tag{4}$$

0-1 SOC Representation with $\mathcal{D} = \mathcal{D}_1$

Following Chebyshev's inequality (see, El Ghaoui et al. (2003); Wagner (2008)), the DR chance constraint (3) is equivalent to

$$\sqrt{y_i^{\mathsf{T}} \Sigma_i y_i} \leq \sqrt{\frac{\alpha_i}{1-\alpha_i}} \left(T_i - \mu_i^{\mathsf{T}} y_i \right), \ \forall i \in I \tag{4}$$

Remarks:

- we can recapture the convexity of chance constraints (3) by employing set $\mathcal{D}_1$ to model the $\tilde{t}_i$ uncertainty.
- The continuous relaxation of DCBP is an SOC program, one of the most computationally tractable nonlinear programs.

0-1 SDP Reformulation with $\mathcal{D} = \mathcal{D}_2$

When $\mathcal{D} = \mathcal{D}_2$, based on existing results of general DR chance constraints (e.g., Zymler et al. (2013), Jiang and Guan (2015))

- DR chance constraints (3) $\Leftrightarrow$ SDP constraints (exact)

Then, DCBP $\Leftrightarrow$ 0-1 SDP reformulation.

However,

- 0-1 SDP cannot be directly solved in solvers

0-1 SDP Reformulation with $\mathcal{D} = \mathcal{D}_2$

When $\mathcal{D} = \mathcal{D}_2$, based on existing results of general DR chance constraints (e.g., Zymler et al. (2013), Jiang and Guan (2015))

- DR chance constraints (3) $\Leftrightarrow$ SDP constraints (exact)

Then, DCBP $\Leftrightarrow$ 0-1 SDP reformulation.

However,

- 0-1 SDP cannot be directly solved in solvers
- Can use a **cutting-plane algorithm** and iteratively generate Benders cuts based on the dual of SDP given fixed y_i

0-1 SDP Reformulation with $\mathcal{D} = \mathcal{D}_2$

When $\mathcal{D} = \mathcal{D}_2$, based on existing results of general DR chance constraints (e.g., Zymler et al. (2013), Jiang and Guan (2015))

- DR chance constraints (3) $\Leftrightarrow$ SDP constraints (exact)

Then, DCBP $\Leftrightarrow$ 0-1 SDP reformulation.

However,

- 0-1 SDP cannot be directly solved in solvers
- Can use a **cutting-plane algorithm** and iteratively generate Benders cuts based on the dual of SDP given fixed y_i
- It takes $\geq$ 6,000 seconds to solve instances with 6 bins and 32 items when $\alpha_i = 0.05$ and $(\gamma_1, \gamma_2) = (1, 2)$

0-1 SDP Reformulation with $\mathcal{D} = \mathcal{D}_2$

When $\mathcal{D} = \mathcal{D}_2$, based on existing results of general DR chance constraints (e.g., Zymler et al. (2013), Jiang and Guan (2015))

- DR chance constraints (3) $\Leftrightarrow$ SDP constraints (exact)

Then, DCBP $\Leftrightarrow$ 0-1 SDP reformulation.

However,

- 0-1 SDP cannot be directly solved in solvers
- Can use a **cutting-plane algorithm** and iteratively generate Benders cuts based on the dual of SDP given fixed y_i
- It takes $\geq$ 6,000 seconds to solve instances with 6 bins and 32 items when $\alpha_i = 0.05$ and $(\gamma_1, \gamma_2) = (1, 2)$

0-1 SDP Reformulation with $\mathcal{D} = \mathcal{D}_2$

When $\mathcal{D} = \mathcal{D}_2$, based on existing results of general DR chance constraints (e.g., Zymler et al. (2013), Jiang and Guan (2015))

- DR chance constraints (3) $\Leftrightarrow$ SDP constraints (exact)

Then, DCBP $\Leftrightarrow$ 0-1 SDP reformulation.

However,

- 0-1 SDP cannot be directly solved in solvers
- Can use a **cutting-plane algorithm** and iteratively generate Benders cuts based on the dual of SDP given fixed y_i
- It takes $\geq$ 6,000 seconds to solve instances with 6 bins and 32 items when $\alpha_i = 0.05$ and $(\gamma_1, \gamma_2) = (1, 2)$

NEXT, we seek

- More tractable reformulation (0-1 SOC program)
- More efficient algorithms (branch-and-cut)

0-1 SOC Reformulation with $\mathcal{D} = \mathcal{D}_2$

Theorem

For each $i \in I$, DRCC (3) with $\mathcal{D} = \mathcal{D}_2$ is equivalent to

$$\mu_i^\top y_i + \left(\sqrt{\gamma_1} + \sqrt{\left(\frac{1-\alpha_i}{\alpha_i}\right)(\gamma_2 - \gamma_1)}\right)\sqrt{y_i^\top \Sigma_i y_i} \ \le \ T_i \tag{5a}$$

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

$$\mu_i^\top y_i + \sqrt{\frac{\gamma_2}{\alpha_i}}\sqrt{y_i^\top \Sigma_i y_i} \ \le \ T_i \tag{5b}$$

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

0-1 SOC Reformulation with $\mathcal{D} = \mathcal{D}_2$

Theorem

For each $i \in I$, DRCC (3) with $\mathcal{D} = \mathcal{D}_2$ is equivalent to

$$\mu_i^\top y_i + \left(\sqrt{\gamma_1} + \sqrt{\left(\frac{1-\alpha_i}{\alpha_i}\right)(\gamma_2 - \gamma_1)}\right)\sqrt{y_i^\top \Sigma_i y_i} \;\leq\; T_i \tag{5a}$$

if $\gamma_1/\gamma_2 \leq \alpha_i$, and is equivalent to

$$\mu_i^\top y_i + \sqrt{\frac{\gamma_2}{\alpha_i}}\sqrt{y_i^\top \Sigma_i y_i} \;\leq\; T_i \tag{5b}$$

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

Remarks:

- The result holds for general covariance matrices. (Σ_i)
- Both (5a) and (5b) are SOC constraints with different coefficients, dependent on values of $\gamma_1, \gamma_2, \alpha_i$.

Proof Sketch

The theorem was proved in two steps:

- **[Step 1]:** Project the random vector $\tilde{t}_i$ and its ambiguity set $\mathcal{D}_2$ from $\mathbb{R}^J$ to the real line, i.e., $\mathbb{R}$.

Proof Sketch

The theorem was proved in two steps:

- **[Step 1]:** Project the random vector $\tilde{t}_i$ 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.

Proof Sketch

The theorem was proved in two steps:

- **[Step 1]:** Project the random vector $\tilde{t}_i$ 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.
- **[Step 2]:** Derive worst-case mean and covariance matrix in $\mathcal{D}_2$ that attain the worst-case probability bound in (3). Then apply Cantelli's inequality to conclude the SOC representation.

Proof Sketch

The theorem was proved in two steps:

- **[Step 1]:** Project the random vector $\tilde{t}_i$ 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.
- **[Step 2]:** Derive worst-case mean and covariance matrix in $\mathcal{D}_2$ that attain the worst-case probability bound in (3). Then apply Cantelli's inequality to conclude the SOC representation.
- Let $\tilde{s}_i = \tilde{t}_i - \mu_i$, $\tilde{\xi}_i = y_i^\top \tilde{s}_i$, $b_i = T_i - \mu_i^\top y_i$.

$$\begin{aligned}
\inf_{\mathbb{P}\in\mathcal{D}_2} \mathbb{P}\{\tilde{t}_i^\top y_i \le T_i\} &= \inf_{\mathbb{P}\in\mathcal{D}_{\tilde{s}_i}} \mathbb{P}\{y_i^\top \tilde{s}_i \le b_i\} \\
&= \inf_{\mathbb{P}\in\mathcal{D}_{\tilde{\xi}_i}} \mathbb{P}\{\tilde{\xi}_i \le b_i\} \\
&= \inf_{(\mu_1,\sigma_1)\in S} \inf_{\mathbb{P}\in\mathcal{D}_1(\mu_1,\sigma_1^2)} \mathbb{P}\{\tilde{\xi}_i \le b_i\}
\end{aligned}$$

Proof Sketch (Continued)

For simplicity, we omit index i for the rest of the proof.

- The above equality 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.

Proof Sketch (Continued)

For simplicity, we omit index i for the rest of the proof.

- The above equality 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

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

Proof Sketch (Continued)

For simplicity, we omit index i for the rest of the proof.

- The above equality 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 states that $\inf_{\mathbb{P} \in \mathcal{D}_2} \mathbb{P}\{\tilde{t}^\top y \leq T\} \geq 1 - \alpha > 0$, we can assume $b \geq \mu_1$ for all $(\mu_1, \sigma_1) \in S$ w.l.o.g.

Proof Sketch (Continued)

- It follows that

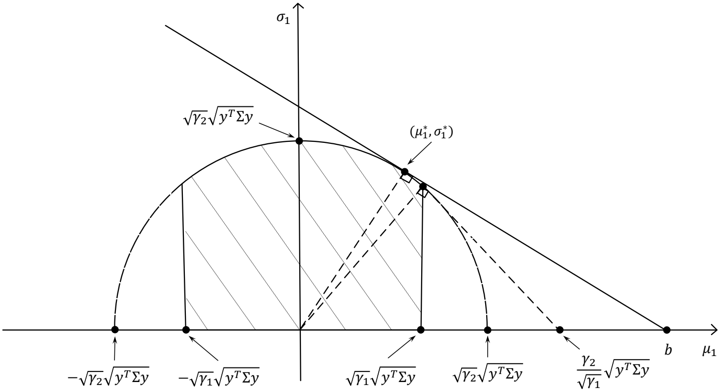
$$\inf_{\mathbb{P}\in\mathcal{D}_2} \mathbb{P}\{\tilde{t}^\top y \leq T\} = \inf_{(\mu_1,\sigma_1)\in S} \frac{(b-\mu_1)^2}{\sigma_1^2 + (b-\mu_1)^2} = \inf_{(\mu_1,\sigma_1)\in S} \frac{1}{\left(\frac{\sigma_1}{b-\mu_1}\right)^2 + 1}.$$

Proof Sketch (Continued)

- It follows that

$$\inf_{\mathbb{P}\in\mathcal{D}_2} \mathbb{P}\{\tilde{t}^\top y \leq T\} = \inf_{(\mu_1,\sigma_1)\in S} \frac{(b-\mu_1)^2}{\sigma_1^2+(b-\mu_1)^2} = \inf_{(\mu_1,\sigma_1)\in S} \frac{1}{\left(\frac{\sigma_1}{b-\mu_1}\right)^2+1}.$$

- The objective function decreases as $\sigma_1/(b-\mu_1)$ increases. Hence, equivalently, we solve $\inf_{(\mu_1,\sigma_1)\in S} -\left(\frac{\sigma_1}{b-\mu_1}\right)$.



0-1 SOC Formulation Summary

The DRCC (3) can be reformulated by three types of 0-1 SOC constraints in the same form of:

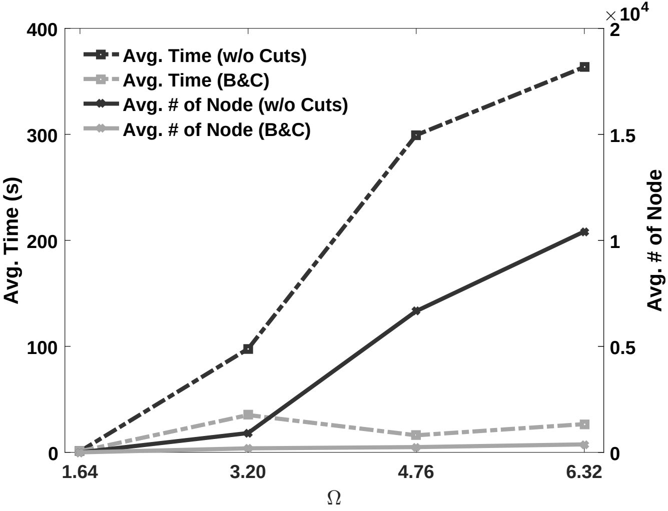
$$(\mu_i)^\mathsf{T} y_i + \Omega_i \sqrt{y_i^\mathsf{T} \Sigma_i y_i} \le T_i,\ \Omega_i \ge 0$$

- Guassian: $\Omega_i = \Phi^{-1}(1-\alpha_i)$
- DCBP1 with $\mathcal{D} = \mathcal{D}_1$: $\Omega_i = \sqrt{(1-\alpha_i)/\alpha_i}$
- DCBP2 with $\mathcal{D} = \mathcal{D}_2$:

$$\Omega_i = \begin{cases} \left(\sqrt{\gamma_1} + \sqrt{(1-\alpha_i)(\gamma_2-\gamma_1)/\alpha_i}\right), & \gamma_1/\gamma_2 \le \alpha_i \\ \sqrt{\gamma_2/\alpha_i}, & \gamma_2/\gamma_2 \ge \alpha_i \end{cases}$$

CPU Time of 0-1 SOC under different Ω_i

We test $I = 6$ and $J = 32$, $\alpha_i = 0.05$, $\forall i \in I$, and vary Ω_i, $\forall i$ in between $[1.64, 6.32]$. ($\Omega_i = 1.64$ for "Gaussian" and $\Omega_i = 6.32$ for "DCBP2".)



Outline

Special Case: When Σ_i is Diagonal

Goal: solve 0-1 SOC constraint:

$$(\mu_i)^\mathsf{T} y_i + \Omega_i \sqrt{y_i^\mathsf{T} \Sigma_i y_i} \leq T_i, \;\; y_i \in \{0, 1\}.$$

- Denote $g_i(y_i) = (\mu_i)^\mathsf{T} y_i + \Omega_i \sqrt{c_i^\mathsf{T} y_i}$, where $c_i \geq \mathbf{0}$ consists of the diagonal term in Σ_i.

Special Case: When Σ_i is Diagonal

Goal: solve 0-1 SOC constraint:

$$(\mu_i)^\mathsf{T} y_i + \Omega_i \sqrt{y_i^\mathsf{T} \Sigma_i y_i} \leq T_i,\ \ y_i \in \{0, 1\}.$$

- Denote $g_i(y_i) = (\mu_i)^\mathsf{T} y_i + \Omega_i \sqrt{c_i^\mathsf{T} y_i}$, where $c_i \geq \mathbf{0}$ consists of the diagonal term in Σ_i.
- $g_i(y_i)$ submodular: if $f(S) + f(T) \geq f(S \cup T) + f(S \cap T)$ for all $S, T \subseteq N$, where $N = \{1, \ldots, n\}$

Special Case: When Σ_i is Diagonal

Goal: solve 0-1 SOC constraint:

$$(\mu_i)^\mathsf{T} y_i + \Omega_i \sqrt{y_i^\mathsf{T} \Sigma_i y_i} \le T_i,\ \ y_i \in \{0,1\}.$$

- Denote $g_i(y_i) = (\mu_i)^\mathsf{T} y_i + \Omega_i \sqrt{c_i^\mathsf{T} y_i}$, where $c_i \ge \mathbf{0}$ consists of the diagonal term in Σ_i.
- $g_i(y_i)$ submodular: if $f(S) + f(T) \ge f(S \cup T) + f(S \cap T)$ for all $S, T \subseteq N$, where $N = \{1, \ldots, n\}$
- Extended polymatroid inequality for $g_i(y_i) \le T_i$ (Atamtürk and Narayanan (2008)):

$$\pi_i^\mathsf{T} y_i \le T_i$$

Special Case: When Σ_i is Diagonal

Goal: solve 0-1 SOC constraint:

$$(\mu_i)^\mathsf{T} y_i + \Omega_i \sqrt{y_i^\mathsf{T} \Sigma_i y_i} \le T_i,\ \ y_i \in \{0,1\}.$$

- Denote $g_i(y_i) = (\mu_i)^\mathsf{T} y_i + \Omega_i \sqrt{c_i^\mathsf{T} y_i}$, where $c_i \ge \mathbf{0}$ consists of the diagonal term in Σ_i.
- $g_i(y_i)$ submodular: if $f(S) + f(T) \ge f(S \cup T) + f(S \cap T)$ for all $S, T \subseteq N$, where $N = \{1, \ldots, n\}$
- Extended polymatroid inequality for $g_i(y_i) \le T_i$ (Atamtürk and Narayanan (2008)):

$$\pi_i^\mathsf{T} y_i \le T_i$$

- **Branch-and-cut (B&C)**: $\pi_i^* = \arg\max_{\pi_i \in EP_{g_i}} \pi_i^\mathsf{T} y_i$ can be found efficiently by the greedy algorithm (Edmonds, 1971).

Outline

Non-diagonal Σ_i: Not Submodular

Example: Suppose that $\mathcal{J} = \{1, 2, 3\}$, $\mu = 0$, 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 $g(y) = \mu^\top y + \sqrt{y^\top \Lambda y}$ (with $\Lambda = \Omega^2 \Sigma$) is not submodular because $h(R \cup \{j\}) - h(R) < h(S \cup \{j\}) - h(S)$, where $R = \{1\}$, $S = \{1, 2\}$, and $j = 3$.

(Necessary) and Sufficient Conditions for Submodularity

We take out the index i for all the variables and parameters for presentation clarity.

Theorem

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 $\Lambda_{rs} \leq 0$ for all $r, s = 1, \ldots, J$ and $r \neq s$.

Proposition

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

Relaxed Submodular Approximation

The 0-1 SOC constraint $g(y) \leq T$ implies another SOC constraint

$$\mu^\top y + \sqrt{y^\top \Delta^{\mathsf{L}} y} \;\leq\; T, \tag{6}$$

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

$$\min_{\Delta} \;\; ||\Delta - \Lambda||_2 \tag{7a}$$

$$\text{s.t.} \;\; 0 \;\preceq\; \Delta \;\preceq\; \Lambda, \tag{7b}$$

$$2\sum_{s=1}^{J} \Delta_{rs} \geq \Delta_{rr}, \quad \forall r = 1, \ldots, J, \tag{7c}$$

$$\Delta_{rs} \leq 0, \quad \forall r, s = 1, \ldots, J \text{ and } r \neq s. \tag{7d}$$

The extended polymatroid cuts for (6) are valid for DCBP.

Conservative Submodular Approximation

Additionally, $g(y) = \mu^\top y + \sqrt{y^\top \Lambda y} \leq T$ is implied by

$$\mu^\top y + \sqrt{y^\top \Delta^{\cup} y} \;\leq\; T, \tag{8}$$

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

$$\min_{\Delta} \;\; ||\Delta - \Lambda||_2 \tag{9a}$$

$$\text{s.t.} \;\; \Delta \;\succeq\; \Lambda, \;\; \text{(7c)–(7d)}. \tag{9b}$$

Conservative Submodular Approximation

Additionally, $g(y) = \mu^\top y + \sqrt{y^\top \Lambda y} \leq T$ is implied by

$$\mu^\top y + \sqrt{y^\top \Delta^{\mathsf{U}} y} \ \leq \ T, \tag{8}$$

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

$$\min_{\Delta} \ ||\Delta - \Lambda||_2 \tag{9a}$$

$$\text{s.t.} \ \ \Delta \ \succeq \ \Lambda, \ \ \text{(7c)–(7d)}. \tag{9b}$$

- The results hold for general 0-1 SOC constraints. We can apply relaxed and conservative submodular approximations (6) and (8) to obtain valid bounds on any 0-1 SOC programs, e.g., the knapsack problem with DRCCs.

Outline

Submodularity through Lifting

We show that the submodularity of $g(y) = \mu^\top y + \sqrt{y^\top \Lambda y} \le T$ holds for general Λ in a lifted (i.e., higher-dimensional) space by

- defining $w_{jk} = y_j y_k$ for all $j, k = 1, \ldots, J$ and augment vector y to vector $v = [y_1, \ldots, y_J, w_{11}, \ldots, w_{1J}, w_{21}, \ldots, w_{JJ}]^\top$;
- reformulating $g(y) \le T$ as $y^\top(\Lambda - \mu\mu^\top)y + 2T\mu^\top y \le T^2$;
- decomposing $(\Lambda - \mu\mu^\top)$ to be the sum of two matrices, one containing all positive entries and the other all nonpositive;

Accordingly, we derive extended polymatroid inequalities $\pi^\top v \le T^2$ with $v = (y, w)$ in the lifted space + McCormick Inequalties to linearize w-variables.

Valid Inequalities in the Lifted Space

Consider the feasible region of DCBP in the lifted space. We prove the following valid inequalities:

$$w_{ijk} \geq y_{ij} + y_{ik} + \sum_{\substack{\ell=1 \\ \ell \neq i}}^{I} w_{\ell jk} - 1 \quad \forall j, k = 1, \ldots, J \tag{10a}$$

$$w_{ijk} \geq y_{ij} + y_{ik} - z_i \quad \forall i = 1, \ldots, I, \ \forall j, k = 1, \ldots, J \tag{10b}$$

$$\sum_{\substack{j=1 \\ j \neq k}}^{J} w_{ijk} \leq \sum_{j=1}^{J} y_{ij} - z_i \quad \forall i = 1, \ldots, I, \ \forall k = 1, \ldots, J \tag{10c}$$

$$\sum_{j=1}^{J} \sum_{k=j+1}^{J} w_{ijk} \geq \sum_{j=1}^{J} y_{ij} - z_i \quad \forall i = 1, \ldots, I. \tag{10d}$$

- The above valid inequalities are *polynomially many* and all coefficients are in *closed-form*. We do not need any separation processes for these inequalities.

Outline

Approaches and Computer Setting

Compute the three 0-1 SOC reformulations:

- Gaussian - Gaussian distributed uncertainty assumption
- DCBP1 -DR model with $\mathcal{D}_1$
- DCBP2 -DR model with $\mathcal{D}_2$, $(\gamma_1, \gamma_2) = (1, 2)$

Appointment allocation setting:

- default: $|I| = 6$ servers, $|J| = 32$ appointments
- $T_i \in [420, 540]$ minutes (7-9 hours)
- $c_i^{\mathrm{z}} = T_i^2/3600 + 3T_i/60$, c_{ij}^{y} vary in between $[0, 18]$

Computer setup:

- GUROBI 5.6.3 in Python 2.7; Windows 7 machine with Intel(R) Core(TM) i7-2600 CPU 3.40 GHz; 8GB memory.
- Cuts added using GUROBI `callback` class `Model.cbCut()`.
- Cuts generated at each branch-and-bound node, for both integer and fractional temp solutions.
- Optimality gap tolerance = threshold for violated cuts = 10^{-4}
- Time limit = 3600 seconds.

Instance Design

In-sample data:

- At each server $i \in I$, $\tilde{t}_{ij}$ follows Gaussian:
  high mean: 25 min, low mean: 12.5 min;
  high variance: std/mean = 1.0, low variance: std/mean = 0.3
- Mix for $j \in J$ (8 hMhV, 8 hMℓV, 8 ℓMℓV, 8 ℓMhV)
- Sample size = 10,000 data points

Instance Design

In-sample data:

- At each server $i \in I$, $\tilde{t}_{ij}$ follows Gaussian:
  high mean: 25 min, low mean: 12.5 min;
  high variance: std/mean $= 1.0$, low variance: std/mean $= 0.3$
- Mix for $j \in J$ (8 hMhV, 8 hMℓV, 8 ℓMℓV, 8 ℓMhV)
- Sample size $=$ 10,000 data points

Moments ambiguity

- The same distribution type but pure hMhV appointments.
- out-of-sample size $=$ 10,000 data points

Distribution type ambiguity

- $\tilde{t}_{ij}$ (Two-point distribution; long-tail):
  $= \mu_{ij} + \frac{(1-p)}{\sqrt{p(1-p)}}\sigma_{ij}$ with probability $p = 0.3$
  $= \mu_{ij} - \frac{\sqrt{p(1-p)}}{(1-p)}\sigma_{ij}$ min with probability $1 - p = 0.7$
- out-of-sample size $=$ 10,000 data points

Outline

Computing 0-1 SOC models with diagonal covariance

Table: Time and solution details for instances with diagonal covariance 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
	Guassian	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		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

Computing 0-1 SOC models with general covariance

Table: CPU time of DCBP2 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

Computing 0-1 SOC models under different problem sizes

Table: CPU time of DCBP2 with general covariance and different sizes

	Method	Inst.	$J = 32$						$J = 40$			
			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

Out-of-Sample Performance I

- Test optimal solutions y_i of Gaussian, DCBP1, DCBP2, and MILP in various out-of-sample instances.
- Reliability of each open server $i =$

$$\frac{\#\text{ of scenarios in which } \tilde{t}_i^{\mathsf{T}} y_i \leq T_i}{N = 10,000}$$

Table: Solution reliability results in simulation sample with misspecified moments (all hMℓV instances)

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

Out-of-Sample Performance II

Table: Solution reliability results in simulation sample with misspecified distribution (two-point distribution)

Model	Server 2	Server 4	Server 5	Server 6
DCBP1	N/A	0.96	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

Conclusions

We investigate

- 0-1 SOC representations of DCBP with cross moments
- fast branch-and-cut (BAC) algorithm for 0-1 SOC models with general covariance matrices using bounds and valid cuts
- the BAC algorithm with extended polymatroid inequalities in the original space scales very well as the problem size grows.

Conclusions

We investigate

- 0-1 SOC representations of DCBP with cross moments
- fast branch-and-cut (BAC) algorithm for 0-1 SOC models with general covariance matrices using bounds and valid cuts
- the BAC algorithm with extended polymatroid inequalities in the original space scales very well as the problem size grows.

Future research

- other applications, e.g., appointment scheduling, production planning, and power system operation.
- under other types of ambiguity sets (moment or density based)
- connections between SOCP, SDP, and submodular optimization.

Conclusions

We investigate

- 0-1 SOC representations of DCBP with cross moments
- fast branch-and-cut (BAC) algorithm for 0-1 SOC models with general covariance matrices using bounds and valid cuts
- the BAC algorithm with extended polymatroid inequalities in the original space scales very well as the problem size grows.

Future research

- other applications, e.g., appointment scheduling, production planning, and power system operation.
- under other types of ambiguity sets (moment or density based)
- connections between SOCP, SDP, and submodular optimization.

Thank you! Any questions?