

Distributionally Robust Optimization and its Tractable Approximations with ROME

Joel Goh Melvyn Sim

Department of Decision Sciences
NUS Business School, Singapore

26 Aug 2009
20th ISMP, Chicago

Outline

- 1 Introduction
- 2 Framework
- 3 Segregated LDR
- 4 Bi-Deflected LDR
- 5 ROME
- 6 Conclusion

Outline

- 1 Introduction
- 2 Framework
- 3 Segregated LDR
- 4 Bi-Deflected LDR
- 5 ROME
- 6 Conclusion

Distributionally Robust Optimization

- An approach toward optimization under uncertainty
 - Partially characterized distribution
 - Historically called minimax stochastic programming (Žáčková, 1966)
 - Motivation: in practice we will never know the actual uncertainty distribution

Distributionally Robust Optimization

- An approach toward optimization under uncertainty
 - Partially characterized distribution
 - Historically called minimax stochastic programming (Žáčková, 1966)
 - Motivation: in practice we will never know the actual uncertainty distribution
- Contrast with classical Robust Optimization (RO)
 - Uncertainties in RO characterized by uncertainty set (support)
 - Ben-Tal and Nemirovski (1998), Bertsimas and Sim (2004)

Research Objectives

- Construct an approximate solution framework for a class of DRO problems. Features:
 - Multi-stage decisions
 - Distribution-free, but partially characterized by support, moments, and directional deviations (Chen, Sim, and Sun, 2007)
 - More flexible decision rules compared to AARC (Ben-Tal, Goryashko, Guslitzer, and Nemirovski, 2004), aka LDRs.

Research Objectives

- Construct an approximate solution framework for a class of DRO problems. Features:
 - Multi-stage decisions
 - Distribution-free, but partially characterized by support, moments, and directional deviations (Chen, Sim, and Sun, 2007)
 - More flexible decision rules compared to AARC (Ben-Tal, Goryashko, Guslitzer, and Nemirovski, 2004), aka LDRs.
- To build a software modeling tool to solve DRO problems within this framework (ROME)

Outline

- 1 Introduction
- 2 Framework**
- 3 Segregated LDR
- 4 Bi-Deflected LDR
- 5 ROME
- 6 Conclusion

Notation

- $\tilde{z} \in \mathcal{R}^N$: uncertainties with distribution $\mathbb{P} \in \mathbb{F}$.
- $\mathbf{x} \in \mathcal{R}^n$: “here-and-now” decision variables.
- $\mathbf{y}^k(\tilde{z}) \in \mathcal{R}^{m_k}$: “wait-and-see” decision rules, $\forall k \in [K]$.

Notation

- $\tilde{z} \in \mathbb{R}^N$: uncertainties with distribution $\mathbb{P} \in \mathbb{F}$.
- $\mathbf{x} \in \mathbb{R}^n$: “here-and-now” decision variables.
- $\mathbf{y}^k(\tilde{z}) \in \mathbb{R}^{m_k}$: “wait-and-see” decision rules, $\forall k \in [K]$.
- Each decision rule need not depend on the full uncertainty vector. We introduce K information index sets, to capture the dependency structure, denoted by $I_k \subseteq [N], \forall k \in [K]$.

$$\mathbf{y}^k \in \mathcal{Y}(m_k, N, I_k) \quad \forall k \in [K]$$

$$\mathcal{Y}(m, N, I) \triangleq \left\{ \mathbf{f} : \mathbb{R}^N \rightarrow \mathbb{R}^m : \mathbf{f} \left(\mathbf{z} + \sum_{i \notin I} \lambda_i \mathbf{e}^i \right) = \mathbf{f}(\mathbf{z}), \forall \boldsymbol{\lambda} \in \mathbb{R}^N \right\}$$

Notation

- $\tilde{z} \in \mathcal{R}^N$: uncertainties with distribution $\mathbb{P} \in \mathbb{F}$.
- $\mathbf{x} \in \mathcal{R}^n$: “here-and-now” decision variables.
- $\mathbf{y}^k(\tilde{z}) \in \mathcal{R}^{m_k}$: “wait-and-see” decision rules, $\forall k \in [K]$.
- Each decision rule need not depend on the full uncertainty vector. We introduce K information index sets, to capture the dependency structure, denoted by $I_k \subseteq [N], \forall k \in [K]$.

$$\mathbf{y}^k \in \mathcal{Y}(m_k, N, I_k) \quad \forall k \in [K]$$

$$\mathcal{Y}(m, N, I) \triangleq \left\{ \mathbf{f} : \mathcal{R}^N \rightarrow \mathcal{R}^m : \mathbf{f} \left(\mathbf{z} + \sum_{i \notin I} \lambda_i \mathbf{e}^i \right) = \mathbf{f}(\mathbf{z}), \forall \boldsymbol{\lambda} \in \mathcal{R}^N \right\}$$

- E.g. $I_1 = \{2, 3\} \implies \mathbf{y}^1 \in \mathcal{Y}(m, N, I_1) \implies \mathbf{y}^1(\tilde{z}) = \mathbf{y}^1(\tilde{z}_2, \tilde{z}_3)$

Notation

- $\tilde{z} \in \mathcal{R}^N$: uncertainties with distribution $\mathbb{P} \in \mathbb{F}$.
- $\mathbf{x} \in \mathcal{R}^n$: “here-and-now” decision variables.
- $\mathbf{y}^k(\tilde{z}) \in \mathcal{R}^{m_k}$: “wait-and-see” decision rules, $\forall k \in [K]$.
- Each decision rule need not depend on the full uncertainty vector. We introduce K information index sets, to capture the dependency structure, denoted by $I_k \subseteq [N], \forall k \in [K]$.

$$\mathbf{y}^k \in \mathcal{Y}(m_k, N, I_k) \quad \forall k \in [K]$$

$$\mathcal{Y}(m, N, I) \triangleq \left\{ \mathbf{f} : \mathcal{R}^N \rightarrow \mathcal{R}^m : \mathbf{f} \left(\mathbf{z} + \sum_{i \notin I} \lambda_i \mathbf{e}^i \right) = \mathbf{f}(\mathbf{z}), \forall \boldsymbol{\lambda} \in \mathcal{R}^N \right\}$$

- E.g. $I_1 = \{2, 3\} \implies \mathbf{y}^1 \in \mathcal{Y}(m, N, I_1) \implies \mathbf{y}^1(\tilde{z}) = \mathbf{y}^1(\tilde{z}_2, \tilde{z}_3)$
- For multi-stage decisions with progressive information revelation, $I_1 \subseteq I_2 \subseteq \dots \subseteq I_K$

Model of Uncertainty

- Actual uncertainty distribution \mathbb{P} lies in a family of distributions \mathbb{F}

Model of Uncertainty

- Actual uncertainty distribution \mathbb{P} lies in a family of distributions \mathbb{F}
- Partially characterized by distributional properties:
 - Support \mathcal{W} : Tractable conic representable, full-dim.
 - Mean Support $\hat{\mathcal{W}}$: Tractable conic representable.
 - Covariance matrix Σ : Positive definite
 - Upper bounds on Directional Deviations for stochastically independent components $(\mathbf{H}_\sigma, \sigma_f, \sigma_b)$.

Formulation of General Model

$$\begin{aligned} Z_{GEN}^* = & \\ \min_{\mathbf{x}, \{\mathbf{y}^k(\cdot)\}_{k=1}^K} & \mathbf{c}^{0'} \mathbf{x} + \sup_{\mathbb{P} \in \mathbb{F}} \mathbb{E}_{\mathbb{P}} \left(\sum_{k=1}^K \mathbf{d}^{0,k'} \mathbf{y}^k(\tilde{\mathbf{z}}) \right) \\ \text{s.t.} & \mathbf{c}^{l'} \mathbf{x} + \sup_{\mathbb{P} \in \mathbb{F}} \mathbb{E}_{\mathbb{P}} \left(\sum_{k=1}^K \mathbf{d}^{l,k'} \mathbf{y}^k(\tilde{\mathbf{z}}) \right) \leq b_l \quad \forall l \in [M] \\ & \mathbf{T}(\tilde{\mathbf{z}}) \mathbf{x} + \sum_{k=1}^K \mathbf{U}^k \mathbf{y}^k(\tilde{\mathbf{z}}) = \mathbf{v}(\tilde{\mathbf{z}}) \\ & \underline{\mathbf{y}}^k \leq \mathbf{y}^k(\tilde{\mathbf{z}}) \leq \bar{\mathbf{y}}^k \quad \forall k \in [K] \\ & \mathbf{x} \geq \mathbf{0} \\ & \mathbf{y}^k \in \mathcal{Y}(m_k, N, I_k) \quad \forall k \in [K] \end{aligned}$$

Linear Approximation of General Model

- General model is intractable.
 - How to choose recourse decisions?

Linear Approximation of General Model

- General model is intractable.
 - How to choose recourse decisions?
- Following Ben-Tal et. al. (2004), we restrict ourselves to LDRs (or AARCs), where the recourse decisions are affinely dependent on the model uncertainties.

Linear Approximation of General Model

- General model is intractable.
 - How to choose recourse decisions?
- Following Ben-Tal et. al. (2004), we restrict ourselves to LDRs (or AARCs), where the recourse decisions are affinely dependent on the model uncertainties.
- Define the space of LDRs

$$\mathcal{L}(m, N, I) \triangleq \left\{ \mathbf{f} : \mathbb{R}^N \rightarrow \mathbb{R}^m : \exists (\mathbf{y}^0, \mathbf{Y}) \in \mathbb{R}^m \times \mathbb{R}^{m \times N} : \begin{array}{l} \mathbf{f}(\mathbf{z}) = \mathbf{y}^0 + \mathbf{Y}\mathbf{z} \\ \mathbf{Y}\mathbf{e}^i = \mathbf{0}, \forall i \notin I \end{array} \right\}$$

Linear Approximation of General Model

- General model is intractable.
 - How to choose recourse decisions?
- Following Ben-Tal et. al. (2004), we restrict ourselves to LDRs (or AARCs), where the recourse decisions are affinely dependent on the model uncertainties.
- Define the space of LDRs

$$\mathcal{L}(m, N, I) \triangleq \left\{ \mathbf{f} : \mathbb{R}^N \rightarrow \mathbb{R}^m : \exists (\mathbf{y}^0, \mathbf{Y}) \in \mathbb{R}^m \times \mathbb{R}^{m \times N} : \begin{array}{l} \mathbf{f}(\mathbf{z}) = \mathbf{y}^0 + \mathbf{Y}\mathbf{z} \\ \mathbf{Y}\mathbf{e}^i = \mathbf{0}, \forall i \notin I \end{array} \right\}$$

- Use this to approximate $\mathcal{Y}(m, N, I)$.

LDR Approximation of General Model

$$\begin{aligned}
 Z_{LDR} = & \\
 \min_{\mathbf{x}, \{\mathbf{y}^k(\cdot)\}_{k=1}^K} & \quad \mathbf{c}^{0'} \mathbf{x} + \sup_{\mathbb{P} \in \mathbb{F}} \mathbb{E}_{\mathbb{P}} \left(\sum_{k=1}^K \mathbf{d}^{0,k'} \mathbf{y}^k(\tilde{\mathbf{z}}) \right) \\
 \text{s.t.} & \quad \mathbf{c}^{l'} \mathbf{x} + \sup_{\mathbb{P} \in \mathbb{F}} \mathbb{E}_{\mathbb{P}} \left(\sum_{k=1}^K \mathbf{d}^{l,k'} \mathbf{y}^k(\tilde{\mathbf{z}}) \right) \leq b_l \quad \forall l \in [M] \\
 & \quad T(\tilde{\mathbf{z}}) \mathbf{x} + \sum_{k=1}^K U^k \mathbf{y}^k(\tilde{\mathbf{z}}) = \mathbf{v}(\tilde{\mathbf{z}}) \\
 & \quad \underline{\mathbf{y}}^k \leq \mathbf{y}^k(\tilde{\mathbf{z}}) \leq \bar{\mathbf{y}}^k \quad \forall k \in [K] \\
 & \quad \mathbf{x} \geq \mathbf{0} \\
 & \quad \mathbf{y}^k \in \mathcal{L}(m_k, N, I_k) \quad \forall k \in [K]
 \end{aligned}$$

LDR Approx. of General Model (Explicit)

$$\begin{aligned}
 Z_{LDR} = & \\
 \min_{\mathbf{x}, \{\mathbf{y}^{0,k}, \mathbf{Y}^k\}_{k=1}^K} & \mathbf{c}^{0'} \mathbf{x} + \sum_{k=1}^K \mathbf{d}^{0,k'} \mathbf{y}^{0,k} + \sup_{\hat{\mathbf{z}} \in \hat{\mathcal{W}}} \left(\sum_{k=1}^K \mathbf{d}^{0,k'} \mathbf{Y}^k \hat{\mathbf{z}} \right) \\
 \text{s.t.} & \mathbf{c}^{l'} \mathbf{x} + \sum_{k=1}^K \mathbf{d}^{l,k'} \mathbf{y}^{0,k} + \sup_{\hat{\mathbf{z}} \in \hat{\mathcal{W}}} \left(\sum_{k=1}^K \mathbf{d}^{l,k'} \mathbf{Y}^k \hat{\mathbf{z}} \right) \leq b_l \quad \forall l \in [M] \\
 & \mathbf{T}^0 \mathbf{x} + \sum_{k=1}^K \mathbf{U}^k \mathbf{y}^{0,k} = \mathbf{v}^0 \\
 & \mathbf{T}^j \mathbf{x} + \sum_{k=1}^K \mathbf{U}^k \mathbf{Y}^k \mathbf{e}^j = \mathbf{v}^j \quad \forall j \in [N] \\
 & \underline{\mathbf{y}}^k \leq \mathbf{y}^{0,k} + \mathbf{Y}^k \mathbf{z} \leq \bar{\mathbf{y}}^k \quad \forall \mathbf{z} \in \mathcal{W} \quad \forall k \in [K] \\
 & \bar{\mathbf{Y}}^k \mathbf{e}^j = \mathbf{0} \quad \forall j \notin I_k, \forall k \in [K] \\
 & \mathbf{x} \geq \mathbf{0}
 \end{aligned}$$

Is LDR Approximation too Conservative?

- LDR approximation of General Model is tractable, but is it too conservative?

Is LDR Approximation too Conservative?

- LDR approximation of General Model is tractable, but is it too conservative?
- We will proceed by using the basic LDR approximation as a starting point, and aim to find other more complex decision rules which are better.

$$Z_{GEN}^* \leq ??? \leq ??? \leq Z_{LDR}$$

Outline

- 1 Introduction
- 2 Framework
- 3 Segregated LDR**
- 4 Bi-Deflected LDR
- 5 ROME
- 6 Conclusion

Segregating the Uncertainties

- Idea: to re-map the **uncertainties** into a higher-dimensional space, and apply an LDR on the new uncertainties, for more flexibility.

Segregating the Uncertainties

- Idea: to re-map the **uncertainties** into a higher-dimensional space, and apply an LDR on the new uncertainties, for more flexibility.
- E.g. for a scalar uncertainty, we can segregate into positive and negative parts (as was done by Chen, Sim, Sun, and Zhang, 2008).

$$\tilde{z} = \underbrace{\tilde{z}^+}_{\tilde{\zeta}_1} - \underbrace{\tilde{z}^-}_{\tilde{\zeta}_2}$$

Segregating the Uncertainties

- Idea: to re-map the **uncertainties** into a higher-dimensional space, and apply an LDR on the new uncertainties, for more flexibility.
- E.g. for a scalar uncertainty, we can segregate into positive and negative parts (as was done by Chen, Sim, Sun, and Zhang, 2008).

$$\tilde{z} = \underbrace{\tilde{z}^+}_{\tilde{\zeta}_1} - \underbrace{\tilde{z}^-}_{\tilde{\zeta}_2}$$

- Can segregate more generally into distinct **segments** of the real line.

SLDR Approximation of General Model

$$\begin{aligned}
 & Z_{SLDR} = \\
 & \min_{\mathbf{x}, \{r^{0,k}, R^k\}_{k=1}^K} \quad c^{0'} \mathbf{x} + \sum_{k=1}^K d^{0,k'} r^{0,k} + \sup_{\hat{\zeta} \in \hat{\mathcal{V}}} \left(\sum_{k=1}^K d^{0,k'} R^k \hat{\zeta} \right) \\
 & \text{s.t.} \quad c^{l'} \mathbf{x} + \sum_{k=1}^K d^{l,k'} r^{0,k} + \sup_{\hat{\zeta} \in \hat{\mathcal{V}}} \left(\sum_{k=1}^K d^{l,k'} R^k \hat{\zeta} \right) \leq b_l \quad \forall l \in [M] \\
 & \quad \mathcal{T}^0 \mathbf{x} + \sum_{k=1}^K U^k r^{0,k} = \nu^0 \\
 & \quad \mathcal{T}^j \mathbf{x} + \sum_{k=1}^K U^k R^k e^j = \nu^j \quad \forall j \in [N_E] \\
 & \quad \underline{y}^k \leq r^{0,k} + R^k \zeta \leq \bar{y}^k \quad \forall \zeta \in \mathcal{V} \quad \forall k \in [K] \\
 & \quad \bar{R}^k e^j = 0 \quad \forall j \notin \Phi_k, \forall k \in [K] \\
 & \quad \mathbf{x} \geq 0
 \end{aligned}$$

Properties of SLDR

Theorem

For the general problem, $Z_{SLDR} \leq Z_{LDR}$

Properties of SLDR

Theorem

For the general problem, $Z_{SLDR} \leq Z_{LDR}$

- Increases the size of the problem (and computational effort needed), but retains the linear structure.

Properties of SLDR

Theorem

For the general problem, $Z_{SLDR} \leq Z_{LDR}$

- Increases the size of the problem (and computational effort needed), but retains the linear structure.
- Can we do better?

$$Z_{GEN}^* \leq ??? \leq Z_{SLDR} \leq Z_{LDR}$$

Properties of SLDR

Theorem

For the general problem, $Z_{SLDR} \leq Z_{LDR}$

- Increases the size of the problem (and computational effort needed), but retains the linear structure.
- Can we do better?

$$Z_{GEN}^* \leq ??? \leq Z_{SLDR} \leq Z_{LDR}$$

- Will base further improvement on the SLDR.

Outline

- 1 Introduction
- 2 Framework
- 3 Segregated LDR
- 4 Bi-Deflected LDR**
- 5 ROME
- 6 Conclusion

Review of Deflected LDR (DLDR)

- Originally introduced by Chen, Sim, Sun, and Zhang (2008) to circumvent restrictiveness imposed by the LDR.
 - The structure of the set of constraints might allow some piecewise-linear decision rules to be used instead.
 - Apply bounds on expected positive part of an LDR to exploit piecewise-linearity.

Review of Deflected LDR (DLDR)

- Originally introduced by Chen, Sim, Sun, and Zhang (2008) to circumvent restrictiveness imposed by the LDR.
 - The structure of the set of constraints might allow some piecewise-linear decision rules to be used instead.
 - Apply bounds on expected positive part of an LDR to exploit piecewise-linearity.
- Bi-deflected LDR seeks to improve and extend the DLDR.
 - Improve: Reduce the objective even further.
 - Extend: Explore modeling scenarios (non-anticipativity, expectations in constraints) which the DLDR didn't handle.

Motivating Example

- Scalar uncertainty \tilde{z} with unknown distribution \mathbb{P} in a family \mathbb{F} .
- \mathbb{F} defined by infinite support, zero mean, and unit variance. Consider the following problem:

$$\begin{aligned} \min_{y(\cdot)} \quad & \sup_{\mathbb{P} \in \mathbb{F}} \mathbb{E}_{\mathbb{P}} (|y(\tilde{z}) - \tilde{z}|) \\ & 0 \leq y(\tilde{z}) \leq 1 \\ & y \in \mathcal{Y}(1, 1, \{1\}) \end{aligned}$$

Motivating Example

- Scalar uncertainty \tilde{z} with unknown distribution \mathbb{P} in a family \mathbb{F} .
- \mathbb{F} defined by infinite support, zero mean, and unit variance. Consider the following problem:

$$\begin{aligned} \min_{y(\cdot)} \quad & \sup_{\mathbb{P} \in \mathbb{F}} \mathbb{E}_{\mathbb{P}} (|y(\tilde{z}) - \tilde{z}|) \\ & 0 \leq y(\tilde{z}) \leq 1 \\ & y \in \mathcal{Y}(1, 1, \{1\}) \end{aligned}$$

- Solution is pretty straightforward: piecewise-linear recourse function:

$$y_{sol}(\tilde{z}) = \begin{cases} \tilde{z} & \text{if } 0 \leq \tilde{z} \leq 1 \\ 0 & \text{if } \tilde{z} \leq 0 \\ 1 & \text{if } \tilde{z} \geq 1 \end{cases}$$

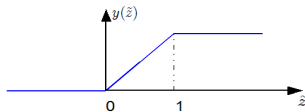
Motivating Example

- Scalar uncertainty \tilde{z} with unknown distribution \mathbb{P} in a family \mathbb{F} .
- \mathbb{F} defined by infinite support, zero mean, and unit variance. Consider the following problem:

$$\begin{aligned} \min_{y(\cdot)} \quad & \sup_{\mathbb{P} \in \mathbb{F}} \mathbb{E}_{\mathbb{P}} (|y(\tilde{z}) - \tilde{z}|) \\ & 0 \leq y(\tilde{z}) \leq 1 \\ & y \in \mathcal{Y}(1, 1, \{1\}) \end{aligned}$$

- Solution is pretty straightforward: piecewise-linear recourse function:

$$y_{sol}(\tilde{z}) = \begin{cases} \tilde{z} & \text{if } 0 \leq \tilde{z} \leq 1 \\ 0 & \text{if } \tilde{z} \leq 0 \\ 1 & \text{if } \tilde{z} \geq 1 \end{cases}$$



Example Reformulated into DRO Framework

- Apply the identities $x = x^+ - x^-$, $|x| = x^+ + x^-$.
- Model reformulates into DRO framework:

$$\begin{aligned}
 \min_{y(\cdot)} \quad & \sup_{\mathbb{P} \in \mathbb{F}} \mathbb{E}_{\mathbb{P}} (|y(\tilde{z}) - \tilde{z}|) \\
 & 0 \leq y(\tilde{z}) \leq 1 \\
 & y \in \mathcal{Y}(1, 1, \{1\}) \\
 & \updownarrow \\
 \min_{y(\cdot), u(\cdot), v(\cdot)} \quad & \sup_{\mathbb{P} \in \mathbb{F}} \mathbb{E}_{\mathbb{P}} (u(\tilde{z}) + v(\tilde{z})) \\
 \text{s.t.} \quad & u(\tilde{z}) - v(\tilde{z}) = y(\tilde{z}) - \tilde{z} \\
 & 0 \leq y(\tilde{z}) \leq 1 \\
 & u(\tilde{z}), v(\tilde{z}) \geq 0 \\
 & y, u, v \in \mathcal{Y}(1, 1, \{1\})
 \end{aligned}$$

Using Different Decision Rules

- Using LDR, problem is infeasible, objective = $+\infty$.

Using Different Decision Rules

- Using LDR, problem is infeasible, objective = $+\infty$.
- Using DLDR (of Chen et. al. 2008), we get the piecewise-linear decision rule:

$$\begin{aligned}\hat{u}_D(z) &= (u^0 + uz)^+ + (v^0 + vz)^- \\ \hat{v}_D(z) &= (v^0 + vz)^+ + (u^0 + uz)^- \\ y(z) &= y^0 + yz\end{aligned}$$

Using Different Decision Rules

- Using LDR, problem is infeasible, objective = $+\infty$.
- Using DLDR (of Chen et. al. 2008), we get the piecewise-linear decision rule:

$$\begin{aligned}\hat{u}_D(z) &= (u^0 + uz)^+ + (v^0 + vz)^- \\ \hat{v}_D(z) &= (v^0 + vz)^+ + (u^0 + uz)^- \\ y(z) &= y^0 + yz\end{aligned}$$

- After applying bounds, becomes an SOCP

$$\begin{aligned}Z_{DLDR} &= \min_{u^0, u, v^0, v} \left\| \begin{pmatrix} u^0 \\ u \end{pmatrix} \right\|_2 + \left\| \begin{pmatrix} v^0 \\ v \end{pmatrix} \right\|_2 \\ &\text{s.t. } u - v = -1 \\ &\quad 0 \leq u^0 - v^0 \leq 1\end{aligned}$$

Using Different Decision Rules

- Using LDR, problem is infeasible, objective = $+\infty$.
- Using DLDR (of Chen et. al. 2008), we get the piecewise-linear decision rule:

$$\begin{aligned}\hat{u}_D(z) &= (u^0 + uz)^+ + (v^0 + vz)^- \\ \hat{v}_D(z) &= (v^0 + vz)^+ + (u^0 + uz)^- \\ y(z) &= y^0 + yz\end{aligned}$$

- After applying bounds, becomes an SOCP

$$\begin{aligned}Z_{DLDR} &= \min_{u^0, u, v^0, v} \left\| \begin{pmatrix} u^0 \\ u \end{pmatrix} \right\|_2 + \left\| \begin{pmatrix} v^0 \\ v \end{pmatrix} \right\|_2 \\ \text{s.t.} & \quad u - v = -1 \\ & \quad 0 \leq u^0 - v^0 \leq 1\end{aligned}$$

- Objective = 1.

Different Decision Rules (II)

- Consider this hypothetical decision rule, which satisfies the model constraints:

$$\begin{aligned}\hat{u}(z) &= (u^0 + uz)^+ + (v^0 + vz)^- + (y^0 + yz)^- \\ \hat{v}(z) &= (v^0 + vz)^+ + (u^0 + uz)^- + (y^0 - 1 + yz)^+ \\ \hat{y}(z) &= (y^0 + yz)^+ - (y^0 - 1 + yz)^+\end{aligned}$$

Different Decision Rules (II)

- Consider this hypothetical decision rule, which satisfies the model constraints:

$$\begin{aligned}\hat{u}(z) &= (u^0 + uz)^+ + (v^0 + vz)^- + (y^0 + yz)^- \\ \hat{v}(z) &= (v^0 + vz)^+ + (u^0 + uz)^- + (y^0 - 1 + yz)^+ \\ \hat{y}(z) &= (y^0 + yz)^+ - (y^0 - 1 + yz)^+\end{aligned}$$

- Applying robust bounds, problem transforms into the SOCP:

$$\begin{aligned} \min_{y^0, y, u^0, u, v^0, v} \quad & \left\| \begin{pmatrix} u^0 \\ u \end{pmatrix} \right\|_2 + \left\| \begin{pmatrix} v^0 \\ v \end{pmatrix} \right\|_2 + \frac{1}{2} \left\| \begin{pmatrix} y^0 \\ y \end{pmatrix} \right\|_2 + \frac{1}{2} \left\| \begin{pmatrix} y^0 - 1 \\ y \end{pmatrix} \right\|_2 - \frac{1}{2} \\ \text{s.t.} \quad & u^0 - v^0 = y^0 \\ & u - v = y - 1 \end{aligned}$$

Different Decision Rules (II)

- Consider this hypothetical decision rule, which satisfies the model constraints:

$$\begin{aligned}\hat{u}(z) &= (u^0 + uz)^+ + (v^0 + vz)^- + (y^0 + yz)^- \\ \hat{v}(z) &= (v^0 + vz)^+ + (u^0 + uz)^- + (y^0 - 1 + yz)^+ \\ \hat{y}(z) &= (y^0 + yz)^+ - (y^0 - 1 + yz)^+\end{aligned}$$

- Applying robust bounds, problem transforms into the SOCP:

$$\begin{aligned}\min_{y^0, y, u^0, u, v^0, v} \quad & \left\| \begin{pmatrix} u^0 \\ u \end{pmatrix} \right\|_2 + \left\| \begin{pmatrix} v^0 \\ v \end{pmatrix} \right\|_2 + \frac{1}{2} \left\| \begin{pmatrix} y^0 \\ y \end{pmatrix} \right\|_2 + \frac{1}{2} \left\| \begin{pmatrix} y^0 - 1 \\ y \end{pmatrix} \right\|_2 - \frac{1}{2} \\ \text{s.t.} \quad & u^0 - v^0 = y^0 \\ & u - v = y - 1\end{aligned}$$

- Objective = 0.707 (better!)

Different Decision Rules (II)

- Consider this hypothetical decision rule, which satisfies the model constraints:

$$\begin{aligned}\hat{u}(z) &= (u^0 + uz)^+ + (v^0 + vz)^- + (y^0 + yz)^- \\ \hat{v}(z) &= (v^0 + vz)^+ + (u^0 + uz)^- + (y^0 - 1 + yz)^+ \\ \hat{y}(z) &= (y^0 + yz)^+ - (y^0 - 1 + yz)^+\end{aligned}$$

- Applying robust bounds, problem transforms into the SOCP:

$$\begin{aligned} \min_{y^0, y, u^0, u, v^0, v} \quad & \left\| \begin{pmatrix} u^0 \\ u \end{pmatrix} \right\|_2 + \left\| \begin{pmatrix} v^0 \\ v \end{pmatrix} \right\|_2 + \frac{1}{2} \left\| \begin{pmatrix} y^0 \\ y \end{pmatrix} \right\|_2 + \frac{1}{2} \left\| \begin{pmatrix} y^0 - 1 \\ y \end{pmatrix} \right\|_2 - \frac{1}{2} \\ \text{s.t.} \quad & u^0 - v^0 = y^0 \\ & u - v = y - 1 \end{aligned}$$

- Objective = 0.707 (better!)
- Aim to find algorithm to automatically predict this decision rule

Two-stage Bi-deflected LDR Problem Setup

- Consider a simpler two-stage model first:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{r}(\cdot)} \quad & \mathbf{c}'\mathbf{x} + \sup_{\mathbb{P} \in \mathbb{F}} \mathbb{E}_{\mathbb{P}} \left(\mathbf{d}'\mathbf{r}(\tilde{\zeta}) \right) \\ \text{s.t.} \quad & \mathbf{T}(\tilde{\zeta})\mathbf{x} + \mathbf{U}\mathbf{r}(\tilde{\zeta}) = \mathbf{v}(\tilde{\zeta}) \\ & \underline{\mathbf{y}} \leq \mathbf{r}(\tilde{\zeta}) \leq \bar{\mathbf{y}} \\ & \mathbf{r} \in \mathcal{L}(m, N_E, [N_E]) \end{aligned}$$

- Define the index sets of non-infinite bounds:

$$\begin{aligned} \underline{J} &= \left\{ i \in [m] : \underline{y}_i > -\infty \right\} \\ \bar{J} &= \left\{ i \in [m] : \bar{y}_i < +\infty \right\} \end{aligned}$$

Two-stage BDLDR Sub-problems

- We solve a series of sub-problems:

$$\begin{array}{ll}
 \min_{\mathbf{p}} & \mathbf{d}'\mathbf{p} \\
 \text{s.t.} & \mathbf{U}\mathbf{p} = \mathbf{0} \\
 & p_i = 1 \\
 & p_j \geq 0 \quad \forall j \in \underline{J} \\
 & p_j \leq 0 \quad \forall j \in \bar{J} \setminus \{i\}
 \end{array}$$

Feasible $i \in \underline{J}^\circ$

$$\begin{array}{ll}
 \min_{\mathbf{q}} & \mathbf{d}'\mathbf{q} \\
 \text{s.t.} & \mathbf{U}\mathbf{q} = \mathbf{0} \\
 & q_i = -1 \\
 & q_j \leq 0 \quad \forall j \in \bar{J} \\
 & q_j \geq 0 \quad \forall j \in \underline{J} \setminus \{i\}
 \end{array}$$

Feasible $i \in \bar{J}^\circ$

Two-stage BDLDR

- Consider an LDR which satisfies the relaxed constraints:

$$\begin{aligned}\mathcal{T}(\tilde{\zeta})\mathbf{x} + \mathbf{U}\mathbf{r}(\tilde{\zeta}) &= \boldsymbol{\nu}(\tilde{\zeta}) \\ r_j(\tilde{\zeta}) &\geq \underline{y}_j & \forall j \in \underline{J} \setminus \underline{J}^\circ \\ r_j(\tilde{\zeta}) &\leq \overline{y}_j & \forall j \in \overline{J} \setminus \overline{J}^\circ\end{aligned}$$

Two-stage BDLDR

- Consider an LDR which satisfies the relaxed constraints:

$$\begin{aligned} \mathcal{T}(\tilde{\zeta})x + Ur(\tilde{\zeta}) &= \nu(\tilde{\zeta}) \\ r_j(\tilde{\zeta}) &\geq \underline{y}_j & \forall j \in \underline{J} \setminus \underline{J}^\circ \\ r_j(\tilde{\zeta}) &\leq \bar{y}_j & \forall j \in \bar{J} \setminus \bar{J}^\circ \end{aligned}$$

- The two-stage BDLDR is defined as:

$$\hat{r}(\tilde{\zeta}) \triangleq \underbrace{r(\tilde{\zeta})}_{\text{Linear Part}} + \sum_{i \in \underline{J}^\circ} \left(r_i(\tilde{\zeta}) - \underline{y}_i \right)^- \underbrace{\bar{p}^i}_{\text{Optimal solution of the } i^{\text{th}} \text{ sub-problem (Deflected Part)}} + \sum_{i \in \bar{J}^\circ} \left(r_i(\tilde{\zeta}) - \bar{y}_i \right)^+ \underbrace{\bar{q}^i}_{\text{Optimal solution of the } i^{\text{th}} \text{ sub-problem (Deflected Part)}}$$

Two-stage BDLDR Properties

Theorem

The Bi-Deflected Linear Decision Rule, $\hat{\mathbf{r}}(\tilde{\zeta})$, satisfies the following properties

- 1 $\mathbf{U}\hat{\mathbf{r}}(\tilde{\zeta}) = \mathbf{U}\mathbf{r}(\tilde{\zeta})$
- 2 $\underline{\mathbf{y}} \leq \hat{\mathbf{r}}(\tilde{\zeta}) \leq \bar{\mathbf{y}}$

- For Reference:

$$\hat{\mathbf{r}}(\tilde{\zeta}) \triangleq \underbrace{\mathbf{r}(\tilde{\zeta})}_{\text{Linear Part}} + \sum_{i \in \underline{J}^o} \left(r_i(\tilde{\zeta}) - \underline{y}_i \right)^- \underbrace{\bar{\mathbf{p}}^i}_{\text{Optimal solution of the } i^{\text{th}} \text{ sub-problem (Deflected Part)}} + \sum_{i \in \bar{J}^o} \left(r_i(\tilde{\zeta}) - \bar{y}_i \right)^+ \underbrace{\bar{\mathbf{q}}^i}$$

BDLDR Approximation of Two-stage Problem

- Using the BDLDR, the problem becomes:

$$\begin{aligned}
 Z_{BDLDR} = & \\
 & \min_{\mathbf{x}, \mathbf{r}(\cdot)} \quad \mathbf{c}'\mathbf{x} + \sup_{\mathbb{P} \in \mathbb{F}} \mathbb{E}_{\mathbb{P}} \left(\mathbf{d}'\mathbf{r}(\tilde{\zeta}) \right) \\
 & + \sum_{i \in \underline{J}_R^\circ} \sup_{\mathbb{P} \in \mathbb{F}} \mathbb{E}_{\mathbb{P}} \left(\left(r_i(\tilde{\zeta}) - \underline{y}_i \right)^- \right) \mathbf{d}'\bar{\mathbf{p}}^i \\
 & + \sum_{i \in \bar{J}_R^\circ} \sup_{\mathbb{P} \in \mathbb{F}} \mathbb{E}_{\mathbb{P}} \left(\left(r_i(\tilde{\zeta}) - \bar{y}_i \right)^+ \right) \mathbf{d}'\bar{\mathbf{q}}^i \\
 & \text{s.t.} \\
 & \mathcal{T}(\tilde{\zeta})\mathbf{x} + \mathbf{U}\mathbf{r}(\tilde{\zeta}) = \boldsymbol{\nu}(\tilde{\zeta}) \\
 & r_j(\tilde{\zeta}) \geq \underline{y}_j \quad \forall j \in \underline{J} \setminus \underline{J}^\circ \\
 & r_j(\tilde{\zeta}) \leq \bar{y}_j \quad \forall j \in \bar{J} \setminus \bar{J}^\circ \\
 & \mathbf{r} \in \mathcal{L}(m, N_E, [N_E])
 \end{aligned}$$

Performance of Two-stage BDLDR

Theorem

For the two-stage problem, $Z_{BDLDR} \leq Z_{DLDR} \leq Z_{LDR}$.

- Proof is a bit tedious, but the basic idea is that the BDLDR includes the DLDR as a special case, and hence is more flexible and therefore yields a lower objective.
- Next, to generalize the BDLDR for use together with non-anticipativity requirements and expectation constraints.

BDLDR for General Model

- Define the following sets $\forall k \in [K]$:

$$\begin{aligned} N^+(k) &= \{j \in [K] : \Phi_k \subseteq \Phi_j\} \\ N^-(k) &= \{j \in [K] : \Phi_j \subseteq \Phi_k\} \end{aligned}$$

BDLDR for General Model

- Define the following sets $\forall k \in [K]$:

$$\begin{aligned} N^+(k) &= \{j \in [K] : \Phi_k \subseteq \Phi_j\} \\ N^-(k) &= \{j \in [K] : \Phi_j \subseteq \Phi_k\} \end{aligned}$$

- Similarly, we have two sub-problems:

$$\begin{aligned} \forall k \in [K], \forall i \in \underline{J}_k \\ \sum_{j \in N^+(k)} \mathbf{U}^j \mathbf{p}^{i,k,j} &= \mathbf{0} \\ p_l^{i,k,j} &\geq 0 \quad \forall l \in \underline{J}_j \quad \forall j \in N^+(k) \\ p_l^{i,k,j} &\leq 0 \quad \forall l \in \bar{J}_j \quad \forall j \in N^+(k) \setminus \{k\} \\ p_l^{i,k,k} &\leq 0 \quad \forall l \in \bar{J}_k \setminus \{i\} \\ p_i^{i,k,k} &= 1 \end{aligned}$$

$$\begin{aligned} \forall k \in [K], \forall i \in \bar{J}_k \\ \sum_{j \in N^+(k)} \mathbf{U}^j \mathbf{q}^{i,k,j} &= \mathbf{0} \\ q_l^{i,k,j} &\leq 0 \quad \forall l \in \bar{J}_j \quad \forall j \in N^+(k) \\ q_l^{i,k,j} &\geq 0 \quad \forall l \in \underline{J}_j \quad \forall j \in N^+(k) \setminus \{k\} \\ q_l^{i,k,k} &\geq 0 \quad \forall l \in \underline{J}_k \setminus \{i\} \\ q_i^{i,k,k} &= -1 \end{aligned}$$

Constructing the BDLDR

- Consider an LDR which satisfies the relaxed constraints

$$\begin{aligned} \mathcal{T}(\tilde{\zeta})x + \sum_{k=1}^K U^k r^k(\tilde{\zeta}) &= \nu(\tilde{\zeta}) \\ r_j^k(\tilde{\zeta}) &\geq \underline{y}_j^k & \forall k \in [K], \forall j \in \underline{J}_k \setminus \underline{J}_k^\circ \\ r_j^k(\tilde{\zeta}) &\leq \bar{y}_j^k & \forall k \in [K], \forall j \in \bar{J}_k \setminus \bar{J}_k^\circ \\ r^k &\in \mathcal{L}(m_k, N_E, \Phi_k) & \forall k \in [K] \end{aligned}$$

- The non-anticipative BDLDR is defined as:

$$\hat{r}^k(\tilde{\zeta}) \triangleq \underbrace{r^k(\tilde{\zeta})}_{\text{Linear Part}} + \sum_{j \in N^-(k)} \left(\sum_{i \in \underline{J}_j^\circ} (r_i^j(\tilde{\zeta}) - \underline{y}_i^j)^- \underbrace{p^{i,j,k}}_{\text{Optimal solution of } i^{\text{th}} \text{ sub-problem (Deflected Part)}} + \sum_{i \in \bar{J}_j^\circ} (r_i^j(\tilde{\zeta}) - \bar{y}_i^j)^+ \underbrace{q^{i,j,k}}_{\text{Optimal solution of } i^{\text{th}} \text{ sub-problem (Deflected Part)}} \right)$$

BDLDR Properties

Theorem

Each non-anticipative BDLDR, $\hat{\mathbf{r}}^k(\tilde{\zeta})$ satisfies the following properties:

- ① $\sum_{k=1}^K \mathbf{U}^k \hat{\mathbf{r}}^k(\tilde{\zeta}) = \sum_{k=1}^K \mathbf{U}^k \mathbf{r}^k(\tilde{\zeta})$
- ② $\underline{\mathbf{y}}^k \leq \hat{\mathbf{r}}^k(\tilde{\zeta}) \leq \bar{\mathbf{y}}^k, \quad \forall k \in [K]$

- For reference:

$$\hat{\mathbf{r}}^k(\tilde{\zeta}) \triangleq \underbrace{\mathbf{r}^k(\tilde{\zeta})}_{\text{Linear Part}} + \sum_{j \in N^-(k)} \left(\sum_{i \in \underline{J}_j^{\circ}} (r_i^j(\tilde{\zeta}) - \underline{y}_i^j)^- \underbrace{\mathbf{p}^{i,j,k}}_{\text{Optimal solution of } i^{\text{th}} \text{ sub-problem (Deflected Part)}} + \sum_{i \in \bar{J}_j^{\circ}} (r_i^j(\tilde{\zeta}) - \bar{y}_i^j)^+ \underbrace{\mathbf{q}^{i,j,k}}_{\text{Optimal solution of } i^{\text{th}} \text{ sub-problem (Deflected Part)}} \right)$$

BDLDR Performance

Theorem

For the general problem, we have

$$Z_{GEN}^* \leq Z_{BDLDR} \leq Z_{SLDR} \leq Z_{LDR}$$

- The k^{th} BDLDR sums over $j \in N^-(k)$,
 - These are the indices which are **contained by** the current (k^{th}) information index set,
 - BDLDR does not violate non-anticipative of the LDR which it is based upon.

Outline

- 1 Introduction
- 2 Framework
- 3 Segregated LDR
- 4 Bi-Deflected LDR
- 5 ROME**
- 6 Conclusion

Designing Software for RO Problems

- RO programs, especially with more complex decision rules (e.g. BDLDR), can be messy, typically involving the following steps:
 - Constructing robust counterparts,
 - Finding deflected components (non-anticipative),
 - Constructing robust bounds.

Designing Software for RO Problems

- RO programs, especially with more complex decision rules (e.g. BDLDR), can be messy, typically involving the following steps:
 - Constructing robust counterparts,
 - Finding deflected components (non-anticipative),
 - Constructing robust bounds.
- Software Design Goals:
 - Mathematically intuitive modeling of RO problems
 - Rapid prototyping of RO ideas
 - Ease transition from RO theory to practice
 - Allow numerical studies

Robust Optimization Made Easy with ROME

- ROME: an algebraic modeling language in the MATLAB environment for modeling Robust Optimization (RO) problems

Robust Optimization Made Easy with ROME

- ROME: an algebraic modeling language in the MATLAB environment for modeling Robust Optimization (RO) problems
- ROME is primarily designed for structured RO problems, within the DRO framework presented.
 - Other MATLAB-based modeling languages (e.g. CVX, YALMIP) solve more general types of problems.
 - ROME focuses on modeling phenomena more specific to RO, e.g. uncertainty description, decision rules, non-anticipativity.

Robust Optimization Made Easy with ROME

- ROME: an algebraic modeling language in the MATLAB environment for modeling Robust Optimization (RO) problems
- ROME is primarily designed for structured RO problems, within the DRO framework presented.
 - Other MATLAB-based modeling languages (e.g. CVX, YALMIP) solve more general types of problems.
 - ROME focuses on modeling phenomena more specific to RO, e.g. uncertainty description, decision rules, non-anticipativity.
- ROME is a modeling language, does not do actual solving.
 - Calls underlying solver engines to do perform solve step
 - Present version supports CPLEX, MOSEK, and SDPT3 solver engines

Example: Robust Inventory Model

- Model a distribution-free, multi-period, inventory control problem with service constraints.

Example: Robust Inventory Model

- Model a distribution-free, multi-period, inventory control problem with service constraints.
- Demand is exogenous, with unknown, but partially characterized distribution in a family \mathbb{F} , defined by:
 - Covariance Matrix: Temporal demand correlation can be modeled by a non-diagonal covariance matrix.
 - Mean: Assume fixed mean (for simplicity).
 - Support: Maximum demand in each period.

Example: Robust Inventory Model

- Model a distribution-free, multi-period, inventory control problem with service constraints.
- Demand is exogenous, with unknown, but partially characterized distribution in a family \mathbb{F} , defined by:
 - Covariance Matrix: Temporal demand correlation can be modeled by a non-diagonal covariance matrix.
 - Mean: Assume fixed mean (for simplicity).
 - Support: Maximum demand in each period.
- Backorders allowed, but in some applications, a penalty cost might not be a good model for stockouts.
 - In our model, we avoid stockouts with a constraint on the **fill-rate**.
 - Fill-rate constraint acts as a service guarantee to the consumers.

Model Parameters

Parameters

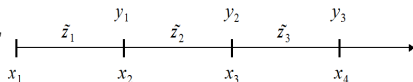
Num Periods : $T \in \mathbb{N}$

Order Cost : $\mathbf{c} \in \mathbb{R}^T$

Holding Cost : $\mathbf{h} \in \mathbb{R}^T$

Min Fill Rate : $\beta \in \mathbb{R}^T$

Max Order Qty : $\mathbf{x}^{MAX} \in \mathbb{R}^T$



Uncertainties

Demand : $\tilde{\mathbf{z}} \in \mathbb{R}^T$

Decisions

Order Quantity : $\mathbf{x}(\tilde{\mathbf{z}}) : x_t(\tilde{\mathbf{z}}) \in \mathcal{L}(1, T, [t-1])$

Inventory Level : $\mathbf{y}(\tilde{\mathbf{z}}) : y_t(\tilde{\mathbf{z}}) \in \mathcal{L}(1, T, [t])$

Robust Fill Rate Constraint

$$\text{Fill Rate} = \frac{\text{Expected Sales}}{\text{Expected Demand}} \geq \beta$$

Robust Fill Rate Constraint

$$\text{Fill Rate} = \frac{\text{Expected Sales}}{\text{Expected Demand}} \geq \beta$$

- Using our notation, the robust (worst-case) version:

$$\inf_{\mathbb{P} \in \mathcal{F}} \mathbb{E}_{\mathbb{P}} (\min \{ \tilde{z}_t, y_{t-1}(\tilde{z}) + x_t(\tilde{z}) \}) \geq \beta_t \mu_t$$

Robust Fill Rate Constraint

$$\text{Fill Rate} = \frac{\text{Expected Sales}}{\text{Expected Demand}} \geq \beta$$

- Using our notation, the robust (worst-case) version:

$$\inf_{\mathbb{P} \in \mathbb{F}} \mathbb{E}_{\mathbb{P}} (\min \{ \tilde{z}_t, y_{t-1}(\tilde{\mathbf{z}}) + x_t(\tilde{\mathbf{z}}) \}) \geq \beta_t \mu_t$$

- Apply inventory balance equation and re-arrange:

$$\sup_{\mathbb{P} \in \mathbb{F}} \mathbb{E}_{\mathbb{P}} (y_t(\tilde{\mathbf{z}})^-) \leq (1 - \beta_t) \mu_t$$

Model Formulation

- Family of uncertainties:

$$\mathbb{F} = \left\{ \mathbb{P} : \mathbb{E}_{\mathbb{P}}(\tilde{z}) = \boldsymbol{\mu}, \mathbb{E}_{\mathbb{P}}(\tilde{z}\tilde{z}') = \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}', \mathbb{P}(0 \leq \tilde{z} \leq \mathbf{z}^{MAX}) = 1 \right\}$$

Model Formulation

- Family of uncertainties:

$$\mathbb{F} = \left\{ \mathbb{P} : \mathbb{E}_{\mathbb{P}}(\tilde{\mathbf{z}}) = \boldsymbol{\mu}, \mathbb{E}_{\mathbb{P}}(\tilde{\mathbf{z}}\tilde{\mathbf{z}}') = \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}', \mathbb{P}(0 \leq \tilde{\mathbf{z}} \leq \mathbf{z}^{MAX}) = 1 \right\}$$

- Robust Inventory Model with Fill Rate constraints:

$$\begin{aligned} \min_{\mathbf{x}(\cdot), \mathbf{y}(\cdot)} \quad & \sup_{\mathbb{P} \in \mathbb{F}} \mathbb{E}_{\mathbb{P}}(\mathbf{c}'\mathbf{x}(\tilde{\mathbf{z}}) + \mathbf{h}'(\mathbf{y}(\tilde{\mathbf{z}}))^+) \\ \text{s.t.} \quad & \sup_{\mathbb{P} \in \mathbb{F}} \mathbb{E}_{\mathbb{P}}(\mathbf{y}(\tilde{\mathbf{z}})^-) \leq (\mathbf{I} - \mathbf{diag}(\boldsymbol{\beta}))\boldsymbol{\mu} \\ & \mathbf{D}\mathbf{y}(\tilde{\mathbf{z}}) - \mathbf{x}(\tilde{\mathbf{z}}) = -\tilde{\mathbf{z}} \\ & \mathbf{0} \leq \mathbf{x}(\tilde{\mathbf{z}}) \leq \mathbf{x}^{MAX} \\ & x_t \in \mathcal{L}(1, T, [t-1]) \quad \forall t \in [T] \\ & y_t \in \mathcal{L}(1, T, [t]) \quad \forall t \in [T] \end{aligned}$$

Model Formulation

- Family of uncertainties:

$$\mathbb{F} = \left\{ \mathbb{P} : \mathbb{E}_{\mathbb{P}}(\tilde{z}) = \boldsymbol{\mu}, \mathbb{E}_{\mathbb{P}}(\tilde{z}\tilde{z}') = \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}', \mathbb{P}(0 \leq \tilde{z} \leq \mathbf{z}^{MAX}) = 1 \right\}$$

- Robust Inventory Model with Fill Rate constraints:

$$\begin{aligned} \min_{\mathbf{x}(\cdot), \mathbf{y}(\cdot)} \quad & \sup_{\mathbb{P} \in \mathbb{F}} \mathbb{E}_{\mathbb{P}}(\mathbf{c}'\mathbf{x}(\tilde{z}) + \mathbf{h}'(\mathbf{y}(\tilde{z}))^+) \\ \text{s.t.} \quad & \sup_{\mathbb{P} \in \mathbb{F}} \mathbb{E}_{\mathbb{P}}(\mathbf{y}(\tilde{z})^-) \leq (\mathbf{I} - \mathbf{diag}(\boldsymbol{\beta}))\boldsymbol{\mu} \\ & \mathbf{D}\mathbf{y}(\tilde{z}) - \mathbf{x}(\tilde{z}) = -\tilde{z} \\ & \mathbf{0} \leq \mathbf{x}(\tilde{z}) \leq \mathbf{x}^{MAX} \\ & x_t \in \mathcal{L}(1, T, [t-1]) \quad \forall t \in [T] \\ & y_t \in \mathcal{L}(1, T, [t]) \quad \forall t \in [T] \end{aligned}$$

$$\mathbf{D} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

Model Transformed into DRO Framework

- After linearizing piecewise-linear terms,

$$\begin{aligned}
 \min_{\mathbf{x}(\cdot), \mathbf{y}(\cdot), \mathbf{r}(\cdot), \mathbf{s}(\cdot)} \quad & \sup_{\mathbb{P} \in \mathbb{F}} \mathbf{E}_{\mathbb{P}} (\mathbf{c}' \mathbf{x}(\tilde{\mathbf{z}}) + \mathbf{h}' \mathbf{r}(\tilde{\mathbf{z}})) \\
 \text{s.t.} \quad & \sup_{\mathbb{P} \in \mathbb{F}} \mathbf{E}_{\mathbb{P}} (\mathbf{s}(\tilde{\mathbf{z}})) \leq (\mathbf{I} - \mathbf{diag}(\boldsymbol{\beta})) \boldsymbol{\mu} \\
 & \mathbf{r}(\tilde{\mathbf{z}}) \geq \mathbf{y}(\tilde{\mathbf{z}}) \\
 & \mathbf{s}(\tilde{\mathbf{z}}) \geq -\mathbf{y}(\tilde{\mathbf{z}}) \\
 & \mathbf{r}(\tilde{\mathbf{z}}), \mathbf{s}(\tilde{\mathbf{z}}) \geq \mathbf{0} \\
 & \mathbf{D}\mathbf{y}(\tilde{\mathbf{z}}) - \mathbf{x}(\tilde{\mathbf{z}}) = -\tilde{\mathbf{z}} \\
 & \mathbf{0} \leq \mathbf{x}(\tilde{\mathbf{z}}) \leq \mathbf{x}^{MAX} \\
 & x_t \in \mathcal{L}(1, T, [t-1]) \quad \forall t \in [T] \\
 & r_t, s_t, y_t \in \mathcal{L}(1, T, [t]) \quad \forall t \in [T]
 \end{aligned}$$

ROME Code for Inventory Problem

```
% model parameters
T = 10; % planning horizon
c = 1 *ones(T, 1); % order cost rate
hcost = 2*ones(T, 1); % holding cost rate
beta = 0.50*ones(T, 1); % minimum fillrate in each period
xMax = 100*ones(T, 1); % maximum order quantity in each period
alpha = 0.5; % temporal autocorrelation factor
L = alpha * tril(ones(T), -1) + eye(T); % autocorrelation matrix

% numerical uncertainty parameters
zMax = 105*ones(T, 1); % maximum demand in each period
zMean = 30*ones(T, 1); % mean demand in each period
zCovar = 20*(L * L'); % temporal demand covariance

% differencing matrix
D = eye(T) - diag(ones(T-1, 1), -1);

% dependency structure
pX = logical([tril(ones(T)), zeros(T, 1)]);
```

ROME Code for Inventory Problem (cont.)

```

% Step 3: BDLDR Method
% -----
h = rome_begin('Robust Inventory (BDLDR)');
tic;

% declare uncertainties
newvar z(T) uncertain nonneg;

% define uncertainty parameters
rome_constraint(z <= zMax); % support
z.set_mean(zMean);         % mean
z.Covar = zCovar;         % covariance

% define LDR variables
newvar x(T, z, 'Pattern', pX) linearrule; % order quantity
newvar y(T, z) linearrule;                % inventory level

% define auxilliary variables
newvar r(T, z) s(T, z) linearrule nonneg;

```

$$\mathbb{F} = \left\{ \begin{array}{l} \mathbb{P} \left(0 \leq \tilde{z} \leq z^{MAX} \right) = 1 \\ \mathbb{E}_{\mathbb{P}} \left(\tilde{z} \right) = \mu, \\ \mathbb{E}_{\mathbb{P}} \left(\tilde{z} \tilde{z}' \right) = \Sigma + \mu \mu' \end{array} \right\}$$

$$\begin{array}{ll} \mathbf{r}(\tilde{z}), \mathbf{s}(\tilde{z}) \geq \mathbf{0} & \\ x_t \in \mathcal{L}(1, T, [t-1]) & \forall t \in [T] \\ r_t, s_t, y_t \in \mathcal{L}(1, T, [t]) & \forall t \in [T] \end{array}$$

ROME Code for Inventory Problem (cont.)

```

% auxilliary constraints
rome_constraint(r >= y); % since r >= y^-+
rome_constraint(s >= -y); % since s >= y^-

% fillrate constraint
rome_constraint(mean(s) <= diag(ones(T, 1) - beta) * zMean);

% inventory balance constraint
rome_constraint(D*y == x - z);

% order quantity constraints
rome_box(x, 0, xMax);

% objective
rome_minimize(c'*mean(x) + hcost'*mean(r));

% solve and display optimal objective
h.solve_deflected;
disp(sprintf('BDLDR Obj = %0.2f, time = %0.2f secs', h.ObjVal, toc));
x_sol_bdlldr = h.eval(x)

rome_end;

```

$$\min_{\mathbf{x}(\cdot), \mathbf{y}(\cdot), \mathbf{r}(\cdot), \mathbf{s}(\cdot)}$$

$$\text{s.t.}$$

$$\sup_{\mathbb{P} \in \mathbb{F}} \mathbb{E}_{\mathbb{P}} (\mathbf{c}' \mathbf{x}(\tilde{\mathbf{z}}) + \mathbf{h}' \mathbf{r}(\tilde{\mathbf{z}}))$$

$$\sup_{\mathbb{P} \in \mathbb{F}} \mathbb{E}_{\mathbb{P}} (\mathbf{s}(\tilde{\mathbf{z}})) \leq (\mathbf{I} - \text{diag}(\boldsymbol{\beta})) \boldsymbol{\mu}$$

$$\mathbf{D} \mathbf{y}(\tilde{\mathbf{z}}) - \mathbf{x}(\tilde{\mathbf{z}}) = -\tilde{\mathbf{z}}$$

$$\mathbf{r}(\tilde{\mathbf{z}}) \geq \mathbf{y}(\tilde{\mathbf{z}})$$

$$\mathbf{s}(\tilde{\mathbf{z}}) \geq -\mathbf{y}(\tilde{\mathbf{z}})$$

$$\mathbf{0} \leq \mathbf{x}(\tilde{\mathbf{z}}) \leq \mathbf{x}^{MAX}$$

ROME Output (MOSEK Solver)

```
EDU>> inventory_fillrate_example
Status: OPTIMAL
LDR Obj = 1747.50, time = 0.25 secs

x_sol_ldr =

  97.500
  0.000 + 0.952*z1
-0.000 + 0.007*z1 + 0.945*z2
  0.000 + 0.005*z1 + 0.010*z2 + 0.937*z3
-0.000 + 0.004*z1 + 0.006*z2 + 0.013*z3 + 0.929*z4
  0.000 + 0.003*z1 + 0.005*z2 + 0.008*z3 + 0.015*z4 + 0.921*z5
-0.000 + 0.004*z1 + 0.004*z2 + 0.006*z3 + 0.009*z4 + 0.018*z5 + 0.911*z6
-0.000 + 0.003*z1 + 0.005*z2 + 0.006*z3 + 0.008*z4 + 0.012*z5 + 0.023*z6 + 0.897*z7
  0.000 + 0.004*z1 + 0.005*z2 + 0.006*z3 + 0.009*z4 + 0.011*z5 + 0.016*z6 + 0.031*z7 + 0.870*z8
-0.000 + 0.007*z1 + 0.007*z2 + 0.009*z3 + 0.011*z4 + 0.015*z5 + 0.019*z6 + 0.031*z7 + 0.058*z8 + 0.795*z9

Status: NEAR_OPTIMAL
BDLDR Obj = 300.48, time = 0.58 secs

x_sol_blddr =

  15.333|
-0.000 + 1.003*z1 + 1.00(-0.000 + 1.003*z1)^- - 1.00(100.000 - 1.003*z1)^-
-0.000 + 0.000*z1 + 1.004*z2 + 1.00(-0.000 + 0.000*z1 + 1.004*z2)^- - 1.00(100.000 - 0.000*z1 - 1.004*z2)^-
-0.000 + 0.000*z1 - 0.000*z2 + 1.004*z3 + 1.00(-0.000 + 0.000*z1 - 0.000*z2 + 1.004*z3)^- - 1.00(100.000 - 0.000
-0.000 - 0.000*z1 - 0.000*z2 + 0.000*z3 + 1.004*z4 + 1.00(-0.000 - 0.000*z1 - 0.000*z2 + 0.000*z3 + 1.004*z4)^-
  0.000 - 0.000*z1 - 0.000*z2 - 0.000*z3 + 0.000*z4 + 1.004*z5 + 1.00(0.000 - 0.000*z1 - 0.000*z2 - 0.000*z3 + (
-0.000 + 0.000*z1 - 0.000*z2 + 0.000*z3 + 0.000*z4 - 0.000*z5 + 1.004*z6 + 1.00(-0.000 + 0.000*z1 - 0.000*z2 + (
  0.000 - 0.000*z1 - 0.000*z2 + 0.000*z3 + 0.000*z4 - 0.000*z5 - 0.000*z6 + 1.004*z7 + 1.00(0.000 - 0.000*z1 - (
-0.000 + 0.000*z1 - 0.000*z2 + 0.000*z3 + 0.000*z4 + 0.000*z5 - 0.000*z6 - 0.000*z7 + 1.004*z8 + 1.00(-0.000 + (
-0.000 + 0.000*z1 + 0.000*z2 - 0.000*z3 + 0.000*z4 + 0.000*z5 + 0.000*z6 + 0.000*z7 + 0.000*z8 + 1.005*z9 + 1.00
```



Outline

- 1 Introduction
- 2 Framework
- 3 Segregated LDR
- 4 Bi-Deflected LDR
- 5 ROME
- 6 Conclusion**

Summary

- Different decision rules for Distributionally Robust Optimization:
 - LDRs: Tractable approximation, but poor performance
 - SLDRs: Improves performance, and retains linear structure
 - BDLDRs: Improves performance beyond SLDR, but makes problem messy

Summary

- Different decision rules for Distributionally Robust Optimization:
 - LDRs: Tractable approximation, but poor performance
 - SLDRs: Improves performance, and retains linear structure
 - BDLDRs: Improves performance beyond SLDR, but makes problem messy
- ROME: Robust Optimization Made Easy
 - MATLAB-based modeling language for Distributionally Robust Optimization problems
 - Aims to allow robust optimization problems to be modeled and solved in a mathematically intuitive way
 - Freely-available for academic use from www.RobustOpt.com (together with User's Guide)