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

Outline

Introduction

Framework

Segregated LDR

Bi-Deflected LDR

ROME

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
- 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.
- To build a software modeling tool to solve DRO problems within this framework (ROME)

Outline

Notation

- $\tilde{\boldsymbol{z}} \in \Re^N$: uncertainties with distribution $\mathbb{P} \in \mathbb{F}$.
- $\boldsymbol{x} \in \Re^n$: "here-and-now" decision variables.
- $\boldsymbol{y}^k(\tilde{\boldsymbol{z}}) \in \Re^{m_k}$: "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 $I_k \subseteq [N], \forall k \in [K]$.

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

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

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

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 $\boldsymbol{\Sigma}$: Positive definite
  - Upper bounds on Directional Deviations for stochastically independent components $(\boldsymbol{H}_{\sigma}, \boldsymbol{\sigma_f}, \boldsymbol{\sigma_b})$.

Formulation of General Model

$$
\begin{array}{lll}
{Z_{GEN}}^* = & & \\
\min\limits_{\boldsymbol{x},\{\boldsymbol{y}^k(\cdot)\}_{k=1}^K} & \boldsymbol{c}^{0\prime}\boldsymbol{x} + \sup\limits_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}}\left(\sum_{k=1}^K \boldsymbol{d}^{0,k\prime}\boldsymbol{y}^k(\tilde{\boldsymbol{z}})\right) & \\
\text{s.t.} & \boldsymbol{c}^{l\prime}\boldsymbol{x} + \sup\limits_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}}\left(\sum_{k=1}^K \boldsymbol{d}^{l,k\prime}\boldsymbol{y}^k(\tilde{\boldsymbol{z}})\right) \leq b_l & \forall l \in [M] \\
& \boldsymbol{T}(\tilde{\boldsymbol{z}})\boldsymbol{x} + \sum_{k=1}^K \boldsymbol{U}^k\boldsymbol{y}^k(\tilde{\boldsymbol{z}}) = \boldsymbol{v}(\tilde{\boldsymbol{z}}) & \\
& \underline{\boldsymbol{y}}^k \leq \boldsymbol{y}^k(\tilde{\boldsymbol{z}}) \leq \overline{\boldsymbol{y}}^k & \forall k \in [K] \\
& \boldsymbol{x} \geq \boldsymbol{0} & \\
& \boldsymbol{y}^k \in \mathcal{Y}(m_k, N, I_k) & \forall k \in [K]
\end{array}
$$

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

$$\mathcal{L}(m,N,I) \triangleq \left\{ \boldsymbol{f} : \Re^N \to \Re^m : \exists \left(\boldsymbol{y}^0, \boldsymbol{Y}\right) \in \Re^m \times \Re^{m \times N} : \begin{array}{c} \boldsymbol{f}(\boldsymbol{z}) = \boldsymbol{y}^0 + \boldsymbol{Y}\boldsymbol{z} \\ \boldsymbol{Y}\boldsymbol{e}^i = \boldsymbol{0}, \forall i \notin I \end{array} \right\}$$

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

LDR Approximation of General Model

$$
\begin{array}{lll}
Z_{LDR} = & & \\
\min\limits_{\boldsymbol{x}, \{\boldsymbol{y}^k(\cdot)\}_{k=1}^K} & \boldsymbol{c}^{0\prime}\boldsymbol{x} + \sup\limits_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}}\left(\sum_{k=1}^K \boldsymbol{d}^{0,k\prime}\boldsymbol{y}^k(\tilde{\boldsymbol{z}})\right) & \\
\text{s.t.} & \boldsymbol{c}^{l\prime}\boldsymbol{x} + \sup\limits_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}}\left(\sum_{k=1}^K \boldsymbol{d}^{l,k\prime}\boldsymbol{y}^k(\tilde{\boldsymbol{z}})\right) \leq b_l & \forall l \in [M] \\
& \boldsymbol{T}(\tilde{\boldsymbol{z}})\boldsymbol{x} + \sum_{k=1}^K \boldsymbol{U}^k \boldsymbol{y}^k(\tilde{\boldsymbol{z}}) = \boldsymbol{v}(\tilde{\boldsymbol{z}}) & \\
& \underline{\boldsymbol{y}}^k \leq \boldsymbol{y}^k(\tilde{\boldsymbol{z}}) \leq \overline{\boldsymbol{y}}^k & \forall k \in [K] \\
& \boldsymbol{x} \geq \boldsymbol{0} & \\
& \boldsymbol{y}^k \in \mathcal{L}(m_k, N, I_k) & \forall k \in [K]
\end{array}
$$

LDR Approx. of General Model (Explicit)

$$
\begin{array}{lll}
Z_{LDR} = & & \\
\min_{\boldsymbol{x}, \{\boldsymbol{y}^{0,k}, \boldsymbol{Y}^k\}_{k=1}^K} & \boldsymbol{c}^{0\prime}\boldsymbol{x} + \sum_{k=1}^K \boldsymbol{d}^{0,k\prime}\boldsymbol{y}^{0,k} + \sup_{\hat{\boldsymbol{z}} \in \hat{\mathcal{W}}} \left(\sum_{k=1}^K \boldsymbol{d}^{0,k\prime}\boldsymbol{Y}^k \hat{\boldsymbol{z}} \right) & \\
\text{s.t.} & \boldsymbol{c}^{l\prime}\boldsymbol{x} + \sum_{k=1}^K \boldsymbol{d}^{l,k\prime}\boldsymbol{y}^{0,k} + \sup_{\hat{\boldsymbol{z}} \in \hat{\mathcal{W}}} \left(\sum_{k=1}^K \boldsymbol{d}^{l,k\prime}\boldsymbol{Y}^k \hat{\boldsymbol{z}} \right) \leq b_l & \forall l \in [M] \\
& \boldsymbol{T}^0\boldsymbol{x} + \sum_{k=1}^K \boldsymbol{U}^k \boldsymbol{y}^{0,k} = \boldsymbol{v}^0 & \\
& \boldsymbol{T}^j\boldsymbol{x} + \sum_{k=1}^K \boldsymbol{U}^k \boldsymbol{Y}^k \boldsymbol{e}^j = \boldsymbol{v}^j & \forall j \in [N] \\
& \underline{\boldsymbol{y}}^k \leq \boldsymbol{y}^{0,k} + \boldsymbol{Y}^k \boldsymbol{z} \leq \overline{\boldsymbol{y}}^k \quad \forall \boldsymbol{z} \in \mathcal{W} & \forall k \in [K] \\
& \boldsymbol{Y}^k \boldsymbol{e}^j = \boldsymbol{0} & \forall j \notin I_k, \forall k \in [K] \\
& \boldsymbol{x} \geq \boldsymbol{0} &
\end{array}
$$

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

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

$$
\begin{array}{lll}
Z_{SLDR} = & & \\
\min\limits_{\boldsymbol{x}, \left\{\boldsymbol{r}^{0,k}, \boldsymbol{R}^k\right\}_{k=1}^K} & \boldsymbol{c}^{0\prime}\boldsymbol{x} + \sum\limits_{k=1}^K \boldsymbol{d}^{0,k\prime}\boldsymbol{r}^{0,k} + \sup\limits_{\hat{\zeta} \in \hat{\mathcal{V}}} \left(\sum\limits_{k=1}^K \boldsymbol{d}^{0,k\prime}\boldsymbol{R}^k\hat{\boldsymbol{\zeta}} \right) & \\
\text{s.t.} & \boldsymbol{c}^{l\prime}\boldsymbol{x} + \sum\limits_{k=1}^K \boldsymbol{d}^{l,k\prime}\boldsymbol{r}^{0,k} + \sup\limits_{\hat{\zeta} \in \hat{\mathcal{V}}} \left(\sum\limits_{k=1}^K \boldsymbol{d}^{l,k\prime}\boldsymbol{R}^k\hat{\boldsymbol{\zeta}} \right) \le b_l & \forall l \in [M] \\
& \boldsymbol{\mathcal{T}}^0\boldsymbol{x} + \sum\limits_{k=1}^K \boldsymbol{U}^k\boldsymbol{r}^{0,k} = \boldsymbol{\nu}^0 & \\
& \boldsymbol{\mathcal{T}}^j\boldsymbol{x} + \sum\limits_{k=1}^K \boldsymbol{U}^k\boldsymbol{R}^k\boldsymbol{e}^j = \boldsymbol{\nu}^j & \forall j \in [N_E] \\
& \underline{\boldsymbol{y}}^k \le \boldsymbol{r}^{0,k} + \boldsymbol{R}^k\boldsymbol{\zeta} \le \overline{\boldsymbol{y}}^k \quad \forall \boldsymbol{\zeta} \in \mathcal{V} & \forall k \in [K] \\
& \boldsymbol{R}^k\boldsymbol{e}^j = \boldsymbol{0} & \forall j \notin \Phi_k, \forall k \in [K] \\
& \boldsymbol{x} \ge \boldsymbol{0} &
\end{array}
$$

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

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.

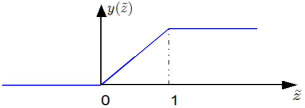
Motivating Example

- 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:

$$\begin{array}{ll} \min\limits_{y(\cdot)} & \sup\limits_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}}\left(|y(\tilde{z}) - \tilde{z}|\right) \\ & 0 \leq y(\tilde{z}) \leq 1 \\ & y \in \mathcal{Y}\left(1, 1, \{1\}\right) \end{array}$$

- 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:

$$\begin{array}{cl} \min\limits_{y(\cdot)} & \sup\limits_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}}\left(|y(\tilde{z}) - \tilde{z}|\right) \\ & 0 \leq y(\tilde{z}) \leq 1 \\ & y \in \mathcal{Y}\left(1, 1, \{1\}\right) \end{array}$$

$$\Updownarrow$$

$$\begin{array}{cl} \min\limits_{y(\cdot),u(\cdot),v(\cdot)} & \sup\limits_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}}\left(u(\tilde{z}) + v(\tilde{z})\right) \\ \text{s.t.} & u(\tilde{z}) - v(\tilde{z}) = y(\tilde{z}) - \tilde{z} \\ & 0 \leq y(\tilde{z}) \leq 1 \\ & u(\tilde{z}), v(\tilde{z}) \geq 0 \\ & y, u, v \in \mathcal{Y}\left(1, 1, \{1\}\right) \end{array}$$

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:

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

- After applying bounds, becomes an SOCP

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

- Objective $= 1$.

Different Decision Rules (II)

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

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

- Applying robust bounds, problem transforms into the SOCP:

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

- 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:

$$
\begin{array}{rl}
\min\limits_{\boldsymbol{x},\boldsymbol{r}(\cdot)} & \boldsymbol{c}'\boldsymbol{x} + \sup\limits_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}}\left(\boldsymbol{d}'\boldsymbol{r}(\tilde{\boldsymbol{\zeta}})\right) \\
\text{s.t.} & \boldsymbol{\mathcal{T}}(\tilde{\boldsymbol{\zeta}})\boldsymbol{x} + \boldsymbol{U}\boldsymbol{r}(\tilde{\boldsymbol{\zeta}}) = \boldsymbol{\nu}(\tilde{\boldsymbol{\zeta}}) \\
& \underline{\boldsymbol{y}} \leq \boldsymbol{r}(\tilde{\boldsymbol{\zeta}}) \leq \overline{\boldsymbol{y}} \\
& \boldsymbol{r} \in \mathcal{L}(m, N_E, [N_E])
\end{array}
$$

- Define the index sets of non-infinite bounds:

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

Two-stage BDLDR Sub-problems

- We solve a series of sub-problems:

$$\underline{\forall i \in \underline{J}}$$

$$\begin{aligned} \min_{\boldsymbol{p}} \quad & \boldsymbol{d}'\boldsymbol{p} \\ \text{s.t.} \quad & \boldsymbol{U}\boldsymbol{p} = \boldsymbol{0} \\ & p_i = 1 \\ & p_j \geq 0 \qquad \forall j \in \underline{J} \\ & p_j \leq 0 \qquad \forall j \in \overline{J} \setminus \{i\} \end{aligned}$$

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

$$\underline{\forall i \in \overline{J}}$$

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

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

Two-stage BDLDR

- Consider an LDR which satisfies the relaxed constraints:

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

- The two-stage BDLDR is defined as:

$$\hat{\boldsymbol{r}}(\tilde{\boldsymbol{\zeta}}) \triangleq \underbrace{\boldsymbol{r}(\tilde{\boldsymbol{\zeta}})}_{\text{Linear Part}} + \sum_{i \in \underline{J}^\circ} \left(r_i(\tilde{\boldsymbol{\zeta}}) - \underline{y}_i \right)^- \underbrace{\bar{\boldsymbol{p}}^i} + \sum_{i \in \overline{J}^\circ} \left(r_i(\tilde{\boldsymbol{\zeta}}) - \overline{y}_i \right)^+ \underbrace{\bar{\boldsymbol{q}}^i}$$

Optimal solution of the i^{th} sub-problem (Deflected Part)

Two-stage BDLDR Properties

Theorem

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

1. $\boldsymbol{U}\hat{\boldsymbol{r}}(\tilde{\boldsymbol{\zeta}}) = \boldsymbol{U}\boldsymbol{r}(\tilde{\boldsymbol{\zeta}})$
2. $\underline{\boldsymbol{y}} \leq \hat{\boldsymbol{r}}(\tilde{\boldsymbol{\zeta}}) \leq \overline{\boldsymbol{y}}$

- For Reference:

$$\hat{\boldsymbol{r}}(\tilde{\boldsymbol{\zeta}}) \triangleq \underbrace{\boldsymbol{r}(\tilde{\boldsymbol{\zeta}})}_{\text{Linear Part}} + \sum_{i \in \underline{J}^\circ} \left(r_i(\tilde{\boldsymbol{\zeta}}) - \underline{y}_i\right)^- \underbrace{\bar{\boldsymbol{p}}^i} + \sum_{i \in \overline{J}^\circ} \left(r_i(\tilde{\boldsymbol{\zeta}}) - \overline{y}_i\right)^+ \underbrace{\bar{\boldsymbol{q}}^i}$$

Optimal solution of the i^{th} sub-problem (Deflected Part)

BDLDR Approximation of Two-stage Problem

- Using the BDLDR, the problem becomes:

$$
\begin{array}{rll}
Z_{BDLDR} = & & \\
\min\limits_{\boldsymbol{x},\boldsymbol{r}(\cdot)} & \boldsymbol{c}'\boldsymbol{x} + \sup\limits_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}}\left(\boldsymbol{d}'\boldsymbol{r}(\tilde{\boldsymbol{\zeta}})\right) & \\
& + \sum\limits_{i\in\underline{J}^{\circ}_{R}} \sup\limits_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}}\left(\left(r_i(\tilde{\boldsymbol{\zeta}}) - \underline{y}_i\right)^{-}\right)\boldsymbol{d}'\bar{\boldsymbol{p}}^{i} & \\
& + \sum\limits_{i\in\overline{J}^{\circ}_{R}} \sup\limits_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}}\left(\left(r_i(\tilde{\boldsymbol{\zeta}}) - \overline{y}_i\right)^{+}\right)\boldsymbol{d}'\bar{\boldsymbol{q}}^{i} & \\
\text{s.t.} & & \\
& \boldsymbol{\mathcal{T}}(\tilde{\boldsymbol{\zeta}})\boldsymbol{x} + \boldsymbol{U}\boldsymbol{r}(\tilde{\boldsymbol{\zeta}}) = \boldsymbol{\nu}(\tilde{\boldsymbol{\zeta}}) & \\
& r_j(\tilde{\boldsymbol{\zeta}}) \geq \underline{y}_j & \forall j \in \underline{J} \setminus \underline{J}^{\circ} \\
& r_j(\tilde{\boldsymbol{\zeta}}) \leq \overline{y}_j & \forall j \in \overline{J} \setminus \overline{J}^{\circ} \\
& \boldsymbol{r} \in \mathcal{L}(m, N_E, [N_E]) &
\end{array}
$$

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 $\forall k \in [K]$:

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

- Similarly, we have two sub-problems:

$$\begin{array}{rcll} \multicolumn{4}{l}{\underline{\forall k \in [K], \forall i \in \underline{J}_k}} \\ \sum_{j \in N^+(k)} \boldsymbol{U}^j \boldsymbol{p}^{i,k,j} &=& \boldsymbol{0} & \\ p_l^{i,k,j} &\geq& 0 \quad \forall l \in \underline{J}_j & \forall j \in N^+(k) \\ p_l^{i,k,j} &\leq& 0 \quad \forall l \in \overline{J}_j & \forall j \in N^+(k) \setminus \{k\} \\ p_l^{i,k,k} &\leq& 0 \quad \forall l \in \overline{J}_k \setminus \{i\} & \\ p_i^{i,k,k} &=& 1 & \end{array}$$

$$\begin{array}{rcll} \multicolumn{4}{l}{\underline{\forall k \in [K], \forall i \in \overline{J}_k}} \\ \sum_{j \in N^+(k)} \boldsymbol{U}^j \boldsymbol{q}^{i,k,j} &=& \boldsymbol{0} & \\ q_l^{i,k,j} &\leq& 0 \quad \forall l \in \overline{J}_j & \forall j \in N^+(k) \\ q_l^{i,k,j} &\geq& 0 \quad \forall l \in \underline{J}_j & \forall j \in N^+(k) \setminus \{k\} \\ q_l^{i,k,k} &\geq& 0 \quad \forall l \in \underline{J}_k \setminus \{i\} & \\ q_i^{i,k,k} &=& -1 & \end{array}$$

Constructing the BDLDR

- Consider an LDR which satisfies the relaxed constraints

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

- The non-anticipative BDLDR is defined as:

$$
\hat{\boldsymbol{r}}^k(\tilde{\boldsymbol{\zeta}}) \triangleq \underbrace{\boldsymbol{r}^k(\tilde{\boldsymbol{\zeta}})}_{\text{Linear Part}} + \sum_{j \in N^-(k)} \Bigg(\sum_{i \in \underline{J}_j^\circ} \left(r_i^j(\tilde{\boldsymbol{\zeta}}) - \underline{y}_i^j \right)^- \underbrace{\boldsymbol{p}^{i,j,k}} + \sum_{i \in \overline{J}_j^\circ} \left(r_i^j(\tilde{\boldsymbol{\zeta}}) - \overline{y}_i^j \right)^+ \underbrace{\boldsymbol{q}^{i,j,k}} \Bigg)
$$

Optimal solution of i^{th} sub-problem (Deflected Part)

BDLDR Properties

Theorem

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

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

- For reference:

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

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

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
- 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 phenonena 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.
- 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

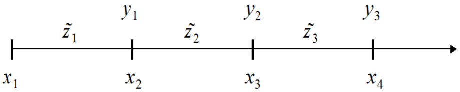
Num Periods : $T \in \mathbb{N}$

Order Cost : $\boldsymbol{c} \in \Re^T$

Holding Cost : $\boldsymbol{h} \in \Re^T$

Min Fill Rate : $\boldsymbol{\beta} \in \Re^T$

Max Order Qty : $\boldsymbol{x}^{MAX} \in \Re^T$



Uncertainties

Demand : $\tilde{\boldsymbol{z}} \in \Re^T$

Decisions

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

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

Robust Fill Rate Constraint

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

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

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

- Apply inventory balance equation and re-arrange:

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

Model Formulation

- Family of uncertainties:

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

- Robust Inventory Model with Fill Rate constraints:

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

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

Model Transformed into DRO Framework

- After linearizing piecewise-linear terms,

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

ROME Code for Inventory Problem

```
% 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
```

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

```
% 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;
```

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

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_bdldr = h.eval(x)

rome_end;
```

$$\begin{array}{cl} \min\limits_{\boldsymbol{x}(\cdot),\boldsymbol{y}(\cdot),\boldsymbol{r}(\cdot),\boldsymbol{s}(\cdot)} & \sup\limits_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}}\left(\boldsymbol{c}'\boldsymbol{x}(\tilde{\boldsymbol{z}}) + \boldsymbol{h}'\boldsymbol{r}(\tilde{\boldsymbol{z}})\right) \\ \text{s.t.} & \sup\limits_{\mathbb{P}\in\mathbb{F}} \mathrm{E}_{\mathbb{P}}\left(\boldsymbol{s}(\tilde{\boldsymbol{z}})\right) \leq (\boldsymbol{I} - \mathbf{diag}\,(\boldsymbol{\beta}))\boldsymbol{\mu} \\ & \boldsymbol{D}\boldsymbol{y}\,(\tilde{\boldsymbol{z}}) - \boldsymbol{x}\,(\tilde{\boldsymbol{z}}) = -\tilde{\boldsymbol{z}} \\ & \boldsymbol{r}\,(\tilde{\boldsymbol{z}}) \geq \boldsymbol{y}\,(\tilde{\boldsymbol{z}}) \\ & \boldsymbol{s}\,(\tilde{\boldsymbol{z}}) \geq -\boldsymbol{y}\,(\tilde{\boldsymbol{z}}) \\ & \boldsymbol{0} \leq \boldsymbol{x}\,(\tilde{\boldsymbol{z}}) \leq \boldsymbol{x}^{MAX} \end{array}$$

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_bdldr =

 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 - 0.0(
-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.(
```

Outline

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)