Robust Multi-Stage Decision Making

Erick Delage[1] **Dan Iancu**[2]

[1]CRC in Decision Making under Uncertainty
HEC Montréal

[2]Graduate School of Business
Stanford University

INFORMS Tutorial
November 1st, 2015

Decisions Under Uncertainty

Decision maker chooses x; u unknown

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & C(x, u) \\ \text{s.t.} \quad & F(x, u) \geqslant 0 \end{aligned}$$

Decisions Under Uncertainty

Decision maker chooses x; u unknown

$$\begin{aligned} \underset{x}{\text{minimize}} \;& C(x, u) \\ \text{s.t.} \;& F(x, u) \geqslant 0 \end{aligned}$$

Stochastic model:

$$\begin{aligned} &\min_{x} \mathbb{E}_{u}\left[C(x, u)\right] \\ &\text{s.t.} \;\; \mathbb{P}[\, F(x, u) \geqslant 0\,] \geqslant 1 - \epsilon \end{aligned}$$

- Distribution for u known
- Well-defined objective : *average* performance
- Good data, future like past, ...

Robust model:

$$\begin{aligned} &\min_{x} \max_{u \in \mathcal{U}} C(x, u) \\ &\text{s.t.} \;\; F(x, u) \geqslant 0, \forall\, u \in \mathcal{U} \end{aligned}$$

- Uncertainty set $\mathcal{U}$ known
- Well-defined objective : *worst-case* performance
- Poor data, non-stationarity, ...

Decisions Under Uncertainty

Decision maker chooses $\mathbf{x}$; $\mathbf{u}$ unknown

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \; & C(\mathbf{x}, \mathbf{u}) \\ \text{s.t.} \; & F(\mathbf{x}, \mathbf{u}) \geqslant 0 \end{aligned}$$

Stochastic model:

$$\begin{aligned} \min_{\mathbf{x}} \; & \mathbb{E}_{\mathbf{u}}\left[C(\mathbf{x}, \mathbf{u})\right] \\ \text{s.t.} \; & \mathbb{P}\left[\, F\big(\mathbf{x}, \mathbf{u}\big) \geqslant 0 \,\right] \geqslant 1 - \epsilon \end{aligned}$$

- Distribution for $\mathbf{u}$ known
- Well-defined objective : *average* performance
- Good data, future like past, ...

Robust model:

$$\begin{aligned} \min_{\mathbf{x}} \max_{\mathbf{u} \in \mathcal{U}} \; & C(\mathbf{x}, \mathbf{u}) \\ \text{s.t.} \; & F(\mathbf{x}, \mathbf{u}) \geqslant 0, \forall\, \mathbf{u} \in \mathcal{U} \end{aligned}$$

- Uncertainty set $\mathcal{U}$ known
- Well-defined objective : *worst-case* performance
- Poor data, non-stationarity, ...
- *Infinitely many constraints*

Robust Optimization

Robust model:

$$\min_{\boldsymbol{x}} \max_{\mathbf{u}\in\mathcal{U}} C(\boldsymbol{x},\mathbf{u})$$
$$\text{s.t. } F(\boldsymbol{x},\mathbf{u}) \geqslant 0, \forall\, \mathbf{u} \in \mathcal{U}$$

- For many classes of $\mathcal{U}$, can reformulate as convex optimization

Robust Optimization

Robust model:

$$\min_{x} \max_{\mathbf{u} \in \mathcal{U}} C(x, \mathbf{u})$$
$$\text{s.t. } F(x, \mathbf{u}) \geqslant 0, \forall\, \mathbf{u} \in \mathcal{U}$$

- For many classes of $\mathcal{U}$, can reformulate as convex optimization
  Soyster [1973], Falk [1976], Ben-Tal and Nemirovski [1998, 1999, 2000, 2002], Ben-Tal et al. [2002], El-Ghaoui and Lebret [1997], El-Ghaoui et al. [1998], Bertsimas and Sim [2003, 2004], Bertsimas et al. [2004], ...

Robust Optimization

Robust model:

$$\min_{\boldsymbol{x}} \max_{\mathbf{u}\in\mathcal{U}} C(\mathbf{x}, \mathbf{u})$$
$$\text{s.t. } F(\mathbf{x}, \mathbf{u}) \geqslant 0, \forall\, \mathbf{u} \in \mathcal{U}$$

- For many classes of $\mathcal{U}$, can reformulate as convex optimization
  Soyster [1973], Falk [1976], Ben-Tal and Nemirovski [1998, 1999, 2000, 2002], Ben-Tal et al. [2002], El-Ghaoui and Lebret [1997], El-Ghaoui et al. [1998], Bertsimas and Sim [2003, 2004], Bertsimas et al. [2004], ...
- Focus on *tractability*, *degree of conservatism*, *probabilistic guarantees*

Robust Optimization

Robust model:

$$\min_{x} \max_{\mathbf{u}\in\mathcal{U}} C(x, \mathbf{u})$$
$$\text{s.t. } F(x, \mathbf{u}) \geqslant 0, \forall\, \mathbf{u} \in \mathcal{U}$$

- For many classes of $\mathcal{U}$, can reformulate as convex optimization
  Soyster [1973], Falk [1976], Ben-Tal and Nemirovski [1998, 1999, 2000, 2002], Ben-Tal et al. [2002], El-Ghaoui and Lebret [1997], El-Ghaoui et al. [1998], Bertsimas and Sim [2003, 2004], Bertsimas et al. [2004], ...
- Focus on *tractability*, *degree of conservatism*, *probabilistic guarantees*
- Some decisions *adjustable* → approach extended to dynamic settings
  Ben-Tal et al. [2004]

Robust Optimization

Robust model:

$$\min_{x} \max_{\mathbf{u}\in\mathcal{U}} C(x, \mathbf{u})$$
$$\text{s.t. } F(x, \mathbf{u}) \geqslant 0, \forall\, \mathbf{u} \in \mathcal{U}$$

Applications...

- inventory management e.g., [Ben-Tal et al., 2005, Bertsimas and Thiele, 2006, Bienstock and Özbay, 2008, ...]
- facility location and transportation [Baron et al., 2011, ...]
- scheduling [Lin et al., 2004, Yamashita et al., 2007, Mittal et al., 2014, ...]
- revenue management [Perakis and Roels, 2010, Adida and Perakis, 2006, ...]
- project management [Wiesemann et al., 2012, Ben-Tal et al., 2009, ...]
- energy generation and distribution [Zhao et al., 2013, Lorca and Sun, 2015, ...]
- portfolio optimization [Goldfarb and Iyengar, 2003, Tütüncü and Koenig, 2004, Ceria and Stubbs, 2006, Pinar and Tütüncü, 2005, Bertsimas and Pachamanova, 2008, ...]
- healthcare [Borfeld et al., 2008, Hanne et al., 2009, Chen et al., 2011, ...]

Book [Ben-Tal et al., 2009]; Review papers: [Bertsimas et al., 2011a, Gabrel et al., 2012]

Robust Optimization

Robust model:

$$\min_{\boldsymbol{x}} \max_{\mathbf{u}\in\mathcal{U}} C(\boldsymbol{x},\mathbf{u})$$
$$\text{s.t. } F(\boldsymbol{x},\mathbf{u}) \geqslant 0, \forall\, \mathbf{u} \in \mathcal{U}$$

Our objectives with this tutorial...

- Discuss static versus adjustable decisions
- Highlight some tractability issues
- Clarify the connection to robust dynamic programming
- Provide motivations for simple policies
- Illustrate time consistency issues

Outline

1. Introduction
2. When to Worry About Adjustable Decisions
3. The Adjustable Robust Counterpart Model
4. Connections With Robust Dynamic Programming (DP)
5. Simple Policies and Their Optimality
6. Should One Worry About Time Consistency?
7. Conclusions and Future Directions

Table of Contents

A Simple Inventory Problem

Consider the following inventory management problem:

$$\begin{aligned}
\underset{x,y}{\text{minimize}} \quad & \sum_{t=1}^{T} \left(\overbrace{c_t x_t}^{\text{ordering cost}} + \overbrace{h_t (y_{t+1})^+}^{\text{holding cost}} + \overbrace{b_t (-y_{t+1})^+}^{\text{backlog cost}} \right) \\
\text{s.t.} \quad & y_{t+1} = y_t + x_t - d_t, \ \forall\, t, && \text{(Stock balance)} \\
& 0 \leqslant x_t \leqslant M_t, \ \forall\, t, && \text{(Min/max order size)} \\
& y_1 = a \,, && \text{(Initial stock level)}
\end{aligned}$$

where

- x_t is number of goods ordered at time t and received at $t+1$
- y_t is number of goods in stock at beginning of time t
- d_t is demand between time t and $t+1$
- a is the initial inventory

A Linear Programming Formulation

This problem can be reformulated using the linear program

$$\begin{aligned}
\underset{x,y,s^+,s^-}{\text{minimize}} \quad & \sum_{t=1}^{T} \left(c_t x_t + h_t s_t^+ + b_t s_t^- \right) \\
\text{s.t.} \quad & s_t^+ \geqslant 0,\ s_t^- \geqslant 0\,,\ \forall\, t, \\
& s_t^+ \geqslant y_{t+1}\,,\ \forall\, t, \\
& s_t^- \geqslant -y_{t+1}\,,\ \forall\, t, \\
& y_{t+1} = y_t + x_t - d_t\,,\ \forall\, t, \\
& 0 \leqslant x_t \leqslant M_t\,,\ \forall\, t,
\end{aligned}$$

where

- s_t^+ is amount of goods held in storage during stage t
- s_t^- is amount of backlogged customer demands during stage t

A Linear Programming Formulation

This problem can be reformulated using the linear program

$$
\begin{aligned}
\underset{x,y,s^+,s^-}{\text{minimize}} \quad & \sum_{t=1}^{T} \left(c_t x_t + h_t s_t^+ + b_t s_t^- \right) \\
\text{s.t.} \quad & s_t^+ \geqslant 0,\ s_t^- \geqslant 0\,,\ \forall\, t, \\
& s_t^+ \geqslant y_{t+1}\,,\ \forall\, t, \\
& s_t^- \geqslant -y_{t+1}\,,\ \forall\, t, \\
& y_{t+1} = y_t + x_t - d_t\,,\ \forall\, t, \\
& 0 \leqslant x_t \leqslant M_t\,,\ \forall\, t,
\end{aligned}
$$

How can we make this model robust to demand perturbations?

Naïve Robustification Scheme

Given that the vector of demand d is assumed to lie in some uncertainty set $\mathcal{U}$, let's consider the robust optimization model:

$$
\begin{aligned}
\underset{x,y,s^+,s^-}{\text{minimize}} \quad & \sum_{t=1}^{T} \left(c_t x_t + h_t s_t^+ + b_t s_t^- \right) \\
\text{s.t.} \quad & s_t^+ \geqslant 0,\ s_t^- \geqslant 0\,,\ \forall\, t \\
& s_t^+ \geqslant y_{t+1}\,,\ \forall\, t \\
& s_t^- \geqslant -y_{t+1}\,,\ \forall\, t \\
& y_{t+1} = y_t + x_t - d_t\,,\ \forall\, d \in \mathcal{U}\,,\ \forall\, t \\
& 0 \leqslant x_t \leqslant M_t\,,\ \forall\, t
\end{aligned}
$$

Naïve Robustification Scheme

Given that the vector of demand d is assumed to lie in some uncertainty set $\mathcal{U}$, let's consider the robust optimization model:

$$
\begin{array}{ll}
\underset{x,y,s^+,s^-}{\text{minimize}} & \sum_{t=1}^{T} \left(c_t x_t + h_t s_t^+ + b_t s_t^-\right) \\
\text{s.t.} & s_t^+ \geqslant 0,\ s_t^- \geqslant 0\,,\ \forall\, t \\
& s_t^+ \geqslant y_{t+1}\,,\ \forall\, t \\
& s_t^- \geqslant -y_{t+1}\,,\ \forall\, t \\
& y_{t+1} = y_t + x_t - d_t\,,\ \forall\, d \in \mathcal{U}\,,\ \forall\, t \\
& 0 \leqslant x_t \leqslant M_t\,,\ \forall\, t
\end{array}
$$

Unfortunately, this makes the model infeasible even when $|\mathcal{U}| = 2$:

$$
\left\{ \begin{array}{l} y_{t+1} = y_t + x_t - d_t^{(1)} \\ y_{t+1} = y_t + x_t - d_t^{(2)} \end{array} \right\} \Rightarrow\ d_t^{(1)} = d_t^{(2)}
$$

A Less Naïve Robustification Scheme

Robustify an alternate linear programming formulation:

$$\begin{aligned}
\underset{x,s^+,s^-}{\text{minimize}} \quad & \sum_{t=1}^{T} \left(c_t x_t + h_t s_t^+ + b_t s_t^- \right) \\
\text{s.t.} \quad & s_t^+ \geqslant 0,\ s_t^- \geqslant 0,\ \forall t, \\
& s_t^+ \geqslant y_1 + \sum_{t'=1}^{t} x_{t'} - d_{t'},\ \forall t, \\
& s_t^- \geqslant -y_1 + \sum_{t'=1}^{t} d_{t'} - x_{t'},\ \forall t, \\
& 0 \leqslant x_t \leqslant M_t \ \forall t\ ,
\end{aligned}$$

where we simply replaced $y_{t+1} := y_1 + \sum_{t'=1}^{t} x_{t'} - d_{t'}$ in order to capture the fact that stock level evolves according to demand.

A Less Naïve Robustification Scheme

Robustify an alternate linear programming formulation:

$$
\begin{aligned}
\underset{x,s^+,s^-}{\text{minimize}} \quad & \sum_{t=1}^{T} \left(c_t x_t + h_t s_t^+ + b_t s_t^- \right) \\
\text{s.t.} \quad & s_t^+ \geqslant 0,\ s_t^- \geqslant 0,\ \forall t, \\
& s_t^+ \geqslant y_1 + \sum_{t'=1}^{t} x_{t'} - d_{t'} \,,\ \forall\, d \in \mathcal{U} \,,\ \forall\, t, \\
& s_t^- \geqslant -y_1 + \sum_{t'=1}^{t} d_{t'} - x_{t'} \,,\ \forall\, d \in \mathcal{U} \,,\ \forall\, t, \\
& 0 \leqslant x_t \leqslant M_t, \forall\, t \,.
\end{aligned}
$$

A Less Naïve Robustification Scheme

Robustify an alternate linear programming formulation:

$$
\begin{aligned}
\underset{x,s^+,s^-}{\text{minimize}} \quad & \sum_{t=1}^{T} \left(c_t x_t + h_t s_t^+ + b_t s_t^- \right) \\
\text{s.t.} \quad & s_t^+ \geqslant 0,\ s_t^- \geqslant 0,\ \forall t, \\
& s_t^+ \geqslant y_1 + \sum_{t'=1}^{t} x_{t'} - d_{t'},\ \forall\, d \in \mathcal{U},\ \forall\, t, \\
& s_t^- \geqslant -y_1 + \sum_{t'=1}^{t} d_{t'} - x_{t'},\ \forall\, d \in \mathcal{U},\ \forall\, t, \\
& 0 \leqslant x_t \leqslant M_t, \forall\, t\ .
\end{aligned}
$$

Still two issues remain:

1. the orders should be adjustable w.r.t. the observed demand
2. (s_t^+, s_t^-) should be fully adjustable (more subtle)

Why (s_t^+, s_t^-) Should Be Fully Adjustable

Why (s_t^+, s_t^-) Should Be Fully Adjustable

Consider the two-stage problem:

<u>Deterministic model</u>:

$$
\begin{aligned}
\min_{x_1} \quad & \overbrace{0.5x_1}^{\text{ordering}} + \overbrace{(x_1 - d_1)^+}^{\text{holding}} \\
& \quad + \underbrace{(d_1 - x_1)^+}_{\text{backlog}} \\
\text{s.t.} \quad & 0 \leqslant x_1 \leqslant 2 \,,
\end{aligned}
$$

Why (s_t^+, s_t^-) Should Be Fully Adjustable

Consider the two-stage problem:

<u>Deterministic model</u>:

$$\begin{aligned} \min_{x_1} \quad & \overbrace{0.5x_1}^{\text{ordering}} + \overbrace{(x_1 - d_1)^+}^{\text{holding}} \\ & + \underbrace{(d_1 - x_1)^+}_{\text{backlog}} \\ \text{s.t.} \quad & 0 \leqslant x_1 \leqslant 2 \,, \end{aligned}$$

<u>Less naïve robust model</u>:

$$\begin{aligned} \min_{x_1, s_1^+, s_1^-} \quad & 0.5x_1 + s_1^+ + s_1^- \\ \text{s.t.} \quad & s_1^+ \geqslant 0,\ s_1^- \geqslant 0 \\ & s_1^+ \geqslant x_1 - d_1 \,,\ \forall\, d_1 \in [0,2] \\ & s_1^- \geqslant d_1 - x_1 \,,\ \forall\, d_1 \in [0,2] \\ & 0 \leqslant x_1 \leqslant 2 \,. \end{aligned}$$

Why (s_t^+, s_t^-) Should Be Fully Adjustable

Consider the two-stage problem:

Deterministic model:

$$\begin{aligned}\min_{x_1} \quad & \overbrace{0.5x_1}^{\text{ordering}} + \overbrace{(x_1 - d_1)^+}^{\text{holding}} \\ & + \underbrace{(d_1 - x_1)^+}_{\text{backlog}} \\ \text{s.t.} \quad & 0 \leqslant x_1 \leqslant 2\ ,\end{aligned}$$

Less naïve robust model:

$$\begin{aligned}\min_{x_1, s_1^+, s_1^-} \quad & 0.5x_1 + s_1^+ + s_1^- \\ \text{s.t.} \quad & s_1^+ \geqslant 0,\ s_1^- \geqslant 0 \\ & s_1^+ \geqslant x_1 \\ & s_1^- \geqslant 2 - x_1 \\ & 0 \leqslant x_1 \leqslant 2\ .\end{aligned}$$

Why (s_t^+, s_t^-) Should Be Fully Adjustable

Consider the two-stage problem:

Deterministic model:

$$\begin{aligned}\min_{x_1} \quad & \overbrace{0.5x_1}^{\text{ordering}} + \overbrace{(x_1 - d_1)^+}^{\text{holding}} \\ & + \underbrace{(d_1 - x_1)^+}_{\text{backlog}} \\ \text{s.t.} \quad & 0 \leqslant x_1 \leqslant 2 \, ,\end{aligned}$$

Less naïve robust model:

$$\begin{aligned}\min_{x_1} \quad & 0.5x_1 + \underbrace{x_1}_{s_1^{+*}} + \underbrace{2 - x_1}_{s_1^{-*}} \\ \text{s.t.} \quad & 0 \leqslant x_1 \leqslant 2 \, .\end{aligned}$$

Why (s_t^+, s_t^-) Should Be Fully Adjustable

Consider the two-stage problem:

Deterministic model:

$$\min_{x_1} \quad \overbrace{0.5x_1}^{\text{ordering}} + \overbrace{(x_1 - d_1)^+}^{\text{holding}} + \underbrace{(d_1 - x_1)^+}_{\text{backlog}}$$

$$\text{s.t.} \quad 0 \leqslant x_1 \leqslant 2 \ ,$$

Less naïve robust model:

$$\min_{x_1} \quad 0.5x_1 + \underbrace{x_1}_{s_1^{+*}} + \underbrace{2 - x_1}_{s_1^{-*}}$$

$$\text{s.t.} \quad 0 \leqslant x_1 \leqslant 2 \ .$$

Conclusions:

- Less naïve robust model states $x_1^* := 0$, $s_1^{+*} := 0$, and $s_1^{-*} := 2$ with worst-case cost of 2

Why (s_t^+, s_t^-) Should Be Fully Adjustable

Consider the two-stage problem:

Deterministic model:

$$\min_{x_1} \quad \overbrace{0.5x_1}^{\text{ordering}} + \overbrace{(x_1 - d_1)^+}^{\text{holding}} + \underbrace{(d_1 - x_1)^+}_{\text{backlog}}$$

$$\text{s.t.} \quad 0 \leqslant x_1 \leqslant 2\,,$$

Less naïve robust model:

$$\min_{x_1} \quad 0.5x_1 + \underbrace{x_1}_{s_1^{+*}} + \underbrace{2 - x_1}_{s_1^{-*}}$$

$$\text{s.t.} \quad 0 \leqslant x_1 \leqslant 2\,.$$

Conclusions:

- Less naïve robust model states $x_1^* := 0$, $s_1^{+*} := 0$, and $s_1^{-*} := 2$ with worst-case cost of 2
- Alternatively, $x_1^{**} := 1$ achieves a total cost lower than 1.5 for all $d_1 \in [0, 2]$

Why (s_t^+, s_t^-) Should Be Fully Adjustable

Consider the two-stage problem:

<u>Deterministic model</u>:

$$\min_{x_1} \quad \overbrace{0.5x_1}^{\text{ordering}} + \overbrace{|x_1 - d_1|}^{\text{holding+backlog}}$$

$$\text{s.t.} \quad 0 \leqslant x_1 \leqslant 2 \,,$$

<u>Less naïve robust model</u>:

$$\min_{x_1} \quad 0.5x_1 + \underbrace{x_1}_{s_1^{+*}} + \underbrace{2 - x_1}_{s_1^{-*}}$$

$$\text{s.t.} \quad 0 \leqslant x_1 \leqslant 2 \,.$$

Conclusions:

- Less naïve robust model states $x_1^* := 0$, $s_1^{+*} := 0$, and $s_1^{-*} := 2$ with worst-case cost of 2
- Alternatively, $x_1^{**} := 1$ achieves a total cost lower than 1.5 for all $d_1 \in [0, 2]$

An Accurate Two-stage Robust Inventory Model

The robust two-stage problem actually takes the form:

$$\begin{aligned} \underset{x_1}{\text{minimize}} \quad & \sup_{d_1 \in [0,2]} \; 0.5x_1 + h(x_1, d_1) \\ \text{s.t.} \quad & 0 \leqslant x_1 \leqslant 2 \,, \end{aligned}$$

An Accurate Two-stage Robust Inventory Model

The robust two-stage problem actually takes the form:

$$\begin{aligned} \underset{x_1}{\text{minimize}} \quad & \sup_{d_1 \in [0,2]} \; 0.5x_1 + h(x_1, d_1) \\ \text{s.t.} \quad & 0 \leqslant x_1 \leqslant 2 \,, \end{aligned}$$

where

$$\begin{aligned} h(x_1, d_1) := \quad \min_{s_1^+, s_1^-} \quad & s_1^+ + s_1^- \\ \text{s.t.} \quad & s_1^+ \geqslant 0, \; s_1^- \geqslant 0 \\ & s_1^+ \geqslant x_1 - d_1 \\ & s_1^- \geqslant -x_1 + d_1 \,. \end{aligned}$$

An Accurate Two-stage Robust Inventory Model

The robust two-stage problem actually takes the form:

$$\begin{aligned} \underset{x_1}{\text{minimize}} \quad & \sup_{d_1 \in [0,2]} \; 0.5x_1 + h(x_1, d_1) \\ \text{s.t.} \quad & 0 \leqslant x_1 \leqslant 2 \;, \end{aligned}$$

where

$$\begin{aligned} h(x_1, d_1) := \quad \min_{s_1^+, s_1^-} \quad & s_1^+ + s_1^- \\ \text{s.t.} \quad & s_1^+ \geqslant 0,\ s_1^- \geqslant 0 \\ & s_1^+ \geqslant x_1 - d_1 \\ & s_1^- \geqslant -x_1 + d_1 \;. \end{aligned}$$

- Note how this form accounts for the fact that s_1^+ and s_1^- might jointly depend on the realized value of d_1.

Alternate Representation of Less Naïve Robust Model

Comparatively, the less naïve robust model was solving:

$$\begin{aligned} \underset{x_1, s_1^+, s_1^-}{\text{minimize}} \quad & \sup_{d_1 \in [0,2]} \; 0.5x_1 + g(x_1, s_1^+, s_1^-, d_1) \\ \text{s.t.} \quad & s_1^+ \geqslant 0, \; s_1^- \geqslant 0 \\ & 0 \leqslant x_1 \leqslant 2 \; , \end{aligned}$$

where

$$g(x_1, s_1^+, s_1^-, d_1) := \begin{cases} s_1^+ + s_1^- & \text{if } s_1^+ \geqslant x_1 - d_1 \text{ and } s_1^- \geqslant d_1 - x_1 \\ \infty & \text{otherwise} \end{cases} \quad .$$

Alternate Representation of Less Naïve Robust Model

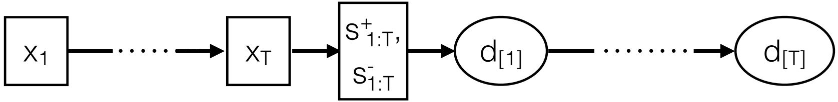
Comparatively, the less naïve robust model was solving:

$$\begin{aligned}
\underset{x_1, s_1^+, s_1^-}{\text{minimize}} \quad & \sup_{d_1 \in [0,2]} \; 0.5x_1 + g(x_1, s_1^+, s_1^-, d_1) \\
\text{s.t.} \quad & s_1^+ \geqslant 0,\; s_1^- \geqslant 0 \\
& 0 \leqslant x_1 \leqslant 2 \,,
\end{aligned}$$

where

$$g(x_1, s_1^+, s_1^-, d_1) := \begin{cases} s_1^+ + s_1^- & \text{if } s_1^+ \geqslant x_1 - d_1 \text{ and } s_1^- \geqslant d_1 - x_1 \\ \infty & \text{otherwise} \end{cases} \quad .$$

Timing characterized by less naïve robust model



Takeaway Message about Adjustable Decisions

When robustifying decision models that either involve

- "implementable" decisions at different time periods (e.g. x_t)
- "auxiliary" decisions such as (s_t^+, s_t^-) that are used to assess overall performance of implemented decisions

one needs to carefully identify the chronology of decisions and observations and employ the adjustable robust counterpart framework introduced in (Ben-Tal et al., 2004)

Accurate timing of decisions and observations in inventory problem

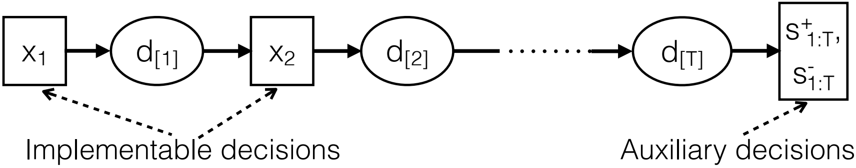


Table of Contents

Applying the ARO Framework to Inventory Management

Applying the ARO Framework to Inventory Management

1. Formulate a deterministic model where implementable and auxiliary decisions can be identified:

$$
\begin{aligned}
\underset{x_t, s_t^+, s_t^-}{\text{minimize}} \quad & \sum_{t=1}^{T} \left(c_t x_t + h_t s_t^+ + b_t s_t^- \right) \\
\text{s.t.} \quad & s_t^+ \geqslant 0,\ s_t^- \geqslant 0,\ \forall t, \\
& s_t^+ \geqslant y_1 + \sum_{t'=1}^{t} x_{t'} - d_{t'},\ \forall\, t, \\
& s_t^- \geqslant -y_1 + \sum_{t'=1}^{t} d_{t'} - x_{t'},\ \forall\, t, \\
& 0 \leqslant x_t \leqslant M_t, \forall\, t\,.
\end{aligned}
$$

Applying the ARO Framework to Inventory Management

1. Formulate a deterministic model where implementable and auxiliary decisions can be identified:

$$
\begin{aligned}
\underset{x_t, s_t^+, s_t^-}{\text{minimize}} \quad & \sum_{t=1}^{T} \left(c_t x_t + h_t s_t^+ + b_t s_t^- \right) \\
\text{s.t.} \quad & s_t^+ \geqslant 0,\ s_t^- \geqslant 0,\ \forall t, \\
& s_t^+ \geqslant y_1 + \sum_{t'=1}^{t} x_{t'} - d_{t'},\ \forall\, t, \\
& s_t^- \geqslant -y_1 + \sum_{t'=1}^{t} d_{t'} - x_{t'},\ \forall\, t, \\
& 0 \leqslant x_t \leqslant M_t, \forall\, t\,.
\end{aligned}
$$

2. Identify an accurate chronology:

$$
x_1 \ \rightarrow\ d_{[1]}\ \rightarrow\ x_2\ \rightarrow \cdots \rightarrow\ x_T\ \rightarrow\ d_{[T]}\ \rightarrow\ (s_{1:T}^+, s_{1:T}^-)
$$

Applying the ARO Framework to Inventory Management

3. Describe the adjustable robust counterpart

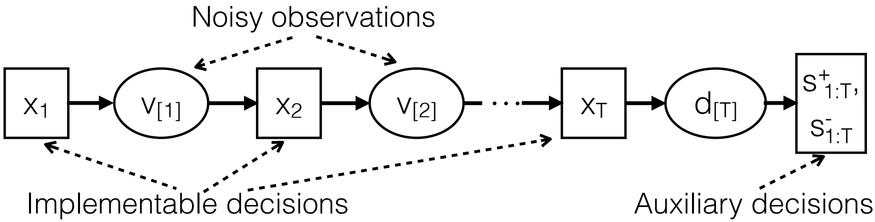
$$\begin{aligned}
\min_{\substack{x_1, \{x_t(\cdot)\}_{t=2}^T, \\ \{s_t^+(\cdot), s_t^-(\cdot)\}_{t=1}^T}} \quad & \sup_{d \in \mathcal{U}} \sum_{t=1}^{T} \left(c_t x_t(d_{[t-1]}) + h_t s_t^+(d) + b_t s_t^-(d) \right) \\
\text{s.t.} \quad & s_t^+(d) \geqslant 0,\ s_t^-(d) \geqslant 0\,,\ \forall\, d \in \mathcal{U},\ \ \forall\, t \\
& s_t^+(d) \geqslant y_1 + \sum_{t'=1}^{t} x_{t'}(d_{[t'-1]}) - d_{t'}\,,\ \forall\, d \in \mathcal{U},\ \ \forall\, t \\
& s_t^-(d) \geqslant -y_1 + \sum_{t'=1}^{t} d_{t'} - x_{t'}(d_{[t'-1]})\,,\ \forall\, d \in \mathcal{U},\ \ \forall\, t \\
& 0 \leqslant x_t(d_{[t-1]}) \leqslant M_t\,,\ \forall\, d \in \mathcal{U},\ \ \forall\, t\,.
\end{aligned}$$

where

- Each $x_t : \mathbb{R}^{t-1} \to \mathbb{R}$ is adjusted to the observed demand $d_{[t-1]}$
- Each $s_t^+ : \mathbb{R}^T \to \mathbb{R}$ and $s_t^- : \mathbb{R}^T \to \mathbb{R}$ are adjusted to entire d

Robust Inventory Management with Noisy Observations

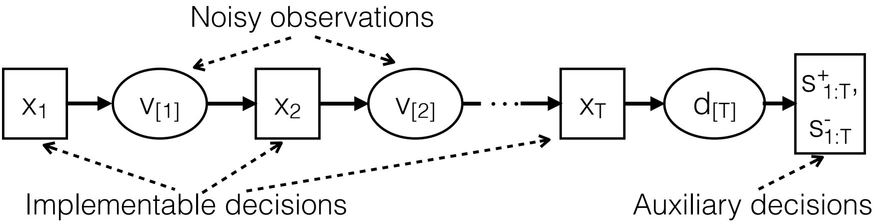
In some situations, the observations that are made are noisy.



where

Robust Inventory Management with Noisy Observations

In some situations, the observations that are made are noisy.

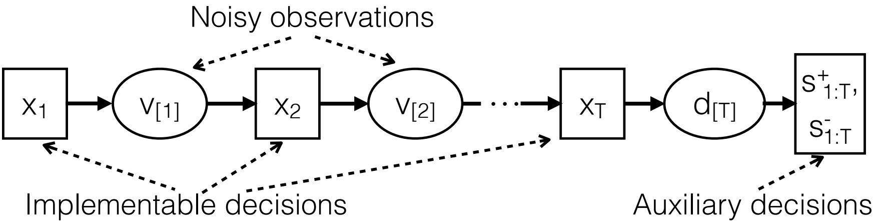


where

- each $s_t^+ : \mathbb{R}^T \to \mathbb{R}$ and $s_t^- : \mathbb{R}^T \to \mathbb{R}$ are still adjusted to entire $\mathbf{d}$

Robust Inventory Management with Noisy Observations

In some situations, the observations that are made are noisy.

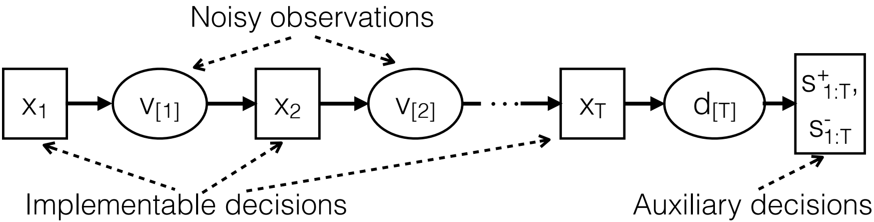


where

- each $s_t^+ : \mathbb{R}^T \to \mathbb{R}$ and $s_t^- : \mathbb{R}^T \to \mathbb{R}$ are still adjusted to entire d
- each $x_t : \mathbb{R}^{t-1} \to \mathbb{R}$ is adjusted to the observations $v_{[t]}$

Robust Inventory Management with Noisy Observations

In some situations, the observations that are made are noisy.

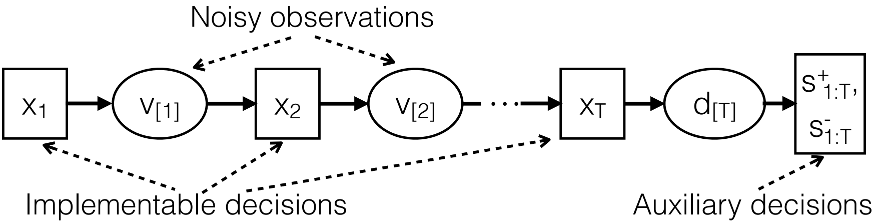


where

- each $s_t^+ : \mathbb{R}^T \to \mathbb{R}$ and $s_t^- : \mathbb{R}^T \to \mathbb{R}$ are still adjusted to entire d
- each $x_t : \mathbb{R}^{t-1} \to \mathbb{R}$ is adjusted to the observations $v_{[t]}$
- each v_t captures what is observed right before ordering x_t
  e.g. $v_t = d_t - \xi_t$ with $\xi_t \in [-d_t, d_t]$., with $\sum_t |\xi_t| \leqslant \Gamma$.

Robust Inventory Management with Noisy Observations

In some situations, the observations that are made are noisy.



where

- each $s_t^+ : \mathbb{R}^T \to \mathbb{R}$ and $s_t^- : \mathbb{R}^T \to \mathbb{R}$ are still adjusted to entire d
- each $x_t : \mathbb{R}^{t-1} \to \mathbb{R}$ is adjusted to the observations $v_{[t]}$
- each v_t captures what is observed right before ordering x_t
  e.g. $v_t = d_t - \xi_t$ with $\xi_t \in [-d_t, d_t]$., with $\sum_t |\xi_t| \leqslant \Gamma$.

See (de Ruiter et al., 2014) for detailed methodology.

Robust Inventory Management with Noisy Observations

$$
\begin{aligned}
\min_{\substack{x_1,\{x_t(\cdot)\}_{t=2}^T,\\ \{s_t^+(\cdot),\, s_t^-(\cdot)\}_{t=1}^T}} \quad & \sup_{(d,v)\in\mathcal{U}'} \; c_1 x_1 + \sum_t \left(c_t x_t(v_{[t]}) + h_t s_t^+(d) + b_t s_t^-(d) \right) \\
\text{s.t.} \quad & s_t^+(d) \geqslant 0,\; s_t^-(d) \geqslant 0\,,\; \forall\, d \in \mathcal{U},\;\; \forall\, t \\
& s_t^+(d) \geqslant y_1 + \sum_{t'=1}^{t} x_{t'}(v_{[t']}) - d_{t'}\,,\; \forall\,(d,v) \in \mathcal{U}',\;\; \forall\, t \\
& s_t^-(d) \geqslant -y_1 + \sum_{t'=1}^{t} d_{t'} - x_{t'}(v_{[t']})\,,\; \forall\,(d,v) \in \mathcal{U}',\;\; \forall\, t \\
& 0 \leqslant x_t(v_{[t-1]}) \leqslant M_t\,,\; \forall\,(d,v) \in \mathcal{U}',\;\; \forall\, t\,,
\end{aligned}
$$

where

$$
\mathcal{U}' := \left\{ (d,v) \in \mathcal{U} \times \mathbb{R}^T \;\middle|\; \exists\, \xi \in \mathbb{R}_+^T,\; v_t = d_t - \xi_t,\; |\xi_t| \leqslant d_t,\; \sum_t |\xi_t| \leqslant \Gamma \right\}
$$

NP-Hardness of Adjustable Robust Optimization

Theorem (Ben-Tal et al., 2004):

Solving a robust multi-stage linear programming model is NP-hard.

NP-Hardness of Adjustable Robust Optimization

Theorem (Ben-Tal et al., 2004):

Solving a robust multi-stage linear programming model is NP-hard.

Sketch of proof:

- Consider the two-stage ARO model where $x : \mathbb{R}^m \to \mathbb{R}^N$:

$$\begin{aligned} \underset{x(\cdot)}{\text{minimize}} \quad & \sup_{u \in [0,1]^m} \sum_{i=1}^{N} (x_i(u) - 1) \\ \text{s.t.} \quad & x_i(u) \geqslant a_{i,k}^\top u + b_{i,k}\,,\ \forall\, u \in [0,1]^m\,, \begin{cases} \forall\, i = 1, \ldots, N\,, \\ \forall\, k = 1, 2, 3. \end{cases} \end{aligned}$$

NP-Hardness of Adjustable Robust Optimization

Theorem (Ben-Tal et al., 2004):

Solving a robust multi-stage linear programming model is NP-hard.

Sketch of proof:

- Consider the two-stage ARO model where $x : \mathbb{R}^m \to \mathbb{R}^N$:

$$\begin{aligned} \underset{x(\cdot)}{\text{minimize}} \quad & \sup_{u\in[0,1]^m} \sum_{i=1}^{N} (x_i(u) - 1) \\ \text{s.t.} \quad & x_i(u) \geqslant a_{i,k}^\top u + b_{i,k}\,,\ \forall\, u \in [0,1]^m\,, \begin{cases} \forall\, i = 1, \ldots, N\,, \\ \forall\, k = 1, 2, 3. \end{cases} \end{aligned}$$

- Checking if optimal value is $\geqslant 0$ can be as difficult as a 3-SAT problem with m variables and N clauses like $v_{j_1} \vee v_{j_2} \vee \bar{v}_{j_3}$.
  - E.g., choose (a, b) such that:

$$\begin{aligned} x_i^*(z) &= \max_k a_{i,k}^\top u + b_{i,k} = \max\{u_{j_1}; u_{j_2}; 1 - u_{j_3}\} \\ &= 1 \text{ if } u_{j_1} = 1,\ u_{j_2} = 1, \text{ or } u_{j_3} = 0, \text{ otherwise it's } < 1. \end{aligned}$$

Table of Contents

Connections With Robust Dynamic Programming

Our dynamic decision problem can also be written:

Connections With Robust Dynamic Programming

Our dynamic decision problem can also be written:

$$\min_{0 \leqslant x_1 \leqslant M_1} \Bigg[c_1 x_1 + \max_{d_1 \in \mathcal{U}_1(\emptyset)} \Big[h_1(y_2)^+ + b_1(-y_2)^+$$

$$+ \min_{0 \leqslant x_2 \leqslant M_2} \Big[c_2 x_2 + \max_{d_2 \in \mathcal{U}_2(d_1)} \Big[h_2(y_3)^+ + b_2(-y_3)^+ + \dots$$

$$+ \min_{0 \leqslant x_t \leqslant M_t} \Big[c_T x_T + \max_{d_T \in \mathcal{U}_T(d_{[T-1]})} [h_T(y_{T+1})^+ + b_T(-y_{T+1})^+] \Big] \dots \Big] \Bigg]$$

Connections With Robust Dynamic Programming

Our dynamic decision problem can also be written:

$$\min_{0\leqslant x_1\leqslant M_1}\Bigg[c_1x_1 + \max_{d_1\in\mathcal{U}_1(\emptyset)}\Big[h_1(y_2)^+ + b_1(-y_2)^+$$

$$+ \min_{0\leqslant x_2\leqslant M_2}\Big[c_2x_2 + \max_{d_2\in\mathcal{U}_2(d_1)}\Big[h_2(y_3)^+ + b_2(-y_3)^+ + \dots$$

$$+ \min_{0\leqslant x_t\leqslant M_t}\Big[c_Tx_T + \max_{d_T\in\mathcal{U}_T(d_{[T-1]})}[h_T(y_{T+1})^+ + b_T(-y_{T+1})^+]\Big]\dots\Big]\Bigg]$$

where:

$$y_{t+1} := y_t + x_t - d_t$$

$$\mathcal{U}_t(d_{[t-1]}) := \left\{d\in\mathbb{R} : \exists\, \xi\in\mathbb{R}^{T-t} \text{ such that } [d_{[t-1]}^\top\ d\ \xi^\top]^\top \in \mathcal{U}\right\}$$

Connections With Robust Dynamic Programming

Our dynamic decision problem can also be written:

$$\min_{0 \leqslant x_1 \leqslant M_1} \Bigg[c_1 x_1 + \max_{d_1 \in \mathcal{U}_1(\emptyset)} \Big[h_1 (y_2)^+ + b_1(-y_2)^+$$

$$+ \min_{0 \leqslant x_2 \leqslant M_2} \Big[c_2 x_2 + \max_{d_2 \in \mathcal{U}_2(d_1)} \Big[h_2 (y_3)^+ + b_2(-y_3)^+ + \ldots$$

$$+ \min_{0 \leqslant x_t \leqslant M_t} \Big[c_T x_T + \max_{d_T \in \mathcal{U}_T(d_{[T-1]})} [h_T (y_{T+1})^+ + b_T(-y_{T+1})^+] \Big] \ldots \Big] \Bigg]$$

where:

$$y_{t+1} := y_t + x_t - d_t$$

$$\mathcal{U}_t(d_{[t-1]}) := \Big\{ d \in \mathbb{R} : \exists\, \xi \in \mathbb{R}^{T-t} \text{ such that } [d_{[t-1]}^\top \; d \; \xi^\top]^\top \in \mathcal{U} \Big\}$$

- nested min-max problems
- proper updating rule: *projection* (analogous to "conditioning")

Bellman Principle; Robust DP Recursions

- The *state* of the system at time t:

$$S_t := \left[y_t \;\; d_{[t-1]}^\top\right]^\top = \left[y_t \;\; d_1 \;\; d_2 \;\; \dots \;\; d_{t-1}\right]^\top \in \mathbb{R}^t$$

Bellman Principle; Robust DP Recursions

- The *state* of the system at time t:

$$S_t := \begin{bmatrix} y_t & d_{[t-1]}^\top \end{bmatrix}^\top = \begin{bmatrix} y_t & d_1 & d_2 & \dots & d_{t-1} \end{bmatrix}^\top \in \mathbb{R}^t$$

- Value function $J_t^*(S_t)$ given by:

$$J_t^*(S_t) = \min_{0 \leqslant x_t \leqslant M_t} \left[c_t x_t + \max_{d_t \in \mathcal{U}_t(d_{[t-1]})} \left[h_t (y_{t+1})^+ + b_t (-y_{t+1})^+ + J_{t+1}^*(S_{t+1}) \right] \right]$$

Bellman Principle; Robust DP Recursions

- The *state* of the system at time t:

$$S_t := \left[y_t \;\; d_{[t-1]}^\top \right]^\top = \left[y_t \;\; d_1 \;\; d_2 \;\; \ldots \;\; d_{t-1} \right]^\top \in \mathbb{R}^t$$

- Value function $J_t^*(S_t)$ given by:

$$J_t^*(S_t) = \min_{0 \leqslant x_t \leqslant M_t} \left[c_t x_t + \max_{d_t \in \mathcal{U}_t(d_{[t-1]})} \left[h_t (y_{t+1})^+ + b_t (-y_{t+1})^+ + J_{t+1}^*(S_{t+1}) \right] \right]$$

Observations:

1. General $\mathcal{U} \longrightarrow$ high-dimensional $S_t \longrightarrow$ curse of dimensionality

Bellman Principle; Robust DP Recursions

- The *state* of the system at time t:

$$S_t := \left[y_t \;\; d_{[t-1]}^\top\right]^\top = \left[y_t \;\; d_1 \;\; d_2 \;\; \ldots \;\; d_{t-1}\right]^\top \in \mathbb{R}^t$$

- Value function $J_t^*(S_t)$ given by:

$$J_t^*(S_t) = \min_{0 \leq x_t \leq M_t} \left[c_t x_t + \max_{d_t \in \mathcal{U}_t(d_{[t-1]})} \left[h_t (y_{t+1})^+ + b_t (-y_{t+1})^+ + J_{t+1}^*(S_{t+1}) \right] \right]$$

Observations:

1. General $\mathcal{U} \longrightarrow$ high-dimensional $S_t \longrightarrow$ curse of dimensionality
2. When $\mathcal{U}$ has special structure, can reduce state space

Bellman Principle; Robust DP Recursions

- The *state* of the system at time t:

$$S_t := \left[y_t \;\; d_{[t-1]}^\top\right]^\top = \left[y_t \;\; d_1 \;\; d_2 \;\; \dots \;\; d_{t-1}\right]^\top \in \mathbb{R}^t$$

- Value function $J_t^*(S_t)$ given by:

$$J_t^*(S_t) = \min_{0 \leqslant x_t \leqslant M_t}\left[c_t x_t + \max_{d_t \in \mathcal{U}_t(d_{[t-1]})}\left[h_t(y_{t+1})^+ + b_t(-y_{t+1})^+ + J_{t+1}^*(S_{t+1})\right]\right]$$

Observations:

1. General $\mathcal{U} \longrightarrow$ high-dimensional $S_t \longrightarrow$ curse of dimensionality
2. When $\mathcal{U}$ has special structure, can reduce state space

$$\mathcal{U}_{\text{h-cube}} = \times_{t=1}^{T}[\underline{d}_t, \bar{d}_t] \quad \rightarrow \quad S_t = y_t$$

$$\mathcal{U}_{\text{budget}} = \left\{d : \exists\, z, \|z\|_\infty \leqslant 1, \|z\|_1 \leqslant \Gamma, d_t = \bar{d}_t + \hat{d}_t z_t\right\} \rightarrow S_t = \left[y_t, \sum_{\tau=1}^{t-1} |z_\tau|\right]$$

Bellman Principle; Robust DP Recursions

- The *state* of the system at time t:

$$S_t := \left[y_t \ d_{[t-1]}^\top\right]^\top = \left[y_t \ d_1 \ d_2 \ \ldots \ d_{t-1}\right]^\top \in \mathbb{R}^t$$

- Value function $J_t^*(S_t)$ given by:

$$J_t^*(S_t) = \min_{0 \leqslant x_t \leqslant M_t} \left[c_t x_t + \max_{d_t \in \mathcal{U}_t(d_{[t-1]})} \left[h_t (y_{t+1})^+ + b_t (-y_{t+1})^+ + J_{t+1}^*(S_{t+1}) \right] \right]$$

Observations:

1. General $\mathcal{U} \longrightarrow$ high-dimensional $S_t \longrightarrow$ curse of dimensionality
2. When $\mathcal{U}$ has special structure, can reduce state space
   - Reduce computational burden
   - Prove structural results, comparative statics

$$x_t^*(y) = \min\big(M_t, \max(0, \ \theta_t - y)\big) \qquad \text{(modified) } \textit{base-stock policy}$$

A More Subtle Point...

A More Subtle Point...

Consider this problem:

$$J^* = \max_{d \in [\underline{d}, \bar{d}]} \min_{0 \leqslant x} f(d, x)$$

$$\big(y_1 \to d_1 \equiv d \to y_2 \equiv y(d) \to x_2 \equiv x\big)$$

A More Subtle Point...

Consider this problem:

$$J^* = \max_{d \in [\underline{d}, \bar{d}]} \min_{0 \leqslant x} f(d, x)$$

- DP yields *Bellman-optimal* policy $x^*(d) = \max\big(0, \theta^* - y(d)\big)$

A More Subtle Point...

Consider this problem:

$$J^* = \max_{d \in [\underline{d}, \bar{d}]} \min_{0 \leqslant x} f(d, x)$$

- DP yields *Bellman-optimal* policy $x^*(d) = \max\big(0, \theta^* - y(d)\big)$
- $x^*(d)$ minimizes $f(x, d)$ *for any* d

A More Subtle Point...

Consider this problem:

$$J^* = \max_{d \in [\underline{d}, \bar{d}]} \min_{0 \leqslant x} f(d, x)$$

- DP yields *Bellman-optimal* policy $x^*(d) = \max\big(0, \theta^* - y(d)\big)$
- $x^*(d)$ minimizes $f(x, d)$ *for any* d
- $x^*(d)$ *sufficient* for achieving J^*, i.e., $J^* = \max_{d \in [\underline{d}, \bar{d}]} f\big(d, x^*(d)\big)$

A More Subtle Point...

Consider this problem:

$$J^* = \max_{d \in [\underline{d}, \bar{d}]} \min_{0 \leqslant x} f(d, x)$$

- DP yields *Bellman-optimal* policy $x^*(d) = \max\big(0, \theta^* - y(d)\big)$
- $x^*(d)$ minimizes $f(x, d)$ *for any* d
- $x^*(d)$ *sufficient* for achieving J^*, i.e., $J^* = \max_{d \in [\underline{d}, \bar{d}]} f\big(d, x^*(d)\big)$

Is $x^*(d)$ necessary for achieving J^*?
Is Bellman optimality necessary in robust dynamic problems?

Is Bellman Optimality **Necessary?**

$$J^* = \max_{d \in [\underline{d}, \bar{d}]} \min_{0 \leqslant x} f(d, x)$$

$\boxed{\text{Is } x^*(d) \text{ necessary for achieving } J^*?}$

Is Bellman Optimality **Necessary?**

$$J^* = \max_{d \in [\underline{d}, \bar{d}]} \min_{0 \leqslant x} f(d, x)$$

$$\boxed{\text{Is } x^*(d) \text{ necessary for achieving } J^*?}$$

Define the set of *worst-case optimal* policies:

$$\mathcal{X}^{\mathrm{wc}} := \left\{ x : [\underline{d}, \bar{d}] \to \mathbb{R}^+ \ : \ f\big(d, x(d)\big) \leqslant J^*, \ \forall\, d \in \mathcal{U} \right\}.$$

- $\mathcal{X}^{\mathrm{wc}}$ non-empty : $x^* \in \mathcal{X}^{\mathrm{WC}}$

Is Bellman Optimality **Necessary?**

$$J^* = \max_{d \in [\underline{d}, \bar{d}]} \min_{0 \leqslant x} f(d, x)$$

$$\boxed{\text{Is } x^*(d) \text{ necessary for achieving } J^*?}$$

Define the set of *worst-case optimal* policies:

$$\mathcal{X}^{\mathsf{wc}} := \left\{ x : [\underline{d}, \bar{d}] \to \mathbb{R}^+ \;:\; f\big(d, x(d)\big) \leqslant J^*, \; \forall\, d \in \mathcal{U} \right\}.$$

- $\mathcal{X}^{\mathsf{wc}}$ non-empty : $x^* \in \mathcal{X}^{\mathsf{wc}}$
- $\mathcal{X}^{\mathsf{wc}}$ contains other policies:

$$x^{\mathsf{aff}}(d) = x^*(\underline{d}) + \frac{x^*(\bar{d}) - x^*(\underline{d})}{\bar{d} - \underline{d}}(d - \underline{d})$$

Is Bellman Optimality **Necessary?**

$$J^* = \max_{d \in [\underline{d}, \bar{d}]} \min_{0 \leqslant x} f(d, x)$$

$$\boxed{\text{Is } x^*(d) \text{ necessary for achieving } J^*?}$$

Define the set of *worst-case optimal* policies:

$$\mathcal{X}^{\text{wc}} := \left\{ x : [\underline{d}, \bar{d}] \to \mathbb{R}^+ \; : \; f\big(d, x(d)\big) \leqslant J^*, \; \forall\, d \in \mathcal{U} \right\}.$$

- $\mathcal{X}^{\text{wc}}$ non-empty : $x^* \in \mathcal{X}^{\text{wc}}$
- $\mathcal{X}^{\text{wc}}$ contains other policies:

$$x^{\text{aff}}(d) = x^*(\underline{d}) + \frac{x^*(\bar{d}) - x^*(\underline{d})}{\bar{d} - \underline{d}} (d - \underline{d})$$

- In fact, $\mathcal{X}^{\text{wc}}$ very degenerate: infinitely many policies

  (e.g., $\lambda\, x^* + (1 - \lambda) x^{\text{aff}}$, for any $\lambda \in [0, 1]$)

Worst-Case Optimality and Degeneracy

- This degeneracy is typical for robust multi-stage problems
  ("If adversary does not play optimally, you don't have to, either...")

Worst-Case Optimality and Degeneracy

- This degeneracy is typical for robust multi-stage problems
  ("If adversary does not play optimally, you don't have to, either...")

- Critically different from stochastic problems

Worst-Case Optimality and Degeneracy

- This degeneracy is typical for robust multi-stage problems
  ("If adversary does not play optimally, you don't have to, either...")

- Critically different from stochastic problems

- A **blessing**: may allow finding policies with *simple structure*
  (Bertsimas et al. [2010], Iancu et al. [2013])

Worst-Case Optimality and Degeneracy

- This degeneracy is typical for robust multi-stage problems
  ("If adversary does not play optimally, you don't have to, either...")

- Critically different from stochastic problems

- A **blessing**: may allow finding policies with *simple structure*
  (Bertsimas et al. [2010], Iancu et al. [2013])

- A **curse**: may yield inefficiencies in the decision process
  (x^* Pareto-dominates x^{aff})

Worst-Case Optimality and Degeneracy

- This degeneracy is typical for robust multi-stage problems
  ("If adversary does not play optimally, you don't have to, either...")

- Critically different from stochastic problems

- A **blessing**: may allow finding policies with *simple structure*
  (Bertsimas et al. [2010], Iancu et al. [2013])

- A **curse**: may yield inefficiencies in the decision process
  (x^* Pareto-dominates x^{aff})

- Worst-case optimal policies *must be implemented with resolving*

Table of Contents

Static Policies

- Simplest possible policies : static (i.e., non-adjustable)
- Very tractable
- Optimal for LPs with *row-wise* uncertainty [Ben-Tal et al., 2009]

$$\min_{x(z)} \quad \max_{z_0 \in \mathcal{Z}_0} \left[c(z_0)^\top x(z) + d(z_0) \right]$$
$$\text{subject to} \quad a_j(z_j)^\top x(z) \leqslant b_j(z_j), \ \forall z_j \in \mathcal{Z}_j, \ \forall j = 1, \ldots, J.$$

- Result extended recently [Bertsimas et al., 2015]
- Good performance under other uncertainty sets [Bertsimas and Goyal, 2010, Bertsimas et al., 2011b, ...]

Affine Decision Rules

- Restrict attention to *affine* decision rules [Ben-Tal et al., 2004]

Affine Decision Rules

- Restrict attention to *affine* decision rules [Ben-Tal et al., 2004]
- *Affinely Adjustable Robust Counterpart* approximation scheme

$$
\begin{aligned}
\min_{\mathcal{X}} \quad & \max_{d \in \mathcal{U}} \sum_{t=1}^{T} c_t(x_t^0 + X_t d) + h_t(s_t^+ + S_t^+ d) + b_t(s_t^- + S_t^- d) \\
\text{s.t.} \quad & s_t^+ + S_t^+ d \geqslant 0, \quad s_t^- + S_t^- d \geqslant 0, \ \forall\, d \in \mathcal{U} \\
& s_t^+ + S_t^+ d \geqslant y_1 + \sum_{\tau=1}^{t} (x_\tau^0 + X_\tau d - d_\tau), \ \forall\, d \in \mathcal{U}, \\
& s_t^- + S_t^- d \geqslant -y_1 - \sum_{\tau=1}^{t} (x_\tau^0 + X_\tau d - d_\tau), \ \forall\, d \in \mathcal{U}, \\
& 0 \leqslant x_t + X_t d \leqslant M_t\,, \ \forall\, d \in \mathcal{U},
\end{aligned}
$$

- Decision variables: coefficients $\mathcal{X} = \left\{x_t^0, X_t, s_t^+, S_t^+, s_t^-, S_t^-\right\}_{t=1}^{T}$
  - $X_t \in \mathbb{R}^{t-1} \times 0^{T-t+1}$ to ensure non-anticipativity

Affine Decision Rules

- Restrict attention to *affine* decision rules [Ben-Tal et al., 2004]
- *Affinely Adjustable Robust Counterpart* approximation scheme

$$\begin{aligned}
\min_{\mathcal{X}} \quad & \max_{d \in \mathcal{U}} \sum_{t=1}^{T} c_t(x_t^0 + X_t d) + h_t(s_t^+ + S_t^+ d) + b_t(s_t^- + S_t^- d) \\
\text{s.t.} \quad & s_t^+ + S_t^+ d \geqslant 0, \quad s_t^- + S_t^- d \geqslant 0, \ \forall\, d \in \mathcal{U} \\
& s_t^+ + S_t^+ d \geqslant y_1 + \sum_{\tau=1}^{t} (x_\tau^0 + X_\tau d - d_\tau), \ \forall\, d \in \mathcal{U}, \\
& s_t^- + S_t^- d \geqslant -y_1 - \sum_{\tau=1}^{t} (x_\tau^0 + X_\tau d - d_\tau), \ \forall\, d \in \mathcal{U}, \\
& 0 \leqslant x_t + X_t d \leqslant M_t\,, \ \forall\, d \in \mathcal{U},
\end{aligned}$$

- **Two** layers of sub-optimality: **policy** *and* **auxiliary variables**

Affine Decision Rules

- Restrict attention to *affine* decision rules [Ben-Tal et al., 2004]
- *Affinely Adjustable Robust Counterpart* approximation scheme
- Tractable under fixed recourse [Ben-Tal et al., 2004]
- Excellent performance in a variety of applications [Ben-Tal et al., 2005, Mani et al., 2006, Adida and Perakis, 2006, Babonneau et al., 2010, ...]
- Optimal for linear problems with simplex uncertainty [Ben-Tal et al., 2009]
- Optimal for our multi-period inventory model under hypercube uncertainty set $\mathcal{U} = \times_{t=1}^{T} [\underline{d}_t, \bar{d}_t]$ [Iancu et al., 2013, Bertsimas et al., 2010]

Piecewise Affine Rules

- One can also use piece-wise affine rules, e.g.,

$$x_t(d_{[t-1]}) := x_t^0 + X_t d_{[t-1]} + \sum_{k=1}^{t-1} \theta_{tk}^{+}(d_k)^{+} + \theta_{tk}^{-}(-d_k)^{+}$$

Piecewise Affine Rules

- One can also use piece-wise affine rules, e.g.,

$$x_t(d_{[t-1]}) := x_t^0 + X_t d_{[t-1]} + \sum_{k=1}^{t-1} \theta_{tk}^+ (d_k)^+ + \theta_{tk}^- (-d_k)^+$$

- These policies are actually affine in $\xi := [d, d^+, d^-]$
  - It would suffice to have a tractable representation for:

$$\text{ConvexHull}\Big(\{(d, d^+, d^-) \mid d \in \mathcal{U},\ d^+ = \max(0; d),\ d^- = \max(0; -d)\}\Big)$$

Piecewise Affine Rules

- One can also use piece-wise affine rules, e.g.,

$$x_t(d_{[t-1]}) := x_t^0 + X_t d_{[t-1]} + \sum_{k=1}^{t-1} \theta_{tk}^+ (d_k)^+ + \theta_{tk}^- (-d_k)^+$$

- These policies are actually affine in $\xi := [d, d^+, d^-]$
  - It would suffice to have a tractable representation for:

$$\text{ConvexHull}\Big(\{(d, d^+, d^-) \,|\, d \in \mathcal{U},\ d^+ = \max(0; d),\ d^- = \max(0; -d)\}\Big)$$

- Such representations known for $\mathcal{U}_{\text{h-cube}}$, $\mathcal{U}_{\text{budget}}$, and other sets based on "absolutely symmetric convex functions" [Ben-Tal et al., 2009]

Piecewise Affine Rules

- One can also use piece-wise affine rules, e.g.,

$$x_t(d_{[t-1]}) := x_t^0 + X_t d_{[t-1]} + \sum_{k=1}^{t-1} \theta_{tk}^+ (d_k)^+ + \theta_{tk}^- (-d_k)^+$$

- These policies are actually affine in $\xi := [d, d^+, d^-]$
  - It would suffice to have a tractable representation for:

$$\text{ConvexHull}\Big(\{(d, d^+, d^-) \mid d \in \mathcal{U},\ d^+ = \max(0; d),\ d^- = \max(0; -d)\}\Big)$$

- Such representations known for $\mathcal{U}_{\text{h-cube}}$, $\mathcal{U}_{\text{budget}}$, and other sets based on "absolutely symmetric convex functions" [Ben-Tal et al., 2009]

- Large literature on other (nonlinear) rules:
  - *segregated rules* [Chen and Zhang, 2009, Goh and Sim, 2010, **?**]
  - *piecewise constant* [Bertsimas and Caramanis, 2010, Bertsimas et al., 2011b]
  - *kernelized* [Chatterjee et al., 2011, Skaf and Boyd, 2009]
  - *polynomial* [Ben-Tal et al., 2009, Bertsimas et al., 2011c]

Table of Contents

What Is Time Consistency?

Definition of time consistency:

Axiomatic property that requires a decision maker's stated preferences over future courses of action to remain consistent with the actual preferred actions when planned-for contingencies arise.

(Refer to Chapter 6 of (Shapiro et al., 2009) for background info.)

What Is Time Consistency?

Definition of time consistency:

Axiomatic property that requires a decision maker's stated preferences over future courses of action to remain consistent with the actual preferred actions when planned-for contingencies arise.

(Refer to Chapter 6 of (Shapiro et al., 2009) for background info.)

Intuitively,

- Relates to the way multi-stage decision models are employed to generate implemented decisions as the future unfolds

What Is Time Consistency?

Definition of time consistency:

Axiomatic property that requires a decision maker's stated preferences over future courses of action to remain consistent with the actual preferred actions when planned-for contingencies arise.
(Refer to Chapter 6 of (Shapiro et al., 2009) for background info.)

Intuitively,

- Relates to the way multi-stage decision models are employed to generate implemented decisions as the future unfolds
- Good news: All robust multi-stage decision models naturally express stage-wise preferences that are time consistent

What Is Time Consistency?

Definition of time consistency:

Axiomatic property that requires a decision maker's stated preferences over future courses of action to remain consistent with the actual preferred actions when planned-for contingencies arise.
(Refer to Chapter 6 of (Shapiro et al., 2009) for background info.)

Intuitively,

- Relates to the way multi-stage decision models are employed to generate implemented decisions as the future unfolds
- Good news: All robust multi-stage decision models naturally express stage-wise preferences that are time consistent
- Bad news: When decisions are implemented in a shrinking horizon fashion, the uncertainty set must be updated according to the "proper" update rule in order to give rise to this said consistency

Time Consistency in Robust “Coffee-vendor” Problem

Robust coffee-vendor problem (Part I):

A coffee stand is operated for one morning in the lobby of two hotels:

- hotel #1 during 7am-9am
- hotel #2 during 9am-11am
- possibility to replenish with fresh coffee at 9am

Time Consistency in Robust "Coffee-vendor" Problem

Robust coffee-vendor problem (Part I):

A coffee stand is operated for one morning in the lobby of two hotels:

- hotel #1 during 7am-9am
- hotel #2 during 9am-11am
- possibility to replenish with fresh coffee at 9am

Given that each hotel has two (hundred) guests, the salesman estimates the following demand:

- one (hundred) cups in hotel #1
- one (hundred) cups in hotel #2
- very unlikely that more than three (hundred) cups of coffee would be needed overall

Time Consistency in Robust "Coffee-vendor" Problem

Robust coffee-vendor problem (Part I):

A coffee stand is operated for one morning in the lobby of two hotels:

- hotel #1 during 7am-9am
- hotel #2 during 9am-11am
- possibility to replenish with fresh coffee at 9am

Given that each hotel has two (hundred) guests, the salesman estimates the following demand:

- one (hundred) cups in hotel #1
- one (hundred) cups in hotel #2
- very unlikely that more than three (hundred) cups of coffee would be needed overall

This situation seems to motivate the following uncertainty set :

$$\mathcal{U} := \{d \in [0, 2]^2 \mid d_1 + d_2 \leqslant 3\}$$

Time Consistency in Robust "Coffee-vendor" Problem

Robust coffee-vendor problem (Part II):

The coffee-vendor signed a contract with the two hotels to serve coffee to every thirsty guests. Here are some financial facts:

- contracts' revenue = 600\$ in total upfront
- production cost at 6am = 1\$ per cup
- production cost at 9am = 4\$ per cup
- salvage cost = 0\$ per cup
- contracted penalty for unserved customers = 10\$ per cup

Time Consistency in Robust "Coffee-vendor" Problem

Robust coffee-vendor problem (Part II):

The coffee-vendor signed a contract with the two hotels to serve coffee to every thirsty guests. Here are some financial facts:

- contracts' revenue = 600\$ in total upfront
- production cost at 6am = 1\$ per cup
- production cost at 9am = 4\$ per cup
- salvage cost = 0\$ per cup
- contracted penalty for unserved customers = 10\$ per cup

This situation seems to motivate the following deterministic model:

$$\underset{x_1 \geqslant 0, x_2 \geqslant 0}{\text{minimize}} \quad \underbrace{x_1 + 4x_2 + 10(d_1 + d_2 - x_1 - x_2)^+}_{\text{total cost (in hundreds of \$)}}$$

Robust Two-stage Coffee-vendor Problem

$$\begin{aligned} \underset{x_1, x_2(\cdot)}{\text{minimize}} \quad & \sup_{d \in \mathcal{U}} \; x_1 + 4x_2(d_1) + 10(d_1 + d_2 - x_1 - x_2(d_1))^+ \\ \text{s.t.} \quad & x_1 \geqslant 0, \;\; x_2(d_1) \geqslant 0 \,, \; \forall \, d \in \mathcal{U} \,, \end{aligned}$$

where $\mathcal{U} := \{d \in [0,2]^2 \,|\, d_1 + d_2 \leqslant 3\}$.

Conclusions that can be drawn based on optimal solution:

- Prepare 3 (hundred) cups at 6am
- Don't refresh the coffee at 9am under any circumstances
- The expected worst-case cost is 300\$ under any circumstances
- This gives an anticipated worst-case profit of $600\$ - 300\$ = 300\$$

Robust Two-stage Coffee-vendor Problem

$$\begin{array}{ll} \underset{x_1, x_2(\cdot)}{\text{minimize}} & \sup_{d \in \mathcal{U}} \; x_1 + 4x_2(d_1) + 10(d_1 + d_2 - x_1 - x_2(d_1))^+ \\ \text{s.t.} & x_1 \geqslant 0, \;\; x_2(d_1) \geqslant 0 \,, \; \forall\, d \in \mathcal{U} \,, \end{array}$$

where $\mathcal{U} := \{d \in [0, 2]^2 \,|\, d_1 + d_2 \leqslant 3\}$.

Conclusions that can be drawn based on optimal solution:

- Prepare 3 (hundred) cups at 6am
- Don't refresh the coffee at 9am under any circumstances
- The expected worst-case cost is 300$ under any circumstances
- This gives an anticipated worst-case profit of 600$ − 300$ = 300$

Is this truly the worst-case profit that will be achieved ?

Scenario #1: Time Consistent Second-stage Preferences

- At 9am, let $d_1 = 2$ be observed, the coffee-vendor changes hotel

Scenario #1: Time Consistent Second-stage Preferences

- At 9am, let $d_1 = 2$ be observed, the coffee-vendor changes hotel
- On the way, he updates the uncertainty set using the proper update rule

$$d_2 \in \{d_2 \in \mathbb{R} \,|\, (2, d_2) \in \mathcal{U}\} \; = \; \{d_2 \in [0,2] \,|\, 2 + d_2 \leqslant 3\} \; = \; [0,1] \, .$$

Scenario #1: Time Consistent Second-stage Preferences

- At 9am, let $d_1 = 2$ be observed, the coffee-vendor changes hotel
- On the way, he updates the uncertainty set using the proper update rule

$$d_2 \in \{d_2 \in \mathbb{R} \,|\, (2, d_2) \in \mathcal{U}\} = \{d_2 \in [0,2] \,|\, 2 + d_2 \leqslant 3\} = [0,1] \,.$$

- He solves the new robust optimization model

$$\begin{aligned} \underset{x_2}{\text{minimize}} \quad & \sup_{d_2 \in [0,1]} \overbrace{3}^{x_1 = 3} + 4x_2 + 10(\overbrace{2 - 3 + d_2 - x_2}^{d_2 - x_2 - 1 \leqslant 0 \,,\, \forall d_2 \in [0,1]})^+ \\ \text{s.t.} \quad & x_2 \geqslant 0 \,, \end{aligned}$$

Scenario #1: Time Consistent Second-stage Preferences

- At 9am, let $d_1 = 2$ be observed, the coffee-vendor changes hotel
- On the way, he updates the uncertainty set using the proper update rule

$$d_2 \in \{d_2 \in \mathbb{R} \,|\, (2, d_2) \in \mathcal{U}\} = \{d_2 \in [0,2] \,|\, 2 + d_2 \leqslant 3\} = [0,1]\,.$$

- He solves the new robust optimization model

$$\begin{aligned} \underset{x_2}{\text{minimize}} \quad & \sup_{d_2 \in [0,1]} \overbrace{3}^{x_1 = 3} + 4x_2 + 10\Big(\overbrace{2 - 3 + d_2 - x_2}^{d_2 - x_2 - 1 \leqslant 0\,,\ \forall\, d_2 \in [0,1]} \Big)^+ \\ \text{s.t.} \quad & x_2 \geqslant 0\,, \end{aligned}$$

- This allows him to confirm that he should not refresh the coffee
  - Worst-case cost stays 300$
  - Worst-case profit stays 600$ − 300$ = 300$

Scenario #1: Time Consistent Second-stage Preferences

- At 9am, let $d_1 = 2$ be observed, the coffee-vendor changes hotel
- On the way, he updates the uncertainty set using the proper update rule

$$d_2 \in \{d_2 \in \mathbb{R} \,|\, (2, d_2) \in \mathcal{U}\} = \{d_2 \in [0,2] \,|\, 2 + d_2 \leqslant 3\} = [0,1] \,.$$

- He solves the new robust optimization model

$$\begin{aligned} \underset{x_2}{\text{minimize}} \quad & \sup_{d_2 \in [0,1]} \overbrace{3}^{x_1 = 3} + 4x_2 + 10\Big(\overbrace{2 - 3 + d_2 - x_2}^{d_2 - x_2 - 1 \leqslant 0\,,\ \forall\, d_2 \in [0,1]} \Big)^+ \\ \text{s.t.} \quad & x_2 \geqslant 0 \,, \end{aligned}$$

- This allows him to confirm that he should not refresh the coffee
  - Worst-case cost stays 300\$
  - Worst-case profit stays 600\$ − 300\$ = 300\$

$\boxed{\text{No surprises.}}$

Scenario #2: Time Inconsistent Second-stage Preferences

- At 9am, let $d_1 = 2$ be observed, the coffee-vendor changes hotel

Scenario #2: Time Inconsistent Second-stage Preferences

- At 9am, let $d_1 = 2$ be observed, the coffee-vendor changes hotel
- Considering that the two hotels have an "independent" number of thirsty guests, he prefers to be safe by updating the uncertainty set to $d_2 \in [0, 2]$ instead of $d_2 \in [0, 1]$

Scenario #2: Time Inconsistent Second-stage Preferences

- At 9am, let $d_1 = 2$ be observed, the coffee-vendor changes hotel
- Considering that the two hotels have an "independent" number of thirsty guests, he prefers to be safe by updating the uncertainty set to $d_2 \in [0, 2]$ instead of $d_2 \in [0, 1]$
- He now solves the new robust optimization model

$$
\begin{array}{ll}
\underset{x_2}{\text{minimize}} & \sup_{d_2 \in [0,2]} \; 3 + 4x_2 + 10\Big(\overbrace{2 - 3 + d_2 - x_2}^{d_2 - x_2 - 1 \leqslant 1 - x_2 \,,\; \forall\, d_2 \in [0,2]} \Big)^+ \\
\text{s.t.} & x_2 \geqslant 0 \,,
\end{array}
$$

Scenario #2: Time Inconsistent Second-stage Preferences

- At 9am, let $d_1 = 2$ be observed, the coffee-vendor changes hotel
- Considering that the two hotels have an "independent" number of thirsty guests, he prefers to be safe by updating the uncertainty set to $d_2 \in [0, 2]$ instead of $d_2 \in [0, 1]$
- He now solves the new robust optimization model

$$\begin{aligned} \underset{x_2}{\text{minimize}} \quad & \sup_{d_2 \in [0,2]} \; 3 + 4x_2 + 10\Big(\overbrace{2 - 3 + d_2 - x_2}^{d_2 - x_2 - 1 \leqslant 1 - x_2 \,,\, \forall d_2 \in [0,2]} \Big)^+ \\ \text{s.t.} \quad & x_2 \geqslant 0 \,, \end{aligned}$$

- This warns him that he should prepare an additional 1 (hundred) cups
  - Worst-case cost is now 700\$ instead of 300\$
  - Worst-case profit is now 600\$ $-$ 700\$ $= -100$\$

Scenario #2: Time Inconsistent Second-stage Preferences

- At 9am, let $d_1 = 2$ be observed, the coffee-vendor changes hotel
- Considering that the two hotels have an "independent" number of thirsty guests, he prefers to be safe by updating the uncertainty set to $d_2 \in [0, 2]$ instead of $d_2 \in [0, 1]$
- He now solves the new robust optimization model

$$\begin{array}{lll} \underset{x_2}{\text{minimize}} & \sup_{d_2 \in [0,2]} & 3 + 4x_2 + 10(\overbrace{2 - 3 + d_2 - x_2}^{d_2 - x_2 - 1 \leqslant 1 - x_2\ ,\ \forall\, d_2 \in [0,2]})^+ \\ \text{s.t.} & x_2 \geqslant 0\ , & \end{array}$$

- This warns him that he should prepare an additional 1 (hundred) cups
  - Worst-case cost is now 700\$ instead of 300\$
  - Worst-case profit is now 600\$ − 700\$ = −100\$

> This issue can arise any time a new confidence region is used to update the uncertainty set dynamically.

One Possible Fix Using Bi-level Modeling

The following model accounts explicitly for time inconsistency

$$\begin{aligned} \underset{x_1, x_2(\cdot)}{\text{minimize}} \quad & \sup_{d \in \mathcal{U}} \; x_1 + 4x_2(d_1) + 10(d_1 + d_2 - x_1 - x_2(d_1))^+ \\ \text{s.t.} \quad & x_1 \geq 0 \\ & x_2(d_1) \in \underset{x_2' \geq 0}{\arg\min} \; \max_{d_2' \in \tilde{\mathcal{U}}_2(d_1)} \; c_2 x_2' + b(d_1 + d_2' - x_1 - x_2')^+ \,, \; \forall\, d_1 \in \mathcal{U}_1 \,, \end{aligned}$$

where

- $\mathcal{U}_1 := \{d_1 \in \mathbb{R} \,|\, \exists\, d_2, (d_1, d_2) \in \mathcal{U}\}$
- $\tilde{\mathcal{U}}_2(d_1)$ models exactly the type of uncertainty set that is used in stage $\#2$ when d_1 is observed

One Possible Fix Using Bi-level Modeling

The following model accounts explicitly for time inconsistency

$$\begin{aligned} \underset{x_1, x_2(\cdot)}{\text{minimize}} \quad & \sup_{d \in \mathcal{U}} \; x_1 + 4x_2(d_1) + 10(d_1 + d_2 - x_1 - x_2(d_1))^+ \\ \text{s.t.} \quad & x_1 \geq 0 \\ & x_2(d_1) \in \underset{x_2' \geq 0}{\arg\min} \; \max_{d_2' \in \tilde{\mathcal{U}}_2(d_1)} \; c_2 x_2' + b(d_1 + d_2' - x_1 - x_2')^+ \,, \; \forall\, d_1 \in \mathcal{U}_1 \,, \end{aligned}$$

where

- $\mathcal{U}_1 := \{d_1 \in \mathbb{R} \,|\, \exists\, d_2, (d_1, d_2) \in \mathcal{U}\}$
- $\tilde{\mathcal{U}}_2(d_1)$ models exactly the type of uncertainty set that is used in stage #2 when d_1 is observed

This model suggests preparing 4 (hundred) cups at 6am, and nothing later. Under scenario #2, the worst-case profit is 200$!!!

One Possible Fix Using Bi-level Modeling

The following model accounts explicitly for time inconsistency

$$\begin{aligned}
\underset{x_1, x_2(\cdot)}{\text{minimize}} \quad & \sup_{d \in \mathcal{U}} \; x_1 + 4x_2(d_1) + 10(d_1 + d_2 - x_1 - x_2(d_1))^+ \\
\text{s.t.} \quad & x_1 \geq 0 \\
& x_2(d_1) \in \underset{x_2' \geq 0}{\arg\min} \max_{d_2' \in \tilde{\mathcal{U}}_2(d_1)} \; c_2 x_2' + b(d_1 + d_2' - x_1 - x_2')^+ , \; \forall \, d_1 \in \mathcal{U}_1 \, ,
\end{aligned}$$

where

- $\mathcal{U}_1 := \{d_1 \in \mathbb{R} \,|\, \exists \, d_2, (d_1, d_2) \in \mathcal{U}\}$
- $\tilde{\mathcal{U}}_2(d_1)$ models exactly the type of uncertainty set that is used in stage #2 when d_1 is observed

This model suggests preparing 4 (hundred) cups at 6am, and nothing later. Under scenario #2, the worst-case profit is 200$!!!

The question remains of how to efficiently resolve these types of models...

Takeaway Message About Time Consistency

- Time consistency can easily become an issue when implementing decisions obtained from robust multi-stage models

Takeaway Message About Time Consistency

- Time consistency can easily become an issue when implementing decisions obtained from robust multi-stage models
- Quality of solution of robust multi-stage decision model is contingent on the assumption that the "proper" update rule is followed:

$$\mathcal{U}_{t:T}(\bar{d}_{[t-1]}) := \{(d_t, \ldots, d_T) \,|\, (\bar{d}_1, \ldots, \bar{d}_{t-1}, d_t, \ldots, d_T) \in \mathcal{U}\}\,.$$

Takeaway Message About Time Consistency

- Time consistency can easily become an issue when implementing decisions obtained from robust multi-stage models
- Quality of solution of robust multi-stage decision model is contingent on the assumption that the "proper" update rule is followed:

$$\mathcal{U}_{t:T}(\bar{d}_{[t-1]}) := \{(d_t, \ldots, d_T) \,|\, (\bar{d}_1, \ldots, \bar{d}_{t-1}, d_t, \ldots, d_T) \in \mathcal{U}\} \,.$$

- Unclear how much can be loss when the uncertainty set is updated differently

Takeaway Message About Time Consistency

- Time consistency can easily become an issue when implementing decisions obtained from robust multi-stage models
- Quality of solution of robust multi-stage decision model is contingent on the assumption that the "proper" update rule is followed:

$$\mathcal{U}_{t:T}(\bar{d}_{[t-1]}) := \{(d_t, \ldots, d_T) \,|\, (\bar{d}_1, \ldots, \bar{d}_{t-1}, d_t, \ldots, d_T) \in \mathcal{U}\}\,.$$

- Unclear how much can be loss when the uncertainty set is updated differently

Should one account for inconsistency directly in the decision model?

Table of Contents

Key Insights About Robust Multi-stage Decision Making

- Two common "pitfalls" might mislead the decision process
  1. treating adjustable decisions as static decisions
  2. updating the uncertainty set differently from the "proper" update rule

Key Insights About Robust Multi-stage Decision Making

- Two common "pitfalls" might mislead the decision process
  1. treating adjustable decisions as static decisions
  2. updating the uncertainty set differently from the "proper" update rule
- Correcting these issues requires skilful modeling and, in theory, significant computational resources

Key Insights About Robust Multi-stage Decision Making

- Two common "pitfalls" might mislead the decision process
  1. treating adjustable decisions as static decisions
  2. updating the uncertainty set differently from the "proper" update rule
- Correcting these issues requires skilful modeling and, in theory, significant computational resources
- From a practical point of view,

Key Insights About Robust Multi-stage Decision Making

- Two common "pitfalls" might mislead the decision process
  1. treating adjustable decisions as static decisions
  2. updating the uncertainty set differently from the "proper" update rule
- Correcting these issues requires skilful modeling and, in theory, significant computational resources
- From a practical point of view,
  1. there is hope that simple policies might often be sufficient because they are worst-case optimal

Key Insights About Robust Multi-stage Decision Making

- Two common "pitfalls" might mislead the decision process
  1. treating adjustable decisions as static decisions
  2. updating the uncertainty set differently from the "proper" update rule
- Correcting these issues requires skilful modeling and, in theory, significant computational resources
- From a practical point of view,
  1. there is hope that simple policies might often be sufficient because they are worst-case optimal
  2. it is unclear to us whether time inconsistent updating should be banished or accounted for in the robust multi-stage model?

Key Insights About Robust Multi-stage Decision Making

- Two common "pitfalls" might mislead the decision process
  1. treating adjustable decisions as static decisions
  2. updating the uncertainty set differently from the "proper" update rule
- Correcting these issues requires skilful modeling and, in theory, significant computational resources
- From a practical point of view,
  1. there is hope that simple policies might often be sufficient because they are worst-case optimal
  2. it is unclear to us whether time inconsistent updating should be banished or accounted for in the robust multi-stage model?
- In any case, it is important to acknowledge these issues in reporting decisions that are prescribed by these models

Future Directions

- Full theoretical understanding of quality of simple policies
- Development of decomposition schemes for accelerating resolution of robust multi-stage problems
- Identify tractable conservative approximation methods for problems with integer or random recourse variables
- Advanced methods that provide the decision maker with better control of temporal correlation, level of conservatism, and time consistency

Thank you for attending! Questions?...

References I

E. Adida and G. Perakis. A robust optimization approach to dynamic pricing and inventory control with no backorders. *Mathematical Programming*, 107:97–129, 2006.

F. Babonneau, J.-P. Vial, and R. Apparigliato. *Uncertainty and Environmental Decision Making*, volume 138 of *International Series in Operations Research & Management Science*, chapter Robust Optimization for Environmental and Energy Planning, pages 79–126. Springer US, 2010.

O. Baron, J. Milner, and H. Naseraldin. Facility location: A robust optimization approach. *Production and Operations Management*, 20(5):772–785, 2011.

A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.

A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25:1–13, 1999.

A. Ben-Tal and A. Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming, Series A*, 88(3):411–424, September 2000.

A. Ben-Tal and A. Nemirovski. Robust optimization - methodology and applications. *Mathematical Programming*, 92(3): 453–480, 2002.

A. Ben-Tal, A. Nemirovski, and C. Roos. Robust solutions of uncertain quadratic and conic-quadratic problems. *SIAM Journal on Optimization*, 13(2):535–560, 2002.

A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351–376, 2004.

A. Ben-Tal, B. Golany, A. Nemirovski, and J.-P. Vial. Retailer-supplier flexible commitments contracts: A robust optimization approach. *Manufacturing & Service Operations Management*, 7(3):248–271, 2005.

A. Ben-Tal, L. El-Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, 2009.

D. Bertsimas and C. Caramanis. Finite adaptability in multistage linear optimization. *Automatic Control, IEEE Transactions on*, 55(12):2751 –2766, dec. 2010. ISSN 0018-9286.

D. Bertsimas and V. Goyal. On the Power of Robust Solutions in Two-Stage Stochastic and Adaptive Optimization Problems. *Mathematics of Operations Research*, 35(2):284–305, 2010.

References II

D. Bertsimas and D. Pachamanova. Robust multiperiod portfolio management in the presence of transaction costs. *Computers and Operations Research*, 35(1):3–17, 2008.

D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming*, 98(1-3):49–71, 2003.

D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

D. Bertsimas and A. Thiele. A robust optimization approach to inventory theory. *Operations Research*, 54(1):150–168, 2006.

D. Bertsimas, D. Pachamanova, and M. Sim. Robust linear optimization under general norms. *Operations Research Letters*, 32 (6):510 – 516, 2004.

D. Bertsimas, D. Iancu, and P. Parrilo. Optimality of affine policies in multistage robust optimization. *Mathematics of Operations Research*, 35(2):363–394, 2010.

D. Bertsimas, C. Caramanis, and D. Brown. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011a.

D. Bertsimas, V. Goyal, and X. Sun. A geometric characterization of the power of finite adaptability in multistage stochastic and adaptive optimization. *Mathematics of Operations Research*, 36(1):24–54, 2011b.

D. Bertsimas, D. Iancu, and P. Parrilo. A hierarchy of near-optimal policies for multistage adaptive optimization. *IEEE Transactions on Automatic Control*, 56(12):2809 –2824, 2011c.

D. Bertsimas, V. Goyal, and B. Lu. A tight characterization of the performance of static solutions in two-stage adjustable robust linear optimization. *Mathematical Programming*, 150(2):281–319, 2015.

D. Bienstock and N. Özbay. Computing robust basestock levels. *Discrete Optimization*, 5(2):389 – 414, 2008.

S. Ceria and R. Stubbs. Incorporating estimation errors into portfolio selection: Robust portfolio construction. *Journal of Asset Management*, 7(2):109–127, July 2006.

D. Chatterjee, P. Hokayem, and J. Lygeros. Stochastic model predictive control with bounded control inputs: a vector space approach. *IEEE Transactions on Automatic Control*, 56(11):2704–2710, 2011.

X. Chen and Y. Zhang. Uncertain Linear Programs: Extended Affinely Adjustable Robust Counterparts. *Operations Research*, 57(6):1469–1482, November/December 2009.

References III

F. de Ruiter, A. Ben-Tal, R. Brekelmans, and D. den Hertog. Adjustable robust optimization with decision rules based on inexact revealed data. Center discussion paper series no. 2014-003, CentER, 2014.

E. Delage and D. Iancu. Robust multistage decision making. *Tutorials in Operations Research*, pages 20–46, 2015.

L. El-Ghaoui and H. Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM Journal on Matrix Analysis and Applications*, 18(4):1035–1064, October 1997.

L. El-Ghaoui, F. Oustry, and H. Lebret. Robust solutions to uncertain semidefinite programs. *SIAM Journal on Optimization*, 9 (1):33–52, 1998.

J. Falk. Exact solutions of inexact linear programs. *Operations Research*, 24(4):783–787, 1976.

V. Gabrel, C. Murat, and A. Thiele. Recent advances in robust optimization and robustness: An overview. Working paper, July 2012.

J. Goh and M. Sim. Distributionally robust optimization and its tractable approximations. *Operations Research*, 58(4-Part-1): 902–917, 2010.

D. Goldfarb and G. Iyengar. Robust Portfolio Selection Problems. *Mathematics of Operations Research*, 28(1):1–38, 2003.

D. Iancu, M. Sharma, and M. Sviridenko. Supermodularity and affine policies in dynamic robust optimization. *Operations Research*, 61(4):941–956, 2013.

X. Lin, S. Janak, and C. Floudas. A new robust optimization approach for scheduling under uncertainty: I. bounded uncertainty. *Computers and Chemical Engineering*, 28:1069–1085, 2004.

A. Lorca and A. Sun. Adaptive robust optimization with dynamic uncertainty sets for multi-period economic dispatch under significant wind. *IEEE Transactions on Power Systems*, 30(4):1702–1713, 2015.

M. Mani, A. Sing, and M. Orshansky. Joint Design-time and Post-silicon Minimization of Parametric Yield Loss Using Adjustable Robust Optimization. In *ICCAD '06: Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design*, pages 19–26. ACM, 2006.

S. Mittal, A. Schulz, and S. Stiller. Robust appointment scheduling. In *APPROX*, 2014.

G. Perakis and G. Roels. Robust controls for network revenue management. *Manufacturing & Service Operations Management*, 12(1):56–76, 2010.

References IV

M. Pinar and R. Tütüncü. Robust profit opportunities in risky financial portfolios. *Operations Research Letters*, 33(4):331 – 340, 2005.

A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming*. MPS / SIAM Series on Optimization. SIAM, 2009.

J. Skaf and S. Boyd. Nonlinear Q-Design for Convex Stochastic Control. *IEEE Transactions on Automatic Control*, 54(10): 2426–2430, October 2009.

A. L. Soyster. Technical Note–Convex Programming with Set-Inclusive Constraints and Applications to Inexact Linear Programming. *Operations Research*, 21(5):1154–1157, 1973.

R. Tütüncü and M. Koenig. Robust asset allocation. *Annals of Operations Research*, 132(1-4):157–187, November 2004.

W. Wiesemann, D. Kuhn, and B. Rustem. Robust resource allocations in temporal networks. *Mathematical Programming*, 135: 437–471, 2012. ISSN 0025-5610.

D. Yamashita, V. Armentano, and M. Laguna. Robust optimization models for project scheduling with resource availability cost. *Journal of Scheduling*, 10(1):67–76, 2007.

C. Zhao, J.Wang, J. Watson, and Y.Guan. Multi-stage robust unit commitment considering wind and demand response uncertainties. *IEEE Transactions on Power Systems*, 28(3):2708–2717, 2013.