Stochastic Optimization
IDA PhD course 2011ht

Stefanie Kosuch
PostDok at TCSLab
www.kosuch.eu/stefanie/

6. Lecture: Two- and Multi-Stage Problems
17. November 2011

Linköping University

1 Two-Stage Programming
- Deterministic reformulation in case of discrete distributions
- Main theoretical result

Outline

Linköping University

Why Two-Stage Programming?

Why Two-Stage Programming?

- Feasibility in all scenarios: Conservative solutions
- Chance-Constraints: Infeasibility of solution possible
- Simple Recourse: No actual "recourse" action

Why Two-Stage Programming?

- Feasibility in all scenarios: Conservative solutions
- Chance-Constraints: Infeasibility of solution possible
- Simple Recourse: No actual "recourse" action
- In the end: feasible solution $\rightarrow$ most realistic?

Basic structure of Two-Stage Programming Problem

Basic structure of Two-Stage Programming Problem

1. First stage: random parameters unknown
2. (Pre)-decision taken based on (statistical) information of uncertain parameters
3. Objective: Minimize expected total cost
4. Between first and second stage: **all** random parameters come to be known
5. Second stage: deterministic problem
6. Make corrective decision ("recourse"):
   - $\rightarrow$ Make solution feasible
   - $\rightarrow$ Decrease total cost

Linköping University

Linear Two-Stage Problem

Linear Two-Stage Problem

$$\min_{x \geq 0} \quad c^T x$$

s.t.

$x \in \mathbb{R}^{n_1}$: decision vector of 1$^{\text{st}}$ stage

Linear Two-Stage Problem

$$\begin{aligned} \min_{x \geq 0} \quad & c^T x \\ \text{s.t.} \quad & Ax \geq b, \end{aligned}$$

$x \in \mathbb{R}^{n_1}$: decision vector of 1^{st} stage

Linear Two-Stage Problem

$$\begin{aligned} \min_{x \geq 0} \quad & c^T x + \mathbb{E}[\mathcal{Q}(x,\chi)] \\ \text{s.t.} \quad & Ax \geq b, \\ & \mathcal{Q}(x,\chi) = \min_{y \geq 0} \quad d^T y \end{aligned}$$

$x \in \mathbb{R}^{n_1}$: decision vector of 1st stage
$y \in \mathbb{R}^{n_2}$: decision vectors of 2nd stage (recourse action)

Linear Two-Stage Problem

$$
\begin{aligned}
\min_{x\geq 0} \quad & c^T x + \mathbb{E}[\mathcal{Q}(x,\chi)] \\
\text{s.t.} \quad & Ax \geq b, \\
& \mathcal{Q}(x,\chi) = \min_{y\geq 0} \quad d^T y \\
& \qquad\qquad\qquad \text{s.t.} \quad T(\chi)x + W(\chi)y \geq h(\chi).
\end{aligned}
$$

$x \in \mathbb{R}^{n_1}$: decision vector of 1st stage
$y \in \mathbb{R}^{n_2}$: decision vectors of 2nd stage (recourse action)
$T(\chi)$: Technology matrix
$W(\chi)$: Recourse matrix

Linear Two-Stage Problem

$$
\begin{aligned}
\min_{x\geq 0} \quad & c^T x + \mathbb{E}[\mathcal{Q}(x,\chi)] \\
\text{s.t.} \quad & Ax \geq b, \\
& \mathcal{Q}(x,\chi) = \min_{y\geq 0} \quad d^T y \\
& \qquad\qquad\qquad \text{s.t.} \quad T(\chi)x + W(\chi)y \geq h(\chi).
\end{aligned}
$$

$x \in \mathbb{R}^{n_1}$: decision vector of 1$^{\text{st}}$ stage
$y \in \mathbb{R}^{n_2}$: decision vectors of 2$^{\text{nd}}$ stage (recourse action)
$T(\chi)$: Technology matrix
$W(\chi)$: Recourse matrix

Outline

1. Two-Stage Programming
   - Deterministic reformulation in case of discrete distributions
   - Main theoretical result

2. Multi-Stage Programming
   - Comparison with Online Optimization
   - Modeling Multi-Stage Stochastic Programming Problems
   - Deterministic Reformulation in case of discrete distributions (Main Idea)

Linear Two-Stage Problem

$$
\begin{aligned}
\min_{x \geq 0} \quad & c^T x + \mathbb{E}[\mathcal{Q}(x,\chi)] \\
\text{s.t.} \quad & Ax \geq b, \\
& \mathcal{Q}(x,\chi) = \min_{y \geq 0} \quad d^T y \\
& \qquad\qquad\qquad \text{s.t.} \quad T(\chi)x + W(\chi)y \geq h(\chi).
\end{aligned}
$$

$x \in \mathbb{R}^{n_1}$: decision vector of 1$^{\text{st}}$ stage
$y \in \mathbb{R}^{n_2}$: decision vectors of 2$^{\text{nd}}$ stage (recourse action)

Linear Two-Stage Problem

$$\begin{aligned} \min_{x \geq 0} \quad & c^T x + \mathbb{E}[\mathcal{Q}(x,\chi)] \\ \text{s.t.} \quad & Ax \geq b, \\ & \mathcal{Q}(x,\chi) = \min_{y \geq 0} \quad d^T y \\ & \qquad\qquad\qquad \text{s.t.} \quad T(\chi)x + W(\chi)y \geq h(\chi). \end{aligned}$$

$x \in \mathbb{R}^{n_1}$: decision vector of 1st stage
$y \in \mathbb{R}^{n_2}$: decision vectors of 2nd stage (recourse action)
$\chi^1, \ldots, \chi^K \in \mathbb{R}^s$: scenarios
$\mathbb{P}\{\chi = \chi^k\} := p^k$: probabilities

Linear Two-Stage Problem

$$
\begin{aligned}
\min_{x \geq 0} \quad & c^T x + \sum_{k=1}^{K} p^k \mathcal{Q}(x, \chi^k) \\
\text{s.t.} \quad & Ax \geq b, \\
& \mathcal{Q}(x, \chi) = \min_{y \geq 0} \quad d^T y \\
& \qquad\qquad\qquad \text{s.t.} \quad T(\chi)x + W(\chi)y \geq h(\chi).
\end{aligned}
$$

$x \in \mathbb{R}^{n_1}$: decision vector of 1st stage
$y \in \mathbb{R}^{n_2}$: decision vectors of 2nd stage (recourse action)
$\chi^1, \dots, \chi^K \in \mathbb{R}^s$: scenarios
$\mathbb{P}\{\chi = \chi^k\} := p^k$: probabilities

Linear Two-Stage Problem

$$
\begin{aligned}
\min_{x \geq 0} \quad & c^T x + \sum_{k=1}^{K} p^k \mathcal{Q}(x, \chi^k) \\
\text{s.t.} \quad & Ax \geq b, \\
& \mathcal{Q}(x, \chi^k) = \min_{y \geq 0} \quad d^T y \\
& \qquad\qquad\qquad \text{s.t.} \quad T(\chi^k)x + W(\chi^k)y \geq h(\chi^k).
\end{aligned}
$$

$x \in \mathbb{R}^{n_1}$: decision vector of 1st stage
$y \in \mathbb{R}^{n_2}$: decision vectors of 2nd stage (recourse action)
$\chi^1, \ldots, \chi^K \in \mathbb{R}^s$: scenarios
$\mathbb{P}\{\chi = \chi^k\} := p^k$: probabilities

Linköping University

Linear Two-Stage Problem

$$
\begin{aligned}
\min_{x\geq 0} \quad & c^T x + \sum_{k=1}^{K} p^k \mathcal{Q}(x,\chi^k) \\
\text{s.t.} \quad & Ax \geq b, \\
& \mathcal{Q}(x,\chi^k) = \min_{y\geq 0} \quad d^T y \\
& \qquad\qquad\qquad \text{s.t.} \quad T(\chi^k)x + W(\chi^k)y \geq h(\chi^k) \\
& (\forall\, k \in \{1,\ldots,K\})\,.
\end{aligned}
$$

$x \in \mathbb{R}^{n_1}$: decision vector of 1$^{\text{st}}$ stage
$y \in \mathbb{R}^{n_2}$: decision vectors of 2$^{\text{nd}}$ stage (recourse action)
$\chi^1,\ldots,\chi^K \in \mathbb{R}^s$: scenarios
$\mathbb{P}\{\chi = \chi^k\} := p^k$: probabilities

Linear Two-Stage Problem

$$
\begin{aligned}
\min_{x \geq 0} \quad & c^T x + \sum_{k=1}^{K} p^k \mathcal{Q}(x, \chi^k) \\
\text{s.t.} \quad & Ax \geq b, \\
& \mathcal{Q}(x, \chi^k) = \min_{y^k \geq 0} \quad d^T y^k \\
& \qquad\qquad\qquad \text{s.t.} \quad T(\chi^k)x + W(\chi^k)y^k \geq h(\chi^k) \\
& (\forall\, k \in \{1, \ldots, K\})\,.
\end{aligned}
$$

$x \in \mathbb{R}^{n_1}$: decision vector of 1^{st} stage
$y \in \mathbb{R}^{n_2}$: decision vectors of 2^{nd} stage (recourse action)
$\chi^1, \ldots, \chi^K \in \mathbb{R}^s$: scenarios
$\mathbb{P}\{\chi = \chi^k\} := p^k$: probabilities

Linköping University

Deterministically Reformulated Linear Two-Stage Problem

$$\min_{x \geq 0} \quad c^T x +$$

$$\text{s.t.} \quad Ax \geq b, \tag{3a}$$

Deterministically Reformulated Linear Two-Stage Problem

$$\begin{aligned} \min_{\substack{x \geq 0 \\ y^k \geq 0}} \quad & c^T x + \sum_{k=1}^{K} p^k \left(d^T y^k \right) \\ \text{s.t.} \quad & Ax \geq b, \end{aligned} \tag{3a}$$

Deterministically Reformulated Linear Two-Stage Problem

$$\min_{\substack{x\geq 0\\ y^k\geq 0}} \quad c^T x + \sum_{k=1}^{K} p^k \left(d^T y^k\right)$$

$$\text{s.t.} \quad Ax \geq b, \tag{3a}$$

Deterministically Reformulated Linear Two-Stage Problem

$$
\begin{aligned}
\min_{\substack{x \geq 0 \\ y^k \geq 0}} \quad & c^T x + \sum_{k=1}^{K} p^k \left(d^T y^k \right) \\
\text{s.t.} \quad & Ax \geq b, \qquad\qquad (3a) \\
& T(\chi^k)x + W(\chi^k)y^k \geq h(\chi^k) \quad (\forall\, k \in \{1, \ldots, K\}) \,.
\end{aligned}
$$

Deterministically Reformulated Linear Two-Stage Problem

$$\begin{aligned} \min_{\substack{x \geq 0 \\ y^k \geq 0}} \quad & c^T x + \sum_{k=1}^{K} p^k \left(d^T y^k \right) \\ \text{s.t.} \quad & Ax \geq b, \\ & T(\chi^k)x + W(\chi^k)y^k \geq h(\chi^k) \quad (\forall\, k \in \{1, \dots, K\}) \,. \end{aligned} \tag{3a}$$

Outline

Linköping University

Definition (Fixed Recourse)

Definition (Fixed Recourse)

A Linear Two Stage Programming Problem is said to have **fixed recourse** if the recourse matrix is deterministic, i.e. $W(\chi) = W$.

Definition (Fixed Recourse)

A Linear Two Stage Programming Problem is said to have **fixed recourse** if the recourse matrix is deterministic, i.e. $W(\chi) = W$.

Theorem

Given: linear two-stage problem

Theorem

Given: linear two-stage problem
Let:

Linköping University

Theorem

Given: linear two-stage problem
Let:

- *fixed recourse*

Theorem

Given: linear two-stage problem
Let:

- *fixed recourse*
- $\mathbb{E}[\chi] \in \mathbb{R}$
- $\{\lambda \in \mathbb{R}^{m_2} : d \geq \lambda W\} \neq \emptyset$

Linköping University

Linear Two-Stage Problem

$$\begin{aligned}
\min_{x\geq 0} \quad & c^T x + \mathbb{E}[\mathcal{Q}(x,\chi)] \\
\text{s.t.} \quad & Ax \geq b, \\
& \mathcal{Q}(x,\chi) = \min_{y\geq 0} \quad d^T y \\
& \qquad\qquad\qquad \text{s.t.} \quad T(\chi)x + W(\chi)y \geq h(\chi).
\end{aligned}$$

$x \in \mathbb{R}^{n_1}$: decision vector of 1st stage
$y \in \mathbb{R}^{n_2}$: decision vectors of 2nd stage (recourse action)
$T(\chi)$: Technology matrix
$W(\chi)$: Recourse matrix

Theorem

Given: linear two-stage problem
Let:

- *fixed recourse*
- $\mathbb{E}[\chi] \in \mathbb{R}$
- $\{\lambda \in \mathbb{R}^{m_2} : d \geq \lambda W\} \neq \emptyset$
  (≈ primal and dual feasible feasibility of second-stage problem)

Linköping University

Theorem

Given: linear two-stage problem
Let:

- *fixed recourse*
- $\mathbb{E}[\chi] \in \mathbb{R}$
- $\{\lambda \in \mathbb{R}^{m_2} : d \geq \lambda W\} \neq \emptyset$
  (≈ primal and dual feasible feasibility of second-stage problem)

Then $\mathbb{E}\left[\mathcal{Q}(x, \chi)\right]$ *is*

Linköping University

Theorem

Given: linear two-stage problem
Let:

- *fixed recourse*
- $\mathbb{E}[\chi] \in \mathbb{R}$
- $\{\lambda \in \mathbb{R}^{m_2} : d \geq \lambda W\} \neq \emptyset$
  (≈ primal and dual feasible feasibility of second-stage problem)

Then $\mathbb{E}\left[\mathcal{Q}(x, \chi)\right]$ *is*

- *real-valued,*

Linköping University

Theorem

Given: linear two-stage problem
Let:

- *fixed recourse*
- $\mathbb{E}[\chi] \in \mathbb{R}$
- $\{\lambda \in \mathbb{R}^{m_2} : d \geq \lambda W\} \neq \emptyset$
  (≈ primal and dual feasible feasibility of second-stage problem)

Then $\mathbb{E}\left[\mathcal{Q}(x,\chi)\right]$ *is*

- *real-valued,*
- *piecewise linear and convex in* x*,*

Theorem

Given: linear two-stage problem
Let:

- *fixed recourse*
- $\mathbb{E}[\chi] \in \mathbb{R}$
- $\{\lambda \in \mathbb{R}^{m_2} : d \geq \lambda W\} \neq \emptyset$
  (≈ primal and dual feasible feasibility of second-stage problem)

Then $\mathbb{E}\left[\mathcal{Q}(x, \chi)\right]$ *is*

- *real-valued,*
- *piecewise linear and convex in* x*,*
- *Lipschitz continuous in* x*,*

Linköping University

Theorem

Given: linear two-stage problem
Let:

- *fixed recourse*
- $\mathbb{E}[\chi] \in \mathbb{R}$
- $\{\lambda \in \mathbb{R}^{m_2} : d \geq \lambda W\} \neq \emptyset$
  (≈ primal and dual feasible feasibility of second-stage problem)

Then $\mathbb{E}\left[\mathcal{Q}(x, \chi)\right]$ *is*

- *real-valued,*
- *piecewise linear and convex in* x*,*
- *Lipschitz continuous in* x*,*
- *sub-differentiable in* x*.*

Linköping University

Theorem

Given: linear two-stage problem
Let:

- *fixed recourse*
- $\mathbb{E}[\chi] \in \mathbb{R}$
- $\{\lambda \in \mathbb{R}^{m_2} : d \geq \lambda W\} \neq \emptyset$
  (≈ primal and dual feasible feasibility of second-stage problem)

Then $\mathbb{E}\left[\mathcal{Q}(x,\chi)\right]$ *is*

- *real-valued,*
- *piecewise linear and convex in* x*,*
- *Lipschitz continuous in* x*,*
- *sub-differentiable in* x*.*

Theorem

Given: linear two-stage problem
Let:

- *fixed recourse*
- $\mathbb{E}[\chi] \in \mathbb{R}$
- $\{\lambda \in \mathbb{R}^{m_2} : d \geq \lambda W\} \neq \emptyset$
  (≈ primal and dual feasible feasibility of second-stage problem)
- χ *continuously distributed*

Then $\mathbb{E}\left[\mathcal{Q}(x,\chi)\right]$ *is*

- *real-valued,*
- *piecewise linear and convex in* x*,*
- *Lipschitz continuous in* x*,*
- *sub-differentiable in* x*.*

Theorem

Given: linear two-stage problem
Let:

- *fixed recourse*
- $\mathbb{E}[\chi] \in \mathbb{R}$
- $\{\lambda \in \mathbb{R}^{m_2} : d \geq \lambda W\} \neq \emptyset$
  (≈ primal and dual feasible feasibility of second-stage problem)
- χ *continuously distributed*

Then $\mathbb{E}\left[\mathcal{Q}(x, \chi)\right]$ *is*

- *real-valued,*
- *piecewise linear and convex in* x*,*
- *Lipschitz continuous in* x*,*
- *continuously differentiable.*

Theorem

Given: linear two-stage problem
Let:

- *fixed recourse*
- $\mathbb{E}[\chi] \in \mathbb{R}$
- $\{\lambda \in \mathbb{R}^{m_2} : d \geq \lambda W\} \neq \emptyset$
  (≈ primal and dual feasible feasibility of second-stage problem)
- χ *continuously distributed*

Then $\mathbb{E}\left[\mathcal{Q}(x, \chi)\right]$ *is*

- *real-valued,*
- *piecewise linear and convex in* x*,*
- *Lipschitz continuous in* x*,*
- *continuously differentiable.*

Theorem

Given: linear two-stage problem
Let:

- *fixed recourse*
- $\mathbb{E}[\chi] \in \mathbb{R}$
- $\{\lambda \in \mathbb{R}^{m_2} : d \geq \lambda W\} \neq \emptyset$
  (≈ primal and dual feasible feasibility of second-stage problem)
- χ *continuously distributed*

Then $\mathbb{E}\left[\mathcal{Q}(x, \chi)\right]$ *is*

- *real-valued,*
- *piecewise linear and convex in x,*
- *Lipschitz continuous in x,*
- *continuously differentiable.*

Theorem

Given: linear two-stage problem
Let:

- *fixed recourse*
- $\mathbb{E}[\chi] \in \mathbb{R}$
- $\{\lambda \in \mathbb{R}^{m_2} : d \geq \lambda W\} \neq \emptyset$
  (≈ primal and dual feasible feasibility of second-stage problem)

Then $\mathbb{E}\left[\mathcal{Q}(x, \chi)\right]$ *is*

- *real-valued,*
- *piecewise linear and convex in* x*,*
- *Lipschitz continuous in* x*,*
- *subdifferentiable in* x*,*

Theorem

Given: linear two-stage problem
Let:

- *fixed recourse*
- $\mathbb{E}[\chi] \in \mathbb{R}$
- $\{\lambda \in \mathbb{R}^{m_2} : d \geq \lambda W\} \neq \emptyset$
  (≈ primal and dual feasible feasibility of second-stage problem)
- χ *finitely discretely distributed*

Then $\mathbb{E}\left[\mathcal{Q}(x,\chi)\right]$ *is*

- *real-valued,*
- *piecewise linear and convex in* x*,*
- *Lipschitz continuous in* x*,*
- *subdifferentiable in* x*,*

Theorem

Given: linear two-stage problem
Let:

- *fixed recourse*
- $\mathbb{E}[\chi] \in \mathbb{R}$
- $\{\lambda \in \mathbb{R}^{m_2} : d \geq \lambda W\} \neq \emptyset$
  (≈ primal and dual feasible feasibility of second-stage problem)
- χ *finitely discretely distributed*

Then $\mathbb{E}\left[\mathcal{Q}(x,\chi)\right]$ *is*

- *real-valued,*
- *piecewise linear and convex in* x*,*
- *Lipschitz continuous in* x*,*
- *subdifferentiable in* x*,*
- *a polyhedral convex function.*

Theorem

Given: linear two-stage problem
Let:

- *fixed recourse*
- $\mathbb{E}[\chi] \in \mathbb{R}$
- $\{\lambda \in \mathbb{R}^{m_2} : d \geq \lambda W\} \neq \emptyset$
  (≈ primal and dual feasible feasibility of second-stage problem)
- χ *finitely discretely distributed*

Then $\mathbb{E}\left[\mathcal{Q}(x,\chi)\right]$ *is*

- *real-valued,*
- *piecewise linear and convex in* x*,*
- *Lipschitz continuous in* x*,*
- *subdifferentiable in* x*,*
- *a polyhedral convex function.*

Theorem

Given: linear two-stage problem
Let:

- *fixed recourse*
- $\mathbb{E}[\chi] \in \mathbb{R}$
- $\{\lambda \in \mathbb{R}^{m_2} : d \geq \lambda W\} \neq \emptyset$
  (≈ primal and dual feasible feasibility of second-stage problem)
- χ *finitely discretely distributed*

Then $\mathbb{E}\left[\mathcal{Q}(x, \chi)\right]$ *is*

- *real-valued,*
- *piecewise linear and convex in* x*,*
- *Lipschitz continuous in* x*,*
- *subdifferentiable in* x*,*
- *a polyhedral convex function.*

Outline

1 Two-Stage Programming
- Deterministic reformulation in case of discrete distributions
- Main theoretical result

2 Multi-Stage Programming
- Comparison with Online Optimization
- Modeling Multi-Stage Stochastic Programming Problems
- Deterministic Reformulation in case of discrete distributions (Main Idea)

Linköping University

Basics of Multi-Stage Programming

Basics of Multi-Stage Programming

- Information obtained in several "stages"

Basics of Multi-Stage Programming

- Information obtained in several "stages"
- Recourse action can be taken in each stage

Basics of Multi-Stage Programming

- Information obtained in several "stages"
- Recourse action can be taken in each stage
- Recourse action depends on:

Linköping University

Basics of Multi-Stage Programming

- Information obtained in several "stages"
- Recourse action can be taken in each stage
- Recourse action depends on:
  - $\rightarrow$ Information available

Basics of Multi-Stage Programming

- Information obtained in several "stages"
- Recourse action can be taken in each stage
- Recourse action depends on:
  - $\rightarrow$ Information available
  - $\rightarrow$ Previous decisions

Basics of Multi-Stage Programming

- Information obtained in several "stages"
- Recourse action can be taken in each stage
- Recourse action depends on:
  - $\rightarrow$ Information available
  - $\rightarrow$ Previous decisions
- Recourse action does not depend on:

Basics of Multi-Stage Programming

- Information obtained in several "stages"
- Recourse action can be taken in each stage
- Recourse action depends on:
  - $\rightarrow$ Information available
  - $\rightarrow$ Previous decisions
- Recourse action does not depend on:
  - $\rightarrow$ Information to be released later ($\neq$ det. multi-period optimization)

Linköping University

Basics of Multi-Stage Programming

- Information obtained in several "stages"
- Recourse action can be taken in each stage
- Recourse action depends on:
  - → Information available
  - → Previous decisions
- Recourse action does not depend on:
  - → Information to be released later ($\neq$ det. multi-period optimization)
- Objective: Minimize **total expected cost**

Basics of Multi-Stage Programming

- Information obtained in several "stages"
- Recourse action can be taken in each stage
- Recourse action depends on:
  - $\rightarrow$ Information available
  - $\rightarrow$ Previous decisions
- Recourse action does not depend on:
  - $\rightarrow$ Information to be released later ($\neq$ det. multi-period optimization)
- Objective: Minimize **total expected cost**
- Crucial: Discretization of time and random distribution(s)

Linköping University

Outline

1. Two-Stage Programming
   - Deterministic reformulation in case of discrete distributions
   - Main theoretical result

2. Multi-Stage Programming
   - Comparison with Online Optimization
   - Modeling Multi-Stage Stochastic Programming Problems
   - Deterministic Reformulation in case of discrete distributions (Main Idea)

Basics of Online Optimization

Basics of Online Optimization

- Information **changes** over time

Basics of Online Optimization

- Information **changes** over time
- Time discretized (**periods**)

Basics of Online Optimization

- Information **changes** over time
- Time discretized (**periods**)
- (New) Decision can be taken in each time period

Basics of Online Optimization

- Information **changes** over time
- Time discretized (**periods**)
- (New) Decision can be taken in each time period
- Decision depends on:

Basics of Online Optimization

- Information **changes** over time
- Time discretized (**periods**)
- (New) Decision can be taken in each time period
- Decision depends on:

  $\rightarrow$ Information available

Linköping University

Basics of Online Optimization

- Information **changes** over time
- Time discretized (**periods**)
- (New) Decision can be taken in each time period
- Decision depends on:
  - $\rightarrow$ Information available
  - $\rightarrow$ Previous decisions

Linköping University

Basics of Online Optimization

- Information **changes** over time
- Time discretized (**periods**)
- (New) Decision can be taken in each time period
- Decision depends on:
  - → Information available
  - → Previous decisions
  - → **"Learning process"**

Basics of Online Optimization

- Information **changes** over time
- Time discretized (**periods**)
- (New) Decision can be taken in each time period
- Decision depends on:
  - → Information available
  - → Previous decisions
  - → **"Learning process"**
- Decision does not depend on:

Linköping University

Basics of Online Optimization

- Information **changes** over time
- Time discretized (**periods**)
- (New) Decision can be taken in each time period
- Decision depends on:
  - $\rightarrow$ Information available
  - $\rightarrow$ Previous decisions
  - $\rightarrow$ **"Learning process"**
- Decision does not depend on:
  - $\rightarrow$ Information to be released later ($\neq$ det. multi-period optimization)

Basics of Online Optimization

- Information **changes** over time
- Time discretized (**periods**)
- (New) Decision can be taken in each time period
- Decision depends on:
  - $\rightarrow$ Information available
  - $\rightarrow$ Previous decisions
  - $\rightarrow$ **"Learning process"**
- Decision does not depend on:
  - $\rightarrow$ Information to be released later ($\neq$ det. multi-period optimization)
- Possible objectives:

Basics of Online Optimization

- Information **changes** over time
- Time discretized (**periods**)
- (New) Decision can be taken in each time period
- Decision depends on:
  - → Information available
  - → Previous decisions
  - → **"Learning process"**
- Decision does not depend on:
  - → Information to be released later ($\neq$ det. multi-period optimization)
- Possible objectives:
  a) Minimize **total cost**

Linköping University

Basics of Online Optimization

- Information **changes** over time
- Time discretized (**periods**)
- (New) Decision can be taken in each time period
- Decision depends on:
  - → Information available
  - → Previous decisions
  - → **"Learning process"**
- Decision does not depend on:
  - → Information to be released later ($\neq$ det. multi-period optimization)
- Possible objectives:
  - a) Minimize **total cost**
  - b) Average cost not higher than r times "optimal" cost

Linköping University

Similarities

Similarities

- Decision taken over time (stages $\leftrightarrow$ periods)

Similarities

- Decision taken over time (stages $\leftrightarrow$ periods)
- Decision does not depend on information available in future

Linköping University

Differences

Solution

Linköping University

Differences

Solution

Multi-Stage Programming:

Differences

Solution

Multi-Stage Programming:

- Decision vector

Differences

Solution

Multi-Stage Programming:

- Decision vector
- Decision for each stage and scenario

Differences

Solution

Multi-Stage Programming:

- Decision vector
- Decision for each stage and scenario

Differences

Solution

Multi-Stage Programming:

- Decision vector
- Decision for each stage and scenario

Online Optimization:

Differences

Solution

Multi-Stage Programming:

- Decision vector
- Decision for each stage and scenario

Online Optimization:

- Policy

Differences

Solution

Multi-Stage Programming:

- Decision vector
- Decision for each stage and scenario

Online Optimization:

- Policy
- Online Algorithm

Linköping University

Differences

Objective

Differences

Objective

Multi-Stage Programming:

Differences

Objective

Multi-Stage Programming:

- Minimize expected total cost

Differences

Objective

Multi-Stage Programming:

- Minimize expected total cost

Differences

Objective

Multi-Stage Programming:

- Minimize expected total cost

Online Optimization:

Differences

Objective

Multi-Stage Programming:

- Minimize expected total cost

Online Optimization:

- Minimize total cost

Differences

Objective

Multi-Stage Programming:

- Minimize expected total cost

Online Optimization:

- Minimize total cost
- Propose "competitive" online algorithm

Differences

Modeling

Differences

Modeling

Multi-Stage Programming:

Differences

Modeling

Multi-Stage Programming:

- Random variables

Differences

Modeling

Multi-Stage Programming:

- Random variables
- Knowledge about distribution

Linköping University

Differences

Modeling

Multi-Stage Programming:

- Random variables
- Knowledge about distribution

Linköping University

Differences

Modeling

Multi-Stage Programming:

- Random variables
- Knowledge about distribution

Online Optimization:

Differences

Modeling

Multi-Stage Programming:

- Random variables
- Knowledge about distribution

Online Optimization:

- Generally no knowledge of structure of input

Linköping University

Differences

Modeling

Multi-Stage Programming:

- Random variables
- Knowledge about distribution

Online Optimization:

- Generally no knowledge of structure of input
- State sequence

Linköping University

Differences

Modeling

Multi-Stage Programming:

- Random variables
- Knowledge about distribution

Online Optimization:

- Generally no knowledge of structure of input
- State sequence
- Deterministic formulation

Two different views

Two different views

Multi-Stage programming is...

Two different views

Multi-Stage programming is...

- ... a different approach to optimization with information over time than...

...online optimization.

Two different views

Multi-Stage programming is...

- ... a different approach to optimization with information over time than...
- ... one way to attack problems in the domain of...

...online optimization.

Which approach to choose?

Questions to be answered:

- Do I have a finite, infinite or undefined number of periods/stages?

Which approach to choose?

Questions to be answered:

- Do I have a finite, infinite or undefined number of periods/stages?
- Is information changing or extended over time?

Which approach to choose?

Questions to be answered:

- Do I have a finite, infinite or undefined number of periods/stages?
- Is information changing or extended over time?
- Do I have statistical information about the uncertain parameters?
  OR: Do I need/want to learn about uncertainties "on the fly"?

Which approach to choose?

Questions to be answered:

- Do I have a finite, infinite or undefined number of periods/stages?
- Is information changing or extended over time?
- Do I have statistical information about the uncertain parameters?
  OR: Do I need/want to learn about uncertainties "on the fly"?
- Is discretization of the random distribution an option?

Which approach to choose?

Questions to be answered:

- Do I have a finite, infinite or undefined number of periods/stages?
- Is information changing or extended over time?
- Do I have statistical information about the uncertain parameters?
  OR: Do I need/want to learn about uncertainties "on the fly"?
- Is discretization of the random distribution an option?
- Do I have the infrastructure and/or time to make decisions online?

Which approach to choose?

Questions to be answered:

- Do I have a finite, infinite or undefined number of periods/stages?
- Is information changing or extended over time?
- Do I have statistical information about the uncertain parameters?
  OR: Do I need/want to learn about uncertainties "on the fly"?
- Is discretization of the random distribution an option?
- Do I have the infrastructure and/or time to make decisions online?
- Will I use my solution once or several times?

Outline

General Linear Multi-Stage Model

Linköping University

General Linear Multi-Stage Model

$$\min_{x_0 \geq 0} \quad c_0^T x_0$$

s.t.

General Linear Multi-Stage Model

$$\begin{aligned} \min_{x_0 \geq 0} \quad & c_0^\top x_0 \\ \text{s.t.} \quad & A_{00} x_0 \geq b_0, \end{aligned}$$

Linköping University

General Linear Multi-Stage Model

$$\min_{x_0 \geq 0} \quad c_0^\intercal x_0 + \mathbb{E}_{\zeta_1}\left[\mathcal{Q}^1(x_0, \zeta_1)\right]$$

$$\text{s.t.} \quad A_{00} x_0 \geq b_0,$$

Linköping University

General Linear Multi-Stage Model

$$\min_{x_0 \geq 0} \quad c_0^\intercal x_0 + \mathbb{E}_{\zeta_1}\left[\mathcal{Q}^1(x_0,\zeta_1)\right]$$
$$\text{s.t.} \quad A_{00}x_0 \geq b_0,$$

$$\mathcal{Q}^1(x_0,\zeta_1) = \min_{x_1 \geq 0} \quad c_1^\intercal x_1$$

$x_1 = x_1(\zeta_1)\ \forall t \geq 1$

Linköping University

General Linear Multi-Stage Model

$$\min_{x_0 \geq 0} \quad c_0^\intercal x_0 + \mathbb{E}_{\zeta_1}\left[\mathcal{Q}^1(x_0, \zeta_1)\right]$$

$$\text{s.t.} \quad A_{00} x_0 \geq b_0,$$

$$\mathcal{Q}^1(x_0, \zeta_1) = \min_{x_1 \geq 0} \quad c_1^\intercal x_1$$

$$\text{s.t.} \quad \sum_{s=0}^{1} A_{1s}(\zeta_1) x_s \geq b_1(\zeta_1).$$

$x_1 = x_1(\zeta_1) \; \forall t \geq 1$

Linköping University

General Linear Multi-Stage Model

$$\min_{x_0 \geq 0} \quad c_0^\intercal x_0 + \mathbb{E}_{\zeta_1}\left[\mathcal{Q}^1(x_{[0]}, \zeta_{[1]})\right]$$

$$\text{s.t.} \quad A_{00}x_0 \geq b_0,$$

$$\mathcal{Q}^T(x_{[T-1]}, \zeta_{[T]}) = \min_{x_T \geq 0} \quad c_T^\intercal x_T$$

$$\text{s.t.} \quad \sum_{s=0}^{T} A_{Ts}(\zeta_{[T]})x_s \geq b_T(\zeta_{[T]}).$$

$\zeta_{[t]} = (\zeta_1, \ldots, \zeta_t)$
$x_{[t-1]} = (x_0, x_1, \ldots, x_{t-1})$
$x_t = x_t(\zeta_{[t]}) \; \forall t \geq 1$

Linköping University

General Linear Multi-Stage Model

$$
\begin{aligned}
\min_{x_0 \geq 0} \quad & c_0^\intercal x_0 + \mathbb{E}_{\zeta_1}\left[\mathcal{Q}^1(x_{[0]}, \zeta_{[1]})\right] \\
\text{s.t.} \quad & A_{00} x_0 \geq b_0, \\
& \left.\begin{aligned}
\mathcal{Q}^t(x_{[t-1]}, \zeta_{[t]}) = \min_{x_t \geq 0} \quad & c_t^\intercal x_t + \mathbb{E}_{\zeta_{t+1}|\zeta_{[t]}}\left[\mathcal{Q}^{t+1}(x_{[t]}, \zeta_{[t+1]})\right] \\
\text{s.t.} \quad & \sum_{s=0}^{t} A_{ts}(\zeta_{[t]}) x_s \geq b_t(\zeta_{[t]}).
\end{aligned}\right\} \quad \begin{matrix}\forall t \in \\ \{1, \ldots, t-1\}\end{matrix} \\
& \begin{aligned}
\mathcal{Q}^T(x_{[T-1]}, \zeta_{[T]}) = \min_{x_T \geq 0} \quad & c_T^\intercal x_T \\
\text{s.t.} \quad & \sum_{s=0}^{T} A_{Ts}(\zeta_{[T]}) x_s \geq b_T(\zeta_{[T]}).
\end{aligned}
\end{aligned}
$$

$\zeta_{[t]} = (\zeta_1, \ldots, \zeta_t)$
$x_{[t-1]} = (x_0, x_1, \ldots, x_{t-1})$
$x_t = x_t(\zeta_{[t]})\ \forall t \geq 1$

Outline

Definition

Scenario

Definition

Scenario

- Sequence of outcomes $\zeta_1^k, \ldots, \zeta_T^k$ of $\zeta_1, \ldots, \zeta_T$

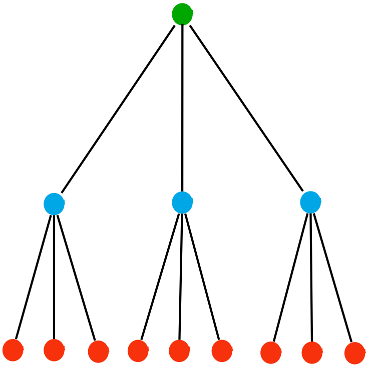
Definition

Scenario

- Sequence of outcomes $\zeta_1^k, \ldots, \zeta_T^k$ of $\zeta_1, \ldots, \zeta_T$
- Outcome $\zeta_{[T]}^k$ of $\zeta_{[T]} = (\zeta_1, \ldots, \zeta_T)$

Linköping University

Scenario Tree



Definition

Scenario

- Sequence of outcomes $\zeta_1^k, \dots, \zeta_T^k$ of $\zeta_1, \dots, \zeta_T$
- Outcome $\zeta_{[T]}^k$ of $\zeta_{[T]} = (\zeta_1, \dots, \zeta_T)$

Definition

Scenario

- Sequence of outcomes $\zeta_1^k, \ldots, \zeta_T^k$ of $\zeta_1, \ldots, \zeta_T$
- Outcome $\zeta_{[T]}^k$ of $\zeta_{[T]} = (\zeta_1, \ldots, \zeta_T)$

Definition

Scenario

- Sequence of outcomes $\zeta_1^k, \ldots, \zeta_T^k$ of $\zeta_1, \ldots, \zeta_T$
- Outcome $\zeta_{[T]}^k$ of $\zeta_{[T]} = (\zeta_1, \ldots, \zeta_T)$

Idea

Definition

Scenario

- Sequence of outcomes $\zeta_1^k, \ldots, \zeta_T^k$ of $\zeta_1, \ldots, \zeta_T$
- Outcome $\zeta_{[T]}^k$ of $\zeta_{[T]} = (\zeta_1, \ldots, \zeta_T)$

Idea

Introduce sequence of decision vectors $(x_0^k, x_1^k, \ldots, x_T^k)$ ($\forall$ scenarios $\zeta_{[T]}^k$)

Definition

Scenario

- Sequence of outcomes $\zeta_1^k, \ldots, \zeta_T^k$ of $\zeta_1, \ldots, \zeta_T$
- Outcome $\zeta_{[T]}^k$ of $\zeta_{[T]} = (\zeta_1, \ldots, \zeta_T)$

Idea

Introduce sequence of decision vectors $(x_0^k, x_1^k, \ldots, x_T^k)$ ($\forall$ scenarios $\zeta_{[T]}^k$)

Problem

Definition

Scenario

- Sequence of outcomes $\zeta_1^k, \ldots, \zeta_T^k$ of $\zeta_1, \ldots, \zeta_T$
- Outcome $\zeta_{[T]}^k$ of $\zeta_{[T]} = (\zeta_1, \ldots, \zeta_T)$

Idea

Introduce sequence of decision vectors $(x_0^k, x_1^k, \ldots, x_T^k)$ ($\forall$ scenarios $\zeta_{[T]}^k$)

Problem

Some decision vectors should be equal (e.g. $x_0^k = x_0^{k'}$ $\forall k, k'$)

Definition

Scenario

- Sequence of outcomes $\zeta_1^k, \ldots, \zeta_T^k$ of $\zeta_1, \ldots, \zeta_T$
- Outcome $\zeta_{[T]}^k$ of $\zeta_{[T]} = (\zeta_1, \ldots, \zeta_T)$

Idea

Introduce sequence of decision vectors $(x_0^k, x_1^k, \ldots, x_T^k)$ ($\forall$ scenarios $\zeta_{[T]}^k$)

Problem

Some decision vectors should be equal (e.g. $x_0^k = x_0^{k'}\ \forall k, k'$)

Solution

Linköping University

Definition

Scenario

- Sequence of outcomes $\zeta_1^k, \ldots, \zeta_T^k$ of $\zeta_1, \ldots, \zeta_T$
- Outcome $\zeta_{[T]}^k$ of $\zeta_{[T]} = (\zeta_1, \ldots, \zeta_T)$

Idea

Introduce sequence of decision vectors $(x_0^k, x_1^k, \ldots, x_T^k)$ ($\forall$ scenarios $\zeta_{[T]}^k$)

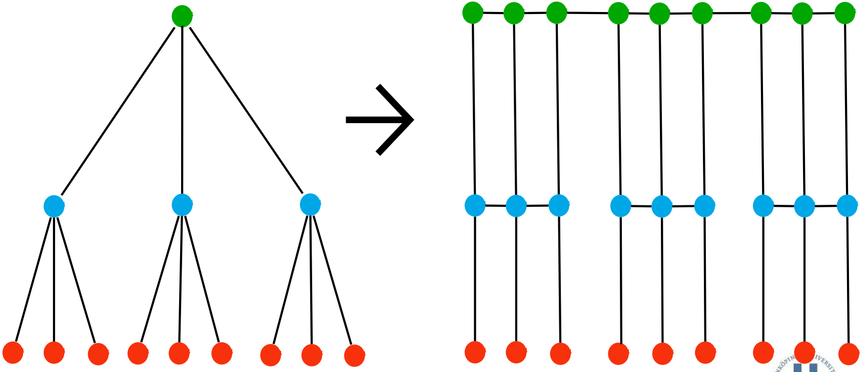
Problem

Some decision vectors should be equal (e.g. $x_0^k = x_0^{k'}\ \forall k, k'$)

Solution

Non-anticipativity constraints: $x_t^k = x_t^{k'}$ if $\zeta_{[t]}^k = \zeta_{[t]}^{k'}$

Scenario Tree



Definition

Scenario

- Sequence of outcomes $\zeta_1^k, \ldots, \zeta_T^k$ of $\zeta_1, \ldots, \zeta_T$
- Outcome $\zeta_{[T]}^k$ of $\zeta_{[T]} = (\zeta_1, \ldots, \zeta_T)$

Idea

Introduce sequence of decision vectors $(x_0^k, x_1^k, \ldots, x_T^k)$ ($\forall$ scenarios $\zeta_{[T]}^k$)

Problem

Some decision vectors should be equal (e.g. $x_0^k = x_0^{k'}$ $\forall k, k'$)

Solution

Non-anticipativity constraints: $x_t^k = x_t^{k'}$ if $\zeta_{[t]}^k = \zeta_{[t]}^{k'}$

QUESTIONS?

What about next week?