Stochastic Optimization
IDA PhD course 2011ht

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

9. Lecture: Stochastic Decomposition
08. December 2011

1. Decomposition Methods
   - L-shaped method (Benders' decomposition)

1 Decomposition Methods
- L-shaped method (Benders' decomposition)

2 Inner Approximation Approaches
- Stochastic Decomposition

Outline

1 Decomposition Methods
- L-shaped method (Benders' decomposition)

2 Inner Approximation Approaches
- Stochastic Decomposition

3 Complexity of Two-Stage Optimization problems

Outline

1 Decomposition Methods
- L-shaped method (Benders' decomposition)

2 Inner Approximation Approaches
- Stochastic Decomposition

3 Complexity of Two-Stage Optimization problems

Linear Two-Stage Problem with fixed recourse

$$\min_{x \geq 0} \quad c^T x + \mathbb{E}[\mathcal{Q}(x, \chi)]$$

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

$$\mathcal{Q}(x, \chi) = \min_{y \geq 0} \quad d^T y$$

$$\text{s.t.} \quad Wy \geq h(\chi) - T(\chi)x.$$

$x \in \mathbb{R}^{n_1}$: 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

Linköping University

Linear Two-Stage Problem with fixed recourse

$$\min_{\substack{x \geq 0 \\ \theta \geq 0}} \quad c^T x + \mathbb{E}[\mathcal{Q}(x, \chi)]$$

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

$$\theta \geq \mathbb{E}[\mathcal{Q}(x, \chi)]$$

$$\mathcal{Q}(x, \chi) = \min_{y \geq 0} \quad d^T y$$

$$\text{s.t.} \quad Wy \geq h(\chi) - T(\chi)x.$$

$x \in \mathbb{R}^{n_1}$: 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 with fixed recourse

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

$x \in \mathbb{R}^{n_1}$: 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 with fixed recourse

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

$x \in \mathbb{R}^{n_1}$: 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

Basic Structure of L-shaped method

Basic Structure of L-shaped method

1) Solve current master problem

Current master problem

$$
\begin{aligned}
\min_{x \geq 0} \quad & c^T x + \theta \\
\text{s.t.} \quad & Ax \geq b. \\
& D_\ell x \geq d_\ell \quad (\ell = 1, \ldots, r) \\
& G_\ell x + \theta \geq g_\ell \quad (\ell = 1, \ldots, s)
\end{aligned}
$$

Linköping University

Basic Structure of L-shaped method

1) Solve current master problem

Basic Structure of L-shaped method

1) Solve current master problem
2) As long as second-stage problem infeasible:
   Add feasibility cuts to master problem.

Basic Structure of L-shaped method

1) Solve current master problem
2) As long as second-stage problem infeasible:
   Add feasibility cuts to master problem.
3) If solution optimal: Stop.

Basic Structure of L-shaped method

1) Solve current master problem
2) As long as second-stage problem infeasible:
   Add feasibility cuts to master problem.
3) If solution optimal: Stop.

Basic Structure of L-shaped method

1) Solve current master problem
2) As long as second-stage problem infeasible:
   Add feasibility cuts to master problem.
3) If solution optimal: Stop.
   Otherwise: Add optimality cut to master problem. Go back to 1).

Optimality cut

Idea

Optimality cut

Idea

Approximate $\theta \geq \mathbb{E}[\mathcal{Q}(x, \chi)]$ by linear inequalities

Optimality cut

Idea

Approximate $\theta \geq \mathbb{E}[\mathcal{Q}(x, \chi)]$ by linear inequalities

- x^ν, θ^ν: Optimal solution of master problem in iteration ν

Optimality cut

Idea

Approximate $\theta \geq \mathbb{E}[\mathcal{Q}(x, \chi)]$ by linear inequalities

- x^ν, θ^ν: Optimal solution of master problem in iteration ν
- π_k^ν: Optimal solution of dual of $\mathcal{Q}(x^\nu, \chi^k)$

Optimality cut

Idea

Approximate $\theta \geq \mathbb{E}[\mathcal{Q}(x, \chi)]$ by linear inequalities

- x^ν, θ^ν: Optimal solution of master problem in iteration ν
- π_k^ν: Optimal solution of dual of $\mathcal{Q}(x^\nu, \chi^k)$

Optimality cut

Idea

Approximate $\theta \geq \mathbb{E}[\mathcal{Q}(x, \chi)]$ by linear inequalities

- x^ν, θ^ν: Optimal solution of master problem in iteration ν
- π_k^ν: Optimal solution of dual of $\mathcal{Q}(x^\nu, \chi^k)$

Optimality cut

$$\theta \geq \sum_{k=1}^{K} p^k (\pi_k^\nu)^T (h(\chi^k) - T(\chi^k)x)$$

Feasibility cut

First:

Feasibility cut

First:

Test feasibility of optimal solution of master problem by computing:

$$\begin{aligned} z_k = \min \quad & \mathbb{1}^T v_k^+ \\ \text{s.t.} \quad & W v_k + v_k^+ \geq h(\chi^k) - T(\chi^k) x^\nu, \\ & v_k, v_k^+ \geq 0. \end{aligned}$$

Feasibility cut

First:

Test feasibility of optimal solution of master problem by computing:

$$\begin{aligned} z_k = \min \quad & \mathbb{1}^T v_k^+ \\ \text{s.t.} \quad & W v_k + v_k^+ \geq h(\chi^k) - T(\chi^k) x^\nu, \\ & v_k, v_k^+ \geq 0. \end{aligned}$$

If $z_k = 0$:

Feasibility cut

First:

Test feasibility of optimal solution of master problem by computing:

$$
\begin{aligned}
z_k = \min \quad & \mathbb{1}^T v_k^+ \\
\text{s.t.} \quad & W v_k + v_k^+ \geq h(\chi^k) - T(\chi^k) x^\nu, \\
& v_k, v_k^+ \geq 0.
\end{aligned}
$$

If $z_k = 0$:

Linköping University

Feasibility cut

First:

Test feasibility of optimal solution of master problem by computing:

$$\begin{aligned} z_k = \min \quad & \mathbb{1}^T v_k^+ \\ \text{s.t.} \quad & W v_k + v_k^+ \geq h(\chi^k) - T(\chi^k) x^\nu, \\ & v_k, v_k^+ \geq 0. \end{aligned}$$

If $z_k = 0$:

x^ν is 2.-s. feasible

Feasibility cut

First:

Test feasibility of optimal solution of master problem by computing:

$$
\begin{aligned}
z_k = \min \quad & \mathbb{1}^T v_k^+ \\
\text{s.t.} \quad & W v_k + v_k^+ \geq h(\chi^k) - T(\chi^k) x^\nu, \\
& v_k, v_k^+ \geq 0.
\end{aligned} \tag{4a}
$$

Feasibility cut

First:

Test feasibility of optimal solution of master problem by computing:

$$\begin{aligned} z_k = \min \quad & \mathbb{1}^T v_k^+ \\ \text{s.t.} \quad & W v_k + v_k^+ \geq h(\chi^k) - T(\chi^k) x^\nu, \\ & v_k, v_k^+ \geq 0. \end{aligned} \tag{4a}$$

If $z_k > 0$:

x^ν is *not* 2.-s. feasible $\Rightarrow$ Add feasibility cut

Feasibility cut II

Theory

Consider dual:

$$\begin{aligned} 0 < z_k = \max \quad & \sigma^T (h(\chi^k) - T(\chi^k) x^\nu) \\ \text{s.t.} \quad & \sigma^T W \leq 0, \\ & \sigma \leq \mathbb{1}. \end{aligned} \tag{5a}$$

Feasibility cut II

Theory

Consider dual:

$$
\begin{aligned}
0 < z_k = \max \quad & \sigma^T(h(\chi^k) - T(\chi^k)x^\nu) \\
\text{s.t.} \quad & \sigma^T W \leq 0, \\
& \sigma \leq \mathbb{1}.
\end{aligned} \tag{5a}
$$

- σ_k^ν: Optimal solution of above dual problem

Feasibility cut II

Theory

Consider dual:

$$\begin{aligned} 0 < z_k = \max \quad & \sigma^T (h(\chi^k) - T(\chi^k) x^\nu) \\ \text{s.t.} \quad & \sigma^T W \leq 0, \\ & \sigma \leq \mathbb{1}. \end{aligned} \tag{5a}$$

- σ_k^ν: Optimal solution of above dual problem

Feasibility cut

$$\sigma_k^{\nu\,T} (h(\chi^k) - T(\chi^k) x) \leq 0$$

Linköping University

L-Shaped Algorithm

L-Shaped Algorithm

$r, s, \nu \leftarrow 0$

Linköping University

L-Shaped Algorithm

$r, s, \nu \leftarrow 0$

`while` $1 \neq 0$ `do`

  $\nu \leftarrow \nu + 1$

`end while`

L-Shaped Algorithm

$r, s, \nu \leftarrow 0$
`while` $1 \neq 0$ `do`
  $\nu \leftarrow \nu + 1$
  `Solve Current Master Problem (CMP):`

`end while`

L-Shaped Algorithm

$r, s, \nu \leftarrow 0$

`while` $1 \neq 0$ `do`

  $\nu \leftarrow \nu + 1$

  `Solve Current Master Problem (CMP):`

$$\begin{aligned} \min_{x \geq 0} \quad & c^T x + \theta \\ \text{s.t.} \quad & Ax \geq b. \\ & D_\ell x \geq d_\ell \quad (\ell = 1, \dots, r) \\ & G_\ell x + \theta \geq g_\ell \quad (\ell = 1, \dots, s) \end{aligned}$$

`end while`

L-Shaped Algorithm

$r, s, \nu \leftarrow 0$
`while` $1 \neq 0$ `do`
  $\nu \leftarrow \nu + 1$
  `Solve Current Master Problem (CMP)` $\rightarrow x^{\nu}$, θ^{ν}

`end while`

L-Shaped Algorithm

$r, s, \nu \leftarrow 0$
`while` $1 \neq 0$ `do`
  $\nu \leftarrow \nu + 1$
  `Solve Current Master Problem (CMP)` $\rightarrow x^{\nu}$, θ^{ν}
  `if` x^{ν} `not 2.-s. feasible then`

  `end if`

`end while`

L-Shaped Algorithm

```
r, s, ν ← 0
while 1 ≠ 0 do
  ν ← ν + 1
  Solve Current Master Problem (CMP)  → x^ν, θ^ν
  if x^ν not 2.-s. feasible then
    Add feasibility cut (r ← r + 1)

  end if

end while
```

L-Shaped Algorithm

$r, s, \nu \leftarrow 0$
`while` $1 \neq 0$ `do`
  $\nu \leftarrow \nu + 1$
  `Solve Current Master Problem (CMP)` $\rightarrow x^\nu$, θ^ν
  `if` x^ν `not 2.-s. feasible then`
    `Add feasibility cut` ($r \leftarrow r + 1$)
    `Go back: Resolve CMP`
  `end if`

`end while`

L-Shaped Algorithm

$r, s, \nu \leftarrow 0$
`while` $1 \neq 0$ `do`
  $\nu \leftarrow \nu + 1$
  `Solve Current Master Problem (CMP)` $\rightarrow x^\nu$, θ^ν
  `if` x^ν `not 2.-s. feasible then`
    `Add feasibility cut` $(r \leftarrow r + 1)$
    `Go back: Resolve CMP`
  `end if`
  `Add optimality cut` $(s \leftarrow s + 1)$

`end while`

L-Shaped Algorithm

$r, s, \nu \leftarrow 0$
`while` $1 \neq 0$ `do`
  $\nu \leftarrow \nu + 1$
  `Solve Current Master Problem (CMP)` $\rightarrow$ x^ν, θ^ν
  `if` x^ν `not 2.-s. feasible then`
    `Add feasibility cut` $(r \leftarrow r + 1)$
    `Go back: Resolve CMP`
  `end if`
  `Add optimality cut` $(s \leftarrow s + 1)$
  `if` x^ν, θ^ν `satisfy optimality cut then`

  `end if`
`end while`

L-Shaped Algorithm

```
r, s, ν ← 0
while 1 ≠ 0 do
  ν ← ν + 1
  Solve Current Master Problem (CMP)  → x^ν, θ^ν
  if x^ν not 2.-s.  feasible then
    Add feasibility cut (r ← r + 1)
    Go back:  Resolve CMP
  end if
  Add optimality cut (s ← s + 1)
  if x^ν, θ^ν satisfy optimality cut then
    STOP. x^ν is optimal solution.


  end if
end while
```

L-Shaped Algorithm

$r, s, \nu \leftarrow 0$
`while` $1 \neq 0$ `do`
  $\nu \leftarrow \nu + 1$
  `Solve Current Master Problem (CMP)` $\rightarrow x^\nu, \theta^\nu$
  `if` x^ν `not 2.-s. feasible then`
    `Add feasibility cut` $(r \leftarrow r + 1)$
    `Go back: Resolve CMP`
  `end if`
  `Add optimality cut` $(s \leftarrow s + 1)$
  `if` x^ν, θ^ν `satisfy optimality cut then`
    `STOP.` x^ν `is optimal solution.`
  `else`
    `Go back: Resolve CMP`
  `end if`
`end while`

Results

Results

- Only finitely many cuts needed to obtain feasibility

Results

- Only finitely many cuts needed to obtain feasibility
- BUT: Number can be large!

Results

- Only finitely many cuts needed to obtain feasibility
- BUT: Number can be large!
- HOWEVER: Feasibility cut has "deepest cut property"

Results

- Only finitely many cuts needed to obtain feasibility
- BUT: Number can be large!
- HOWEVER: Feasibility cut has "deepest cut property"
- Algorithm stops after finitely many iterations

Outline

Inner Approximation

Inner Approximation

- Randomized Solution Algorithm
- Sampling during solution process
- Either: Find good solution over iterations
- Or: Problem approximated over iterations

Inner Approximation

- Randomized Solution Algorithm
- Sampling during solution process
- Either: Find good solution over iterations
- Or: Problem approximated over iterations
- Famous examples:

  $\rightarrow$ Stochastic gradient algorithm (Stochastic approximation)
  $\rightarrow$ Stochastic Decomposition

Outline

Used in case where...

Used in case where...

- ...underlying distribution is continuous.

Used in case where...

- ...underlying distribution is continuous.
- ...underlying discrete distribution intractable.

Used in case where...

- ...underlying distribution is continuous.
- ...underlying discrete distribution intractable.
- ...SAA error bound too pessimistic.

Linköping University

Used in case where...

- ...underlying distribution is continuous.
- ...underlying discrete distribution intractable.
- ...SAA error bound too pessimistic.

Used in case where...

- ...underlying distribution is continuous.
- ...underlying discrete distribution intractable.
- ...SAA error bound too pessimistic.

Reference

Used in case where...

- ...underlying distribution is continuous.
- ...underlying discrete distribution intractable.
- ...SAA error bound too pessimistic.

Reference

Julia L. Higle and Suvrajeet Sen
Stochastic decomposition: An algorithm for two-stage linear programs with recourse.
Mathematics of Operations Research 16(3):650–669, 1991

Basic idea

Basic idea

- Basically: L-shaped method

Basic idea

- Basically: L-shaped method
- BUT: set of considered scenarios continuously extended

Basic idea

- Basically: L-shaped method
- BUT: set of considered scenarios continuously extended
- $\Rightarrow$ Cuts computed based on "incomplete" information

Problem considered in iteration ν

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

Problem considered in iteration ν

$$\begin{aligned}
\min_{\substack{x \geq 0 \\ \theta \geq 0}} \quad & c^T x + \theta \\
\text{s.t.} \quad & Ax \geq b, \\
& \theta \geq \sum_{k=1}^{\nu'} \frac{1}{\nu} \mathcal{Q}(x, \chi^k) \\
& \mathcal{Q}(x, \chi) = \min_{y \geq 0} \quad d^T y \\
& \qquad\qquad\qquad \text{s.t.} \quad Wy \geq h(\chi) - T(\chi)x.
\end{aligned}$$

Problem considered in iteration ν

$$
\begin{aligned}
\min_{\substack{x\geq 0\\ \theta\geq 0}} \quad & c^T x + \theta \\
\text{s.t.} \quad & Ax \geq b, \\
& \theta \geq \sum_{k=1}^{\nu'} \frac{1}{\nu} \mathcal{Q}(x, \chi^k) \\
& \mathcal{Q}(x,\chi) = \min_{y\geq 0} \quad d^T y \\
& \qquad\qquad\qquad \text{s.t.} \quad Wy \geq h(\chi) - T(\chi)x.
\end{aligned}
$$

χ^k: sample from iteration k

Basic idea

- Basically: L-shaped method
- BUT: set of considered scenarios continuously extended
- $\Rightarrow$ Cuts computed based on "incomplete" information

Basic idea

- Basically: L-shaped method
- BUT: set of considered scenarios continuously extended
- $\Rightarrow$ Cuts computed based on "incomplete" information
- $\Rightarrow$ 2.-s. feasibility and optimality only with certain probability

Basic idea

- Basically: L-shaped method
- BUT: set of considered scenarios continuously extended
- $\Rightarrow$ Cuts computed based on "incomplete" information
- $\Rightarrow$ 2.-s. feasibility and optimality only with certain probability

Basic idea

- Basically: L-shaped method
- BUT: set of considered scenarios continuously extended
- $\Rightarrow$ Cuts computed based on "incomplete" information
- $\Rightarrow$ 2.-s. feasibility and optimality only with certain probability
- In each iteration: Only (exactly) solve 2.-s. problem for last added outcome

Definition

A two-stage stochastic programming problem has

relative complete recourse

if

- $\forall$ feasible 1.-s. solutions $x \in \mathbb{R}^n$ and
- $\forall\ \hat{\chi} \in \Omega$

$\exists$ **a feasible 2.-s.-solution**.

Definition

A two-stage stochastic programming problem has

relative complete recourse

if

- $\forall$ feasible 1.-s. solutions $x \in \mathbb{R}^n$ and
- $\forall \ \hat{\chi} \in \Omega$

$\exists$ **a feasible 2.-s.-solution**.

Definition

A two-stage stochastic programming problem has

relative complete recourse

if

- $\forall$ feasible 1.-s. solutions $x \in \mathbb{R}^n$ and
- $\forall\ \hat{\chi} \in \Omega$

$\exists$ **a feasible 2.-s.-solution**.

Definition

A two-stage stochastic programming problem has

relative complete recourse

if

- $\forall$ feasible 1.-s. solutions $x \in \mathbb{R}^n$ and
- $\forall\ \hat{\chi} \in \Omega$

$\exists$ **a feasible 2.-s.-solution**.

In other words:

Definition

A two-stage stochastic programming problem has

relative complete recourse

if

- $\forall$ feasible 1.-s. solutions $x \in \mathbb{R}^n$ and
- $\forall\, \hat{\chi} \in \Omega$

$\exists$ **a feasible 2.-s.-solution**.

In other words:

$\forall\, x \in X$ and $\forall\, \hat{\chi} \in \Omega$:

Definition

A two-stage stochastic programming problem has

relative complete recourse

if

- $\forall$ feasible 1.-s. solutions $x \in \mathbb{R}^n$ and
- $\forall\ \hat{\chi} \in \Omega$

$\exists$ **a feasible 2.-s.-solution**.

In other words:

$\forall\, x \in X$ and $\forall\, \hat{\chi} \in \Omega$:

- $\mathcal{Q}(x, \chi) < \infty$.

Definition

A two-stage stochastic programming problem has

relative complete recourse

if

- $\forall$ feasible 1.-s. solutions $x \in \mathbb{R}^n$ and
- $\forall\ \hat{\chi} \in \Omega$

$\exists$ **a feasible 2.-s.-solution**.

In other words:

$\forall x \in X$ and $\forall \hat{\chi} \in \Omega$:

- $\mathcal{Q}(x, \chi) < \infty$.
- x is 2.-s. feasible.

Definition

A two-stage stochastic programming problem has

relative complete recourse

if

- $\forall$ feasible 1.-s. solutions $x \in \mathbb{R}^n$ and
- $\forall\, \hat{\chi} \in \Omega$

$\exists$ **a feasible 2.-s.-solution**.

In other words:

$\forall\, x \in X$ and $\forall\, \hat{\chi} \in \Omega$:

- $\mathcal{Q}(x, \chi) < \infty$.
- x is 2.-s. feasible.
- $h(\hat{\chi}) - T(\hat{\chi})x \in \text{pos}\, W(\hat{\chi}) := \{t | \exists y \geq 0 : W(\hat{\chi})y \geq t\}$.

Definition

A two-stage stochastic programming problem has

relative complete recourse

if

- $\forall$ feasible 1.-s. solutions $x \in \mathbb{R}^n$ and
- $\forall\ \hat{\chi} \in \Omega$

$\exists$ **a feasible 2.-s.-solution**.

Definition

A two-stage stochastic programming problem has

relative complete recourse

if

- $\forall$ feasible 1.-s. solutions $x \in \mathbb{R}^n$ and
- $\forall\ \hat{\chi} \in \Omega$

$\exists$ **a feasible 2.-s.-solution**.

Consequently:

Linköping University

Definition

A two-stage stochastic programming problem has

relative complete recourse

if

- $\forall$ feasible 1.-s. solutions $x \in \mathbb{R}^n$ and
- $\forall\ \hat{\chi} \in \Omega$

$\exists$ **a feasible 2.-s.-solution**.

Consequently:

No feasibility cuts needed!

Assumptions

Assumptions

- Relatively complete recourse.

Assumptions

- Relatively complete recourse.
- Fixed recourse.

Assumptions

- Relatively complete recourse.
- Fixed recourse.
- $X \times \Omega$ is compact.

Assumptions

- Relatively complete recourse.
- Fixed recourse.
- $X \times \Omega$ is compact.
- Deterministic technology matrix.

Assumptions

- Relatively complete recourse.
- Fixed recourse.
- $X \times \Omega$ is compact.
- ~~Deterministic technology matrix.~~

Assumptions

- Relatively complete recourse.
- Fixed recourse.
- $X \times \Omega$ is compact.
- ~~Deterministic technology matrix.~~
- $\forall x \in X \; \mathcal{Q}(x, \chi) \geq 0$ (w.p.1)

Linköping University

Optimality cut

In iteration ν (after master problem has been solved)

Optimality cut

In iteration ν (after master problem has been solved)

- Draw sample χ^ν of χ

Optimality cut

In iteration ν (after master problem has been solved)

- Draw sample χ^ν of χ
- Solve 2.-s. problem:

$$\begin{aligned} \max_{\pi \geq 0} \quad & \pi^T (h(\chi^\nu) - T(\chi^\nu) x^\nu) \\ \text{s.t.} \quad & \pi^T W \leq d. \end{aligned}$$

Optimality cut

In iteration ν (after master problem has been solved)

- Draw sample χ^ν of χ
- Solve 2.-s. problem: $(\rightarrow \pi_\nu^\nu)$

$$\begin{aligned} \max_{\pi \geq 0} \quad & \pi^T (h(\chi^\nu) - T(\chi^\nu) x^\nu) \\ \text{s.t.} \quad & \pi^T W \leq d. \end{aligned}$$

ersity

Optimality cut

In iteration ν (after master problem has been solved)

- Draw sample χ^ν of χ
- Solve 2.-s. problem: $(\rightarrow \pi^\nu_\nu)$

$$\begin{aligned} \max_{\pi \geq 0} \quad & \pi^T(h(\chi^\nu) - T(\chi^\nu)x^\nu) \\ \text{s.t.} \quad & \pi^T W \leq d. \end{aligned}$$

- Add π^ν_ν to list of solutions ($V^\nu \leftarrow V^{\nu-1} \cup \{\pi^\nu_\nu\}$)

Optimality cut

In iteration ν (after master problem has been solved)

- Draw sample χ^ν of χ
- Solve 2.-s. problem: $(\rightarrow \pi^\nu_\nu)$

$$\begin{aligned} \max_{\pi \geq 0} \quad & \pi^T (h(\chi^\nu) - T(\chi^\nu) x^\nu) \\ \text{s.t.} \quad & \pi^T W \leq d. \end{aligned}$$

- Add π^ν_ν to list of solutions ($V^\nu \leftarrow V^{\nu-1} \cup \{\pi^\nu_\nu\}$)
- $\forall k = 1, \ldots, \nu - 1$ solve:

$$\begin{aligned} \max_{\pi \geq 0} \quad & \pi^T (h(\chi^k) - T(\chi^k) x^\nu) \\ \text{s.t.} \quad & \pi^T \in V^\nu. \end{aligned}$$

ersity

Optimality cut

In iteration ν (after master problem has been solved)

- Draw sample χ^ν of χ
- Solve 2.-s. problem: $(\rightarrow \pi_\nu^\nu)$

$$\begin{aligned} \max_{\pi \geq 0} \quad & \pi^T (h(\chi^\nu) - T(\chi^\nu) x^\nu) \\ \text{s.t.} \quad & \pi^T W \leq d. \end{aligned}$$

- Add π_ν^ν to list of solutions $(V^\nu \leftarrow V^{\nu-1} \cup \{\pi_\nu^\nu\})$
- $\forall k = 1, \ldots, \nu - 1$ solve: $(\rightarrow \pi_k^\nu,\ k = 1, \ldots, \nu - 1)$

$$\begin{aligned} \max_{\pi \geq 0} \quad & \pi^T (h(\chi^k) - T(\chi^k) x^\nu) \\ \text{s.t.} \quad & \pi^T \in V^\nu. \end{aligned}$$

ersity

Optimality cut II

$$\tilde{\pi}_k \quad := \quad \arg\max_{\pi \geq 0} \quad \{\pi^T(h(\chi^k) - T(\chi^k)x) | \quad \pi^T W \leq d\}$$

Optimality cut II

$$\tilde{\pi}_k \quad := \quad \arg\max_{\pi \geq 0} \quad \{\pi^T(h(\chi^k) - T(\chi^k)x) | \quad \pi^T W \leq d\}$$

Theory

Optimality cut II

$$\tilde{\pi}_k \quad := \quad \arg\max_{\pi \geq 0} \quad \{\pi^T(h(\chi^k) - T(\chi^k)x) | \quad \pi^T W \leq d\}$$

Theory

1) $\mathcal{Q}(x, \chi^k) = \tilde{\pi}_k(h(\chi^k) - T(\chi^k)x)$

Optimality cut II

$$\tilde{\pi}_k \quad := \quad \arg\max_{\pi \geq 0} \quad \{\pi^T(h(\chi^k) - T(\chi^k)x) | \quad \pi^T W \leq d\}$$

Theory

1) $\mathcal{Q}(x, \chi^k) = \tilde{\pi}_k(h(\chi^k) - T(\chi^k)x) \quad \geq \quad \pi_k^\nu(h(\chi^k) - T(\chi^k)x)$

Optimality cut II

$$\tilde{\pi}_k \quad := \quad \arg\max_{\pi \geq 0} \quad \{\pi^T(h(\chi^k) - T(\chi^k)x) | \quad \pi^T W \leq d\}$$

Theory

1) $\mathcal{Q}(x, \chi^k) = \tilde{\pi}_k(h(\chi^k) - T(\chi^k)x) \quad \geq \quad \pi_k^\nu(h(\chi^k) - T(\chi^k)x)$

2)

$$\theta \geq \frac{1}{\nu}\sum_{k=1}^{\nu} \mathcal{Q}(x, \chi^k) \quad \Rightarrow \quad \theta \geq \frac{1}{\nu}\sum_{k=1}^{\nu} \pi_k^\nu(h(\chi^k) - T(\chi^k)x)$$

Optimality cut II

$$\tilde{\pi}_k \quad := \quad \arg\max_{\pi \geq 0} \quad \{\pi^T(h(\chi^k) - T(\chi^k)x) | \quad \pi^T W \leq d\}$$

Theory

1) $\mathcal{Q}(x, \chi^k) = \tilde{\pi}_k(h(\chi^k) - T(\chi^k)x) \quad \geq \quad \pi_k^\nu(h(\chi^k) - T(\chi^k)x)$
2) $$\theta \geq \frac{1}{\nu}\sum_{k=1}^{\nu} \mathcal{Q}(x, \chi^k) \quad \Rightarrow \quad \theta \geq \frac{1}{\nu}\sum_{k=1}^{\nu} \pi_k^\nu(h(\chi^k) - T(\chi^k)x)$$

New Optimality cut

ersity

Optimality cut II

$$\tilde{\pi}_k \quad := \quad \arg\max_{\pi \geq 0} \quad \{\pi^T(h(\chi^k) - T(\chi^k)x)| \quad \pi^T W \leq d\}$$

Theory

1) $\mathcal{Q}(x, \chi^k) = \tilde{\pi}_k(h(\chi^k) - T(\chi^k)x) \quad \geq \quad \pi_k^\nu(h(\chi^k) - T(\chi^k)x)$

2)

$$\theta \geq \frac{1}{\nu}\sum_{k=1}^{\nu} \mathcal{Q}(x, \chi^k) \quad \Rightarrow \quad \theta \geq \frac{1}{\nu}\sum_{k=1}^{\nu} \pi_k^\nu(h(\chi^k) - T(\chi^k)x)$$

New Optimality cut

$$\theta \geq \frac{1}{\nu}\sum_{k=1}^{\nu} \pi_k^\nu(h(\chi^k) - T(\chi^k)x)$$

Optimality cut III

$$\tilde{\pi}_k \quad := \quad \arg\max_{\pi \geq 0} \quad \{\pi^T(h(\chi^k) - T(\chi^k)x) | \quad \pi^T W \leq d\}$$

Needed

Optimality cut III

$$\tilde{\pi}_k \quad := \quad \arg\max_{\pi \geq 0} \quad \{\pi^T(h(\chi^k) - T(\chi^k)x) | \quad \pi^T W \leq d\}$$

Needed

$\forall \nu' > \nu$:

$$\theta \geq \frac{1}{\nu'} \sum_{k=1}^{\nu'} \mathcal{Q}(x, \chi^k) \quad \Rightarrow \quad \theta \geq \frac{1}{\nu} \sum_{k=1}^{\nu} \pi_k^{\nu}(h(\chi^k) - T(\chi^k)x)$$

Optimality cut III

$$\tilde{\pi}_k \quad := \quad \arg\max_{\pi \geq 0} \quad \{\pi^T(h(\chi^k) - T(\chi^k)x)| \quad \pi^T W \leq d\}$$

However...

$\forall \nu' > \nu$:

$$\theta \geq \frac{1}{\nu'} \sum_{k=1}^{\nu'} \mathcal{Q}(x, \chi^k) \quad \not\Rightarrow \quad \theta \geq \frac{1}{\nu} \sum_{k=1}^{\nu} \pi_k^{\nu}(h(\chi^k) - T(\chi^k)x)$$

Optimality cut III

$$\tilde{\pi}_k \quad := \quad \arg\max_{\pi \geq 0} \quad \{\pi^T (h(\chi^k) - T(\chi^k)x) | \quad \pi^T W \leq d\}$$

However...

$\forall \nu' > \nu$:

$$\theta \geq \frac{1}{\nu'} \sum_{k=1}^{\nu'} \mathcal{Q}(x, \chi^k) \quad \Rightarrow \quad \theta \geq \frac{1}{\nu'} \sum_{k=1}^{\nu} \pi_k^{\nu} (h(\chi^k) - T(\chi^k)x)$$

Optimality cut II

$$\tilde{\pi}_k \quad := \quad \arg\max_{\pi \geq 0} \quad \{\pi^T(h(\chi^k) - T(\chi^k)x) | \quad \pi^T W \leq d\}$$

Theory

1) $\mathcal{Q}(x, \chi^k) = \tilde{\pi}_k(h(\chi^k) - T(\chi^k)x) \quad \geq \quad \pi_k^\nu(h(\chi^k) - T(\chi^k)x)$

2)

$$\theta \geq \frac{1}{\nu}\sum_{k=1}^{\nu} \mathcal{Q}(x, \chi^k) \quad \Rightarrow \quad \theta \geq \frac{1}{\nu}\sum_{k=1}^{\nu} \pi_k^\nu(h(\chi^k) - T(\chi^k)x)$$

New Optimality cut

$$\theta \geq \frac{1}{\nu}\sum_{k=1}^{\nu} \pi_k^\nu(h(\chi^k) - T(\chi^k)x)$$

ersity

Optimality cut III

$$\tilde{\pi}_k \quad := \quad \arg\max_{\pi \geq 0} \quad \{\pi^T(h(\chi^k) - T(\chi^k)x)| \quad \pi^T W \leq d\}$$

However...

$\forall \nu' > \nu$:

$$\theta \geq \frac{1}{\nu'} \sum_{k=1}^{\nu'} \mathcal{Q}(x, \chi^k) \quad \Rightarrow \quad \theta \geq \frac{1}{\nu'} \sum_{k=1}^{\nu} \pi_k^{\nu}(h(\chi^k) - T(\chi^k)x)$$

Linköping University

Optimality cut III

$$\tilde{\pi}_k \quad := \quad \arg\max_{\pi \geq 0} \quad \{\pi^T(h(\chi^k) - T(\chi^k)x) | \quad \pi^T W \leq d\}$$

However...

$\forall \nu' > \nu$:

$$\theta \geq \frac{1}{\nu'} \sum_{k=1}^{\nu'} \mathcal{Q}(x, \chi^k) \quad \Rightarrow \quad \theta \geq \frac{1}{\nu'} \sum_{k=1}^{\nu} \pi_k^{\nu}(h(\chi^k) - T(\chi^k)x)$$

Update existing optimality cuts...

Linköping University

Optimality cut III

$$\tilde{\pi}_k \quad := \quad \arg\max_{\pi \geq 0} \quad \{\pi^T(h(\chi^k) - T(\chi^k)x) | \quad \pi^T W \leq d\}$$

However...

$\forall \nu' > \nu$:

$$\theta \geq \frac{1}{\nu'} \sum_{k=1}^{\nu'} \mathcal{Q}(x, \chi^k) \quad \Rightarrow \quad \theta \geq \frac{1}{\nu'} \sum_{k=1}^{\nu} \pi_k^{\nu}(h(\chi^k) - T(\chi^k)x)$$

Update existing optimality cuts...

... by multiplying right hand side by $\frac{\nu-1}{\nu}$ (in iteration ν).

Theorem (Consistency)

Theorem (Consistency)

- $\{x^{\nu_n}\}_{n=1}^{\infty}$: *infinite subsequence of* $\{x^{\nu}\}_{\nu=1}^{\infty}$

Linköping University

Theorem (Consistency)

- $\{x^{\nu_n}\}_{n=1}^{\infty}$: *infinite subsequence of* $\{x^{\nu}\}_{\nu=1}^{\infty}$
- $x^{\nu_n} \longrightarrow \hat{x}$

Linköping University

Theorem (Consistency)

- $\{x^{\nu_n}\}_{n=1}^{\infty}$: *infinite subsequence of* $\{x^{\nu}\}_{\nu=1}^{\infty}$
- $x^{\nu_n} \longrightarrow \hat{x}$

Linköping University

Theorem (Consistency)

- $\{x^{\nu_n}\}_{n=1}^{\infty}$: *infinite subsequence of* $\{x^{\nu}\}_{\nu=1}^{\infty}$
- $x^{\nu_n} \to \hat{x}$

Then (w.p.1):

$$\frac{1}{\nu_n} \sum_{t=1}^{\nu_n} \pi_t^{\nu_n} (h(\chi^t) - T(\chi^t) x^{\nu_n}) \to \mathbb{E}[\mathcal{Q}(\hat{x}, \chi)]$$

Theorem (Convergence)

Theorem (Convergence)

$\exists$ **infinite subsequence** $\{x^{\nu_n}\}_{n=1}^{\infty}$ *of* $\{x^{\nu}\}_{\nu=1}^{\infty}$

s.t. (w.p.1)

every **accumulation point** *of* $\{x^{\nu_n}\}_{n=1}^{\infty}$ *is an* **optimal solution**.

Theorem (Convergence)

$\exists$ **infinite subsequence** $\{x^{\nu_n}\}_{n=1}^{\infty}$ *of* $\{x^{\nu}\}_{\nu=1}^{\infty}$

s.t. (w.p.1)

every **accumulation point** *of* $\{x^{\nu_n}\}_{n=1}^{\infty}$ *is an* **optimal solution**.

Theorem (Convergence)

$\exists$ **infinite subsequence** $\{x^{\nu_n}\}_{n=1}^{\infty}$ *of* $\{x^{\nu}\}_{\nu=1}^{\infty}$

s.t. (w.p.1)

every **accumulation point** *of* $\{x^{\nu_n}\}_{n=1}^{\infty}$ *is an* **optimal solution**.

Problem

Theorem (Convergence)

$\exists$ **infinite subsequence** $\{x^{\nu_n}\}_{n=1}^{\infty}$ *of* $\{x^{\nu}\}_{\nu=1}^{\infty}$

s.t. (w.p.1)

every **accumulation point** *of* $\{x^{\nu_n}\}_{n=1}^{\infty}$ *is an* **optimal solution**.

Problem

How to identify this subsequence?

Theorem (Convergence)

$\exists$ **infinite subsequence** $\{x^{\nu_n}\}_{n=1}^{\infty}$ *of* $\{x^{\nu}\}_{\nu=1}^{\infty}$

s.t. (w.p.1)

every **accumulation point** *of* $\{x^{\nu_n}\}_{n=1}^{\infty}$ *is an* **optimal solution**.

Problem

How to identify this subsequence?

Solution (Higle, Sen '91)

Theorem (Convergence)

$\exists$ **infinite subsequence** $\{x^{\nu_n}\}_{n=1}^{\infty}$ *of* $\{x^{\nu}\}_{\nu=1}^{\infty}$

s.t. (w.p.1)

every **accumulation point** *of* $\{x^{\nu_n}\}_{n=1}^{\infty}$ *is an* **optimal solution**.

Problem

How to identify this subsequence?

Solution (Higle, Sen '91)

Take iterates whose estimated objective value is "sufficient low".

Linköping University

Outline

Definition

- $\sharp P$**:** Counting problems associated with problems in NP

Definition

- $\sharp P$**:** Counting problems associated with problems in NP
- $\sharp P$-**hard:**
  Every problem in $\sharp P$ can be reduced to it

Definition

- $\sharp P$**:** Counting problems associated with problems in NP
- $\sharp P$-**hard:**
  Every problem in $\sharp P$ can be reduced to it

Definition

- $\sharp P$**:** Counting problems associated with problems in NP
- $\sharp P$-**hard:**
  Every problem in $\sharp P$ can be reduced to it

$\sharp P$-hard problems

Definition

- $\sharp P$**:** Counting problems associated with problems in NP
- $\sharp P$-**hard:**
  Every problem in $\sharp P$ can be reduced to it

$\sharp P$-hard problems

- "How many graph colorings using k colors are there for a particular graph G?"

Definition

- $\sharp P$**:** Counting problems associated with problems in NP
- $\sharp P$-**hard:**
  Every problem in $\sharp P$ can be reduced to it

$\sharp P$-hard problems

- "How many graph colorings using k colors are there for a particular graph G?"
- "How many perfect matchings are there for a given bipartite graph?"

Definition

- $\sharp P$: Counting problems associated with problems in NP
- $\sharp P$-**hard:**
  Every problem in $\sharp P$ can be reduced to it

$\sharp P$-hard problems

- "How many graph colorings using k colors are there for a particular graph G?"
- "How many perfect matchings are there for a given bipartite graph?"

Definition

- $\sharp P$**:** Counting problems associated with problems in NP
- $\sharp P$-**hard:**
  Every problem in $\sharp P$ can be reduced to it

$\sharp P$-hard problems

- "How many graph colorings using k colors are there for a particular graph G?"
- "How many perfect matchings are there for a given bipartite graph?"

$\sharp P$-hard problem solvable in pol. time $\Rightarrow P = NP$

Theorem (Dyer, Stougie 2003)

Theorem (Dyer, Stougie 2003)

Linear Two-Stage Stochastic Programming with **discretely distributed** *parameters is* $\sharp P$*-hard.*

Theorem (Dyer, Stougie 2003)

Linear Two-Stage Stochastic Programming with **discretely distributed** *parameters is* $\sharp P$*-hard.*

Reference

Theorem (Dyer, Stougie 2003)

Linear Two-Stage Stochastic Programming with **discretely distributed** *parameters is* $\sharp P$*-hard.*

Reference

Martin Dyer, Leen Stougie
Computational complexity of stochastic programming problems. (2003)
http://www.win.tue.nl/bs/spor/2003-20.pdf

Theorem (Dyer, Stougie 2003)

Linear Two-Stage Stochastic Programming with discretely distributed parameters is $\sharp P$-hard.

Theorem (Dyer, Stougie 2003)

Linear Two-Stage Stochastic Programming with discretely distributed parameters is $\sharp P$-hard.

Proof

Theorem (Dyer, Stougie 2003)

Linear Two-Stage Stochastic Programming with discretely distributed parameters is $\sharp P$-hard.

Proof

Reduction from **Graph reliability problem**:

Linköping University

Theorem (Dyer, Stougie 2003)

Linear Two-Stage Stochastic Programming with discretely distributed parameters is $\sharp P$-hard.

Proof

Reduction from **Graph reliability problem**:
Given:

Linköping University

Theorem (Dyer, Stougie 2003)

Linear Two-Stage Stochastic Programming with discretely distributed parameters is $\sharp P$-hard.

Proof

Reduction from **Graph reliability problem**:
Given:

- Directed graph $G = (V, E)$ with random edges

Linköping University

Theorem (Dyer, Stougie 2003)

Linear Two-Stage Stochastic Programming with discretely distributed parameters is $\sharp P$-hard.

Proof

Reduction from **Graph reliability problem**:
Given:

- Directed graph $G = (V, E)$ with random edges
- $\forall e \in E$: $\mathbb{P}\{e \in E\} = \frac{1}{2}$

Theorem (Dyer, Stougie 2003)

Linear Two-Stage Stochastic Programming with discretely distributed parameters is $\sharp P$-hard.

Proof

Reduction from **Graph reliability problem**:
Given:

- Directed graph $G = (V, E)$ with random edges
- $\forall e \in E$: $\mathbb{P}\{e \in E\} = \frac{1}{2}$
- $u, v \in V$

Theorem (Dyer, Stougie 2003)

Linear Two-Stage Stochastic Programming with discretely distributed parameters is $\sharp P$-hard.

Proof

Reduction from **Graph reliability problem**:
Given:

- Directed graph $G = (V, E)$ with random edges
- $\forall e \in E$: $\mathbb{P}\{e \in E\} = \frac{1}{2}$
- $u, v \in V$

Linköping University

Theorem (Dyer, Stougie 2003)

Linear Two-Stage Stochastic Programming with discretely distributed parameters is $\sharp P$-hard.

Proof

Reduction from **Graph reliability problem**:
Given:

- Directed graph $G = (V, E)$ with random edges
- $\forall e \in E$: $\mathbb{P}\{e \in E\} = \frac{1}{2}$
- $u, v \in V$

Compute:

$$\mathbb{P}\{\exists \textbf{ u-v-path in G}\}$$

Linköping University

Theorem (Dyer, Stougie 2003)

Theorem (Dyer, Stougie 2003)

Linear Two-Stage Stochastic Programming with **continuously distributed** *parameters is $\sharp P$-hard.*

Theorem (Dyer, Stougie 2003)

Linear Two-Stage Stochastic Programming with **continuously distributed** *parameters is $\sharp P$-hard.*

Reference

Martin Dyer, Leen Stougie
Computational complexity of stochastic programming problems. (2003)
http://www.win.tue.nl/bs/spor/2003-20.pdf

QUESTIONS?

What about next week?