Stochastic Optimization
IDA PhD course 2011ht

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

8. Lecture: Decomposition Methods
01. December 2011

1. Outer and Inner Approximation Approaches
   - Sample Average Approximation
   - Stochastic Gradient method

Linköping University

Outline

Outer versus Inner Approximation

Outer versus Inner Approximation

Problem Approximation $\leftrightarrow$ Randomized Algorithm

Outer versus Inner Approximation

Problem Approximation $\leftrightarrow$ Randomized Algorithm

Sampling done before solution $\leftrightarrow$ Sampling during solution process

Outer versus Inner Approximation

Problem Approximation $\leftrightarrow$ Randomized Algorithm

Sampling done before solution $\leftrightarrow$ Sampling during solution process

Deterministic reformulation $\leftrightarrow$ Deterministic problems during solution

Outline

Linköping University

Stochastic Programming Model → Deterministic Equivalent Model

Stochastic Programming Model → Deterministic Equivalent Model

$$\min_{x \in X} \quad \mathbb{E}[f(x, \chi)]$$

$\chi \in \Omega \subseteq \mathbb{R}^s$: random vector

Stochastic Programming Model → Deterministic Equivalent Model

$$\min_{x \in X} \quad \mathbb{E}[f(x, \chi)]$$

$\chi \in \Omega \subseteq \mathbb{R}^s$: random vector

$\xrightarrow{SAA}$

Stochastic Programming Model → Deterministic Equivalent Model

$$\min_{x \in X} \quad \mathbb{E}[f(x, \chi)] \quad \xrightarrow{SAA} \quad \min_{x \in X} \quad \frac{1}{N} \sum_{k=1}^{N} f(x, \chi^k)$$

$\chi \in \Omega \subseteq \mathbb{R}^s$: random vector

Stochastic Programming Model $\rightarrow$ Deterministic Equivalent Model

$$\min_{x \in X} \quad \mathbb{E}[f(x,\chi)]$$

$\chi \in \Omega \subseteq \mathbb{R}^s$: random vector

$$\xrightarrow{SAA}$$

$$\min_{x \in X} \quad \frac{1}{N}\sum_{k=1}^{N} f(x,\chi^k)$$

$\chi^1,\ldots,\chi^N$: random sample

Stochastic Programming Model → Deterministic Equivalent Model

$$\min_{x \in X} \quad \mathbb{E}[f(x,\chi)] \quad \xrightarrow{SAA} \quad \min_{x \in X} \quad \frac{1}{N}\sum_{k=1}^{N} f(x,\chi^k)$$

$\chi \in \Omega \subseteq \mathbb{R}^s$: random vector

$\chi^1,\ldots,\chi^N$: random sample

Main Result

Linköping University

Stochastic Programming Model → Deterministic Equivalent Model

$$\min_{x \in X} \quad \mathbb{E}[f(x,\chi)] \quad \xrightarrow{SAA} \quad \min_{x \in X} \quad \frac{1}{N}\sum_{k=1}^{N} f(x,\chi^k)$$

$\chi \in \Omega \subseteq \mathbb{R}^s$: random vector

$\chi^1, \dots, \chi^N$: random sample

Main Result

Under some mild (technical) assumptions:

Linköping University

Stochastic Programming Model → Deterministic Equivalent Model

$$\min_{x \in X} \quad \mathbb{E}[f(x,\chi)] \quad \xrightarrow{SAA} \quad \min_{x \in X} \quad \frac{1}{N}\sum_{k=1}^{N} f(x,\chi^k)$$

$\chi \in \Omega \subseteq \mathbb{R}^s$: random vector

$\chi^1, \ldots, \chi^N$: random sample

Main Result

Under some mild (technical) assumptions:

$\forall \epsilon > 0$:

Stochastic Programming Model → Deterministic Equivalent Model

$$\min_{x \in X} \quad \mathbb{E}[f(x,\chi)]$$

$\chi \in \Omega \subseteq \mathbb{R}^s$: random vector

$$\xrightarrow{SAA}$$

$$\min_{x \in X} \quad \frac{1}{N} \sum_{k=1}^{N} f(x, \chi^k)$$

$\chi^1, \ldots, \chi^N$: random sample

Main Result

Under some mild (technical) assumptions:

$\forall \epsilon > 0$:

$\mathbb{P}\{\mathrm{dist}_A(\hat{x}_N) \leq \epsilon\}$ increases exponentially with N

Linköping University

Outline

Basic Idea

Basic Idea

- Basically: Gradient method

Basic Idea

- Basically: Gradient method
- At each iteration: Sample random parameters

Basic Idea

- Basically: Gradient method
- At each iteration: Sample random parameters
- Compute new solution based on this sample

Linköping University

Basic Idea

- Basically: Gradient method
- At each iteration: Sample random parameters
- Compute new solution based on this sample
- Use gradient of function **inside** expectation

Basic Idea

- Basically: Gradient method
- At each iteration: Sample random parameters
- Compute new solution based on this sample
- Use gradient of function **inside** expectation
- Hopefully:

Basic Idea

- Basically: Gradient method
- At each iteration: Sample random parameters
- Compute new solution based on this sample
- Use gradient of function **inside** expectation
- Hopefully:

1) $x^k \to x^*$ as $k \to \infty$ w.h.p.

Linköping University

Problem type (mostly)

Problem type (mostly)

$$\begin{aligned} \min \quad & \mathbb{E}[f(x,\chi)] \\ \text{s.t.} \quad & x \in X \end{aligned}$$

Problem type (mostly)

$$\begin{aligned} \min \quad & \mathbb{E}[f(x,\chi)] \\ \text{s.t.} \quad & x \in X \end{aligned}$$

Problem type (mostly)

$$\begin{aligned} \min \quad & \mathbb{E}[f(x,\chi)] \\ \text{s.t.} \quad & x \in X \end{aligned}$$

Where:

Linköping University

Problem type (mostly)

$$\begin{aligned} \min \quad & \mathbb{E}[f(x,\chi)] \\ \text{s.t.} \quad & x \in X \end{aligned}$$

Where:

- X independent of distribution of random vector

Linköping University

Problem type (mostly)

$$\begin{array}{ll} \min & \mathbb{E}[f(x,\chi)] \\ \text{s.t.} & x \in X \end{array}$$

Where:

- X independent of distribution of random vector
- X convex set

Linköping University

Problem type (mostly)

$$\begin{aligned} \min \quad & \mathbb{E}[f(x,\chi)] \\ \text{s.t.} \quad & x \in X \end{aligned}$$

Where:

- X independent of distribution of random vector
- X convex set
- $f(\cdot,\chi)$ convex

Problem type (mostly)

$$\begin{aligned} \min \quad & \mathbb{E}[f(x,\chi)] \\ \text{s.t.} \quad & x \in X \end{aligned}$$

Where:

- X independent of distribution of random vector
- X convex set
- $f(\cdot,\chi)$ convex
- $f(\cdot,\chi)$ differentiable (nearly everywhere)

Stochastic Gradient Algorithm

Stochastic Gradient Algorithm

$k \leftarrow 0$
Choose x^0 in X

Linköping University

Stochastic Gradient Algorithm

$k \leftarrow 0$
Choose x^0 in X
while $k < K_{max}$ do
  $k \leftarrow k + 1$

end while

Stochastic Gradient Algorithm

$k \leftarrow 0$
Choose x^0 in X
while $k < K_{max}$ do
  $k \leftarrow k + 1$
  Draw $\chi^k = (\chi_1^k, ..., \chi_n^k)$

end while

Stochastic Gradient Algorithm

```
k ← 0
Choose x^0 in X
while k < K_max do
   k ← k + 1
   Draw χ^k = (χ_1^k, ..., χ_n^k)
   Update x^k as follows:
```

$$x^{k+1} \leftarrow x^k + \epsilon^k r^k$$

```
end while
```

- $r^k = \nabla_x f(x, \chi^k)$

Stochastic Gradient Algorithm

$k \leftarrow 0$
Choose x^0 in X
while $k < K_{max}$ do
  $k \leftarrow k + 1$
  Draw $\chi^k = (\chi_1^k, ..., \chi_n^k)$
  Update x^k as follows:

$$x^{k+1} \leftarrow x^k + \epsilon^k r^k$$

end while

- $r^k = \nabla_x f(x, \chi^k)$
- $(\epsilon^k)_{k \in \mathbb{N}}$ is a σ-sequence

Stochastic Gradient Algorithm

$k \leftarrow 0$
Choose x^0 in X
while $k < K_{max}$ do
  $k \leftarrow k + 1$
  Draw $\chi^k = (\chi^k_1, ..., \chi^k_n)$
  Update x^k as follows:

$$x^{k+1} \leftarrow x^k + \epsilon^k r^k$$

end while

Linköping University

- $r^k = \nabla_x f(x, \chi^k)$
- $(\epsilon^k)_{k \in \mathbb{N}}$ is a σ-sequence

Stochastic Gradient Algorithm

$k \leftarrow 0$
Choose x^0 in X
while $k < K_{max}$ do
  $k \leftarrow k + 1$
  Draw $\chi^k = (\chi_1^k, ..., \chi_n^k)$
  Update x^k as follows:

$$x^{k+1} \leftarrow x^k + \epsilon^k r^k$$

  Project x^{k+1} on X
end while

- $r^k = \nabla_x f(x, \chi^k)$
- $(\epsilon^k)_{k \in \mathbb{N}}$ is a σ-sequence

Stochastic Gradient Algorithm

$k \leftarrow 0$
Choose x^0 in X
while $k < K_{max}$ do
  $k \leftarrow k + 1$
  Draw $\chi^k = (\chi_1^k, ..., \chi_n^k)$
  Update x^k as follows:

$$x^{k+1} \leftarrow x^k + \epsilon^k r^k$$

  Project x^{k+1} on X
end while
return Best found or last solution

Linköping University

Assumptions

Assumptions

1) $\forall x \in X$: $f(x, \chi)$ is a random variable with finite expectation

Assumptions

1) $\forall x \in X$: $f(x, \chi)$ is a random variable with finite expectation
2) $\forall \chi \in \Omega$: $f(\cdot, \chi)$ is convex, proper, differentiable

Assumptions

1) $\forall x \in X$: $f(x, \chi)$ is a random variable with finite expectation
2) $\forall \chi \in \Omega$: $f(\cdot, \chi)$ is convex, proper, differentiable
3) $\exists m > 0$ s.t.

$$\forall x \in X, \forall \chi \in \Omega : \|\nabla_x f(x, \chi)\| \leq m$$

Assumptions

1) $\forall x \in X$: $f(x, \chi)$ is a random variable with finite expectation
2) $\forall \chi \in \Omega$: $f(\cdot, \chi)$ is convex, proper, differentiable
3) $\exists m > 0$ s.t.
$$\forall x \in X, \forall \chi \in \Omega : \|\nabla_x f(x, \chi)\| \leq m$$
4) $\exists$ set of optimal solutions X^* and $c > 0$ s.t.:
$$\forall x \in X, x^* \in X^* : \quad \mathbb{E}[f(x, \chi)] - \mathbb{E}[f(x^*, \chi)] \geq c \cdot (\mathrm{dist}_{X^*}(x))^2$$

Assumptions

1) $\forall x \in X$: $f(x, \chi)$ is a random variable with finite expectation
2) $\forall \chi \in \Omega$: $f(\cdot, \chi)$ is convex, proper, differentiable
3) $\exists m > 0$ s.t.

$$\forall x \in X, \forall \chi \in \Omega : \|\nabla_x f(x, \chi)\| \leq m$$

4) $\exists$ set of optimal solutions X^* and $c > 0$ s.t.:

$$\forall x \in X, x^* \in X^* : \quad \mathbb{E}[f(x, \chi)] - \mathbb{E}[f(x^*, \chi)] \geq c \cdot (\text{dist}_{X^*}(x))^2$$

Theorem

Under assumptions 1) - 4) we have:

$$\lim_{k \to \infty} \mathbb{E}\left[(dist_{X^*}(x^k))^2\right] = 0$$

Linköping University

Assumptions

1) $\forall x \in X$: $f(x, \chi)$ is a random variable with finite expectation
2) $\forall \chi \in \Omega$: $f(\cdot, \chi)$ is convex, proper, differentiable
3) $\exists m > 0$ s.t.

$$\forall x \in X, \forall \chi \in \Omega : \|\nabla_x f(x, \chi)\| \leq m$$

4) $\exists$ set of optimal solutions X^* and $c > 0$ s.t.:

$$\forall x \in X, x^* \in X^* : \quad \mathbb{E}[f(x, \chi)] - \mathbb{E}[f(x^*, \chi)] \geq c \cdot (\text{dist}_{X^*}(x))^2$$

Assumptions

1) $\forall x \in X$: $f(x, \chi)$ is a random variable with finite expectation
2) $\forall \chi \in \Omega$: $f(\cdot, \chi)$ is convex, proper, differentiable
3) $\exists m > 0$ s.t.

$$\forall x \in X, \forall \chi \in \Omega : \|\nabla_x f(x, \chi)\| \leq m$$

4) $\exists$ set of optimal solutions X^* and $c > 0$ s.t.:

$$\forall x \in X, x^* \in X^* : \quad \mathbb{E}[f(x, \chi)] - \mathbb{E}[f(x^*, \chi)] \geq c \cdot (\text{dist}_{X^*}(x))^2$$

Theorem

Let $d^0 = (dist_{X^*}(x^0))^2$ *and* $\epsilon^k = \frac{1}{ck + \frac{m^2}{cd^0}}$.

- $r^k = \nabla_x f(x, \chi^k)$
- $(\epsilon^k)_{k\in\mathbb{N}}$ is a σ-sequence

Stochastic Gradient Algorithm

```
k ← 0
Choose x^0 in X
while k < K_max do
  k ← k + 1
  Draw χ^k = (χ^k_1, ..., χ^k_n)
  Update x^k as follows:
```

$$x^{k+1} \leftarrow x^k + \epsilon^k r^k$$

```
  Project x^{k+1} on X
end while
return  Best found or last solution
```

Linköping University

Assumptions

1) $\forall x \in X$: $f(x, \chi)$ is a random variable with finite expectation
2) $\forall \chi \in \Omega$: $f(\cdot, \chi)$ is convex, proper, differentiable
3) $\exists m > 0$ s.t.
$$\forall x \in X, \forall \chi \in \Omega : \|\nabla_x f(x, \chi)\| \leq m$$
4) $\exists$ set of optimal solutions X^* and $c > 0$ s.t.:
$$\forall x \in X, x^* \in X^* : \quad \mathbb{E}[f(x, \chi)] - \mathbb{E}[f(x^*, \chi)] \geq c \cdot (\text{dist}_{X^*}(x))^2$$

Theorem

Let $d^0 = (dist_{X^*}(x^0))^2$ *and* $\epsilon^k = \frac{1}{ck + \frac{m^2}{cd^0}}$.

ersity

Assumptions

1) $\forall x \in X$: $f(x,\chi)$ is a random variable with finite expectation
2) $\forall \chi \in \Omega$: $f(\cdot,\chi)$ is convex, proper, differentiable
3) $\exists m > 0$ s.t.
$$\forall x \in X, \forall \chi \in \Omega : \|\nabla_x f(x,\chi)\| \leq m$$
4) $\exists$ set of optimal solutions X^* and $c > 0$ s.t.:
$$\forall x \in X, x^* \in X^* : \quad \mathbb{E}[f(x,\chi)] - \mathbb{E}[f(x^*,\chi)] \geq c \cdot (\text{dist}_{X^*}(x))^2$$

Theorem

Let $d^0 = (dist_{X^*}(x^0))^2$ *and* $\epsilon^k = \frac{1}{ck + \frac{m^2}{cd^0}}$.
Under assumptions 1) - 4) we have:
$$\mathbb{E}\left[(dist_{X^*}(x^k))^2\right] \leq \frac{1}{\frac{c^2}{m^2}k + \frac{1}{d^0}}$$

Outline

Outline

Linköping University

The beginning

George Dantzig
Linear programming under uncertainty. (1955)
Management Science 1:197–206

Linköping University

The beginning

George Dantzig
Linear programming under uncertainty. (1955)
Management Science 1:197–206

- Two-Stage and Simple recourse problems

The beginning

George Dantzig
Linear programming under uncertainty. (1955)
Management Science 1:197–206

- Two-Stage and Simple recourse problems
- Finite number of scenarios

Linköping University

The beginning

George Dantzig
Linear programming under uncertainty. (1955)
Management Science 1:197–206

- Two-Stage and Simple recourse problems
- Finite number of scenarios
- Deterministic Reformulation

The beginning

George Dantzig
Linear programming under uncertainty. (1955)
Management Science 1:197–206

- Two-Stage and Simple recourse problems
- Finite number of scenarios
- Deterministic Reformulation
- No use of special structure

George Dantzig and Philip Wolfe
Decomposition principle for linear programs.
Operations Research 8(1):101–111, 1960

George Dantzig and Philip Wolfe
Decomposition principle for linear programs.
Operations Research 8(1):101–111, 1960

- Solution method for linear problems of special structure

George Dantzig and Philip Wolfe
Decomposition principle for linear programs.
Operations Research 8(1):101–111, 1960

George Dantzig and Albert Madansky
On the solution of two-stage linear programs under uncertainty.
Proc. 4th Berkeley Symposium on Mathematical Statistics and Probability 1:165–176, 1961

- Solution method for linear problems of special structure

George Dantzig and Philip Wolfe
Decomposition principle for linear programs.
Operations Research 8(1):101–111, 1960

George Dantzig and Albert Madansky
On the solution of two-stage linear programs under uncertainty.
Proc. 4th Berkeley Symposium on Mathematical Statistics and Probability 1:165–176, 1961

- Solution method for linear problems of special structure
- Applicable to dual of Linear Two-Stage Problem with fixed recourse

Theorem

Given: linear two-stage problem
Let:

- *fixed recourse*
- $\{\lambda \in \mathbb{R}^{m_2} : d \geq \lambda W\} \neq \emptyset$
  ($\rightarrow$ second-stage problem primal and dual feasible)

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

Richard Van Slyke and Roger J-B. Wets
L-shaped linear programs with applications to optimal control and stochastic programming.
MSIAM Journal on Applied Mathematics 17(4):638–663, 1969

Linköping University

Richard Van Slyke and Roger J-B. Wets
L-shaped linear programs with applications to optimal control and stochastic programming.
MSIAM Journal on Applied Mathematics 17(4):638–663, 1969

- Solution method that makes use of special problem structures

Richard Van Slyke and Roger J-B. Wets
L-shaped linear programs with applications to optimal control and stochastic programming.
MSIAM Journal on Applied Mathematics 17(4):638–663, 1969

- Solution method that makes use of special problem structures
- Reduced computing time

Richard Van Slyke and Roger J-B. Wets
L-shaped linear programs with applications to optimal control and stochastic programming.
MSIAM Journal on Applied Mathematics 17(4):638–663, 1969

- Solution method that makes use of special problem structures
- Reduced computing time
- Restriction to problems with random right hand side

Richard Van Slyke and Roger J-B. Wets
L-shaped linear programs with applications to optimal control and stochastic programming.
MSIAM Journal on Applied Mathematics 17(4):638–663, 1969

- Solution method that makes use of special problem structures
- Reduced computing time
- Restriction to problems with random right hand side
- Extension to fixed recourse possible

John R. Birge and Franois V. Louveaux
A multicut algorithm for two-stage stochastic linear programs.
European Journal of Operational Research 34(3):384–392, 1988

John R. Birge and Franois V. Louveaux
A multicut algorithm for two-stage stochastic linear programs.
European Journal of Operational Research 34(3):384–392, 1988

- Modification of L-shaped method

John R. Birge and Franois V. Louveaux
A multicut algorithm for two-stage stochastic linear programs.
European Journal of Operational Research 34(3):384–392, 1988

- Modification of L-shaped method
- Add several cuts per iteration

John R. Birge and Franois V. Louveaux
A multicut algorithm for two-stage stochastic linear programs.
European Journal of Operational Research 34(3):384–392, 1988

- Modification of L-shaped method
- Add several cuts per iteration
- Advantage: Less iterations needed

John R. Birge and Franois V. Louveaux
A multicut algorithm for two-stage stochastic linear programs.
European Journal of Operational Research 34(3):384–392, 1988

- Modification of L-shaped method
- Add several cuts per iteration
- Advantage: Less iterations needed
- Disadvantage: Size of subproblems might slow algorithm down

Andrzej Ruszczyński
Regularized decomposition of stochastic programs: Algorithmic techniques and numerical results.
Technical Report WP-93-21, International Institute for Applied Systems Analysis (IIASA), Austria 1993

Andrzej Ruszczyński
Regularized decomposition of stochastic programs: Algorithmic techniques and numerical results.
Technical Report WP-93-21, International Institute for Applied Systems Analysis (IIASA), Austria 1993

- Aim: Reduce size of master problem

Andrzej Ruszczyński
Regularized decomposition of stochastic programs: Algorithmic techniques and numerical results.
Technical Report WP-93-21, International Institute for Applied Systems Analysis (IIASA), Austria 1993

- Aim: Reduce size of master problem
- Idea: Regularize step size

Andrzej Ruszczyński
Regularized decomposition of stochastic programs: Algorithmic techniques and numerical results.
Technical Report WP-93-21, International Institute for Applied Systems Analysis (IIASA), Austria 1993

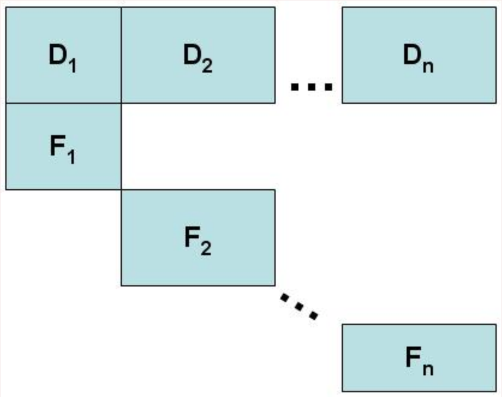
- Aim: Reduce size of master problem
- Idea: Regularize step size
- Advantage: Reduced number of iterations due to stabilizing effect

Andrzej Ruszczyński
Regularized decomposition of stochastic programs: Algorithmic techniques and numerical results.
Technical Report WP-93-21, International Institute for Applied Systems Analysis (IIASA), Austria 1993

- Aim: Reduce size of master problem
- Idea: Regularize step size
- Advantage: Reduced number of iterations due to stabilizing effect
- Result: At most $n_1 + 2K$ cuts need to be stored

Outline

Constraint Matrix type:



Linköping University

Dantzig-Wolfe Decomposition and Stochastic Programming

Dantzig-Wolfe Decomposition and Stochastic Programming

Dantzig-Wolfe Decomposition method
to solve dual of
Linear Two-Stage problems with fixed recourse.

Outline

Linköping University

Linear Two-Stage Problem with fixed recourse

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

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

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

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

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

Linear Two-Stage Problem with fixed recourse

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

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

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

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

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

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

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

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

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

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

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

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

Linear Two-Stage Problem with fixed recourse

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

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

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

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

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

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

Linköping University

Idea

Idea

Start by solving:

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

Idea

Start by solving:

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

Then:

Idea

Start by solving:

$$\min_{x \geq 0} \quad c^T x$$
$$\text{s.t.} \quad Ax \geq b.$$

Then:

- Iteratively add valid cuts (constraints) to the problem

Idea

Start by solving:

$$\min_{x \geq 0} \quad c^T x$$
$$\text{s.t.} \quad Ax \geq b.$$

Then:

- Iteratively add valid cuts (constraints) to the problem

Ensure that optimal solution of *master problem*...

Idea

Start by solving:

$$\min_{x \geq 0} \quad c^T x$$
$$\text{s.t.} \quad Ax \geq b.$$

Then:

- Iteratively add valid cuts (constraints) to the problem

Ensure that optimal solution of *master problem*...

- ..."renders" 2.-s. problem feasible (2.-s. feasible) .

Idea

Start by solving:

$$\min_{x\geq 0} \quad c^T x$$
$$\text{s.t.} \quad Ax \geq b.$$

Then:

- Iteratively add valid cuts (constraints) to the problem

Ensure that optimal solution of *master problem*...

- ..."renders" 2.-s. problem feasible (2.-s. feasible) .
- ...is optimal for original problem .

Linköping University

Idea

Start by solving:

$$\min_{x \geq 0} \quad c^T x$$
$$\text{s.t.} \quad Ax \geq b.$$

Then:

- Iteratively add valid cuts (constraints) to the problem

Ensure that optimal solution of *master problem*...

- ..."renders" 2.-s. problem feasible (2.-s. feasible) $\rightarrow$ Feasibility cut.
- ...is optimal for original problem .

Idea

Start by solving:

$$\min_{x \geq 0} \quad c^T x$$
$$\text{s.t.} \quad Ax \geq b.$$

Then:

- Iteratively add valid cuts (constraints) to the problem

Ensure that optimal solution of *master problem*...

- ..."renders" 2.-s. problem feasible (2.-s. feasible) $\rightarrow$ Feasibility cut.
- ...is optimal for original problem $\rightarrow$ Optimality cut.

Linköping University

Basic Structure of L-shaped method

Basic Structure of L-shaped method

1) Solve current master problem

Current master problem

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

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

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

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)
$\chi^1, \ldots, \chi^K \in \mathbb{R}^s$: scenarios
$\mathbb{P}\{\chi = \chi^k\} := p^k$: probabilities

Linköping University

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

Use: Dual of second-stage problem

Linköping University

Optimality cut

Idea

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

Use: Dual of second-stage problem

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

Optimality cut

Idea

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

Use: Dual of second-stage problem

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

Optimality cut II

Optimality cut II

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

Linköping University

Optimality cut II

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

Linköping University

Optimality cut II

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

Linköping University

Optimality cut II

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

Theory

Linköping University

Optimality cut II

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

Theory

By Duality:

$$\mathcal{Q}(x^\nu, \chi^k) = (\pi_k^\nu)^T (h(\chi^k) - T(\chi^k) x^\nu)$$

Linköping University

Optimality cut II

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

Theory

By Duality:

$$\mathcal{Q}(x^\nu, \chi^k) = (\pi_k^\nu)^T (h(\chi^k) - T(\chi^k) x^\nu)$$

$\forall$ 2.-s. feasible x $\exists \pi_k$ s.t.:

$$\mathcal{Q}(x, \chi^k) = (\pi_k)^T (h(\chi^k) - T(\chi^k) x)$$

Linköping University

Optimality cut II

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

Theory

By Duality:

$$\mathcal{Q}(x^\nu, \chi^k) = (\pi_k^\nu)^T (h(\chi^k) - T(\chi^k) x^\nu)$$

$\forall$ 2.-s. feasible x $\exists \pi_k$ s.t.:

$$\mathcal{Q}(x, \chi^k) = (\pi_k)^T (h(\chi^k) - T(\chi^k) x)$$

By optimality of π_k:

$$(\pi_k)^T (h(\chi^k) - T(\chi^k) x) \geq (\pi_k^\nu)^T (h(\chi^k) - T(\chi^k) x)$$

Linköping University

Optimality cut II

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

Theory

By Duality:

$$\mathcal{Q}(x^\nu, \chi^k) = (\pi_k^\nu)^T (h(\chi^k) - T(\chi^k) x^\nu)$$

$\forall$ 2.-s. feasible x $\exists \pi_k$ s.t.:

$$\mathcal{Q}(x, \chi^k) = (\pi_k)^T (h(\chi^k) - T(\chi^k) x)$$

By optimality of π_k[a]:

$$(\pi_k)^T (h(\chi^k) - T(\chi^k) x) \geq (\pi_k^\nu)^T (h(\chi^k) - T(\chi^k) x)$$

[a] Used: Feasible set of 2.-s. dual independent of x

Optimality cut II

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

Theory (continued)

Linköping University

Optimality cut II

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

Theory (continued)

$\Longrightarrow \forall$ 2.-s. feasible x:

Optimality cut II

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

Theory (continued)

$\Longrightarrow \forall$ 2.-s. feasible x:

$$\mathcal{Q}(x, \chi^k) \geq (\pi_k^\nu)^T (h(\chi^k) - T(\chi^k)x)$$

Linköping University

Optimality cut II

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

Theory (continued)

$\Longrightarrow \forall$ 2.-s. feasible x:

$$\mathcal{Q}(x, \chi^k) \geq (\pi_k^\nu)^T (h(\chi^k) - T(\chi^k)x)$$

$\Longrightarrow \forall$ 2.-s. feasible x:

Linköping University

Optimality cut II

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

Theory (continued)

$\Longrightarrow \forall$ 2.-s. feasible x:

$$\mathcal{Q}(x, \chi^k) \geq (\pi_k^\nu)^T (h(\chi^k) - T(\chi^k)x)$$

$\Longrightarrow \forall$ 2.-s. feasible x:

$$\mathbb{E}[\mathcal{Q}(x, \chi)] = \sum_{k=1}^{K} p^k \mathcal{Q}(x, \chi^k) \geq \sum_{k=1}^{K} p^k (\pi_k^\nu)^T (h(\chi^k) - T(\chi^k)x)$$

Optimality cut III

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

Linköping University

Optimality cut III

- 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

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

Linköping University

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:

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

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 $z_k = 0$:

x^ν is 2.-s. feasible

Linköping University

Feasibility cut

First:

Test feasibility of optimal solution of master problem by computing:

$$\begin{aligned} z_k = \min \quad & \mathbb{1}^T v_k^+ \\ \text{s.t.} \quad & W v_k + v_k^+ \geq h(\chi^k) - T(\chi^k) x^\nu, \\ & v_k, v_k^+ \geq 0. \end{aligned} \tag{6a}$$

Feasibility cut

First:

Test feasibility of optimal solution of master problem by computing:

$$\begin{aligned} z_k = \min \quad & \mathbb{1}^T v_k^+ \\ \text{s.t.} \quad & W v_k + v_k^+ \geq h(\chi^k) - T(\chi^k) x^\nu, \\ & v_k, v_k^+ \geq 0. \end{aligned} \tag{6a}$$

If $z_k > 0$:

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

Feasibility cut II

Theory

Consider dual:

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

Feasibility cut II

Theory

Consider dual:

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

- σ_k^ν: Optimal solution of above dual problem

Feasibility cut II

Theory

Consider dual:

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

- 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

$r, s, \nu \leftarrow 0$

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

$r, s, \nu \leftarrow 0$
`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` $(r \leftarrow r + 1)$

  `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, 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



  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 ($s \leftarrow s + 1$)
  if x^ν, θ^ν satisfy optimality cut then
    STOP. x^ν is optimal solution.

  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.
  else
    Go back:  Resolve CMP
  end if
end while
```

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

Linköping University

QUESTIONS?

What about next week?