Introduction to robust optimization

Michael POSS

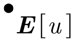
May 30, 2017

Outline

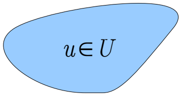
Robust optimization

1. **How much do we know ?**

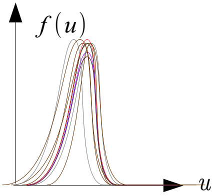
Mean value (Deterministic)



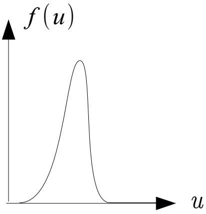
Robust



Distributionally robust



Stochastic



Robust optimization

2. **Worst-case approach**

static VS adjustable

Static decisions $\dashrightarrow$ uncertainty revealed

Complexity Easy for LP ☺, $\mathcal{NP}$-hard for combinatorial optimization ☹

MILP reformulation ☺

Two-stages decisions $\dashrightarrow$ uncertainty revealed $\dashrightarrow$ more decisions

Complexity $\mathcal{NP}$-hard for LP ☹, decomposition algorithms ☺

Multi-stages decisions $\dashrightarrow$ uncertainty $\dashrightarrow$ decisions $\dashrightarrow$ uncertainty $\dashrightarrow$ $\cdots$

Complexity $\mathcal{NP}$-hard for LP ☹, cannot be solved to optimality ☹

static VS adjustable

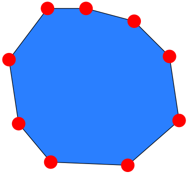
Static	decisions ⇢ uncertainty revealed
Complexity	Easy for LP ☺, $\mathcal{NP}$-hard for combinatorial optimization ☹
	MILP reformulation ☺
Two-stages	decisions ⇢ uncertainty revealed ⇢ more decisions
Complexity	$\mathcal{NP}$-hard for LP ☹, decomposition algorithms ☺
Multi-stages	decisions ⇢ uncertainty ⇢ decisions ⇢ uncertainty ⇢ $\cdots$
Complexity	$\mathcal{NP}$-hard for LP ☹, cannot be solved to optimality ☹

static VS adjustable

Static	decisions ⇢ uncertainty revealed
Complexity	Easy for LP ☺, $\mathcal{NP}$-hard for combinatorial optimization ☹
	MILP reformulation ☺
Two-stages	decisions ⇢ uncertainty revealed ⇢ more decisions
Complexity	$\mathcal{NP}$-hard for LP ☹, decomposition algorithms ☺
Multi-stages	decisions ⇢ uncertainty ⇢ decisions ⇢ uncertainty ⇢ $\cdots$
Complexity	$\mathcal{NP}$-hard for LP ☹, cannot be solved to optimality ☹

discrete uncertainty VS convex uncertainty

$\mathcal{U} = \text{vertices}(\mathcal{P})$



Observation

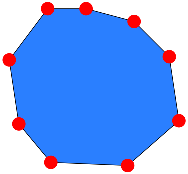
In many cases, $\mathcal{U} \sim \mathcal{P}$.

Exceptions:

- robust constraints $f(x,u) \leq b$ and f non-concave in u
- multi-stages problems with integer adjustable variables

discrete uncertainty VS convex uncertainty

$\mathcal{U} = \text{vertices}(\mathcal{P})$



Observation

In many cases, $\mathcal{U} \sim \mathcal{P}$.

Exceptions:

- robust constraints $f(x, u) \leq b$ and f non-concave in u
- multi-stages problems with integer adjustable variables

Outline

Robust combinatorial optimization

Combinatorial problem

- $\mathcal{X} \subseteq \{0,1\}^n$, $u_0 \in \mathbb{R}^n$

$$CO \qquad \min_{x \in \mathcal{X}} u_0^T x.$$

Robust counterparts with cost uncertainty

1. $\mathcal{X} \subseteq \{0,1\}^n, \mathcal{U} \subset \mathbb{R}^n$

$$\mathcal{U}\text{-}CO \qquad \min_{x \in \mathcal{X}} \max_{u \in \mathcal{U}} u_0^T x$$

2. Regret version:

$$\min_{x \in \mathcal{X}} \max_{u \in \mathcal{U}} \left(u_0^T x - \min_{y \in \mathcal{X}} u_0^T y \right)$$

Robust combinatorial optimization

Combinatorial problem

- $\mathcal{X} \subseteq \{0,1\}^n$, $u_0 \in \mathbb{R}^n$

$$CO \qquad \min_{x \in \mathcal{X}} u_0^T x.$$

Robust counterparts with cost uncertainty

1. $\mathcal{X} \subseteq \{0,1\}^n, \mathcal{U} \subset \mathbb{R}^n$

$$\mathcal{U}\text{-}CO \qquad \min_{x \in \mathcal{X}} \max_{u \in \mathcal{U}} u_0^T x$$

2. Regret version:

$$\min_{x \in \mathcal{X}} \max_{u \in \mathcal{U}} \left(u_0^T x - \min_{y \in \mathcal{X}} u_0^T y \right)$$

$$= \min_{x \in \mathcal{X}} \max_{u \in \mathcal{U}} \min_{y \in \mathcal{X}} \left(u_0^T x - u_0^T y \right)$$

Robust combinatorial optimization

Combinatorial problem

- $\mathcal{X} \subseteq \{0,1\}^n$, $u_0 \in \mathbb{R}^n$

$$CO \qquad \min_{x \in \mathcal{X}} u_0^T x.$$

Robust counterparts with cost uncertainty

1. $\mathcal{X} \subseteq \{0,1\}^n, \mathcal{U} \subset \mathbb{R}^n$

$$\mathcal{U}\text{-}CO \qquad \min_{x \in \mathcal{X}} \max_{u \in \mathcal{U}} u_0^T x$$

2. Regret version:

$$\min_{x \in \mathcal{X}} \max_{u \in \mathcal{U}} \left(u_0^T x - \min_{y \in \mathcal{X}} u_0^T y \right)$$

$$= \qquad \min_{x \in \mathcal{X}} \max_{u \in \mathcal{U}} \min_{y \in \mathcal{X}} \left(u_0^T x - u_0^T y \right)$$

Robust combinatorial optimization

Combinatorial problem

- $\mathcal{X} \subseteq \{0,1\}^n$, $u_0 \in \mathbb{R}^n$

$$CO \qquad \min_{x \in \mathcal{X}} u_0^T x.$$

Robust counterparts with cost uncertainty

1. $\mathcal{X} \subseteq \{0,1\}^n, \mathcal{U} \subset \mathbb{R}^n$

$$\mathcal{U}\text{-}CO \qquad \min_{x \in \mathcal{X}} \max_{u \in \mathcal{U}} u_0^T x$$

2. Regret version:

$$\min_{x \in \mathcal{X}} \max_{u \in \mathcal{U}} \left(u_0^T x - \min_{y \in \mathcal{X}} u_0^T y \right)$$

$$= \quad \min_{x \in \mathcal{X}} \max_{u \in \mathcal{U}} \min_{y \in \mathcal{X}} \left(u_0^T x - u_0^T y \right)$$

General robust counterpart

$\mathcal{X} = \mathcal{X}^{comb} \cap \mathcal{X}^{num}$:

$\mathcal{X}^{comb}$ Combinatorial nature, **known**.

$\mathcal{X}^{num}$ Numerical uncertainty: $u_j^T x \leq b_j, \; j = 1, \ldots, m$, **uncertain**.

Robust counterpart

$$\min \left\{ \quad : \right. \qquad (1)$$

$$\mathcal{U}\text{-}CO \qquad u_j^T x \leq b_j, \quad j = 1, \ldots, m, \; u_j \in \mathcal{U}_j, \qquad (2)$$

$$(3)$$

Examples: knapsack, constrained shortest path

General robust counterpart

$\mathcal{X} = \mathcal{X}^{comb} \cap \mathcal{X}^{num}$:

$\mathcal{X}^{comb}$ Combinatorial nature, **known**.

$\mathcal{X}^{num}$ Numerical uncertainty: $u_j^T x \leq b_j$, $j = 1, \dots, m$, **uncertain**.

Robust counterpart

$$\mathcal{U}\text{-}CO \qquad \min \left\{ \begin{array}{ll} \max_{u_0 \in \mathcal{U}_0} u_0^T x & : \qquad (1) \\ u_j^T x \leq b_j, \quad j = 1, \dots, m, \ u_j \in \mathcal{U}_j, & \qquad (2) \\ x \in \mathcal{X}^{comb} & \end{array} \right\}.$$

Examples: knapsack, constrained shortest path

General robust counterpart

$\mathcal{X} = \mathcal{X}^{comb} \cap \mathcal{X}^{num}$:

$\mathcal{X}^{comb}$ Combinatorial nature, **known**.

$\mathcal{X}^{num}$ Numerical uncertainty: $u_j^T x \leq b_j,\ j = 1, \dots, m$, **uncertain**.

Robust counterpart

$$\mathcal{U}\text{-}CO \qquad \min \left\{ \max_{u_0 \in \mathcal{U}_0} u_0^T x \ : \right. \tag{1}$$

$$u_j^T x \leq b_j, \quad j = 1, \dots, m,\ u_j \in \mathcal{U}_j, \tag{2}$$

$$a_k^T x \leq d_k, \quad k = 1, \dots, \ell \tag{3}$$

$$\left. x \in \{0,1\}^n \quad \right\} \tag{4}$$

Examples: knapsack, constrained shortest path

General robust counterpart

$\mathcal{X} = \mathcal{X}^{comb} \cap \mathcal{X}^{num}$:

$\mathcal{X}^{comb}$ Combinatorial nature, **known**.

$\mathcal{X}^{num}$ Numerical uncertainty: $u_j^T x \leq b_j,\ j = 1, \dots, m$, **uncertain**.

Robust counterpart

$\mathcal{U}$-CO

$$\min \Big\{ \ z \ : \tag{1}$$

$$u_j^T x \leq b_j, \quad j = 1, \dots, m, \ u_j \in \mathcal{U}_j, \tag{2}$$

$$u_0^T x \leq z, \quad u_0 \in \mathcal{U}_0 \tag{3}$$

$$a_k^T x \leq d_k, \quad k = 1, \dots, \ell \tag{4}$$

$$x \in \{0, 1\}^n \quad \Big\} \tag{5}$$

Examples: knapsack, constrained shortest path

General robust counterpart

$\mathcal{X} = \mathcal{X}^{comb} \cap \mathcal{X}^{num}$:

$\mathcal{X}^{comb}$ Combinatorial nature, **known**.

$\mathcal{X}^{num}$ Numerical uncertainty: $u_j^T x \leq b_j,\ j = 1, \ldots, m$, **uncertain**.

Robust counterpart

$\mathcal{U}$-CO

$$\min \left\{ \quad : \right. \tag{1}$$

$$u_j^T x \leq b_j, \quad j = 1, \ldots, m,\ u_j \in \mathcal{U}_j, \tag{2}$$

$$u_0^T x \leq z, \quad u_0 \in \mathcal{U}_0 \tag{3}$$

$$a_k^T x \leq d_k, \quad k = 1, \ldots, \ell \tag{4}$$

$$\left. x \in \{0, 1\}^n \quad \right\} \tag{5}$$

Examples: knapsack, constrained shortest path

discrete uncertainty: $\mathcal{U}$-CO is hard [Kouvelis and Yu, 2013]

Theorem

The robust shortest path, assignment, spanning tree, ... are $\mathcal{NP}$-hard even when $|\mathcal{U}| = 2$.

Proof.

1. SELECTION PROBLEM: $\min_{S \subseteq N, |S|=p} \sum_{i \in S} u_i$
2. ROBUST SEL. PROB.: $\min_{S \subseteq N, |S|=p} \max_{u \in \mathcal{U}} \sum_{i \in S} u_i$
3. PARTITION PROBLEM: $\min_{S \subseteq N, |S|=|N|/2} \max \left(\sum_{i \in S} a_i, \sum_{i \in N \setminus S} a_i \right)$
4. Reduction: $p = \frac{|N|}{2}$, and $\mathcal{U} = \{u^1, u^2\}$ such that

$$u_i^1 = a_i \quad \text{and} \quad u_i^2 = \frac{2}{|N|} \sum_k a_k - a_i$$

$$\Rightarrow \quad \max_{u \in \mathcal{U}} \sum_{i \in S} u_i = \max \left(\sum_{i \in S} a_i, \sum_{i \in N \setminus S} a_i \right)$$

discrete uncertainty: $\mathcal{U}$-CO is hard [Kouvelis and Yu, 2013]

Theorem

The robust shortest path, assignment, spanning tree, ... are $\mathcal{NP}$-hard even when $|\mathcal{U}| = 2$.

Proof.

1. SELECTION PROBLEM: $\min_{S \subseteq N, |S|=p} \sum_{i \in S} u_i$
2. ROBUST SEL. PROB.: $\min_{S \subseteq N, |S|=p} \max_{u \in \mathcal{U}} \sum_{i \in S} u_i$
3. PARTITION PROBLEM: $\min_{S \subseteq N, |S|=|N|/2} \max\left(\sum_{i \in S} a_i, \sum_{i \in N \setminus S} a_i \right)$
4. Reduction: $p = \frac{|N|}{2}$, and $\mathcal{U} = \{u^1, u^2\}$ such that

$$u_i^1 = a_i \qquad \text{and} \qquad u_i^2 = \frac{2}{|N|} \sum_k a_k - a_i$$

$$\Rightarrow \quad \max_{u \in \mathcal{U}} \sum_{i \in S} u_i = \max\left(\sum_{i \in S} a_i, \sum_{i \in N \setminus S} a_i \right)$$

discrete uncertainty: $\mathcal{U}$-CO is hard [Kouvelis and Yu, 2013]

Theorem

The robust shortest path, assignment, spanning tree, ... are $\mathcal{NP}$-hard even when $|\mathcal{U}| = 2$.

Proof.

1. SELECTION PROBLEM: $\min_{S \subseteq N, |S|=p} \sum_{i \in S} u_i$
2. ROBUST SEL. PROB.: $\min_{S \subseteq N, |S|=p} \max_{u \in \mathcal{U}} \sum_{i \in S} u_i$
3. PARTITION PROBLEM: $\min_{S \subseteq N, |S|=|N|/2} \max \left(\sum_{i \in S} a_i, \sum_{i \in N \setminus S} a_i \right)$
4. Reduction: $p = \frac{|N|}{2}$, and $\mathcal{U} = \{u^1, u^2\}$ such that

$$u_i^1 = a_i \quad \text{and} \quad u_i^2 = \frac{2}{|N|} \sum_k a_k - a_i$$

$$\Rightarrow \quad \max_{u \in \mathcal{U}} \sum_{i \in S} u_i = \max \left(\sum_{i \in S} a_i, \sum_{i \in N \setminus S} a_i \right)$$

discrete uncertainty: $\mathcal{U}$-CO is hard [Kouvelis and Yu, 2013]

Theorem

The robust shortest path, assignment, spanning tree, ... are $\mathcal{NP}$-hard even when $|\mathcal{U}| = 2$.

Proof.

1. SELECTION PROBLEM: $\min_{S\subseteq N,|S|=p} \sum_{i\in S} u_i$
2. ROBUST SEL. PROB.: $\min_{S\subseteq N,|S|=p} \max_{u\in\mathcal{U}} \sum_{i\in S} u_i$
3. PARTITION PROBLEM: $\min_{S\subseteq N,|S|=|N|/2} \max\left(\sum_{i\in S} a_i, \sum_{i\in N\setminus S} a_i\right)$
4. Reduction: $p = \frac{|N|}{2}$, and $\mathcal{U} = \{u^1, u^2\}$ such that

$$u_i^1 = a_i \quad \text{and} \quad u_i^2 = \frac{2}{|N|}\sum_k a_k - a_i$$

$$\Rightarrow \quad \max_{u\in\mathcal{U}} \sum_{i\in S} u_i = \max\left(\sum_{i\in S} a_i, \sum_{i\in N\setminus S} a_i\right)$$

discrete uncertainty: $\mathcal{U}$-CO is hard [Kouvelis and Yu, 2013]

Theorem

The robust shortest path, assignment, spanning tree, ... are $\mathcal{NP}$-hard even when $|\mathcal{U}| = 2$.

Proof.

1. SELECTION PROBLEM: $\min_{S\subseteq N, |S|=p} \sum_{i\in S} u_i$
2. ROBUST SEL. PROB.: $\min_{S\subseteq N, |S|=p} \max_{u\in\mathcal{U}} \sum_{i\in S} u_i$
3. PARTITION PROBLEM: $\min_{S\subseteq N, |S|=|N|/2} \max\left(\sum_{i\in S} a_i, \sum_{i\in N\setminus S} a_i\right)$
4. Reduction: $p = \frac{|N|}{2}$, and $\mathcal{U} = \{u^1, u^2\}$ such that

$$u_i^1 = a_i \qquad \text{and} \qquad u_i^2 = \frac{2}{|N|}\sum_k a_k - a_i$$

$$\Rightarrow \quad \max_{u\in\mathcal{U}} \sum_{i\in S} u_i = \max\left(\sum_{i\in S} a_i, \sum_{i\in N\setminus S} a_i\right)$$

polyhedral uncertainty: $\mathcal{U}$-CO is still hard (but solvable)

Theorem

The robust shortest path, assignment, spanning tree, ... are $\mathcal{NP}$-hard even when $\mathcal{U}$ has a compact description.

Proof.

1. $\mathcal{U} = \text{conv}(u^1, u^2) \Rightarrow n$ equalities and 2 inequalities
2. $u^T x \leq b, \quad u \in \mathcal{U} \quad \Leftrightarrow \quad u^T x \leq b, \quad u \in \text{ext}(\mathcal{U})$

□

Theorem (Ben-Tal and Nemirovski [1998])

Problem $\mathcal{U}$-CO is equivalent to a mixed-integer linear program.

polyhedral uncertainty: $\mathcal{U}$-CO is still hard (but solvable)

Theorem

The robust shortest path, assignment, spanning tree, ... are $\mathcal{NP}$-hard even when $\mathcal{U}$ has a compact description.

Proof.

1. $\mathcal{U} = \text{conv}(u^1, u^2) \Rightarrow n$ equalities and 2 inequalities
2. $u^T x \leq b, \quad u \in \mathcal{U} \quad \Leftrightarrow \quad u^T x \leq b, \quad u \in \text{ext}(\mathcal{U})$

□

Theorem (Ben-Tal and Nemirovski [1998])

Problem $\mathcal{U}$-CO is equivalent to a mixed-integer linear program.

polyhedral uncertainty: $\mathcal{U}$-CO is still hard (but solvable)

Theorem

The robust shortest path, assignment, spanning tree, ... are $\mathcal{NP}$-hard even when $\mathcal{U}$ has a compact description.

Proof.

1. $\mathcal{U} = \text{conv}(u^1, u^2) \Rightarrow n$ equalities and 2 inequalities
2. $u^T x \leq b, \quad u \in \mathcal{U} \quad \Leftrightarrow \quad u^T x \leq b, \quad u \in \text{ext}(\mathcal{U})$

□

Theorem (Ben-Tal and Nemirovski [1998])

Problem $\mathcal{U}$-CO is equivalent to a mixed-integer linear program.

polyhedral uncertainty: $\mathcal{U}$-CO is still hard (but solvable)

Theorem

The robust shortest path, assignment, spanning tree, ... are $\mathcal{NP}$-hard even when $\mathcal{U}$ has a compact description.

Proof.

1. $\mathcal{U} = \text{conv}(u^1, u^2) \Rightarrow n$ equalities and 2 inequalities
2. $u^T x \leq b, \quad u \in \mathcal{U} \quad \Leftrightarrow \quad u^T x \leq b, \quad u \in \text{ext}(\mathcal{U})$

□

Theorem (Ben-Tal and Nemirovski [1998])

Problem $\mathcal{U}$-CO is equivalent to a mixed-integer linear program.

Dualization - cost uncertainty

Theorem (Ben-Tal and Nemirovski [1998])

Consider $\alpha \in \mathbb{R}^{l\times n}$ and $\beta \in \mathbb{R}^{l}$ that define polytope

$$\mathcal{U} := \{u \in \mathbb{R}^n_+ \ : \ \alpha_k^T u \leq \beta_k, \ k = 1, \dots, l\}.$$

Problem $\min\limits_{x\in\mathcal{X}} \max\limits_{u\in\mathcal{U}} u^T x$ is equivalent to a compact MILP.

Proof.

Dualizing the inner maximization: $\min\limits_{x\in\mathcal{X}} \max\limits_{u\in\mathcal{U}} u^T x =$

$$\min_{x\in\mathcal{X}} \min \left\{ \sum_{k=1}^{l} \beta_k z_k \ : \ \sum_{k=1}^{l} \alpha_{ki} z_k \geq x_i, i = 1, \dots, n, z \geq 0 \right\}, \qquad \square$$

Robust constraint (e.g. the knapsack)

Theorem (Ben-Tal and Nemirovski [1998])

Consider $\alpha \in \mathbb{R}^{l \times n}$ and $\beta \in \mathbb{R}^{l}$ that define polytope

$$\mathcal{U} := \{u \in \mathbb{R}^n_+ \ : \ \alpha_k^T u \leq \beta_k, \ k = 1, \ldots, l\}.$$

Problem $\min_{x \in \mathcal{X}} \max_{u \in \mathcal{U}} u^T x$ is equivalent to a compact MILP.

Proof.

Dualizing the inner maximization: $\min_{x \in \mathcal{X}} \max_{u \in \mathcal{U}} u^T x =$

$$\min_{x \in \mathcal{X}} \min \left\{ \sum_{k=1}^{l} \beta_k z_k \ : \ \sum_{k=1}^{l} \alpha_{ki} z_k \geq x_i, i = 1, \ldots, n, z \geq 0 \right\},$$

□

Robust constraint (e.g. the knapsack)

Dualization - cost uncertainty

Theorem (Ben-Tal and Nemirovski [1998])

Consider $\alpha \in \mathbb{R}^{l \times n}$ *and* $\beta \in \mathbb{R}^l$ *that define polytope*

$$\mathcal{U} := \{u \in \mathbb{R}^n_+ \ : \ \alpha_k^T u \le \beta_k, \ k = 1, \ldots, l\}.$$

Problem $\min_{x \in \mathcal{X}} \max_{u \in \mathcal{U}} u^T x$ *is equivalent to a compact MILP.*

Proof.

Dualizing the inner maximization: $\min_{x \in \mathcal{X}} \max_{u \in \mathcal{U}} u^T x =$

$$\min_{x \in \mathcal{X}} \min \left\{ \sum_{k=1}^{l} \beta_k z_k \ : \ \sum_{k=1}^{l} \alpha_{ki} z_k \ge x_i, i = 1, \ldots, n, z \ge 0 \right\},$$

□

Robust constraint (e.g. the knapsack)

Cutting plane algorithms [Bertsimas et al., 2016]

$\mathcal{U}_0^* \subset \mathcal{U}_0$, $\mathcal{U}_j^* \subset \mathcal{U}_j$

Master problem

$$
MP \quad \min \left\{ \begin{array}{ll} z \; : & \\ u_j^T x \le b_j, & j = 1, \ldots, m, \; u_j \in \mathcal{U}_j^*, \\ u_0^T x \le z, & u_0 \in \mathcal{U}_0^*, \\ a_k^T x \le d_k, & k = 1, \ldots, \ell \\ x \in \{0,1\}^n & \end{array} \right\}
$$

1. **Solve** $MP \rightarrow$ get $\tilde{x}, \tilde{z}$
2. **Solve** $\max_{u_0 \in \mathcal{U}_0} u_0^T \tilde{x}$ **and** $\max_{u_j \in \mathcal{U}_j} u_j^T \tilde{x} \rightarrow$ get $\tilde{u}_0, \ldots, \tilde{u}_m$
3. **If** $\tilde{u}_0^T \tilde{x} > \tilde{z}$ **or** $\tilde{u}_j^T \tilde{x} > b_j$ **then**
   - $\mathcal{U}_0^* \leftarrow \mathcal{U}_0^* \cup \{\tilde{u}_0\}$ and $\mathcal{U}_j^* \leftarrow \mathcal{U}_j^* \cup \{\tilde{u}_j\}$
   - go back to 1

Cutting plane algorithms [Bertsimas et al., 2016]

$\mathcal{U}_0^* \subset \mathcal{U}_0$, $\mathcal{U}_j^* \subset \mathcal{U}_j$

Master problem

$$MP \qquad \min \left\{ \begin{array}{ll} z \; : & \\ u_j^T x \le b_j, & j = 1, \dots, m, \; u_j \in \mathcal{U}_j^*, \\ u_0^T x \le z, & u_0 \in \mathcal{U}_0^*, \\ a_k^T x \le d_k, & k = 1, \dots, \ell \\ x \in \{0,1\}^n & \end{array} \right\}$$

1. **Solve** $MP \rightarrow$ get $\tilde{x}, \tilde{z}$
2. **Solve** $\max_{u_0 \in \mathcal{U}_0} u_0^T \tilde{x}$ **and** $\max_{u_j \in \mathcal{U}_j} u_j^T \tilde{x} \rightarrow$ get $\tilde{u}_0, \dots, \tilde{u}_m$
3. **If** $\tilde{u}_0^T \tilde{x} > \tilde{z}$ **or** $\tilde{u}_j^T \tilde{x} > b_j$ **then**
   - $\mathcal{U}_0^* \leftarrow \mathcal{U}_0^* \cup \{\tilde{u}_0\}$ **and** $\mathcal{U}_0^* \leftarrow \mathcal{U}_j^* \cup \{\tilde{u}_j\}$
   - go back to 1

Cutting plane algorithms [Bertsimas et al., 2016]

$\mathcal{U}_0^* \subset \mathcal{U}_0, \mathcal{U}_j^* \subset \mathcal{U}_j$

Master problem

$$
MP \qquad \min \left\{ \begin{array}{ll} z \ : & \\ u_j^T x \le b_j, & j = 1, \ldots, m, \ u_j \in \mathcal{U}_j^*, \\ u_0^T x \le z, & u_0 \in \mathcal{U}_0^*, \\ a_k^T x \le d_k, & k = 1, \ldots, \ell \\ x \in \{0,1\}^n & \end{array} \right\}
$$

1. **Solve** $MP \rightarrow$ get $\tilde{x}, \tilde{z}$
2. **Solve** $\max_{u_0 \in \mathcal{U}_0} u_0^T \tilde{x}$ **and** $\max_{u_j \in \mathcal{U}_j} u_j^T \tilde{x} \rightarrow$ get $\tilde{u}_0, \ldots, \tilde{u}_m$
3. **If** $\tilde{u}_0^T \tilde{x} > \tilde{z}$ **or** $\tilde{u}_j^T \tilde{x} > b_j$ **then**
   - $\mathcal{U}_0^* \leftarrow \mathcal{U}_0^* \cup \{\tilde{u}_0\}$ **and** $\mathcal{U}_0^* \leftarrow \mathcal{U}_j^* \cup \{\tilde{u}_j\}$
   - go back to 1

Cutting plane algorithms [Bertsimas et al., 2016]

$\mathcal{U}_0^* \subset \mathcal{U}_0, \mathcal{U}_j^* \subset \mathcal{U}_j$

Master problem

$$
MP \qquad \min \left\{ \begin{array}{ll} z \ : & \\ u_j^T x \leq b_j, & j = 1, \ldots, m, \ u_j \in \mathcal{U}_j^*, \\ u_0^T x \leq z, & u_0 \in \mathcal{U}_0^*, \\ a_k^T x \leq d_k, & k = 1, \ldots, \ell \\ x \in \{0,1\}^n & \end{array} \right\}
$$

1. **Solve** $MP \to$ get $\tilde{x}, \tilde{z}$
2. **Solve** $\max_{u_0 \in \mathcal{U}_0} u_0^T \tilde{x}$ **and** $\max_{u_j \in \mathcal{U}_j} u_j^T \tilde{x} \to$ get $\tilde{u}_0, \ldots, \tilde{u}_m$
3. **If** $\tilde{u}_0^T \tilde{x} > \tilde{z}$ **or** $\tilde{u}_j^T \tilde{x} > b_j$ **then**
   - $\mathcal{U}_0^* \leftarrow \mathcal{U}_0^* \cup \{\tilde{u}_0\}$ **and** $\mathcal{U}_0^* \leftarrow \mathcal{U}_j^* \cup \{\tilde{u}_j\}$
   - go back to 1

Cutting plane algorithms [Bertsimas et al., 2016]

$\mathcal{U}_0^* \subset \mathcal{U}_0$, $\mathcal{U}_j^* \subset \mathcal{U}_j$

Master problem

$$
MP \qquad \min \left\{ \begin{array}{ll} z : & \\ u_j^T x \le b_j, & j = 1, \dots, m,\ u_j \in \mathcal{U}_j^*, \\ u_0^T x \le z, & u_0 \in \mathcal{U}_0^*, \\ a_k^T x \le d_k, & k = 1, \dots, \ell \\ x \in \{0,1\}^n & \end{array} \right\}
$$

1. **Solve** $MP \rightarrow$ get $\tilde{x}, \tilde{z}$
2. **Solve** $\max_{u_0 \in \mathcal{U}_0} u_0^T \tilde{x}$ **and** $\max_{u_j \in \mathcal{U}_j} u_j^T \tilde{x} \rightarrow$ get $\tilde{u}_0, \dots, \tilde{u}_m$
3. **If** $\tilde{u}_0^T \tilde{x} > \tilde{z}$ **or** $\tilde{u}_j^T \tilde{x} > b_j$ **then**
   - $\mathcal{U}_0^* \leftarrow \mathcal{U}_0^* \cup \{\tilde{u}_0\}$ **and** $\mathcal{U}_0^* \leftarrow \mathcal{U}_j^* \cup \{\tilde{u}_j\}$
   - go back to 1

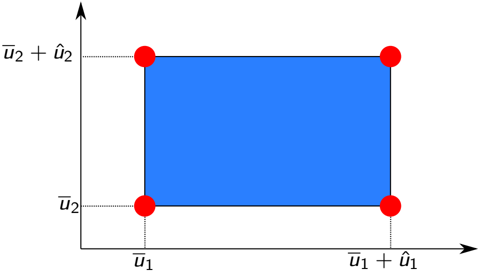
Simpler structure: $\mathcal{U}^\Gamma$-robust combinatorial optimization

- $\mathcal{U}$ = vertices($\mathcal{P}$): good, but need "simpler" $\mathcal{P}$

$$\mathcal{U}^\Gamma = \left\{ \overline{u}_i \le u_i \le \overline{u}_i + \hat{u}_i, i = 1, \dots, n, \sum_{i=1}^{n} \frac{u_i - \overline{u}_i}{\hat{u}_i} \le \right\}$$

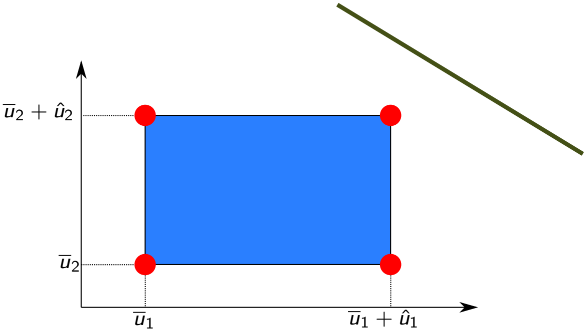
Simpler structure: $\mathcal{U}^\Gamma$-robust combinatorial optimization

- $\mathcal{U}$ = vertices($\mathcal{P}$): good, but need "simpler" $\mathcal{P}$



$$\mathcal{U}^\Gamma = \left\{ \overline{u}_i \leq u_i \leq \overline{u}_i + \hat{u}_i, i = 1, \ldots, n, \sum_{i=1}^{n} \frac{u_i - \overline{u}_i}{\hat{u}_i} \leq \right\}$$

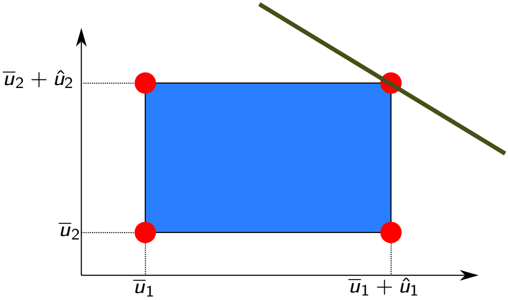
- $\mathcal{U}$ = vertices($\mathcal{P}$): good, but need "simpler" $\mathcal{P}$



$$\mathcal{U}^\Gamma = \left\{ \overline{u}_i \leq u_i \leq \overline{u}_i + \hat{u}_i, i = 1, \dots, n, \sum_{i=1}^{n} \frac{u_i - \overline{u}_i}{\hat{u}_i} \leq \Gamma \right\}$$

Simpler structure: $\mathcal{U}^\Gamma$-robust combinatorial optimization

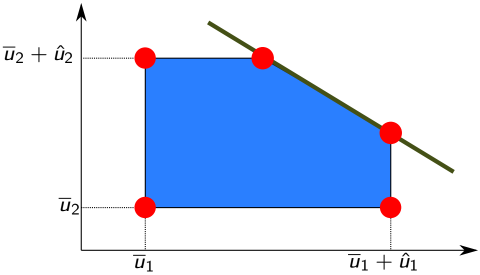
- $\mathcal{U}$ = vertices($\mathcal{P}$): good, but need "simpler" $\mathcal{P}$



$$\mathcal{U}^\Gamma = \left\{ \overline{u}_i \le u_i \le \overline{u}_i + \hat{u}_i, i = 1, \dots, n, \sum_{i=1}^{n} \frac{u_i - \overline{u}_i}{\hat{u}_i} \le 2 \right\}$$

Simpler structure: $\mathcal{U}^\Gamma$-robust combinatorial optimization

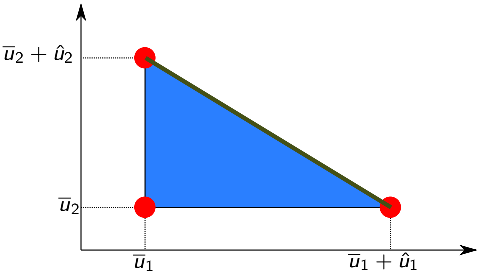
- $\mathcal{U}$ = vertices($\mathcal{P}$): good, but need "simpler" $\mathcal{P}$



$$\mathcal{U}^\Gamma = \left\{ \overline{u}_i \le u_i \le \overline{u}_i + \hat{u}_i, i = 1, \ldots, n, \sum_{i=1}^{n} \frac{u_i - \overline{u}_i}{\hat{u}_i} \le 1.5 \right\}$$

Simpler structure: $\mathcal{U}^\Gamma$-robust combinatorial optimization

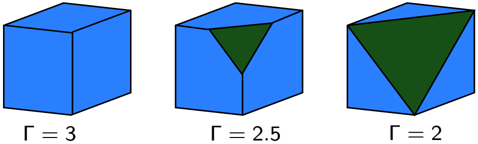
- $\mathcal{U}$ = vertices($\mathcal{P}$): good, but need "simpler" $\mathcal{P}$



$$\mathcal{U}^\Gamma = \left\{ \overline{u}_i \le u_i \le \overline{u}_i + \hat{u}_i, i = 1, \dots, n, \sum_{i=1}^{n} \frac{u_i - \overline{u}_i}{\hat{u}_i} \le 1 \right\}$$

Iterative algorithms for $\mathcal{U}^\Gamma$

$$\mathcal{P} = \left\{ \overline{u}_i \leq u_i \leq \overline{u}_i + \hat{u}_i, i = 1, \ldots, n, \sum_{i=1}^{n} \frac{u_i - \overline{u}_i}{\hat{u}_i} \leq \Gamma \right\}$$



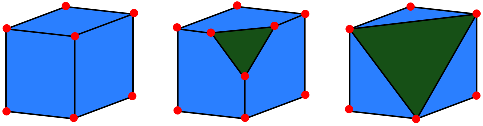
Theorem (Bertsimas and Sim [2003], Goetzmann et al. [2011], Álvarez-Miranda et al. [2013], Lee and Kwon [2014])

Cost uncertainty $\mathcal{U}^\Gamma$-*CO* $\Rightarrow$ *solving* $\sim n/2$ *problems CO*.

Numerical uncertainty $\mathcal{U}^\Gamma$-*CO* $\Rightarrow$ *solving* $\sim (n/2)^m$ *problems CO*.

Iterative algorithms for $\mathcal{U}^\Gamma$

$$\mathcal{U}^\Gamma = vertices\left(\left\{\overline{u}_i \leq u_i \leq \overline{u}_i + \hat{u}_i, i = 1, \dots, n, \sum_{i=1}^{n} \frac{u_i - \overline{u}_i}{\hat{u}_i} \leq \Gamma\right\}\right)$$



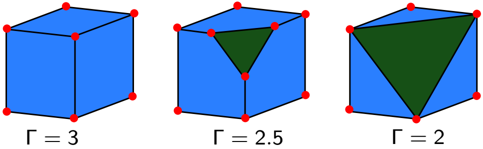
Theorem (Bertsimas and Sim [2003], Goetzmann et al. [2011], Álvarez-Miranda et al. [2013], Lee and Kwon [2014])

Cost uncertainty $\mathcal{U}^\Gamma$-*CO* $\Rightarrow$ *solving* $\sim n/2$ *problems CO*.

Numerical uncertainty $\mathcal{U}^\Gamma$-*CO* $\Rightarrow$ *solving* $\sim (n/2)^m$ *problems CO*.

Iterative algorithms for $\mathcal{U}^\Gamma$

$$\mathcal{U}^\Gamma = vertices\left(\left\{\overline{u}_i \le u_i \le \overline{u}_i + \hat{u}_i, i = 1, \dots, n, \sum_{i=1}^{n} \frac{u_i - \overline{u}_i}{\hat{u}_i} \le \Gamma\right\}\right)$$



Theorem (Bertsimas and Sim [2003], Goetzmann et al. [2011], Álvarez-Miranda et al. [2013], Lee and Kwon [2014])

Cost uncertainty $\mathcal{U}^\Gamma$-*CO* $\Rightarrow$ *solving* $\sim n/2$ *problems CO.*

Numerical uncertainty $\mathcal{U}^\Gamma$-*CO* $\Rightarrow$ *solving* $\sim (n/2)^m$ *problems CO.*

Other convex $\mathcal{U}$ (recall that $\mathcal{U} \Leftrightarrow \text{conv}(\mathcal{U})$)

Total deviation

$$\left\{ \overline{u} \leq u \leq \overline{u} + \hat{u}, \sum_{i=1}^{n} (u_i - \overline{u}_i) \leq \Omega \right\} \Rightarrow \text{solving 2 problems } CO$$

Knapsack uncertainty [Poss, 2017])

$$\left\{ \overline{u} \leq u \leq \overline{u} + \hat{u}, \sum_{i=1}^{n} a_i u_i \leq b \right\} \Rightarrow \text{solving } n \text{ problems } CO$$

Decision-dependent [Poss, 2013, 2014, Nohadani and Sharma, 2016]

$$\left\{ \overline{u} \leq u \leq \overline{u} + \hat{u}, \sum_{i=1}^{n} a_i u_i \leq b(x) \right\} \Rightarrow \text{solving } n \text{ problems } CO$$

Axis-parallel Ellipsoids [Mokarami and Hashemi, 2015]

$$\left\{ \sum_{i=1}^{n} \left(\frac{u_i - \overline{u}_i}{\hat{u}_i} \right)^2 \leq \Omega \right\} \Rightarrow \text{solving } n \max_i \hat{u}_i \text{ problems } CO$$

Other convex $\mathcal{U}$ (recall that $\mathcal{U} \Leftrightarrow \text{conv}(\mathcal{U})$)

Total deviation

$$\left\{ \overline{u} \leq u \leq \overline{u} + \hat{u}, \sum_{i=1}^{n} (u_i - \overline{u}_i) \leq \Omega \right\} \Rightarrow \text{solving 2 problems } CO$$

Knapsack uncertainty [Poss, 2017])

$$\left\{ \overline{u} \leq u \leq \overline{u} + \hat{u}, \sum_{i=1}^{n} a_i u_i \leq b \right\} \Rightarrow \text{solving } n \text{ problems } CO$$

Decision-dependent [Poss, 2013, 2014, Nohadani and Sharma, 2016]

$$\left\{ \overline{u} \leq u \leq \overline{u} + \hat{u}, \sum_{i=1}^{n} a_i u_i \leq b(x) \right\} \Rightarrow \text{solving } n \text{ problems } CO$$

Axis-parallel Ellipsoids [Mokarami and Hashemi, 2015]

$$\left\{ \sum_{i=1}^{n} \left(\frac{u_i - \overline{u}_i}{\hat{u}_i} \right)^2 \leq \Omega \right\} \Rightarrow \text{solving } n \max_i \hat{u}_i \text{ problems } CO$$

Other convex $\mathcal{U}$ (recall that $\mathcal{U} \Leftrightarrow \text{conv}(\mathcal{U})$)

Total deviation

$\left\{\overline{u} \le u \le \overline{u} + \hat{u}, \sum_{i=1}^{n} (u_i - \overline{u}_i) \le \Omega \right\}$ ⇒ solving 2 problems CO

Knapsack uncertainty [Poss, 2017])

$\left\{\overline{u} \le u \le \overline{u} + \hat{u}, \sum_{i=1}^{n} a_i u_i \le b \right\}$ ⇒ solving n problems CO

Decision-dependent [Poss, 2013, 2014, Nohadani and Sharma, 2016]

$\left\{\overline{u} \le u \le \overline{u} + \hat{u}, \sum_{i=1}^{n} a_i u_i \le b(x) \right\}$ ⇒ solving n problems CO

Axis-parallel Ellipsoids [Mokarami and Hashemi, 2015]

$\left\{\sum_{i=1}^{n} \left(\frac{u_i - \overline{u}_i}{\hat{u}_i}\right)^2 \le \Omega \right\}$ ⇒ solving $n \max_i \hat{u}_i$ problems CO

Other convex $\mathcal{U}$ (recall that $\mathcal{U} \Leftrightarrow \text{conv}(\mathcal{U})$)

Total deviation

$$\left\{\overline{u} \leq u \leq \overline{u} + \hat{u}, \sum_{i=1}^{n}(u_i - \overline{u}_i) \leq \Omega\right\} \Rightarrow \text{solving 2 problems } CO$$

Knapsack uncertainty [Poss, 2017])

$$\left\{\overline{u} \leq u \leq \overline{u} + \hat{u}, \sum_{i=1}^{n} a_i u_i \leq b\right\} \Rightarrow \text{solving } n \text{ problems } CO$$

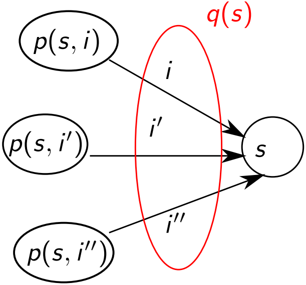
Decision-dependent [Poss, 2013, 2014, Nohadani and Sharma, 2016]

$$\left\{\overline{u} \leq u \leq \overline{u} + \hat{u}, \sum_{i=1}^{n} a_i u_i \leq b(x)\right\} \Rightarrow \text{solving } n \text{ problems } CO$$

Axis-parallel Ellipsoids [Mokarami and Hashemi, 2015]

$$\left\{\sum_{i=1}^{n}\left(\frac{u_i - \overline{u}_i}{\hat{u}_i}\right)^2 \leq \Omega\right\} \Rightarrow \text{solving } n \max_i \hat{u}_i \text{ problems } CO$$

Dynamic Programming [Klopfenstein and Nace, 2008, Monaci et al., 2013, Poss, 2014]



Classical recurrence

$F(s)$ = cheapest cost up to state s; $F(O) = 0$

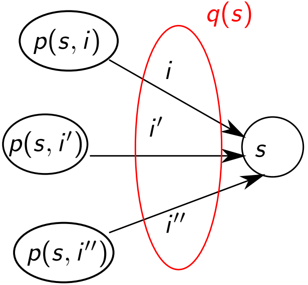
$$F(s) = \min_{i \in q(s)} \{F(p(s,i)) + u_i\}, \quad s \in S \backslash O$$

Robust recurrence

$F(s,\alpha)$ = cheapest cost up to state s with α remaning deviations; $F(O,\alpha) = 0$

$$\begin{cases} F(s,\alpha) = \min_{i \in q(s)} \{\max(F(p(s,i),\alpha) + \overline{u}_i, F(p(s,i),\alpha-1) + \overline{u}_i + \hat{u}_i)\}, \\ \qquad\qquad s \in S \backslash O, 1 \le \alpha \le \Gamma, \\ F(s,0) = \min_{i \in q(s)} \{F(p(s,i),0) + \overline{u}_i\}, \qquad s \in S \backslash O. \end{cases}$$

Dynamic Programming [Klopfenstein and Nace, 2008, Monaci et al., 2013, Poss, 2014]



Classical recurrence

$F(s)$ = cheapest cost up to state s; $F(O) = 0$

$$F(s) = \min_{i \in q(s)} \{F(p(s,i)) + u_i\}, \quad s \in S \backslash O$$

Robust recurrence

$F(s,\alpha)$ = cheapest cost up to state s with α remaning deviations; $F(O,\alpha) = 0$

$$\begin{cases} F(s,\alpha) = \min_{i \in q(s)} \{\max(F(p(s,i),\alpha) + \overline{u}_i, F(p(s,i),\alpha-1) + \overline{u}_i + \hat{u}_i)\}, \\ \qquad s \in S \backslash O, 1 \leq \alpha \leq \Gamma, \\ F(s,0) = \min_{i \in q(s)} \{F(p(s,i),0) + \overline{u}_i\}, \qquad s \in S \backslash O. \end{cases}$$

Are all problems easy?

Hard problems must have one of

1. non-constant number of robust "linear" constraints
2. "non-linear" constraints/cost function

Theorem (Pessoa et al. [2015])

$\mathcal{U}^{\Gamma}$-robust shortest path with time windows is $\mathcal{NP}$-hard in the strong sense.

Theorem (Bougeret et al. [2016])

Minimizing the weighted sum of completion times is $\mathcal{NP}$-hard in the strong sense.

Are all problems easy?

Hard problems must have one of

1. non-constant number of robust "linear" constraints
2. "non-linear" constraints/cost function

Theorem (Pessoa et al. [2015])

$\mathcal{U}^\Gamma$-robust shortest path with time windows is $\mathcal{NP}$-hard in the strong sense.

Theorem (Bougeret et al. [2016])

Minimizing the weighted sum of completion times is $\mathcal{NP}$-hard in the strong sense.

Are all problems easy?

Hard problems must have one of

1. non-constant number of robust “linear” constraints
2. “non-linear” constraints/cost function

Theorem (Pessoa et al. [2015])

$\mathcal{U}^\Gamma$-robust shortest path with time windows is $\mathcal{NP}$-hard in the strong sense.

Theorem (Bougeret et al. [2016])

Minimizing the weighted sum of completion times is $\mathcal{NP}$-hard in the strong sense.

Are all problems easy?

Hard problems must have one of

1. non-constant number of robust “linear” constraints
2. “non-linear” constraints/cost function

Theorem (Pessoa et al. [2015])

$\mathcal{U}^{\Gamma}$-robust shortest path with time windows is $\mathcal{NP}$-hard in the strong sense.

Theorem (Bougeret et al. [2016])

Minimizing the weighted sum of completion times is $\mathcal{NP}$-hard in the strong sense.

$\mathcal{U}^\Gamma$-*TWSP* is $\mathcal{NP}$-hard in the strong sense

ROBUST PATH WITH DEADLINES ($\mathcal{U}^\Gamma$-*PD*)
Input: Graph $D = (N, A)$, $\hat{u}_a$, Γ, $\overline{u} = 0$.
Question: There exists a path $p = o \rightsquigarrow i_2 \rightsquigarrow i_3 \rightsquigarrow \cdots \rightsquigarrow d$

$$\sum_{k=1}^{h-1} u_{i_k i_{k+1}} \leq \overline{b}_{i_h}, \text{ for each } h = 1, \ldots, l, \ u \in \mathcal{U}^\Gamma?$$

INDEPENDENT SET (*IS*)
Input: An undirected graph $G = (V, E)$ and a positive integer K.
Question: There exists $W \subseteq V$ such that $|W| \geq K$ and $\{i, j\} \nsubseteq W$ for each $\{i, j\} \in E$?

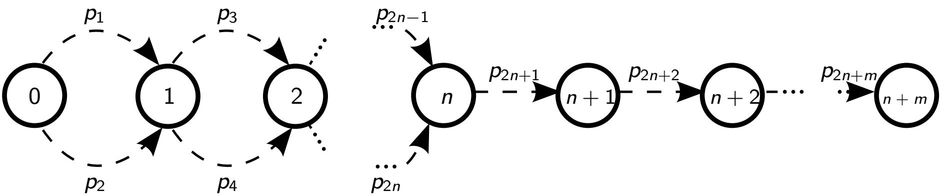
$\mathcal{U}^\Gamma$-*TWSP* is $\mathcal{NP}$-hard in the strong sense

ROBUST PATH WITH DEADLINES ($\mathcal{U}^\Gamma$-*PD*)
Input: Graph $D = (N, A)$, $\hat{u}_a$, Γ, $\overline{u} = 0$.
Question: There exists a path $p = o \rightsquigarrow i_2 \rightsquigarrow i_3 \rightsquigarrow \cdots \rightsquigarrow d$

$$\sum_{k=1}^{h-1} u_{i_k i_{k+1}} \leq \overline{b}_{i_h}, \text{ for each } h = 1, \ldots, l, \; u \in \mathcal{U}^\Gamma?$$

INDEPENDENT SET (*IS*)
Input: An undirected graph $G = (V, E)$ and a positive integer K.
Question: There exists $W \subseteq V$ such that $|W| \geq K$ and $\{i, j\} \not\subseteq W$ for each $\{i, j\} \in E$?

We are given an instance of *IS* with $|V| = n$ nodes and $|E| = m$



Set $W \subseteq V$ corresponds to path p_W:

- p_W contains p_{2i} iff $i \in W$
- p_W contains p_{2i-1} iff $i \notin W$

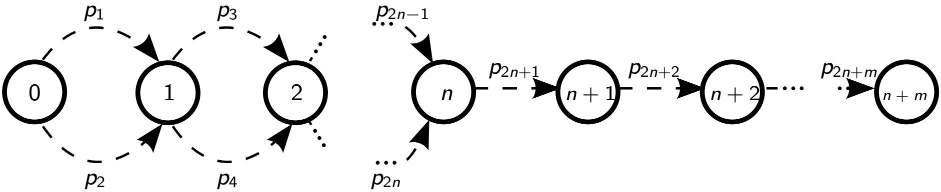
Observation

$$\sum_{k=1}^{h-1} u_{i_k i_{k+1}} \le \overline{b}_{i_h}, \ \forall u \in \mathcal{U}^\Gamma \quad \Leftrightarrow \max_{u \in \mathcal{U}^\Gamma} \sum_{k=1}^{h-1} u_{i_k i_{k+1}} \le \overline{b}_{i_h}$$

Parameters $\hat{u}$ and $\overline{b}$ are chosen such that

1. $\max_{u \in \mathcal{U}^\Gamma} \sum_{k=1}^{n-1} u_{i_k i_{k+1}} \le \overline{b}_n$ for $p_W \Leftrightarrow \quad |W| \ge K$
2. $\max_{u \in \mathcal{U}^\Gamma} \sum_{k=1}^{n+h-1} u_{i_k i_{k+1}} \le \overline{b}_{n+h}$ for $p_W \Leftrightarrow \quad e_h = \{i, j\} \not\subseteq W$

We are given an instance of *IS* with $|V| = n$ nodes and $|E| = m$



Set $W \subseteq V$ corresponds to path p_W:

- p_W contains p_{2i} iff $i \in W$
- p_W contains p_{2i-1} iff $i \notin W$

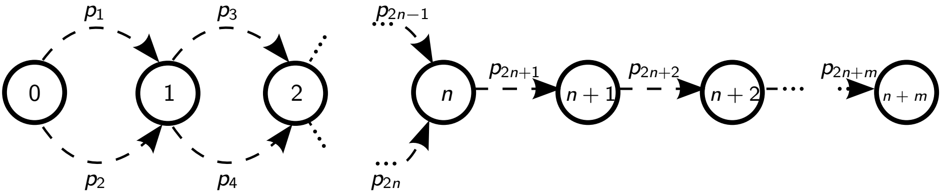
Observation

$$\sum_{k=1}^{h-1} u_{i_k i_{k+1}} \leq \overline{b}_{i_h}, \ \forall u \in \mathcal{U}^\Gamma \quad \Leftrightarrow \max_{u\in\mathcal{U}^\Gamma} \sum_{k=1}^{h-1} u_{i_k i_{k+1}} \leq \overline{b}_{i_h}$$

Parameters $\hat{u}$ and $\overline{b}$ are chosen such that

1. $\max_{u\in\mathcal{U}^\Gamma} \sum_{k=1}^{n-1} u_{i_k i_{k+1}} \leq \overline{b}_n$ for $p_W \Leftrightarrow \quad |W| \geq K$
2. $\max_{u\in\mathcal{U}^\Gamma} \sum_{k=1}^{n+h-1} u_{i_k i_{k+1}} \leq \overline{b}_{n+h}$ for $p_W \Leftrightarrow \quad e_h = \{i,j\} \not\subseteq W$

We are given an instance of IS with $|V| = n$ nodes and $|E| = m$



Set $W \subseteq V$ corresponds to path p_W:

- p_W contains p_{2i} iff $i \in W$
- p_W contains p_{2i-1} iff $i \notin W$

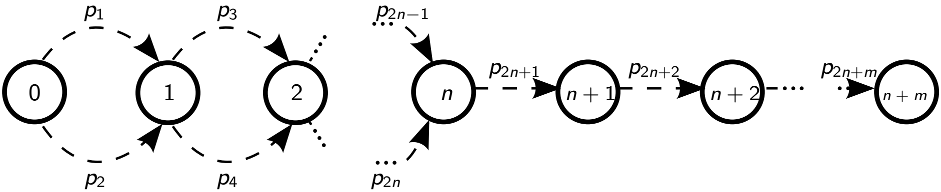
Observation

$$\sum_{k=1}^{h-1} u_{i_k i_{k+1}} \leq \overline{b}_{i_h}, \ \forall u \in \mathcal{U}^\Gamma \quad \Leftrightarrow \max_{u \in \mathcal{U}^\Gamma} \sum_{k=1}^{h-1} u_{i_k i_{k+1}} \leq \overline{b}_{i_h}$$

Parameters $\hat{u}$ and $\overline{b}$ are chosen such that

1. $\max_{u \in \mathcal{U}^\Gamma} \sum_{k=1}^{n-1} u_{i_k i_{k+1}} \leq \overline{b}_n$ for $p_W \Leftrightarrow \quad |W| \geq K$
2. $\max_{u \in \mathcal{U}^\Gamma} \sum_{k=1}^{n+h-1} u_{i_k i_{k+1}} \leq \overline{b}_{n+h}$ for $p_W \Leftrightarrow \quad e_h = \{i, j\} \not\subseteq W$

We are given an instance of IS with $|V| = n$ nodes and $|E| = m$



Set $W \subseteq V$ corresponds to path p_W:

- p_W contains p_{2i} iff $i \in W$
- p_W contains p_{2i-1} iff $i \notin W$

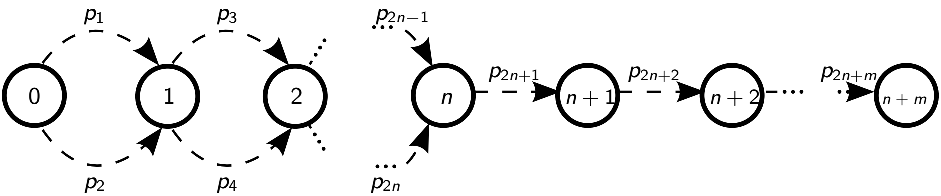
Observation

$$\sum_{k=1}^{h-1} u_{i_k i_{k+1}} \leq \overline{b}_{i_h}, \ \forall u \in \mathcal{U}^\Gamma \quad \Leftrightarrow \max_{u \in \mathcal{U}^\Gamma} \sum_{k=1}^{h-1} u_{i_k i_{k+1}} \leq \overline{b}_{i_h}$$

Parameters $\hat{u}$ and $\overline{b}$ are chosen such that

1. $\max_{u \in \mathcal{U}^\Gamma} \sum_{k=1}^{n-1} u_{i_k i_{k+1}} \leq \overline{b}_n$ for $p_W \Leftrightarrow \quad |W| \geq K$
2. $\max_{u \in \mathcal{U}^\Gamma} \sum_{k=1}^{n+h-1} u_{i_k i_{k+1}} \leq \overline{b}_{n+h}$ for $p_W \Leftrightarrow \quad e_h = \{i, j\} \not\subseteq W$

We are given an instance of IS with $|V| = n$ nodes and $|E| = m$



Set $W \subseteq V$ corresponds to path p_W:

- p_W contains p_{2i} iff $i \in W$
- p_W contains p_{2i-1} iff $i \notin W$

Observation

$$\sum_{k=1}^{h-1} u_{i_k i_{k+1}} \leq \overline{b}_{i_h}, \ \forall u \in \mathcal{U}^\Gamma \quad \Leftrightarrow \max_{u \in \mathcal{U}^\Gamma} \sum_{k=1}^{h-1} u_{i_k i_{k+1}} \leq \overline{b}_{i_h}$$

Parameters $\hat{u}$ and $\overline{b}$ are chosen such that

1. $\max_{u \in \mathcal{U}^\Gamma} \sum_{k=1}^{n-1} u_{i_k i_{k+1}} \leq \overline{b}_n$ for $p_W \Leftrightarrow \quad |W| \geq K$
2. $\max_{u \in \mathcal{U}^\Gamma} \sum_{k=1}^{n+h-1} u_{i_k i_{k+1}} \leq \overline{b}_{n+h}$ for $p_W \Leftrightarrow \quad e_h = \{i, j\} \nsubseteq W$

Cutting plane algorithms 2

Master problem

$$MP \qquad \min \left\{ \begin{array}{ll} c^T x \ : & \\ f(x,u) \le 0, & u \in \mathcal{U}^*, \\ a_k^T x \le d_k, & k = 1, \ldots, \ell \\ x \in \{0,1\}^n & \end{array} \right\}$$

1. **solve** $MP \rightarrow$ get $\tilde{x}$; **solve** $\max_{u \in \mathcal{U}} f(\tilde{x}, u) \rightarrow$ get $\tilde{u}$
2. **If** $f(\tilde{x}, \tilde{u}) > 0$ **then** $\mathcal{U}^* \leftarrow \mathcal{U}^* \cup \{\tilde{u}\}$; go back to 1

Examples [Agra et al., 2016]

Minimizing tardiness $f(x,u) = \sum_{i=1}^{n} w_i \max\{C_i(x,u) - d_i, 0\}$

Lot-sizing $f(x,u) = \sum_{i=1}^{n} \max \left\{ h_i(\sum_{j=1}^{i} x_j - \sum_{j=1}^{i} u_j), p_i(\sum_{j=1}^{i} u_j - \sum_{j=1}^{i} x_j) \right\}$

Cutting plane algorithms 2

Master problem

$$MP \qquad \min \left\{ \begin{array}{ll} c^T x \ : & \\ f(x,u) \leq 0, & u \in \mathcal{U}^*, \\ a_k^T x \leq d_k, & k = 1, \dots, \ell \\ x \in \{0,1\}^n & \end{array} \right\}$$

1. **solve** $MP \rightarrow$ get $\tilde{x}$; **solve** $\max_{u \in \mathcal{U}} f(\tilde{x}, u) \rightarrow$ get $\tilde{u}$
2. **If** $f(\tilde{x}, \tilde{u}) > 0$ **then** $\mathcal{U}^* \leftarrow \mathcal{U}^* \cup \{\tilde{u}\}$; go back to 1

Examples [Agra et al., 2016]

Minimizing tardiness $f(x,u) = \sum_{i=1}^{n} w_i \max\{C_i(x,u) - d_i, 0\}$

Lot-sizing $f(x,u) = \sum_{i=1}^{n} \max\left\{ h_i(\sum_{j=1}^{i} x_j - \sum_{j=1}^{i} u_j), p_i(\sum_{j=1}^{i} u_j - \sum_{j=1}^{i} x_j) \right\}$

Cutting plane algorithms 2

Master problem

$$
MP \qquad \min \left\{ \begin{array}{ll} c^T x & : \\ f(x,u) \le 0, & u \in \mathcal{U}^*, \\ a_k^T x \le d_k, & k = 1, \ldots, \ell \\ x \in \{0,1\}^n & \end{array} \right\}
$$

1. **solve** $MP \rightarrow$ get $\tilde{x}$; **solve** $\max_{u \in \mathcal{U}} f(\tilde{x}, u) \rightarrow$ get $\tilde{u}$
2. **If** $f(\tilde{x}, \tilde{u}) > 0$ **then** $\mathcal{U}^* \leftarrow \mathcal{U}^* \cup \{\tilde{u}\}$; go back to 1

Examples [Agra et al., 2016]

Minimizing tardiness $f(x,u) = \sum_{i=1}^{n} w_i \max\{C_i(x,u) - d_i, 0\}$

Lot-sizing $f(x,u) = \sum_{i=1}^{n} \max \left\{ h_i\left(\sum_{j=1}^{i} x_j - \sum_{j=1}^{i} u_j\right), p_i\left(\sum_{j=1}^{i} u_j - \sum_{j=1}^{i} x_j\right) \right\}$

Cutting plane algorithms 2

Master problem

$$MP \qquad \min\left\{ \begin{array}{ll} c^T x & : \\ f(x,u) \leq 0, & u \in \mathcal{U}^*, \\ a_k^T x \leq d_k, & k = 1, \dots, \ell \\ x \in \{0,1\}^n & \end{array} \right\}$$

1. **solve** $MP \rightarrow$ get $\tilde{x}$; **solve** $\max_{u \in \mathcal{U}} f(\tilde{x}, u) \rightarrow$ get $\tilde{u}$
2. **If** $f(\tilde{x}, \tilde{u}) > 0$ **then** $\mathcal{U}^* \leftarrow \mathcal{U}^* \cup \{\tilde{u}\}$; go back to 1

Examples [Agra et al., 2016]

Minimizing tardiness $f(x,u) = \sum_{i=1}^{n} w_i \max\{C_i(x,u) - d_i, 0\}$

Lot-sizing $f(x,u) = \sum_{i=1}^{n} \max\left\{ h_i(\sum_{j=1}^{i} x_i - \sum_{j=1}^{i} u_i), p_i(\sum_{j=1}^{i} u_i - \sum_{j=1}^{i} x_i) \right\}$

Cutting plane algorithms 2

Master problem

$$MP \qquad \min \left\{ \begin{array}{ll} c^T x \; : & \\ f(x,u) \le 0, & u \in \mathcal{U}^*, \\ a_k^T x \le d_k, & k = 1, \ldots, \ell \\ x \in \{0,1\}^n & \end{array} \right\}$$

1. **solve** $MP \rightarrow$ get $\tilde{x}$; **solve** $\max_{u \in \mathcal{U}} f(\tilde{x}, u) \rightarrow$ get $\tilde{u}$
2. **If** $f(\tilde{x}, \tilde{u}) > 0$ **then** $\mathcal{U}^* \leftarrow \mathcal{U}^* \cup \{\tilde{u}\}$; go back to 1

Examples [Agra et al., 2016]

Minimizing tardiness $f(x,u) = \sum_{i=1}^{n} w_i \max\{C_i(x,u) - d_i, 0\}$

Lot-sizing $f(x,u) = \sum_{i=1}^{n} \max \left\{ h_i\left(\sum_{j=1}^{i} x_i - \sum_{j=1}^{i} u_i\right), p_i\left(\sum_{j=1}^{i} u_i - \sum_{j=1}^{i} x_i\right) \right\}$

Cookbook for static problems

Dualization

good easy to apply

bad breaks combinatorial structure (e.g. shortest path)

Cutting plane algorithms (branch-and-cut)

good handle non-linear functions

bad implementation effort

Iterative algorithms, dynamic programming

good good theoretical bounds

bad solving n^s problems can be too much

Cookbook for static problems

Dualization

good easy to apply

bad breaks combinatorial structure (e.g. shortest path)

Cutting plane algorithms (branch-and-cut)

good handle non-linear functions

bad implementation effort

Iterative algorithms, dynamic programming

good good theoretical bounds

bad solving n^s problems can be too much

Cookbook for static problems

Dualization

good easy to apply

bad breaks combinatorial structure (e.g. shortest path)

Cutting plane algorithms (branch-and-cut)

good handle non-linear functions

bad implementation effort

Iterative algorithms, dynamic programming

good good theoretical bounds

bad solving n^s problems can be too much

Open questions

Knapsack/budget uncertainty

- Easy problems that turn $\mathcal{NP}$-hard
- Approximation algorithms

Scheduling seems to be a good niche.

Ellipsoidal uncertainty

Axis-parallel $\mathcal{NP}$-hard in general? (known FPTAS)

General Approximation algorithms

Open questions

Knapsack/budget uncertainty

- Easy problems that turn $\mathcal{NP}$-hard
- Approximation algorithms

Scheduling seems to be a good niche.

Ellipsoidal uncertainty

Axis-parallel $\mathcal{NP}$-hard in general? (known FPTAS)

General Approximation algorithms

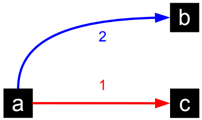
Outline

2-stages example: network design

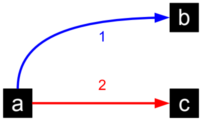
Demands vectors $\{u_1, \dots, u_n\}$ that must be routed **non-simultaneously** on a network to be designed.

$\Rightarrow$ two-stages program:

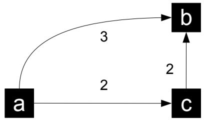
1. capacities
2. routing.



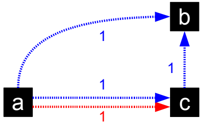
Demands for scenario 1



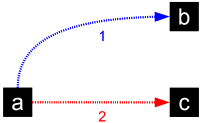
Demands for scenario 2



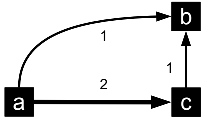
Capacity cost per unit



Routing for scenario 1



Routing for scenario 2



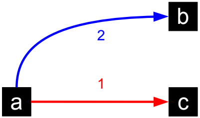
Capacity installation

2-stages example: network design

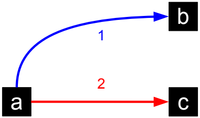
Demands vectors $\{u_1, \ldots, u_n\}$ that must be routed **non-simultaneously** on a network to be designed.

$\Rightarrow$ two-stages program:

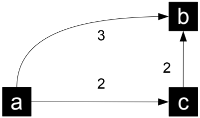
1. capacities
2. routing.



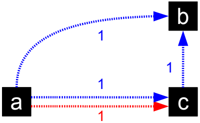
Demands for scenario 1



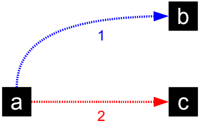
Demands for scenario 2



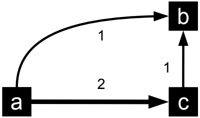
Capacity cost per unit



Routing for scenario 1



Routing for scenario 2



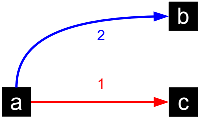
Capacity installation

2-stages example: network design

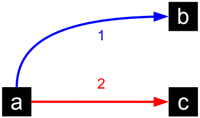
Demands vectors $\{u_1, \ldots, u_n\}$ that must be routed **non-simultaneously** on a network to be designed.

$\Rightarrow$ two-stages program:

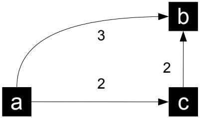
1. capacities
2. routing.



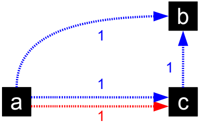
Demands for scenario 1



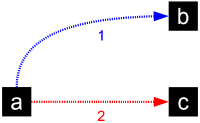
Demands for scenario 2



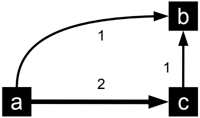
Capacity cost per unit



Routing for scenario 1



Routing for scenario 2



Capacity installation

multistage example: lot sizing

Given

- Production costs c
- Uncertain demands vectors
  $u_1 = (u_{11}, u_{12}, \ldots, u_{1t}), \ldots, u_n = (u_{n1}, u_{n2}, \ldots, u_{nt})$
- Storage costs h

Compute

- A production plan that minimizes the costs

Variables

- $y_i(u)$ production at period i for demand scenario u
- $x_i(u)$ stock at the end of period i for demand scenario u

$$
\begin{array}{lll}
\min & \gamma & \\
\text{s.t.} & \gamma \geq \sum_{i=1}^{t} (c_i y_i(u) + h_i x_i(u)) & u \in \mathcal{U} \\
& x_{i+1}(u) = x_i(u) + y_i(u) - u_i & i = 1, \dots, t, \; u \in \mathcal{U} \\
& x, y \geq 0 &
\end{array}
$$

Something is wrong !

Variables

- $y_i(u)$ production at period i for demand scenario u
- $x_i(u)$ stock at the end of period i for demand scenario u

$$\begin{array}{lll} \min & \gamma & \\ \text{s.t.} & \gamma \geq \sum_{i=1}^{t}(c_i y_i(u) + h_i x_i(u)) & u \in \mathcal{U} \\ & x_{i+1}(u) = x_i(u) + y_i(u) - u_i & i = 1, \dots, t, \ u \in \mathcal{U} \\ & x, y \geq 0 & \end{array}$$

Something is wrong !

Non-anticipativity - Example

Consider a lot-sizing problem with

- two different products A and B
- at most 1 unit of product (A and B together) can be produced at each period
- two time periods
- we know the demand of the current period at the beginning of the period
- two scenarios u and u' defined as follows:

$$u = \left[\begin{array}{lcc} & t=1 & t=2 \\ \hline A: & 0 & 2 \\ B: & 0 & 0 \end{array}\right], \qquad u' = \left[\begin{array}{lcc} & t=1 & t=2 \\ \hline A: & 0 & 0 \\ B: & 0 & 2 \end{array}\right],$$

Question Propose a feasible production plan

Answer The problem is infeasible !

Why? Because scenarios u and u' cannot be distinguished at the beginning of period 1, i.e.

$$u^1 = u'^1$$

Non-anticipativity - Example

Consider a lot-sizing problem with

- two different products A and B
- at most 1 unit of product (A and B together) can be produced at each period
- two time periods
- we know the demand of the current period at the beginning of the period
- two scenarios u and u' defined as follows:

$$u = \left[\begin{array}{c|cc} & t=1 & t=2 \\ \hline A: & 0 & 2 \\ B: & 0 & 0 \end{array}\right], \qquad u' = \left[\begin{array}{c|cc} & t=1 & t=2 \\ \hline A: & 0 & 0 \\ B: & 0 & 2 \end{array}\right],$$

Question Propose a feasible production plan

Answer The problem is infeasible !

Why? Because scenarios u and u' cannot be distinguished at the beginning of period 1, i.e.

$$u^1 = u'^1$$

Non-anticipativity - Example

Consider a lot-sizing problem with

- two different products A and B
- at most 1 unit of product (A and B together) can be produced at each period
- two time periods
- we know the demand of the current period at the beginning of the period
- two scenarios u and u' defined as follows:

$$u = \left[\begin{array}{ccc} & t=1 & t=2 \\ \hline A: & 0 & 2 \\ B: & 0 & 0 \end{array}\right], \qquad u' = \left[\begin{array}{ccc} & t=1 & t=2 \\ \hline A: & 0 & 0 \\ B: & 0 & 2 \end{array}\right],$$

Question Propose a feasible production plan

Answer The problem is infeasible !

Why? Because scenarios u and u' cannot be distinguished at the beginning of period 1, i.e.

$$u^1 = u'^1$$

Non-anticipativity - Example

Consider a lot-sizing problem with

- two different products A and B
- at most 1 unit of product (A and B together) can be produced at each period
- two time periods
- we know the demand of the current period at the beginning of the period
- two scenarios u and u' defined as follows:

$$u = \left[\begin{array}{lcc} & t=1 & t=2 \\ \hline A: & 1 & 2 \\ B: & 0 & 0 \end{array}\right], \qquad u' = \left[\begin{array}{lcc} & t=1 & t=2 \\ \hline A: & 0 & 0 \\ B: & 1 & 2 \end{array}\right],$$

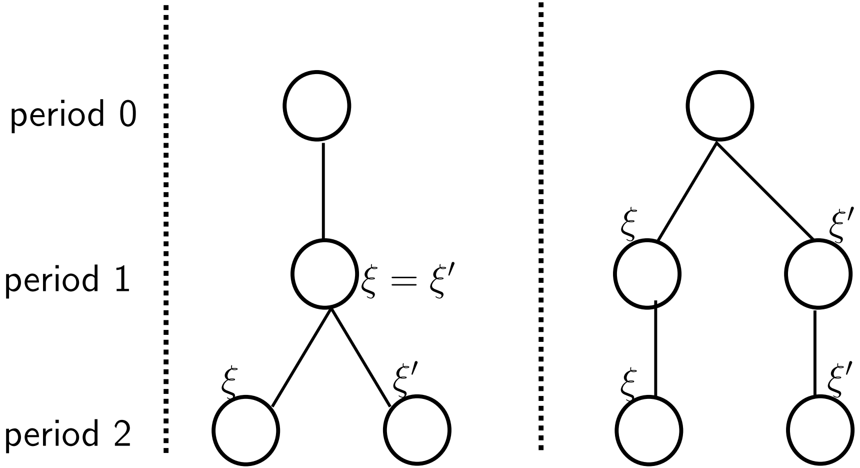
Question Propose a feasible production plan

Answer The problem is infeasible !

Why? Because scenarios u and u' cannot be distinguished at the beginning of period 1, i.e.

$$u^1 = u'^1$$

Graphical representation - scenario tree



multistage example: lot sizing - formulation

Variables

- $y_i(u)$ production at period i for demand scenario u
- $x_i(u)$ stock at the end of period i for demand scenario u

$$
\begin{aligned}
\min \quad & \gamma \\
\text{s.t.} \quad & \gamma \geq \sum_{i=1}^{t} (c_i y_i(u) + h_i x_i(u)) && u \in \mathcal{U} \\
& x_{i+1}(u) = x_i(u) + y_i(u) - u_i && i = 1, \ldots, t, \; u \in \mathcal{U} \\
& x, y \geq 0
\end{aligned}
$$

Variables

- $y_i(u)$ production at period i for demand scenario u
- $x_i(u)$ stock at the end of period i for demand scenario u

$$
\begin{aligned}
\min \quad & \gamma \\
\text{s.t.} \quad & \gamma \geq \sum_{i=1}^{t} (c_i y_i(u) + h_i x_i(u)) && u \in \mathcal{U} \\
& x_{i+1}(u) = x_i(u) + y_i(u) - u_i && i = 1, \ldots, t,\ u \in \mathcal{U} \\
& y_i(u) = y_i(u') && i = 1, \ldots, t,\ u, u' \in \mathcal{U}, u^i = u'^i \\
& x, y \geq 0
\end{aligned}
$$

Variables

- $y_i(u)$ production at period i for demand scenario u
- $x_i(u)$ stock at the end of period i for demand scenario u

$$
\begin{aligned}
\min \quad & \gamma \\
\text{s.t.} \quad & \gamma \geq \sum_{i=1}^{t} (c_i y_i(u^i) + h_i x_i(u)) && u \in \mathcal{U} \\
& x_{i+1}(u) = x_i(u) + y_i(u^i) - u_i && i = 1, \ldots, t, \; u \in \mathcal{U} \\
& x, y \geq 0
\end{aligned}
$$

2-stages integer example: knapsack

Given a capacity C, and a set of items I with profits c and weights $w(u)$, **find** the subset of items $N \subseteq I$ that maximizes its profit

such that
for each $u \in \mathcal{U}$, we can remove items in $K(u)$ from N and the total weight satisfies

$$\sum_{n \in N \setminus K(u)} w_n(u) \leq C$$

.

multistage integer example: lot sizing

Variables

- $y_i(u)$ production at period i for demand scenario u
- $x_i(u)$ stock at the end of period i for demand scenario u
- $z_i(u)$ allowing production for period i for demand scenario u

$$
\begin{aligned}
\min \quad & \gamma \\
\text{s.t.} \quad & \gamma \geq \sum_{i=1}^{t} (c_i y_i(u^i) + h_i x_i(u)) & u \in \mathcal{U} \\
& x_{i+1}(u) = x_i(u) + y_i(u^i) - u_i & i = 1, \dots, t, \; u \in \mathcal{U} \\
& y_i(u^i) \leq M z_i(u^i) & i = 1, \dots, t, \; u \in \mathcal{U} \\
& x, y \geq 0 \\
& z \in \{0, 1\}^{t|\mathcal{U}|}
\end{aligned}
$$

Outline

Exact solution procedure

$$\begin{aligned} & \min && c^T x \\ & \text{s.t.} && x \in \mathcal{X} \\ (P) & && A(u)x + Ey(u) \le b \quad u \in \mathcal{U} \end{aligned} \tag{6}$$

where $A(u) = A^0 + \sum A_k u_k$.

Lemma

We can replace (6) by

$$A(u)x + Ey(u) \le b \quad u \in \text{ext}(\mathcal{U}).$$

Idea of the proof:

$$A(u^*)x^* + Ey(u^*) \le b \Leftrightarrow \sum_{s=1}^{\text{ext}(\mathcal{U})} \lambda_s (A(u_s)x^* + Ey(u_s)) \le \sum_{s=1}^{\text{ext}(\mathcal{U})} \lambda_s b.$$

Exact solution procedure

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & x \in \mathcal{X} \\ (P) \qquad & A(u)x + Ey(u) \leq b \quad u \in \mathcal{U} \end{aligned} \tag{6}$$

where $A(u) = A^0 + \sum A_k u_k$.

Lemma

We can replace (6) by

$$A(u)x + Ey(u) \leq b \quad u \in \text{ext}(\mathcal{U}).$$

Idea of the proof:

$$A(u^*)x^* + Ey(u^*) \leq b \Leftrightarrow \sum_{s=1}^{\text{ext}(\mathcal{U})} \lambda_s \left(A(u_s)x^* + Ey(u_s)\right) \leq \sum_{s=1}^{\text{ext}(\mathcal{U})} \lambda_s b.$$

Master problem

$$\mathcal{U}^*\text{-}LSP' \quad \begin{array}{ll} \min & c^T x \\ \text{s.t.} & x \in \mathcal{X}. \\ & \text{Constraints corresponding to } u \in \mathcal{U}^* \end{array}$$

Separation

$$(SPL) \quad \begin{array}{lll} \max & (b - A^0 x^*)^T \pi - \sum_{k \in K} (A^{1k} x^*)^T v^k & \\ \text{s.t.} & u \in \mathcal{U} & \\ & E^T \pi = 0 & \\ & \mathbf{1}^T \pi = 1 & \\ & v_m^k \geq \pi_m - (1 - u^k) & k \in K, m \in M \\ & v_m^k \leq u^k & k \in K, m \in M \\ & \pi, v_m^k \geq 0, & \\ & u \in \{0,1\}^K. & \end{array}$$

Two different approaches

Benders $$(b - A(u^*)x)^T \pi^* \leq 0. \quad (7)$$

Row and column generation $$A(u^*)x + Ey(u^*) \leq b. \quad (8)$$

Algorithm 1: *RG* and *RCG*

repeat
- solve $\mathcal{U}^*$-*LSP'*;
- let x^* be an optimal solution;
- solve (*SPL*);
- let (u^*, π^*) be an optimal solution and z^* be the optimal solution cost;
- **if** $z^* > 0$ **then**
  - *RG*: add constraint (7) to $\mathcal{U}^*$-*LSP'*;
  - *RCG*: add constraint (8) to $\mathcal{U}^*$-*LSP'*;

until $z^* > 0$;

Two different approaches

Benders $$(b - A(u^*)x)^T \pi^* \leq 0. \quad (7)$$

Row and column generation $$A(u^*)x + Ey(u^*) \leq b. \quad (8)$$

Algorithm 2: *RG* and *RCG*

repeat
- solve $\mathcal{U}^*$-LSP';
- let x^* be an optimal solution;
- solve (SPL);
- let (u^*, π^*) be an optimal solution and z^* be the optimal solution cost;
- **if** $z^* > 0$ **then**
  - *RG*: add constraint (7) to $\mathcal{U}^*$-LSP';
  - *RCG*: add constraint (8) to $\mathcal{U}^*$-LSP';

until $z^* > 0$;

Numerical results

K	Γ	t_{RCG}	t_{SPL} (%)	iter	t_{RG}	$t_{P'}$
30	2	150	64	18	4967	13
30	3	301	78	19	**T**	213
30	4	1500	90	27	**T**	**M**
30	5	1344	91	25	**T**	**M**
40	2	365	69	21	6523	49
40	3	1037	88	22	**T**	**M**
40	4	6879	96	30	**T**	**M**
40	5	5866	95	31	**T**	**M**
40	6	**T**	–	–	**T**	**M**
50	2	694	73	23	**T**	98
50	3	4446	94	27	**T**	**M**
50	4	22645	98	35	**T**	**M**
50	5	**T**	–	–	**T**	**M**
50	6	**T**	–	–	**T**	**M**

Table: Results from Ayoub and Poss (2013) on a network design problem (Janos - 26/84).

Outline

Decision rules

$$
\begin{aligned}
\min \quad & c^T x \\
\text{s.t.} \quad & x \in \mathcal{X} \\
& A_t(u)x + \sum_{s=1}^{t} E_{ts} y_s(u^s) \leq b_t \quad t = 1, \ldots, T, \; u \in \mathcal{U}
\end{aligned}
$$

- We cannot use the previous decomposition anymore
- We can use decision rules, e.g.

$$
y(u) = y_0 + \sum_{k \in K} y_k u_k.
$$

- The problem gets the structure of a static robust problem.
- Can be dualized.
- More complex decision rules exist. Some can lead to exact reformulations; others can be approximated efficiently.
- Decision rules are "heuristic": they provide feasible solutions, possibly suboptimal.

Decision rules

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & x \in \mathcal{X} \\ & A_t(u)x + \sum_{s=1}^{t} E_{ts} y_s(u^s) \leq b_t \quad t = 1, \ldots, T, \; u \in \mathcal{U} \end{aligned}$$

- We cannot use the previous decomposition anymore
- We can use decision rules, e.g.

$$y(u) = y_0 + \sum_{k \in K} y_k u_k.$$

- The problem gets the structure of a static robust problem.
- Can be dualized.
- More complex decision rules exist. Some can lead to exact reformulations; others can be approximated efficiently.
- Decision rules are "heuristic": they provide feasible solutions, possibly suboptimal.

Decision rules

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & x \in \mathcal{X} \\ & A_t(u)x + \sum_{s=1}^{t} E_{ts} y_s(u^s) \le b_t \quad t = 1, \dots, T, \; u \in \mathcal{U} \end{aligned}$$

- We cannot use the previous decomposition anymore
- We can use decision rules, e.g.

$$y(u) = y_0 + \sum_{k \in K} y_k u_k.$$

- The problem gets the structure of a static robust problem.
- Can be dualized.
- More complex decision rules exist. Some can lead to exact reformulations; others can be approximated efficiently.
- Decision rules are "heuristic": they provide feasible solutions, possibly suboptimal.

Decision rules

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & x \in \mathcal{X} \\ & A_t(u)x + \sum_{s=1}^{t} E_{ts} y_s(u^s) \leq b_t \quad t = 1, \ldots, T, \; u \in \mathcal{U} \end{aligned}$$

- We cannot use the previous decomposition anymore
- We can use decision rules, e.g.

$$y(u) = y_0 + \sum_{k \in K} y_k u_k.$$

- The problem gets the structure of a static robust problem.
- Can be dualized.
- More complex decision rules exist. Some can lead to exact reformulations; others can be approximated efficiently.
- Decision rules are "heuristic": they provide feasible solutions, possibly suboptimal.

Decision rules

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & x \in \mathcal{X} \\ & A_t(u)x + \sum_{s=1}^{t} E_{ts} y_s(u^s) \leq b_t \quad t = 1, \dots, T, \; u \in \mathcal{U} \end{aligned}$$

- We cannot use the previous decomposition anymore
- We can use decision rules, e.g.

$$y(u) = y_0 + \sum_{k \in K} y_k u_k.$$

- The problem gets the structure of a static robust problem.
- Can be dualized.
- More complex decision rules exist. Some can lead to exact reformulations; others can be approximated efficiently.
- Decision rules are "heuristic": they provide feasible solutions, possibly suboptimal.

Decision rules

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & x \in \mathcal{X} \\ & A_t(u)x + \sum_{s=1}^{t} E_{ts} y_s(u^s) \leq b_t \quad t = 1, \ldots, T, \; u \in \mathcal{U} \end{aligned}$$

- We cannot use the previous decomposition anymore
- We can use decision rules, e.g.

$$y(u) = y_0 + \sum_{k \in K} y_k u_k.$$

- The problem gets the structure of a static robust problem.
- Can be dualized.
- More complex decision rules exist. Some can lead to exact reformulations; others can be approximated efficiently.
- Decision rules are "heuristic": they provide feasible solutions, possibly suboptimal.

Decision rules: Example for network design problem

Static $y_{ka}(u) = y_{ka} u_k$

Affine $y_{ka}(u) = y_{ka0} + \sum_{h \in K} y_{kah} u_h$

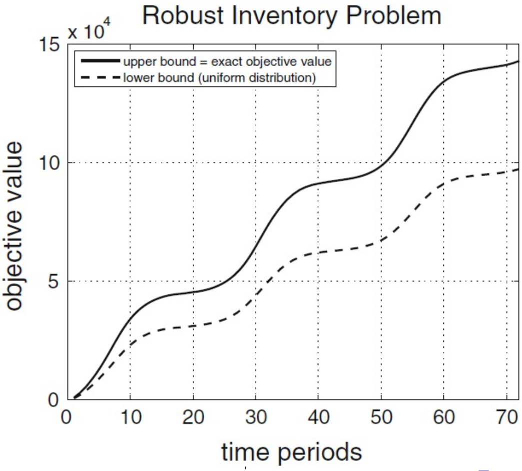
Dynamic $y_{ka}(u)$ is an arbitrary function

polska	0.25	2.612E+02	12.4	$\geq$ 0.0
	0.1	2.874E+02	12.8	$\geq$ 0.0
	0.05	2.935E+02	10.9	$\geq$ 0.0
nobel-us	0.25	2.949E+05	10.5	$\geq$ 0.0
	0.1	3.156E+05	9.2	$\geq$ 0.0
	0.05	3.198E+05	7.9	$\geq$ 0.0
atlanta	0.25	2.001E+05	4.7	5.4
	0.1	2.096E+05	3.4	3.6
	0.05	2.117E+05	2.7	2.7
newyork	0.25	9.852E+02	0.0	0.0
	0.1	9.852E+02	0.0	0.0
	0.05	9.852E+02	0.0	0.0
france	0.25	1.040E+01	7.7	$\geq$ 0.0
	0.1	1.100E+01	6.4	$\geq$ 0.0
	0.05	1.120E+01	$\geq$ 5.4	$\geq$ 0.0

Dual bound

Question: Can we obtain some guarantee on the quality of the affine solution ?

Answer: Using a dual model ...



Outline

What about integer adjustable variables ?

Notation $u^s = (u_1, \ldots, u_s)$

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & x \in \mathcal{X} \\ & A_t(u)x + \sum_{s=1}^{t} E_{ts} y_s(u^s) \le b_t(u) \quad && t = 1, \ldots, T,\ u \in \mathcal{U} \qquad (9) \\ & y(u) \in \mathbb{R}^{L_1} \times \mathbb{Z}^{L_2} && u \in \mathcal{U} \end{aligned}$$

Observation

Constraints (9) are not equivalent to

$$A_t(u)x + \sum_{s=1}^{t} E_{ts} y_s(u^s) \le b_t(u) \qquad t = 1, \ldots, T,\ u \in \text{ext}(\mathcal{U})$$

What about integer adjustable variables ?

Notation $u^s = (u_1, \dots, u_s)$

$$
\begin{aligned}
\min \quad & c^T x \\
\text{s.t.} \quad & x \in \mathcal{X} \\
& A_t(u)x + \sum_{s=1}^{t} E_{ts} y_s(u^s) \le b_t(u) \qquad t = 1, \dots, T, \; u \in \mathcal{U} \qquad (9) \\
& y(u) \in \mathbb{R}^{L_1} \times \mathbb{Z}^{L_2} \qquad u \in \mathcal{U}
\end{aligned}
$$

Observation

Constraints (9) are not equivalent to

$$
A_t(u)x + \sum_{s=1}^{t} E_{ts} y_s(u^s) \le b_t(u) \qquad t = 1, \dots, T, \; u \in \text{ext}(\mathcal{U})
$$

What about integer adjustable variables ?

Notation $u^s = (u_1, \ldots, u_s)$

$$
\begin{aligned}
\min \quad & c^T x \\
\text{s.t.} \quad & x \in \mathcal{X} \\
& A_t(u)x + \sum_{s=1}^{t} E_{ts} y_s(u^s) \leq b_t(u) \quad && t = 1, \ldots, T, \ u \in \mathcal{U} \qquad (9) \\
& y(u) \in \mathbb{R}^{L_1} \times \mathbb{Z}^{L_2} && u \in \mathcal{U}
\end{aligned}
$$

Observation

Constraints (9) are not equivalent to

$$A_t(u)x + \sum_{s=1}^{t} E_{ts} y_s(u^s) \leq b_t(u) \qquad t = 1, \ldots, T, \ u \in \text{ext}(\mathcal{U})$$

2-stages example: knapsack

Given

Set	N
Capacity	C
Weights	u
Profit	c
Removal limit	K

Solve

$$\max \left\{ \begin{array}{lll} & \sum_{i\in N} c_i x_i & \\ \text{s.t.} & \sum_{i\in N} u_i(x_i - y_i(u)) \le C & u \in \mathcal{U} \\ & \sum_{i\in N} y_i(u) \le K & u \in \mathcal{U} \\ & x, y(u) \in \{0,1\} & \end{array} \right\}$$

Example ($\mathcal{U} \neq \text{ext}(\mathcal{U})$)

Parameters $N = \{1,2\}$, $\overline{u}_i = 0, \hat{u}_i = 1, c_i = 1$, $C = 0$, $\Gamma = K = 1$

$\mathcal{U}^\Gamma$ opt: $x_1 = 1, x_2 = 0$ with cost 1, worst u: $(0.5, 0.5)$

$\text{ext}(\mathcal{U}^\Gamma)$ opt: $x_1 = x_2 = 1$ with cost 2, worst u: $(1,0)$

2-stages example: knapsack

Given

Set N

Capacity C

Weights u

Profit c

Removal limit K

Solve

$$\max \left\{ \begin{array}{lll} & \sum_{i \in N} c_i x_i & \\ \text{s.t.} & \sum_{i \in N} u_i(x_i - y_i(u)) \leq C & u \in \mathcal{U} \\ & \sum_{i \in N} y_i(u) \leq K & u \in \mathcal{U} \\ & x, y(u) \in \{0,1\} & \end{array} \right\}$$

Example ($\mathcal{U} \neq \text{ext}(\mathcal{U})$)

Parameters $N = \{1,2\}, \quad \overline{u}_i = 0, \hat{u}_i = 1, c_i = 1, \quad C = 0, \quad \Gamma = K = 1$

$\mathcal{U}^\Gamma$ opt: $x_1 = 1, x_2 = 0$ with cost 1, worst u: $(0.5, 0.5)$

$\text{ext}(\mathcal{U}^\Gamma)$ opt: $x_1 = x_2 = 1$ with cost 2, worst u: $(1, 0)$

2-stages example: knapsack

Given

Set	N
Capacity	C
Weights	u
Profit	c
Removal limit	K

Solve

$$\max \left\{ \begin{array}{rll} & \sum_{i\in N} c_i x_i & \\ \text{s.t.} & \sum_{i\in N} u_i(x_i - y_i(u)) \leq C & u \in \mathcal{U} \\ & \sum_{i\in N} y_i(u) \leq K & u \in \mathcal{U} \\ & x, y(u) \in \{0,1\} & \end{array} \right\}$$

Example ($\mathcal{U} \neq \text{ext}(\mathcal{U})$)

Parameters $N = \{1,2\}$, $\overline{u}_i = 0, \hat{u}_i = 1, c_i = 1$, $C = 0$, $\Gamma = K = 1$

$\mathcal{U}^\Gamma$ opt: $x_1 = 1, x_2 = 0$ with cost 1, worst u: $(0.5, 0.5)$

$\text{ext}(\mathcal{U}^\Gamma)$ opt: $x_1 = x_2 = 1$ with cost 2, worst u: $(1, 0)$

2-stages example: knapsack

Given

Set N

Capacity C

Weights u

Profit c

Removal limit K

Solve

$$\max \left\{ \begin{array}{ll} \sum_{i \in N} c_i x_i & \\ \text{s.t.} \sum_{i \in N} u_i (x_i - y_i(u)) \leq C & u \in \mathcal{U} \\ \sum_{i \in N} y_i(u) \leq K & u \in \mathcal{U} \\ x, y(u) \in \{0,1\} & \end{array} \right\}$$

Example ($\mathcal{U} \neq \text{ext}(\mathcal{U})$)

Parameters $N = \{1,2\}, \quad \overline{u}_i = 0, \hat{u}_i = 1, c_i = 1, \quad C = 0, \quad \Gamma = K = 1$

$\mathcal{U}^\Gamma$ opt: $x_1 = 1, x_2 = 0$ with cost 1, worst u: $(0.5, 0.5)$

$\text{ext}(\mathcal{U}^\Gamma)$ opt: $x_1 = x_2 = 1$ with cost 2, worst u: $(1, 0)$

2-stages example: knapsack

Given

Set	N
Capacity	C
Weights	u
Profit	c
Removal limit	K

Solve

$$\max \left\{ \begin{array}{ll} \sum_{i \in N} c_i x_i & \\ \text{s.t.} \sum_{i \in N} u_i(x_i - y_i(u)) \leq C & u \in \mathcal{U} \\ \sum_{i \in N} y_i(u) \leq K & u \in \mathcal{U} \\ x, y(u) \in \{0,1\} & \end{array} \right\}$$

Example ($\mathcal{U} \neq \text{ext}(\mathcal{U})$)

Parameters $N = \{1,2\}, \quad \overline{u}_i = 0, \hat{u}_i = 1, c_i = 1, \quad C = 0, \quad \Gamma = K = 1$

$\mathcal{U}^\Gamma$ opt: $x_1 = 1, x_2 = 0$ with cost 1, worst u: $(0.5, 0.5)$

$\text{ext}(\mathcal{U}^\Gamma)$ opt: $x_1 = x_2 = 1$ with cost 2, worst u: $(1,0)$

2-stages example: knapsack

Given

Set	N
Capacity	C
Weights	u
Profit	c
Removal limit	K

Solve

$$\max \left\{ \begin{array}{ll} \sum_{i \in N} c_i x_i & \\ \text{s.t.} \sum_{i \in N} u_i(x_i - y_i(u)) \leq C & u \in \mathcal{U} \\ \sum_{i \in N} y_i(u) \leq K & u \in \mathcal{U} \\ x, y(u) \in \{0,1\} & \end{array} \right\}$$

Example ($\mathcal{U} \neq \text{ext}(\mathcal{U})$)

Parameters $N = \{1,2\}$, $\overline{u}_i = 0, \hat{u}_i = 1, c_i = 1$, $C = 0$, $\Gamma = K = 1$

$\mathcal{U}^\Gamma$ opt: $x_1 = 1, x_2 = 0$ with cost 1, worst u: $(0.5, 0.5)$

$\text{ext}(\mathcal{U}^\Gamma)$ opt: $x_1 = x_2 = 1$ with cost 2, worst u: $(1, 0)$

2-stages example: knapsack

Given

Set N

Capacity C

Weights u

Profit c

Removal limit K

Solve

$$\max \left\{ \begin{aligned} & \sum_{i \in N} c_i x_i \\ \text{s.t.} & \sum_{i \in N} u_i (x_i - y_i(u)) \leq C & u \in \mathcal{U} \\ & \sum_{i \in N} y_i(u) \leq K & u \in \mathcal{U} \\ & x, y(u) \in \{0,1\} \end{aligned} \right\}$$

Example ($\mathcal{U} \neq \text{ext}(\mathcal{U})$)

Parameters $N = \{1,2\}, \quad \overline{u}_i = 0, \hat{u}_i = 1, c_i = 1, \quad C = 0, \quad \Gamma = K = 1$

$\mathcal{U}^\Gamma$ opt: $x_1 = 1, x_2 = 0$ with cost 1, worst u: $(0.5, 0.5)$

$\text{ext}(\mathcal{U}^\Gamma)$ opt: $x_1 = x_2 = 1$ with cost 2, worst u: $(1, 0)$

What to do ?

Three lines of research have been proposed in the litterature:

1. Partitioning the uncertainty set.
   - $\mathcal{U} = \mathcal{U}^1 \cup \ldots \cup \mathcal{U}^n$
   - Constraints

$$A_t(u)x + \sum_{s=1}^{t} E_{ts} y_s(u^s) \leq b_t(u) \qquad t = 1, \ldots, T, \; u \in \mathcal{U}$$

   become

$$A_t(u)x + \sum_{s=1}^{t} E_{ts} y_{s1} \leq b_t(u) \qquad t = 1, \ldots, T, \; u \in \mathcal{U}^1$$
$$\ldots$$
$$A_t(u)x + \sum_{s=1}^{t} E_{ts} y_{sn} \leq b_t(u) \qquad t = 1, \ldots, T, \; u \in \mathcal{U}^n$$

2. Row-and-column generation algorithms by Zhao and Zeng [2012]
   Assumptions
   - Problems with complete recourse
   - $\mathcal{X}(\mathcal{U}) = \mathcal{X}(ext(\mathcal{U}))$

   Algorithms Nested row-and-column generation algorithms.
3. Non-linear decision rules proposed by Bertsimas and Georghiou [2015]

What to do ?

Three lines of research have been proposed in the litterature:

1. Partitioning the uncertainty set.
   - $\mathcal{U} = \mathcal{U}^1 \cup \ldots \cup \mathcal{U}^n$
   - Constraints

$$A_t(u)x + \sum_{s=1}^{t} E_{ts} y_s(u^s) \leq b_t(u) \qquad t = 1, \ldots, T, \; u \in \mathcal{U}$$

   become

$$A_t(u)x + \sum_{s=1}^{t} E_{ts} y_{s1} \leq b_t(u) \qquad t = 1, \ldots, T, \; u \in \mathcal{U}^1$$
$$\ldots$$
$$A_t(u)x + \sum_{s=1}^{t} E_{ts} y_{sn} \leq b_t(u) \qquad t = 1, \ldots, T, \; u \in \mathcal{U}^n$$

2. Row-and-column generation algorithms by Zhao and Zeng [2012]

   Assumptions
   - Problems with complete recourse
   - $\mathcal{K}(\mathcal{U}) = \mathcal{K}(\text{ext}(\mathcal{U}))$

   Algorithms Nested row-and-column generation algorithms.

3. Non-linear decision rules proposed by Bertsimas and Georghiou [2015]

What to do ?

Three lines of research have been proposed in the litterature:

1. Partitioning the uncertainty set.
   - $\mathcal{U} = \mathcal{U}^1 \cup \ldots \cup \mathcal{U}^n$
   - Constraints

$$A_t(u)x + \sum_{s=1}^{t} E_{ts} y_s(u^s) \leq b_t(u) \qquad t = 1, \ldots, T, \; u \in \mathcal{U}$$

   become

$$A_t(u)x + \sum_{s=1}^{t} E_{ts} y_{s1} \leq b_t(u) \qquad t = 1, \ldots, T, \; u \in \mathcal{U}^1$$
$$\ldots$$
$$A_t(u)x + \sum_{s=1}^{t} E_{ts} y_{sn} \leq b_t(u) \qquad t = 1, \ldots, T, \; u \in \mathcal{U}^n$$

2. Row-and-column generation algorithms by Zhao and Zeng [2012]
   Assumptions
   - Problems with complete recourse
   - $\mathcal{K}(\mathcal{U}) = \mathcal{K}(\text{ext}(\mathcal{U}))$

   Algorithms Nested row-and-column generation algorithms.
3. Non-linear decision rules proposed by Bertsimas and Georghiou [2015]

Dynamic partition [Bertsimas and Dunning, 2016, Postek and den Hertog, 2016]

Partition $\mathcal{P} = \mathcal{U}^1 \cup \cdots \cup \mathcal{U}^n$

Heuristic bound $\mathcal{U}\text{-}CO(\mathcal{P})$

Algorithm

1. Solve $\mathcal{U}\text{-}CO(\mathcal{P})$
2. Refine $\mathcal{P}$, go back to 1

Partition step

- active vectors u lie in different subsets

$\Rightarrow$ Voronoi diagrams

$\mathcal{U}\text{-}CO(\mathcal{P})$ dimensions increases linearly with $|\mathcal{P}|$

Dynamic partition [Bertsimas and Dunning, 2016, Postek and den Hertog, 2016]

Partition $\mathcal{P} = \mathcal{U}^1 \cup \cdots \cup \mathcal{U}^n$

Heuristic bound $\mathcal{U}\text{-}CO(\mathcal{P})$

Algorithm

1. Solve $\mathcal{U}\text{-}CO(\mathcal{P})$
2. Refine $\mathcal{P}$, go back to 1

Partition step

- active vectors u lie in different subsets

$\Rightarrow$ Voronoi diagrams

$\mathcal{U}\text{-}CO(\mathcal{P})$ dimensions increases linearly with $|\mathcal{P}|$

Dynamic partition [Bertsimas and Dunning, 2016, Postek and den Hertog, 2016]

Partition $\mathcal{P} = \mathcal{U}^1 \cup \cdots \cup \mathcal{U}^n$

Heuristic bound $\mathcal{U}\text{-}CO(\mathcal{P})$

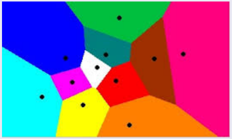
Algorithm

1. Solve $\mathcal{U}\text{-}CO(\mathcal{P})$
2. Refine $\mathcal{P}$, go back to 1

Partition step

- active vectors u lie in different subsets

$\Rightarrow$ Voronoi diagrams



$\mathcal{U}\text{-}CO(\mathcal{P})$ dimensions increases linearly with $|\mathcal{P}|$

Dynamic partition [Bertsimas and Dunning, 2016, Postek and den Hertog, 2016]

Partition $\mathcal{P} = \mathcal{U}^1 \cup \cdots \cup \mathcal{U}^n$

Heuristic bound $\mathcal{U}\text{-}CO(\mathcal{P})$

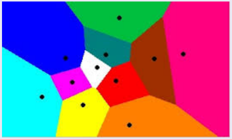
Algorithm

1. Solve $\mathcal{U}\text{-}CO(\mathcal{P})$
2. Refine $\mathcal{P}$, go back to 1

Partition step

- active vectors u lie in different subsets

$\Rightarrow$ Voronoi diagrams



$\mathcal{U}\text{-}CO(\mathcal{P})$ dimensions increases linearly with $|\mathcal{P}|$

Comparison of Bertsimas and Georghiou [2015], Bertsimas and Dunning [2016], Postek and den Hertog [2016] on lot-sizing.

$w_i^n(u)$ order a fixed amount q_n at time i

Comparison of Bertsimas and Georghiou [2015], Bertsimas and Dunning [2016], Postek and den Hertog [2016] on lot-sizing.

$w_i^n(u)$ order a fixed amount q_n at time i

Method		T = 4	6	8	10
Our method (2 iter.)	Gap (%)	13.0	10.3	11.6	14.9
	Time (s)	0.0	0.5	7.7	108.6
Our method (3 iter.)	Gap (%)	11.4	9.3	11.3	14.9
	Time (s)	0.2	2.0	52.4	309.3
Postek and Den Hertog (2014)	Gap (%)	11.5	14.1	15.7	15.7
	Time (s)	0.4	1.6	10.8	77.8
Bertsimas and Georghiou (2015)	Gap (%)	17.2	34.5	37.6	-
	Time (s)	3381	9181	28743	-

Concluding remarks

Static problems

- Numerical solution by **dualization** or **decomposition algorithms**.
- $\mathcal{U}$ "nice" structure and non-linear objective $\Rightarrow$ interesting open problems

Adjustable problems

- Hot topic
- Very hard to solve!
- Even good generic heuristic approaches would be interesting.

Concluding remarks

Static problems

- Numerical solution by **dualization** or **decomposition algorithms**.
- $\mathcal{U}$ "nice" structure and non-linear objective $\Rightarrow$ interesting open problems

Adjustable problems

- Hot topic
- Very hard to solve!
- Even good generic heuristic approaches would be interesting.

SI EJCO: Robust Combinatorial Optimization

- valid inequalities for robust MILPs,
- decomposition algorithms for robust MILPs,
- constraint programming approaches to robust combinatorial optimization,
- heuristic and meta-heuristic algorithms for hard robust combinatorial problems,
- ad-hoc combinatorial algorithms,
- novel applications of robust combinatorial optimization,
- multi-stage integer robust optimization,
- recoverable robust optimization,

Deadline: July 15 2017

References I

Agostinho Agra, Marcio C. Santos, Dritan Nace, and Michael Poss. A dynamic programming approach for a class of robust optimization problems. *SIAM Journal on Optimization*, (3):1799–1823, 2016.

E. Álvarez-Miranda, I. Ljubić, and P. Toth. A note on the bertsimas & sim algorithm for robust combinatorial optimization problems. *4OR*, 11(4): 349–360, 2013.

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

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

Dimitris Bertsimas and Iain Dunning. Multistage robust mixed-integer optimization with adaptive partitions, 2016. URL `http://dx.doi.org/10.1287/opre.2016.1515`.

References II

Dimitris Bertsimas and Angelos Georghiou. Design of near optimal decision rules in multistage adaptive mixed-integer optimization. *Operations Research*, 63(3):610–627, 2015. doi: 10.1287/opre.2015.1365. URL http://dx.doi.org/10.1287/opre.2015.1365.

Dimitris Bertsimas, Iain Dunning, and Miles Lubin. Reformulation versus cutting-planes for robust optimization. *Computational Management Science*, 13(2):195–217, 2016.

M. Bougeret, Artur A. Pessoa, and M. Poss. Robust scheduling with budgeted uncertainty, 2016. Submitted.

K.-S. Goetzmann, S. Stiller, and C. Telha. Optimization over integers with robustness in cost and few constraints. In *WAOA*, pages 89–101, 2011.

O. Klopfenstein and D. Nace. A robust approach to the chance-constrained knapsack problem. *Oper. Res. Lett.*, 36(5):628–632, 2008.

P. Kouvelis and G. Yu. *Robust discrete optimization and its applications*, volume 14. Springer Science & Business Media, 2013.

Taehan Lee and Changhyun Kwon. A short note on the robust combinatorial optimization problems with cardinality constrained uncertainty. *4OR*, pages 373–378, 2014.

Shaghayegh Mokarami and S Mehdi Hashemi. Constrained shortest path with uncertain transit times. *Journal of Global Optimization*, 63(1): 149–163, 2015.

M. Monaci, U. Pferschy, and P. Serafini. Exact solution of the robust knapsack problem. *Computers & OR*, 40(11):2625–2631, 2013.

Omid Nohadani and Kartikey Sharma. Optimization under decision-dependent uncertainty. *arXiv preprint arXiv:1611.07992*, 2016.

A. A. Pessoa, L. Di Puglia Pugliese, F. Guerriero, and M. Poss. Robust constrained shortest path problems under budgeted uncertainty. *Networks*, 66(2):98–111, 2015.

M. Poss. Robust combinatorial optimization with variable budgeted uncertainty. *4OR*, 11(1):75–92, 2013.

References IV

M. Poss. Robust combinatorial optimization with variable cost uncertainty. *European Journal of Operational Research*, 237(3):836–845, 2014.

Michael Poss. Robust combinatorial optimization with knapsack uncertainty. 2017. Available at `hal.archives-ouvertes.fr/tel-01421260`.

Krzysztof Postek and Dick den Hertog. Multistage adjustable robust mixed-integer optimization via iterative splitting of the uncertainty set. *INFORMS Journal on Computing*, 28(3):553–574, 2016. doi: 10.1287/ijoc.2016.0696. URL `http://dx.doi.org/10.1287/ijoc.2016.0696`.

Long Zhao and Bo Zeng. An exact algorithm for two-stage robust optimization with mixed integer recourse problems. *submitted, available on Optimization-Online. org*, 2012.