Decomposition in adjustable robust optimization: illustrations on network design problems

Michael Poss

joint work with Artur Alves Pessoa and Josette Ayoub

Heudiasyc UMR CNRS 7253, Université de Technologie de Compiègne, France

22nd of May 2014

Outline

1. Introduction
2. Models
3. Robust network design: basic models
4. Exact solutions with fixed recourse
5. Affine decision rules and random recourse

Outline

Knapsack problem under uncertainty

$$\begin{aligned} \min \quad & \sum_{i \in N} c_i x_i \\ \text{s.t.} \quad & \sum_{i \in N} a_i x_i \leq b \\ & x \in \{0,1\}^n \end{aligned}$$

Suppose that the parameters (c, a) are uncertain:

- They vary over time
- They must be predicted from historical data
- They cannot be measured with enough accuracy
- ...

Let's do something clever (and useful)!

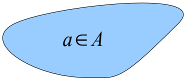
How much do we know?

$$\underbrace{\text{Stochastic programming}}_{\text{A lot}} \quad \Leftrightarrow \quad \underbrace{\text{Robust programming}}_{\text{A little}}$$

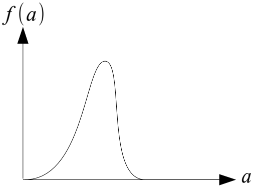
Mean value (Deterministic)

$\bullet\ \boldsymbol{E}[a]$

Robust



Stochastic



How much do we know?

Robust pr. Uncertain parameters are merely assumed to belong to an uncertainty set Ξ $\Rightarrow$ one wishes to optimize some worst-case objective over the uncertainty set

Stochastic pr. Uncertain parameters are precisely described by probability distributions $\Rightarrow$ one wishes to optimize some expectation, variance, Value-at-risk, ...

How much do we know?

Robust pr. Uncertain parameters are merely assumed to belong to an uncertainty set $\Xi \Rightarrow$ one wishes to optimize some worst-case objective over the uncertainty set

Distributionally robust pr. Uncertain parameters are described by classes of probability distributions $\mathcal{F} \Rightarrow$ one wishes to optimize some worst-case objective over the ambiguity classes

- $\mathcal{F}$ is a singleton Stochastic programming
- $\mathcal{F}$ contains all distributions over Ξ Robust optimization

Stochastic pr. Uncertain parameters are precisely described by probability distributions $\Rightarrow$ one wishes to optimize some expectation, variance, Value-at-risk, ...

When do we take decisions?

Now All decisions must be taken before the uncertainty is known with precision ⇒ probability constraints, (static) robust optimization

Delayed Some decisions may be delayed until the uncertainty is revealed ⇒ multi-stage stochastic programming, adjustable robust optimization

Why Robust Optimization (RO)

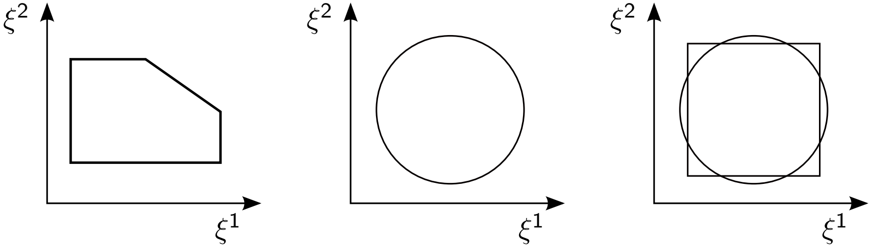
Stochastic programming suffers from two drawbacks:

- Precise information about the distribution is required.
- The resulting optimization problems are usually very large-scale (and intractable).

Robust Optimization (and distributionally RO):

- Require less information about how parameters vary.
- Leads to tractable optimization problems (less true for distributionally RO).

Uncertainty sets



(a) $\|\xi\|_\infty \leq 1, \|\xi\|_1 \leq \kappa_1$ (b) $\|\xi\|_2 \leq \kappa_2$ (c) $\|\xi\|_\infty \leq 1, \|\xi\|_2 \leq \kappa_2$

Outline

Consider a static robust linear optimization problem

$$
\begin{aligned}
\min \quad & c^T \mathbf{x} \\
\text{s.t.} \quad & \mathbf{x} \in \mathcal{X} \\
& A(\xi)\mathbf{x} \le b \quad \xi \in \Xi.
\end{aligned} \tag{1}
$$

with the following notation

ξ : uncertain parameter
$\Xi \subset \mathbb{R}^K$: uncertainty polytope
$\mathcal{X} \subset \mathbb{R}^I$: intersection of a polyhedron with integer restrictions
A : affine function of ξ, i.e., $A(\xi) := A^0 + \sum_k A^{1k}\xi^k$.

Consider a static robust linear optimization problem

$$\begin{aligned} \min \quad & c^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{X} \\ & A(\xi)\mathbf{x} \le b \quad \xi \in \Xi. \end{aligned} \tag{1}$$

with the following notation

ξ	:	uncertain parameter
$\Xi \subset \mathbb{R}^K$	:	uncertainty polytope
$\mathcal{X} \subset \mathbb{R}^I$	:	intersection of a polyhedron with integer restrictions
A	:	affine function of ξ, i.e., $A(\xi) := A^0 + \sum_k A^{1k}\xi^k$.

Can be solved in two ways:

Cutting-planes Let $\Xi^* \subset \Xi$. We relax (1) to $A(\xi)x \le b \quad \xi \in \Xi^*$. We generate additional constraints on demand by solving $\max_{\xi \in \Xi} A_i(\xi)x$ for each constraint i.

Consider a static robust linear optimization problem

$$
\begin{aligned}
\min \quad & c^T \mathbf{x} \\
\text{s.t.} \quad & \mathbf{x} \in \mathcal{X} \\
& A(\xi)\mathbf{x} \le b \quad \xi \in \Xi.
\end{aligned} \qquad (1)
$$

with the following notation

ξ	:	uncertain parameter
$\Xi \subset \mathbb{R}^K$	:	uncertainty polytope
$\mathcal{X} \subset \mathbb{R}^I$	:	intersection of a polyhedron with integer restrictions
A	:	affine function of ξ, i.e., $A(\xi) := A^0 + \sum_k A^{1k}\xi^k$.

Can be solved in two ways:

Cutting-planes Let $\Xi^* \subset \Xi$. We relax (1) to $A(\xi)x \le b \quad \xi \in \Xi^*$. We generate additional constraints on demand by solving $\max_{\xi \in \Xi} A_i(\xi)x$ for each constraint i.

Dualization Using duality in linear programming, we can reformulate (1) as a polynomial number of linear constraints.

Dualization

Theorem

Let $\Xi := \left\{ \xi \in \mathbb{R}^K : \xi \geq 0, \sum_{k \in K} e_j^k \xi^k \leq d_j,\ j \in J \right\}$. *A vector* $\mathbf{x}$ *is feasible for*

$$a^T(\xi)\mathbf{x} \leq b(\xi), \quad \xi \in \Xi$$

if and only if there exists dual variables $\mathbf{z} \in \mathbb{R}^J$ *such that* $\mathbf{x}$ *is feasible for the system of constraints*

$$\sum_{i \in I} a_i^0 \mathbf{x_i} + \sum_{j \in J} d_j \mathbf{z_j} \leq b^0$$

$$\sum_{j \in J} e_j^k \mathbf{z_j} \geq \sum_{i \in I} a_i^k \mathbf{x_i} - b^k, \quad k \in K$$

$$\mathbf{z} \geq 0.$$

Proof.

$$
\begin{aligned}
a^T(\xi)x \le b(\xi), \quad \xi \in \Xi \quad &\Leftrightarrow \quad \sum_{i\in I}(a_i^0 + \sum_{k\in K} a_i^k \xi^k)x_i \le b^0 + \sum_{k\in K} b^k \xi^k, \quad \xi \in \Xi \\
&\Leftrightarrow \quad \sum_{i\in I} a_i^0 x_i + \max_{\xi\in\Xi} \sum_{k\in K}(\sum_{i\in I} a_i^k x_i - b^k)\xi^k \le b^0 \\
&\Leftrightarrow \quad \sum_{i\in I} a_i^0 x_i + \min_{z\ge 0} \sum_{j\in J} d_j z_j \le b^0 \\
&\qquad\qquad \text{s.t.} \sum_{j\in J} e_j^k z_j \ge \sum_{i\in I} a_i^k x_i - b^k, \quad k \in K \\
&\Leftrightarrow \quad \sum_{i\in I} a_i^0 x_i + \sum_{j\in J} d_j z_j \le b^0 \\
&\qquad \sum_{j\in J} e_j^k z_j \ge \sum_{i\in I} a_i^k x_i - b^k, \quad k \in K \\
&\qquad z \ge 0.
\end{aligned}
$$

Which approach is best ?

We refer to the recent paper:

Dimitris Bertsimas, Iain Dunning, and Miles Lubin: "Reformulations versus cutting planes for robust optimization: A computational and machine learning perspective". Available at Optimization Online.

	Polyhedral	Ellipsoidal
Reformulation	1.56	**1.43**
Cutting-plane	**1.49**	1.67

Table : Results for linear programs.

	Polyhedral	Ellipsoidal
1. Reformulation	**1.66**	2.24
2. Cutting-plane	2.96	2.65
3. Cutting-plane & root	1.74	**1.74**
4. Cutting-plane & heur.	3.02	2.66
5. Cutting-plane & root & heur.	1.82	1.79

Table : Results for mixed-integer linear programs.

Adjustable situation

$$
\begin{aligned}
& \min && c^T \mathbf{x} \\
& \text{s.t.} && \mathbf{x} \in \mathcal{X} \\
(P) \quad & && A(\xi)\mathbf{x} + E(\xi)\mathbf{y}(\xi) \leq b \quad \xi \in \Xi.
\end{aligned}
$$

where y represents the vector of adjustable variables.

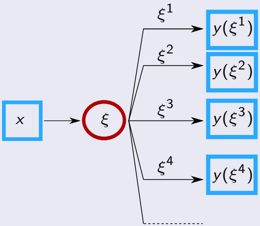
Adjustable situation

$$
\begin{aligned}
& \min \quad c^T \mathbf{x} \\
& \text{s.t.} \quad \mathbf{x} \in \mathcal{X} \\
(P) \quad & \quad A(\xi)\mathbf{x} + E(\xi)\mathbf{y}(\xi) \leq b \quad \xi \in \Xi.
\end{aligned}
$$

where y represents the vector of adjustable variables.

Remark

We consider 2-stage problems only: network design problems, facility location problems, ...



Remark

We suppose that all recourse variables y are continuous. Very recent works from Bertsimas et al. (2014) and Hanasusanto et al. (2014) consider the case of integer recourse (both available at Optimization Online):

- *Dimitris Bertsimas and Angelos Georghiou "Design of Near Optimal Decision Rules in Multistage Adaptive Mixed-Integer Optimization".*
- *Grani A. Hanasusanto, Daniel Kuhn, and Wolfram Wiesemann "Two-Stage Robust Integer Programming"*

Remark

We suppose that all recourse variables y are continuous. Very recent works from Bertsimas et al. (2014) and Hanasusanto et al. (2014) consider the case of integer recourse (both available at Optimization Online):

- *Dimitris Bertsimas and Angelos Georghiou "Design of Near Optimal Decision Rules in Multistage Adaptive Mixed-Integer Optimization".*
- *Grani A. Hanasusanto, Daniel Kuhn, and Wolfram Wiesemann "Two-Stage Robust Integer Programming"*

We present next approaches for two types of problems:

- Exact solution algorithms for (P) when E is constant.
- Solution algorithms based on affine decision rules when E depends affinely on ξ.

Affine decision rules

A classical approach for (P) is to restrict y to affine functions of ξ:

$$\mathbf{y}(\xi) = \mathbf{y^0} + \sum_{k \in K} \mathbf{y^k} \xi^k.$$

when E is constant, (P) becomes

$$\begin{aligned} \min \quad & c^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{X} \\ & \left(A^0 + \sum_k A^{1k} \xi^k\right) \mathbf{x} + E\left(\mathbf{y^0} + \sum_{k \in K} \mathbf{y^k} \xi^k\right) \leq b \quad \xi \in \Xi. \end{aligned}$$

Disadvantage Provides a conservative solution. When Ξ is a simplex, the conservative solution is optimal.

Advantage The resulting optimization problems have the same structure as static RO problems.

Affine decision rules

Theorem

Let Ξ be an uncertainty polytope, A be an affine function of ξ, $\mathbf{x}$ be a vector of optimization variables, and

$$\mathbf{y}(\xi) = \mathbf{y^0} + \sum_{k \in K} \mathbf{y^k} \xi^k$$

be a vector of affine adjustable optimization variables. Then, robust constraint

$$A(\xi)\mathbf{x} + E(\xi)\mathbf{y}(\xi) \leq b \quad \xi \in \Xi$$

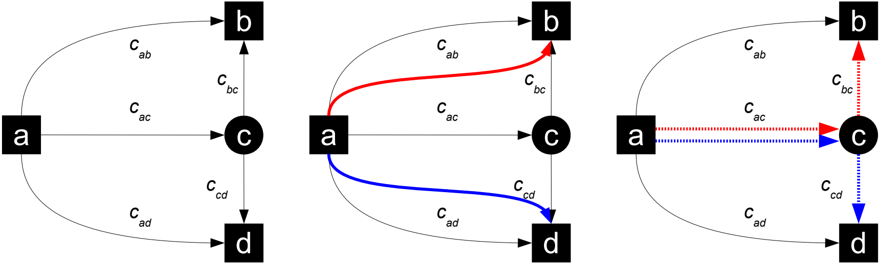
is equivalent to

$$\hat{A}(\xi)\hat{\mathbf{x}} \leq b \quad \xi \in \Xi,$$

where $\hat{A}$ is an affine function and $\hat{\mathbf{x}}$ is a vector of optimization variables.

Outline

Classical network design problem



Commodities are routed according to shortest paths. In the example, we assume $c_{ac} + c_{bc} \leq c_{ab}$ and $c_{ac} + c_{cd} \leq c_{ad}$.
Solution cost: $\xi^{ab}(c_{ac} + c_{bc}) + \xi^{ad}(c_{ac} + c_{cd})$.

Robust network design problem: **dynamic** recourse/routing

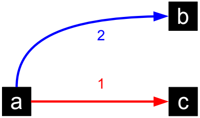
Demands vectors $\xi_1, \ldots, \xi_n$ must be routed **non-simultaneously**.
The problem becomes a two-stages program:

1. decide of the capacities
2. decide of the routing.

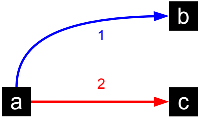
Robust network design problem: **dynamic** recourse/routing

Demands vectors $\xi_1, \ldots, \xi_n$ must be routed **non-simultaneously**.
The problem becomes a two-stages program:

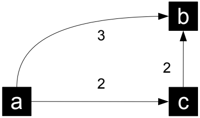
1. decide of the capacities
2. decide of the routing.



Demands for scenario 1



Demands for scenario 2



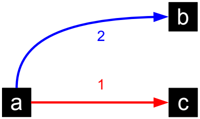
Capacity cost per unit

Robust network design problem: **dynamic** recourse/routing

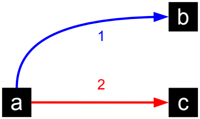
Demands vectors $\xi_1, \dots, \xi_n$ must be routed **non-simultaneously**.

The problem becomes a two-stages program:

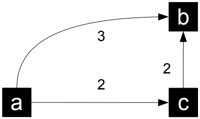
1. decide of the capacities
2. decide of the routing.



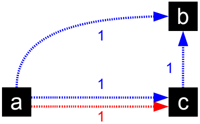
Demands for scenario 1



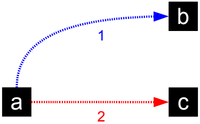
Demands for scenario 2



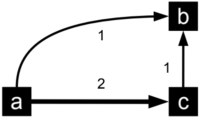
Capacity cost per unit



Routing for scenario 1



Routing for scenario 2



Capacity installation

Robust network design problem: **static** recourse/routing

Allowing for dynamic/arbitrary recourse may not be a good idea:

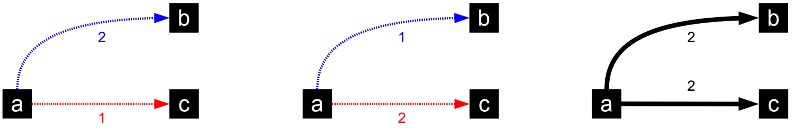
- Yield NP-hard optimization problems, see Chekuri et al. (2007), Gupta et al. (2001), Mattia (2010).
- Unpractical to change completely the routing according to the demand variations.

Robust network design problem: **static** recourse/routing

Allowing for dynamic/arbitrary recourse may not be a good idea:

- Yield NP-hard optimization problems, see Chekuri et al. (2007), Gupta et al. (2001), Mattia (2010).
- Unpractical to change completely the routing according to the demand variations.

$\Rightarrow$ Introduction of static routing described by routing templates.



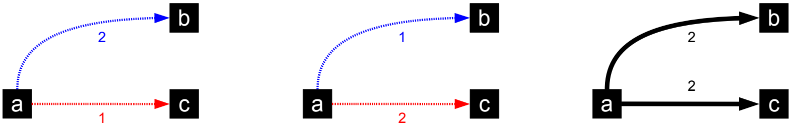
Routing for scenario 1 | Routing for scenario 2 | Capacity installation

Robust network design problem: **static** recourse/routing

Allowing for dynamic/arbitrary recourse may not be a good idea:

- Yield NP-hard optimization problems, see Chekuri et al. (2007), Gupta et al. (2001), Mattia (2010).
- Unpractical to change completely the routing according to the demand variations.

⇒ Introduction of static routing described by routing templates.



Routing for scenario 1 | Routing for scenario 2 | Capacity installation

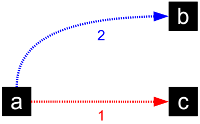
$$\text{Cost\_Static} = 10 > 9 = \text{Cost\_Dynamic}$$

Robust network design problem: **static** recourse/routing

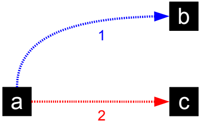
Allowing for dynamic/arbitrary recourse may not be a good idea:

- Yield NP-hard optimization problems, see Chekuri et al. (2007), Gupta et al. (2001), Mattia (2010).
- Unpractical to change completely the routing according to the demand variations.

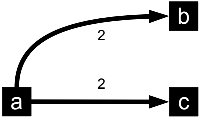
⇒ Introduction of static routing described by routing templates.



Routing for scenario 1



Routing for scenario 2



Capacity installation

$$\text{Cost\_Static} = 10 > 9 = \text{Cost\_Dynamic}$$

See Ben-Ameur (2007) and Scutellà (2009,2010) for intermediary (NP-hard) routing frameworks.

Using the parlance of robust optimization

First stage: capacity variables $\mathbf{x}$

Second stage: flow variables $\mathbf{y}$

Using the parlance of robust optimization

First stage: capacity variables $\mathbf{x}$

Second stage: flow variables $\mathbf{y}$

$$\min \sum_{a \in A} \kappa_a \mathbf{x_a}$$

$$\text{s.t.} \sum_{a \in \delta^+(v)} \mathbf{y_a^k}(\xi) - \sum_{a \in \delta^-(v)} \mathbf{y_a^k}(\xi) = \begin{cases} -\xi^k & v = s(k) \\ \xi^k & v = t(k), \\ 0 & \text{else} \end{cases} \qquad v \in V, k \in K, \ \xi \in \Xi$$

$$\sum_{k \in K} \mathbf{y_a^k}(\xi) \leq \mathbf{x_a}, \qquad a \in A, \xi \in \Xi$$

$$\mathbf{y_a^k}(\xi) \geq 0, \qquad a \in A, k \in K, \ \xi \in \Xi$$

$$\mathbf{x_a} \geq 0, \qquad a \in A,$$

Routing function

Dynamic routing: flow function $y^k : \Xi \to \mathbb{R}^{A \times K}$ is **arbitrary**.

Routing function

Static routing: flow functions f^k are **independent linear** functions for each commodity:

$$y_a^k(\xi) := \mathbf{y_a^k}\xi^k, \qquad a \in A,\ k \in K$$

for some routing template vector $y^k \in \mathbb{R}^A$.

Dynamic routing: flow function $y^k : \Xi \to \mathbb{R}^{A\times K}$ is **arbitrary**.

Routing function

Static routing: flow functions f^k are **independent linear** functions for each commodity:

$$y_a^k(\xi) := \mathbf{y_a^k}\xi^k, \qquad a \in A,\ k \in K$$

for some routing template vector $y^k \in \mathbb{R}^A$.

Affine routing: flow functions y^k are **affine** functions:

$$y_a^k(\xi) := \mathbf{y_a^{0k}} + \sum_{h \in K} \mathbf{y_a^{kh}}\xi^h, \qquad a \in A,\ k \in K$$

Dynamic routing: flow function $y^k : \Xi \to \mathbb{R}^{A \times K}$ is **arbitrary**.

Outline

Exact solution procedure

$$
\begin{aligned}
&\min \quad c^T \mathbf{x} \\
&\text{s.t.} \quad \mathbf{x} \in \mathcal{X} \\
(P) \qquad & A(\xi)\mathbf{x} + E\mathbf{y}(\xi) \leq b \quad \xi \in \Xi. \qquad (2)
\end{aligned}
$$

Let $\mathcal{K}(\Xi)$ be the set defined by (2).

Lemma

It holds that $\mathcal{K}(\Xi) = \mathcal{K}(\text{vert}(\Xi))$.

Exact solution procedure

$$
\begin{aligned}
(P) \qquad \min \quad & c^T \mathbf{x} \\
\text{s.t.} \quad & \mathbf{x} \in \mathcal{X} \\
& A(\xi)\mathbf{x} + E\mathbf{y}(\xi) \leq b \quad \xi \in \Xi. \qquad (2)
\end{aligned}
$$

Let $\mathcal{K}(\Xi)$ be the set defined by (2).

Lemma

It holds that $\mathcal{K}(\Xi) = \mathcal{K}(\text{vert}(\Xi))$.

Idea of the proof:

$$A(\xi^*)x^* + Ey(\xi^*) \leq b \Leftrightarrow \sum_{s=1}^{\text{vert}(\Xi)} \lambda_s \left(A(\xi_s)x^* + Ey(\xi_s)\right) \leq \sum_{s=1}^{\text{vert}(\Xi)} \lambda_s b.$$

Finite reformulation

$$\begin{aligned} \min \quad & c^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{X} \\ & A(\xi)\mathbf{x} + E\mathbf{y}(\xi) \leq b \quad \xi \in \text{vert}(\Xi). \end{aligned}$$

We shall consider an initial subset Ξ^* of Ξ and generate additional scenarios during decomposition algorithms:

Master problem

$$(MP') \quad \begin{aligned} \min \quad & c^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{X}. \\ & \text{Constraints corresponding to } \xi \in \Xi^* \end{aligned}$$

Separation

Using the Farkas' lemma, we obtain:

Theorem

Let $x^* \in \mathbb{R}^n$ be given. Vector x^* belongs to $\mathcal{K}(\Xi)$ if and only if the optimal solution of the following optimization problem is non-positive

$$
(SP) \quad \begin{array}{ll} \max & (b - A(\xi)x^*)^T \pi \\ s.t. & \xi \in \Xi \\ & E^T \pi = 0 \\ & \mathbf{1}^T \pi = 1 \\ & \pi \geq 0. \end{array}
$$

Bilinear optimization problems are very difficult to solve. Mattia (2013) proposed a MILP reformulation based on complementary slackness conditions $\Rightarrow$ still very hard to solve.

We restrict ourselves to polytopes that can be obtained as affine projection of $(0,1)$ polytopes (polytopes whose extreme points are binary vectors).

Assumption

There exists a $(0,1)$ polytope Ω and an affine projection $\mathcal{A}$ such that $\Xi = \mathcal{A}(\Omega)$.

We restrict ourselves to polytopes that can be obtained as affine projection of $(0,1)$ polytopes (polytopes whose extreme points are binary vectors).

Assumption

There exists a $(0,1)$ polytope Ω and an affine projection $\mathcal{A}$ such that $\Xi = \mathcal{A}(\Omega)$.

Example

Let $\text{vert}(\Xi) = \{\xi_1, \ldots, \xi_s\}$ and let Ω be the unit simplex in $\mathbb{R}^s$. Then, for any $\omega \in \Omega$,

$$\mathcal{A}(\omega) = \left(\sum_{j=1}^{s} \xi_j^1 \omega_j, \ldots, \sum_{j=1}^{s} \xi_j^K \omega_j \right)$$

We restrict ourselves to polytopes that can be obtained as affine projection of $(0,1)$ polytopes (polytopes whose extreme points are binary vectors).

Assumption

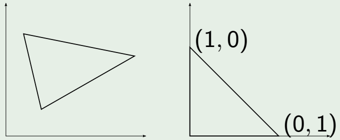
There exists a $(0,1)$ polytope Ω and an affine projection $\mathcal{A}$ such that $\Xi = \mathcal{A}(\Omega)$.

Example

Let $\text{vert}(\Xi) = \{\xi_1, \ldots, \xi_s\}$ and let Ω be the unit simplex in $\mathbb{R}^s$. Then, for any $\omega \in \Omega$,

$$\mathcal{A}(\omega) = \left(\sum_{j=1}^{s} \xi_j^1 \omega_j, \ldots, \sum_{j=1}^{s} \xi_j^K \omega_j \right)$$

Example



(a) Ξ (b) Ω

Figure : Example of polytope Ξ affinely equivalent to a $(0,1)$ polytope.

Separation 2

Theorem

Let $x^* \in \mathbb{R}^n$ be given. Vector x^* belongs to $\mathcal{K}(\Xi)$ if and only if the optimal solution of the following optimization problem is non-positive

$$
\begin{aligned}
& & \max \quad & (b - A^0 x^*)^T \pi - \sum_{k \in K} (A^{1k} x^*)^T \mathbf{v^k} & \\
&(SPL) & s.t. \quad & \xi \in \Xi & \\
& & & E^T \pi = 0 & \\
& & & \mathbf{1}^T \pi = 1 & \\
& & & \mathbf{v_m^k} \geq \pi_{\mathbf{m}} - (1 - \xi^{\mathbf{k}}) & k \in K, m \in M \\
& & & \mathbf{v_m^k} \leq \xi^{\mathbf{k}} & k \in K, m \in M \\
& & & \pi, \mathbf{v_m^k} \geq 0, & \\
& & & \xi \in \{0, 1\}^K. &
\end{aligned}
$$

Two different approaches

Benders
$$(b - A(\xi^*)x)^T \pi^* \leq 0. \quad (3)$$

Row and column generation
$$A(\xi^*)x + Ey(\xi^*) \leq b. \quad (4)$$

Two different approaches

Benders	$(b - A(\xi^*)x)^T \pi^* \leq 0.$	(3)
Row and column generation	$A(\xi^*)x + Ey(\xi^*) \leq b.$	(4)

Algorithm 2: *RG* and *RCG*

repeat
- solve (MP');
- let x^* be an optimal solution;
- solve (SPL);
- let (ξ^*, π^*) be an optimal solution and z^* be the optimal solution cost;
- **if** $z^* > 0$ **then**
  - *RG*: add constraint (3) to (MP');
  - *RCG*: add constraint (4) to (MP');

until $z^* > 0$;

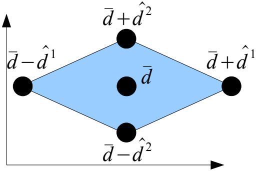
Application to network design

- SNDlib networks: janos-us, sun, and giul39

Application to network design

- SNDlib networks: janos-us, sun, and giul39
- Two kinds of polytopes:

$$\Xi = \begin{cases} \xi^k \in [\bar{\xi}^k - \sigma_-^k \hat{\xi}^k, \bar{\xi}^k + \sigma_+^k \hat{\xi}^k] \text{ for all } k \in K \\ \sum_{k \in K} \sigma_-^k + \sigma_+^k \leq \Gamma \end{cases}$$



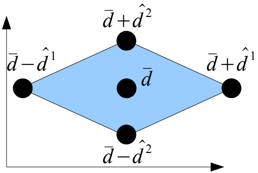
Ξ for $\Gamma = 1$

Details

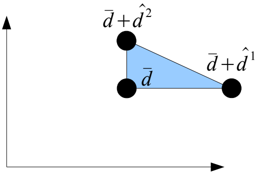
- SNDlib networks: janos-us, sun, and giul39
- Two kinds of polytopes:

$$\Xi = \left\{ \begin{array}{l} \xi^k \in [\bar{\xi}^k - \sigma_-^k \hat{\xi}^k, \bar{\xi}^k + \sigma_+^k \hat{\xi}^k] \text{ for all } k \in K \\ \sum_{k \in K} \sigma_-^k + \sigma_+^k \leq \Gamma \end{array} \right.$$

$$\Xi^+ = \left\{ \begin{array}{l} \xi^k \in [\bar{\xi}^k, \bar{\xi}^k + \sigma_+^k \hat{\xi}^k] \text{ for all } k \in K \\ \sum_{k \in K} \sigma_+^k \leq \Gamma \end{array} \right.$$



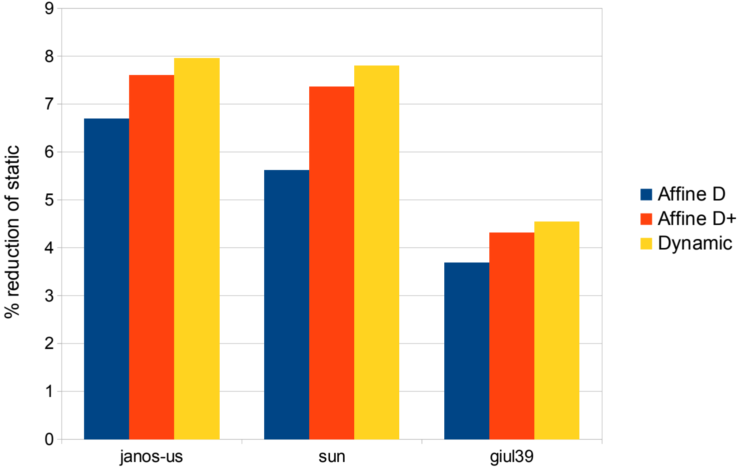
Ξ for $\Gamma = 1$



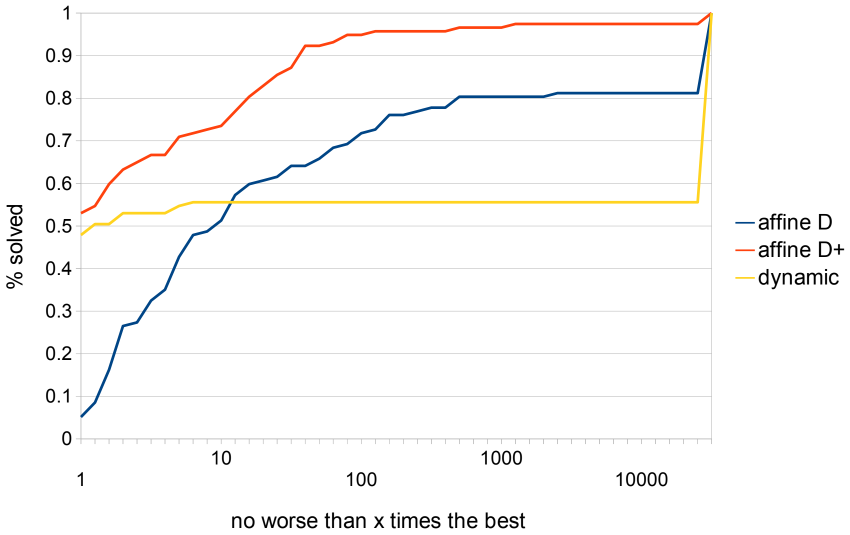
Ξ^+ for $\Gamma = 1$

Average cost reduction

$$|K| \in \{10, 20, 30, 40, 50\} \qquad \Gamma \in \{1, 2, 3, 4, 5, 6\}$$



Solution times: Performance profile



Numerical results

K	Γ	opt_{stat}	$gap_{dyn}(\%)$	t_{RCG}	t_{SPL} (%)	iter	t_{RG}	$t_{P'}$
30	2	672665	8.7	150	64	18	4967	13
30	3	699279	7.0	301	78	19	**T**	213
30	4	699590	2.9	1500	90	27	**T**	**M**
30	5	699590	0.7	1344	91	25	**T**	**M**
40	2	732850	8.8	365	69	21	6523	49
40	3	763505	7.6	1037	88	22	**T**	**M**
40	4	766293	4.1	6879	96	30	**T**	**M**
40	5	766293	1.5	5866	95	31	**T**	**M**
40	6	766293	–	**T**	–	–	**T**	**M**
50	2	793295	8.9	694	73	23	**T**	98
50	3	827405	8.2	4446	94	27	**T**	**M**
50	4	839656	6.0	22645	98	35	**T**	**M**
50	5	841295	–	**T**	–	–	**T**	**M**
50	6	841295	–	**T**	–	–	**T**	**M**

Table : Results for janos-us.

Attention: not applicable to multi-stage problems

ξ^t Information available at period t (i.e. $(\xi_1, \ldots, \xi_{t-1})$)

y_t Decision at period t

$$
\begin{aligned}
\min \quad & c^T \mathbf{x} \\
\text{s.t.} \quad & \mathbf{x} \in \mathcal{X} \\
& A(\xi)\mathbf{x} + E\mathbf{y}(\xi) \le b \quad \xi \in \Xi \\
& y_t(\xi) = y_t(\xi') \quad \text{for all } \xi, \xi' \in \Xi \text{ s.t. } \xi^t = \xi'^t
\end{aligned} \tag{5}
$$

Constraints (5) prevents us from using Farkas' Lemma as before!

Outline

Random recourse: general case

$$\begin{aligned} \min \quad & c^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{X} \\ & A(\xi)\mathbf{x} + E(\xi)y(\xi) \le b, \quad \xi \in \Xi, \end{aligned} \tag{6}$$

where $E(\xi)$ and $y(\xi)$ are affine functions of ξ:

$$\begin{aligned} \min \quad & c^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{X} \\ & \left(A^0 + \sum_k A^{1k}\xi^k\right)\mathbf{x} + \left(E^0 + \sum_k E^{1k}\xi^k\right)\left(\mathbf{y^0} + \sum_{k\in K} \mathbf{y^k}\xi^k\right) \le b, \quad \xi \in \Xi \end{aligned}$$

Example

Random recourse cost

$$r^T(\xi)y(\xi) \le \theta, \quad \xi \in \Xi.$$

Random recourse: general case

$$\begin{aligned} \min \quad & c^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{X} \\ & A(\xi)\mathbf{x} + E(\xi)y(\xi) \leq b, \quad \xi \in \Xi, \end{aligned} \tag{6}$$

where $E(\xi)$ and $y(\xi)$ are affine functions of ξ:

$$\begin{aligned} \min \quad & c^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{X} \\ & \left(A^0 + \sum_k A^{1k} \xi^k\right)\mathbf{x} + \left(E^0 + \sum_k E^{1k} \xi^k\right)\left(\mathbf{y^0} + \sum_{k \in K} \mathbf{y^k} \xi^k\right) \leq b, \quad \xi \in \Xi \end{aligned}$$

Example

Random recourse cost

$$r^T(\xi)y(\xi) \leq \theta, \quad \xi \in \Xi.$$

"Easy" case: Ξ is an ellipsoid

$$
\begin{aligned}
& \left(A^0 + \sum_k A^{1k}\xi^k\right)\mathbf{x} + \left(E^0 + \sum_k E^{1k}\xi^k\right)\left(\mathbf{y^0} + \sum_{k\in K}\mathbf{y^k}\xi^k\right) \le b, && \xi \in \Xi \\
\Leftrightarrow \quad & 0 \le \alpha(\mathbf{x},\mathbf{y}) + 2\xi^T\beta(\mathbf{x},\mathbf{y}) + \xi^T\Gamma(\mathbf{x},\mathbf{y})\xi, && \xi \in \Xi
\end{aligned} \tag{7}
$$

"Easy" case: Ξ is an ellipsoid

$$\left(A^0 + \sum_k A^{1k}\xi^k\right)\mathbf{x} + \left(E^0 + \sum_k E^{1k}\xi^k\right)\left(\mathbf{y^0} + \sum_{k\in K}\mathbf{y^k}\xi^k\right) \le b, \quad \xi \in \Xi$$

$$\Leftrightarrow \quad 0 \le \alpha(\mathbf{x},\mathbf{y}) + 2\xi^T\beta(\mathbf{x},\mathbf{y}) + \xi^T\Gamma(\mathbf{x},\mathbf{y})\xi, \qquad \xi \in \Xi \tag{7}$$

Theorem (Ben-Tal et al.)

Let $\Xi = \{\|\xi\|_2 \le \kappa_2\}$. *Constraint* (7) *is equivalent to*

$$\begin{pmatrix} \Gamma(\mathbf{x},\mathbf{y}) + \kappa_2^{-2}\mathbf{v}\,\mathrm{Id} & \beta(\mathbf{x},\mathbf{y}) \\ \beta^T(\mathbf{x},\mathbf{y}) & \alpha(\mathbf{x},\mathbf{y}) - \mathbf{v} \end{pmatrix} \succeq 0$$

$$\mathbf{v} \ge 0,$$

where Id *is the* $|K| \times |K|$ *identity matrix.*

Conic reformulation

Ellipsoidal uncertainty Semi-definite programming (SDP) reformulation

Other uncertainty No exact reformulation, SDP can provide an upper bound.

Conic reformulation

Ellipsoidal uncertainty Semi-definite programming (SDP) reformulation

Other uncertainty No exact reformulation, SDP can provide an upper bound.

$\Rightarrow$ Our objective is to address with decomposition algorithms for problems that satisfy the following assumption:

Assumption

$$A(\xi)x + E(\xi)y(\xi) \leq b \Rightarrow \begin{array}{l} r(\xi)g^T y(\xi) \leq \theta \\ A(\xi)x + E'y(\xi) \leq b \end{array}$$

Robust network design problem

Classical problem:

Input

- Directed graph $G = (V, A)$
- Capacity cost c
- Set of point-to-point commodities K
- Demand $d^k = \bar{\xi}^k + \xi^k \hat{\xi}^k$
- Uncertainty set Ξ
- *Outsourcing costs* r

Output

- Network capacities $\mathbf{x}$
- Flow (routing templates) $\mathbf{f}$
- *Outsourced fraction* $\mathbf{g}$

Robust network design problem

Classical problem:

Input

- Directed graph $G = (V, A)$
- Capacity cost c
- Set of point-to-point commodities K
- Demand $d^k = \bar{\xi}^k + \xi^k \hat{\xi}^k$
- Uncertainty set Ξ
- *Outsourcing costs* r

Output

- Network capacities $\mathbf{x}$
- Flow (routing templates) $\mathbf{f}$
- *Outsourced fraction* $\mathbf{g}$
- $\mathbf{g^k} = 1 - \sum_{a \in \delta^-(t(k))} \mathbf{y_a^k} + \sum_{a \in \delta^+(t(k))} \mathbf{y_a^k}$

Robust network design problem

Classical problem:

Input

- Directed graph $G = (V, A)$
- Capacity cost c
- Set of point-to-point commodities K
- Demand $d^k = \bar{\xi}^k + \xi^k \hat{\xi}^k$
- Uncertainty set Ξ
- *Outsourcing costs r*

Output

- Network capacities $\mathbf{x}$
- Flow (routing templates) $\mathbf{f}$
- *Outsourced fraction* $\mathbf{g}$
- $\mathbf{g^k} = 1 - \sum_{a \in \delta^-(t(k))} \mathbf{y^k_a} + \sum_{a \in \delta^+(t(k))} \mathbf{y^k_a}$

We consider uncertain outsourcing cost $r(\xi)$

design variables	$\mathbf{x} \geq 0$
flow variables	$\mathbf{f} \geq 0$
outsourcing variables	$\mathbf{g} \geq 0$
outsourcing cost	$\theta \geq 0$

$$\min \quad \sum_{a \in A} \kappa_a \mathbf{x_a} + \theta$$

$$\text{s.t.} \quad \theta \geq r(\xi) \sum_{k \in K} d^k(\xi) \mathbf{g^k} \qquad \xi \in \Xi \qquad \mathit{COST}$$

$$\sum_{a \in \delta^-(v)} \mathbf{y_a^k} - \sum_{a \in \delta^+(v)} \mathbf{y_a^k} = 0, \qquad k \in K, v \in V^* \qquad \mathit{FLOW}$$

$$\sum_{k \in K} d^k(\xi) \mathbf{y_a^k} \leq \mathbf{x_a} \qquad a \in A, \xi \in \Xi \qquad \mathit{CAPACITY}$$

$$\mathbf{g^k} = 1 - \sum_{a \in \delta^-(t(k))} \mathbf{y_a^k} + \sum_{a \in \delta^+(t(k))} \mathbf{y_a^k} \qquad k \in K \qquad \mathit{REJECTION}$$

Master problem

- Ξ_0^* and Ξ_a^* are finite subsets of Ξ
- Relax capacity and cost constraints outside these subsets

$$
\begin{aligned}
& \min && \sum_{a \in A} \kappa_a \mathbf{x_a} + \theta && \\
(MP) \quad & \text{s.t.} && \theta \geq r(\xi) \sum_{k \in K} d^k(\xi) \mathbf{g^k} && \xi \in \Xi_0^* \\
& && \sum_{k \in K} d^k(\xi) \mathbf{y_a^k} \leq \mathbf{x_a} && a \in A, \xi \in \Xi_a^* \\
& && FLOW, REJECTION, NON-NEGATIVITY &&
\end{aligned}
$$

Separation problem

Given a solution $(\theta^*, x^*, f^*, g^*)$ to (MP), check for missing constraints:

$$\text{CAPACITY} \quad -x_a^* + \max_{\xi \in \Xi} \sum_{k \in K} d^k(\xi) f_a^{*k} \quad \Rightarrow \quad \text{OK}$$

$$\text{COST} \quad -\theta^* + \max_{\xi \in \Xi} r(\xi) \sum_{k \in K} d^k(\xi) g^{*k} \quad \Rightarrow \quad ?$$

Separation problem

Given a solution $(\theta^*, x^*, f^*, g^*)$ to (MP), check for missing constraints:

$$\text{CAPACITY} \quad -x_a^* + \max_{\xi\in\Xi} \sum_{k\in K} d^k(\xi) f_a^{*k} \quad \Rightarrow \quad \text{OK}$$

$$\text{COST} \quad -\theta^* + \max_{\xi\in\Xi} r(\xi) \sum_{k\in K} d^k(\xi) g^{*k} \quad \Rightarrow \quad ?$$

$$\sim -\theta^* + \max_{\xi\in\Xi} \ln\left(r(\xi)\right) + \ln\left(\sum_{k\in K} d^k(\xi) g^{*k}\right)$$

Separation problem

Given a solution $(\theta^*, x^*, f^*, g^*)$ to (MP), check for missing constraints:

$$\text{CAPACITY} \quad -x_a^* + \max_{\xi \in \Xi} \sum_{k \in K} d^k(\xi) f_a^{*k} \quad \Rightarrow \quad \text{OK}$$

$$\text{COST} \quad -\theta^* + \max_{\xi \in \Xi} r(\xi) \sum_{k \in K} d^k(\xi) g^{*k} \quad \Rightarrow \quad ?$$

$$\sim -\theta^* + \max_{\xi \in \Xi} \ln \left(r(\xi) \right) + \ln \left(\sum_{k \in K} d^k(\xi) g^{*k} \right)$$

We solve COST by a cutting-plane algorithm (embedded into another cutting plane algorithm)

We compare:

- *CP* Cut generation for cost and capacity constraints
- *CP+D* Dualization of capacity constraint and cut generation for cost constraints
- *SDP* Semi-definite reformulation in the ellipsoidal case

Instances:

	$\|V\|$	$\|A\|$	$\|K\|$
di-yuan	11	84	22
pdh	11	68	24
polska	12	36	66
nobel-us	14	42	91
atlanta	15	44	210
newyork	16	98	240
france	25	90	300
india35	35	160	595
germany50	50	176	662
cost266	37	114	1332

Solution times

Instances	Budgeted		Ellipsoid + box		Ellipsoid		
	CP	*CP+D*	*CP*	*CP+D*	*CP*	*CP+D*	*SDP*
di-yuan	0.1	0.1	0.1	6.1	0.2	10.8	17.3
pdh	0.1	0.1	1.4	114	0.1	4.4	10.9
polska	0.1	0.1	0.4	49.7	0.5	10.9	68.9
nobel-us	0.1	0.2	0.8	193	0.3	12	276
atlanta	1.4	4.1	4	260	4.4	75.8	**T**
newyork	0.9	1.1	2.9	**T**	10.1	450	**M**
France	2.9	4.2	74.3	**T**	18.8	271	**M**
india35	20.9	12.6	76.4	**T**	571	1750	**M**
germany	5.8	13.9	8.7	**T**	138	**T**	**M**
cost266	154	61.8	1480	**T**	**T**	**T**	**M**

Table : Solution times for $\epsilon = 0.05$.

Solution costs

Instances	Best	(*RND*)			(RND^0)		
		Bud.	Ell.+Box	Ell.	Bud.	Ell.+Box	Ell.
di-yuan	4.74+06	0.8	**0**	12.8	11.6	11.6	36.5
pdh	2.45+07	**0**	1.3	16.6	2.2	2.2	45.6
polska	3.74+02	1.0	**0**	7.2	21.4	18.9	23.4
nobel-us	4.82+06	3.0	**0**	7.1	33.7	30.3	35.3
atlanta	1.86+08	12.4	**0**	16.6	50.4	38.7	52.5
newyork	2.67+05	5.0	**0**	9.3	38.6	34.0	39.7
France	2.14+04	5.3	**0**	5.3	31.4	19.8	22.2
india35	3.43+03	2.2	**0**	2.8	39.2	22.2	22.9
germany	6.19+05	7.6	**0**	7.3	41.3	26.9	32.7
cost266	1.39+07	8.8	**0**	*N.A.*	50.5	24.5	24.7

Table : Solution costs for $\epsilon = 0.05$.

Thank's for your attention