

# Robust Optimization: Static Case

V. Leclère (ENPC)

November 24, 2023

# Contents

- 1 Introduction and motivations
  - How to add uncertainty in an optimization problem
  - Why shall you do Robust Optimization ?
- 2 Solving the robust optimization problem
- 3 Robust optimization for Linear Programm
  - Reformulating the problem
  - Ellipsoidal uncertainty set
  - Polyhedral uncertainty set
  - Cardinality constrained LP
- 4 Probability guarantee
- 5 Robust Combinatorial Problem
- 6 Conclusion

# Contents

- 1 Introduction and motivations
  - How to add uncertainty in an optimization problem
  - Why shall you do Robust Optimization ?
- 2 Solving the robust optimization problem
- 3 Robust optimization for Linear Programm
  - Reformulating the problem
  - Ellipsoidal uncertainty set
  - Polyhedral uncertainty set
  - Cardinality constrained LP
- 4 Probability guarantee
- 5 Robust Combinatorial Problem
- 6 Conclusion

# An optimization problem

A generic optimization problem can be written

$$\begin{aligned} \min_x \quad & L(x) \\ \text{s.t.} \quad & g(x) \leq 0 \end{aligned}$$

where

- $x$  is the decision variable
- $L$  is the objective function
- $g$  is the constraint function

# An optimization problem with uncertainty

Adding uncertainty  $\xi$  in the mix

$$\begin{aligned} \min_x \quad & L(x, \tilde{\xi}) \\ \text{s.t.} \quad & g(x, \tilde{\xi}) \leq 0 \end{aligned}$$

Remarks:

- $\tilde{\xi}$  is unknown. Two main way of modelling it:
  - $\tilde{\xi} \in R$  with a known uncertainty set  $R$ , and a pessimistic approach. This is the **robust optimization** approach (RO).
  - $\tilde{\xi}$  is a random variable with known probability law. This is the **Stochastic Programming** approach (SP).
- Cost is not well defined.
  - RO:  $\max_{\xi \in R} L(x, \xi)$ .
  - SP:  $\mathbb{E}[L(x, \xi)]$ .
- Constraints are not well defined.
  - RO:  $g(x, \xi) \leq 0, \quad \forall \xi \in R$ .
  - SP:  $g(x, \xi) \leq 0, \quad \mathbb{P}\text{-a.s.}$

# An optimization problem with uncertainty

Adding uncertainty  $\xi$  in the mix

$$\begin{aligned} \min_x \quad & L(x, \tilde{\xi}) \\ \text{s.t.} \quad & g(x, \tilde{\xi}) \leq 0 \end{aligned}$$

Remarks:

- $\tilde{\xi}$  is unknown. Two main way of modelling it:
  - $\tilde{\xi} \in R$  with a known uncertainty set  $R$ , and a pessimistic approach. This is the **robust optimization** approach (RO).
  - $\tilde{\xi}$  is a random variable with known probability law. This is the **Stochastic Programming** approach (SP).
- Cost is not well defined.
  - RO:  $\max_{\xi \in R} L(x, \xi)$ .
  - SP:  $\mathbb{E}[L(x, \xi)]$ .
- Constraints are not well defined.
  - RO:  $g(x, \xi) \leq 0, \quad \forall \xi \in R$ .
  - SP:  $g(x, \xi) \leq 0, \quad \mathbb{P}\text{-a.s.}$

# An optimization problem with uncertainty

Adding uncertainty  $\xi$  in the mix

$$\begin{aligned} \min_x \quad & L(x, \tilde{\xi}) \\ \text{s.t.} \quad & g(x, \tilde{\xi}) \leq 0 \end{aligned}$$

Remarks:

- $\tilde{\xi}$  is unknown. Two main way of modelling it:
  - $\tilde{\xi} \in R$  with a known uncertainty set  $R$ , and a pessimistic approach. This is the **robust optimization** approach (RO).
  - $\tilde{\xi}$  is a random variable with known probability law. This is the **Stochastic Programming** approach (SP).
- Cost is not well defined.
  - RO:  $\max_{\xi \in R} L(x, \xi)$ .
  - SP:  $\mathbb{E}[L(x, \xi)]$ .
- Constraints are not well defined.
  - RO:  $g(x, \xi) \leq 0, \quad \forall \xi \in R$ .
  - SP:  $g(x, \xi) \leq 0, \quad \mathbb{P} - a.s.$

# Requirements and limits

- Stochastic optimization:
  - requires a law of the uncertainty  $\xi$
  - can be hard to solve (generally require discretizing the support and blowing up the dimension of the problem)
  - there exists specific methods (like Bender's decomposition)
- Robust optimization:
  - requires an uncertainty set  $R$
  - can be overly conservative, even for reasonable  $R$
  - complexity strongly depend on the choice of  $R$
- Distributionally robust optimization:
  - is a mix between robust and stochastic optimization
  - consists in solving a stochastic optimization problem where the law is chosen in a robust way
  - is a fast growing fields with multiple recent results
  - but is still hard to implement than other approaches



# Requirements and limits

- Stochastic optimization:
  - requires a law of the uncertainty  $\xi$
  - can be hard to solve (generally require discretizing the support and blowing up the dimension of the problem)
  - there exists specific methods (like Bender's decomposition)
- Robust optimization:
  - requires an uncertainty set  $R$
  - can be overly conservative, even for reasonable  $R$
  - complexity strongly depend on the choice of  $R$
- Distributionally robust optimization:
  - is a mix between robust and stochastic optimization
  - consists in solving a stochastic optimization problem where the law is chosen in a robust way
  - is a fast growing fields with multiple recent results
  - but is still hard to implement than other approaches

# Requirements and limits

- Stochastic optimization:
  - requires a law of the uncertainty  $\xi$
  - can be hard to solve (generally require discretizing the support and blowing up the dimension of the problem)
  - there exists specific methods (like Bender's decomposition)
- Robust optimization:
  - requires an uncertainty set  $R$
  - can be overly conservative, even for reasonable  $R$
  - complexity strongly depend on the choice of  $R$
- Distributionally robust optimization:
  - is a mix between robust and stochastic optimization
  - consists in solving a stochastic optimization problem where the law is chosen in a robust way
  - is a fast growing fields with multiple recent results
  - but is still hard to implement than other approaches

# Contents

- 1 Introduction and motivations
  - How to add uncertainty in an optimization problem
  - Why shall you do Robust Optimization ?
- 2 Solving the robust optimization problem
- 3 Robust optimization for Linear Programm
  - Reformulating the problem
  - Ellipsoidal uncertainty set
  - Polyhedral uncertainty set
  - Cardinality constrained LP
- 4 Probability guarantee
- 5 Robust Combinatorial Problem
- 6 Conclusion

# Some numerical tests on real-life LPs

From Ben-Tal and Nemirovski

- take LP from Netlib library
- look at non-integer coefficients, assuming that they are not known with perfect certainty
- What happens if you change them by 0.1% ?
  - constraints can be violated by up to 450%
  - $\mathbb{P}(\text{violation} > 0) = 0.5$
  - $\mathbb{P}(\text{violation} > 150\%) = 0.18$
  - $\mathbb{E}[\text{violation}] = 125\%$

# What do you want from robust optimization ?

- finding a solution that is less sensible to modified data, without a great increase of price
- choosing an uncertainty set  $R$  that:
  - offer robustness guarantee
  - yield an easily solved optimization problem

# Contents

- 1 Introduction and motivations
  - How to add uncertainty in an optimization problem
  - Why shall you do Robust Optimization ?
- 2 Solving the robust optimization problem
- 3 Robust optimization for Linear Programm
  - Reformulating the problem
  - Ellipsoidal uncertainty set
  - Polyhedral uncertainty set
  - Cardinality constrained LP
- 4 Probability guarantee
- 5 Robust Combinatorial Problem
- 6 Conclusion

# Solving a robust optimization problem

The robust optimization problem we want to solve is<sup>1</sup>

$$\begin{aligned} \min_x \quad & L(x) \\ \text{s.t.} \quad & g(x, \xi) \leq 0 \quad \forall \xi \in R \end{aligned}$$

Two main approaches are possible:

Constraint generation: replace  $R$  by a finite set of  $\xi$ , that is we replace an "infinite number of constraints" by a finite number of them.

Reformulation: replace  $g(x, \xi) \leq 0 \quad \forall \xi \in R$ ,  
by  $\sup_{\xi \in R} g(x, \xi) \leq 0$ ,  
then explicit the sup.

---

<sup>1</sup>For simplicity reason we dropped w.l.o.g. the uncertainty in the objective.

# Solving a robust optimization problem

The robust optimization problem we want to solve is<sup>1</sup>

$$\begin{aligned} \min_x \quad & L(x) \\ \text{s.t.} \quad & g(x, \xi) \leq 0 \quad \forall \xi \in R \end{aligned}$$

Two main approaches are possible:

Constraint generation: replace  $R$  by a finite set of  $\xi$ , that is we replace an "infinite number of constraints" by a finite number of them.

Reformulation: replace  $g(x, \xi) \leq 0 \quad \forall \xi \in R$ ,  
by  $\sup_{\xi \in R} g(x, \xi) \leq 0$ ,  
then explicit the sup.

---

<sup>1</sup>For simplicity reason we dropped w.l.o.g. the uncertainty in the objective.



# Solving a robust optimization problem

The robust optimization problem we want to solve is<sup>1</sup>

$$\begin{aligned} \min_x \quad & L(x) \\ \text{s.t.} \quad & g(x, \xi) \leq 0 \quad \forall \xi \in R \end{aligned}$$

Two main approaches are possible:

**Constraint generation:** replace  $R$  by a finite set of  $\xi$ , that is we replace an "infinite number of constraints" by a finite number of them.

**Reformulation:** replace  $g(x, \xi) \leq 0 \quad \forall \xi \in R$ ,  
by  $\sup_{\xi \in R} g(x, \xi) \leq 0$ ,  
then explicit the sup.

---

<sup>1</sup>For simplicity reason we dropped w.l.o.g. the uncertainty in the objective.

# Solving a robust optimization problem

The robust optimization problem we want to solve is<sup>1</sup>

$$\begin{aligned} \min_x \quad & L(x) \\ \text{s.t.} \quad & g(x, \xi) \leq 0 \quad \forall \xi \in R \end{aligned}$$

Two main approaches are possible:

**Constraint generation:** replace  $R$  by a finite set of  $\xi$ , that is we replace an "infinite number of constraints" by a finite number of them.

**Reformulation:** replace  $g(x, \xi) \leq 0 \quad \forall \xi \in R$ ,  
by  $\sup_{\xi \in R} g(x, \xi) \leq 0$ ,  
then explicit the sup.

---

<sup>1</sup>For simplicity reason we dropped w.l.o.g. the uncertainty in the objective.

# Constraint generation algorithm

**Data:** Problem parameters, reference uncertainty  $\xi_0$

**Result:** approximate value with gap;

**for**  $k \in \mathbb{N}$  **do**

    solve  $\tilde{v} = \min_x \{L(x) \mid g(x, \xi_k) \forall k \leq k\} \rightsquigarrow x_k$ ;

    solve  $s = \max_{\xi \in R} g(x_k, \xi) \rightsquigarrow \xi_{k+1}$ ;

**if**  $s \leq 0$  **then**

        Robust optimization problem solved,  
        with value  $\tilde{v}$  and optimal solution  $x_k$

## Algorithm 1: Constraint Generation Algorithm

Note that we are solving a problem similar to the deterministic problem with an increasing number of constraints.

This is easy to implement and can be numerically efficient.

# Constraint generation algorithm

**Data:** Problem parameters, reference uncertainty  $\xi_0$

**Result:** approximate value with gap;

**for**  $k \in \mathbb{N}$  **do**

    solve  $\tilde{v} = \min_x \{L(x) \mid g(x, \xi_k) \forall k \leq k\} \rightsquigarrow x_k$ ;

    solve  $s = \max_{\xi \in R} g(x_k, \xi) \rightsquigarrow \xi_{k+1}$ ;

**if**  $s \leq 0$  **then**

        Robust optimization problem solved,  
        with value  $\tilde{v}$  and optimal solution  $x_k$

## Algorithm 1: Constraint Generation Algorithm

Note that we are solving a problem similar to the deterministic problem with an increasing number of constraints.

This is easy to implement and can be numerically efficient.

# Constraint generation algorithm

**Data:** Problem parameters, reference uncertainty  $\xi_0$

**Result:** approximate value with gap;

**for**  $k \in \mathbb{N}$  **do**

    solve  $\tilde{v} = \min_x \{L(x) \mid g(x, \xi_k) \forall k \leq k\} \rightsquigarrow x_k$ ;

    solve  $s = \max_{\xi \in R} g(x_k, \xi) \rightsquigarrow \xi_{k+1}$ ;

**if**  $s \leq 0$  **then**

        Robust optimization problem solved,  
        with value  $\tilde{v}$  and optimal solution  $x_k$

## Algorithm 1: Constraint Generation Algorithm

Note that we are solving a problem similar to the deterministic problem with an increasing number of constraints.

This is easy to implement and can be numerically efficient.

# Reformulation principle

We can write the robust optimization problem as

$$\begin{aligned} \min_x \quad & L(x) \\ \text{s.t.} \quad & \sup_{\xi \in R} g(x, \xi) \leq 0 \end{aligned}$$

Now, there are two ways of simplifying this problem:

- we can explicitly compute  $\bar{g}(x) = \sup_{\xi \in R} g(x, \xi)$ ;
  - by duality we can write  $\sup_{\xi \in R} g(x, \xi) = \min_{\eta \in Q} h(x, \eta)$
- $\min_{\eta \in Q} h(x, \eta) \leq 0$  is equivalent to  $\exists \eta$  such that  $h(x, \eta) \leq 0$ , i.e. just add  $\eta$  as a variable in your optimization problem

# Reformulation principle

We can write the robust optimization problem as

$$\begin{aligned} \min_x \quad & L(x) \\ \text{s.t.} \quad & \sup_{\xi \in R} g(x, \xi) \leq 0 \end{aligned}$$

Now, there are two ways of simplifying this problem:

- we can explicitly compute  $\bar{g}(x) = \sup_{\xi \in R} g(x, \xi)$ ;
  - by duality we can write  $\sup_{\xi \in R} g(x, \xi) = \min_{\eta \in Q} h(x, \eta)$
- $\min_{\eta \in Q} h(x, \eta) \leq 0$  is equivalent to  $\exists \eta$  such that  $h(x, \eta) \leq 0$ , i.e. just add  $\eta$  as a variable in your optimization problem

# Contents

- 1 Introduction and motivations
  - How to add uncertainty in an optimization problem
  - Why shall you do Robust Optimization ?
- 2 Solving the robust optimization problem
- 3 Robust optimization for Linear Programm**
  - Reformulating the problem
  - Ellipsoidal uncertainty set
  - Polyhedral uncertainty set
  - Cardinality constrained LP
- 4 Probability guarantee
- 5 Robust Combinatorial Problem
- 6 Conclusion



# Contents

- 1 Introduction and motivations
  - How to add uncertainty in an optimization problem
  - Why shall you do Robust Optimization ?
- 2 Solving the robust optimization problem
- 3 Robust optimization for Linear Programm**
  - Reformulating the problem
  - Ellipsoidal uncertainty set
  - Polyhedral uncertainty set
  - Cardinality constrained LP
- 4 Probability guarantee
- 5 Robust Combinatorial Problem
- 6 Conclusion

# Canonization of the problem

We consider

$$\begin{aligned} \min_{x \geq 0} \quad & \max_{(A,b,c) \in R} \quad c^T x \\ \text{s.t.} \quad & Ax \leq b \end{aligned}$$

Without loss of generality we can consider a deterministic cost:

$$\begin{aligned} \min_{x \geq 0, \theta} \quad & \theta \\ \text{s.t.} \quad & Ax \leq b & \forall (A, b, c) \in R \\ & c^T x \leq \theta & \forall (A, b, c) \in R \end{aligned}$$

That can be written as

$$\begin{aligned} \min_{x \geq 0, \theta} \quad & \theta \\ \text{s.t.} \quad & a_i^T x - b_i \leq 0 & \forall (A, b, c) \in R, \forall i \in [m] \end{aligned}$$

# Canonization of the problem

We consider

$$\begin{aligned} \min_{x \geq 0} \quad & \max_{(A,b,c) \in R} \quad c^T x \\ \text{s.t.} \quad & Ax \leq b \end{aligned}$$

Without loss of generality we can consider a deterministic cost:

$$\begin{aligned} \min_{x \geq 0, \theta} \quad & \theta \\ \text{s.t.} \quad & Ax \leq b & \forall (A, b, c) \in R \\ & c^T x \leq \theta & \forall (A, b, c) \in R \end{aligned}$$

That can be written as

$$\begin{aligned} \min_{x \geq 0, \theta} \quad & \theta \\ \text{s.t.} \quad & a_i^T x - b_i \leq 0 & \forall (A, b, c) \in R, \forall i \in [m] \end{aligned}$$

# Canonization of the problem

We consider

$$\begin{aligned} \min_{x \geq 0} \quad & \max_{(A,b,c) \in R} \quad c^T x \\ \text{s.t.} \quad & Ax \leq b \end{aligned}$$

Without loss of generality we can consider a deterministic cost:

$$\begin{aligned} \min_{x \geq 0, \theta} \quad & \theta \\ \text{s.t.} \quad & Ax \leq b & \forall (A, b, c) \in R \\ & c^T x \leq \theta & \forall (A, b, c) \in R \end{aligned}$$

That can be written as

$$\begin{aligned} \min_{x \geq 0, \theta} \quad & \theta \\ \text{s.t.} \quad & a_i^T x - b_i \leq 0 & \forall (A, b, c) \in R, \forall i \in [m] \end{aligned}$$

# Canonization of the problem



We now consider

$$\begin{aligned} \min_{x \geq 0} \quad & c^\top x \\ \text{s.t.} \quad & a_i^\top x - b_i \leq 0 \qquad \forall (A, b) \in R, \forall i \in [m] \end{aligned}$$

Let  $R_i$  be the projection of  $R$  onto coordinate  $i$ .

We have in particular  $R \subset R_1 \times \cdots \times R_m$ .

But note that, in the robust constraint,  $R$  can be replaced by  $R_1 \times \cdots \times R_m$ , indeed,

$$\begin{aligned} f_i(x, \xi_i) \leq 0, \quad \forall i \in [m], \quad \forall \xi \in R \\ \iff f_i(x, \xi_i) \leq 0, \quad \forall i \in [m], \forall \xi \in R_1 \times \cdots \times R_m \\ \iff f_i(x, \xi_i) \leq 0, \quad \forall \xi_i \in R_i \quad \forall i \in [m] \end{aligned}$$

# Canonization of the problem



We now consider

$$\begin{aligned} \min_{x \geq 0} \quad & c^\top x \\ \text{s.t.} \quad & a_i^\top x - b_i \leq 0 \qquad \forall (A, b) \in R, \forall i \in [m] \end{aligned}$$

Let  $R_i$  be the projection of  $R$  onto coordinate  $i$ .

We have in particular  $R \subset R_1 \times \dots \times R_m$ .

But note that, in the robust constraint,  $R$  can be replaced by  $R_1 \times \dots \times R_m$ , indeed,

$$\begin{aligned} f_i(x, \xi_i) \leq 0, \quad \forall i \in [m], \quad \forall \xi \in R \\ \iff f_i(x, \xi_i) \leq 0, \quad \forall i \in [m], \forall \xi \in R_1 \times \dots \times R_m \\ \iff f_i(x, \xi_i) \leq 0, \quad \forall \xi_i \in R_i \quad \forall i \in [m] \end{aligned}$$

# Canonization of the problem



We now consider

$$\begin{aligned} \min_{x \geq 0} \quad & c^\top x \\ \text{s.t.} \quad & a_i^\top x - b_i \leq 0 \qquad \forall (A, b) \in R, \forall i \in [m] \end{aligned}$$

Let  $R_i$  be the projection of  $R$  onto coordinate  $i$ .

We have in particular  $R \subset R_1 \times \dots \times R_m$ .

But note that, in the robust constraint,  $R$  can be replaced by  $R_1 \times \dots \times R_m$ , indeed,

$$\begin{aligned} f_i(x, \xi_i) \leq 0, \quad \forall i \in [m], \quad \forall \xi \in R \\ \iff f_i(x, \xi_i) \leq 0, \quad \forall i \in [m], \forall \xi \in R_1 \times \dots \times R_m \\ \iff f_i(x, \xi_i) \leq 0, \quad \forall \xi_i \in R_i \quad \forall i \in [m] \end{aligned}$$

# Canonization of the problem



We now consider

$$\begin{array}{ll} \min_{x \geq 0} & c^\top x \\ \text{s.t.} & a_i^\top x - b_i \leq 0 \end{array} \quad \forall (a_i, b_i) \in R_i, \forall i \in [m]$$



# Canonization of the problem



We now consider

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

$$s.t. \quad a^T x - b \leq 0$$

$$\forall (a, b) \in R,$$

# Canonization of the problem



We now consider

$$\begin{aligned} \min_{x \geq 0} \quad & c^\top x \\ \text{s.t.} \quad & a^\top x - b \leq 0 \qquad \forall (a, b) \in R, \end{aligned}$$

To model correlation we set

$$a = \bar{a} + P\zeta \qquad b = \bar{b} + p^\top \zeta$$

where  $(\bar{a}, \bar{b})$  are the nominal value, and  $\zeta$  is the primitive/residual uncertainty.

# Canonization of the problem



We now consider

$$\begin{array}{ll} \min_{x \geq 0} & c^\top x \\ \text{s.t.} & a^\top x - b \leq 0 \end{array} \quad \forall (a, b) \in R,$$

To model correlation we set

$$a = \bar{a} + P\zeta \quad b = \bar{b} + p^\top \zeta$$

where  $(\bar{a}, \bar{b})$  are the nominal value, and  $\zeta$  is the primitive/residual uncertainty.  
The robust constraint now reads

$$(\bar{a}^\top x - \bar{b}) + (P^\top x - p)^\top \zeta \leq 0 \quad \forall \zeta \in \mathcal{Z}$$

# Canonization of the problem

## IV

Example: assume that  $a$  is a random variable with mean  $\bar{a}$  and covariance  $\Sigma$ . Then, a natural reformulation would be

$$a = \bar{a} + \Sigma^{1/2}\zeta,$$

so that  $\zeta$  is centered with uncorrelated coordinates.

Finally, w.l.o.g. we assume that  $b$  is deterministic (can be obtained by adding a variable  $x_{n+1}$  constrained to be equal to 1).

# Canonization of the problem

## IV

Example: assume that  $a$  is a random variable with mean  $\bar{a}$  and covariance  $\Sigma$ . Then, a natural reformulation would be

$$a = \bar{a} + \Sigma^{1/2}\zeta,$$

so that  $\zeta$  is centered with uncorrelated coordinates.

Finally, w.l.o.g. we assume that  $b$  is deterministic (can be obtained by adding a variable  $x_{n+1}$  constrained to be equal to 1).

# Contents

- 1 Introduction and motivations
  - How to add uncertainty in an optimization problem
  - Why shall you do Robust Optimization ?
- 2 Solving the robust optimization problem
- 3 Robust optimization for Linear Programm**
  - Reformulating the problem
  - Ellipsoidal uncertainty set**
  - Polyhedral uncertainty set
  - Cardinality constrained LP
- 4 Probability guarantee
- 5 Robust Combinatorial Problem
- 6 Conclusion

# An explicit worst case value

- We consider an ellipsoidal uncertainty set

$$R = \left\{ \xi = \{\bar{a} + P\zeta\}; \quad | \quad \|\zeta\|_2 \leq \rho \right\}$$

- Here we can, for a given  $x$ , explicitly compute

$$\begin{aligned} \sup_{\xi \in R} \xi^T x &= \bar{a}^T x + \sup_{\|\zeta\|_2 \leq \rho} (P\zeta)^T x \\ &= \bar{a}^T x + \rho \|P^T x\|_2 \end{aligned}$$

- Hence, constraint

$$\sup_{\xi \in R} \xi^T x \leq b$$

can be written

$$\bar{a}^T x + \rho \|P^T x\|_2 \leq b$$

# An explicit worst case value

- We consider an ellipsoidal uncertainty set

$$R = \left\{ \xi = \{\bar{a} + P\zeta\}; \quad | \quad \|\zeta\|_2 \leq \rho \right\}$$

- Here we can, for a given  $x$ , explicitly compute

$$\begin{aligned} \sup_{\xi \in R} \xi^T x &= \bar{a}^T x + \sup_{\|\zeta\|_2 \leq \rho} (P\zeta)^T x \\ &= \bar{a}^T x + \rho \|P^T x\|_2 \end{aligned}$$

- Hence, constraint

$$\sup_{\xi \in R} \xi^T x \leq b$$

can be written

$$\bar{a}^T x + \rho \|P^T x\|_2 \leq b$$



# An explicit worst case value

- We consider an ellipsoidal uncertainty set

$$R = \left\{ \xi = \{\bar{a} + P\zeta\}_i \mid \|\zeta\|_2 \leq \rho \right\}$$

- Here we can, for a given  $x$ , explicitly compute

$$\begin{aligned} \sup_{\xi \in R} \xi^\top x &= \bar{a}^\top x + \sup_{\|\zeta\|_2 \leq \rho} (P\zeta)^\top x \\ &= \bar{a}^\top x + \rho \|P^\top x\|_2 \end{aligned}$$

- Hence, constraint

$$\sup_{\xi \in R} \xi^\top x \leq b$$

can be written

$$\bar{a}^\top x + \rho \|P^\top x\|_2 \leq b$$

# SOCP problem

- An Second Order Cone Programming constraint is a constraint of the form

$$\|Ax + b\|_2 \leq c^T x + d$$

- An SOCP problem is a (continuous) optimization problem with linear cost and linear and SOCP constraints
- There exists powerful software to solve SOCP (e.g. CPLEX, Gurobi, MOSEK...) with dedicated interior points methods
- There exist a duality theory akin to the LP duality theory
- If a robust optimization problem can be cast as an SOCP the formulation is deemed efficient

# Contents

- 1 Introduction and motivations
  - How to add uncertainty in an optimization problem
  - Why shall you do Robust Optimization ?
- 2 Solving the robust optimization problem
- 3 Robust optimization for Linear Programm**
  - Reformulating the problem
  - Ellipsoidal uncertainty set
  - Polyhedral uncertainty set**
  - Cardinality constrained LP
- 4 Probability guarantee
- 5 Robust Combinatorial Problem
- 6 Conclusion

# Linear duality: recalls

- Recall that, if finite,

$$\begin{aligned} \max_{\xi} \quad & \xi^T x \\ \text{s.t.} \quad & D\xi \leq d \end{aligned}$$

as the same value as

$$\begin{aligned} \min_{\eta} \quad & \eta^T d \\ \text{s.t.} \quad & \eta^T D = x \\ & \eta \geq 0 \end{aligned}$$

- Thus,

$$\begin{aligned} \sup_{\xi: D\xi \leq d} \xi^T x \leq b & \iff \min_{\eta \geq 0: \eta^T D = x} \eta^T d \leq b \\ & \iff \exists \eta \geq 0, \eta^T D = x, \eta^T d \leq b \end{aligned}$$

# Linear duality: recalls

- Recall that, if finite,

$$\begin{aligned} \max_{\xi} \quad & \xi^T x \\ \text{s.t.} \quad & D\xi \leq d \end{aligned}$$

as the same value as

$$\begin{aligned} \min_{\eta} \quad & \eta^T d \\ \text{s.t.} \quad & \eta^T D = x \\ & \eta \geq 0 \end{aligned}$$

- Thus,

$$\begin{aligned} \sup_{\xi: D\xi \leq d} \xi^T x \leq b & \iff \min_{\eta \geq 0: \eta^T D = x} \eta^T d \leq b \\ & \iff \exists \eta \geq 0, \eta^T D = x, \eta^T d \leq b \end{aligned}$$

# Polyhedral uncertainty

- We consider a polyhedral uncertainty set

$$R = \left\{ a \mid Da \leq d \right\}$$

- Then the robust optimization problem

$$\begin{aligned} \min_{x \geq 0} \quad & c^T x \\ \text{s.t.} \quad & \sup_{a \in R} a^T x \leq b \end{aligned}$$

reads

$$\begin{aligned} \min_{x \geq 0, \eta \geq 0} \quad & c^T x \\ \text{s.t.} \quad & \eta^T d \leq b \\ & \eta^T D = x \end{aligned}$$

# Contents

- 1 Introduction and motivations
  - How to add uncertainty in an optimization problem
  - Why shall you do Robust Optimization ?
- 2 Solving the robust optimization problem
- 3 Robust optimization for Linear Programm**
  - Reformulating the problem
  - Ellipsoidal uncertainty set
  - Polyhedral uncertainty set
  - **Cardinality constrained LP**
- 4 Probability guarantee
- 5 Robust Combinatorial Problem
- 6 Conclusion

# Soyster model

The problem

$$\begin{aligned} \min_x \quad & c^T x \\ & \tilde{A}x \leq b \\ & \underline{x} \leq x \leq \bar{x} \end{aligned} \quad \forall \tilde{A} \in R$$

where each coefficient  $\tilde{a}_{ij} \in [\bar{a}_{ij} - \delta_{ij}, \bar{a}_{ij} + \delta_{ij}]$



# Soyster model

The problem

$$\begin{aligned} \min_x \quad & c^T x \\ & \sup_{\tilde{A} \in R} \tilde{A}x \leq b \\ & \underline{x} \leq x \leq \bar{x} \end{aligned}$$

where each coefficient  $\tilde{a}_{ij} \in [\bar{a}_{ij} - \delta_{ij}, \bar{a}_{ij} + \delta_{ij}]$

# Soyster model

The problem

$$\begin{aligned} \min_x \quad & c^T x \\ & \sup_{\tilde{A} \in R} \tilde{A}x \leq b \\ & \underline{x} \leq x \leq \bar{x} \end{aligned}$$

where each coefficient  $\tilde{a}_{ij} \in [\bar{a}_{ij} - \delta_{ij}, \bar{a}_{ij} + \delta_{ij}]$   
can be written

$$\begin{aligned} \min_x \quad & c^T x \\ & \sum_j \bar{a}_{ij} x_j + \sum_j \delta_{ij} |x_j| \leq b_i \quad \forall i \\ & \underline{x} \leq x \leq \bar{x} \end{aligned}$$

# Soyster model

The problem

$$\begin{aligned} \min_x \quad & c^T x \\ & \sup_{\tilde{A} \in R} \tilde{A}x \leq b \\ & \underline{x} \leq x \leq \bar{x} \end{aligned}$$

where each coefficient  $\tilde{a}_{ij} \in [\bar{a}_{ij} - \delta_{ij}, \bar{a}_{ij} + \delta_{ij}]$   
can be written

$$\begin{aligned} \min_x \quad & c^T x \\ & \sum_j \bar{a}_{ij} x_j + \sum_j \delta_{ij} y_j \leq b_i \quad \forall i \\ & \underline{x} \leq x \leq \bar{x} \\ & y_j \geq x_j, \quad y_j \geq -x_j \end{aligned}$$

# Cardinality constrained LP

Soyster's model is over conservative, we want to consider a model where only  $\Gamma_i$  coefficient per line have non-zero errors, leading to

$$\min_{x,y} c^T x$$

$$\sum_j \bar{a}_{ij} x_j + \max_{S_i: |S_i|=\Gamma_i} \sum_{j \in S_i} \delta_{ij} y_j \leq b_i \quad \forall i$$

$$\underline{x} \leq x \leq \bar{x}$$

$$y_j \geq x_j, \quad y_j \geq -x_j$$

# Cardinality constrained LP

Soyster's model is over conservative, we want to consider a model where only  $\Gamma_i$  coefficient per line have non-zero errors, leading to

$$\begin{aligned}
 & \min_{x,y} c^T x \\
 & \sum_j \bar{a}_{ij} x_j + \beta_i \leq b_i \quad \forall i \\
 & \max_{S_i: |S_i|=\Gamma_i} \sum_{j \in S_i} \delta_{ij} y_j \leq \beta_i \\
 & \underline{x} \leq x \leq \bar{x} \\
 & y_j \geq x_j, \quad y_j \geq -x_j
 \end{aligned}$$

# Cardinality constrained LP



This means that, for line  $i$  we take a margin of

$$\beta_i(x, \Gamma_i) := \max_{S_i: |S_i|=\Gamma_i} \sum_{j \in S_i} \delta_{ij} |x_j|$$

which can be obtained as

$$\begin{aligned} \beta_i(x, \Gamma_i) = \max_{z \geq 0} \quad & \sum_j \delta_{ij} |x_j| z_{ij} \\ & \sum_j z_{ij} \leq \Gamma_i \\ & z_{ij} \in \{0, 1\} \end{aligned}$$

# Cardinality constrained LP



This means that, for line  $i$  we take a margin of

$$\beta_i(x, \Gamma_i) := \max_{S_i: |S_i|=\Gamma_i} \sum_{j \in S_i} \delta_{ij} |x_j|$$

which can be obtained as

$$\begin{aligned} \beta_i(x, \Gamma_i) = \max_{z \geq 0} \quad & \sum_j \delta_{ij} |x_j| z_{ij} \\ & \sum_j z_{ij} \leq \Gamma_i \\ & z_{ij} \leq 1 \end{aligned}$$

# Cardinality constrained LP



This means that, for line  $i$  we take a margin of

$$\beta_i(x, \Gamma_i) := \max_{S_i: |S_i|=\Gamma_i} \sum_{j \in S_i} \delta_{ij} |x_j|$$

which can be obtained as

$$\begin{aligned} \beta_i(x, \Gamma_i) = \max_{z \geq 0} \quad & \sum_j \delta_{ij} |x_j| z_{ij} \\ & \sum_j z_{ij} \leq \Gamma_i && [\lambda_i] \\ & z_{ij} \leq 1 && [\mu_{ij}] \end{aligned}$$

This LP can be then dualized to be integrated in the original LP.



## Cardinality constrained LP



$$\beta_i(x, \Gamma_i) = \max_{z \geq 0} \sum_j \delta_{ij} |x_j| z_{ij}$$
$$\sum_j z_{ij} \leq \Gamma_i \quad [\lambda_i]$$
$$z_{ij} \leq 1 \quad [\mu_{ij}]$$

## Cardinality constrained LP



$$\begin{aligned}\beta_i(x, \Gamma_i) = \max_{z \geq 0} \quad & \sum_j \delta_{ij} |x_j| z_{ij} \\ & \sum_j z_{ij} \leq \Gamma_i && [\lambda_i] \\ & z_{ij} \leq 1 && [\mu_{ij}]\end{aligned}$$

$$\beta_i(x, \Gamma_i) = \max_{z \geq 0} \min_{\lambda, \mu \geq 0} \sum_j \delta_{ij} |x_j| z_{ij} + \lambda_i \left( \Gamma_i - \sum_j z_{ij} \right) + \sum_j \mu_{ij} (1 - z_{ij})$$

## Cardinality constrained LP



$$\beta_i(x, \Gamma_i) = \max_{z \geq 0} \sum_j \delta_{ij} |x_j| z_{ij}$$

$$\sum_j z_{ij} \leq \Gamma_i \quad [\lambda_i]$$

$$z_{ij} \leq 1 \quad [\mu_{ij}]$$

$$\beta_i(x, \Gamma_i) = \max_{z \geq 0} \min_{\lambda, \mu \geq 0} \sum_j \delta_{ij} |x_j| z_{ij} + \lambda_i \left( \Gamma_i - \sum_j z_{ij} \right) + \sum_j \mu_{ij} (1 - z_{ij})$$

$$= \min_{\lambda, \mu \geq 0} \max_{z \geq 0} \lambda_i \Gamma_i + \sum_j \mu_{ij} + \sum_j z_{ij} (\delta_{ij} |x_j| - \lambda_i - \mu_{ij})$$

## Cardinality constrained LP



$$\beta_i(x, \Gamma_i) = \max_{z \geq 0} \sum_j \delta_{ij} |x_j| z_{ij}$$

$$\sum_j z_{ij} \leq \Gamma_i \quad [\lambda_i]$$

$$z_{ij} \leq 1 \quad [\mu_{ij}]$$

$$\beta_i(x, \Gamma_i) = \max_{z \geq 0} \min_{\lambda, \mu \geq 0} \sum_j \delta_{ij} |x_j| z_{ij} + \lambda_i \left( \Gamma_i - \sum_j z_{ij} \right) + \sum_j \mu_{ij} (1 - z_{ij})$$

$$= \min_{\lambda, \mu \geq 0} \lambda_i \Gamma_i + \sum_j \mu_{ij}$$

$$\text{s.t.} \quad \delta_{ij} |x_j| \leq \lambda_i + \mu_{ij}$$

## Cardinality constrained LP

## IV

In the end we obtain

$$\begin{aligned}
 & \min_{x, \beta, \lambda, \mu} c^T x \\
 & \sum_j \bar{a}_{ij} x_j + \beta_i \leq b_i \quad \forall i \\
 & \lambda_i \Gamma_i + \sum_j \mu_{ij} \leq \beta_i \quad \forall i \\
 & \delta_{ij} x_j \leq \lambda_i + \mu_{ij} \quad \forall i, j \\
 & -\delta_{ij} x_j \leq \lambda_i + \mu_{ij} \quad \forall i, j \\
 & \lambda \geq 0, \quad \mu \geq 0 \\
 & \underline{x} \leq x \leq \bar{x}
 \end{aligned}$$

# Contents

- 1 Introduction and motivations
  - How to add uncertainty in an optimization problem
  - Why shall you do Robust Optimization ?
- 2 Solving the robust optimization problem
- 3 Robust optimization for Linear Programm
  - Reformulating the problem
  - Ellipsoidal uncertainty set
  - Polyhedral uncertainty set
  - Cardinality constrained LP
- 4 Probability guarantee
- 5 Robust Combinatorial Problem
- 6 Conclusion

# Robust constraint implying a probabilistic guarantee

## Definition

We say that, for a given set of probability measures  $\mathbb{P} \in \mathcal{P}$ , the constraint

$$g(x, \xi) \leq 0, \quad \forall \xi \in R,$$

implies a probabilistic guarantee of level  $1 - \varepsilon$  if, for all  $\mathbb{P} \in \mathcal{P}$ ,

$$\mathbb{P}(g(x, \xi) \leq 0) \geq 1 - \varepsilon.$$

# Probability guarantee for ellipsoidal uncertainty

- We consider a linear constraint

$$\sum_j \tilde{a}_{ij} x_j \leq b_i, \quad \forall i \in [m]$$

- We assume that  $\tilde{a}_{ij} = \bar{a}_{ij}(1 + \varepsilon \xi_{ij})$  where  $\xi_{ij}$  is a random variable with mean 0, contained in  $[-1, 1]$ , and independent in  $j$ .
- Then the robust constraint

$$\sum_j \bar{a}_{ij} x_j + \varepsilon \Omega \sqrt{\sum_j \bar{a}_{ij}^2 x_j^2} \leq b_i^+, \quad \forall i \in [m]$$

implies a probabilistic guarantee of level  $1 - e^{-\Omega^2/2}$ .



# Contents

- 1 Introduction and motivations
  - How to add uncertainty in an optimization problem
  - Why shall you do Robust Optimization ?
- 2 Solving the robust optimization problem
- 3 Robust optimization for Linear Programm
  - Reformulating the problem
  - Ellipsoidal uncertainty set
  - Polyhedral uncertainty set
  - Cardinality constrained LP
- 4 Probability guarantee
- 5 Robust Combinatorial Problem**
- 6 Conclusion

# A combinatorial optimization problem with uncertain cost

We consider a combinatorial optimization problem:

$$\begin{aligned} \min_{x \in \{0,1\}^N} \quad & \max_{\tilde{c} \in R} \tilde{c}^\top x \\ \text{s.t.} \quad & x \in X \end{aligned}$$

where  $R$  is such that each  $\tilde{c}_i \in [\bar{c}_i, \bar{c}_i + \delta_i]$ , with at most  $\Gamma$  coefficient deviating from  $\bar{c}_i$ . Thus, the problem reads

$$\begin{aligned} (P) \quad \min_{x \in \{0,1\}^N} \quad & \bar{c}^\top x + \max_{|S| \leq \Gamma} \sum_{i \in S} \delta_i x_i \\ \text{s.t.} \quad & x \in X \end{aligned}$$

wlog we assume that the  $i$  are ordered by decreasing cost uncertainty span:  
 $\delta_1 \geq \delta_2 \geq \dots \geq \delta_n$ .

# A combinatorial optimization problem with uncertain cost

We consider a combinatorial optimization problem:

$$\begin{aligned} & \min_{x \in \{0,1\}^N} \max_{\tilde{c} \in R} \tilde{c}^\top x \\ & \text{s.t. } x \in X \end{aligned}$$

where  $R$  is such that each  $\tilde{c}_i \in [\bar{c}_i, \bar{c}_i + \delta_i]$ , with at most  $\Gamma$  coefficient deviating from  $\bar{c}_i$ . Thus, the problem reads

$$\begin{aligned} (P) \quad & \min_{x \in \{0,1\}^N} \bar{c}^\top x + \max_{|S| \leq \Gamma} \sum_{i \in S} \delta_i x_i \\ & \text{s.t. } x \in X \end{aligned}$$

wlog we assume that the  $i$  are ordered by decreasing cost uncertainty span:  
 $\delta_1 \geq \delta_2 \geq \dots \geq \delta_n$ .

# Solving the robust combinatorial problem

We can write  $(P)$  as

$$\begin{aligned} \min_{x \in \{0,1\}^N} \quad & \max_{\zeta \in [0,1]^n} \quad \bar{c}^\top x + \sum_{i=1}^n \delta_i x_i \zeta_i \\ \text{s.t.} \quad & x \in X \\ & \sum_{i=1}^n \zeta_i \leq \Gamma \end{aligned}$$

For a given  $x \in X$  we dualize the inner maximization LP problem

# Solving the robust combinatorial problem



Thus we can write  $(P)$  as

$$\min_{x, y, \theta} \quad \bar{c}^T x + \Gamma \theta + \sum_{j=1}^n y_j$$

$$\text{s.t.} \quad x \in X$$

$$y_j + \theta \geq \delta_j x_j$$

$$y_j, \theta \geq 0$$

Note that an optimal solution satisfies

$$y_j = (\delta_j x_j - \theta)^+ = (\delta_j - \theta)^+ x_j$$

as  $x_j \in \{0, 1\}$ , and  $\theta \geq 0$ .

# Solving the robust combinatorial problem



Thus we can write  $(P)$  as

$$\begin{aligned} \min_{x, y, \theta} \quad & \bar{c}^\top x + \Gamma \theta + \sum_{j=1}^n y_j \\ \text{s.t.} \quad & x \in X \\ & y_j + \theta \geq \delta_j x_j \\ & y_j, \theta \geq 0 \end{aligned}$$

Note that an optimal solution satisfies

$$y_j = (\delta_j x_j - \theta)^+ = (\delta_j - \theta)^+ x_j$$

as  $x_j \in \{0, 1\}$ , and  $\theta \geq 0$ .

# Solving the robust combinatorial problem



Thus we can write  $(P)$  as

$$\begin{aligned} \min_{\theta \geq 0} \min_x \quad & \bar{c}^\top x + \Gamma\theta + \sum_{j=1}^n x_j(\delta_j - \theta)^+ \\ \text{s.t.} \quad & x \in X \end{aligned}$$

We can now decompose the problem for  $\theta \in [\delta_\ell, \delta_{\ell-1}]$  where  $\delta_{n+1} = 0$  and  $\delta_0 = +\infty$ .  
Therefore, we have

$$\text{val}(P) = \min_{\ell \in [n]} Z^\ell$$

where

$$Z^\ell = \min_{x \in X, \theta \in [\delta_\ell, \delta_{\ell-1}]} \bar{c}^\top x + \Gamma\theta + \sum_{j=1}^{\ell-1} x_j(\delta_j - \theta)$$

# Solving the robust combinatorial problem



Thus we can write  $(P)$  as

$$\begin{aligned} \min_{\theta \geq 0} \min_x \quad & \bar{c}^\top x + \Gamma\theta + \sum_{j=1}^n x_j(\delta_j - \theta)^+ \\ \text{s.t.} \quad & x \in X \end{aligned}$$

We can now decompose the problem for  $\theta \in [\delta_\ell, \delta_{\ell-1}]$  where  $\delta_{n+1} = 0$  and  $\delta_0 = +\infty$ .  
Therefore, we have

$$\text{val}(P) = \min_{\ell \in [n]} Z^\ell$$

where

$$Z^\ell = \min_{x \in X, \theta \in [\delta_\ell, \delta_{\ell-1}]} \bar{c}^\top x + \Gamma\theta + \sum_{j=1}^{\ell-1} x_j(\delta_j - \theta)$$



# Solving the robust combinatorial problem



Thus we can write  $(P)$  as

$$\begin{aligned} \min_{\theta \geq 0} \min_x \quad & \bar{c}^\top x + \Gamma\theta + \sum_{j=1}^n x_j(\delta_j - \theta)^+ \\ \text{s.t.} \quad & x \in X \end{aligned}$$

We can now decompose the problem for  $\theta \in [\delta_\ell, \delta_{\ell-1}]$  where  $\delta_{n+1} = 0$  and  $\delta_0 = +\infty$ . Therefore, we have

$$\text{val}(P) = \min_{\ell \in [n]} Z^\ell$$

where

$$Z^\ell = \min_{x \in X, \theta \in [\delta_\ell, \delta_{\ell-1}]} \bar{c}^\top x + \Gamma\theta + \sum_{j=1}^{\ell-1} x_j(\delta_j - \theta)$$

## Solving the robust combinatorial problem

## IV

As the problem is linear in  $\theta$  we have that

$$Z^\ell = \min_{x \in X, \theta \in [\delta_\ell, \delta_{\ell-1}]} \bar{c}^\top x + \Gamma \theta + \sum_{j=1}^{\ell-1} x_j (\delta_j - \theta)$$

is attained for  $\theta = \delta_\ell$  or  $\theta = \delta_{\ell-1}$ .

So in the end, we have

$$\text{val}(P) = \min_{\ell \in [n]} G^\ell$$

where

$$G^\ell = \Gamma \delta_\ell + \min_{x \in X} \left\{ \bar{c}^\top x + \sum_{j=1}^{\ell} \underbrace{(\delta_j - \delta_\ell) x_j}_{\geq 0} \right\}$$

# Algorithm for the robust problem

- ① For  $\ell \in [n]$ , solve

$$G^\ell = \Gamma \delta_\ell + \min_{x \in X} \left\{ \bar{c}^\top x + \sum_{i=1}^{\ell} (\delta_i - \delta_\ell) x_j \right\}$$

with optimal solution  $x_\ell$

- ② Set  $\ell^* \in \arg \min_{\ell \in [n]} G^\ell$
- ③ Return  $val(P) = G^{\ell^*}$  and  $x^* = x_\ell$

# Contents

- 1 Introduction and motivations
  - How to add uncertainty in an optimization problem
  - Why shall you do Robust Optimization ?
- 2 Solving the robust optimization problem
- 3 Robust optimization for Linear Programm
  - Reformulating the problem
  - Ellipsoidal uncertainty set
  - Polyhedral uncertainty set
  - Cardinality constrained LP
- 4 Probability guarantee
- 5 Robust Combinatorial Problem
- 6 Conclusion

# Why do robust optimization ?

- Because you want to account for some uncertainty
- Because you want to have a solution that resists to changes in data
- Because your data is unprecise and robustness yield better out-of-sample result
- Because you do not have the law of the uncertainty
- Because you can control the robustness level
- Because your problem is "one-shot"

# Which uncertainty set to choose ?


- An uncertainty set that is computationally tractable
- An uncertainty set that yields good results
- An uncertainty set that have some theoretical soundness
- An uncertainty set that take available data into account
- Select uncertainty set / level through cross-validation


# Is there some theoretical results ?


- Yes: with some assumption over the randomness (e.g. bounded and symmetric around  $\bar{a}$ ) some uncertainty set (e.g. ellipsoidal) have a probabilistic guarantee:


$$\forall \xi \in R_\varepsilon, \quad g(x, \xi) \leq 0 \quad \implies \quad \mathbb{P}(g(x, \xi) \leq 0) \geq 1 - \varepsilon$$


- Yes: in some cases approximation scheme for nominal problem can be extended to robust problem (e.g. cardinal uncertainty in combinatorial problem)
- Yes: using relevant data we can use statistical tools to construct a robust set  $R$  that imply a probabilistic guarantee

 **D. Bertsimas, D. Brown, C. Caramanis**  
Theory and applications of robust optimization  
*Siam Review, 2011.*

 **D. Bertsimas and D. Den Hertog**  
Robust and adaptive optimization  
*Dynamic Ideas, 2022.*

 **BL Gorissen, I. Yanikoglu and D. den Hertog**  
A practical guide to robust optimization  
*Omega, 2015.*

 **D. Bertsimas and M. Sim**  
The price of robustness  
*Operations research, 2004.*

 **A. Ben Tal, L El Ghaoui, A. Nemirovski**  
Robust optimization  
*Springer, 2009.*