Chapter 2:

Robust Counterpart of Linear Programs

General robust LP

$$
\text{(Robust counterpart)} \quad \underset{x}{\text{maximize}} \quad \min_{z \in \mathcal{Z}} \; h(x,z) \qquad \text{subject to} \quad g_j(x,z) \leq 0 \,, \forall\, z \in \mathcal{Z}, \;, \forall\, j = 1, ..., J \,.
$$

We assume that the nominal problem is an LP

$$
\begin{aligned}
h(x,z) \quad &:= c(z)^T x + d(z) \\
g_j(x,z) \quad &:= a_j(z)^T x - b_j(z) \,,
\end{aligned}
$$

And that all functions are affine in « *z* »

$$
\begin{aligned}
c(z) &:= (P_0 z + p_0) \qquad \& \qquad d(z) = q_0^T z + r_0 \,, \\
a_j(z) &:= (P_j z + p_j) \qquad \& \qquad b_j(z) = q_j^T z + r_j \,,
\end{aligned}
$$

In other words, we are left with the following LP-RC

$$
\text{(LP-RC)} \quad \underset{x}{\text{maximize}} \quad \min_{z \in \mathcal{Z}} \; z^T P_0^T x + q_0^T z + p_0^T x + r_0 \tag{2.1a}
$$

$$
\text{subject to} \quad z^T P_j^T x + p_j^T x \leq q_j^T z + r_j \,, \forall\, z \in \mathcal{Z}, \;, \forall\, j = 1, ..., J \,. \tag{2.1b}
$$

NP-hardness for general uncertainty sets

- Take the robust counterpart optimization problem

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & c^T x \\ \text{subject to} \quad & (a+z)^T x \leq b \,, \forall\, z \in \mathcal{Z} \ , \end{aligned}$$

- Verifying for a fixed « x » whether the following claim is true is NP-hard in general, and in particular when the uncertain vector contains integer variables

$$z^T x \leq b - a^T x \ , \ \forall\, z \in \mathcal{Z} \quad \Leftrightarrow \quad \max_{z \in \mathcal{Z}} z^T x \leq b - a^T x$$

Scenario based uncertainty

- Consider the following robust counterpart

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & c^T x \\ \text{subject to} \quad & (a + z)^T x \leq b \,,\, \forall\, z \in \mathcal{Z} \,, \end{aligned}$$

with scenario based uncertainty

$$\mathcal{Z} := \left\{ \bar{z}_1, \bar{z}_2, ..., \bar{z}_K \right\}$$

- Then, one can reduce the problem to

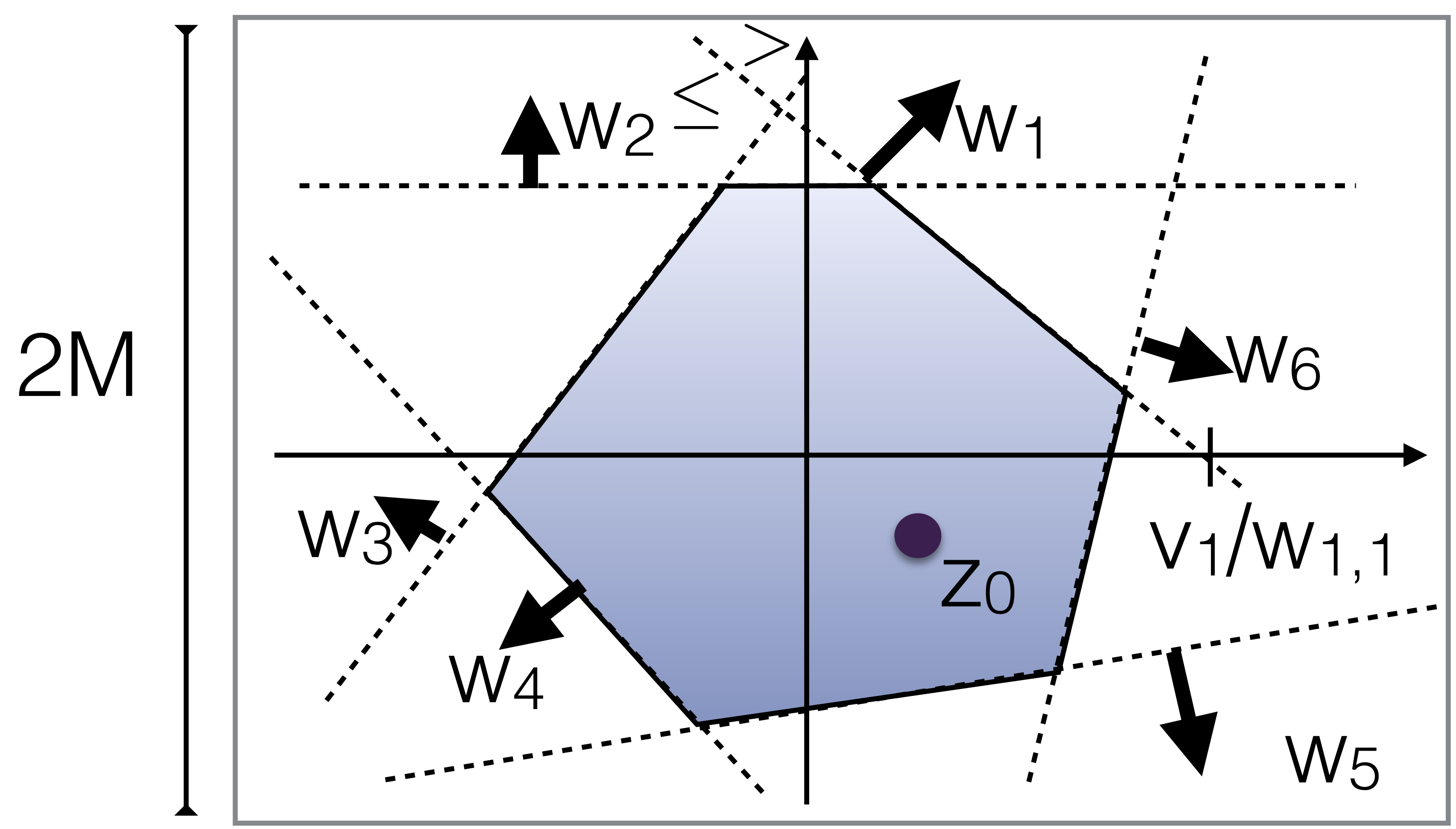
$$\begin{aligned} \underset{x}{\text{maximize}} \quad & c^T x \\ \text{subject to} \quad & (a + \bar{z}_i)^T x \leq b \,,\, \forall i = 1, ..., K \end{aligned}$$

Polyhedral uncertainty

Assumption 2.2. : The uncertainty set $\mathcal{Z}$ is a non-empty and bounded polyhedron that can be defined according to

$$\mathcal{Z} := \{z \in \mathbb{R}^m \,|\, w_i^T z \le v_i \,,\, \forall\, i = 1, ..., s\}\,,$$

where for each $i = 1, ..., s$, we have that $w_i \in \mathbb{R}^{1\times m}$ and $v_i \in \mathbb{R}$ capture a facet of the polyhedron through the expression $w_i^T z = v_i$. Moreover, since $\mathcal{Z}$ is non-empty, there must exist a $z_0 \in \mathcal{Z}$ and since it is bounded there must exist some $M > 0$ such that $\mathcal{Z} = \mathcal{Z} \cap \{z \in \mathbb{R}^m \,|\, -M \le z \le M\}$.



LP reformulation for LP-RC with polyhedral set

Verifying whether $\forall z \in \mathcal{Z},\ z^T x \leq b - a^T x$ is equivalent to evaluating the optimal value of the following problem

$$(\Psi :=) \qquad \underset{z}{\text{maximize}} \quad x^T z \tag{2.3a}$$

$$\text{subject to} \quad Wz \leq v \tag{2.3b}$$

Theorem 2.3. :*(LP Duality see Chapter 4 of [16]) Under assumption 2.2, the optimal value of linear program* (2.3) *is equal to the optimal value of the following dual problem*

$$(\Upsilon^* :=) \qquad \underset{\lambda}{\text{minimize}} \quad v^T \lambda \tag{2.4a}$$

$$\text{subject to} \quad W^T \lambda = x \tag{2.4b}$$

$$\lambda \geq 0 \tag{2.4c}$$

where $\lambda \in \mathbb{R}^s$*. Moreover, problem* (2.4) *has a feasible solution.*

Weak vs. Strong duality

- Weak duality : $\Psi \leq \Upsilon^*$
- Proof of weak duality:

$$\begin{aligned}
\Psi &:= \max_{z: Wz \leq v} x^T z \\
&= \max_{z} \min_{\lambda: \lambda \geq 0} x^T z + \lambda^T (v - Wz) \\
&\leq \min_{\lambda: \lambda \geq 0} \max_{z} x^T z + \lambda^T (v - Wz) \\
&= \min_{\lambda: \lambda \geq 0, x = W^T \lambda} v^T \lambda = \Upsilon^*
\end{aligned}$$

- The challenge of Theorem 2.3 is to prove strong duality
- Strong duality does not necessarily apply when objective is non-linear

Example: box uncertainty

Consider the robust optimization problem:

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & c^T x \\ \text{subject to} \quad & (a+z)^T x \leq b \,, \forall\, z \in \mathcal{Z} \\ & 0 \leq x \leq 1 \,, \end{aligned}$$

with $\mathcal{Z} := \{z \in \mathbb{R}^n | -\hat{z} \leq z \leq \hat{z}\}$

- Formulate an equivalent finite dimensional linear program
- Implement this linear program (a.k.a. the reduced form of the model) using RSOME (incomplete Colab file)

Implementation in RSOME

(see complete Colab file)

- Robust counterpart:

```
#Create model
model = ro.Model('simpleExample_rawrobust')
x=model.dvar(n)

#Create uncertain vector
z= model.rvar(n)
#Create uncertainty set
boxSet= (z>=zBarMinus, z <= zBarPlus)

model.max(c@x)
#Robustify the constraint
model.st(((a+z)@x<=b).forall(boxSet))
model.st(x>=0)
model.st(x<=1)
model.solve(my_solver)
```

- Reduced form:

```
#Create model
model = ro.Model('simpleExample_redrobust')
x=model.dvar(n)
#Create auxiliary variables
lambdaPlus=model.dvar(n)
lambdaMinus=model.dvar(n)

model.max(c@x)
#Modify the deterministic constraint
model.st(a@x + zBarPlus@lambdaPlus -zBarMinus@lambdaMinus <=b)
#Add constraints from dual representation of worst-case optimization
model.st(lambdaPlus-lambdaMinus == x)
model.st(lambdaPlus>=0)
model.st(lambdaMinus>=0)

model.st(x>= 0)
model.st(x<=1)

model.solve(my_solver)
```

Example: box uncertainty (reformulation #2)

Consider the robust optimization problem:

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & c^T x \\ \text{subject to} \quad & (a+z)^T x \leq b \,, \forall\, z \in \mathcal{Z} \\ & 0 \leq x \leq 1 \,, \end{aligned}$$

with $\mathcal{Z} := \{z \in \mathbb{R}^n | -\hat{z} \leq z \leq \hat{z}\}$

- Formulate an equivalent finite dimensional linear program using the equivalent uncertainty set definition:

$$\mathcal{Z} = \left\{ z \in \mathbb{R}^n \middle| \exists \Delta^+ \mathbb{R}^n, \Delta^- \mathbb{R}^n, \begin{array}{c} \Delta^+ \geq 0, \Delta^- \geq 0, \\ z = \Delta^+ - \Delta^-, \\ \Delta^+ + \Delta^- \leq \hat{z} \end{array} \right\}$$

Equivalent LP reformulation for LP-RC

Theorem 2.7. : *The LP-RC problem, with a polyhedral $\mathcal{Z}$ described through $Wz \leq v$ (as in assumption 2.2), is equivalent to the following linear program*

$$
\begin{aligned}
\underset{x,\{\lambda^{(j)}\}_{j=0}^{J}}{\text{maximize}} \quad & p_0^T x + r_0 - v^T \lambda^{(0)} \\
\text{subject to} \quad & W^T \lambda^{(0)} = -P_0^T x - q_0 \\
& p_j^T x + v^T \lambda^{(j)} \leq r_j \,, \forall\, j = 1, \ldots, J \\
& W^T \lambda^{(j)} = P_j^T x - q_j \,, \forall\, j = 1, \ldots, J \\
& \lambda^{(j)} \geq 0 \,, \forall\, j = 0, \ldots, J
\end{aligned}
$$

where $\lambda^{(j)} \in \mathbb{R}^s$ are additional certificates that need to be optimized jointly with x.

SOCP reformulation for LP-RC with ellipsoidal uncertainty

Verifying whether $\forall z \in \mathcal{Z},\ z^T x \leq b - a^T x$ with

$$\mathcal{Z} := \{z \in \mathbb{R}^m | z^T \Sigma^{-1} z \leq \gamma^2\}$$

and $\Sigma \succ 0$. is equivalent to evaluating the optimal value of the following problem

$$\Psi := \max_{z : z^T \Sigma^{-1} z \leq \gamma^2} x^T z$$

One can demonstrate using Cauchy-Schwartz inequality

$$a^T b \leq \|a\|_2 \|b\|_2$$

that this is equivalent to

$$\Psi = \gamma \sqrt{x^T \Sigma x} = \gamma \|\Sigma^{1/2} x\|_2$$

SOCP reformulation for LP-RC with polyhedral set ellipsoidal uncertainty

Theorem. *The LP-RC problem, with ellipsoidal set $\mathcal{Z}$ described is equivalent to the following second order cone program*

$$
\begin{aligned}
\underset{x}{\text{maximize}} \quad & p_0^T x + r_0 - \gamma \|\Sigma^{1/2}(P_0^T x + q_0)\|_2 \\
\text{subject to} \quad & p_j^T x + \gamma \|\Sigma^{1/2}(P_j^T x - q_j)\|_2 \leq r_j \,,\ \forall\, j = 1, \ldots, J\,.
\end{aligned}
$$