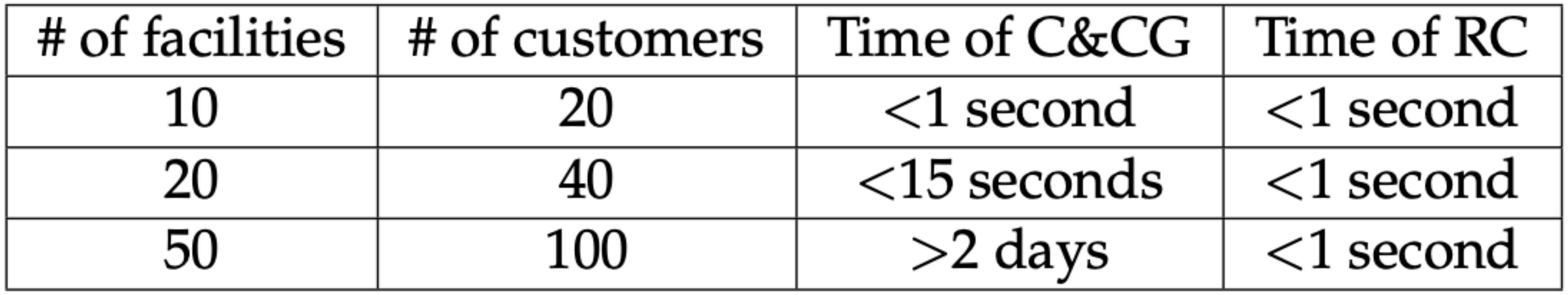
Chapter 5:

Value of Flexibility using Tractable Decision Rules

Limitations of exact methods

1. No guarantee that convergence will be achieved in a reasonable amount of time

Solution time for a robust facility location problem



# of facilities	# of customers	Time of C&CG	Time of RC
10	20	<1 second	<1 second
20	40	<15 seconds	<1 second
50	100	>2 days	<1 second

2. Cannot be applied to multi-stage problems

We will therefore talk about approximation schemes, starting with linear decision rules:

$$x_t(\bar{v}) := x_t + X_t\bar{v} \text{ for some } x_t \in \mathbb{R}^n \text{ and } X_t \in \mathbb{R}^{n\times\nu}$$

Simple newsvendor example

- Recall the inventory management problem:

$$\begin{aligned} \min_{x\in[0,2]} \quad \sup_{z\in[0,2]} \quad \min_{y} \; & 0.5x + y \\ \text{s.t. } & y \geq x - z \\ & y \geq z - x. \end{aligned}$$

Simple newsvendor example

- Recall the inventory management problem:

$$
\begin{aligned}
\min_{x \in [0,2]} \; 0.5x + \sup_{z \in [0,2]} \; \min_{y} \; & y \\
\text{s.t. } & y \geq x - z \\
& y \geq z - x.
\end{aligned}
$$

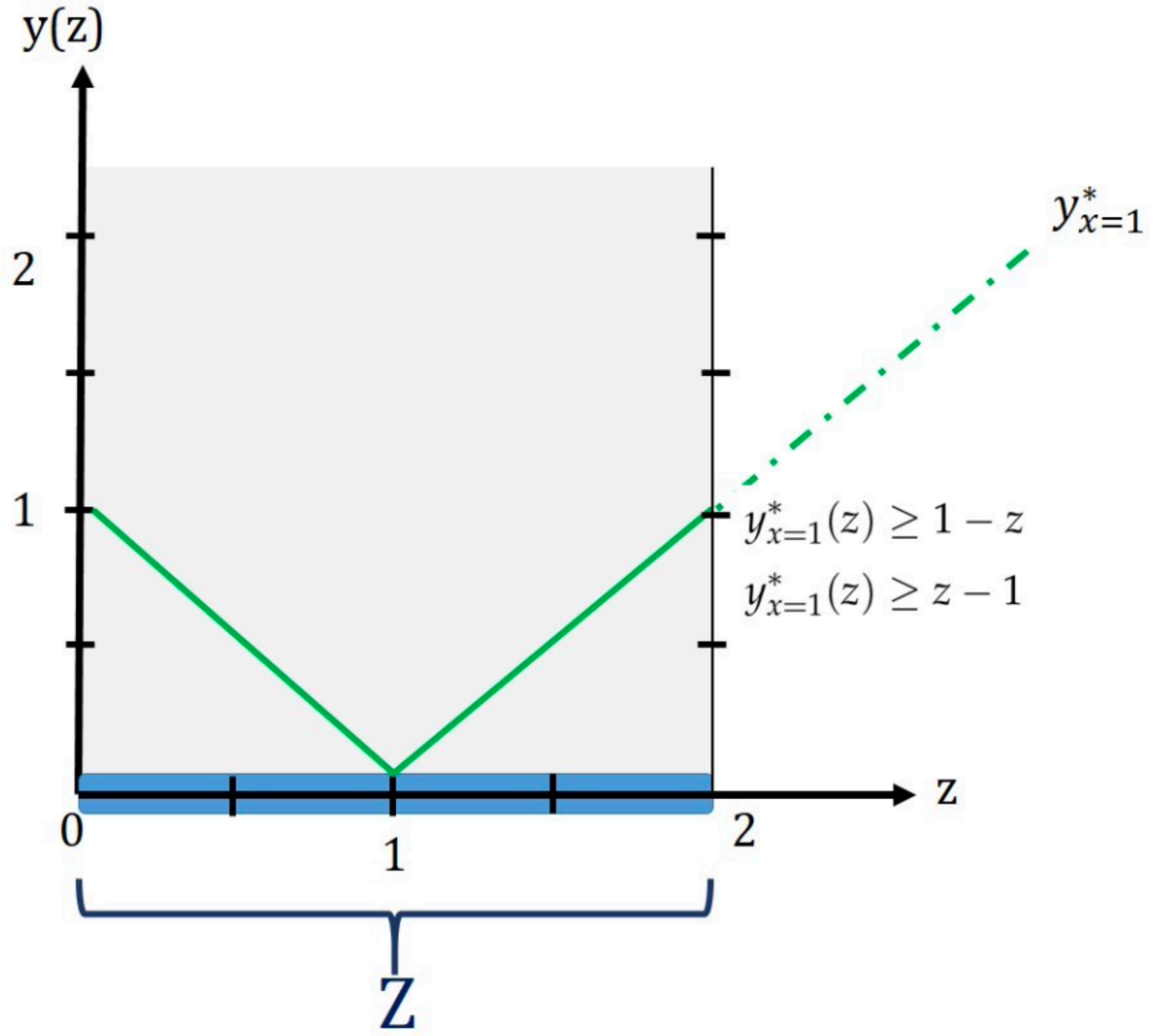
- For any fixed ordering amount, what does the policy of y*(z) look like for different realizations of the demand?

Optimal recourse if $x = 1$

$$\min_y \; y$$

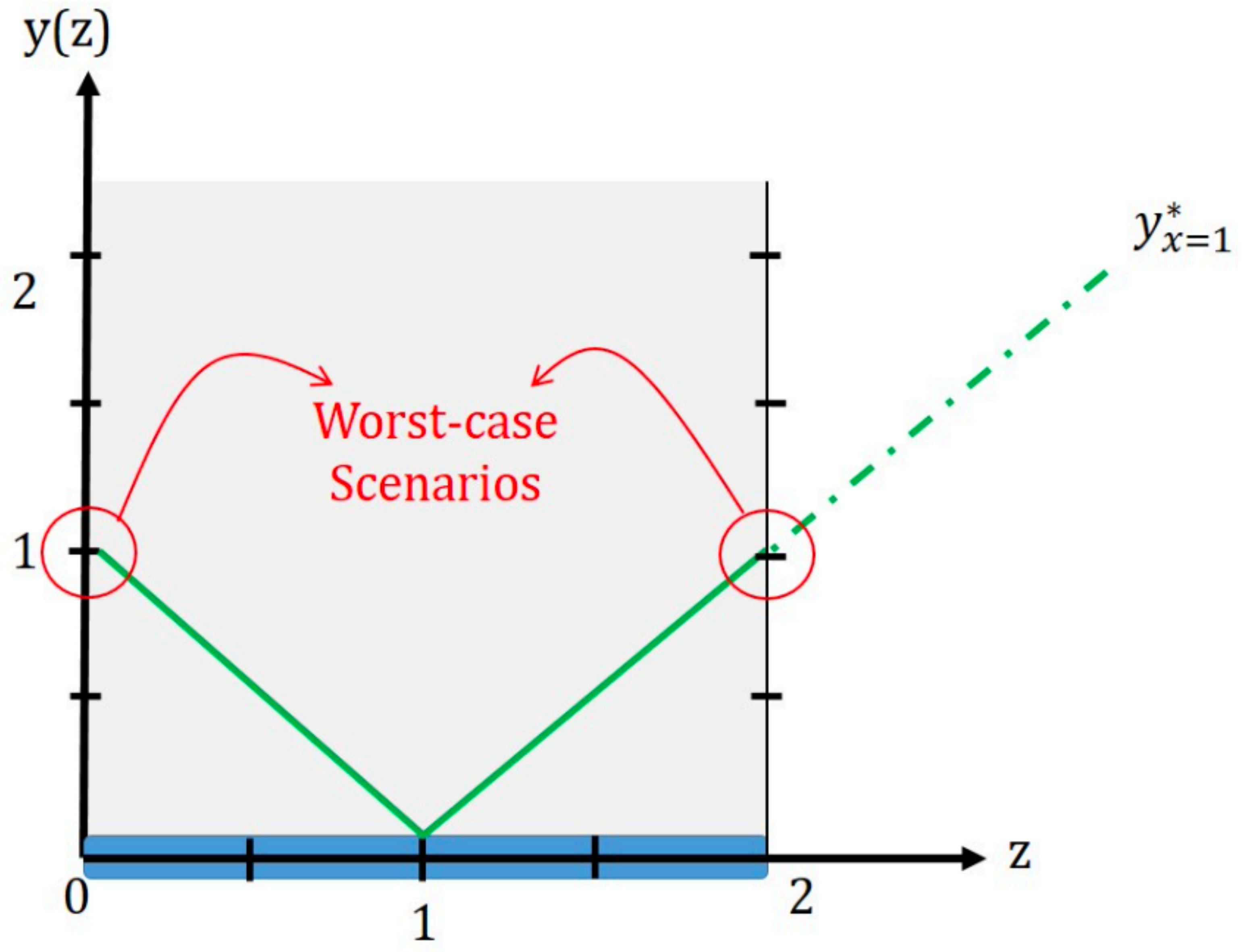
$$\text{s.t. } y \geq 1 - z$$

$$y \geq z - 1.$$



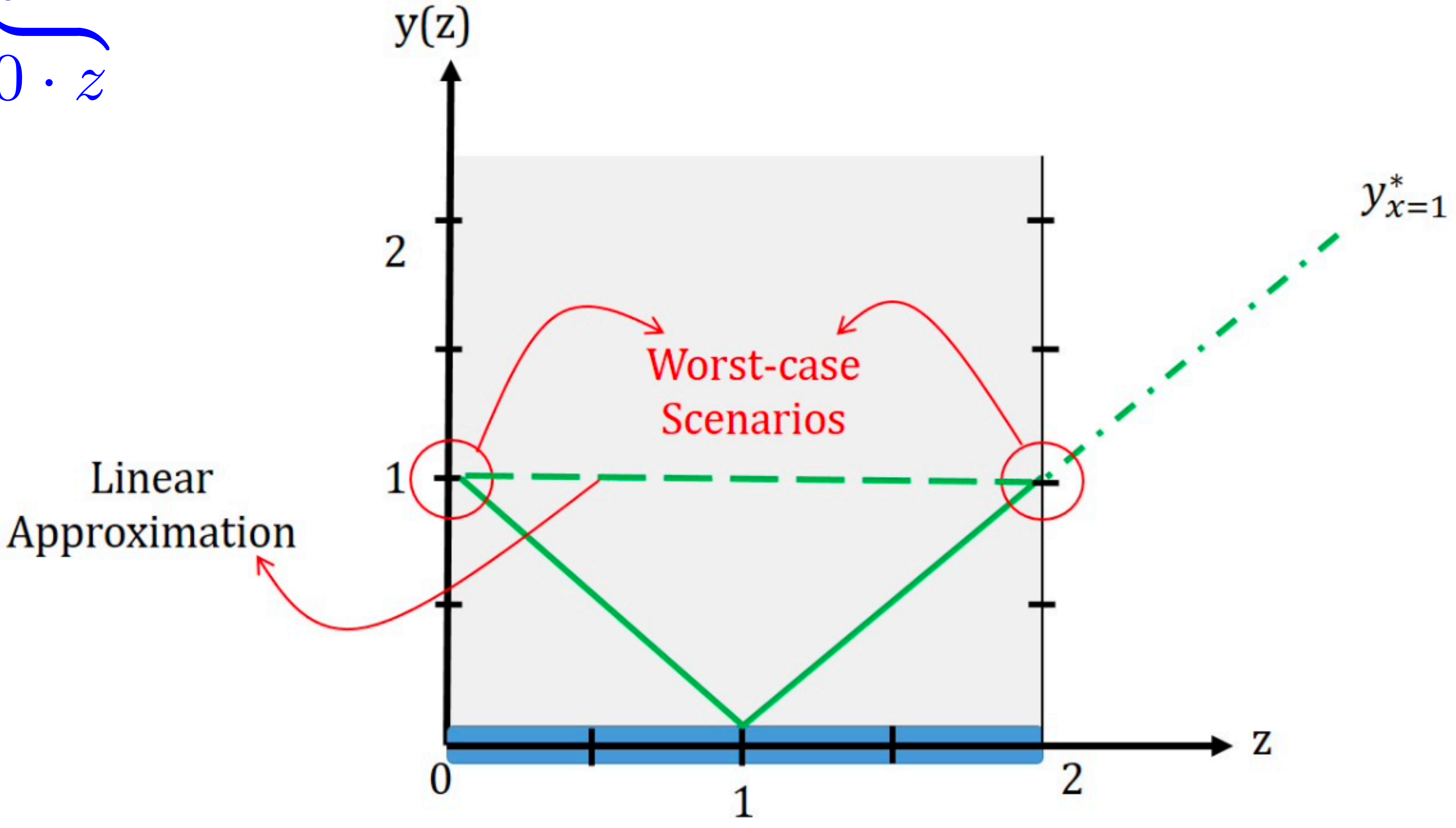
Optimal recourse if $x = 1$

$$\sup_{z \in [0,2]} \min_{y} y$$
$$\text{s.t. } y \geq 1 - z$$
$$y \geq z - 1.$$



Optimal recourse if $x = 1$

$$\sup_{z \in [0,2]} \overbrace{1 + 0 \cdot z}^{\bar{y} + \hat{y} z}$$

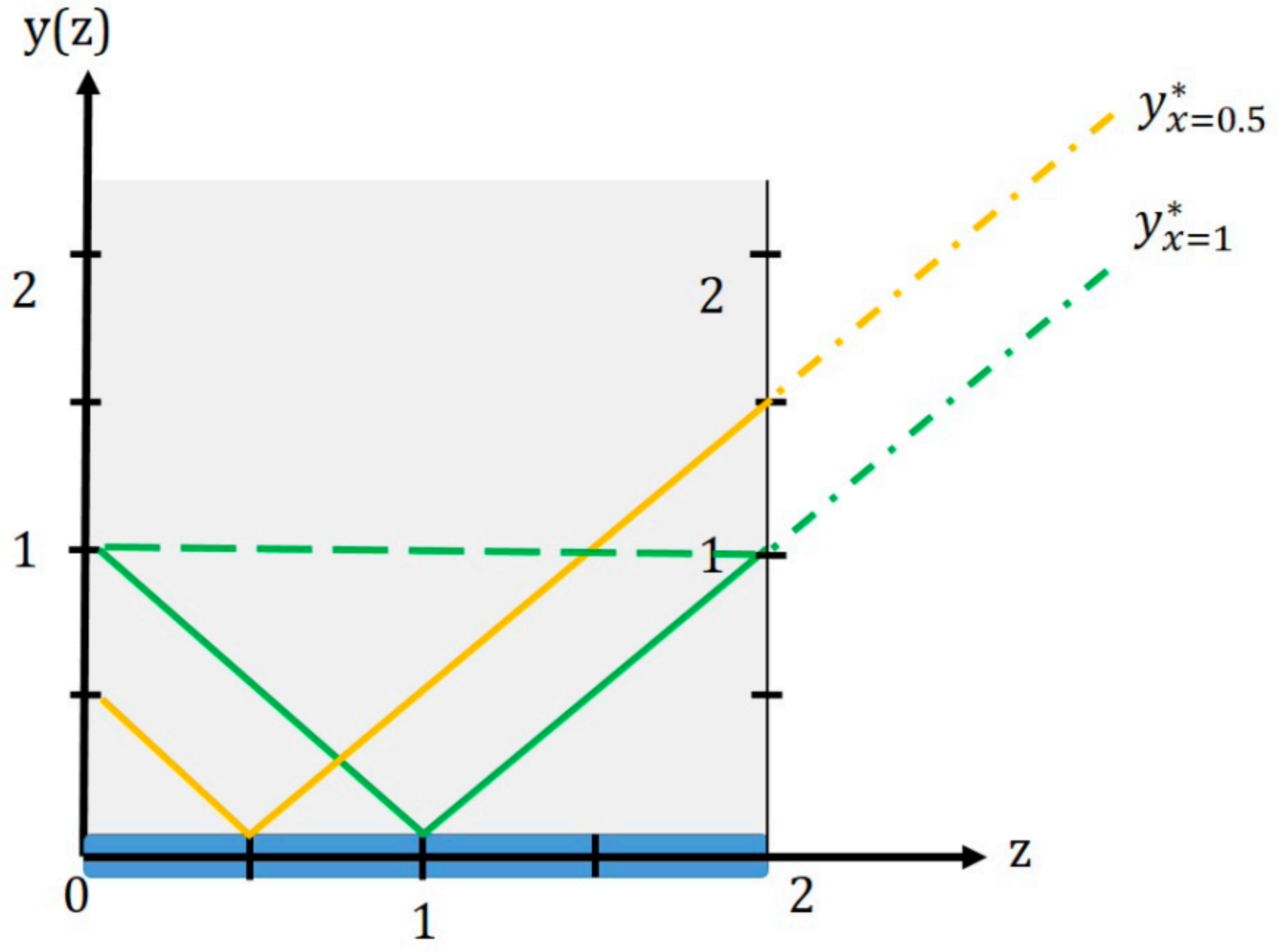


Optimal recourse if $x = 0.5$

$$\min_{y} \; y$$

$$\text{s.t. } y \geq 0.5 - z$$

$$y \geq z - 0.5.$$



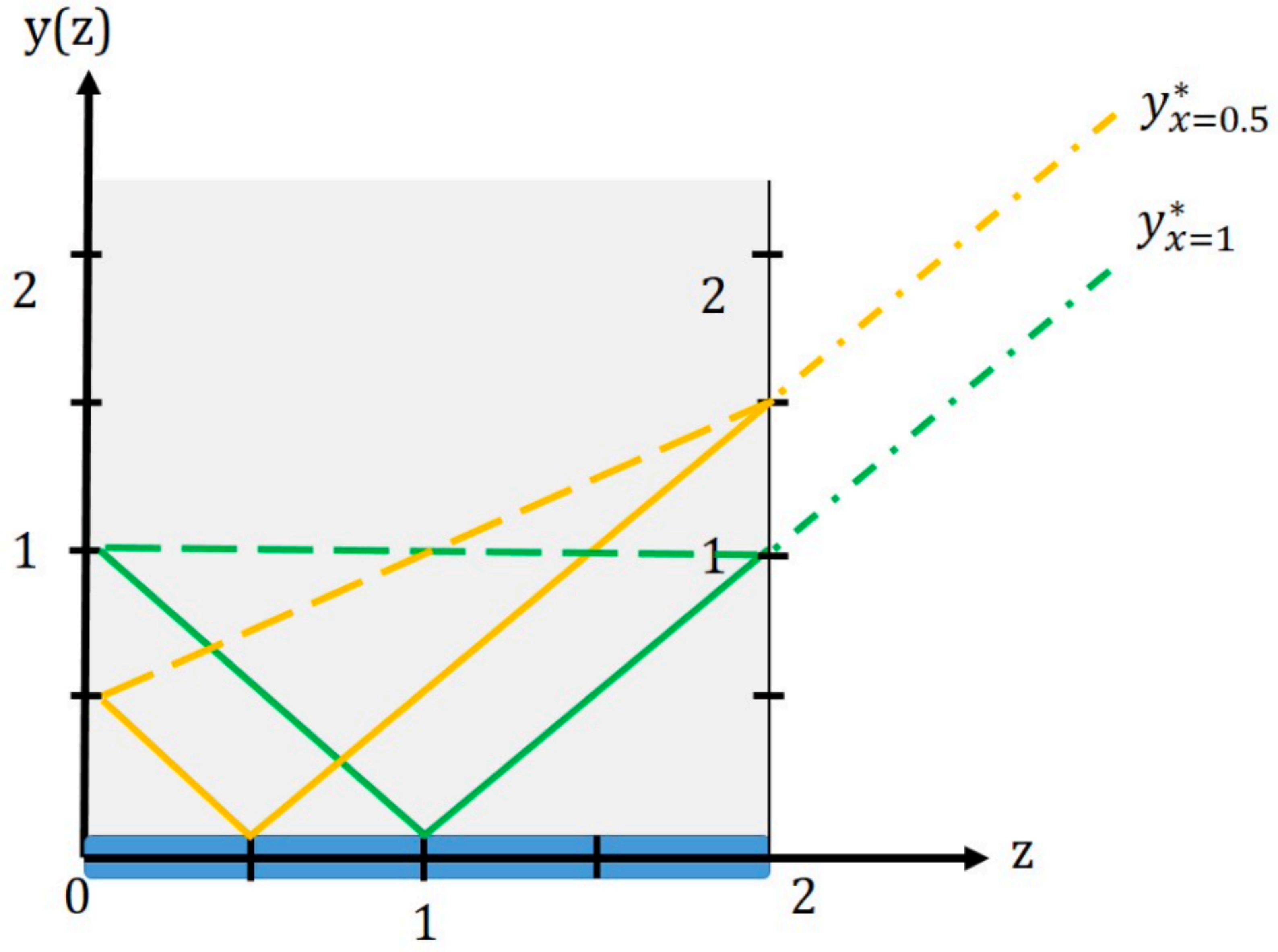
Optimal recourse if $x = 0.5$

$$\sup_{z\in[0,2]} \min_{y} y$$

$$\text{s.t. } y \geq 0.5 - z$$

$$y \geq z - 0.5.$$

$$\sup_{z\in[0,2]} \overbrace{0.5 + 0.5 \cdot z}^{\bar{y}+\hat{y}z}$$



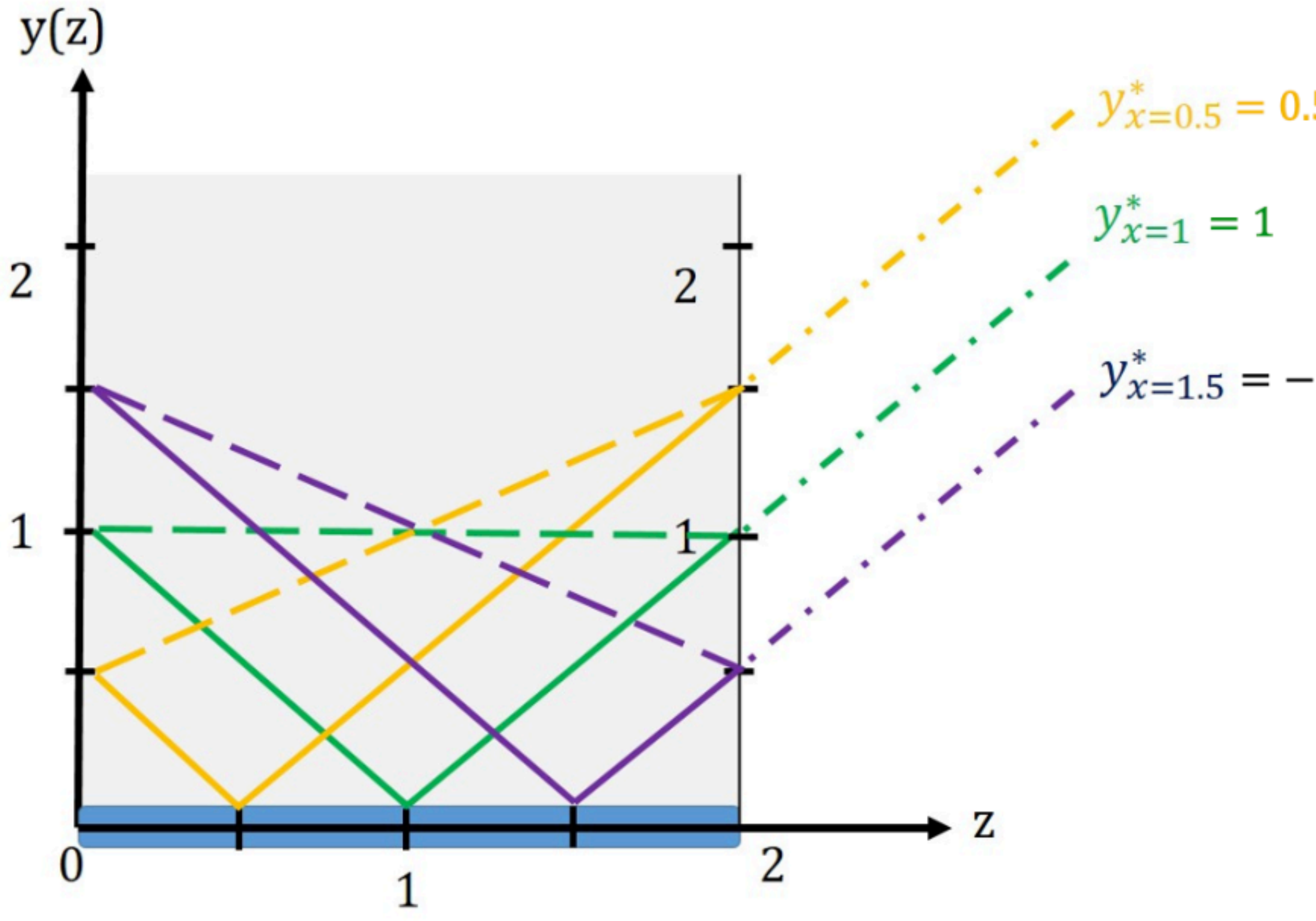
Optimal recourse if $x = 1.5$

$$\sup_{z\in[0,2]} \min_y y$$

$$\text{s.t. } y \geq 1.5 - z$$

$$y \geq z - 1.5.$$

$$\sup_{z\in[0,2]} \overbrace{1.5 - 0.5 \cdot z}^{\bar{y}+\hat{y}z}$$



Affinely Adjustable Robust Counterpart Model

Multi-stage ARC:

$$\underset{x_1,\{x_t(\cdot)\}_{t=2}^T}{\text{maximize}} \quad \inf_{z\in\mathcal{Z}} \; c_1(z)^T x_1 + \sum_{t=2}^{T} c_t(z)^T x_t(v_t(z)) + d(z) \tag{4.8a}$$

$$\text{subject to} \quad a_{j1}(z)^T x_1 + \sum_{t=2}^{T} a_{jt}(z)^T x_t(v_t(z)) \leq b_j(z)\,,\ \forall\, z \in \mathcal{Z}\,,\ \forall\, j = 1,\ldots,J \tag{4.8b}$$

AARC:

$$\underset{\{x_t\}_{t=1}^T,\{X_t\}_{t=2}^T}{\text{maximize}} \quad \inf_{z\in\mathcal{Z}} \; c_1(z)^T x_1 + \sum_{t=1}^{T} c_t(z)^T (x_t + X_t v_t(z)) + d(z) \tag{5.1b}$$

$$\text{subject to} \quad a_{j1}^T x_1 + \sum_{t=1}^{T} a_{jt}(z)^T (x_t + X_t v_t(z)) \leq b_j(z)\,,\ \forall\, z \in \mathcal{Z}\,,\ \forall\, j = 1,\ldots,J \tag{5.1c}$$

Assumption 5.2.

1. ARC model has fixed recourse property:
   $c_t(z) := c_t$ and $a_{jt}(z) := a_{jt}, \forall t = 2, \cdots, T; j = 1, \cdots, J.$
2. All observations are linear functions of z:
   $v_t(z) := V_t z, \forall t = 2, \cdots, T$ for some $V_t \in \mathbb{R}^{v \times m}$.

$$\max_{x_1, \{x_t\}_{t=2}^T, \{X_t\}_{t=2}^T} \quad \inf_{z \in \mathcal{Z}} \ c_1(z)^T x_1 + \sum_{t=2} c_t^T \left(x_t + X_t V_t z\right) + d(z)$$

$$\text{s. t.} \quad a_{j1}(z)^T x_1 + \sum_{t=2}^{T} a_{jt}^T \left(x_t + X_t V_t z\right) \leq b_j(z) \,, \forall z \in \mathcal{Z} \,, \forall j \,.$$

Theorem 5.3.

Let the ARC satisfy Assumption 5.2., and $\mathcal{Z}$ be a non-empty bounded polyhedron, then AARC

1. can be described as a robust linear program.
2. provides a <u>"conservative" approximation</u>[a] of ARC.

[a]Model (A) is a conservative approximation of model (B) if all feasible solutions of (A) are feasible solutions in (B) and, if the objective function of (A) underestimates the performance measured by the objective function of (B) for all feasible solutions of (A).

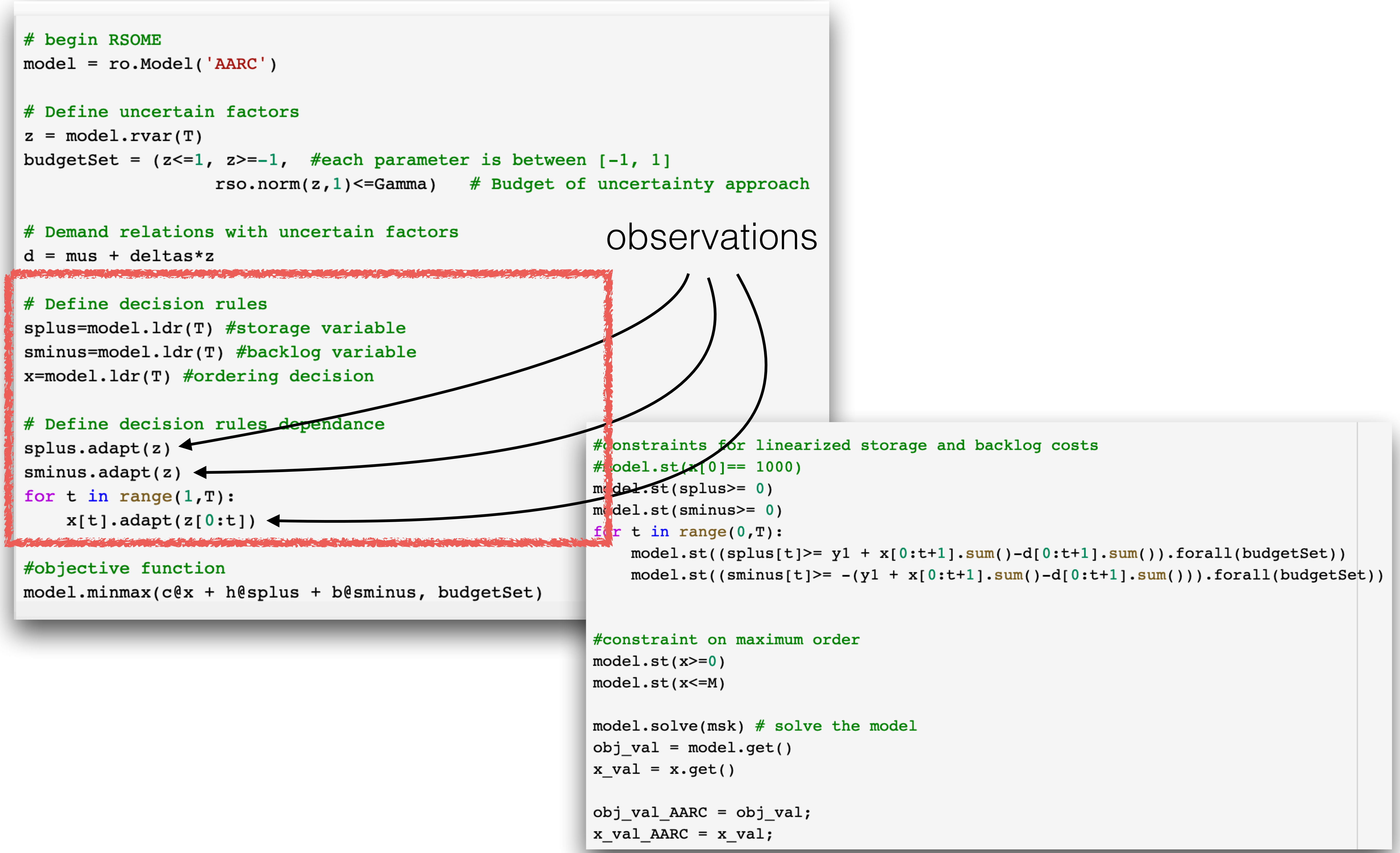
Inventory problem

$$\begin{array}{ll}\underset{\substack{x_1,\{x_t,X_t\}_{t=2}^T,\\ \{s_t^+,S_t^+\}_{t=1}^T,\{s_t^-,S_t^-\}_{t=1}^T}}{\text{minimize}} & \sup_{d\in\mathcal{U}} \; c_1x_1 + \sum_t c_t(x_t + X_tV_td) + h_t(s_t^+ + S_t^+d) + b_t(s_t^- + S_t^-d) \\ \text{subject to} & s_t^+ + S_t^+d \geq 0,\; s_t^- + S_t^-d \geq 0\,,\; \forall\, d\in\mathcal{U},\; \forall\, t \\ & s_t^+ + S_t^+d \geq y_1 + \sum_{t'=1}^{t} x_{t'} + X_{t'}V_{t'}d - d_{t'}\,,\; \forall\, d\in\mathcal{U},\; \forall\, t \\ & s_t^- + S_t^-d \geq -y_1 + \sum_{t'=1}^{t} d_{t'} - (x_{t'} + X_{t'}V_{t'}d)\,,\; \forall\, d\in\mathcal{U},\; \forall\, t \\ & 0 \leq x_t + X_tV_td \leq M\,,\; \forall\, d\in\mathcal{U},\; \forall\, t\,,\end{array}$$

where each $X_t \in \mathbb{R}^{1\times m}$, $S_t^+ \in \mathbb{R}^{1\times m}$, $S_t^- \in \mathbb{R}^{1\times m}$, and where $V_t := \left[\begin{array}{cc} \boldsymbol{I}_{t-1} & \mathbf{0}_{t,T-t+1} \\ \mathbf{0}_{T-t\times t-1} & \mathbf{0}_{T-t\times T-t}\end{array}\right]$ such that $V_td = \left[\begin{array}{cccccc} d_1 & \dots & d_{t-1} & 0 & \dots & 0\end{array}\right]$.

Let's implement this model in RSOME !

Implementation in RSOME



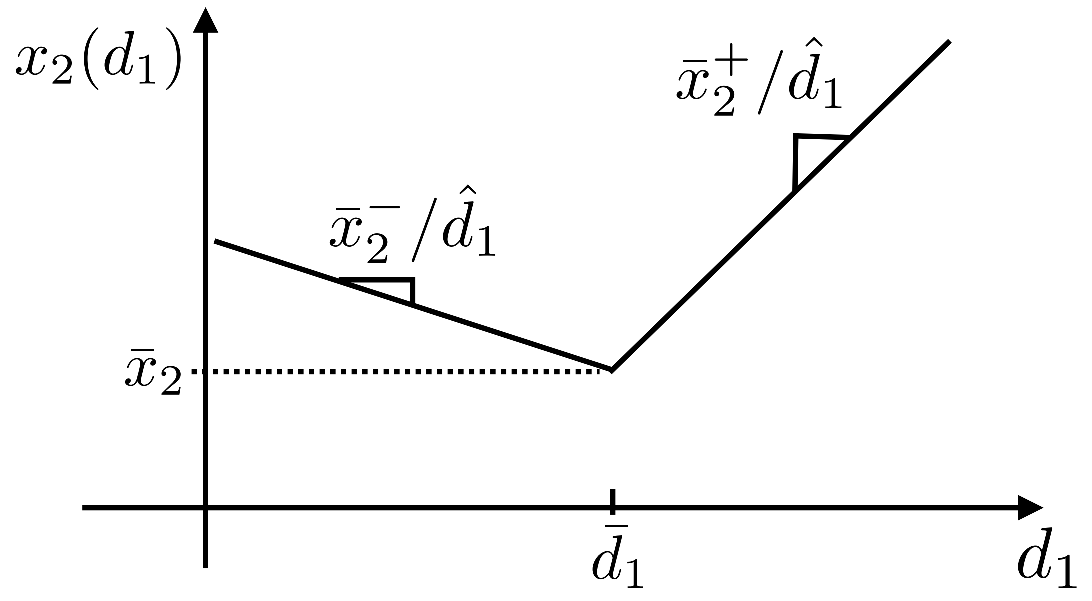
```python
# begin RSOME
model = ro.Model('AARC')

# Define uncertain factors
z = model.rvar(T)
budgetSet = (z<=1, z>=-1,  #each parameter is between [-1, 1]
             rso.norm(z,1)<=Gamma)   # Budget of uncertainty approach

# Demand relations with uncertain factors
d = mus + deltas*z

# Define decision rules
splus=model.ldr(T) #storage variable
sminus=model.ldr(T) #backlog variable
x=model.ldr(T) #ordering decision

# Define decision rules dependance
splus.adapt(z)
sminus.adapt(z)
for t in range(1,T):
    x[t].adapt(z[0:t])

#objective function
model.minmax(c@x + h@splus + b@sminus, budgetSet)
```

```python
#constraints for linearized storage and backlog costs
#model.st(x[0]== 1000)
model.st(splus>= 0)
model.st(sminus>= 0)
for t in range(0,T):
    model.st((splus[t]>= y1 + x[0:t+1].sum()-d[0:t+1].sum()).forall(budgetSet))
    model.st((sminus[t]>= -(y1 + x[0:t+1].sum()-d[0:t+1].sum())).forall(budgetSet))


#constraint on maximum order
model.st(x>=0)
model.st(x<=M)

model.solve(msk) # solve the model
obj_val = model.get()
x_val = x.get()

obj_val_AARC = obj_val;
x_val_AARC = x_val;
```

Application to Inventory Management Problem

- At time t=2, we wish the order to be proportional to how much the demand deviated from the nominal value with different rates if deviation is positive or negative

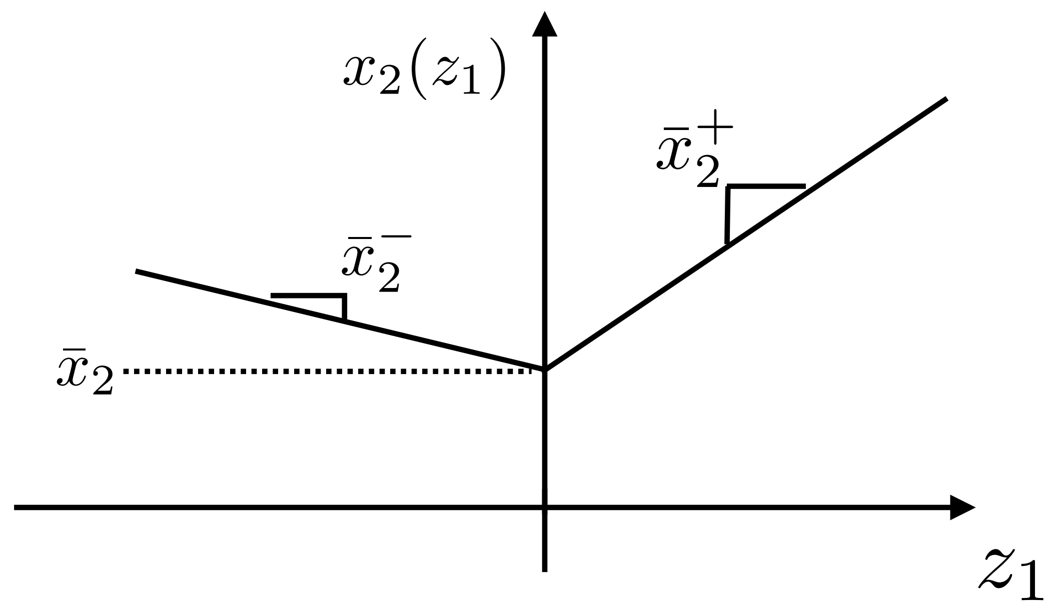
$$x_2(d_1) := \bar{x}_2 + \bar{x}_2^+ \max(0; (d_1 - \bar{d}_1)/\hat{d}_1) + \bar{x}_2^- \max(0; (\bar{d}_1 - d_1)/\hat{d}_1)$$



Application to Inventory Management Problem

- At time t=2, we wish the order to be proportional to how much the demand deviated from the nominal value with different rates if deviation is positive or negative

$$x_2(d_1) := \bar{x}_2 + \bar{x}_2^+ \max(0; z_1) + \bar{x}_2^- \max(0; -z_1)$$



Application to Inventory Management Problem

- At time t=2,

$$x_2(d_1) := \bar{x}_2 + \bar{x}_2^+ \max(0; z_1) + \bar{x}_2^- \max(0; -z_1)$$

- At time t=3,

$$x_3(d_{[2]}) := \bar{x}_3 + \bar{x}_{31}^+ \max(0; z_1) + \bar{x}_{31}^+ \max(0; -z_1)$$
$$+ \bar{x}_{32}^+ \max(0; z_2) + \bar{x}_{31}^+ \max(0; z_2)$$

…

- At time t=T,

$$x_T(d_{[T-1]}) := \bar{x}_T + \bar{x}_{T1}^+ \max(0; z_1) + \bar{x}_{T1}^+ \max(0; -z_1)$$
$$\cdots + \bar{x}_{T,T-1}^+ \max(0; z_{T-1}) + \bar{x}_{3,T-1}^+ \max(0; z_{T-1})$$

Lifted AARC model

- In order to reduce conservativeness, one can attempt to use piecewise linear decision rules:

$$x_t(v_t(z)) := \bar{x}_t + \sum_{k=1}^{\nu} \theta_{tk}^{+} \max(0,\ v_{tk}(z)) + \theta_{tk}^{-} \max(0,\ -v_{tk}(z))$$

- When z is progressively revealed, we obtain:

$(LAARC)$

$$\begin{array}{ll} \underset{\{x_t\}_{t=1}^T, \{X_t^+, X_t^-\}_{t=2}^T}{\text{maximize}} & \inf_{(z,z^+,z^-)\in\mathcal{Z}'} \; c_1(z)^T x_1 + \sum_{t=1}^{T} c_t(z)^T (x_t + X_t^+ V_t z^+ + X_t^- V_t z^-) + d(z) \\ \text{subject to} & a_{j1}^T x_1 + \sum_{t=1}^{T} a_{jt}(z)^T (x_t + X_t^+ V_t z^+ + X_t^- V_t z^-) \leq b_j(z)\,,\ \forall\,(z, z^+, z^-) \in \mathcal{Z}'\,,\ \forall\, j \end{array}$$

where

$$\mathcal{Z}' := \{(z, z^+, z^-) \in \mathbb{R}^{3m} \,|\, z \in \mathcal{Z},\ z_i^+ = \max(0; z_i),\ z_i^- = \max(0; -z_i),\ \forall\, i = 1, \ldots, m\}$$

but this lifted uncertainty set is not convex.

Convexification of LAARC

Theorem 5.7. : *The Lifted AARC model is equivalent to*

$$\begin{array}{ll} \underset{\{x_t\}_{t=1}^T,\{X_t^+,X_t^-\}_{t=2}^T}{\text{maximize}} & \inf_{(z,z^+,z^-)\in\mathcal{Z}''} \; c_1(z)^T x_1 + \sum_{t=1}^T c_t(z)^T (x_t + X_t^+ V_t z^+ + X_t^- V_t z^-) + d(z) \\ \text{subject to} & a_{j1}^T x_1 + \sum_{t=1}^T a_{jt}(z)^T (x_t + X_t^+ V_t z^+ + X_t^- V_t z^-) \leq b_j(z)\,,\; \forall\,(z,z^+,z^-)\in\mathcal{Z}''\,,\;\forall\, j \end{array}$$

where $\mathcal{Z}'' := ConvexHull(\mathcal{Z}')$. Therefore, if the convex hull of $\mathcal{Z}'$ can be described with a finite number of linear constraints then the Lifted AARC model can be solved effectively.

- This result exploits the fact that with linear functions there is always a worst-case solution on one of the vertices of the uncertainty set (see Exercise 2.1).

Convex Hull is Representable for Budgeted Uncertainty Set

Proposition 5.10. : *Let $\mathcal{Z}$ be the budgeted uncertainty set. Then the uncertainty set $ConvexHull(\mathcal{Z}')$ can be represented using the following tractable form*

$$ConvexHull(\mathcal{Z}') = \left\{ (z, z^+, z^-) \in \mathbb{R}^{3m} \;\middle|\; \begin{array}{c} z^+ + z^- \leq 1 \\ \sum_{i=1}^m z_i^+ + z_i^- \leq \Gamma \\ z = z^+ - z^- \\ 0 \leq z^+ \\ 0 \leq z^- \end{array} \right\}.$$

Application to Inventory Management Problem

$$\min_{\substack{x_1, \{x_t, X_t^+, , X_t^-\}_{t=2}^T, \\ \{r_t, R_t^+, R_t^-\}_{t=1}^T, \\ \{s_t, S_t^+, S_t^-\}_{t=1}^T}} \quad \sup_{(z^+,z^-)\in\mathcal{Z}''} \; c_1 x_1 + \textstyle\sum_t c_t(x_t + X_t^+ z_{[t-1]}^+ + X_t^- z_{[t-1]}^-) \\ + h_t(r_t + R_t^+ z^+ + R_t^- z^-) + b_t(s_t + S_t^+ z^+ + S_t^- z^-)$$

$$\text{subject to} \quad r_t + R_t^+ z^+ + R_t^- z^- \geq 0,\; s_t + S_t^+ z^+ + S_t^- z^- \geq 0\,,\; \forall\,(z^+,z^-)\in\mathcal{Z}'',\; \forall$$

$$r_t + R_t^+ z^+ + R_t^- z^- \geq y_1 + \sum_{t'=1}^{t} x_{t'} + X_{t'}^+ z_{[t'-1]}^+ + X_{t'}^- z_{[t'-1]}^- - d_{t'}(z^+,z^-)\,,\; \forall\,(z^+,z^-)\in\mathcal{Z}'',\; \forall\, t$$

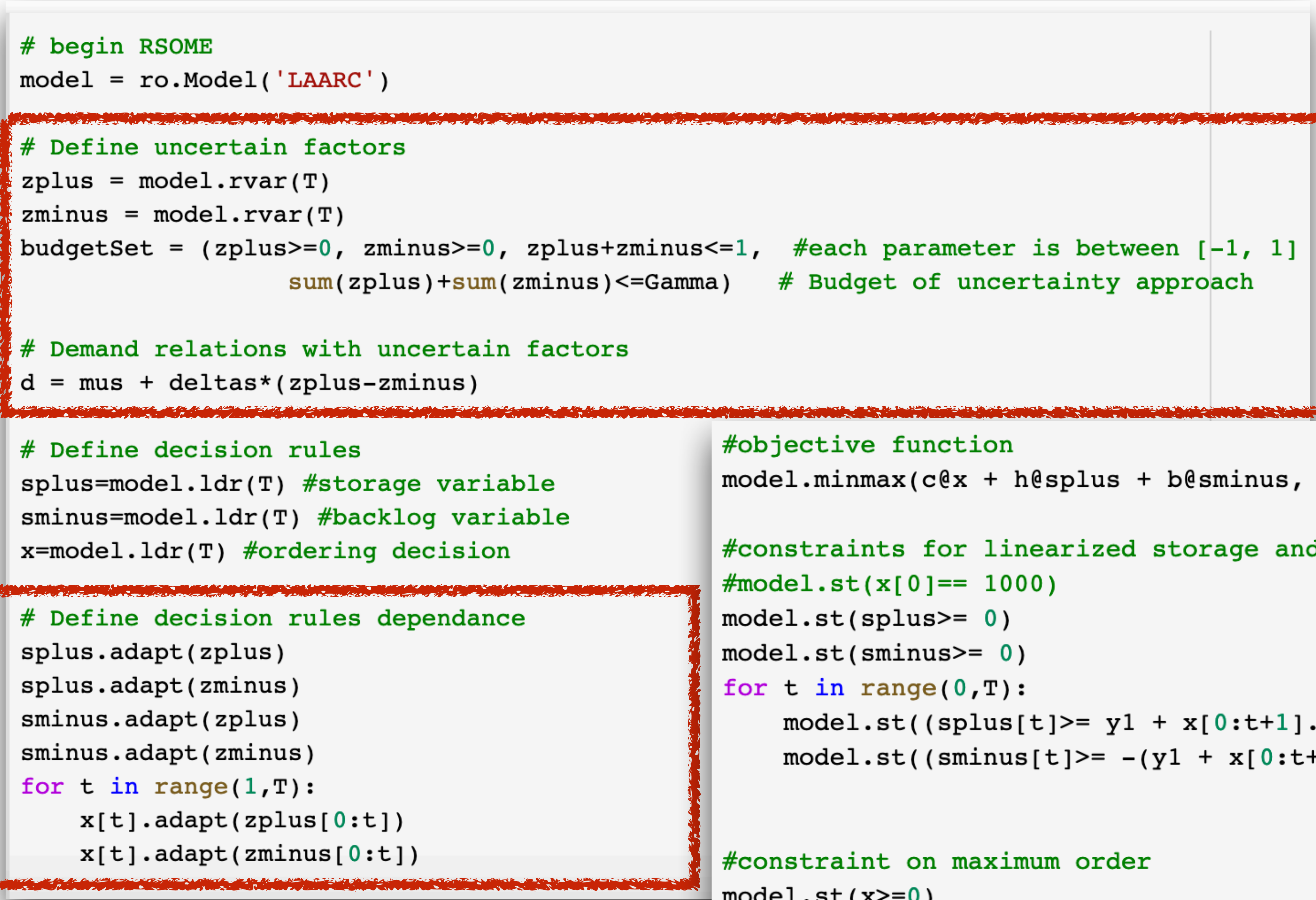
$$s_t + S_t^+ z^+ + S_t^- z^- \geq -y_1 + \sum_{t'=1}^{t} d_{t'}(z^+,z^-) - (x_{t'} + X_{t'}^+ z_{[t'-1]}^+ + X_{t'}^- z_{[t'-1]}^-)\,,\; \forall\,(z^+,z^-)\in\mathcal{Z}'',\; \forall\, t$$

$$0 \leq x_t + X_t^+ z_{[t-1]}^+ + X_t^- z_{[t-1]}^- \leq M\,,\; \forall\,(z^+,z^-)\in\mathcal{Z}'',\; \forall\, t\,,$$

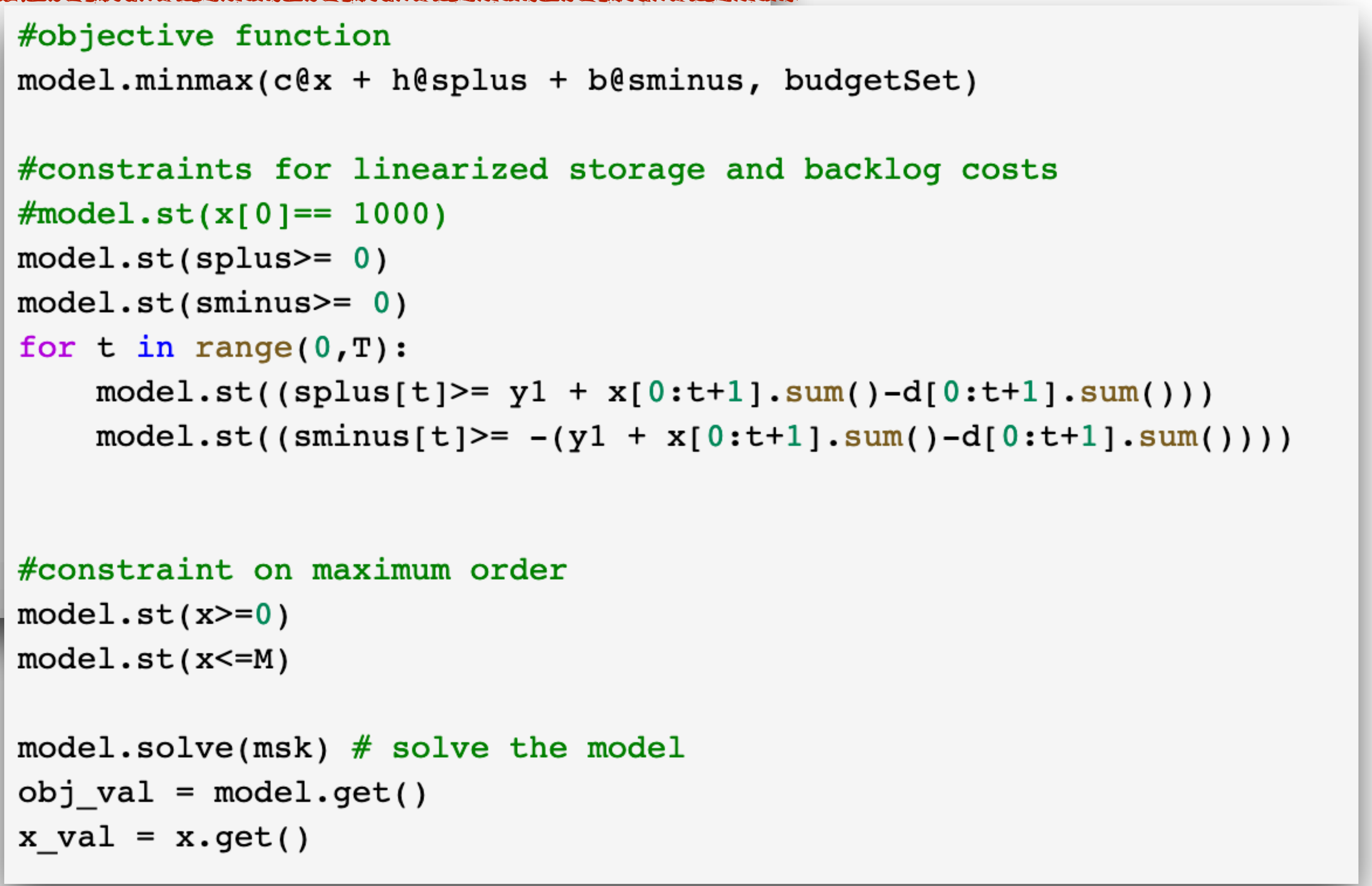
where $d_t(z^+,z^-) := \bar{d}_j + \hat{d}_j(z_j^+ - z_j^-)$ and where

$$\mathcal{Z}' := \{(z^+,z^-)\in\mathbb{R}^{2m} \,|\, z^+ \geq 0,\; z^- \geq 0,\; z^+ + z^- \leq 1,\; \sum_i z_i^+ + z_i^- \leq \Gamma\}\,.$$

Implementation in RSOME



```
# begin RSOME
model = ro.Model('LAARC')

# Define uncertain factors
zplus = model.rvar(T)
zminus = model.rvar(T)
budgetSet = (zplus>=0, zminus>=0, zplus+zminus<=1,  #each parameter is between [-1, 1]
                  sum(zplus)+sum(zminus)<=Gamma)   # Budget of uncertainty approach

# Demand relations with uncertain factors
d = mus + deltas*(zplus-zminus)

# Define decision rules
splus=model.ldr(T) #storage variable
sminus=model.ldr(T) #backlog variable
x=model.ldr(T) #ordering decision

# Define decision rules dependance
splus.adapt(zplus)
splus.adapt(zminus)
sminus.adapt(zplus)
sminus.adapt(zminus)
for t in range(1,T):
    x[t].adapt(zplus[0:t])
    x[t].adapt(zminus[0:t])
```



```
#objective function
model.minmax(c@x + h@splus + b@sminus, budgetSet)

#constraints for linearized storage and backlog costs
#model.st(x[0]== 1000)
model.st(splus>= 0)
model.st(sminus>= 0)
for t in range(0,T):
    model.st((splus[t]>= y1 + x[0:t+1].sum()-d[0:t+1].sum()))
    model.st((sminus[t]>= -(y1 + x[0:t+1].sum()-d[0:t+1].sum())))


#constraint on maximum order
model.st(x>=0)
model.st(x<=M)

model.solve(msk) # solve the model
obj_val = model.get()
x_val = x.get()
```