Homework Assignment #1

Optimization Under Uncertainty – 096335

April 17, 2023

Problem 1

(a) Consider the following uncertain linear robust optimization problem

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^3}{\text{minimize}} \quad & \mathbf{c}^T\mathbf{x} \\ \text{subject to} \quad & \mathbf{a}(\boldsymbol{\xi})^T\mathbf{x} \le b, \quad \forall \boldsymbol{\xi} \in \mathcal{U}, \\ & \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \ge 0. \end{aligned}$$

Find the robust counterpart (RC) of the problem, where the uncertainty set is given as

$$\mathcal{U} = \left\{ \boldsymbol{\xi} \in \mathbb{R}^3 \colon \mathbf{a}(\boldsymbol{\xi}) = \mathbf{a}_0 + \boldsymbol{\xi},\ \|\boldsymbol{\xi}\|_1 \le 3,\ \|\boldsymbol{\xi}\|_\infty \le 2 \right\},$$

where $\mathbf{a}_0 \in \mathbb{R}^3$ is a given nominal data.

(b) Consider the following uncertain linear optimization problem

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad & \mathbf{c}^T\mathbf{x} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} \le \boldsymbol{\xi}, \quad \forall \boldsymbol{\xi} \in \mathcal{U}, \end{aligned}$$

where the uncertainty set is a polyhedron of the form $\mathcal{U} = \{\boldsymbol{\xi} \in \mathbb{R}^m \colon \mathbf{D}\boldsymbol{\xi} \ge \mathbf{d}\}$. Find the robust counterpart (RC) of the problem.

Hint: Treat each of the m constraints separately and use the notion of duality as seen in Lecture 1, slides 16–20.

Problem 2

Consider the following min-cost flow problem: we have a graph $G = (V, E)$ with nodes $V = \{1, 2, \ldots, n\}$ and we must send a unit of flow from node 1 to node n. Each edge $(i, j) \in E$ has capacity C_{ij} and a per-unit flow cost of c_{ij} . The min-cost flow problem corresponds to the following linear optimization problem:

$$\begin{aligned} \underset{x_{ij}\in\mathbb{R}}{\text{minimize}} \quad & \sum_{(i,j)\in E} c_{ij}x_{ij} \\ \text{subject to} \quad & \sum_{(1,j)\in E} x_{1j} = 1 \\ & \sum_{(i,n)\in E} x_{in} = 1 \\ & \sum_{\{i\colon (i,j)\in E\}} x_{ij} = \sum_{\{k\colon (j,k)\in E\}} x_{jk}, \quad \forall j \in V \setminus \{1, n\}, \\ & 0 \le x_{ij} \le C_{ij}, \quad \forall (i,j) \in E, \end{aligned}$$

where the decision variable x_{ij} gives the flow sent from node i to node j.

(a) Consider the case that the flow costs c_{ij} are not known exactly, but instead are uncertain. Formulate the robust optimization problem for each of the following uncertainty sets and find the corresponding robust counterpart.

1. Box uncertainty: $\mathcal{U} = \left\{ \mathbf{c} \in \mathbb{R}^{|E|} \colon c_{ij} = \mu_{ij} + \delta_{ij}\gamma_{ij},\ \|\gamma\|_\infty \leq \Gamma \right\}$.
2. Polyhedral uncertainty: $\mathcal{U} = \left\{ \mathbf{c} \in \mathbb{R}^{|E|} \colon c_{ij} = \mu_{ij} + \delta_{ij}\gamma_{ij},\ \|\gamma\|_1 \leq \Gamma \right\}$.
3. Ellipsoidal uncertainty: $\mathcal{U} = \left\{ \mathbf{c} \in \mathbb{R}^{|E|} \colon c_{ij} = \mu_{ij} + \delta_{ij}\gamma_{ij},\ \|\gamma\|_2 \leq \Gamma \right\}$.

Hint: in order to deal with the uncertainty in the cost vector $\mathbf{c}$, recall that any optimization problem of the form

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad & \mathbf{c}^T\mathbf{x} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} \geq \mathbf{b}, \end{aligned}$$

is equivalent to

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^n, v\in\mathbb{R}}{\text{minimize}} \quad & v \\ \text{subject to} \quad & \mathbf{c}^T\mathbf{x} \leq v, \\ & \mathbf{A}\mathbf{x} \geq \mathbf{b}. \end{aligned}$$

(b) Suppose that G is a bidirectional fully connected graph, with all edge capacities set to $C_{ij} = \frac{1}{2}$. Suppose that the nominal costs μ_{ij} are drawn from Uniform $(0, 10)$, and the deviations δ_{ij} are drawn from Uniform $(0, \mu_{ij})$. Generate a single instance of the problem (μ and δ).

1. Implement and solve the robust optimization problems for $n = 10$ and $\Gamma = 1$ (meaning, find their optimal solution and value) from section (a) using `python` (use `PICOS` as the optimization interface, with `CVXOPT`, `Gurobi` or `CVXPY` at the optimization solvers), for each of the three uncertainty sets and the nominal model where $c_{ij} = \mu_{ij}$ for all $(i, j) \in E$.

   Instructions: in order to create the graph G, you can use the `python` package `networkx`, which is compatible with `PICOS`. Additionally, `PICOS` can generate maximum flow problems. Click here for more information. You can use the attached code that implements the nominal model using `PICOS` and `networkx`.

```python
import picos as pic
import networkx as nx

def flow_problem_nominal(n, Gamma, G, mu, delta):  # n is the number of vertices, Gamma is a parameter, G is a networkx graph, mu and delta are defined as in the exercise

    P = pic.Problem()  # generate an optimization problem P using PICOS

    # adding the flow variables to the problem P
    x = {}
    for e in G.edges():
        x[e] = P.add_variable('x[{0}]'.format(e), 1)  # for each edge e in the graph create a scalar variable named 'x[e]'

    v = P.add_variable('v', 1)  # v is a scalar variable for the value of the flow
    z = P.add_variable('z', 0)  # z is a scalar auxiliary variable initialized as 0

    # Creating the flow constraint
    flowCons = P.add_constraint(pic.flow_Constraint(
        G, x, '1', str(n), 1, capacity='capacity', graphName='G'))  # '1' is the source node, str(n) is the sink node, with capacity as set in G

    # Creating the uncertainty constraint z <= v where z = c^T * x and c can be uncertain (as suggested in the hint in section (a)), and this is equivalent to: minimize c^T*x
    # For the nominal scenario - this constraint is actually certain since c is fixed at mu
    for e in G.edges():
        z = z + mu[e] * x[e]
    P.add_constraint(z <= v)

    # Solving the problem
    P.set_objective('min', v)  # set a objective function for minimization
```

```
    sol = P.solve(verbose=0, solver='cvxopt')  # set solver
    flow = pic.tools.eval_dict(x)

    # print(P)
    print('The solution found is', sol["status"])
    for e in G.edges():
        print('For the edge', str(e), 'the value is', flow[e])
    print('The optimal objective value is: ', P.obj_value())

    return flow, P.obj_value()


# Create the graph
n = 10  # This must be changed manually
G = nx.DiGraph()  # creates a directed graph
for i in range(1, n + 1):  # creates the directed edges of the graph
    for j in range(1, n + 1):
        G.add_edge(str(i), str(j), capacity=1 / 2)  # the edges are ('i', '
    j', capacity)
```

2. Repeat the experiments for $n = 20, 30, 40, 50, 60$ (while $\Gamma = 1$), and also for $\Gamma = 0.9, 0.6, 0.3, 0.1$ (while $n = 10$). For each of the four models, what happens to the robust total cost (meaning, the optimal value of the problem) while n increases? What happens while Γ decreases? Add a supporting plot or a table, and explain your results.
3. Assume that $n = 10$ and $\Gamma = 1$. Using the optimal solutions obtained in the previous section, approximate now the total cost (meaning, $\mathbf{c}^T\mathbf{x}$ where $\mathbf{x}$ is the obtained optimal solution), for each of the four models. Do this by generating running 10000 realizations of the uncertain per-unit flow costs, where in each realization the per-unit flow costs come from the distribution

$$c_{ij} \sim 2 \cdot \mathrm{Beta}\left(\alpha, \beta\right) \cdot \delta_{ij} + \left(\mu_{ij} - \delta_{ij}\right),$$

for $(\alpha, \beta) = \left(\frac{1}{2}, \frac{1}{2}\right), (5, 1), (2, 2), (2, 5)$ (40000 realizations in total), using the single instance of μ and δ you generated earlier.

For each of the four models, and for each of the four distributions, write down the average, standard deviation and mode of the total costs across the realizations, and also the (estimated) probability that the total cost is higher than the predicted on (meaning, the percentage of the realizations in which $\mathbf{c}^T\mathbf{x} > v$, where v is the optimal value of the robust problem from previous section). Add supporting tables and provide explanations for the trends you observe.

Problem 3

Consider the optimization problem

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad & \mathbf{c}^T\mathbf{x} \\ \text{subject to} \quad & \mathbf{a}\left(\boldsymbol{\xi}\right)^T\mathbf{x} \leq b\left(\boldsymbol{\xi}\right), \quad \forall \boldsymbol{\xi} \in \mathcal{U}, \end{aligned}$$

where the uncertainty set $\mathcal{U}$ is given by

$$\mathcal{U} = \left\{ \boldsymbol{\xi} \in \mathbb{R}^{n+1} \colon \begin{bmatrix} \mathbf{a}\left(\boldsymbol{\xi}\right) \\ b\left(\boldsymbol{\xi}\right) \end{bmatrix} = \begin{bmatrix} \mathbf{a}^0 \\ b^0 \end{bmatrix} + \mathbf{P}\boldsymbol{\xi} \quad \text{where} \quad \boldsymbol{\xi}^T\mathbf{Q}_j\boldsymbol{\xi} \leq \rho^2 \quad \forall 1 \leq j \leq J \right\},$$

for some $\mathbf{P} \in \mathbb{R}^{(n+1)\times(n+1)}$ and $\mathbf{Q}_j \in \mathbb{R}^{(n+1)\times(n+1)}$ such that $\mathbf{Q}_j \succeq 0$ and $\sum_{j\in J} \mathbf{Q}_j \succ 0$, $\rho > 0$ and a finite set J. Find a robust counterpart (RC) of the problem.

Hint: Use duality in order to solve the optimization problem in the constraint, and explain why strong duality holds.

Problem 4

The problem here is based on the one found in https://stoprog.org/SavedLinks/IBM_StoExt_problems/node4.php#SECTION00412.

The problem consists of determining the product mix for a furniture shop with two workstations $j = 1, 2$: carpentry and finishing. The availability of labor in man-hours at the two stations is limited. There are four product classes $i = 1, 2, 3, 4$, each consuming a certain number of man-hours at the two stations. Each product earns a certain profit and the shop has the option to purchase labor from outside. The objective is to maximize the profit.

We consider the following variables: x_i is the amount of product class i produced, v_j are the man-hours purchased from outside for workstation j. The parameters are: c_j is the profit per unit of product class j, q_j is the cost per man-hour for labor for workstation j, $\tilde{t}_{ji}$ are random man-hours at work station j required per unit of product class i and $\tilde{h}_j$ are random man-hours available at workstation j. We can not purchase more than 600 man hours for each workstation.

The problem can be formulated as:

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^4,\mathbf{v}\in\mathbb{R}^2}{\text{maximize}} \quad & \mathbf{c}^T\mathbf{x} - \mathbf{q}^T\mathbf{v} \\ \text{subject to} \quad & \tilde{\mathbf{t}}_j\mathbf{x} \le \tilde{h}_j + \mathbf{v}_j, \ \forall j = 1, 2, \\ & \mathbf{x}, \ \mathbf{v} \ge 0, \\ & \mathbf{v} \le 600. \end{aligned}$$

We can eliminate the variable $\mathbf{v}$ and obtain an equivalent problem:

$$\text{(P)} \qquad \begin{aligned} -\underset{\mathbf{x}\in\mathbb{R}^4}{\text{minimize}} \quad & \max\left\{ q_1\left(\tilde{\mathbf{t}}_1^T\mathbf{x} - \tilde{h}_1\right), q_2\left(\tilde{\mathbf{t}}_2^T\mathbf{x} - \tilde{h}_2\right), \sum_{j=1}^{2} q_j\left(\tilde{\mathbf{t}}_j\mathbf{x} - \tilde{h}_j\right), 0 \right\} - \mathbf{c}^T\mathbf{x} \\ \text{subject to} \quad & \tilde{\mathbf{t}}_j^T\mathbf{x} - \tilde{h}_j \le 600, \ \forall j = 1, 2, \\ & \mathbf{x} \ge 0. \end{aligned}$$

The problem's parameters are as follows:

$$\begin{aligned} \mathbf{c} &= \begin{pmatrix} 12, & 20, & 18, & 40 \end{pmatrix}, \\ \mathbf{q} &= \begin{pmatrix} 1, & 1 \end{pmatrix}, \end{aligned}$$

and $\boldsymbol{\xi} \equiv \begin{pmatrix} \tilde{\mathbf{t}} \\ \tilde{\mathbf{h}} \end{pmatrix} \in \mathbb{R}^{10}$ is the vector of uncertain parameters, where $\tilde{\mathbf{t}}$ is the concatenation of $\tilde{\mathbf{t}}_1$ and $\tilde{\mathbf{t}}_2$ into a single column vector. Assume that the upper and lower bounds of $\tilde{\mathbf{t}}$ and $\tilde{\mathbf{h}}$ define a box with center $\boldsymbol{\xi}^0 = (4, 9, 7, 10, 1, 1, 3, 40, 6000, 4000) \in \mathbb{R}^{10}$ and width $\mathbf{b} = \left(\frac{1}{2}, 1, 1, 1, \frac{1}{5}, \frac{1}{5}, \frac{1}{2}, 4, 300, 150\right) \in \mathbb{R}^{10}$. We define the following uncertainty sets:

$$\begin{aligned} \mathcal{U}_{\text{objective}} &= \left\{ \boldsymbol{\xi} \equiv \begin{pmatrix} \tilde{\mathbf{t}} \\ \tilde{\mathbf{h}} \end{pmatrix} \in \mathbb{R}^{10} \colon \left|\boldsymbol{\xi} - \boldsymbol{\xi}^0\right| \le \mathbf{b} \right\}, \\ \mathcal{U}_{\text{constraints}} &= \left\{ \boldsymbol{\xi} \equiv \begin{pmatrix} \tilde{\mathbf{t}} \\ \tilde{\mathbf{h}} \end{pmatrix} \in \mathbb{R}^{10} \colon \left|\boldsymbol{\xi} - \boldsymbol{\xi}^0\right| \le \mathbf{b}, \ \left\|\mathbf{B}^{-1}\left(\boldsymbol{\xi} - \boldsymbol{\xi}^0\right)\right\| \le r \right\}. \end{aligned}$$

where $\mathbf{B} = \operatorname{diag}(\mathbf{b})$ and for some $r \ge 0$, where the set $\mathcal{U}_{\text{objective}}$ is the uncertainty set of the uncertain parameters $\tilde{\mathbf{t}}$ and $\tilde{\mathbf{h}}$ in the objective function of Problem (P), and the set $\mathcal{U}_{\text{constraints}}$ is the uncertainty set of the same uncertain parameters in the constraints.

(a) Define suitable matrices $\mathbf{S}_j$, $j = 1, 2, 3, 4$, such that the uncertain Problem (P) can be written equivalently as

$$\begin{aligned} -\underset{\mathbf{x}\in\mathbb{R}^5, z\in\mathbb{R}}{\text{minimize}} \quad & -\mathbf{c}^T\mathbf{x} + z \\ \text{s.t.} \quad & \max_{\boldsymbol{\xi}\in\mathcal{U}_{\text{objective}}} \left\{\mathbf{x}^T\mathbf{S}_j\boldsymbol{\xi}\right\} \le z, \quad \forall j = 1, 2, 3, 4, \\ & \max_{\boldsymbol{\xi}\in\mathcal{U}_{\text{constraints}}} \left\{\mathbf{x}^T\mathbf{S}_j\boldsymbol{\xi}\right\} \le 600, \quad \forall j = 1, 2, 3, 4, \\ & \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4 \ge 0, \quad \mathbf{x}_5 = -1. \end{aligned}$$

Explain your derivations according to what we saw in the lectures and tutorials. In addition, what could be the advantages of defining different uncertainty sets for the objective function and for the constraints?

(b) Find the robust counterpart of Problem (P) from section (a).

Hint: The uncertainty set $\mathcal{U}_{\text{objective}}$ is equivalent to

$$\mathcal{U}_{\text{objective}} = \left\{ \boldsymbol{\xi} \in \mathbb{R}^{10} \colon \boldsymbol{\xi} = \boldsymbol{\xi}^0 + \mathbf{b} \right\},$$

which is a box uncertainty in the objective function of the form introduced in Problem 2(a).

(c) For any value of r in the range `numpy.linspace(0, 2.5, num = 70)`, implement and solve the robust optimization problem from the previous section (meaning, find its optimal solution $\mathbf{x}$ and its optimal value).

For implementing and solving the optimization problem, you can use the `CVXPY` package, that provides interface and solves convex optimization problems (similarly to `PICOS` with `CVXOPT`). See https://www.cvxpy.org/ for `CVXPY` documentation.

Create a graph that plots the change in profit (optimal value) of the robust problem, as a function of r. Explain your results.

(d) Now, assume that the uncertain parameters come from the following distributions: $\tilde{h}_1$ follows a truncated normal distribution with mean 6000 and variance $\sigma^2 = 100$ supported on $[5700, 6300]$, and $\tilde{h}_2$ follows a truncated normal distribution with mean 4000 and variance $\sigma^2 = 50$ supported on $[3850, 4150]$. Meaning,

$$\tilde{\mathbf{h}} = \big(5700 \leq N(6000, 100) \leq 6300, \quad 3850 \leq N(4000, 50) \leq 4150\big).$$

In addition, $\tilde{\mathbf{t}}$ comes from

$$\begin{aligned}
\tilde{\mathbf{t}}_1 &= \big(U(3.5, 4.5), \quad U(8, 10), \quad U(6, 8), \quad U(9, 11)\big), \\
\tilde{\mathbf{t}}_2 &= \big(U(0.8, 1.2), \quad U(0.8, 1.2), \quad U(2.5, 3.5), \quad U(36, 44)\big).
\end{aligned}$$

Using the optimal solutions obtained in the previous section, approximate the expected profit of the solutions for each value of r. That is, the value of the objective function of Problem (P), where $\mathbf{x}$ is the obtained optimal solution. Do this by generating 10000 realizations of the uncertain parameters, and calculate the profit of the given solution for each realizations.

For each of the values of r, write down the average and standard deviation of profit across the realizations, and also the (estimated) probability that the profit is smaller than the predicted one (meaning, the percentage of the realizations in which the profit is below the optimal profit value of the robust problem from previous section). Add supporting graphs or tables and provide explanations for the trends you observe.