15.093/6.215 Optimization Methods

Problem Set 3

Due: November 4, 2021

Problem 1: Solving a network simplex problem. (10 points) Bertsimas & Tsitsiklis, Exercise 7.9.

Solution. See attached pdf.

Problem 2: Turning robust optimization constraint into tractable linear constraints. (10 points) In this problem, we consider the robust optimization constraint of the form:

$$\left(\bar{a} + \overline{P}z\right)^\top x \leq b, \quad \forall z \in U, \tag{ROC}$$

where $\bar{a} \in \mathbf{R}^n, \overline{P} \in \mathbf{R}^{n\times p}, b \in \mathbf{R}$ are known problem data, $x \in \mathbf{R}^n$ is the decision variable, but the vector $z \in \mathbf{R}^p$ is uncertain and lies in some uncertainty set U. For the following uncertainty sets, convert (ROC) into a set containing linear constraints. *Hint: Duality Trick.*

(a) Box uncertainty set: $U = \left\{z \in \mathbf{R}^p \mid \|z\|_\infty = \max_{i\in\{1,\ldots,p\}} |z_i| \leq \rho\right\}$, where $\rho > 0$ is a known constant. (3 points)

(b) Polyhedral uncertainty set: $U = \{z \in \mathbf{R}^p \mid Dz \leq d\}$, where $D \in \mathbf{R}^{m\times p}, d \in \mathbf{R}^m$ are known. The set U is known to be nonempty and bounded. (4 points)

(c) l_1−uncertainty set: $U = \{z \in \mathbf{R}^p \mid \|z\|_1 = \sum_{i=1}^p |z_i| \leq \rho\}$, where $\rho > 0$ is a known constant. (3 points)

Solution. First, let us rewrite (ROC) as follows:

$$\max_{z\in U} \left(\underbrace{\bar{P}x}_{=c_x}\right)' z \leq \underbrace{b - \bar{a}'x}_{=d_x}.$$

So, for the rest of the solution we will focus on solving $\max_{z\in U} c_x' z$.

(a) We want to solve,

$$\begin{array}{ll} \text{maximize} & c_x' z \\ \text{subject to} & \|z\|_\infty \leq \rho. \end{array}$$

Note that by setting $z = \rho\mathbf{sgn}(c_x)$, where $\mathbf{sgn}(\cdot)$ is the sign vector for any input vector $(\cdot)$, we can ensure that the objective is maximized. In a compact notation, the optimal value is: $c_x'\rho\mathbf{sgn}(c_x) = \rho\|c_x\|_1$. So the equivalent constraint is:

$$\rho\|c_x\|_1 \leq d_x,$$

and then using the epigraph approach, we can convert it to linear constraints.

(b) The dual of $\max_{z\in U} c_x' z$. is

$$\begin{array}{ll} \text{minimize} & d'\lambda \\ \text{subject to} & D'\lambda = c_x \\ & \lambda \geq 0, \end{array}$$

and as strong duality holds, we have the robust counterpart:

$$\begin{aligned} d'\lambda &\leq b - x'\bar{a} \\ D'\lambda &= p'x \\ \lambda &\geq 0. \end{aligned}$$

(c) In this case, we want to solve:

$$\begin{array}{ll} \text{maximize} & c_x' z \\ \text{subject to} & \|z\|_1 \leq \rho. \end{array}$$

Here, the objective can be interpreted as a weighted average of the $c_x s$, where z_i are the weights to be assigned where $\sum_i |z_i| \leq \rho$. The maximum of the weighted average is obtained by first finding the $c_{x,i}$ with the largest absolute value, assigning maximum weight ρ to the corresponding z_i, and set rest of the components of z to zero. This leads to the optimal value: $\rho\|c_x\|_\infty$. So the equivalent constraint is again:

$$\rho\|c_x\|_\infty \leq d_x,$$

and then using the epigraph approach, we can convert it to linear constraints.

Problem 3: Moving problems. (10 points) Suppose you are planning to move to your new house. You have n items of size a_j, $j = 1, \ldots, n$, that need to be moved. You have rented a truck that has size Q and you have bought m boxes. Box i has size b_i, $i = 1, \ldots, m$. Formulate an integer programming problem in order to decide whether the move is possible. Specify what each integer variable models.

Solution Define decision variables as follows:

$$x_i = \begin{cases} 1 & \text{if box } i \text{ is used} \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ij} = \begin{cases} 1 & \text{if item } j \text{ is put in box } i \\ 0 & \text{otherwise} \end{cases}$$

The boxes we use must fit in the truck, so we require that $\sum_{i=1}^m b_i x_i \leq Q$.

The items in box i must fit in the box, so for each box i we require $\sum_{j=1}^n a_j y_{ij} \leq b_i$.

Each item goes exactly in one box, so for each item j we require $\sum_{i=1}^n y_{ij} = 1$.

Finally, item j can be placed in box i only if box i is being used, so for all i and j we require $y_{ij} \leq x_i$.

Therefore, an integer linear program that determines the minimum number of boxes need to move

$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_{i=1}^m x_i & \\ \text{subject to} & \displaystyle\sum_{i=1}^m b_i x_i \leq Q & \\ & \displaystyle\sum_{j=1}^n a_j y_{ij} \leq b_i & \text{for } i = 1, 2, \ldots, m \\ & \displaystyle\sum_{i=1}^n y_{ij} = 1 & \text{for } j = 1, 2, \ldots, n \\ & y_{ij} \leq x_i & \text{for } i = 1, 2, \ldots, m,\ j = 1, 2, \ldots, n \\ & x_i \in \{0, 1\} & \text{for } i = 1, 2, \ldots, m \\ & y_{ij} \in \{0, 1\} & \text{for } i = 1, 2, \ldots, m,\ j = 1, 2, \ldots, n \end{array}$$

Problem 4: Relax or not? (10 points) Give an MIP formulation of each of the following problems. For each problem, can the integer constraints be relaxed in your formulation? Either prove it or give an example where the optimal solution of your relaxed problem is not integer (you can chose specific numerical values of the problem's parameters).

1. (6 points) We have 3 different coin denominations with values $v_1 = 1$, $v_2 = 2$ and $v_3 = 5$. The objective is to provide exact change for some amount $C \in \mathbb{N}$ using as few coins as possible.
2. (4 points) Let $n > 2$. Suppose there are n people to be assigned to n tasks. Every task has to be completed and each task has to be handled by only one person. Here $c_{ij} > 0$ measures the benefits gained by assigning the person i to the task j. The objective here is to maximize the overall benefits by devising the optimal assignment pattern.

Solution.

1. (6 points) We have 3 different coin denominations with values $v_1 = 1$, $v_2 = 2$ and $v_3 = 5$. The objective is to provide exact change for some amount $C \in \mathbb{N}$ using as few coins as possible.
   Solution.
   $$\begin{aligned} \text{minimize} \quad & n_1 + n_2 + n_3 \\ \text{subject to} \quad & n_1 + 2n_2 + 5n_5 = C \\ & n_1, n_2, n_3 \in \mathbb{N} \end{aligned}$$
   This IP can not be relaxed. For $C = 1$, the optimal integer solution is $(1, 0, 0)$ with cost 1, while the optimal LP solution in $(0, 0, 1/5)$ with cost $1/5 < 1$.
2. (4 points) Let $n > 2$. Suppose there are n people to be assigned to n tasks. Every task has to be completed and each task has to be handled by only one person. Here $c_{ij} > 0$ measures the benefits gained by assigning the person i to the task j. The objective here is to maximize the overall benefits by devising the optimal assignment pattern.
   Solution.
   $$\begin{aligned} \text{minimize} \quad & \sum_{i,j} c_{ij} x_{ij} \\ \text{subject to} \quad & \sum_i x_{ij} = 1 \quad \text{for } j = 1, 2, \ldots, n \\ & \sum_j x_{ij} = 1 \quad \text{for } i = 1, 2, \ldots, n \\ & x_{ij} \in \{0, 1\} \quad \text{for } i, j = 1, 2, \ldots, n \end{aligned}$$
   As seen in the lectures, we can convert this problem to a network flow problem. The problem's data is integer, therefore, the problem can be relaxed.
   Common mistakes. You need to argue properly why it is a network flow problem (like shown in the lecture) by, for example, drawing the corresponding network flow.

Problem 5: The tournament problem. (10 points) Bertsimas & Tsitsiklis, Exercise 7.3.

Solution. Given,

- n: number of teams
- k: number of games
- x_i : number of wins by team i
- X: set of all possible outcome vectors
- $x = (x_1, \ldots, x_n)$

We want to find out given some x, is $x \in X$?

Denote $G = (\mathcal{G}, \mathcal{A})$ where $\mathcal{N} = \{1, \ldots, n\}$, $\mathcal{A} = \{(i, j) \in \mathcal{N} \times \mathcal{N} : i < j\}$. Denote by y_{ij} the number of times team i defeats team j with $i < j$. Fixing team i, total number of times team i defeats teams $j = i+1, \ldots, n$ is: $\sum_{j:(i,j)\in\mathcal{A}} y_{ij}$. Also,k total number of times team i beats team j with $j < i$ is $k - y_{ji}$. Then total number of times team i beats teams $j = 1, \ldots, i-1$ is $(i-1)k - \sum_{j:(j,i)\in\mathcal{A}} y_{ji}$.

So, the total number of wins by team i must be equal to

$$x_i = \sum_{j:(i,j)\in\mathcal{A}} y_{ij} + (i-1)k - \sum_{j:(j,i)\in\mathcal{A}} y_{ji},$$

where we must have $0 \leq y_{ij} \leq k$ for any $(i, j) \in \mathcal{A}$.

So, the resultant network flow problem is:

$$\begin{array}{ll} \text{minimize} & 0 \\ \text{subject to} & x_i = \sum_{j:(i,j)\in\mathcal{A}} y_{ij} + (i-1)k - \sum_{j:(j,i)\in\mathcal{A}} y_{ji} \qquad \forall i \in \mathcal{N} \\ & 0 \leq y_{ij} \leq k, \qquad \forall (i,j) \in \mathcal{A}. \end{array}$$

And if the above problem instance is feasible, then $x \in X$.