University of Waterloo – CO 370 Midterm – Fall 2019

Tue Oct 29, 2019 – 10am to 11:20am (80 minutes)
Closed book – no calculators, no materials allowed

Solutions

Question 1 [3 marks] Circle the correct answer. No justification necessary.

1. In Julia/JuMP, the expression

$$\sum_{i=1}^{5} 3i$$

is written

(a) `sum(i = 1:5, 3 * i)`

(b) **(correct)** `sum(3 * i for i = 1:5)`

(c) `sum(3 * i for i = [1, 5])`

(d) `sum(i = [1, 5], 3 * i)`

2. In Julia/JuMP, the constraint

$$x + y \leq 5$$

can be written (for a model `md` with variables `x` and `y`)

(a) **(correct)** `@constraint(md, x + y <= 5)`

(b) `@constraint(md, x + y, <=, 5)`

(c) `@constraint(md, x + y) <= 5`

(d) `@constraint(md, sum(x, y)) <= 5`

3. In Julia/JuMP, if a variable was declared using `@variable(md, x)` for a model `md`, its value can be accessed by writing

(a) `&x`

(b) `x.value`

(c) `x.value()`

(d) **(correct)** `value(x)`

Question 2 [6 marks] Consider the following linear programming problem

$$\begin{array}{lccccccccc}
\min & & & 3x_2 & + & 2x_3 & + & x_4 & & \\
\text{s.t.} & & & x_2 & + & x_3 & & & \geq & 4 \\
& x_1 & + & x_2 & - & x_3 & + & \boldsymbol{a}\, x_4 & \geq & 4 \\
& & & 2x_2 & - & x_3 & & & \geq & 8 \\
& x_1 & - & x_2 & & & & & \geq & 4 \\
& & & & & x_3 & + & \boldsymbol{b}\, x_4 & \geq & 3 \\
& x_1 & , & x_2 & , & x_3 & , & x_4 & \in & \mathbb{R}
\end{array} \tag{P}$$

where $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}$ are constants. We do not know the value of $\boldsymbol{a}$ and $\boldsymbol{b}$, but we are told that $y^* = \begin{pmatrix} 2, & 0, & 0.5, & 0, & 0.5 \end{pmatrix}^T$ is a dual optimal solution. Let x^* be a primal optimal solution.

1. Write the dual of (P).
2. Find the numerical value of x_2^* and x_3^*.
3. Find the numerical value of the objective function at x^*.
4. Find the numerical value of x_4^*.
5. Find the numerical value of $\boldsymbol{b}$.

Note that $\boldsymbol{a}$ and $\boldsymbol{b}$ can appear in the answer to (1), but the answers to (2)-(5) must be numerical.

Solution: 1. The dual of (P) is

$$\begin{array}{lcccccccccccc}
\max & 4y_1 & + & 4y_2 & + & 8y_3 & + & 4y_4 & + & 3y_5 & & \\
\text{s.t.} & & & y_2 & & & + & y_4 & & & = & 0 \\
& y_1 & + & y_2 & + & 2y_3 & - & y_4 & & & = & 3 \\
& y_1 & - & y_2 & - & y_3 & & & + & y_5 & = & 2 \\
& & & ay_2 & & & & & + & by_5 & = & 1 \\
& y_1 & , & y_2 & , & y_3 & , & y_4 & , & y_5 & \geq & 0
\end{array} \tag{D}$$

2. By complementary slackness, the 1st, 3rd and 5th constraints of (P) are tight. In particular, the 1st and 3rd give

$$\left\{ \begin{array}{ccccc} x_2^* & + & x_3^* & = & 4 \\ 2x_2^* & - & x_3^* & = & 8 \end{array} \right. \Leftrightarrow \left\{ \begin{array}{ccc} x_2^* & = & 4 \\ x_3^* & = & 0 \end{array} \right.$$

3. The objective function in (D) gives us

$$z^* = 4 \cdot 2 + 4 \cdot 0 + 8 \cdot 0.5 + 4 \cdot 0 + 3 \cdot 0.5 = 13.5.$$

By strong duality, z^* is also the optimal objective function value for (P). Using the objective function in (P), we obtain that

$$z^* = 13.5 = 2 \cdot 4 + 2 \cdot 0 + x_4^* \quad \Rightarrow \quad x_4^* = 1.5$$

4. Using the fact that the 5th constraint is tight, we get

$$x_3^* + bx_4^* = 3 \quad \Leftrightarrow \quad 0 + b \cdot 1.5 = 3 \quad \Leftrightarrow \quad b = 2.$$

3. and 4. (alternative approach) We compute b first. The last constraint of (D) is $ay_2 + by_5 = 1$. We know that the given dual solution $y^* = \begin{pmatrix} 2, & 0, & 0.5, & 0, & 0.5 \end{pmatrix}^T$ satisfies this constraint, yielding $a \cdot 0 + b \cdot 0.5 = 1$ hence $b = 2$.

Now, using the fact that the 5th constraint is tight, we get

$$x_3^* + bx_4^* = 3 \quad \Leftrightarrow \quad 0 + 2 \cdot x_4^* = 3 \quad \Leftrightarrow \quad x_4^* = 1.5$$

Question 3 [6 marks] Consider a set of N objects indexed $\{1, \ldots, N\}$. For each pair $\{i, j\}$ of objects, we are given a constant $D_{ij} \in \mathbb{R}$, which measures the "affinity" between the objects. For all $i, j \in \{1, \ldots, N\}$, we have:
(i) $D_{ij} = D_{ji}$,
(ii) $D_{ii} = 0$,
(iii) D_{ij} can be positive, negative or zero.
We want to select a subset of K objects (out of the N objects) that maximizes the sum of the pairwise affinities D_{ij} between the selected objects.

For illustrative purposes, consider an example, with $N = 4$, $K = 3$, and

$$D = \begin{pmatrix} 0 & 3 & 2 & 2 \\ 3 & 0 & -1 & 1 \\ 2 & -1 & 0 & 4 \\ 2 & 1 & 4 & 0 \end{pmatrix}.$$

A subset $\{1, 2, 3\}$ would have a sum $D_{12} + D_{13} + D_{23} = 3 + 2 + (-1) = 4$. However, a subset $\{1, 3, 4\}$ would be better, with a sum $D_{13} + D_{14} + D_{34} = 2 + 2 + 4 = 8$.

Given constants $N > 0$, $K > 0$ and $D_{ij} \in \mathbb{R}$ for all $i, j \in \{1, \ldots, N\}$, model this problem as an integer programming problem. (Remark: your IP must be valid for all N, K and D, not just for the example above, which is given only to illustrate the problem statement.)

Solution:
Variables:

$$x_i = \begin{cases} 1 \text{ if object } i \text{ is taken,} \\ 0 \text{ otherwise,} \end{cases} \quad \text{for } i = 1, \ldots, N.$$

$$z_{ij} = \begin{cases} 1 \text{ if objects } i \text{ and } j \text{ are both taken,} \\ 0 \text{ otherwise,} \end{cases} \quad \text{for } i = 1, \ldots, N, \;\; j = 1, \ldots, i-1$$

Model:

$$\begin{array}{llcll}
\max & \sum_{i=1}^{N} \sum_{j=1}^{i-1} d_{ij} z_{ij} & & & \\
\text{s.t.} & \sum_{i=1}^{N} x_i & = & K & \\
& x_i + x_j & \leq & z_{ij} + 1 & \forall i = 1, \ldots, N, \; \forall j = 1, \ldots, i-1 \\
& x_i & \geq & z_{ij} & \forall i = 1, \ldots, N, \; \forall j = 1, \ldots, i-1 \\
& x_j & \geq & z_{ij} & \forall i = 1, \ldots, N, \; \forall j = 1, \ldots, i-1 \\
& x_j & \in & \{0, 1\} & \forall i = 1, \ldots, N \\
& z_{ij} & \in & \{0, 1\} & \forall i = 1, \ldots, N, \; \forall j = 1, \ldots, i-1
\end{array}$$

Explanation: Regarding the objective function, we maximize the sum of the affinities. Note that we sum over all i, j such that $j < i$, because we defined z_{ij} for those indices only. We could have defined z_{ij} for all i, j, in which case we would have had to simply divide the corresponding sum by two.

The first constraint $\sum_{i=1}^{N} x_i = K$ ensures that we select exactly K objects. The rest of the constraints ensure that z_{ij} is consistent with the value of x_i. Specifically, for every z_{ij}, we must force its value to 1 if both x_i and x_j are 1:

$$z_{ij} \geq x_i + x_j - 1 \tag{1}$$

and we force its value to 0 otherwise:

$$z_{ij} \leq x_i \quad , \qquad z_{ij} \leq x_j \tag{2}$$

or as an alternative

$$z_{ij} \leq \frac{1}{2}(x_i + x_j).$$

Remark that all the above constraints are necessary because, since d_{ij} can be of any sign, we don't know if individual z_{ij} are minimized or maximized. Also, while there is (at least) one alternative way to implement (2), $z_{ij} \geq \frac{1}{2}(x_i + x_j)$ is **not** a valid alternative to (1), since having $x_i = 1$ and $x_j = 0$ would already force $z_{ij} = 1$.