15.093: Integer Programming and Combinatorial Optimization **Spring 2021**

Assignment 1: Integer programming modeling and implementation

Assigned: February 22; Due: March 04, 11:59 pm.

Problem 1: Emergency response [50 pts]

You are planning the deployment of emergency shelters in response to a natural disaster. Given available resources, you have capacity to build only 10 shelters across the city. You have identified 55 potential sites, each with a fixed capacity (i.e., a maximum number of residents that it can assist). All residents will need to access one of the shelters you will construct. You aim to determine which shelters to build to optimize the relief services provided to the population.

The city under consideration can be modeled as a 20×10 miles rectangle, with 200 major residential areas. We index the residential areas by $i = 1, \cdots, 200$ and the shelter sites by $j = 1, \cdots, 55$. We consider the "Manhattan distance": if area i's coordinates are (x_i, y_i) and shelter j's coordinates are in (x_j, y_j), the distance from i to j is given by $d_{ij} = |x_i - x_j| + |y_i - y_j|$.

You have access to the following data files:

- Pb1_areas.csv: A matrix of size 200×3 that indicates, for each area, (i) its x-coordinate, in miles (0 to 20), (ii) its y-coordinate, in miles (0 to 10) and (iii) its number of residents.
- Pb1_shelters.csv: A matrix of size 55×3 that indicates, for each candidate shelter, (i) its x-coordinate, in miles (0 to 20), (ii) its y-coordinate, in miles (0 to 10) and (iii) its capacity.

a. Formulate an integer programming model that optimizes the selection of shelters to minimize the total distance across all residents required to access their assigned shelter. **[10 pts]**

b. To simplify operations, all residents from the same area need to be assigned to the same shelter. Modify your formulation accordingly. **[10 pts]**

c. For equity reasons, you want to avoid negatively impacting any area. Modify your formulation (from Question b.) to minimize the largest distance traveled by any resident. **[10 pts]**

d. Implement the three models computationally. For each one, report the average distance and the largest distance traveled, across all residents. Provide a plot showing the density of the distance traveled by the residents, for each solution. Comment on your results. **[20 pts]**

Problem 2: Course scheduling [50 pts]

You are planning the course schedule at the Sloan School for next Semester. To simplify the problem, you make the following assumptions.

On the operations side, Sloan offers 30 courses. Most are offered only once, but some are repeated in several sessions (for a total of 55 sessions). Each session has the same schedule on Mondays/Tuesdays and Wednesdays/Thursdays; hence, we consider the problem over two days (e.g., Mondays and Tuesdays). Each day has six 90-minute time slots (two morning slots, three afternoon slots and one evening slot). Sloan has access to 30 classrooms, each with limited seating capacity.

On the demand side, we consider five groups of students: (i) first-year MBA students, (ii) first-year MBAN students; (iii) first-year ORC SM students; (iv) first-year ORC PhD students; and (v) other students. We divide students into smaller cohorts, for two reasons. First, due to the large size of the program, first-year MBA students are divided into "oceans" of 60 students each, who take core courses together. Second, other students have different interests in electives, and cohorts designate smaller groups with similar interests (these can be calibrated from historical course selections, for instance). Ultimately, we consider 57 cohorts: 6 MBA cohorts of 60 students each, 7 MBAN cohorts of 10 students each, 2 ORC SM cohorts of 5 students each, 2 ORC PhD cohorts of 10 students each, and 40 cohorts of 12 students each, capturing students in other programs.

You define the following inputs, decision variables and constraints:

Sets

$$\begin{aligned}
\mathcal{S}&: \text{ set of student cohorts}\\
\mathcal{C}&: \text{ set of courses}\\
\mathcal{M}_s&: \text{ subset of mandatory courses for cohort } s \in \mathcal{S}\\
\mathcal{I}&: \text{ set of course sessions}\\
\mathcal{J}_c&: \text{ subset of course sessions belonging to course } c \in \mathcal{C}\\
\mathcal{D}&: \text{ set of days}\\
\mathcal{T}&: \text{ set of time blocks within a day}\\
\mathcal{R}&: \text{ set of classrooms}
\end{aligned}$$

Parameters

$$\begin{aligned}
N_s&: \text{ number of students in cohort } s \in \mathcal{S}\\
Q_r&: \text{ capacity of room } r \in \mathcal{R}\\
U_c&: \text{ number of units of course } c \in \mathcal{C}\\
m_s&: \text{ minimum number of units of cohort } s \in \mathcal{S}\\
M_s&: \text{ maximum number of units of cohort } s \in \mathcal{S}
\end{aligned}$$

Decision variables

$$x_{jdtr} = \begin{cases} 1 & \text{if session } j \in \mathcal{J} \text{ is offered on day } d \in \mathcal{D} \text{ at time } t \in \mathcal{T} \text{ in room } r \in \mathcal{R} \\ 0 & \text{otherwise} \end{cases}$$

$$z_{sjdt} = \begin{cases} 1 & \text{if student group } s \in \mathcal{S} \text{ takes session } j \in \mathcal{J} \text{ on day } d \in \mathcal{D} \text{ at time } t \in \mathcal{T} \\ 0 & \text{otherwise} \end{cases}$$

Constraints

$$z_{sjdt} \leq \sum_{r \in \mathcal{R}} x_{jdtr}, \qquad \forall s \in \mathcal{S},\ \forall j \in \mathcal{I},\ \forall d \in \mathcal{D},\ \forall t \in \mathcal{T} \tag{1}$$

$$\sum_{j \in \mathcal{I}} z_{sjdt} \leq 1, \qquad \forall s \in \mathcal{S},\ \forall d \in \mathcal{D},\ \forall t \in \mathcal{T} \tag{2}$$

$$\sum_{d\in\mathcal{D}}\sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{J}_c} z_{sjdt} \leq 1, \qquad \forall s\in\mathcal{S},\ \forall c\in\mathcal{C} \tag{3}$$

$$\sum_{d\in\mathcal{D}}\sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{J}_c} z_{sjdt} \geq 1, \qquad \forall s\in\mathcal{S},\ \forall c\in\mathcal{M}_s \tag{4}$$

$$\sum_{j\in\mathcal{I}} x_{jdtr} \leq 1, \qquad \forall r\in\mathcal{R},\ \forall d\in\mathcal{D},\ \forall t\in\mathcal{T} \tag{5}$$

$$\sum_{r\in\mathcal{R}}\sum_{d\in\mathcal{D}}\sum_{t\in\mathcal{T}} x_{jdtr} = 1, \qquad \forall j\in\mathcal{I} \tag{6}$$

$$\sum_{s\in\mathcal{S}} N_s z_{sjdt} \leq \sum_{r\in\mathcal{R}} Q_r x_{jdtr}, \qquad \forall j\in\mathcal{I},\ \forall d\in\mathcal{D},\ \forall t\in\mathcal{T} \tag{7}$$

You have access to the following data files:

- Pb2_cohorts.csv: A matrix of size 57×4 reporting, for each cohort (column 1), the number of students (column 2), and the minimum and maximum number of units (columns 3 and 4).
- Pb2_courses.csv: A matrix of size 30×3 reporting, for each course (column 1), the number of units (column 2), and for which cohort it is mandatory (column 3).
- Pb2_rooms.csv: A matrix of size 20×2 reporting, for each room (column 1), the maximum number of students it can host (column 2).
- Pb2_sessions.csv: A matrix of size 55×2 reporting, for each session (column 1), the course it corresponds to (column 2).

You also have access to an initial implementation of the problem, in the HW1-Pb2-skeleton.jl file. Feel free to use and modify the code as you see fit through this problem.

a. Interpret briefly each constraint from a practical standpoint. **[10 pts]**

b. Add constraints enforcing the minimum and maximum numbers of units for each student cohort $s \in \mathcal{S}$. **[10 pts]**

c. The objective of the problem is to create many course options but also to prioritize attractive ones. Denote by $\mathcal{P}_s$ the subset of courses that are "preferred" by cohort $s \in \mathcal{S}$. Formulate the objective of maximizing, first, the total number of "preferred" courses taken across the university, and, second, the number of "non-preferred" courses. You can weight the number of "non-preferred" courses by $\lambda = 0.1$. For instance, if cohort $s \in \mathcal{S}$ takes 3 "preferred" courses and 2 "non-preferred" ones, it contributes $3.2N_s$ to the objective. You can find the relevant data in Pb2_preferences.csv, where each row indicates a cohort, each column indicates a course, and a "1" indicates a "preferred" course.**[10 pts]**

d. Implement the full model. Report your computational experience in detail, describing the various steps of the solution algorithm, their computational times, and their impact. You can include screenshots of the solver's log to support the discussion. **[15 pts]**

e. In order to increase the model's scalability and reduce computational times, propose a decomposition heuristic that breaks down the problem into two (or more) smaller sub-problems. Discuss your approach qualitatively (do not formulate or implement new models). What is attractive in your approach? Are you guaranteed to find an optimal solution of the overall problem? Are you guaranteed to find a feasible solution? Why or why not? **[5 pts]**