CO 370 Fall 2019: Homework 1

Due: October 9th by 2:00pm

Instructions You will be graded not only on correctness, but also on clarity of exposition. You are allowed to talk with classmates about the assignment as long as (1) you acknowledge the people you collaborate with, (2) you write your solutions on your own, and (3) you are able to fully explain your solutions. In the models, always give a clear definition to your decision variables (in most cases, this means that you must explain what they represent in plain words). In case you run into trouble (a question is ambiguous, data provided have an issue, problem with the implementation, etc.), it is your responsibility to ask me or your TAs for clarifications in a timely manner.

Homework submission Questions 1, 2 and 3 are to be submitted on Crowdmark. The model file for Question 4 is to be submitted on Learn.

Question 1 [2 marks] Model the following optimization problems as linear programming (LP) problems.

(a) For a vector $x \in \mathbb{R}^n$, the 1-norm of x (denoted as $||x||_1$) is defined as $||x||_1 := \sum_{i=1}^n |x_i|$, i.e. it is equal to the sum of the absolute values of its coordinates.

Suppose you want to solve the following optimization problem:

$$\begin{aligned} \min \quad & ||x||_1 \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

for some given matrix A with m rows and n columns, and some given $b \in \mathbb{R}^m$.

Find a reformulation of the above optimization problem as an LP.

Solution: Since $x \geq 0$, we have $x_i = |x_i|$ for all i. The problem can thus be modeled as

$$\begin{aligned} \min \quad & \textstyle\sum_{i=1}^n x_i \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0. \end{aligned}$$

Solution if we did not have $x \geq 0$: (this solution is acceptable as well) Note that $|x_i| = \max\{x_i, -x_i\}$. In particular, this implies that $|x_i| \geq x_i$ and $|x_i| \geq -x_i$. We introduce a variable z_i to represent $|x_i|$, for $i = 1, \ldots, n$, and have the following model

$$\begin{aligned} \min \quad & \textstyle\sum_{i=1}^n z_i \\ \text{s.t.} \quad & z_i \geq x_i \quad \forall i = 1, \ldots, n \\ & z_i \geq -x_i \quad \forall i = 1, \ldots, n \\ & Ax = b \\ & x \geq 0 \end{aligned} \tag{1}$$

Since the objective function tries to *minimize* the sum of the z_i's, in an optimal solution we will have that, for each i, (at least) one of the inequalities $z_i \geq x_i$ and $z_i \geq -x_i$ holds tight (otherwise, we could strictly decrease the value of z_i, improving the objective function value). It follows that in an optimal solution $z_i = \max\{x_i, -x_i\}$, i.e., $z_i = |x_i|$.

(b) Let $f(x)$ be a function defined as:

$$f(x) := \min_{i=1,2,3} \{a_i x + d_i\}$$

for some fixed given $a_1, a_2, a_3, d_1, d_2, d_3 \in \mathbb{R}$. Formulate the problem of maximizing $f(x)$ as an LP.

Solution: We introduce a variable z, and write the following model

$$\begin{aligned} \max \quad & z \\ \text{s.t.} \quad & z \leq a_i x + d_i \quad \forall i = 1, 2, 3 \end{aligned} \tag{2}$$

Since the objective function tries to *maximize* z, in an optimal solution we will have (at least) one of the inequalities $z \leq a_i x + d_i$ holding tight (otherwise, we could strictly increase the value of z, improving the objective function value). It follows that in an optimal solution $z = \min_{i=1,2,3}\{a_i x + d_i\}$, i.e., $z = f(x)$.

Question 2 [4 marks] As a part of a ride-sharing app you are designing, you need to solve the following problem. You are given a set of drivers D, and a set of clients C. Each client $c \in C$ is requesting one trip, and each driver $d \in D$ can fulfill the trip request of (at most) one client $c \in C$. Because of their respective locations, you cannot just assign any driver to any client. Instead, a given driver $d \in D$ can only be assigned to a client in the subset $C_d \subseteq C$. For simplicity, you can assume that you also know, for a given client $c \in C$, the set $D_c \subseteq D$ of drivers that can drive c. You want to maximize the number of trips requests that are fulfilled. Formulate the problem as a linear programming problem. Justify why you can model it as an LP (as opposed to an IP).

Solution: The problem can be formulated as the following maximum matching problem: Let G be a bipartite graph with vertex sets (C, D) such that each client $c \in C$ is connected to the drivers $d \in D_c$. (This automatically implies that each driver d is connected to the clients $c \in C_d$.) The problem of maximizing the number of fullfilled trip-requests is equivalent to finding a maximum matching for the bipartite graph G. As we saw in class, a bipartite matching problem can be formulated as a max-flow model. Letting $V' := C \cup D \cup \{s, t\}$ and $E' := E \cup \{sq : q \in C\} \cup \{qt : q \in D\}$ where E is the set of edges in G, we have

$$\begin{aligned} \max \quad & f_x(s) \\ \text{s.t.} \quad & f_x(q) = 0, \quad \forall q \in C \cup D \\ & 0 \leq x_e \leq 1, \quad \forall e \in E' \end{aligned} \tag{P}$$

i.e.

$$\begin{aligned} \max \quad & \textstyle\sum_{c \in C} x_{sc} \\ \text{s.t.} \quad & x_{sc} - \textstyle\sum_{d \in D_c} x_{cd} = 0, \quad \forall c \in C \\ & \textstyle\sum_{c \in C_d} x_{cd} - x_{dt} = 0, \quad \forall d \in D \\ & 0 \leq x_e \leq 1, \quad \forall e \in E' \end{aligned} \tag{P}$$

Now since all capacities are integer, there exists x^* optimal for (P) such that x^* contains only integer values. Moreover, since $0 \leq x_e \leq 1$, it implies that

$$\forall e \in E', \ x_e^* \in \{0, 1\}$$

and so x^* induces a matching on G with maximum size.

Question 3 [4 marks] We are scheduling the duties for a set of doctors $\{0, \ldots, D-1\}$ in an emergency room. We are concerned with a period of T days numbered 0 to $(T-1)$ (e.g. $T = 31$ for a month, or $T = 365$ for a year). For each of these days, we will need to know the corresponding day-of-the-week, represented by a number between 0 (Monday) and 6 (Sunday). The day-of-the-week corresponding to the first day is given by F. For example, if $F = 2$ then the period starts on a Wednesday. There are two shifts per day: the daytime shift, and the following nighttime shift. Our period of concern starts with the daytime shift for day 0, and ends with the nighttime shift for day $(T-1)$. We are given the following data and rules:

- For each day-of-the-week $w \in \{0, \ldots, 6\}$, we know the minimum necessary staffing level of the emergency room: $L_{\texttt{day},w}$ for the daytime, and $L_{\texttt{night},w}$ for the following nighttime.
- For each day $t \in \{0, \ldots, T-1\}$, $A_{dt} = 1$ if the doctor $d \in \{0, \ldots, D-1\}$ is available that day (both daytime and subsequent nighttime), and $A_{dt} = 0$ otherwise (neither daytime nor nighttime).
- If a doctor is on duty during a nighttime, that doctor cannot be on duty the following daytime.
- If a doctor is on duty during a nighttime, that doctor cannot be on duty for the two following nightimes.
- There is a system of points to measure how taxing the schedule is on each doctor. For each day-of-the-week $w \in \{0, \ldots, 6\}$, a doctor gets $P_{\texttt{day},w}$ points for being on duty during the daytime, and $P_{\texttt{night},w}$ for being on duty during the nighttime.

Given a schedule, let $P_{\max}$ be the largest number of points any of the doctors gets. Design a schedule that minimizes $P_{\max}$.

Solution:

Variables:

$$x_{dt} = \begin{cases} 1 \text{ if doctor } d \text{ is on duty for the daytime of day } t, \\ 0 \text{ otherwise.} \end{cases}$$

$$y_{dt} = \begin{cases} 1 \text{ if doctor } d \text{ is on duty for the nighttime of day } t, \\ 0 \text{ otherwise.} \end{cases}$$

$$z = P_{\max} \text{ for the current schedule.}$$

Linear constraints:

(1). Given a day t, the corresponding day-of-the-week $w \in \{0, \ldots, 6\}$ is $(t + F) \bmod 7$. We thus need at least $L_{\texttt{day},(t+F) \bmod 7}$ daytime doctors and $L_{\texttt{night},(t+F) \bmod 7}$ nighttime doctors:

$$\sum_{d=0}^{D-1} x_{dt} \geq L_{\texttt{day},(t+F) \bmod 7}, \quad \forall t = 0, \ldots, T-1$$

$$\sum_{d=0}^{D-1} y_{dt} \geq L_{\texttt{night},(t+F) \bmod 7}, \quad \forall t = 0, \ldots, T-1$$

(2). If $A_{dt} = 0$, doctor d is unavailable on day t, both daytime and nighttime:

$$x_{dt} \leq A_{dt}, \quad \forall d = 0, \ldots, D-1, \ \forall t = 0, \ldots, T-1$$

$$y_{dt} \leq A_{dt}, \quad \forall d = 0, \ldots, D-1, \ \forall t = 0, \ldots, T-1$$

(3). If a doctor d is on duty during a nighttime t, that doctor cannot be on duty on daytime $t+1$:

$$y_{d,t} + x_{d,t+1} \leq 1, \quad \forall d = 0, \ldots, D-1, \ \forall t = 0, \ldots T-2$$

Notice that t only goes to $T-2$ because we need $x_{d,t+1}$ to be defined.

(4). If a doctor d is on duty during a nighttime t, that doctor cannot be on duty for the nightimes $t+1$ and $t+2$.

$$y_{d,t} + y_{d,t+1} + y_{d,t+2} \leq 1, \quad \forall d = 0, \ldots, D-1, \ \forall t = 0, \ldots T-3$$

Notice that t only goes to $T-3$ because we need $y_{d,t+2}$ to be defined. Remark that the above constraint already implies that if $y_{d,T-2} = 1$, then $y_{d,T-1} = 0$, so we need no special treatment for the implications of nighttime $T-2$ on nighttime $T-1$.

(5). We will be minimizing a variable z that represents P_max. Therefore, z must be set to be larger-than-or-equal-to the number of points of each doctor. Those are given by a sum over all days. The points associated to each daytime/nighttime depend on the day-of-the-week, so again we use $w = (t+F) \bmod 7$.

$$z \geq \sum_{t=0}^{T-1} P_{\texttt{day},(t+F) \bmod 7} \, x_{d,t} + \sum_{t=0}^{T-1} P_{\texttt{night},(t+F) \bmod 7} \, y_{d,t}, \quad \forall d = 0, \ldots, D-1$$

Model:

$$
\begin{aligned}
\min \quad & z \\
& \sum_{d=0}^{D-1} x_{dt} \geq L_{\texttt{day},(t+F) \bmod 7}, && \forall t = 0, \ldots, T-1 \\
& \sum_{d=0}^{D-1} y_{dt} \geq L_{\texttt{night},(t+F) \bmod 7}, && \forall t = 0, \ldots, T-1 \\
& x_{dt} \leq A_{dt}, && \forall d = 0, \ldots, D-1, \ \forall t = 0, \ldots, T-1 \\
& y_{dt} \leq A_{dt}, && \forall d = 0, \ldots, D-1, \ \forall t = 0, \ldots, T-1 \\
& y_{d,t} + x_{d,t+1} \leq 1, && \forall d = 0, \ldots, D-1, \ \forall t = 0, \ldots T-2 \\
& y_{d,t} + y_{d,t+1} + y_{d,t+2} \leq 1, && \forall d = 0, \ldots, D-1, \ \forall t = 0, \ldots T-3 \\
& z \geq \sum_{t=0}^{T-1} P_{\texttt{day},(t+F) \bmod 7} \, x_{d,t} + \sum_{t=0}^{T-1} P_{\texttt{night},(t+F) \bmod 7} \, y_{d,t}, && \forall d = 0, \ldots, D-1 \\
& x_{dt} \in \{0,1\}, && \forall d = 0, \ldots, D-1, \ \forall t = 0, \ldots, T-1 \\
& y_{dt} \in \{0,1\}, && \forall d = 0, \ldots, D-1, \ \forall t = 0, \ldots, T-1 \\
& z \in \mathbb{R}
\end{aligned}
$$

Question 4 [5 marks] Implement the model of Question 3 in Julia.

The grading of this question will be partially automated, so <u>it is very important that you follow exactly the</u>

specifications given here. At least one mark will be automatically subtracted if your code does not fully follow the specifications. If your code cannot be run after minor modifications, you will get 0 marks for the question.

The data will be given to your model in the form of a data file to be included. Example files can be downloaded here: `https://www.math.uwaterloo.ca/~lpoirrie/co370/d/hw1-data-00.jl`
`https://www.math.uwaterloo.ca/~lpoirrie/co370/d/hw1-data-01.jl`

The following table gives the correspondence between mathematical notation and Julia names in the data file:

Mathematical notation	Julia notation
D	`D`
T	`T`
F	`F`
$L_{\mathtt{day},w}$	`Lday[w]`
$L_{\mathtt{night},w}$	`Lnight[w]`
A_{dt}	`A[d, t]`
$P_{\mathtt{day},w}$	`Pday[w]`
$P_{\mathtt{night},w}$	`Pnight[w]`

Carefully check the following:

- Your model must be in a file called `hw1-model.jl`
- Your model file must be uploaded to Learn by the project deadline.
- Your model must still work if the data is changed (it will be tested with multiple variants of the data file).
- You are responsible for solving the model in the file `hw1-model.jl`. In other words, `hw1-model.jl` contains a call to `optimize!()`.
- After the model file is run, the two arrays `daytime` and `nighttime` must take values such that (1) `daytime[d, t] == true` if the doctor `d` is on duty on the daytime of day `t`, and `false` otherwise. (2) `nighttime[d, t] == true` if the doctor `d` is on duty on the nighttime of day `t`, and `false` otherwise. Note that `daytime` and `nighttime` are just arrays of Booleans (false or true), not model variables. You will thus typically have, in the file `hw1-model.jl` after the model is solved, some code similar to:

```
if value(some_model_variable) > 0.5
     daytime[d, t] = true
else
     daytime[d, t] = false
end
```

. Your model will be tested in the following way:

```
include("hw1-data-XX.jl")
include("hw1-model.jl")
include("hw1-check.jl")
```

where `XX` is a set of data. This means that your model file must not itself `include("")` a data file! Some of the basic checks can be downloaded here: `https://www.math.uwaterloo.ca/~lpoirrie/co370/d/hw1-check.jl`

Solution:

```
using JuMP
using Cbc

md = Model(with_optimizer(Cbc.Optimizer))

@variable(md, day[d in 0:(D-1), t in 0:(T-1)] <= A[d, t], Bin)
@variable(md, night[d in 0:(D-1), t in 0:(T-1)] <= A[d, t], Bin)
@variable(md, pmax)

@objective(md, Min, pmax)

@constraint(md, [t in 0:(T-1)],
    sum(day[d, t] for d in 0:(D-1)) >= Lday[mod(t + F, 7)])

@constraint(md, [t in 0:(T-1)],
    sum(night[d, t] for d in 0:(D-1)) >= Lnight[mod(t + F, 7)])

@constraint(md, [d in 0:(D-1), t in 0:(T-2)],
    night[d, t] + day[d, t + 1] <= 1)

@constraint(md, [d in 0:(D-1), t in 0:(T-3)],
    night[d, t] + night[d, t + 1] + night[d, t + 2] <= 1)

@constraint(md, [d in 0:(D-1)],
    pmax >= sum(Pday[mod(t + F, 7)] * day[d, t]
        + Pnight[mod(t + F, 7)] * night[d, t] for t in 0:(T-1)))

optimize!(md)

for d = 0:(D-1), t = 0:(T-1)
    daytime[d, t] = (value(day[d, t]) >= 0.5)
    nighttime[d, t] = (value(night[d, t]) >= 0.5)
end
```