Homework 6 Solutions IEORE4004 (Optimization M&M)
Instructor: Christian Kroer

Problem 1

Part a

There are multiple ways to model this problem. I have listed one way.

Notation

- i denotes arc i (could have used (i, j)).
- Set A is the set of all arcs.
- Set A_1 is the set of arcs from source to blue nodes.
- Set A_2 is the set of arcs from blue nodes to red nodes.
- Set A_3 is the set of arcs from red nodes to sink. ($A = A_1 \cup A_2 \cup A_3$).
- p denotes route (or path) p.
- Set P denotes the set of 8 routes (or paths).

Decision variables

- x_p denotes the number of trucks on route p for all $p \in P$.
- z_i denotes whether we use arc i ($z_i = 1$) or not ($z_i = 0$) for all $i \in A_1$. Note that I only declare z_i for arcs in set A_1 since only arcs in A_1 have fixed costs. Declaring z_i for arcs in sets A_2 and A_3 is not incorrect but will be computationally inefficient since we will be declaring (binary) variables that are not needed.
- w_i denotes the number of trucks on arc i for all $i \in A$.

Parameters

- c_i^f denotes the fixed cost of using arc i for all $i \in A_1$.
- c_i^v denotes the variable cost of using arc i for all $i \in A_3$.
- u denotes the capacity of each arc in A_2. (It is a constant for all $i \in A_2$. Therefore, I do not have u_i. Instead, I have u).
- a_{pi} denotes whether route p uses arc i ($a_{pi} = 1$) or not ($a_{pi} = 0$) for all $p \in P$ and all $i \in A$. As we will see in constraints, this parameter is very useful. Note that a_{pi} is a parameter (and not a variable) since we know this information before hand. Therefore, even if this is binary, it is not an issue computationally.
- R_p denotes the revenue per truck on route p for all $p \in P$.

Objective

- Our objective is to maximize profit, which equals revenue minus cost.
- Revenue equals
$$\sum_{p \in P} R_p x_p.$$
- Cost can be split into fixed cost and variable cost.
  - Fixed cost equals
$$\sum_{i \in A_1} c_i^f z_i.$$
  - Variable cost equals
$$\sum_{i \in A_3} c_i^v w_i.$$
- Hence, the objective is
$$\underset{x,z,w}{\text{maximize}} \left\{ \sum_{p \in P} R_p x_p - \sum_{i \in A_1} c_i^f z_i - \sum_{i \in A_3} c_i^v w_i \right\}. \tag{1}$$
- Note that we optimize over all the decision variables x, z, w.

Constraints

- **Capacity**: Each arc in set A_2 has a capacity of u trucks. Mathematically,
$$w_i \leq u \ \forall i \in A_2. \tag{2}$$
- **Non-negativity**: All variables should be non-negative. Mathematically,
$$x_r \geq 0 \ \forall r \in P \tag{3}$$
$$w_i \geq 0 \ \forall i \in A. \tag{4}$$
I do not impose non-negativity on variables z_i since I will impose binary constraints on them (which will impose non-negativity implicitly). However, imposing non-negativity on z_i is not incorrect.
- **Binary**: By definition, z_i is binary. Mathematically,
$$z_i \in \{0, 1\} \ \forall i \in A_1. \tag{5}$$
- **Integrality**: By definition, both x_r and w_i should be integers. However, since the problem states to only use binary and continuous variables, we do not impose integer constraints on x_r and w_i. This relates to the extra credit question. Since integer constraints are computationally expensive, we choose not to impose them and solve the resulting optimization model (which will be relatively easy to solve since we do not impose integrality). If the resulting solution satisfies the integer constraints, then we have a win-win situation (that is, we found an integer solution without imposing integer constraints). But, in general, it is not always true that we will get an integer solution without imposing integrality. This raises the following question: For the given problem, will we get an integer solution without imposing integrality? We will explore this issue in the extra credit section.

- **Supply**: We can use up to 100 trucks. Mathematically,

$$\sum_{p \in P} x_p \leq 100. \tag{6}$$

Writing $\sum_{i \in A} w_i \leq 100$ is incorrect since then, we will be counting trucks multiple times. Wrting $\sum_{i \in A_1} w_i \leq 100$ is correct. Similarly, writing either $\sum_{i \in A_2} w_i \leq 100$ or $\sum_{i \in A_3} w_i \leq 100$ is correct.

- **Relational constraints**: We need to ensure that our variables are related correctly. We have 3 class of variables (x, z, w). I will relate x to w and w to z. Accordingly, x and z will be related implicitly. Mathematically, x and w are related as follows:

$$w_i = \sum_{p \in P} a_{pi} x_p \ \forall i \in A. \tag{7}$$

Moreover, w and z are related as follows:

$$w_i \leq M z_i \ \forall i \in A_1, \tag{8}$$

where M is a big number ($M = 100$ works since we have 100 trucks in total).

- **Flow balance constraints**: We do not need to state the flow balance constraints explicitly since they will be satisfied (implicitly). Reason for this is that we are using paths to formulate. Stating them is not incorrect but computationally inefficient.

Formulation

Combining Equations (1) to (8), we get the following formulation:

$$\begin{aligned}
\underset{x,z,w}{\text{maximize}} \quad & \left\{ \sum_{p \in P} R_p x_p - \sum_{i \in A_1} c_i^f z_i - \sum_{i \in A_3} c_i^v w_i \right\} \\
\text{subject to} \quad & w_i \leq u \ \forall i \in A_2 \\
& x_r \geq 0 \ \forall r \in P \\
& w_i \geq 0 \ \forall i \in A \\
& z_i \in \{0, 1\} \ \forall i \in A_1 \\
& \sum_{p \in P} x_p \leq 100 \\
& w_i = \sum_{p \in P} a_{pi} x_p \ \forall i \in A \\
& w_i \leq M z_i \ \forall i \in A_1.
\end{aligned}$$

Part b

See solution file *p1.2.lp*. The optimal solution is to send 10 trucks on route 1, 30 trucks on route 2, 30 trucks on route 5, and 30 trucks on route 7.

Problem 2

Part a. Let $x_{ij} = 1$ if frequency j is used in city i, otherwise $x_{ij} = 0$; let $y_j = 1$ if frequency j is bought, otherwise $y_i = 0$; and let $z_i = 1$ if city i is broadcasted, otherwise $z_i = 0$. Let n be the number of cities and let $c = \$500,000$ be the cost of purchasing each frequency. The general problem can be formulated as the following:

$$
\begin{aligned}
\text{maximize} \quad & \textstyle\sum_{i=1}^{n} p_i z_i - \sum_{j=1}^{n} c y_j \\
\text{subject to} \quad & \textstyle\sum_{i=1}^{n} x_{ij} \le n \cdot y_j \quad \text{for all } j \in [n] \\
& \textstyle\sum_{j=1}^{n} x_{ij} \le n \cdot z_i \quad \text{for all } i \in [n] \\
& \textstyle\sum_{j=1}^{n} x_{ij} \le 1 \quad \text{for all } i \in [n] \\
& \textstyle\sum_{j=1}^{n} x_{ij} \ge z_i \quad \text{for all } i \in [n] \\
& x_{ij} + x_{tj} \le 1 \quad \text{for all } j \in [n], (i,t) \in E \\
& y_j \in \{0,1\} \quad \text{for all } j \in [n] \\
& z_i \in \{0,1\} \quad \text{for all } i \in [n] \\
& x_{ij} \in \{0,1\} \quad \text{for all } i,j \in [n]
\end{aligned}
$$

Part b. See script *p2.2.lp* and solution file *p2.2.sol*. The maximum net-profit is \$750,000. In particular, city 1 and 7 are broadcasted with the first type of frequency; city 2,3 and 6 are broadcasted with the second type of frequency.
* If assumed all cities need to boradcasted, the optimal net-profit is \$450,000. In particular, city 1 and 7 are broadcasted with the first type of frequency; city 2,3 and 6 are broadcasted with the second type of frequency; city 4 and 5 are broadcasted with the third type of frequency.

Problem 3

Let $x = 1$ denote the case where a marina will be built and $x = 0$ denote the case where a swimming-tennis complex will be built. Let v be number of apartments and u be number of houses.

In units of \$1,000, the total profit would be

$$48v + 46u - 40(u + v) - 1200x - 2800(1 - x)$$

The constraint must satisfy $u \ge 3v$ if $x = 1$. This "if" constraint can be formulated as

$$u \ge 3v - M(1 - x).$$

Lastly, the total number of apartments and houses must satisfy $u + v \le 10000$

The whole formulation can be written as

$$
\begin{aligned}
&\text{maximize } 48v + 46u - 40(u + v) - 1200x - 2800(1 - x) \\
&\text{subject to } u + v \le 10000 \\
&\qquad\qquad u \ge 3v - M(1 - x) \\
&\qquad\qquad x \in \{0,1\}, u, v \in \mathbb{Z}_+
\end{aligned}
$$

Problem 4

Let $A_{i,j} = 1$ if value j can be created into wallet by using $\le i$ number of coins, and let $A_{i,j} = 0$ otherwise. The recursion can be written as

$$
\begin{aligned}
&A_{i+1,j} = \max\{A_{i,j}, A_{i,j-1}, A_{i,j-2}, A_{i,j-5}\} \\
&A_{i,0} = 1, \forall i \ge 0 \\
&A_{0,j} = 0, \forall j \ge 1
\end{aligned}
$$

Solving above recursion and find the smallest j such that $A_{5j} = 0$, that is 23.

More generally, suppose we have m type of coins and a wallet can fit at most k coins,

$$
\begin{aligned}
&A_{i+1,j} = \max\left\{A_{i,j}, A_{i,j-v(1)}, A_{i,j-v(2)},, ..., A_{i,j-v(m)}\right\} \\
&A_{i,0} = 1, \forall i \geq 0 \\
&A_{\ell,j} = 0, \forall j \geq 1, \ell \leq 0
\end{aligned}
$$

where $v(i)$ denotes the value of the $i-th$ type of coin. After completing computation of recursion, scan the k-th row to find the smallest j such that $A_{k,j} = 0$. For the input file, the solution is 1509.