MATH3290 Mathematical Modeling 2021/22

Assignment #3

Due Date: 5pm, Mar. 27

Note For Problems 1, 2 and 3, give the steps clearly and hand in your assignment to the *assignment box* in LSB; for Problem 4, pack your codes and results in a zip-file **[your name]-[your student ID]-assign3.zip**, then email it to TA (**zqwang@math.cuhk.edu.hk**). "**(Optional)**" means this problem is optional and solving or not depends on your own will.

Problem 1

The advertising alternatives for a company include television, radio, and newspaper advertisements. The costs and estimates for audience coverage are given in table 1.

	Television	Newspaper	Radio
Cost per advertisement	\$2000	\$600	\$300
Audience per advertisement	100,000	40,000	18,000

Table 1: Data set for Problem 1.

The local newspaper limits the number of weekly advertisements from a single company to *ten*. Moreover, in order to balance the advertising among the three types of media, *no more than half* of the total number of advertisements should occur on the radio, and at least 10% should occur on television. The weekly advertising budget is \$18,200. How many advertisements should be run in each of the three types of media to maximize the total audience?

Solve it by the *Simplex method*. In each step, clearly state the entering and leaving variables, independent and dependent variables, and the current value of objective function.

Problem 2

Your company sells a product whose demands over the next *four* months are 100, 140, 210 and 180 units respectively. You can stock just enough supply to meet the demand each month, or you can overstock to meet the demand for two or more consecutive months. In the latter case, a holding cost of \$1.2 is charged per overstocked unit per month. You estimate the unit purchase prices for the next 4 months are \$15, \$12, \$10 and \$14, respectively. A set-up cost of \$200 is incurred each time a purchase order is placed. For example, if your company wants to purchase enough products for next *two* months at the beginning of the first month, it needs to pay

$$\$200 + \$15 * (100 + 140) + \$1.2 * 140 = \$3968;$$

if your company wants to purchase enough products for next *three* months at the beginning of the first month, then it needs to pay

$$\$200 + \$15 * (100 + 140 + 210) + \$1.2 * 140 + \$1.2 * 210 * 2 = \$7622.$$

Your company wants to develop a purchasing plan that will minimize the total costs of ordering, purchasing and holding an item in stock.

(a) Find the optimal solution by *Dijkstra's algorithm*. Write down the steps clearly.

(b) Find the optimal solution by *Dynamic Programming* (with backward recursion). Write down the steps clearly.

Problem 3

Find the maximum flow from s to t in the graph shown in fig. 1 using the Ford and Fulkerson algorithm. Show the steps clearly.

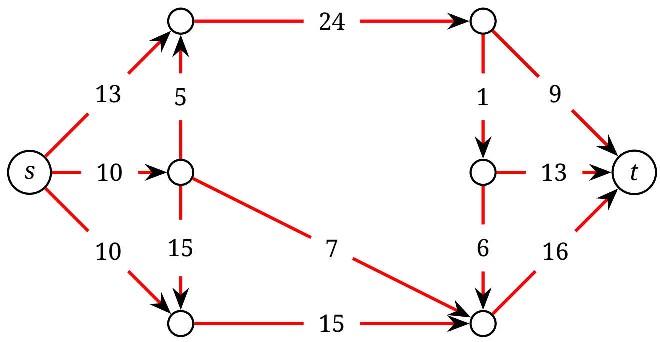

Figure 1: Graph for Problem 3.

Problem 4

(Programming exercise) In this problem, you are required to write a computer program to solve the matrix 0-1 problem. Your task is to submit your **COMPLETE** code of the program with specific format of both input and output. Before you do the exercise, please read the following instructions **CAREFULLY**.

- **Description:** Given two integers m and n and two sequences of numbers $\{r_i\}_{i=1}^m$ and $\{s_j\}_{j=1}^n$, which are the row and column sums of a 0-1 matrix, respectively. Can you find such $m \times n$ 0-1 matrix? If the answer is yes, print out one of this matrix. Otherwise, print out 'No'.

  a. Every entry in the 0-1 matrix should **ONLY** be 0 or 1.

  b. The sequences $\{r_i\}_{i=1}^m$ and $\{s_j\}_{j=1}^n$ should satisfy: $0 \le r_i \le n$ and $0 \le s_j \le m$ for any $i = 1, \cdots, m$ and $j = 1, \cdots, n$.

  c. Additionally, the sum of all r_i should equal to the sum of all s_j.

  You are provided a m.file called **zeroone.m** and you are asked to complete a MATLAB function called zeroone(m,n,r,s).

- **Input:** There are four input-data: m, n, r and s, where m and n are two positive integers. r is an $m \times 1$ vector and s is a $n \times 1$ vector. The vectors r and s represent the row and the column sums of the 0-1 matrix, respectively.

- **Output:** If you can find such a 0-1 matrix, print out 'Yes' and one of the possible matrix. Otherwise, **JUST** print out 'No'.

- **Input Example:** (Type the corresponding commands in the Command Window of MATLAB when you test the code yourself.)
  ```
  » m = 4; n = 6;
  r = [ 3; 2; 3; 4];
  s = [ 3; 2; 2; 3; 1; 1];
  [ flag, matrix ] = zeroone(m,n,r,s)
  ```

- **Output Example:**
  ```
  flag =

  Yes

  matrix =
  ```

1 0 0 1 1 0
1 1 0 0 0 0
1 0 1 1 0 0
0 1 1 1 0 1

- **Remark:** The above **INPUT** and **OUTPUT** example is only for your reference. You should test following examples to verify your program.
- **Your work:** (Test the following dataset and print out your results.)
- **Dataset 1:**
  m = 5; n = 5;
  r = [2; 1; 2; 1; 3];
  s = [2; 1; 3; 1; 2];

- **Dataset 2:**
  m = 6; n = 8;
  r = [4; 3; 2; 1; 3; 1];
  s = [2; 1; 3; 1; 2; 4; 2; 2];

- **Dataset 3:**
  m = 4; n = 4;
  r = [1; 1; 1; 1];
  s = [1; 1; 1; 1];

- **Dataset 4:**
  m = 7; n = 7;
  r = [1; 2; 3; 3; 2; 2; 1];
  s = [2; 5; 1; 1; 1; 1; 1];

Problem 5

(Optional) After solving this problem, you will understand why the simplex method in the lecture note works. The goal of the simplex method is solving the following optimization problem:

$$\begin{aligned} \text{Maximize} \quad & c \cdot x = \sum_{i=1}^{n} c[i]x[i] \\ \text{s.t.} \quad & Ax \le b, \\ & x \ge \mathbf{0}, \end{aligned}$$

where $v[i]$ is the i-th element of the vector v, $a \le b$ for two vectors a and b means $a[i] \le b[i]$, and $\mathbf{0}$ is the zero-vector. Let $b \in \mathbb{R}^m$, we simply *assume* that $x = \mathbf{0}$ is a feasible solution, which implies $b \ge \mathbf{0}$ from the constraint $Ax \le b$. We construct a tableau format of the optimization problem as

$$\left[\begin{array}{ccc|c} A & E_m & \mathbf{0} & b \\ -c^{\top} & \mathbf{0} & 1 & 0 \end{array}\right], \tag{$\star$}$$

where E_m is the identity matrix of size m. This tableau format is equivalent to

$$\begin{bmatrix} A & E_m & \mathbf{0} \\ -c^{\top} & \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix},$$

where x is the vector of decision variables, y is the vector of slack variables, and z is the objective variable. Recall that we need $x \ge \mathbf{0}$, $y \ge \mathbf{0}$ and $z \ge 0$ (this comes from $x = \mathbf{0}$ is a feasible solution, then z the optimal value of $c \cdot x$ must be greater than 0), this leads that any *non-negative* solutions of the *indefinite linear system* eq. ($\star$) are feasible solutions. In the Linear Algebra course, we have learned that *row operations* will not change solutions (exist or not) of the original system. Then, each iteration of the simplex method is essentially making row operations. At the beginning of each iteration, we can find an identity matrix of size $m+1$ as a sub-matrix of the tableau format, and the column indices are exactly corresponding to current *dependent variables*. Try to explain the following statements.

(a) At the beginning of the k-th iteration, after *switching columns*, the tableau format could be written as

$$\left[\begin{array}{ccc|c} A_k & E_m & \mathbf{0} & b_k \\ c_k^\top & \mathbf{0} & 1 & t_k \end{array}\right],$$

where $b_k \geq \mathbf{0}$ and $t_k \geq 0$. If $c_k \geq \mathbf{0}$, then optimal solution x^* of the original optimization problem is b_k by extracting corresponding entries, and the optimal value is t_k.

(b) If $c_k \geq 0$ does not hold, a *most negative* rule is adopted to select the *entering variable*. Suppose that the entering variable is the last entry of c_k (applying column switches can always make it happen), the tableau format is written as

$$\left[\begin{array}{cccc|c} B_k & \alpha_k & E_m & \mathbf{0} & b_k \\ d_k^\top & e_k & \mathbf{0} & 1 & t_k \end{array}\right],$$

where $e_k < 0$. Show that if $\alpha_k \leq \mathbf{0}$, then the optimization problem does *not* have a *bounded* optimal value (i.e., there are feasible solutions $\{x_n\}$ such that $c \cdot x_n \to \infty$).

(c) Suppose the *leaving variable* selected by the *smallest positive ratio* rule is the first entry of α_k, the tableau format is rewritten as

$$\left[\begin{array}{ccccc|c} C_k & \beta_k & 1 & \mathbf{0} & 0 & l_k \\ D_k & \gamma_k & \mathbf{0} & E_{m-1} & \mathbf{0} & b_k' \\ d_k^\top & e_k & 0 & \mathbf{0} & 1 & t_k \end{array}\right],$$

where $\beta_k > 0$. Making row operations, the tableau format is transformed into

$$\left[\begin{array}{ccccc|c} \frac{C_k}{\beta_k} & {\color{red}1} & \frac{1}{\beta_k} & \mathbf{0} & 0 & \frac{l_k}{\beta_k} \\ D_k - \frac{\gamma_k C_k}{\beta_k} & 0 & -\frac{\gamma_k}{\beta_k} & {\color{red}E_{m-1}} & \mathbf{0} & b_k' - \frac{\gamma_k l_k}{\beta_k} \\ d_k^\top - \frac{e_k C_k}{\beta_k} & 0 & -\frac{e_k}{\beta_k} & \mathbf{0} & {\color{red}1} & t_k - \frac{e_k l_k}{\beta_k} \end{array}\right].$$

Show that

$$b_k' - \frac{\gamma_k l_k}{\beta_k} \geq \mathbf{0}.$$

(d) From the expression

$$t_{k+1} = t_k - \frac{e_k l_k}{\beta_k}$$

explain why the *most negative* and *smallest positive ratio* rules are introduced here.