Fall 2020
ECE 236A
Professor: Christina Fragouli
TAs: Mine Dogan and Kaan Ozkara

Linear Programming

Solutions of Homework 1

Problem 1 (`2 points`, Exer. 5 in *Linear Programming Exercises*): Consider the linear program

$$\begin{array}{ll} \text{minimize} & c_1x_1 + c_2x_2 + c_3x_3 \\ \text{subject to} & x_1 + x_2 \geq 1 \\ & x_1 + 2x_2 \leq 3 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0. \end{array}$$

Give the optimal value and the optimal set for the following values of c: $c = (-1, 0, 1)$, $c = (0, 1, 0)$, $c = (0, 0, -1)$.

Solution: (1) $c = (-1, 0, 1)$: optimal solution $x^* = (3, 0, 0)$, optimal value $p^* = -3$.
(2) $c = (0, 1, 0)$: optimal set $X_{opt} = \{x \in \mathbf{R}^3 | 1 \leq x_1 \leq 3, x_2 = 0, x_3 \geq 0\}$, and optimal value $p^* = 0$.
(3) $c = (0, 0, -1)$: optimal set $X_{opt} = \emptyset$, and optimal value $p^* = -\infty$.

Problem 2 (`3 points`): A young investor is planning to invest in the stocks of the Metalco company during the week. He has some predicted values for the buying and selling prices of the shares during the week. The prices are shown in the following table.

Operation	Monday	Tuesday	Wednesday	Thursday	Friday
Buy ($/share)	3	1.7	2.4	2.5	1.8
Sell ($/share)	2.1	2.4	3	2	3.1

The investor starts the week with $120 and he wants to formulate a buying-selling strategy through a linear program to maximize the total amount he would have at the end of the week. The following trading opportunities and restriction apply:

1. He starts the week with no shares owned in Metalco.
2. On any given day, he can only sell shares that he owns from previous trading days. For example, on Wednesday, he can only sell shares that he still owns after trading on Monday and Tuesday.
3. No borrowing is allowed. This means that on any given day, the trader can only buy shares using money he had at the beginning of the week $\pm$ any profit/loss he incurred from the previous days of trading.

For simplicity, the amount of shares that can be bought can take non-integer values. Formulate a linear program that maximizes the amount of money the trader has at the end of the week.

Solution: Let s_i $(i = 1, 2, 3, 4, 5)$ be the shares sold on the i-th day of the week. Similarly let b_i $(i = 1, 2, 3, 4, 5)$ be the shares bought by the investor on the i-th day of the week. Then the LP can be formulated as:

$$\begin{aligned}
\text{maximize} \quad & 2.1s_1 + 2.4s_2 + 3s_3 + 2s_4 + 3.1s_5 - 3b_1 - 1.7b_2 - 2.4b_3 - 2.5b_4 - 1.8b_5 \\
\text{subject to} \quad & 3b_1 \leq 120 \\
& 1.7b_2 \leq 120 + 2.1s_1 - 3b_1 \\
& 2.4b_3 \leq 120 + 2.4s_2 + 2.1s_1 - 1.7b_2 - 3b_1 \\
& 2.5b_4 \leq 120 + 3s_3 + 2.4s_2 + 2.1s_1 - 2.4b_3 - 1.7b_2 - 3b_1 \\
& b_5 \leq 0 \\
& s_1 \leq 0 \\
& \sum_{j=1}^{i} s_j \leq \sum_{j=1}^{i-1} b_j, \;\; i = 2, 3, 4, 5 \\
& b_i \geq 0, \;\; i = 1, 2, 3, 4, 5. \\
& s_i \geq 0, \;\; i = 1, 2, 3, 4, 5.
\end{aligned}$$

Problem 3 (`3 points`): An advertising agency has M products to advertise at N locations (e.g., it can choose to advertise a product on Facebook, Google, Twitter, or on TV.). The advertising agency can choose the advertising time $x_{i,j}$ for product i being advertised at location j (i.e., how long each product is advertised at each location.). The cost of advertising is c_j per unit time when the advertisement is placed at location j. For product i being advertised at location j, the multiplier effect is $u_{i,j}$. The multiplier effect represents that a unit time long advertising can generate an extra $u_{i,j}$ profit because of sales increase. The advertising agency has an advertising budget b_i for product i, indicating that the cost for advertising product i can not exceed this level. Each location j has a total available advertising time t_j, indicating that the total advertising time for all the products at location j cannot exceed this time level. The goal of the advertising agency is to maximize its total net profit (profit of sales increase minus the advertising cost). We assume that $u_{i,j}$, c_j, b_i, t_j are fixed constants. Please formulate an LP problem based on the provided information.

Solution: We can formulate the linear programming problem as follows:

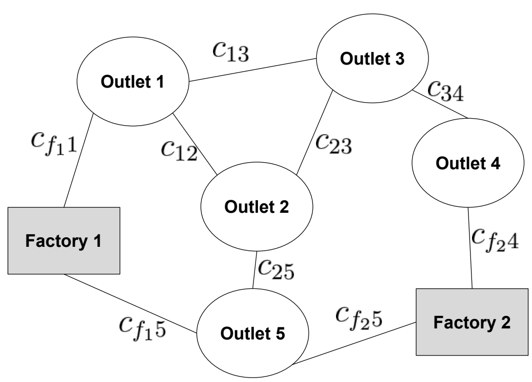
$$\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{M} \sum_{j=1}^{N} (u_{i,j} - c_j) x_{i,j} \\
\text{subject to} \quad & \sum_{j=1}^{N} c_j x_{i,j} \leq b_i, \quad i = 1, 2, \cdots, M \\
& \sum_{i=1}^{M} x_{i,j} \leq t_j, \quad j = 1, 2, \cdots, N \\
& x_{i,j} \geq 0, \quad i = 1, 2, \cdots, M, \; j = 1, 2, \cdots, N,
\end{aligned}$$

where the objective function describes to maximize the total net profit; the first set of constraints describe the advertising budget constraints for each product; the second set of constraints describe the advertising time constraints for each advertising location; and the third constraints describe that the advertising time cannot be negative. The variables are $x_{i,j}$, $i = 1, 2, \cdots, M$, $j = 1, 2, \cdots, N$.

<u>Problem 4</u> (`2 points`): Linear programming can be used to optimize the cost of goods transportation between different selling points. The following is a simplified version of such an approach. Solodrex manufactures a brand of cheese in 2 factories and sells its production through 5 sales outlets in California. The demands of the market have changed in different areas this month and therefore, this weekend Solodrex intends to produce and redistribute cheese stocks to its 5 sales outlets. Current stocks and the needed stocks at each outlet are given in the table below.

	Current Stock (lb)	Needed Stock (lb)
Outlet 1	1,200	2,450
Outlet 2	1,800	1,100
Outlet 3	1,200	1,600
Outlet 4	1,500	3,300
Outlet 5	1,100	2,100

The two factories of Solodrex (Factory 1 and Factory 2) can manufacture cheese at a cost of p_1 and p_2 \$ per lb, respectively. The manufactured stock as well as the stock available at each outlet can be moved through the roads connecting them which are shown in the figure. The cost per lb of transportation through these roads (in either direction) is also shown.



Write an LP that will enable Solodrex to minimize the cost needed to meet the new market requirement.

<u>Solution</u>: Let x_1, x_2 be the amount of cheese produced from Factory 1 and Factory 2, respectively. Let t_{ij} be the amount of cheese moved from node i to node j. We denote by $N(i)$ the set of neighboring nodes (outlets and/or factories) to the i-th node. For example $N(1) = \{f_1, 2, 3\}$. For Factory 1, we have $N(f_1) = \{1, 5\}$. We denote the current avaialable stock vector by s and the demanded stocks vector by n. These vectors are given in Table 1.

1. we have the following simple constraints on the variables:

$$
\begin{aligned}
&t_{ij} \geq 0 \quad \forall i, j \in \{1,2,3,4,5,f_1,f_2\},\ i \neq j \\
&x_i \geq 0, \quad i = 1,2
\end{aligned}
$$

2. The factories have to at least produce the amount of cheese that is not available in the market, therefore we have:

$$x_1 + x_2 \geq \sum_{i=1}^{5} n_i - \sum_{i=1}^{5} s_i$$

3. The goods transported from the factory are at most equal to the amount produced in that factory plus any good transported into the factory:

$$\sum_{j \in N(f_i)} t_{f_i j} \leq x_i + \sum_{j \in N(f_i)} t_{j f_i}, \quad i = 1,2$$

4. For each Outlet, the difference between the incoming and outgoing amount of cheese should be greater than or equal to the difference between its needed and current stocks.

$$\sum_{j \in N(i)} t_{ji} - \sum_{j \in N(i)} t_{ij} \geq n_i - s_i, \quad i = 1,2,3,4,5$$

Finally, the objective function should minimize the sum of transportation costs and production costs:

$$\text{minimize} \ \ x_1 p_1 + x_2 p_2 \ + \sum_{\substack{i,j \in \{f_1,f_2,1,2,\cdots,5\}, \\ i \neq j}} c_{ij} t_{ij}$$

where $c_{ij} = c_{ji} \quad \forall i, j \in \{1,2,3,4,5,f_1,f_2\}$.

<u>Problem 5</u> (`2 points`, Exer. 12 in *Linear Programming Exercises*):

We are given p matrices $A_i \in \mathbf{R}^{n \times n}$, and we would like to find a single matrix $X \in \mathbf{R}^{n \times n}$ that we can use as an approximate right-inverse for each matrix A_i, *i.e.*, we would like to have

$$A_i X \approx I,\ i = 1, \cdots, p$$

We can do this by solving the following optimization problem with X as variable:

$$\text{minimize} \ \max_{i=1,\cdots,p} \|I - A_i X\|_\infty. \tag{1}$$

Here $\|H\|_\infty$ is the 'infinity-norm' or 'max-row-sum norm' of a matrix H, defined as

$$\|H\|_\infty = \max_{i=1,\cdots,m} \sum_{j=1}^{n} |H_{ij}|$$

if $H \in \mathbf{R}^{m \times n}$. Express problem (1) as an LP. You don't have to reduce the LP to a canonical form, as long as you are clear about what the variables are, what the meaning is of any auxiliary variables that you introduce, and why the LP is equivalent to the problem (1).

Solution:

We can express the problem as the LP

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & -S_k \leq I - A_k X \leq S_k, \quad k = 1, \cdots, p \\ & S_k \mathbf{1} \leq t\mathbf{1}, \quad k = 1, \cdots, p. \end{array}$$

The variables are $X \in \mathbf{R}^{n \times n}$, p matrices $S_k \in \mathbf{R}^{n \times n}$, and the scalar t. The inequalities in the first constraints are componentwise inequalities between *matrices*, *i.e.*, they mean

$$-(S_k)_{ij} \leq (I - A_k X)_{ij} \leq (S_k)_{ij}.$$

for $i = 1, \cdots, n$ and for $j = 1, \cdots, n$. These pairs of inequalities are equivalent to

$$|(I - A_k X)_{ij}| \leq (S_k)_{ij}.$$

The second inequality is a componentwise inequality between *vectors*, *i.e.*, it means

$$(S_k \mathbf{1})_i = \sum_{j=1}^{n} (S_k)_{ij} \leq t$$

for $i = 1, \cdots, n$.

Problem 6 (`5 points`, Exer. 6 [b, e, f, g] in *Linear Programming Exercises*): For each of the following LPs, express the optimal value and the optimal solution in terms of the problem parameters (c, k). If the optimal solution is not unique, it is sufficient to give one optimal solution.

(i)

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & -\mathbf{1} \leq x \leq \mathbf{1}. \end{array}$$

The variable is $x \in \mathbf{R}^n$.

(ii)

$$\begin{array}{ll} \text{maximize} & c^T x \\ \text{subject to} & \mathbf{1}^T x = k \\ & 0 \leq x \leq \mathbf{1}. \end{array}$$

The variable is $x \in \mathbf{R}^n$. k is an integer with $1 \leq k \leq n$.

(iii)

$$\begin{array}{ll} \text{maximize} & c^T x \\ \text{subject to} & \mathbf{1}^T x \leq k \\ & 0 \leq x \leq \mathbf{1}. \end{array}$$

The variable is $x \in \mathbf{R}^n$. k is an integer with $1 \leq k \leq n$.

(iv)

$$\begin{array}{ll} \text{maximize} & c^T x \\ \text{subject to} & d^T x = \alpha \\ & 0 \leq x \leq \mathbf{1}. \end{array}$$

The variable is $x \in \mathbf{R}^n$. α and the components of d are positive.

Solution: (i) We can optimize over each x_i independently of the other variables. The optimal value of x_i is

$$x_i^* = \begin{cases} 1, c_i \leq 0 \\ -1, c_i > 0, \end{cases}$$

and the optimal value is

$$p^* = -\sum_{i=1}^{n} |c_i| = -||c||_1.$$

(ii) The optimal value is the sum of the largest k components of c. Assume the coefficients of c are sorted in decreasing order:

$$c_1 \geq c_2 \geq \cdots \geq c_n.$$

The optimal value of the LP is

$$p^* = c_1 + c_2 + \cdots + c_k.$$

The optimal solution x^* has all components zero, except

$$x_1^* = \cdots = x_k^* = 1.$$

(iii) The optimal value is the sum of the largest k positive components of c. As in (c), assume the coefficients of c are sorted in decreasing order. If $c_1 < 0$, the optimal $x^* = 0$. Otherwise, define

$$j = min\{max\{i = 1, \cdots, n | c_i \geq 0\}, k\}.$$

Then the optimal value is

$$p^* = c_1 + c_2 + \cdots + c_j.$$

The optimal x^* has all components zero except

$$x_1^* = \cdots = x_j^* = 1.$$

(iv) Using a change of variables $y_i = d_i x_i$ we can reduce the problem to

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{n} (c_i/d_i) y_i \\ \text{subject to} & \mathbf{1}^T y = \alpha \\ & 0 \leq y_i \leq d_i, \quad \forall i \in \{1, \cdots, n\} \end{array} \qquad (2)$$

The optimal solution is obtained by first sorting the ratios c_i/d_i so that

$$\frac{c_1}{d_1} \geq \frac{c_2}{d_2} \geq \cdots \geq \frac{c_n}{d_n}$$

Let

$$\bar{k} = \max\left\{ k \in \{0, 1, \cdots, n-1\} \middle| \sum_{i=1}^{k} d_i \leq \alpha \right\}.$$

Then the optimal solution $y^\star$ is

$$y_i^\star = \begin{cases} d_i & i \leq \bar{k} \\ \alpha - \sum_{j=1}^{\bar{k}} d_j & i = \bar{k}+1 \\ 0 & i > \bar{k}+1 \end{cases}.$$

Going back to the original variables we find that the optimal solution is

$$x_i^\star = \begin{cases} 1 & i \leq \bar{k} \\ (\alpha - \sum_{j=1}^{\bar{k}} d_j)/d_i & i = \bar{k}+1 \\ 0 & i > \bar{k}+1 \end{cases}.$$

<u>Problem 7</u> (`2 points`, Exer. 9 in *Linear Programming Exercises*): Formulate the following problems as LPs:

(a) minimize $\|Ax - b\|_1$ subject to $\|x\|_\infty \leq 1$.

(b) minimize $\|x\|_1$ subject to $\|Ax - b\|_\infty \leq 1$.

(c) minimize $\|Ax - b\|_1 + \|x\|_\infty$.

In each problem, $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbf{R}^m$ are given, and $x \in \mathbf{R}^n$ is the optimization variable.

<u>Solution</u>:

(a) The problem is equivalent to the LP

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m y_i \\ \text{subject to} & -y \leq Ax - b \leq y \\ & -\mathbf{1} \leq x \leq \mathbf{1}. \end{array}$$

The variables are $x \in \mathbf{R}^n$ and $y \in \mathbf{R}^m$. In matrix notation:

$$\begin{array}{ll} \text{minimize} & \bar{c}^T \bar{x} \\ \text{subject to} & \bar{A}\bar{x} \leq \bar{b}, \end{array}$$

where

$$\bar{x} = \begin{bmatrix} x \\ y \end{bmatrix}, \quad \bar{c} = \begin{bmatrix} 0 \\ \mathbf{1} \end{bmatrix}, \quad \bar{A} = \begin{bmatrix} A & -I \\ -A & -I \\ I & 0 \\ -I & 0 \end{bmatrix}, \quad \bar{b} = \begin{bmatrix} b \\ -b \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix}.$$

(b) If we introduce a new variable $y \in \mathbf{R}^n$, we can express the problem as

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T y \\ \text{subject to} & -y \leq x \leq y \\ & -\mathbf{1} \leq Ax - b \leq \mathbf{1}, \end{array}$$

which is an LP in x and y. In matrix notation, the problem is

$$\begin{array}{ll} \text{minimize} & \bar{c}^T \bar{x} \\ \text{subject to} & \bar{A}\bar{x} \leq \bar{b}, \end{array}$$

where

$$\bar{x} = \begin{bmatrix} x \\ y \end{bmatrix}, \quad \bar{c} = \begin{bmatrix} 0 \\ \mathbf{1} \end{bmatrix}, \quad \bar{A} = \begin{bmatrix} I & -I \\ -I & -I \\ A & 0 \\ -A & 0 \end{bmatrix}, \quad \bar{b} = \begin{bmatrix} 0 \\ 0 \\ b + \mathbf{1} \\ b - \mathbf{1} \end{bmatrix}.$$

Another good solution is to write x as the difference of two nonnegative vectors $x = x^+ - x^-$, and to express the problem as

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T x^+ + \mathbf{1}^T x^- \\ \text{subject to} & -\mathbf{1} \leq Ax^+ - Ax^- - b \leq \mathbf{1} \\ & x^+ \geq 0, \quad x^- \geq 0, \end{array}$$

which is an LP in $x^+ \in \mathbf{R}^n$ and $x^- \in \mathbf{R}^n$. In matrix notation,

$$\begin{array}{ll} \text{minimize} & \bar{c}^T \bar{x} \\ \text{subject to} & \bar{A}\bar{x} \leq \bar{b}, \end{array}$$

where

$$\bar{x} = \begin{bmatrix} x^+ \\ x^- \end{bmatrix}, \quad \bar{c} = \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix}, \quad \bar{A} = \begin{bmatrix} A & -A \\ -A & A \\ -I & 0 \\ 0 & -I \end{bmatrix}, \quad \bar{b} = \begin{bmatrix} \mathbf{1} + b \\ \mathbf{1} - b \\ 0 \\ 0 \end{bmatrix}.$$

(c) We can introduce new variables $y \in \mathbf{R}^m$ and $t \in \mathbf{R}$ and write the problem as

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T y + t \\ \text{subject to} & -y \leq Ax - b \leq y \\ & -t\mathbf{1} \leq x \leq t\mathbf{1}, \end{array}$$

which is an LP in x, y, and t. In matrix notation:

$$\begin{array}{ll} \text{minimize} & \bar{c}^T \bar{x} \\ \text{subject to} & \bar{A}\bar{x} \leq \bar{b}, \end{array}$$

where

$$\bar{x} = \begin{bmatrix} x \\ y \\ t \end{bmatrix}, \quad \bar{c} = \begin{bmatrix} 0 \\ \mathbf{1} \\ 1 \end{bmatrix}, \quad \bar{A} = \begin{bmatrix} A & -I & 0 \\ -A & -I & 0 \\ I & 0 & -\mathbf{1} \\ -I & 0 & -\mathbf{1} \end{bmatrix}, \quad \bar{b} = \begin{bmatrix} b \\ -b \\ 0 \\ 0 \end{bmatrix}.$$

Problem 8 (`1 points`, Exer. 10 in *Linear Programming Exercises*): Formulate the following problems as LPs:

(a) Given $A \in \mathbf{R}^{m \times n}$, $b \in \mathbf{R}^m$,

$$\text{minimize} \quad \sum_{i=1}^{m} \max\{0, a_i^T x + b_i\}.$$

The variable is $x \in \mathbf{R}^n$.

(b) Given $p+1$ matrices $A_0, A_1, \cdots, A_p \in \mathbf{R}^{m\times n}$, find the vector $x \in \mathbf{R}^p$ that minimizes

$$\max_{\|y\|_1=1} \|(A_0 + x_1 A_1 + \cdots + x_p A_p)y\|_1. \tag{3}$$

Hint: *you can use the identity* $\max_{\|y\|_1=1} \|Ay\|_1 = \max_{j=1,\cdots,n} \sum_{i=1,\cdots,m} |A_{ij}|$.

<u>**Solution**</u>:

(a) The problem can be expressed as the LP:

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T s \\ \text{subject to} & s \geq Ax + b, \\ & s \geq 0, \end{array}$$

with the variable $s \in \mathbb{R}^m$.

(b) We first note that for $A \in \mathbf{R}^{m\times n}$,

$$\max_{\|y\|_1=1} \|Ay\|_1 = \max_{j=1,\cdots,n} \sum_{i=1,\cdots,m} |A_{ij}|$$

(i.e., we add the absolute values in each column of A, and then take the maximum of those column sums).

You are not required to prove the identity given in the hint, however, we provide the following proof for completeness.

Since $\|z\|_1 = \max_{\|u\|_\infty \leq 1} u^T z$, we can write

$$\begin{aligned} \max_{\|y\|_1=1} \|Ay\|_1 &= \max_{\|u\|_\infty \leq 1} \max_{\|y\|_1=1} u^T A y \\ &= \max_{\|u\|_\infty \leq 1} \max_{\|y\|_1=1} \sum_{j=1}^{n} y_j \left(\sum_{i=1}^{m} A_{ij} u_i \right) \\ &= \max_{\|u\|_\infty \leq 1} \max_{j=1,\cdots,n} \left| \sum_{i=1}^{m} A_{ij} u_i \right| \\ &= \max_{j=1,\cdots,n} \max_{\|u\|_\infty \leq 1} \left| \sum_{i=1}^{m} A_{ij} u_i \right| \\ &= \max_{j=1,\cdots,n} \sum_{i=1}^{m} |A_{ij}| \end{aligned} \tag{4}$$

Using this expression we can formulate the problem as

$$\text{minimize} \quad \max_{j=1,\cdots,n} \sum_{i=1}^{m} |(A_0 + A_1 x_1 + \cdots + A_p x_p)_{ij}|$$

which can be formulated as an LP

$$
\begin{array}{ll}
\text{minimize} & t \\
\text{subject to} & -s_{ij} \leq (A_0 + A_1 x_1 + \cdots + A_p x_p)_{ij} \leq s_{ij} \\
 & \sum_{i=1}^{m} s_{ij} \leq t,\ j = 1, \cdots, n.
\end{array}
$$

The variables are $x, t \in \mathbb{R}$ and $S \in \mathbb{R}^{m \times n}$.