Algorithm Design and Analysis Assignment 4
Linear Programming

Xinmiao Zhang
202028013229129

December 17, 2020

1 Traveling Trump Problem

Suppose you are the U.S. presidential candidate Donald Trump, who want to hold election rallies in four swing states: *Georgia*(1), *Pennsylvania*(2), *Michigan*(3) and *Florida*(4). It is the last day before election and because of shortage of funds, you need to save money and try to travel through the shortest path, visit each state exactly once and return to the starting point. Distance between every two states can be writen as c_{ij}, $i \in [0,3]$,$j \in [i,4]$, i, j is integer. *Washington DC*(0) is your starting point. Please formulate this problem as an ILP. (*Hint*: You can think about this problem in terms of the constraint that **visiting each state exactly once**.)

1.1 Symbol Definition

The symbols using in linear programming expression is noted as follows:

Symbol	Meaning
c_{ij}	The distance between city i and $j (0 \le i < j \le 4)$
x_{ij}	If visit the road between city i and j, then $x_{ij} = 1$, else $x_{ij} = 0 (0 \le i < j \le 4)$

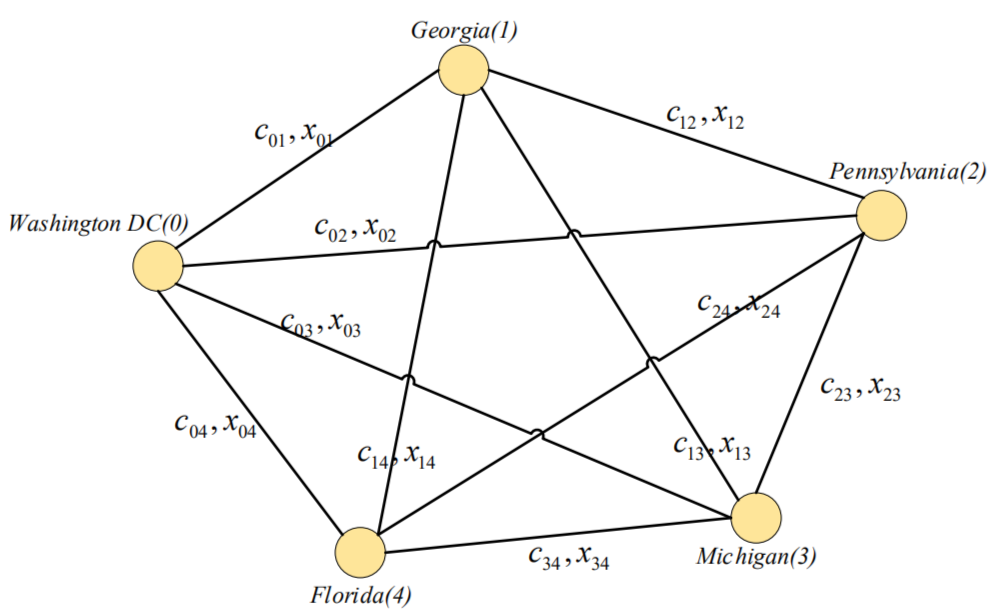


Figure 1: Symbol Definition in Problem 1

1.2 Linear Programming

We can obtain a linear programming goal and restriction as follows:

$$
\begin{aligned}
\min \quad & \sum_{i=0}^{3} \sum_{j=i+1}^{4} x_{ij} c_{ij} && && \text{(LP1)} \\
\text{s.t.} \quad & \sum_{j=0}^{i-1} x_{ji} + \sum_{j=i+1}^{4} x_{ij} = 2, && i = 0, 1, 2, 3, 4 && (1) \\
& x_{ij} = 0/1. && i = 0, 1, 2, 3; i \leq j \leq 4, j \in \mathbb{Z} && (2) \\
& && && (3)
\end{aligned}
$$

1.3 Explanation

The last sentence in problem gives us some enlightment. The constraint of the problem is Trump have to visit each state exactly once. Then for one city, there must be one way for arriving in, and another one for leaving. It is also can be proved that, if we have a path where exactly arrive in and leave each city once, it is a routine which have visited all the cities in graph exactly once. Considering once we have found a path, the direction or the travelling order have no impact to the result, as a consequence, we only need to find the undirected path which minimize the path length.

So for each road between two city i and j, if we visit it then x_{ij} is 1, else it is 0. Then the constraint above can be showed as for each city, the sum of x_{ij} of roads adjacent to it will be 2. (One for in and another for out) So that is the reason for constraint(1).

It is trivial to write the programming goal and constraint(2) according to the framework we have already proposed. Details will be hidden here.

2 Profit Maximization

Your factory produces three kinds of product: A, B and C. All of them need two kinds of raw materials: nickel and aluminum. The profit and cost of each kind of product are shown in the following table.

Product	Profit($)	Nickel(kg)	Aluminum(kg)
A	10	3	4
B	5	3	2
C	15	1	8

You only have 100 kg of nickle and 200 kg of aluminum in stock. How to arrange production to maximize profits? Please formulate this problem as a LP and transform it into dual form. Then you may solve both primal and dual problems using GLPK or Gurobi or other similar tools.

2.1 Symbol Definition

The symbols using in linear programming expression is noted as follows:

Symbol	Meaning
x_1	amount of production A
x_2	amount of production B
x_3	amount of production C

2.2 Linear Programming

We can obtain a linear programming goal and restriction as follows:

$$\begin{aligned} \max \quad & w = 10x_1 + 5x_2 + 15x_3 && \text{(LP2)} \\ \text{s.t.} \quad & 3x_1 + 3x_2 + x_3 \leq 100 && (4) \\ & 4x_1 + 2x_2 + 8x_3 \leq 200 && (5) \\ & x_1, x_2, x_3 \geq 0 && (6) \end{aligned}$$

2.3 Explanation

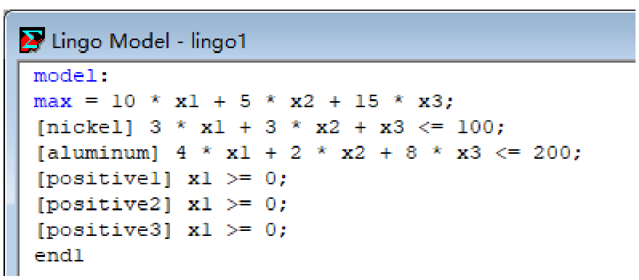
The problem is trivial, we save the explanation here.

2.4 Dual problem and Solution by Lingo

The dual of original Linear Programming is as follows:

$$\begin{aligned} \min \quad & z = 100y_1 + 200y_2 && \text{(LP3)} \\ \text{s.t.} \quad & 3y_1 + 4y_2 \geq 10 && (7) \\ & 3y_1 + 2y_2 \geq 5 && (8) \\ & y_1 + 8y_2 \geq 15 && (9) \\ & y_1, y_2 \geq 0 && (10) \end{aligned}$$

Lingo is a powerful application which can be used to quickly solve the linear programming problem. We use the lingo to solve (LP2) first, and using the code is as following:




```
model:
max = 10 * x1 + 5 * x2 + 15 * x3;
[nickel] 3 * x1 + 3 * x2 + x3 <= 100;
[aluminum] 4 * x1 + 2 * x2 + 8 * x3 <= 200;
[positive1] x1 >= 0;
[positive2] x1 >= 0;
[positive3] x1 >= 0;
endl
```

Figure 2: (LP2) in Lingo Code

And the solution is:

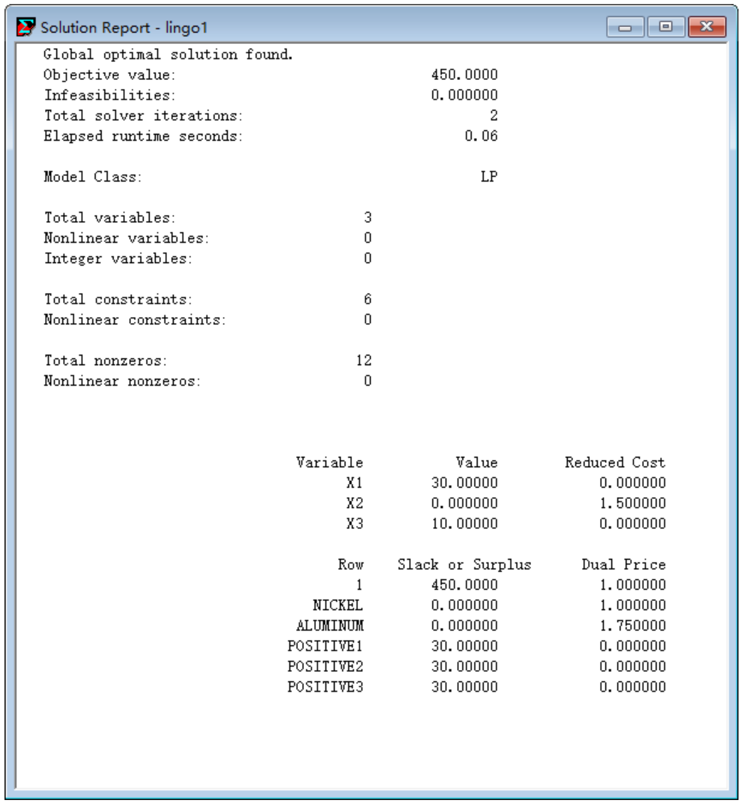


Figure 3: (LP2) result solved by Lingo

It can be seen when we produce 30 pieces of A, 0 pieces of B and 10 pieces of C, the profit can reach the maximization 450. Also we can obtain the status report when solving (LP2):

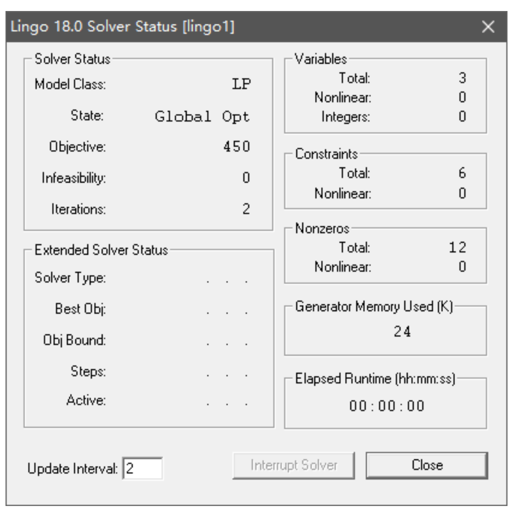


Figure 4: (LP2) solving status

It used 24KB and 2 iterations to finish the calculation, which is fast and efficient.

Also, we use the Lingo to solve the dual of (LP2), and the code used is as following:

```
Lingo Model - lingo1
model:
min = 100 * y1 + 200 * y2;
[constraint1] 3 * y1 + 4 * y2 >= 10;
[constraint2] 3 * y1 + 2 * y2 >= 5;
[constraint3] y1 + 8 * y2 >= 15;
[positive1] y1 >= 0;
[positive2] y2 >= 0;
end
```

Figure 5: (LP3) in Lingo Code

And the solution is:

```
Solution Report - lingo1
Global optimal solution found.
Objective value:                              450.0000
Infeasibilities:                              0.000000
Total solver iterations:                             2
Elapsed runtime seconds:                          0.08

Model Class:                                        LP

Total variables:                      2
Nonlinear variables:                  0
Integer variables:                    0

Total constraints:                    6
Nonlinear constraints:                0

Total nonzeros:                      10
Nonlinear nonzeros:                   0


                             Variable           Value        Reduced Cost
                                   Y1        1.000000            0.000000
                                   Y2        1.750000            0.000000

                                  Row    Slack or Surplus      Dual Price
                                    1        450.0000           -1.000000
                          CONSTRAINT1        0.000000           -30.00000
                          CONSTRAINT2        1.500000            0.000000
                          CONSTRAINT3        0.000000           -10.00000
                            POSITIVE1        1.000000            0.000000
                            POSITIVE2        1.750000            0.000000
```

Figure 6: (LP3) result solved by Lingo

It can be seen when $y_1 = 1.00, y_2 = 1.75$, the optimized goal can be minimized, which is also 450. Also we can obtain the status report when solving (LP3):

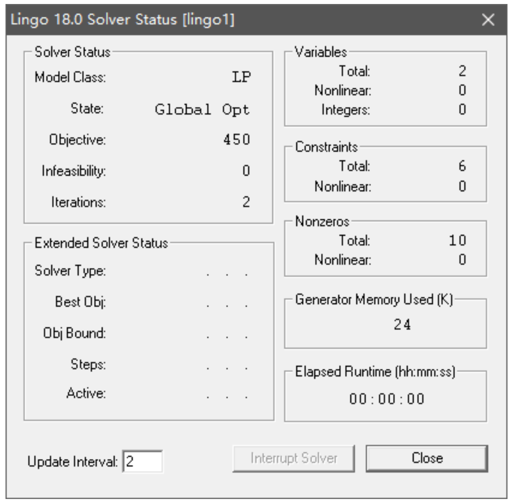


Figure 7: (LP3) solving status

It used 24KB and 2 iterations to finish the calculation, which is same as (LP2).

3 Cutting Paper Minimization

Your factory has expanded its bussiness. Suppose you have an unlimited number of large rolls of paper, of width W meters per roll (W is a positive integer). However, different m customers demands are for smaller width of paper; in particular, customer i needs b_i rolls of paper of width w_i, $i = 1, 2, ..., m$. We assume that $w_i \leq W$ for each i, and each w_i is an integer.

Smaller rolls are obtained by slicing a large roll in a certain way. You can slice one roll of paper for different customers only if their total width does not exceed W. The goal of you is to minimize the number of large rolls used while satisfying customer demand. Please formulate this problem as an ILP. Assume that there is no cost for slicing.

3.1 Symbol Definition

The symbols using in linear programming expression is noted as follows:

Symbol	Meaning
N	**Constant**: estimated the upper limitation of number of large paper
B	**Constant**: total number of smaller paper required
x_{ij}	number of using the jth$(1 \leq j \leq N)$ large paper to produce ith$(1 \leq i \leq m)$ kind of smaller paper

Constant Estimation:

For B, it is the total smaller pages customers required, we can evaluate its value from its definition:

$$B = \sum_{i=1}^{m} b_i$$

For N, considering the situation, where we using the large paper to produce the same type of smaller paper, then we can evaluate the upper limitation N as follows:

$$N = \sum_{i=1}^{m} b_i / \lfloor \frac{W}{w_i} \rfloor$$

3.2 Linear Programming

We can formulate the original problem as a *ILP* problem, its goal and restriction as follows:

$$\min \quad \sum_{i=0}^{N}[B^i(\sum_{j=0}^{m} x_{ij})] \tag{LP4}$$

$$\text{s.t.} \quad \sum_{j=1}^{N} x_{ij} = b_i, \qquad i = 1,2,..,m. \tag{11}$$

$$\sum_{i=1}^{m} x_{ij} w_i \le W, \qquad j = 1,2,..,N. \tag{12}$$

$$0 \le x_{ij} \le b_i, \qquad i = 1,2,..,m; j = 1,2,..,N \tag{13}$$

$$x_{ij} \in \mathbb{Z} \qquad i = 1,2,..,m; j = 1,2,..,N \tag{14}$$

3.3 Explanation

Suppose the number of large papers is n, apparently n is our optimized goal, we want to minimize n. And because N is just a value we estimate really roughly, so we have $\min n \le N$.

Now let's consider about the following matrix $X = (x_{ij})_{m\times N}$.

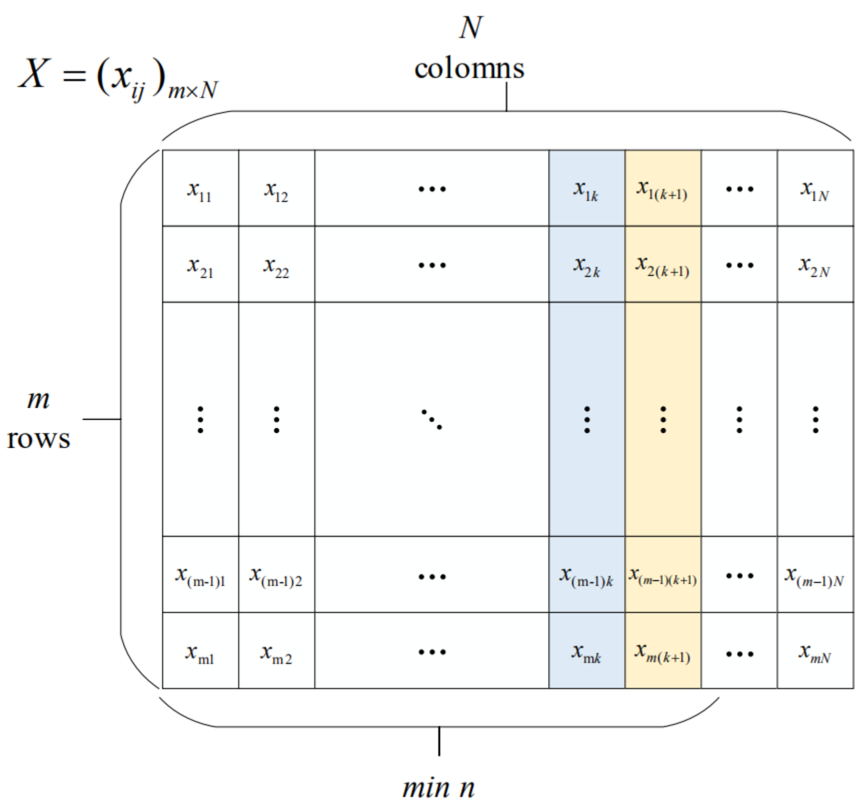


Figure 8: Matrix $X = (x_{ij})$

We can see from this matrix that, for row i, this vector shows how the N large paper produce b_i smaller paper for customer i. So we can make the constraint(10), for every row we should satisfy this equation.

Then for column j, this vector shows how the jth large paper produces smaller papers. It will at most produce m kinds of smaller paper, and their total width can not exceed W. So we can make the constraint(11), for every column we should satisfy this inequation.

Then it is time when for us to consider about the goal of our ILP. That is to say, to minimize n. We can further transform this problem, that is, how to make the *all-zero column* on the right side of matrix as much as possible?

We can use a weighted sum to show this. Notice that the value in one column can not exceed the production amount B, so we carefully design a goal as follows:

$$\sum_{i=0}^{N}[B^i(\sum_{j=0}^{m} x_{ij})]$$

For instance, consider about the kth column(blue in figure 8) and $k+1$th column(yellow in figure 8), if there is one value, say $x_{t(k+1)}$ is not zero, then its weighted value is at least B^{k+1}, which must larger than sum of k line's weighted value $B^k \sum_{i=1}^{m} x_{ik}$.

Using this optimized goal, we can get the minimized n.

4 Reformulation Problems with Absolute Values

Consider the problem:

$$\begin{gathered}\text{minimize } 2|x_1| + x_2 \\ \text{subject to } x_1 + x_2 \geq 4\end{gathered}$$

Please reformulate this problem as a LP without absolute values.

4.1 Symbol Definition

Symbol	Meaning
u	brand new variable to remove absolute value
v	brand new variable to remove absolute value

4.2 Linear Programming

The original LP problem can be transformed as follows:

$$\begin{aligned} \min \quad & 2u + 2v + x_2 && \text{(LP5)} \\ \text{s.t.} \quad & u - v + x_2 \geq 4 && (15) \\ & u, v, x_2 \geq 0 && (16) \\ & && (17) \end{aligned}$$

4.3 Explanation

Suppoes that $|x_1| = u + v$, $x_1 = u - v$, then we have:

$$\begin{cases} u = & (|x_1| + x_1)/2 \\ v = & (|x_1| - x_1)/2 \end{cases}$$

Apparently $u \geq 0$ and $v \geq 0$.(It can considered by supposing x_1 as positive and negative) And we can rewrite the LP goal and constraint accordingly.

5 Lawyer Recruitment for Trump

The U.S. presidential election is over, but Donald Trump refuses to accept defeat. Suppose you are Trump's election campaign manager and you are asked to recruit a group of lawyers for him to initiate litigation against the results. It is estimated that litigation will be initiated in N states, and the i(th) state needs at least L_i lawyers.

The number of law firms is F. Lawyers from the j(th) law firm can offer legal services in several states S_j and the recruitment fee for one lawyer from the j(th) law firm is C_j. Note that S_j is a subset of $N = \{1, 2, .., n\}$ and the union of S_j equals to N.

Your boss wants you to save money so your need to formulate this problem as an *ILP* and your goal is minimizing the recruitment fee of enough lawyers.

5.1 Symbol Definition

Symbol we use in the linear programming has already listed as follows:

Symbol	Meaning
s_{ij}	**Constant**: the jth element in set $S_i (1 \le i \le F)$
t_i	**Constant**: the size of $S_i (t_i = \|S_i\|)$
x_{ij}	the number of people from ith firm to jth state$(1 \le i \le F, 1 \le j \le N)$

5.2 Linear Programming

Using symbols defined before we have the following linear programming:

$$\min \quad \sum_{i=1}^{F} C_i \sum_{j=1}^{N} x_{ij} \tag{LP6}$$

$$\text{s.t.} \quad \sum_{i=1}^{F} x_{ij} \ge L_j, \qquad j = 1, 2, .., n \tag{18}$$

$$\sum_{j=1}^{n} x_{ij} - \sum_{j=1}^{t_i} x_{is_{ij}} = 0, \qquad i = 1, 2, .., F \tag{19}$$

$$x_{ij} \ge 0, \qquad i = 1, 2, .., F \quad j = 1, 2, .., N \tag{20}$$

$$x_{ij} \in \mathbb{Z}, \qquad i = 1, 2, .., F \quad j = 1, 2, .., N. \tag{21}$$

5.3 Explanation

First of all, we need to guarantee that for each state, there is enough lawyer to get this dirty job done, which formed constraint(18).

Second, we must guarantee all the x_{ij} appear when the firm i can handle the case in jth state. So we make sure that the total cases in one state minus all the position should have value is zero, which form constraint(19).

Constraint(20) and (21) is the basic requirements for *ILP*.

Finally, the ILP goal can be weighted sum for each different firm.