Homework Assignment (Problem Set) 1:

Xinyu Zhao

5 questions

Question 1:

SteelCo manufactures three types of steel at two different steel mills. During a given month, Mill 1 has 300 hours of blast furnace time available, whereas Mill 2 has 400 hours. Because of differences in the furnaces at each mill, the time and cost to produce a ton of steel differs for each mill and are shown in the following table. Each month, SteelCo must manufacture a total of at least 500 tons of Steel 1, 600 tons of Steel 2, and 400 tons of Steel 3 to meet demand; however, the total amount of Steel 2 manufactured should not exceed the combined amount of Steel 1 and Steel 3. Also, in order to maintain a roughly uniform usage of the two mills, management's policy is that the percentage of available blast furnace capacity (time) used at each mill should be the same. Clearly formulate a linear program (LP) to minimize the cost of manufacturing the desired steel.

Table 1

Mill	Steel 1		Steel 2		Steel 3	
	Cost ($)	Time (Min)	Cost ($)	Time (Min)	Cost ($)	Time (Min)
Mill 1	11	21	12	23	15	29
Mill 2	13	23	10	17	12	31

Solution:

x_1 = Amount Mill 1 produce Steel 1;

x_2 = Amount Mill 1 produce Steel 2;

x_3 = Amount Mill 1 produce Steel 3;

y_1 = Amount Mill 2 produce Steel 1;

y_2 = Amount Mill 2 produce Steel 2;

y_3 = Amount Mill 2 produce Steel 3;

LP problem: Minimize $11*x_1+12*x_2+15*x_3+13*y_1+10*y_2+12*y_3$

Constraints:

$x_1,x_2,x_3,y_1,y_2,y_3>=0$

$21*x_1+23*x_2+29*x_3<=18000$

$23*y_1+17*y_2+31*y_3<=24000$

$x_1+y_1>=500$; $x_2+y_2>=600$; $x_3+y_3>=400$

$x_2+y_2<=x_1+y_1+x_3+y_3$

$(21*x_1+23*x_2+29*x_3)/18000=(23*y_1+17*y_2+31*y_3)/24000$

```
In [6]: import pulp

        # Define the problem
        prob = pulp.LpProblem("Minimize_Cost", pulp.LpMinimize)

        # Define decision variables
        x1 = pulp.LpVariable('Amount Mill 1 produce Steel 1', lowBound=0)  # x1 >= 0
        x2 = pulp.LpVariable('Amount Mill 1 produce Steel 2', lowBound=0)  # x2 >= 0
        x3 = pulp.LpVariable('Amount Mill 1 produce Steel 3', lowBound=0)  # x3 >= 0
        y1 = pulp.LpVariable('Amount Mill 2 produce Steel 1', lowBound=0)  # y1 >= 0
        y2 = pulp.LpVariable('Amount Mill 2 produce Steel 2', lowBound=0)  # y2 >= 0
        y3 = pulp.LpVariable('Amount Mill 2 produce Steel 3', lowBound=0)  # y3 >= 0

        # Objective function:
        prob += 11 * x1 + 12 * x2 + 15 * x3 + 13 * y1 + 10 * y2 + 12 * y3

        # Constraints
        prob += 21 * x1 + 23 * x2 + 29 * x3 <= 18000
        prob += 23 * y1 + 17 * y2 + 31 * y3 <= 24000
        prob += x1 + y1 >= 500
        prob += x2 + y2 >= 600
        prob += x3 + y3 >= 400
        prob += x2 + y2 <= x1 + y1 + x3 + y3

        # Add the ratio constraint: (21*x1 + 23*x2 + 29*x3)/18000 = (23*x1 + 17*x2 + 31*x3)/24000
        # Multiply by the denominators to eliminate fractions
        prob += 24000 * (21 * x1 + 23 * x2 + 29 * x3) == 18000 * (23 * y1 + 17 * y2 + 31 * y3)

        # Solve the problem
        status=prob.solve()

        print(pulp.LpStatus[status])

        # Output the results
        result = {v.name: v.varValue for v in prob.variables()}
        total_cost = pulp.value(prob.objective)

        result, total_cost

        Optimal

Out[6]: ({'Amount_Mill_1_produce_Steel_1': 500.0,
          'Amount_Mill_1_produce_Steel_2': 180.41958,
          'Amount_Mill_1_produce_Steel_3': 0.0,
          'Amount_Mill_2_produce_Steel_1': 0.0,
          'Amount_Mill_2_produce_Steel_2': 419.58042,
          'Amount_Mill_2_produce_Steel_3': 400.0},
         16660.83916)
```

Solving the linear minimization problem, we get

Amount_Mill_1_produce_Steel_1: 500.0,

Amount_Mill_1_produce_Steel_2: 180.42,

Amount_Mill_1_produce_Steel_3: 0.0,

Amount_Mill_2_produce_Steel_1: 0.0,

Amount_Mill_2_produce_Steel_2: 419.58,

Amount_Mill_2_produce_Steel_3: 400

With the minimum cost 16660.83916

Question 2:

Consider the following linear program:

Max $Z = -4x_1 + 2x_2$
Subject To
$-2x_1 + 2x_2 \leq 7$
$x_1 \geq 2$
$x_1 - 4x_2 \leq 0$
$2x_1 + 2x_2 \geq 10$
$x_1, x_2 \geq 0$

Part A: Write the LP in standard equality form.

Part B: Solve the original LP graphically (<u>to scale</u>). Clearly identify the <u>feasible region</u> and, if one or more exist, the <u>optimal solution(s)</u> (provide exact values for x_1, x_2, and Z).

Solution:

Max $Z = -4x_1 + 2x_2$

Subject to

$-2x_1+2x_2+s_1=7$ ($s1\geq0$)

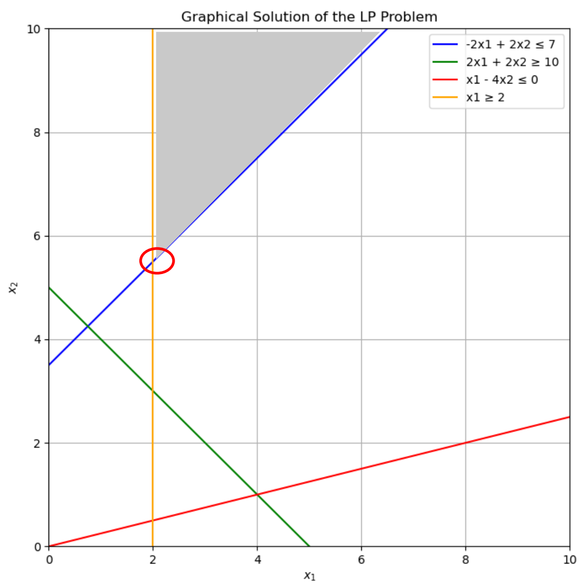
$x_1-s_2 = 2$ ($s2\geq0$)

$x_1 - 4x_2+s_3 = 0$ ($s3\geq0$)

$2x_1+2x_2-s_4=10$ ($s4\geq0$)

$x1,x2,s1,s2,s3,s4\geq0$

The grey area is the feasible region and the optimal solution is (x1=2, x2=5.5, Z=3)



Question 3:

At the beginning of month 1, Finco has $700 in cash. At the beginning of months 1, 2, 3, and 4, Finco receives certain revenues, after which it pays bills (see Table 2 below). Any money left over may be invested for one month at the interest rate of 0.1% per month; for two months at 0.5% per month; for three months at 1% per month; or for four months at 2% per month. Use linear programming to determine an investment strategy that maximizes cash on hand at the beginning of month 5. Formulate an LP to maximize Finco's profit.

Table 2

Month	Revenues ($)	Bills ($)
1	500	400
2	900	300
3	300	600
4	400	350

Solution:

Let $I_{i,j}$ denote as Amount of cash invested at the beginning of month t for m months.

Let C_z denoted as cash on hand at the z month.

Maximize C5=C4+1.001·I4,1+1.005·I3,2+1.01·I2,3+1.02·I1,4

Constraint:

Month 1: C1 = 700+500-400-(I1,1+I1,2+I1,3+I1,4)

Month 2: C2 = C1+900-300-(I2,1+I2,2+I2,3)+1.001*I1,1

Month 3: C3 = C2+300-600-(I3,1+I3,2)+1.001*I2,1+1.005*I1,2

Month 4: C4 = C3 + 400-350-I4,1+1.001*I3,1+1.005*I2,2+1.01*I1,3

Non-negativity: Everything greater than 0

```python
In [19]: from pulp import LpMaximize, LpProblem, LpVariable

# Define the problem
problem = LpProblem("Maximize_Cash_at_Beginning_of_Month_5", LpMaximize)

# Define decision variables
I = {
    (t, m): LpVariable(f"I_{t}_{m}", lowBound=0) for t in range(1, 5) for m in range(1, 6 - t)
}
C = {t: LpVariable(f"C_{t}", lowBound=0) for t in range(1, 6)}

# Objective function: Maximize C5
problem += (
    C[4] + 1.001 * I[4, 1] + 1.005 * I[3, 2] + 1.01 * I[2, 3] + 1.02 * I[1, 4],
    "Total_Cash_at_Beginning_of_Month_5",
)

# Constraints for cash flow
problem += C[1] == 700 + 500 - 400 - sum(I[1, m] for m in range(1, 5)), "Cash_Flow_Month_1"
problem += (
    C[2] == C[1] + 900 - 300 - sum(I[2, m] for m in range(1, 4)) + 1.001 * I[1, 1],
    "Cash_Flow_Month_2",
)
problem += (
    C[3] == C[2] + 300 - 600 - sum(I[3, m] for m in range(1, 3)) + 1.001 * I[2, 1] + 1.005 * I[1, 2],
    "Cash_Flow_Month_3",
)
problem += (
    C[4] == C[3] + 400 - 350 - sum(I[4, m] for m in range(1, 2)) + 1.001 * I[3, 1] + 1.005 * I[2, 2] + 1.01 * I[1, 3],
    "Cash_Flow_Month_4",
)

# Solve the problem
problem.solve()

# Extract results
result = {
    "Maximize_Cash": pulp.value(problem.objective),
    "Investments": {(t, m): I[t, m].varValue for t in range(1, 5) for m in range(1, 6 - t)},
}

result
```

```
Out[19]: {'Maximize_Cash': 1169.352697,
 'Investments': {(1, 1): 0.0,
  (1, 2): 0.0,
  (1, 3): 0.0,
  (1, 4): 800.0,
  (2, 1): 299.7003,
  (2, 2): 0.0,
  (2, 3): 300.2997,
  (3, 1): 0.0,
  (3, 2): 0.0,
  (4, 1): 50.0}}
```

From this code, we create multiple decision variables, and optimize the problem. We then set constraint at each step and finally solve the linear problem. The result show we should invest all \$800 left over for 4 months at month one; \$300 for 1 month and the rest \$300 for 3 month at month two; the \$300 saved for 1 month will offset the extra bill at month 3.

Question 4:

Turkeyco produces two types of turkey cutlets for sale to fast-food restaurants. Each type of cutlet consists of white meat and dark meat. Cutlet 1 sells for $4/lb and must consist of at least 70% white meat. Cutlet 2 sells for $3/lb and must consist of at least 60% white meat. At most, 50 lb of cutlet 1 and 30 lb of cutlet 2 can be sold. The two types of turkey used to manufacture the cutlets are purchased from the GobbleGobble Turkey Farm. Each type 1 turkey costs $10 and yields 5 lb of white meat and 2 lb of dark meat. Each type 2 turkey costs $8 and yields 3 lb of white meat and 3 lb of dark meat.

Part A: Formulate an LP to maximize Turkeyco's profit.

Part B: Solve the LP (provide exact values for all variables and the optimal objective function).

Solution:

Let t1 = number of type 1 turkey bought and t2 = number of type 2 turkey bought

Let c1 = weight of cutlet 1 sold and c2 = weight of cutlet 2 sold

Maximize Z = 4*c1 + 2*c2 - 10*t1 - 8*t2

Constraints:

c1<=50; c2<=30

5*t1+3*t2 >= c1*0.7+c2*0.6

2*t1+3*t2 >= c1*0.3+c2*0.4

Non-negative, t1,t2,c1,c2 >= 0

Answer: Max profit 177.555 when selling 50 cutlet 1 and 30 cutlet 2 and buying 8.6667 type 1 turkey and 3.2222 type 2 turkey

```
In [24]: import pulp

         # Define the problem
         prob = pulp.LpProblem("Maximize Profit", pulp.LpMaximize)

         # Define decision variables
         t1 = pulp.LpVariable('Turkey Type 1', lowBound=0)
         t2 = pulp.LpVariable('Turkey Type 2', lowBound=0)
         c1 = pulp.LpVariable('Cutlet 1', lowBound=0)
         c2 = pulp.LpVariable('Cutlet 2', lowBound=0)

         # Objective function:
         prob += 4*c1+3*c2-10*t1-8*t2

         # Constraints
         prob += c1<=50
         prob += c2<=30
         prob += 5*t1+3*t2>=c1*0.7+c2*0.6
         prob += 2*t1+3*t2>=c1*0.3+c2*0.4


         # Solve the problem
         status=prob.solve()

         print(pulp.LpStatus[status])

         # Output the results
         result = {v.name: v.varValue for v in prob.variables()}
         total_profit = pulp.value(prob.objective)

         result, total_profit

         Optimal

Out[24]: ({'Cutlet_1': 50.0,
           'Cutlet_2': 30.0,
           'Turkey_Type_1': 8.6666667,
           'Turkey_Type_2': 3.2222222},
          177.55555539999997)
```

Question 5:

A company wants to plan production for the ensuing year to minimize the combined cost of production and inventory costs. In each quarter of the year, demand is anticipated to be 130, 160, 250, and 150 units, respectively. The plant can produce a maximum of 200 units each quarter. The product can be manufactured at a cost of $15 per unit during the first quarter, however the manufacturing cost is expected to rise by $1 per quarter. Excess production can be stored from one quarter to the next at a cost of $1.50 per unit, but the storage facility can hold a maximum of 60 units. How should the production be scheduled so as to minimize the total costs?

Part A: Formulate an LP model to minimize costs.

Part B: Solve the LP (provide exact values for all variables and the optimal objective function).

Solution:

Let p1 = production during first quarter, p2=production during second quarter, p3 = production during third quarter, p4 = production of fourth quarter

Extra Storage of first quarter is p1-130

Extra Storage of second quarter is p2+(p1-130) – 160

Extra Storage of third quarter is p3+(p2+(p1-130) – 160)-250

Minimize 15*p1 + 16*p2 + 17*p3 + 18*p4 + 1.5*(p1-130)+1.5*(p2+(p1-130) – 160)+1.5*(p3+(p2+(p1-130) – 160)-250)

Constraints:

p1<=200; p2<=200; p3<=200; p4<=200 #cannot produce over 200

p1-130 <= 60 #cannot store more than 60

p2+(p1-130) – 160 <= 60 #cannot store more than 60

p3+(p2+(p1-130) – 160)-250 <= 60 #cannot store more than 60

p1>= 130 #Demand must be met

p2+(p1-130)>= 160 #Demand must be met

p3+(p2+(p1-130) – 160) >= 250 #Demand must be met

p4+(p3+(p2+(p1-130) – 160)-250)>=150 #Demand must be met

p1,p2,p3,p4>=0 #non-negative constraint

In [26]:

```python
import pulp

# Define the problem
prob = pulp.LpProblem("Minimize Cost", pulp.LpMinimize)

# Define decision variables
p1 = pulp.LpVariable('production of quarter 1', lowBound=0)
p2 = pulp.LpVariable('production of quarter 2', lowBound=0)
p3 = pulp.LpVariable('production of quarter 3', lowBound=0)
p4 = pulp.LpVariable('production of quarter 4', lowBound=0)

# Objective function:
prob += 15*p1+16*p2+17*p3+18*p4+1.5*(p1-130)+1.5*(p2+(p1-130)-160)+1.5*(p3+(p2+(p1-130)-160)-250)

# Constraints
prob += p1<=200
prob += p2<=200
prob += p3<=200
prob += p4<=200
prob += p1-130<=60
prob += p2+(p1-130)-160<=60
prob += p3+(p2+(p1-130)-160)-250<=60
prob += p1>=130
prob += p2+(p1-130)>=160
prob += p3+(p2+(p1-130)-160)>=250
prob += p4+(p3+(p2+(p1-130)-160)-250)>=150


# Solve the problem
status=prob.solve()

print(pulp.LpStatus[status])

# Output the results
result = {v.name: v.varValue for v in prob.variables()}
total_cost = pulp.value(prob.objective)

result, total_cost
```

```
Optimal
```

Out[26]:

```
({'production_of_quarter_1': 140.0,
  'production_of_quarter_2': 200.0,
  'production_of_quarter_3': 200.0,
  'production_of_quarter_4': 150.0},
 11490.0)
```