NATIONAL UNIVERSITY OF SINGAPORE

BDC5101: Deterministic Operations Research
Assignment 1
Swetha Narayanan – A0074604J

1a)

Parts of the original model which were not linear before:

a) Objective function has a quadratic term : $3x_1^2 - 2x_2$
b) One of the constraints is not linear : $0 <= |x_1| <= 1$

1b)

Constraint 1 : $| x_1| + | x_2| <= 1.5$
This is equivalent to the following constraints:
$-1.5 <= x_2 + x_1 <= 1.5$
$-1.5 <= x_2 - x_1 <= 1.5$

Constraint 2 : $0 <= | x_1| <= 1$
This is equivalent to the following constraint:
$-1 <= x_1 <= 1$

Constraint 3 : $0 <= | x_2| <= 1$
This is equivalent to the following constraint:
$-1 <= x_2 <= 1$

General LP Form
min $3x_1 - 2x_2$

s.t
$-x_2 - x_1 <= 1.5$
$x_2 + x_1 <= 1.5$
$x_1 - x_2 <= 1.5$
$x_2 - x_1 <= 1.5$

$-1 <= x_1 <= 1$
$-1 <= x_2 <= 1$

1c)

Screenshots from the code :

```
In [20]:
for v in prob.variables():
        print (v.name, "=", v.varValue, "\tReduced Cost =", v.dj)

print ("objective=", value(prob.objective))

print ("\nSensitivity Analysis\nConstraint\t\tShadow Price\tSlack")
for name, c in prob.constraints.items():
        print (name, ":", c, "\t", c.pi, "\t\t", c.slack)

print("Status:", LpStatus[prob.status])

value(x1)
value(x2)
```

```
x1 = -1.0       Reduced Cost = 1.0
x2 = 0.5        Reduced Cost = 0.0
objective= -4.0

Sensitivity Analysis
Constraint              Shadow Price    Slack
_C1 : x1 + x2 <= 1.5     0.0             2.0
_C2 : x1 + x2 >= -1.5    0.0             -1.0
_C3 : x1 + x2 <= 1.5     0.0             2.0
_C4 : -x1 + x2 <= 1.5    -2.0            -0.0
_C5 : -x1 + x2 >= -1.5   0.0             -3.0
Status: Optimal

Out[20]: 0.5
```

Optimal value of model : -4.0
Values of d.v @Optimal Value:
$x_1 = -1.0$
$x_2 = 0.5$
Tight binding constraints @Optimal value:
$|x_1| + |x_2| <= 1.5$
$|x_1| <= 1$

2a)

LP Model

Minimize total cost

$$Z = 32000*(r_1 + r_2 + r_3 + r_4 + r_5 + r_6) + 40000*(o_1 + o_2 + o_3 + o_4 + o_5 + o_6) + 5000*(i_1 + i_2 + i_3 + i_4 + i_5 + i_6)$$

s.t

$r_j <= 50000$ in month j (where j = 1,2,3,4,5,6)
$o_j <= 25000$ in month j (where j = 1,2,3,4,5,6)
$r_1 + o_1 + 4500 - i_1 = 50000$ month 1
$r_2 + o_2 + i_1 - i_2 = 25000$ month 2
$r_3 + o_3 + i_2 - i_3 = 25000$ month 3
$r_4 + o_4 + i_3 - i_4 = 42000$ month 4
$r_5 + o_5 + i_4 - i_5 = 55000$ month 5
$r_6 + o_6 + i_5 - i_6 = 67000$ month 6

$r_j, o_j, i_j >= 0$

2b)

Results and Production Plan

Total Cost (Objective) = $8,465,000000.0

Month	Before	Jan	Feb	Mar	Apr	May	June
Demand	-	50000	25000	25000	42000	55000	67000
Inventory storage	4500	0	0	0	5000	0	0
Normal Production	-	45500	25000	25000	47000	50000	50000
Overtime Production	-	0	0	0	0	0	17000
Demand Met	-	Normal + Inventory = 45500+4500 = 50000	Normal = 25000	Normal = 25000	Normal = 42000	Normal +Inventory = 50000+5000 = 55000	Normal +Overtime =50000+17000 =67000

Screenshots from the model:

```python
from pulp import *
import pandas as pd
```

```python
# declare your variables
# r(j) - Regular production of EVs in month j where j = 1,2,3,4,5,6
# o(j) - Overtime production of EVs in month j where j = 1,2,3,4,5,6
# i(j) - Extra EVs carried over as inventory in month j where j = 1,2,3,4,5,6
r1 = LpVariable("r1", 0 ,None,LpContinuous) # 0<= r1 <= Infinity
r2 = LpVariable("r2", 0 ,None,LpContinuous) # 0<= r2 <= Infinity
r3 = LpVariable("r3", 0 ,None,LpContinuous) # 0<= r3 <= Infinity
r4 = LpVariable("r4", 0 ,None,LpContinuous) # 0<= r4 <= Infinity
r5 = LpVariable("r5", 0 ,None,LpContinuous) # 0<= r5 <= Infinity
r6 = LpVariable("r6", 0 ,None,LpContinuous) # 0<= r6 <= Infinity
i1 = LpVariable("i1", 0 ,None,LpContinuous) # 0<= i1 <= Infinity
i2 = LpVariable("i2", 0 ,None,LpContinuous) # 0<= i2 <= Infinity
i3 = LpVariable("i3", 0 ,None,LpContinuous) # 0<= i3 <= Infinity
i4 = LpVariable("i4", 0 ,None,LpContinuous) # 0<= i4 <= Infinity
i5 = LpVariable("i5", 0 ,None,LpContinuous) # 0<= i5 <= Infinity
i6 = LpVariable("i6", 0 ,None,LpContinuous) # 0<= i6 <= Infinity
o1 = LpVariable("o1", 0 ,None,LpContinuous) # 0<= o1 <= Infinity
o2 = LpVariable("o2", 0 ,None,LpContinuous) # 0<= o2 <= Infinity
o3 = LpVariable("o3", 0 ,None,LpContinuous) # 0<= o3 <= Infinity
o4 = LpVariable("o4", 0 ,None,LpContinuous) # 0<= o4 <= Infinity
o5 = LpVariable("o5", 0 ,None,LpContinuous) # 0<= o5 <= Infinity
o6 = LpVariable("o6", 0 ,None,LpContinuous) # 0<= o6 <= Infinity
```

```python
# defines the objective function to maximize
prob += 32000*(r1 + r2+ r3+ r4+ r5+ r6)+40000*(o1 + o2+ o3+ o4+ o5+ o6)+ 5000*(i1 + i2+ i3+ i4+ i5+ i6)
#where
#Regular Production cost of per EV : $32000
#Overtime Production cost of per EV : $40000
#Inventory cost of per EV per month : $5000
```

```python
# defines the constraints

#Production capacity constraints for regular and overtime production
prob += r1 <= 50000
prob += r2 <= 50000
prob += r3 <= 50000
prob += r4 <= 50000
prob += r5 <= 50000
prob += r6 <= 50000
prob += o1 <= 25000
prob += o2 <= 25000
prob += o3 <= 25000
prob += o4 <= 25000
prob += o5 <= 25000
prob += o6 <= 25000

#Meet the demand
prob += r1 + o1 + 4500 - i1 == 50000
prob += r2 + o2 + i1 - i2 == 25000
prob += r3 + o3 + i2 - i3 == 25000
prob += r4 + o4 + i3 - i4 == 42000
prob += r5 + o5 + i4 - i5 == 55000
prob += r6 + o6 + i5 - i6  == 67000
```

```
for v in prob.variables():
        print (v.name, "=", v.varValue, "\tReduced Cost =", v.dj)

print ("objective=", value(prob.objective))

print ("\nSensitivity Analysis\nConstraint\t\tShadow Price\tSlack")
for name, c in prob.constraints.items():
        print (name, ":", c, "\t", c.pi, "\t\t", c.slack)

print("Status:", LpStatus[prob.status])
```

```
i1 = 0.0        Reduced Cost = 5000.0
i2 = 0.0        Reduced Cost = 5000.0
i3 = 0.0        Reduced Cost = 5000.0
i4 = 5000.0     Reduced Cost = 0.0
i5 = 0.0        Reduced Cost = 2000.0
i6 = 0.0        Reduced Cost = 45000.0
o1 = 0.0        Reduced Cost = 8000.0
o2 = 0.0        Reduced Cost = 8000.0
o3 = 0.0        Reduced Cost = 8000.0
o4 = 0.0        Reduced Cost = 8000.0
o5 = 0.0        Reduced Cost = 3000.0
o6 = 17000.0    Reduced Cost = 0.0
r1 = 45500.0    Reduced Cost = 0.0
r2 = 25000.0    Reduced Cost = 0.0
r3 = 25000.0    Reduced Cost = 0.0
r4 = 47000.0    Reduced Cost = 0.0
r5 = 50000.0    Reduced Cost = 0.0
r6 = 50000.0    Reduced Cost = 0.0
objective= 8465000000.0
```

```
Sensitivity Analysis
Constraint                  Shadow Price    Slack
_C1 : r1 <= 50000            0.0             4500.0
_C2 : r2 <= 50000            0.0             25000.0
_C3 : r3 <= 50000            0.0             25000.0
_C4 : r4 <= 50000            0.0             3000.0
_C5 : r5 <= 50000            -5000.0                 -0.0
_C6 : r6 <= 50000            -8000.0                 -0.0
_C7 : o1 <= 25000            0.0             25000.0
_C8 : o2 <= 25000            0.0             25000.0
_C9 : o3 <= 25000            0.0             25000.0
_C10 : o4 <= 25000           0.0             25000.0
_C11 : o5 <= 25000           0.0             25000.0
_C12 : o6 <= 25000           0.0             8000.0
_C13 : -i1 + o1 + r1 = 45500      32000.0                -0.0
_C14 : i1 - i2 + o2 + r2 = 25000         32000.0                 -0.0
_C15 : i2 - i3 + o3 + r3 = 25000         32000.0                 -0.0
_C16 : i3 - i4 + o4 + r4 = 42000         32000.0                 -0.0
_C17 : i4 - i5 + o5 + r5 = 55000         37000.0                 -0.0
_C18 : i5 - i6 + o6 + r6 = 67000         40000.0                 -0.0
Status: Optimal
```

2c)

Month	Before	Jan	Feb	Mar	Apr	May	June
Demand	-	60000	85000	85000	100000	55000	87000
Inventory storage	4500	45000	35000	25000	0	12000	0
Normal Production	-	50000	50000	50000	50000	50000	50000
Overtime Production	-	50500	25000	25000	25000	17000	25000
Demand Met	-	Normal + Inventory + Overtime = 50000+4500+5500 = 60000	Normal + Inventory = 50000 + 35000 = 85000	Normal + Inventory = 50000+ 35000 = 85000	Normal + Overtime + Inventory =50000+25000 +25000 =100000	Normal + Overtime = 50000+5000 = 55000	Normal +Overtime + Inventory =50000+25000+ 12000 =87000

Its impossible to meet the demand for Feb, Mar, Apr without increasing regular or overtime capacity. The solution is infeasible with current capacity constraints.

At-least for the month of Jan, overtime production capacity needs to be increased from 25000 to 50500 units to meet demands for Feb, Mar and April. The company can look into shadow prices and sensitivity analysis to determine how much more profit can be obtained by increasing capacity.

2d)

- Estimate the demand curve based on the industry
- Forecast demand based on past demand
- Monte Carlo simulation for variance of demand
- Study research papers on industry demand trends for that company

3a)

We assume the following amount of raw materials are available:

a) M1 : 48
b) M2 : 20
c) M3 : 8

Raw material mix for each type of product (using dual form)

a) To produce product A, we need 8 of M1, 4 of M2 and 2 M3
b) To produce product B, we need 6 of M1, 2 of M2 and 1.5 M3
c) To produce product C, we need 1 of M1, 1.5 of M2 and 0.5 M3

3b)

Standard form

Maximize $60x_1 + 30x_2 + 20x_3 + 0s_1 + 0s_2 + 0s_3$

s.t

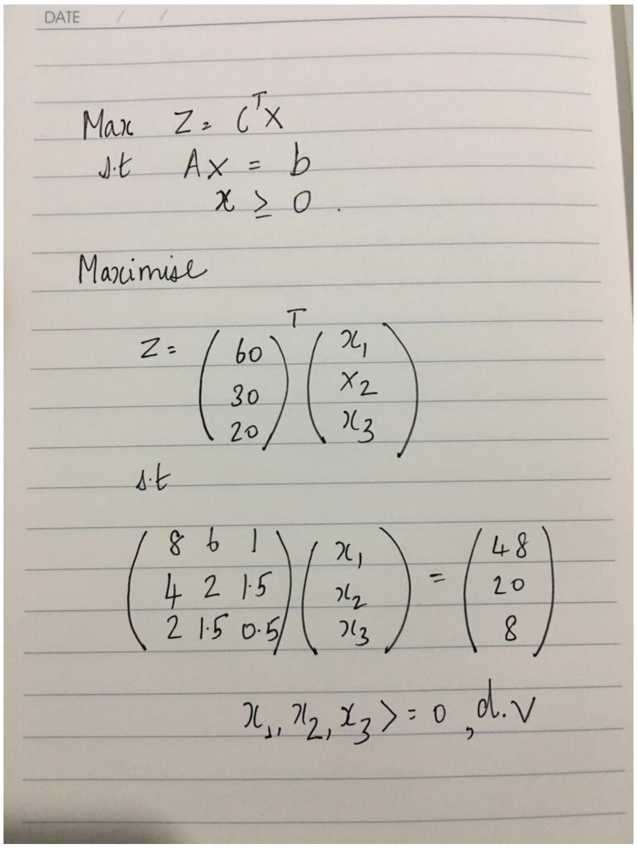
$$\begin{aligned}
8x_1 + 6x_2 + x_3 + s_1 &= 48 \\
4x_1 + 2x_2 + 1.5x_3 + s_2 &= 20 \\
2x_1 + 1.5x_2 + 0.5x_3 + s_3 &= 8
\end{aligned}$$

$x_1, x_2, x_3, s_1, s_2, s_3 >= 0$

where

x_1, x_2, x_3 are no. of products of A,B,C
s_1, s_2, s_3 are slack variables

Matrix Form



Max $Z = C^T X$

s.t $AX = b$

$X \geq 0$.

Maximise

$$Z = \begin{pmatrix} 60 \\ 30 \\ 20 \end{pmatrix}^T \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

s.t

$$\begin{pmatrix} 8 & 6 & 1 \\ 4 & 2 & 1.5 \\ 2 & 1.5 & 0.5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 48 \\ 20 \\ 8 \end{pmatrix}$$

$x_1, x_2, x_3 >= 0$, d.v

3c)

Basic Feasible Solutions (1)

$s_1 = 48$
$s_2 = 20$
$s_3 = 8$
$x_1 = 0$
$x_2 = 0$
$x_3 = 0$

Objective Function Value : 0

Basic Feasible Solutions (2)

$s_1 = 16$
$s_2 = 4$
$x_1 = 4$
$s_3 = 0$
$x_2 = 0$

$x_3 = 0$

Objective Function Value: 240

Basic variables: After applying Gauss-Jordan method, it's a variable with co-efficient of 1 in one equation and coefficient of 0 in other equations.

The above solutions are basic and feasible (satisfy the constraints) ->

$$8x_1 + 6x_2 + x_3 + s_1 = 48$$
$$4x_1 + 2x_2 + 1.5x_3 + s_2 = 20$$
$$2x_1 + 1.5x_2 + 0.5x_3 + s_3 = 8$$

Hence they are qualifying basic feasible solutions, but they are not necessarily optimal.

3d) **Dual Form**

Minimize $W = 48y_1 + 20y_2 + 8y_3$

s.t

$$8y_1 + 4y_2 + 2y_3 >= 60$$
$$6y_1 + 2y_2 + 1.5y_3 >= 30$$
$$y_1 + 1.5y_2 + 0.5y_3 >= 20$$

where y1, y2, y3 are the amount of raw materials M1, M2, M3 used respectively

Interpretation:
The dual problem: a market seeks to eliminate the arbitrage opportunities by choosing the right prices for resources. Shadow prices or the solution to the dual problem can be interpreted as the opportunity costs associated with consuming the firm's resources. Say we were renting out the raw materials used to produce A,B,C, we want the rent rate to be at-least the opportunity cost of producing A,B,C.

We want to minimize the use of raw materials y1, y2, y3 such that

The cost of producing product A is at-least 60$
The cost of producing product B is at-least 30$
The cost of producing product C is at-least 20$

3e)

According to the weak duality theorem of LP,

The (primal) objective value of any feasible primal solution is no larger than the (dual) objective value of any feasible dual solution. Hence, primal optimal value <= dual optimal value

Since a feasible solution is always less than or equal to the optimal solution for a primal maximization problem, any feasible solution to the primal problem corresponds to a lower bound on any solution to the dual problem.

primal feasible value <= primal optimal value <= dual optimal value

If primal has a feasible solution of $x_1 = 1$, $x_2 = 1$, $x_3 = 1$,
the optimal value is $60x_1 + 30x_2 + 20x_3 = 110$

Dual model's lower bound is 110.

3f)

Optimal Solution to the primal : Objective Function Value: 280
$x_1 = 2$
$x_2 = 0$
$x_3 = 8$

```
# defines the constraints
prob += 8*x1+6*x2+ x3 <= 48
prob += 4*x1+2*x2+ 1.5*x3 <= 20
prob += 2*x1+1.5*x2+ 0.5*x3 <= 8
```

```
#solve
prob.writeLP("Assignment1Prob3.lp")
prob.solve()
```

```
1
```

```
for v in prob.variables():
        print (v.name, "=", v.varValue, "\tReduced Cost =", v.dj)

print ("objective=", value(prob.objective))

print ("\nSensitivity Analysis\nConstraint\t\tShadow Price\tSlack")
for name, c in prob.constraints.items():
        print (name, ":", c, "\t", c.pi, "\t\t", c.slack)

print("Status:", LpStatus[prob.status])
```

```
x1 = 2.0        Reduced Cost = -0.0
x2 = 0.0        Reduced Cost = -5.0
x3 = 8.0        Reduced Cost = -0.0
objective= 280.0

Sensitivity Analysis
Constraint              Shadow Price    Slack
_C1 : 8*x1 + 6*x2 + x3 <= 48      -0.0          24.0
_C2 : 4*x1 + 2*x2 + 1.5*x3 <= 20.0      10.0            -0.0
_C3 : 2*x1 + 1.5*x2 + 0.5*x3 <= 8.0     10.0            -0.0
Status: Optimal
```

From the solution, we get the shadow prices, or the dual variables y_1, y_2, y_3
Price of raw material M1 (y_1) = 0
Price of raw material M2 (y_2) = 10$

Price of raw material M3 (y_3) = 10$

Optimal solution to the dual objective = $48y_1 + 20y_2 + 8y_3 = 280$

If there is another company wishing to buy all the raw materials from this company, they should offer a minimum price of 280$ as a whole and 10$ each for M2 and M3.

4a)

Total Revenue of A & B = $23x_1 + 40x_2$ (x_1 , x_2 are no. of units of product A and B)
Total Cost of Product A :
= x_1 * (5 (cost of raw material L) + 2 (no of units of raw material L) + 10 (cost of raw material W) * 1 (no of units of raw material W))
= $20x_1$
Total Cost of Product B :
= x_2 * (5 (cost of raw material L) * 3 (no of units of raw material L) + 10 (cost of raw material W) * 2 (no of units of raw material W))
= $35x_2$

Profit = Revenue – Cost = $23x_1 + 40x_2 - 20x_1 - 35x_2 = 3x_1 + 5x_2$

Amount of Raw Material L Available = 25
Amount of Raw Material W Available = 15

LP Model:

Maximize $Z = 3x_1 + 5x_2$

s.t

$2x_1 + 3x_2 <= 25$
$x_1 + 2x_2 <= 15$

$x_1, x_2 >= 0$, d.v

where x_1, x_2 are number of units of Product A and B respectively,

Screenshots from Model:

In [1]:
```
from pulp import *
import pandas as pd
```

In [6]:
```
# declare your variables
x1 = LpVariable("x1", 0,None,LpContinuous)   # 0<= x1 <= Infinity
x2 = LpVariable("x2", 0,None,LpContinuous) # 0<= x2 <= Infinity
```

In [7]:
```
# defines the problem
prob = LpProblem("problem", LpMaximize)
```

In [8]:
```
# defines the objective function to maximize
prob += 3*x1+5*x2
```

In [9]:
```
# defines the constraints
prob += 2*x1 + 3*x2 <= 25
prob += x1 + 2*x2 <= 15
```

In [10]:
```
#solve
prob.writeLP("Assignment1Prob4.lp")
prob.solve()
```

Out[10]:
```
1
```

In [11]:
```
for v in prob.variables():
        print (v.name, "=", v.varValue, "\tReduced Cost =", v.dj)

print ("objective=", value(prob.objective))

print ("\nSensitivity Analysis\nConstraint\t\tShadow Price\tSlack")
for name, c in prob.constraints.items():
        print (name, ":", c, "\t", c.pi, "\t\t", c.slack)

print("Status:", LpStatus[prob.status])

value(x1)
value(x2)
```

```
x1 = 5.0        Reduced Cost = -0.0
x2 = 5.0        Reduced Cost = -0.0
objective= 40.0

Sensitivity Analysis
Constraint              Shadow Price    Slack
_C1 : 2*x1 + 3*x2 <= 25          1.0             -0.0
_C2 : x1 + 2*x2 <= 15    1.0             -0.0
Status: Optimal
```

Out[11]:
```
5.0
```

Optimal Profit : 40$

No. of optimal units of Product A : 5

No. of optimal units of Product B : 5

4b)

- Shadow prices are 1$ each
- Profit is increased by 1$ if no of raw materials L is increased by 1 unit
- Profit is increased by 1$ if no of raw materials W is increased by 1 unit

5a)

1. Linear Programming optimal solution is always a corner point of the feasible region while Integer Programming optimal solution need not be a corner point. Simplex method is efficient because of this corner-point feasible guarantee for LP problems. Without this guarantee, IP problems are more challenging to find optimal solution.
2. The objective function value is no better or sometimes worse, for the optimal IP solution than for the optimal LP solution
3. Since IP solutions take a long time to solve – exponential complexity compared to Simplex (cos of (1)), we sometimes have to come up with a heuristic approach to solve IP problems

5b)

Weak Duality : Any feasible solution to the dual problem corresponds to an upper bound on any solution to the primal problem

Strong Duality : If an LP model has an optimal solution, then its dual also has a solution and the respective optimal value are equal.

To put it mathematically,

Weak duality: If x' is a feasible solution to (P) and y'is a feasible solution to (D), then:

$$c^T x' \leq b^T y'$$

If equality holds in the above inequality, then x' is an optimal solution to (P) and y' is an optimal solution to (D).

Strong duality: If there exists an optimal solution x′ for (P), then there exists an optimal solution y′ for (D) and the value of x′ in (P) equals the value of y′ in (D)

5c)

What are shadow prices and how do we use/interpret it?

1. When the right-hand side of a constraint in ≤ form gives the amount available of a certain resource, the shadow price for that resource is the rate at which the optimal value of the objective function of the primal problems could be

increased by increasing the amount of this resource being made available by 1 unit.

2. Shadow prices can be interpreted as the opportunity costs associated with consuming a firm's resources. If we value the firm's total resources at the shadow prices, we find their value is exactly equal to the optimal value of the objective function of the firm's profit maximization problem. Imagining a situation where the firm does not own the productive capacity but needs to rent it out, the shadow prices is equivalent to the rates it will charge for renting the resources.

How do we calculate it?

1. The decision variables of the optimal solutions to the dual give the shadow prices of the primal.
2. The shadow prices of the dual give the decision variables of the primal

5d) Typical features we could leverage to assess difficulty of a model in general:

1. No of decision variables
2. No of constraints
3. Are the decision variables decimal or strictly integer values?
4. Is the objective function linear or non-linear?
5. Is the constraint linear or non linear?
6. If the constraint is non-linear, can it be converted to a linear constraint so as to use LP?
7. Can the model be solved in polynomial time? What heuristic algorithms can we use to get greedy quicker solutions?