IE 400 Principles of Engineering Management

Project Report

Group 3

Name	Student ID	Section
Rüzgar Ayan	21801984	3

From the project document: “A unit of an acceptable chemotherapy regimen is 100 centilitres (cl) and must include one or more of these drugs.”

Assumption: The acceptable regimens must be of exactly that size, not less. For example, a regimen of 90 centilitres is not acceptable in my models but only 100 centilitres are acceptable. Also note that I needed to increase this default number of 100 centilitres in parts b and c due to infeasible solutions.

1 Part a)

1.1 Decision Variables:

X_i:The amount of drug i used, $i \in \{1,..., 7\}$

```
X = {}
for i in range(1,8):
    X[i] = m.addVar(vtype=GRB.CONTINUOUS, name="X"+str(i))
```

1.2 Parameters:

P_i:Patient features, $i \in \{1,..., 9\}$

Y_i:Whether or not the drug i is used, this is a parameter since we use the base regimen for this part, $i \in \{1,..., 7\}$

Min_i:The minimum amount of drug i that can be used, $i \in \{1,..., 7\}$

Max_i:The maximum amount of drug i that can be used, $i \in \{1,..., 7\}$

$total$:The total size of an acceptable regimen

These parameters are defined as follows in the code:

```
P = {1:1, 2:0, 3:1, 4:0, 5:0, 6:0, 7:0, 8:0, 9:1}
Y = {1:1, 2:0, 3:1, 4:1, 5:0, 6:0, 7:1}
minDosages = {1:20, 2:10, 3:20, 4:10, 5:10, 6:20, 7:20}
maxDosages = {1:80, 2:50, 3:100, 4:100, 5:70, 6:90, 7:50}
totalDosageCl = 100
```

1.3 Objective:

We are trying to maximize the Q-score function given in the assignment:

$$
\begin{aligned}
Q(\mathbf{p},\mathbf{y},\mathbf{x}) = & -5p_1 - 0.5p_2 - 12p_3 - 8p_4 - 5p_5 - 5p_6 - 1p_7 - 3p_8 - 2p_9 \\
& - 5y_1 - 6y_2 - 4y_3 - 4y_4 - 8y_5 - 6y_6 - 7y_7 \\
& + 0.28x_1 + 0.30x_2 + 0.25x_3 + 0.17x_4 + 0.31x_5 + 0.246x_6 + 0.40x_7
\end{aligned}
$$

In the code, this is written as:

```
m.setObjective(5*-P[1] + 0.5*-P[2] + 12*-P[3] + 8*-P[4] + 5*-P[5] +
5*-P[6] + 1*-P[7] + 3*-P[8] + 2*-P[9]

+ 5*-Y[1] + 6*-Y[2] + 4*-Y[3] + 4*-Y[4] + 8*-Y[5] + 6*-Y[6] + 7*-Y[7]

+ 0.28*X[1] + 0.30*X[2] + 0.25*X[3] + 0.17*X[4] + 0.31*X[5] + 0.246*X[6] +
0.40*X[7]
, GRB.MAXIMIZE)
```

1.4 Constraints:

$X_i = 0,\ i \in \{2, 5, 6\}$: These three drug types are not included in the base regimen and adding new drug types is not allowed in this part, so we force these amounts to be 0. Note that we could just not use these 3 decision variables at all and use these like parameters, but I still included them since it looks more complete.

```
fixX2 = m.addConstr(X[2] == 0)
fixX5 = m.addConstr(X[5] == 0)
fixX6 = m.addConstr(X[6] == 0)
```

$X_i \geq Min_i,\ i \in \{1, 3, 4, 7\}$

$X_i \leq Max_i,\ i \in \{1, 3, 4, 7\}$

These four drug types are included in the base regimen and we must definitely use them since removing drugs is not a possibility in this part. So, we force the amounts of these drugs to be in the given intervals.

```
minX1 = m.addConstr(X[1] >= minDosages[1])
maxX1 = m.addConstr(X[1] <= maxDosages[1])

minX3 = m.addConstr(X[3] >= minDosages[3])
maxX3 = m.addConstr(X[3] <= maxDosages[3])

minX4 = m.addConstr(X[4] >= minDosages[4])
maxX4 = m.addConstr(X[4] <= maxDosages[4])

minX7 = m.addConstr(X[7] >= minDosages[7])
maxX7 = m.addConstr(X[7] <= maxDosages[7])
```

$\sum_{i=1}^{7} X_i = total$: The sum of all the drug amounts must be equal to 100 centiliters to be an acceptable regimen.

```
totalDosage = m.addConstr(quicksum([X[i] for i in range(1,8)]) ==
totalDosageCl)
```

1.5 Result and Comments:

Here, the initial dosages of the drugs in the base regimen does not matter. Because we can change them as we want and there is no cost. So, we can easily solve this optimization without computation. Firstly, all the four drugs in the base regiment must at least have their minimal values which are 20, 20, 10 and 20 which already makes up 70 cl. Then we want to give the remaining 30cl to the drug with the highest value between these four and that one is X_7 . The result is below as expected:

```
Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0   -6.7000000e+00   0.000000e+00   0.000000e+00      0s

Solved in 0 iterations and 0.01 seconds (0.00 work units)
Optimal objective -6.700000000e+00

Variable            X
-------------------------
         X1          20
         X3          20
         X4          10
         X7          50
```

2 Part b)

2.1 Decision Variables:

X_i:The amount of drug i used, $i \in \{1,..., 7\}$

Y_i:Whether or not the drug i used, $i \in \{1,..., 7\}$

I_i:The amount of increase in the dosage of drug i from the base regimen, $i \in \{1,..., 7\}$

D_i:The amount of decrease in the dosage of drug i from the base regimen, $i \in \{1,..., 7\}$

```
Y = {}
for i in range(1,8):
    Y[i] = m.addVar(vtype=GRB.BINARY, name="Y"+str(i))

I = {}
for i in range(1,8):
    I[i] = m.addVar(vtype=GRB.CONTINUOUS, name="I"+str(i))

D = {}
for i in range(1,8):
    D[i] = m.addVar(vtype=GRB.CONTINUOUS, name="D"+str(i))

X = {}
for i in range(1,8):
    X[i] = m.addVar(vtype=GRB.CONTINUOUS, name="X"+str(i))
```

2.2 Parameters:

P_i:Patient features, $i \in \{1,..., 9\}$

QOL_P_i:The coefficients of the P variables in the Q-score function, $i \in \{1,..., 9\}$

QOL_Y_i:The coefficients of the Y variables in the Q-score function, $i \in \{1,..., 7\}$

QOL_X_i:The coefficients of the X variables in the Q-score function, $i \in \{1,..., 7\}$

Min_i:The minimum amount of drug i that can be used, $i \in \{1,..., 7\}$

Max_i:The maximum amount of drug i that can be used, $i \in \{1,..., 7\}$

$baseDosage_i$:The amount of drug i used in the base regimen, $i \in \{1,..., 7\}$

$total$:The total size of an acceptable regimen
$Q\_Threshold$:The desired value of Q-score for the optimization
$fixedCosts_i$:The inflicted cost when the drug i is added or removed from the base regimen, $i \in \{1,..., 7\}$
$unitCosts_i$:The inflicted cost for each unit of drug i that is added or removed from the base regimen, $i \in \{1,..., 7\}$

These parameters are defined as follows in the code:

```
P = {1:1, 2:0, 3:1, 4:0, 5:0, 6:0, 7:0, 8:0, 9:1}
QOL_P_parameters = {1:-5, 2:-0.5, 3:-12, 4:-8, 5:-5, 6:-5, 7:-1, 8:-3,
9:-2}
QOL_Y_parameters = {1:-5, 2:-6, 3:-4, 4:-4, 5:-8, 6:-6, 7:-7}
QOL_X_parameters = {1:0.28, 2:0.30, 3:0.25, 4:0.17, 5:0.31, 6:0.246,
7:0.40}

minDosages = {1:20, 2:10, 3:20, 4:10, 5:10, 6:20, 7:20}
maxDosages = {1:80, 2:50, 3:100, 4:100, 5:70, 6:90, 7:50}
baseRegimenDosages = {1:20, 2:0, 3:30, 4:15, 5:0, 6:0, 7:35}
totalDosageCl = 275

Q_threshold = 35

fixedCosts = {1:25, 2:50, 3:10, 4:25, 5:20, 6:30, 7:40}
unitCosts = {1:1, 2:2, 3:1, 4:3, 5:2, 6:1, 7:1}
```

2.3 Objective:

We are trying to minimize the following cost function:

$$fixedCosts_1 * (1 - Y_1) + fixedCosts_2 * Y_2 + fixedCosts_3 * (1 - Y_3) + fixedCosts_4 * (1 - Y_4) + fixedCosts_5 * Y_5 + fixedCosts_6 * Y_6 + fixedCosts_7 * (1 - Y_7) + \sum_{i=1}^{7} unitCosts_i (I_i + D_i)$$

In the code, this is written as follows by introducing another kind of parameter:

```
isDrugInBaseRegimen = {1:True, 2:False, 3:True, 4:True, 5:False, 6:False,
7:True}

m.setObjective(
quicksum([(fixedCosts[i] * (1 - Y[i]) if isDrugInBaseRegimen[i] else
fixedCosts[i] * Y[i]) for i in range(1,8)])
+ quicksum([unitCosts[i] * (I[i] + D[i]) for i in range(1,8)])
, GRB.MINIMIZE)
```

2.4 Constraints:

$X_i = baseDosage_i + I_i - D_i,\ i \in \{1,..., 7\}$: Here, the decision variable X_i was actually not necessary, but we introduce it as a helper variable and force it to have the above value.

$X_i \geq minDosages_i * Y_i,\ i \in \{1,..., 7\}$: When Y_i is 1, this forces the amount of drug to be more than the minimum dosage specified.

$X_i \leq maxDosages_i * Y_i,\ i \in \{1,..., 7\}$: When Y_i is 1, this forces the amount of drug to be less than the maximum dosage specified. When Y_i is 0, together with the minimum constraints above, these force X_i to be 0.

```
helperXConstraints = {}
minDrugConstraint = {}
maxDrugConstraint = {}
for i in range(1,8):
    helperXConstraints[i] = m.addConstr(X[i] == baseRegimenDosages[i] +
I[i] - D[i])
    minDrugConstraint[i] = m.addConstr(X[i] >= minDosages[i] * Y[i])
    maxDrugConstraint[i] = m.addConstr(X[i] <= maxDosages[i] * Y[i])
```

$\sum_{i=1}^{7} X_i = total$: The sum of all the drug amounts must be equal to 275 centilitres (100 centilitres made the problem infeasible for this part) to be an acceptable regimen.

```
totalDosageConstraint = m.addConstr(quicksum([X[i] for i in range(1,8)])
== totalDosageCl)
```

$\sum_{i=1}^{9} QOL\_P_i * P_i + \sum_{i=1}^{7} QOL\_Y_i * Y_i + \sum_{i=1}^{9} QOL\_X_i * X_i \geq Q\_Threshold$: The Q-score, or the quality of the life of the patient, is larger than the given threshold. I have added a few new parameters to be able to write this constraint easier. In code,

```
QOFConstraint = m.addConstr(
quicksum([QOL_P_parameters[i] * P[i] for i in range(1,10)])
+ quicksum([QOL_Y_parameters[i] * Y[i] for i in range(1,8)])
+ quicksum([QOL_X_parameters[i] * X[i] for i in range(1,8)])
>= Q_threshold
)
```

2.5 Result and Comments:

As explained in the project questions email, there were two interpretations of fixed costs. I modelled the problem such that these fixed costs are incurred only when a drug is completely removed or added when it does not exist. Also, I was getting infeasible solutions with the default volume of 100cl, so I increased it up to 275cl. The results that I have obtained are:

```
Explored 1 nodes (8 simplex iterations) in 0.01 seconds (0.00 work units)
Thread count was 12 (of 12 available processors)

Solution count 1: 235

Optimal solution found (tolerance 1.00e-04)
Best objective 2.350000000000e+02, best bound 2.350000000000e+02, gap 0.0000%

  Variable         X
-------------------------
        Y1         1
        Y3         1
        Y4         1
```

Y7	1
I1	60
I3	70
I4	30
I7	15
X1	80
X3	100
X4	45
X7	50

First obverse that this model will never increase and decrease a drug at the same time because it is just an unnecessary cost that the model is already trying to minimize. We see that in the optimal result, the drug dosages are all increased.

3 Part c)

3.1 Decision Variables:

In addition to the decision variables from part b, add 4 new binary variables, A,B,C and E (the variable name D was already used). These are going to be used to model IF-THEN and EITHER-OR constraints later.

```
A = m.addVar(vtype=GRB.BINARY, name="A")
B = m.addVar(vtype=GRB.BINARY, name="B")
C = m.addVar(vtype=GRB.BINARY, name="C")
E = m.addVar(vtype=GRB.BINARY, name="E")
```

3.2 Parameters:

Again, the parameters from the previous model are used.

3.3 Objective:

Again, the objective function from the previous model is used with minimization.

3.4 Constraints:

All the constraints from the previous model are used with the following extra constraints.

- Melphalan and Oxaliplatin, when combined in wrong amounts, decrease the effect of the therapy. Hence, when combined, the total amount of these two chemicals should be less than 70 cl and greater than 50cl in any regimen:

This translates to $(Y_1 + Y_2 \geq 2) \Rightarrow (X_1 + X_2 \geq 50)$ and also $(Y_1 + Y_2 \geq 2) \Rightarrow (X_1 + X_2 \leq 70)$. After converting $Y_1 + Y_2 \geq 2$ into $Y_1 + Y_2 - 1 > 0$, we can convert these two IF-THEN constraints into EITHER-OR constraints. They become $(Y_1 + Y_2 - 1 \leq 0)\ OR\ (50 - X_1 - X_2 \leq 0)$ and $(Y_1 + Y_2 - 1 \leq 0)\ OR\ (X_1 + X_2 - 70 \leq 0)$ which are then modelled with the addition of the newly added binary decision variables. I directly give the constraint from the code, where M is large constant (for example, I have chosen 10000 as it was easily sufficient):

```
extraContstraint1_1 = m.addConstr(Y[1] + Y[2] - 1 <= M * A)
extraContstraint1_2 = m.addConstr(50 - X[1] - X[2] <= M * (1 - A))

extraContstraint2_1 = m.addConstr(Y[1] + Y[2] - 1 <= M * B)
extraContstraint2_2 = m.addConstr(X[1] + X[2] - 70 <= M * (1 - B))
```

- Either Epirubicin should be included in the regimen or the dosage of Decitabine should be less than 25cl.

This translates to $(Y_5 \geq 1)\ OR\ (X_3 \leq 25)$, which I convert to
$(1 - Y_5 \leq 0)\ OR\ (X_3 - 25 \leq 0)$ and write the constraints as:

```
extraContstraint3_1 = m.addConstr(1 - Y[5] <= M * C)
extraContstraint3_2 = m.addConstr(X[3] - 25 <= M * (1 - C))
```

- If both Pentostatin and Lomoustine are included in the regimen, then at least one of the Thiotepa and Epirubicin should also be chosen.

This translates to $(Y_4 + Y_6 \geq 2) \Rightarrow (Y_5 + Y_7 \geq 1)$, which I convert to
$(Y_4 + Y_6 - 1 > 0) \Rightarrow (1 - Y_5 - Y_7 \leq 0)$ and then again convert to
$(Y_4 + Y_6 - 1 \leq 0)\ OR\ (1 - Y_5 - Y_7 \leq 0)$, which is written as:

```
extraContstraint4_1 = m.addConstr(Y[4] + Y[6] - 1 <= M * E)
extraContstraint4_2 = m.addConstr(1 - Y[5] - Y[7] <= M * (1 - E))
```

3.5 Result and Comments:

In the optimal solution of the part b, we see that the only new constraint that is not satisfied is: “Either Epirubicin should be included in the regimen or the dosage of Decitabine should be less than 25cl”. In that optimal solution, the dosage of Decitabine was 100cl. With the new constraints added, now we see that optimal solution includes drug 5 as well in order to not reduce the amount of Decitabine below 25. Also, in order to create some space for drug 5, the optimal solution decreased the amount of drug 4.

With the new constraints added, the cost that we were trying to minimize increased from 235 to 282.5 as excepted. It could have stayed the same if the optimal solution of the previous part was already satisfying all these new constraints of part c.

```
Explored 1 nodes (25 simplex iterations) in 0.02 seconds (0.00 work units)
Thread count was 12 (of 12 available processors)

Solution count 3: 282.5 325 425.781

Optimal solution found (tolerance 1.00e-04)
Best objective 2.825000000000e+02, best bound 2.825000000000e+02, gap 0.0000%

   Variable            X
-------------------------
         Y1            1
         Y3            1
         Y4            1
         Y5            1
         Y7            1
         I1           60
         I3       34.375
         I5           70
```

I7	15
D4	4.375
X1	80
X3	64.375
X4	10.625
X5	70
X7	50
E	1