Assignment 1

University College Dublin

School of Electronic and Electrical Engineering

Optimisation Techniques- EEEN40580

Student: Aidan O'Sullivan

Student Number: 20209138

15th of October 2021

1.

Yes, I have read and understood UCD's policy on plagiarism and I hereby declare that this work is my own

Signed: Aidan O'Sullivan **Date:** 15/10/2021

My personal question variables:

$\alpha = 3$

$\beta = 8$

2. (A)

The following equations were calculated using my personal values of α = 3 and β = 8.

Objective Function

Min OF = $8x_1 + 10x_2$

Constraints:

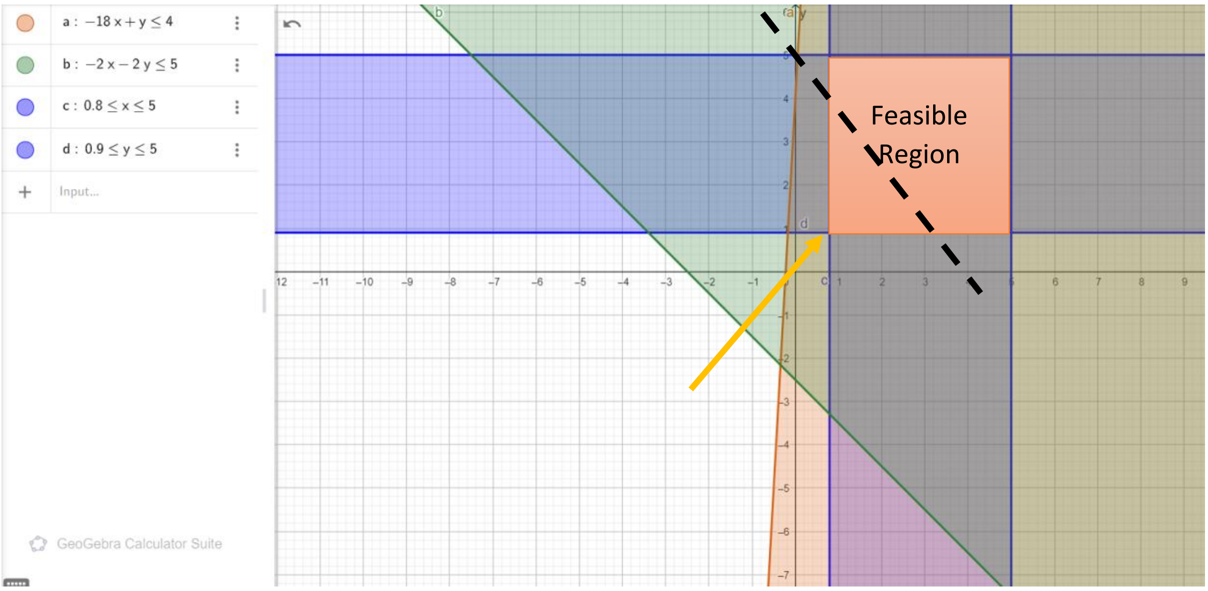
$x_2 \geq 18x_1 + 4$ → $-18x_1 + x_2 \leq 4$

$2x_2 \geq -2x_1 - 5$ → $-2x_1 - 2x_2 \leq 5$

$0.8 \leq x_1 \leq 5$

$0.9 \leq x_2 \leq 5$

These equations were then graphed on a free software called GeoGebra. The shaded area shows the region that is within each constraint. Where the shaded area for each constraint overlaps is known as the feasible region, shown in orange and marked on the graph below. The objective function is marked as the black dashed line in the image. As the objective function is being minimized this line was moved downwards until it reached the bottom corner of the feasible region, indicated with a yellow arrow. This gives the smallest possible values of x_1 and x_2 that will result in a minimization of the objective function.



The graphical method gave the following results:

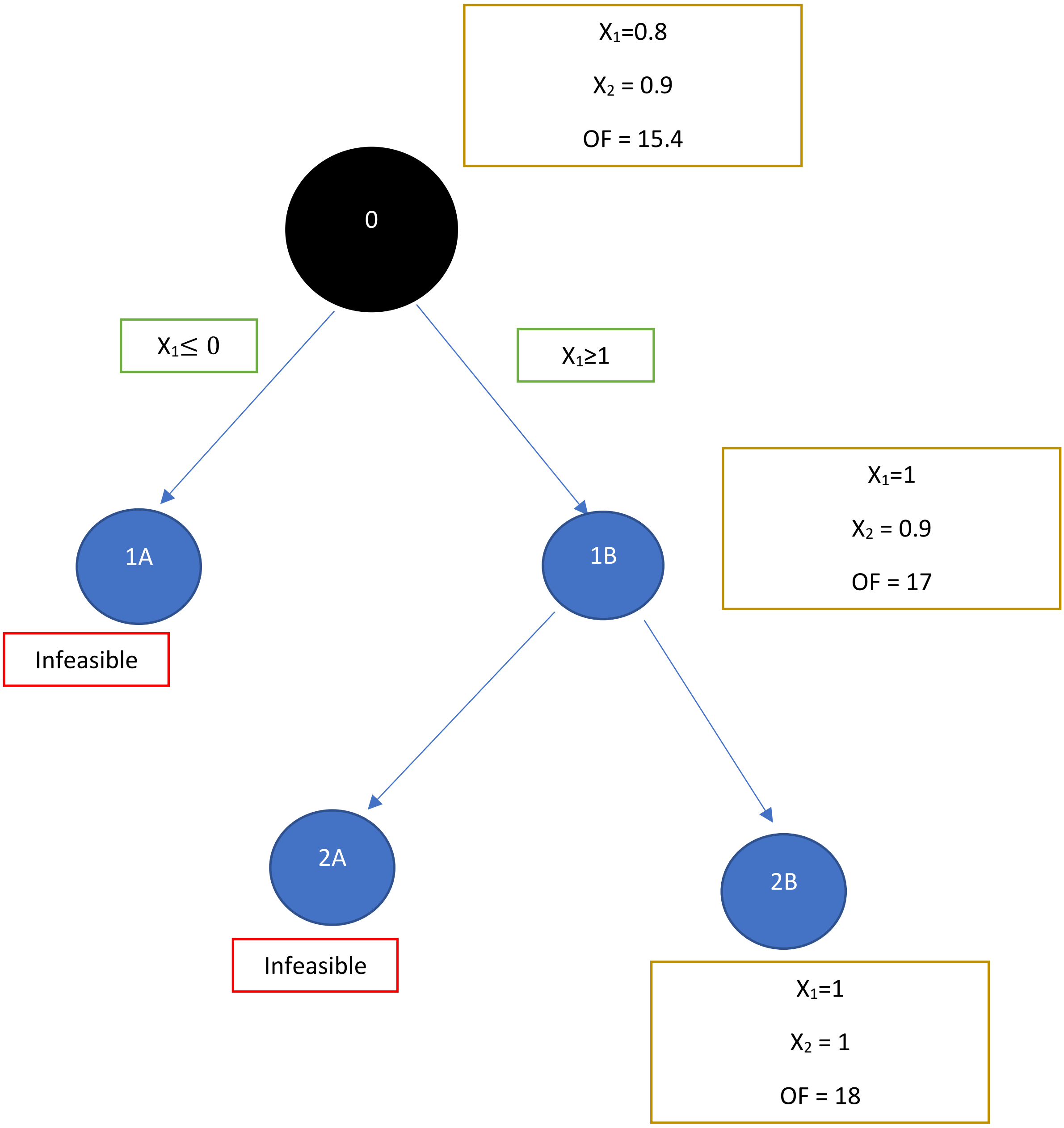
$X_1 = 0.8$

$X_2 = 0.9$

Min OF = $8(0.8) + 10(0.9) = 15.4$

2. (B)

The branch and bound method was used to minimize the objective function and to find an integer solution of x_1 and x_2.



Starting at point “0” on the branching diagram seen above, two solutions for x_1 and x_2 were found. These were computed using the program marked 2 (B) in the Jupyter script. Neither values were integers so the constraints for x_1 were relaxed in the two corresponding branches. The branch leading to 1A was infeasible, as $X_1 \leq 0$ was outside the possible range for x_1 of $0.8 \leq x_1 \leq 5$. The branch to 1B provided a feasible solution and the new x_1 constraints of $1 \leq x_1 \leq 5$ were changed in the python program. This gave a new set of values for x_1 and x_2. However, x_2 was still not a feasible solution, as it is not an integer, so more branching was required, this time relaxing the constraints of x_2.

The branch to 2A was again infeasible but the branch to 2B could be computed with constraints for x_1 and x_2 of:

$1 \leq x_1 \leq 5$

$1 \leq x_2 \leq 5$

The program computed an objective function of 18 with two integer values for x_1 and x_2. The objective function had detreated due to the added constraints, as it was larger than what it originally was when the objective is minimisation. This was still necessary as the question required integer values for x_1 and x_2.

3. (A)

<u>Overlapping Circles</u>

The distance between the centre of both circles is calculated as follows:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

If the combined distance of both radiuses is larger than distance d, then the circles are overlapping. U is a binary variable (either 0 or 1) and M is a sufficiently big positive number that ensures that this constraint gets enforced in cases where U = 1.

$$R_1 + R_2 > d + (1 - U)M$$

If the combined distance of both radiuses are less than or equal to distance d, then the circles are not overlapping. When U = 0 the following constraint gets enforced:

$$R_1 + R_2 \leq d + UM$$

Only one of these constraints are enforced and it is dependant on the binary value of U. This is how a conditional statement can be represented in linear programming.

3. (B)

The following conditions can be used to discern whether two rectangles overlap or not:

(1)

$w_1 + y_1 > y_2 + (1 - U_1)M$ overlapping

$w_1 + y_1 < y_2 + U_1M$ Not overlapping

(2)

$w_2 + y_2 > y_1 + (1 - U_2)M$ overlapping

$w_2 + y_2 < y_1 + U_2M$ Not overlapping

(3)

$L_1 + x_1 > x_2 + (1 - U_3)M$ overlapping

$L_1 + x_1 < x_2 + U_3 M$ Not overlapping

(4)

$L_2 + x_2 > x_1 + (1 - U_4)M$ overlapping

$L_1 + x_2 < x_1 + U_4 M$ Not overlapping

To overlap the following condition must be true:

$U_1 + U_2 + U_3 + U_4 = 4$

In this case all the overlapping conditions are enforced. This is similar to OR logic operation, where all conditions must be true for the two rectangles to be overlapping.

When the rectangles are not overlapping:

$U_1 + U_2 + U_3 + U_4 < 4$

In this case any or all overlapping conditions are not enforced, implying that the two rectangles do not overlap

3. (C)

The following conditions can be used to discern whether a rectangle and a circle overlap or not:

(1)

$R_1 + y_1 > y_2 + (1 - U_1)M$ overlapping

$R_1 + y_1 < y_2 + (1 - U_1)M$ Not overlapping

(2)

$w_2 + y_2 > y_1 + (1 - U_2)M$ overlapping

$w_2 + y_2 < y_1 + U_2M$ Not overlapping

(3)

$R_1 + x_1 > x_2 + (1 - U_3)M$ overlapping

$R_1 + x_1 < x_2 + U_3M$ Not overlapping

(4)

$L_2 + x_2 > x_1 + (1 - U_4)M$ overlapping

$L_2 + x_2 < x_1 + U_4M$ Not overlapping

To overlap the following condition must be true:

$U_1 + U_2 + U_3 + U_4 = 4$

In this case all the overlapping conditions are enforced. This is similar to OR logic operation, where all conditions must be true for the two rectangles to be overlapping.

When the rectangle and circle are not overlapping:

$U_1 + U_2 + U_3 + U_4 < 4$

In this case any or all overlapping conditions are not enforced, implying that the two rectangles do not overlap

4.

For question 4 the solution is shown in the Python script that was solved using Pyomo.

In my case the total budget of the company is:

$$20 + 2(3 + 1) + 0.5(8 + 1) = 32.5\ million\ Euro$$

In the Jupyter notebook a description of how each problem was solved is provided. Each model maximised the profit, within their respective constraints. The same budget applied to each part of Q4. A binary variable was used to indicate if each location was chosen. For parts C and D a new.dat file was created that split the locations into two sets, depending if they were located on the north or south side.

The constraint used for all parts to ensure that the budget is not exceeded:

$$\sum u_i * cost_i \leq 32.5\ million$$

Where the binary variable u will ensure that only selected locations will contribute to the total cost incurred by the company.

The objective function is to maximize the amount of profit:

$$\max OF \sum u_i * profit_i$$