BACHELOR'S DEGREE IN COMPUTER SCIENCE AND ENGINEERING
UNIVERSITY CARLOS III OF MADRID - UC3M

YEAR 2019/2020

LAB 1. MODELLING LINEAR PROGRAMMING TASKS

HEURISTICS AND OPTIMIZATION

Marina Torelli Postigo G88 100383479@alumnos.uc3m.es

1. Table of contents

2. Description and discussion of the models

Part 1

In this first part, the goal is to minimize the average waiting time in all queues of the museum. That is done by assigning a number of vending machines, turnstiles and employees to each of the three entrances to the museum.

It is asked that we model the problem as an Integer Linear Programming Task. In order to do that, we will define the following:

- An objective function
- A decision variable
- The constraints necessary

Decision variables:

We have 9 decision variables, one for each element of each entrance (3*3=9).

Therefore, the variables are:

- East – vending machine
- East – turnstile
- East – employee
- West – vending machine
- West – turnstile
- West – employee
- North – vending machine
- North – turnstile
- North – employee

In order to simplify formulas, we will create sets for them. They are the following:

M (main entrance): east

W: west

N: north

S (both secondary entrances): west north

Now we also need the parameters that have been given to us in the statement:
T_i : Average waiting time (minutes per visitor)
R_i : Reduction of average waiting time (minutes per visitor)
C_i : Cost (€ per day)

Constants:
n(number of elements) = 3 : vending machines, turnstiles and employees

Constraints:

1. The total investment for purchasing vending machines, turnstiles and hiring employees for all entrances should not be larger than €1000 per day.

$$\sum_{i=1}^{n} M_i C_i + \sum_{i=1}^{n} s_i C_i \ \leq 1000 \quad \forall s \in S$$

2. The main entrance should not require an expenditure greater than 10% over the expenditure of any of the secondary entrances.

$$p \sum_{i=1}^{n} s_i C_i - \sum_{i=1}^{n} M_i C_i \geq 0.1 \quad \forall s \in S$$

p stands for percentage and it is a constant with value p=1.1. It is used to calculate the 10% in the secondary entrances,

3. The sum of the number of vending machines and turnstiles of every secondary entrance shall be less than the sum of the number of vending machines and turnstiles of the main entrance.

$$\sum_{i=1}^{2} M_i - \sum_{i=1}^{2} s_i \ \geq 1 \quad \forall s \in S$$

4. The number of turnstiles in the north entrance shall be less than in the west entrance.

$$\sum_{i=2}^{2} W_i - \sum_{i=2}^{2} N_i \geq 1$$

5. There shall be at least two vending machines, two turnstiles and two employees in the main entrance.

$$M_i \ \geq 2 \quad \forall i \in [1, n]$$

6. There shall be at least one vending machine, one turnstile and one employee in each of the secondary entrances.

$$s_i \ \geq 1 \quad \forall s \in S$$

7. It is expected that the reduction in the average waiting time is larger in the main entrance than in any of the secondary entrances.

$$\sum_{i=1}^{n} M_i R_i - \sum_{i=1}^{n} s_i R_i \ \geq 1 \quad \forall s \in S$$

Objective function:

The objective function we have defined is in form of minimization. It provides the average waiting time of all three entrances. In order to calculate that.

$$\left[\sum_{i=1}^{n} T_i - \left(\sum_{j=1}^{n} E_j R_j\right)\right] / \, 3$$

Part 2

In this part we are faced with a new optimization problem. This time we need to assign 8 robots to 17 galleries in the museum. These robots are in charge of presenting each item in the gallery they are assigned to. We are also given several constraints that shall be fulfilled. The goal is to minimize the time required to introduce all items in each gallery of the museum.

We need to merge the solution to this problem to the solution of Part 1 of the assignment in order to provide the final solution and minimization problem. This means we need to solve the assignment so that the average waiting time of visitors to access the museum is minimized and so is the time required to introduce all objects in each gallery of the museum.

However, we have solved this second part of the assignment without considering the first part. It is only when we had both parts correctly solved separately that we merged the objective functions together to obtain the final solution. Thus, let us get into the process of solving this second part.

As in the previous one, we are asked to model the problem as an Integer Programming task.

We found that the best way of representing this problem was using a matrix, where the rows are the 8 different robots, and the columns are the 17 galleries. This is a representation of the matrix:

Galleries Robots	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
R1																	
R2																	
R3																	
R4																	
R5																	
R6																	
R7																	
R8																	

We have defined the decision variable as a binary number:

- If '1': there is a robot
- If '0': there is not a robot

This way, the matrix will be filled with 0s and 1s and it will not be difficult to use this variable in all our constraints.

Constraints:

1. It is not allowed to have more than one robot assigned to the same gallery, and each gallery shall have one robot assigned.

$$\sum_{i=1}^{m} X_{ij} = 1 \quad \forall j \in [1, n]$$

2. Each robot shall be assigned to at least two galleries and never more than three.

$$\sum_{j=1}^{n} X_{ij} \geq 2 \quad \forall i \in [1, m]$$

3. Robots R3, R5 and R6 cannot be assigned to any of the halls in the west side of the museum.

$$\sum_{i=1}^{10} X_{ij} = 0 \quad for\, j = 3, 5, 6$$

4. Likewise, robots R2 and R4 cannot be assigned to any of the galleries in the east side.

$$\sum_{i=11}^{17} X_{ij} = 0 \quad for\, j = 2, 4$$

5. Only robots assigned to either gallery A, B or both, can be assigned to the halls C, D or both.

$$\left[2 * \left(\sum_{j=1}^{2} X_{ij}\right)\right] - \left(\sum_{j=3}^{4} X_{ij}\right) \geq 0 \quad for\, \forall i \in [1,8]$$

6. Obviously, each gallery requires a specific amount of energy to introduce all objects in it, so that only robots that can support such amount shall be assigned to it.

$$M_i - \sum_{i=1}^{17} E_i X_{ij} N_j \geq 0 \quad for\, \forall i \in [1,8]$$

7. Finally, the galleries of the west side are larger and thus, it is required that presentations there take 10% longer than presentations in the east side.

3. Analysis of results

Now, we are going to cover some questions that were proposed in the assignment statement. They are the following:

1. What is the average waiting time of a visitor through the main entrance?

   This question can be easily solved using our MathProg implementation. The only thing we need to do is to change our Objective Function so that instead of calculating the average of waiting time for all three entrances, it only calculates the waiting time for the main entrance. We can then run the files and obtain a solution.

   Thus, the average waiting time of a visitor through the main entrance is 107 minutes.

2. What is the average waiting time of a visitor through the secondary entrances?

   The process for calculating these times is the same as for the last question. Thus, the average waiting time for the west entrance is 78 minutes and the average waiting time for the north entrance is 68 minutes.

3. How long will it take a visitor to visit all galleries?

   Again, we can calculate the answer to this question by just modifying our Objective Function on MathProg. We need to eliminate the part of it that involves the average waiting time for the entrances and just keep the formula where we calculate the minimization of the time it takes for all robots to introduce all items in the galleries they are assigned to. This time is also the time that will take a visitor to visit all galleries. When we run the files, we obtain that it will take the visitor 269 minutes to visit all galleries.

4. What robots will be in charge of the presentation in each gallery?

   The output file we obtain when running our MathProg files provides us with the values assigned to our decision variables. Since our 'assign' variable is a binary one, we just need to look in the output file for those variables' assignations (e.g. assign[r1,D]) that have '1' as their activity.

   In order to provide a more graphical and easier to understand representation of the assignation of the robots to the galleries, we have created the following table:

Galleries / Robots	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
R1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
R2	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
R3	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
R4	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0
R5	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
R6	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
R7	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R8	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0