Lab assignment 1: *Modeling Linear Programming Tasks*

Heuristics & Optimization 2019-2020 (Group 89)

Raul Olmedo Checa (100346073@alumnos.uc3m.es)

October 27, 2019

Contents

Chapter 1

Introduction

1.1 About this lab assignment

This report contains all the information related to the first lab assignment of the heuristics and optimization course. In particular, the task required for this assignment was to model and solve using two different tools an optimization problem regarding several aspects of a museum. The assignment and therefore this document is organized into two sections:

- Part 1: model and solve using a spreadsheet software (in particular the LP solver included in LibreOffice Calc) how the museum should organize its resources to reduce the waiting time in the queues.
- Part 2: model and solve, taking into account the previous part, how a set of robots should be set up to minimize the time spent explaining the galleries to the visitors of the museum (using GNU's Linear Programming Kit and MathProg as the modeling language).

1.2 Document contents

This document has been divided into four chapters trying to follow the structure proposed in the assignment's manual. Below there is a description of the content for each one of them:

- Chapter 1 (the current one): provides an overall description of the work carried out in the laboratory and the structure of the document.
- Chapter 2: the models of the two parts are presented to the reader alongside with the objective functions and the constraints of each part of the assignment. All design decisions carried out during the practice are stated in this section.
- Chapter 3: analyzes the results obtained in the previous chapter and answers the questions proposed in the assignment's guide.
- Chapter 4 explains the difficulties we have faced while completing this lab assignment, both from a technical and a conceptual standpoint.

Chapter 2

Problem Modelling

2.1 Part 1: Entrance waiting time

The museum wants to reduce the average waiting time in each of the entrances by acquiring a certain amount of resources.

- The entrances are classified according to its cardinal orientation, therefore we obtain three classes: *north*, *east* and *west* (note: from this point onwards, to simplify the formulas, we denote them as E_1, E_2 and E_3.
- The resources available to the museum are *vending machines*, *turnstiles* and *employees*. Once again, to simplify the formulae we write them as RS_1, RS_2 and RS_3.

2.1.1 Decision variables definition

The proposed model contains 9 decision variables which denote how many resources of each type (RS) should be placed into each entrance (E). The equation 2.2 presents them in mathematical language.

$$\forall i \in RS, j \in E; \vec{N} \in \mathbb{Z} : \quad N_{i,j} \rightleftharpoons \#\text{ resource } i \text{ on entrance } j; \tag{2.1}$$

where

RS = set of available resources

E = set of entrances

2.1.2 Objective function definition

The objective of this statement is to minimize the average waiting time in all entrances of the museum.

Since the approach we took to define our objective function was to maximize the time reduction in all entrances, in order to obtain the minimum waiting time we need to: substract the maximized value obtained in the objective function to the sum of the average waiting time in all of them, then we divide it by three to obtain the mean value of the total waiting time per entrance.
The formula 2.3 explains this in a formal way.

The objective function proposed is:

$$max\ z = \sum_{i \in E} \left(\sum_{j \in RS} \left(RE_j \vec{N}_{i,j} \right) \right) \tag{2.2}$$

where

RE = set defining the minutes reduced by each resource type j

$\vec{N}$ = vector containing the final number of resources per entrance i and per type j

The following formula obtains the optimal solution required by the museum:

$$\overline{average\ waiting\ time} = \frac{\sum_{i \in A}(A_i) - z(\vec{N})}{3} \tag{2.3}$$

where

A = set of the average waiting time in minutes per entrance, provided by the museum

z = objective function defined in 2.2

2.1.3 Constraints

Cost

The total cost of the resources placed in the three entrances must be lower or equal to 1000€.

$$\sum_{i \in E} \left(\sum_{j \in C} (C_j N_{j,i}) \right) \leq 1000 \tag{2.4}$$

where

C = set of costs for each resource j in an 8-hour day

Cost ratio

The main entrance (the eastern one) should not cost more than 10% of any of the secondary ones.

$$\sum_{j \in RS} (C_j N_{j,k}) \leq 1.1 \times \sum_{j \in RS} (C_j N_{j,i}) \quad \forall k \in E^m,\ i \in E^s \tag{2.5}$$

where

E^m = set containing the main entrance of the museum; $E^m \subseteq E$

E^s = set of secondary entrances (*north* and *west*); $E^s \subseteq E$

Sum of resources

The sum of *turnstiles* and *vending machines* in any secondary entrance shall be less than the sum of the mentioned resources in the main one. The plus one is added to the left side of the inequality to force the constraint to be *less than* instead of *less or equal than.*

$$\sum_{j \in RS^s} (N_{j,i}) + 1 \leq \sum_{j \in RS^s} (N_{j,k}) \quad \forall k \in E^m,\ i \in E^s \tag{2.6}$$

where

RS^s = set of secondary resources (*vending machines* and *turnstiles*); $RS^s \subseteq RS$

Turnstiles in secondary entrances

The number of turnstiles present in the *north* entrance must be lower than the ones placed on the *west* side. Once again the plus one is added to force the *less than* behaviour while using *less or equal than.*

$$N_{turnstile,north} + 1 \leq N_{turnstile,west} \tag{2.7}$$

Amount of resources in the main entrance

In the main entrance, at least two resources of each type (*vending machines*, *turnstiles* and *employees*) should be placed.

$$N_{i,j} \geq 2 \quad \forall i \in RS,\ j \in E^m \tag{2.8}$$

Amount of resources in the secondary entrances

Regarding the secondary accesses, at least one of each type of resource should be present.

$$N_{i,j} \geq 1 \quad \forall i \in RS,\ j \in E^s \tag{2.9}$$

Time reduction rate

The ratio of waiting time reduced by the placed resources in the main entrance should be bigger than the time reduction in any of the secondary entrances.

$$\sum_{j \in RS} (RE_j N_{j,k}) \geq 1 + \sum_{j \in RS} (RE_j N_{j,i}) \quad \forall k \in E^m,\ i \in E^s \tag{2.10}$$

Decision variables must be positive

$$N_{i,j} \geq 0 \quad \forall i \in RS, \quad j \in E \tag{2.11}$$

2.1.4 Complete LP task

$$\begin{gathered}
max\ z = \sum_{i \in E} \left(\sum_{j \in RS} (RE_j N_{i,j}) \right) \\
s.t. \\
\sum_{i \in E} \left(\sum_{j \in C} (C_j N_{j,i}) \right) \leq 1000 \\
\sum_{j \in RS} (C_j N_{j,k}) \leq 1.1 \times \sum_{j \in RS} (C_j N_{j,i}) \quad \forall k \in E^m,\ i \in E^s \\
\sum_{j \in RS^s} (N_{j,i}) + 1 \leq \sum_{j \in RS^s} (N_{j,k}) \quad \forall k \in E^m,\ i \in E^s \\
N_{turnstile,north} + 1 \leq N_{turnstile,west} \\
N_{i,j} \geq 2 \quad \forall i \in RS,\ j \in E^m \\
N_{i,j} \geq 1 \quad \forall i \in RS,\ j \in E^s \\
\sum_{j \in RS} (RE_j N_{j,k}) \geq 1 + \sum_{j \in RS} (RE_j N_{j,i}) \quad \forall k \in E^m,\ i \in E^s \\
N_{i,j} \geq 0 \quad \forall i \in RS, \quad j \in E
\end{gathered} \tag{2.12}$$

2.2 Part 2: Robots assignment

This time, the task requires the assignment of a set of robots to a set of galleries. The museum wants to minimize the time that these robots spend introducing the different objects in the galleries to the visitors, while also taking into account the first part of the problem, optimizing in this way the overall time that a visitor spends on the museum (both waiting in the queue and visiting it).

- There are eight robots $R1$ to $R8$ that have different properties (*energy*, *maximum energy* and *presentation time*).
- The museum is divided into seventeen galleries denoted A to Q, each one of them has a different amount of items to be presented by the robots.

2.2.1 Decision variables definition

This model has 145 decision variables, since the previous model is included, there are 9 variables that correspond to the first part.

136 binary variables have been defined to solve the robot assignment problem. The following formula explains the assignment in an explicit way.

$$X_{i,j} = \begin{cases} 1 & \text{if robot } i \text{ is assigned to gallery } j \\ 0 & \text{otherwise} \end{cases}, \quad i \in R, \quad j \in G \tag{2.13}$$

where

R: set of robots

G: set of galleries

2.2.2 Objective function definition

This time the objective is to minimize the time needed for the robots to show the different galleries based on the amount of time needed to present an item and the number of items in each one of them.

Once we reached this part, we realized that the objective function 2.2 of part 1 could have been done differently, in this formula the first part (which solves the previous problem) has been rewritten to include the mean average waiting time and transformed into a minimization formulae.
The second part of the formula is exclusive for solving the assignment problem proposed on this part: for each robot, the time needed to present all items in the gallery is computed (taking into account if the robot is assigned to the gallery or not).

Therefore, the objective function proposed is:

$$\min z = \frac{\sum_{i \in E} \left(T_i - \sum_{j \in RS} \left(RE_j N_{j,i} \right) \right)}{3} + \sum_{i \in R} \left(\sum_{j \in G} \left(P_i I_j X_{i,j} \right) \right) \tag{2.14}$$

where

T: average waiting time per entrance i

P: presentation time required by each robot i

I: number of items per gallery j

2.2.3 Constraints

Robots assigned to a gallery

Every gallery cannot have more than one robot assigned to it and every one of them must have one (in other words, each gallery must have one and only one robot assigned). To model this restriction, it had to be split up into two different constraints to mimic the *equal than* relation.

$$\sum_{i \in R} X_{i,j} \leq 1 \quad \forall j \in G \tag{2.15}$$

$$\sum_{i \in R} X_{i,j} \geq 1 \quad \forall j \in G \tag{2.16}$$

where

R: set of robots

G: set of galleries

Galleries assigned to a robot

Each robot should be assigned in at least two galleries, but no more than three. Once again, we split the restriction into two parts.

$$\sum_{j\in G} X_{i,j} \geq 2 \quad \forall i \in R \tag{2.17}$$

$$\sum_{j\in G} X_{i,j} \leq 3 \quad \forall i \in R \tag{2.18}$$

Robot placement (West)

There are three robots ($R3$, $R5$ and $R6$)that cannot be assigned into the western galleries, so the sum of those decision variables must be zero.

$$\sum_{j\in G^w} X_{i,j} \leq 0 \quad \forall i \in R^e \tag{2.19}$$

where

G^w = galleries placed in the western part of the building; $G^w \subseteq G$

R^e = robots forbidden in the western part of the building (allowed only on the east); $R^e \subseteq R$

Robot placement (East)

There are two robots ($R2$ and $R4$) that cannot be assigned into the east side galleries, so the sum of those decision variables must be zero.

$$\sum_{j\in G^e} X_{i,j} \leq 0 \quad \forall i \in R^w \tag{2.20}$$

where

G^e = galleries placed in the eastern part of the building; $G^e \subseteq G$

R^w = robots forbidden in the east part of the building (allowed only on the west); $R^w \subseteq R$

Special-robots placement

If a robot is placed on gallery A and/or gallery B, that robot is the only one allowed to be placed on gallery C and/or gallery B. **e.g.** if a robot is placed on A, it could be placed in either C, D or both, but if a robot is placed on gallery E we cannot assign it to gallery D.

$$\sum_{j\in G^{A,B}} X_{i,j} \geq \sum_{j\in G^{C,D}} X_{i,j} \quad \forall i \in R \tag{2.21}$$

where

$G^{A,B}$ = galleries A and B; $G^{A,B} \subseteq G$

$R^{C,D}$ = galleries C and D; $G^{C,D} \subseteq G$

Maximum energy

Each robot has a maximum energy available that cannot be surpassed, it is calculated by multiplying the energy required to present an item by the number of items of the gallery.

$$\sum_{j\in G} X_{i,j} I_j EN_i \leq M_i \quad \forall i \in R \tag{2.22}$$

where

EN = energy required to present an item by the robot i

M = maximum energy allowed for the robot i

Gallery size

The galleries placed on the West are bigger than the ones in the East, so the presentation time should be 10% longer than in the latter.

$$\sum_{i \in R^w} X_{i,j} I_j T_i \geq 1.1 \times \sum_{i \in R^e} X_{i,j} I_j T_i \quad \forall i \in G^w \tag{2.23}$$

Decision variables must be 0 or 1 (binary variables)

$$X_{i,j} \in \{0,1\} \quad \forall i \in R \quad \forall j \in G \tag{2.24}$$

2.2.4 Complete LP task

$$\begin{gathered}
\min z = \frac{\sum_{i \in E} \left(T_i - \sum_{j \in RS} (RE_j N_{j,i})\right)}{3} + \sum_{i \in R} \left(\sum_{j \in G} (P_i I_j X_{i,j}) \right) \\
\sum_{i \in R} X_{i,j} \leq 1 \quad \forall j \in G \\
\sum_{i \in R} X_{i,j} \geq 1 \quad \forall j \in G \\
\sum_{j \in G} X_{i,j} \geq 2 \quad \forall i \in R \\
\sum_{j \in G} X_{i,j} \leq 3 \quad \forall i \in R \\
\sum_{j \in G^w} X_{i,j} \leq 0 \quad \forall i \in R^e \\
\sum_{j \in G^e} X_{i,j} \leq 0 \quad \forall i \in R^w \\
\sum_{j \in G^{A,B}} X_{i,j} \geq \sum_{j \in G^{C,D}} X_{i,j} \quad \forall i \in R \\
\sum_{j \in G} X_{i,j} I_j EN_i \leq M_i \quad \forall i \in R \\
\sum_{i \in R^w} X_{i,j} I_j T_i \geq 1.1 \times \sum_{i \in R^e} X_{i,j} I_j T_i \quad \forall i \in G^w \\
X_{i,j} \in \{0,1\} \quad \forall i \in R \quad \forall j \in G
\end{gathered} \tag{2.25}$$

Chapter 3

Analysis of results

3.1 Interpretation of decision variables

3.1.1 Part 1

The values for the decision variables and the result of the objective function obtained after solving the problem with LibreOffice Calc are presented here.

Table 3.1: Decision variables values for part 1

	RE_1	RE_2	RE_3
E_1	3	3	2
E_2	2	2	3
E_3	1	4	2

$$z(\vec{N}) = 67 \tag{3.1}$$

The value of the objective function means that the maximum reduction that the museum can perform is 67 minutes. To obtain the minimum waiting time in all queues we still need to perform another operation. Substituting the values obtained in equation 2.3:

$$T_{min} = \frac{320 - 67}{3} = 84.\overline{3}\ min \tag{3.2}$$

Therefore, we can see that the minimum average time in minutes that a person must wait in the queue while satisfying all restrictions is $84.\overline{3}\ min$.

3.1.2 Part 2

This table shows the values obtained in the second part of the assignment, which solve the assignment problem presented by the museum.

Table 3.2: Decision variables values for part 2

	RE_1	RE_2	RE_3	$R1$	$R2$	$R3$	$R4$	$R5$	$R6$	$R7$	$R8$
E_1	3	3	2								
E_2	2	2	3								
E_3	1	4	2								
G_1				1	0	0	0	0	0	0	0
G_2				0	0	0	0	0	0	1	0
G_3				1	0	0	0	0	0	0	0
G_4				0	0	0	0	0	0	1	0
G_5				0	0	0	1	0	0	0	0
G_6				0	1	0	0	0	0	0	0
G_7				0	1	0	0	0	0	0	0
G_8				0	0	0	0	0	0	0	1
G_9				0	0	0	1	0	0	0	0
G_{10}				0	0	0	1	0	0	0	0
G_{11}				0	0	0	0	1	0	0	0
G_{12}				0	0	0	0	0	1	0	0
G_{13}				0	0	0	0	1	0	0	0
G_{14}				0	0	1	0	0	0	0	0
G_{15}				0	0	1	0	0	0	0	0
G_{16}				0	0	0	0	0	1	0	0
G_{17}				0	0	0	0	0	0	0	1

$$\begin{gathered} R1:\{A,C\} \quad R2:\{F,G\} \quad R3:\{N,O\} \quad R4:\{E,I,J\} \\ R5:\{K,M\} \quad R6:\{L,P\} \quad R7:\{B,D\} \quad R8:\{H,Q\} \end{gathered} \tag{3.3}$$

$$z(\vec{X}) = 246.\overline{6} \tag{3.4}$$

The value of the objective function means that the minimum time that a visitor will spend on the museum (including the average waiting time in the queue) is $246.\overline{6}minutes$. This value contains the solution to the previous part ($84.\overline{3}\ min$) but including the time needed by the robots to introduce all items in all galleries of the museum.

The set of galleries assigned to each robot has been included under the table to make the assignments clear to the reader, is the same information included in 3.2.

3.2 Checking the correctness

3.2.1 Part 1

To check that the solution is correct, the values obtained for the decision variables are used as an input for the constraints of the LP problem as follows:

Cost This restriction involves constraint 1:

$$\begin{pmatrix} N_{RS_1,E_1} \\ \vdots \\ N_{RS_3,E_1} \end{pmatrix} \begin{pmatrix} 32 & 40 & 64 \end{pmatrix} + \begin{pmatrix} N_{RS_1,E_2} \\ \vdots \\ N_{RS_3,E_2} \end{pmatrix} \begin{pmatrix} 32 & 40 & 64 \end{pmatrix} + \begin{pmatrix} N_{RS_1,E_3} \\ \vdots \\ N_{RS_3,E_3} \end{pmatrix} \begin{pmatrix} 32 & 40 & 64 \end{pmatrix} = 1000 \leq 1000 \tag{3.5}$$

Cost ratio This restriction involves constraints 2 to 3:

$$\begin{pmatrix} N_{RS_1,E_2} \\ \vdots \\ N_{RS_3,E_2} \end{pmatrix} \begin{pmatrix} 32 & 40 & 64 \end{pmatrix} \leq 1.1 \times \begin{pmatrix} N_{RS_1,E_1} & N_{RS_1,E_3} \\ \vdots & \vdots \\ N_{RS_3,E_1} & N_{RS_3,E_3} \end{pmatrix} \begin{pmatrix} 32 & 40 & 64 \end{pmatrix} = \begin{pmatrix} 344 \\ 344 \end{pmatrix} \leq \begin{pmatrix} 369.6 \\ 352 \end{pmatrix} \tag{3.6}$$

Sum of resources This restriction involves constraints 4 to 5:

$$\begin{pmatrix} N_{RS_1,E_1} \\ N_{RS_2,E_1} \end{pmatrix} + \begin{pmatrix} N_{RS_1,E_3} \\ N_{RS_2,E_3} \end{pmatrix} + \begin{pmatrix} 1_{2\times1} \end{pmatrix} \leq \begin{pmatrix} N_{RS_1,E_2} \\ N_{RS_2,E_2} \end{pmatrix} = \begin{pmatrix} 5 \\ 6 \end{pmatrix} \leq \begin{pmatrix} 6 \\ 6 \end{pmatrix} \tag{3.7}$$

Number of turnstiles in secondary entrances This restriction involves constraint 6:

$$N_{RS_2,E_1} + 1 \leq N_{RS_2,E_3} = 3 \leq 3 \tag{3.8}$$

Amount of resources in main entrance This restriction involves constraint 7 to 9:

$$\begin{pmatrix} N_{RS_1,E_2} \\ \vdots N_{RS_3,E_2} \end{pmatrix} \geq \begin{pmatrix} 2_{3\times1} \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \\ 2 \end{pmatrix} \geq \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix} \tag{3.9}$$

Amount of resources in secondary entrances This restriction involves constraint 10 to 15:

$$\begin{pmatrix} N_{RS_1,E_1} & N_{RS_1,E_3} \\ \vdots & \vdots \\ N_{RS_3,E_1} & N_{RS_3,E_3} \end{pmatrix} \geq \begin{pmatrix} 1_{3\times2} \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 2 & 4 \\ 3 & 2 \end{pmatrix} \geq \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix} \tag{3.10}$$

Time reduction This restriction involves constraints 16 to 17:

$$\begin{pmatrix} N_{RS_1,E_2} \\ \vdots \\ N_{RS_3,E_2} \end{pmatrix} \begin{pmatrix} 2 & 3 & 4 \end{pmatrix} \geq \begin{pmatrix} 1_{3\times1} \end{pmatrix} + \begin{pmatrix} N_{RS_1,E_1} & N_{RS_1,E_3} \\ \vdots & \vdots \\ N_{RS_3,E_1} & N_{RS_3,E_3} \end{pmatrix} \begin{pmatrix} 2 & 3 & 4 \end{pmatrix} = \begin{pmatrix} 23 \\ 23 \end{pmatrix} \leq \begin{pmatrix} 23 \\ 23 \end{pmatrix} \tag{3.11}$$

As shown in the previous demonstrations, all constraints are satisfied so that the solution is correct.

3.2.2 Part 2

This problem contains the solution and the constraints of the first part, since the values are the same we do not need to perform the demonstrations made before.

Since the remaining part is of binary type, the demonstration can be made almost instantly taking into account the solution set in 3.3:

Robots assigned to a gallery This restriction involves constraints 18 to 51. It can be checked by looking at the solution set, no gallery is in two different sets at the same time, meaning that every one of them has one and only one robot assigned.

Galleries assigned to a robot This restriction involves constraints 52 to 67. This constraint can be checked at a first glance too, no set has a cardinality bigger than three nor smaller than two.

Robot placement (West) This restriction involves constraints 68 to 70:

$$\begin{pmatrix} X_{R3,G_1} & \cdots & X_{R3,G_{10}} \\ X_{R5,G_1} & \cdots & X_{R5,G_{10}} \\ X_{R6,G_1} & \cdots & X_{R6,G_{10}} \end{pmatrix} \begin{pmatrix} 1_{3\times10} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \leq \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{3.12}$$

Robot placement (East) This restriction involves constraints 71 to 72:

$$\begin{pmatrix} X_{R2,G_{11}} & \cdots & X_{R3,G_{17}} \\ X_{R4,G_{11}} & \cdots & X_{R5,G_{17}} \end{pmatrix} \begin{pmatrix} 1_{2\times6} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \leq \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{3.13}$$

Special-robots placement This restriction involves constraints 73 to 80:

$$\begin{pmatrix} X_{R1,G_1} & X_{R1,G_2} \\ \vdots & \vdots \\ X_{R8,G_1} & X_{R8,G_2} \end{pmatrix} \left(1_{8\times 2}\right) \geq \begin{pmatrix} X_{R1,G_3} & X_{R1,G_4} \\ \vdots & \vdots \\ X_{R8,G_3} & X_{R8,G_4} \end{pmatrix} \left(1_{8\times 2}\right) = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \geq \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \tag{3.14}$$

Maximum energy This restriction involves constraints 81 to 88:

$$\begin{pmatrix} X_{R1,G_1} & \cdots & X_{R1,G_{17}} \\ \vdots & & \vdots \\ X_{R7,G_1} & \cdots & X_{R1,G_{17}} \end{pmatrix} \begin{pmatrix} 5 \\ 5 \\ 5 \\ 6 \\ 6 \\ 4 \\ 4 \\ 4 \\ 7 \\ 7 \\ 3 \\ 3 \\ 3 \\ 2 \\ 2 \\ 2 \\ 2 \end{pmatrix} \begin{pmatrix} 7 & 5 & 3 & 1 & 2 & 4 & 4 & 5 \end{pmatrix} \left(1_{8\times 1}\right) \leq \begin{pmatrix} 100 \\ 90 \\ 95 \\ 40 \\ 45 \\ 75 \\ 85 \\ 60 \end{pmatrix} = \begin{pmatrix} 70 \\ 40 \\ 12 \\ 20 \\ 12 \\ 20 \\ 44 \\ 30 \end{pmatrix} \leq \begin{pmatrix} 100 \\ 90 \\ 95 \\ 40 \\ 45 \\ 75 \\ 85 \\ 60 \end{pmatrix} \tag{3.15}$$

Gallery size This restriction involves constraints 89 to 98.

Due to the number of constraints, the formalization of the checking has not been included, although the final results are below as proof of correctness.

$$212 \geq 1.1 \times 57 = 212 \geq 62.27 \tag{3.16}$$

As shown in the previous demonstrations, all constraints are satisfied so that the solution is, at least, correct (although there is no way to assure that it's the optimal by looking at them).

3.3 Most relevant constraints

Since the relevance in this case means that the more constraints produced, the more restricted the feasible region becomes, the set of constraints related to cost and time reduction are the most relevant ones. In the second part, the group that yields the most number of constraints is the one related to the robot placement.

3.4 Problem complexity

3.4.1 Part 1

For this part we defined 9 decision variables and 17 constraints. The complexity will increase if we add more entrances or resources to use. This is caused by the additional constraints that would created in the LP task.

The number of decision variables $m \times n$ comes from the m entrances and n resources. The number of constraints will come from m_s secondary entrances and n entrances, the following formula formalizes the number: $3 \times m_s + (m_s \times n) + n + 2$.

3.4.2 Part 2

In this part the number of binary decision variables is 136 and the number of constraints 81. In this case, the problem's complexity increases if we add more galleries or robots, but if we change the amount of items per gallery for example, it will not increase it (since the first problem is embedded in this one, by changing the variables mentioned in part 1 it will also increase its complexity). Once again, this depends on the number of constraints generated by those additions.

The number of decision variables comes from the m entrances, n resources, o galleries and p robots, it is $m \times n + o \times p$.

3.5 Specific questions

The average waiting time for a visitor depending on the entrance is:

78 min for the *north* entrance

107 min for the *east* entrance

68 min for the *west* entrance

The time required for a visitor to go through all galleries and all items is 269 min.
The robots in charge of each gallery have already been included in the formula 3.3.

3.6 Pros and cons of LibreOffice and GLPK

The main advantages we found in LibreOffice Calc are the visualization of the computed constraint values (for a fast visual check) and the option to watch the objective function and constraints change in real time, which helps with the understanding of the problem. However, when the problem includes a large constraint set or a high amount of decision variables it can become tedious to model and solve in the spreadsheet. Another minor drawback we found was the slow speed of the program itself while running on macOS, the scrolling seemed laggy and pretty slow.

MathProg/GLPK seems like a more natural way for computer scientists to solve and model LP problems, since the modularity you can achieve allows some interesting approaches by reusing models and changing data files. Although we could not use all GLPK's directives available, the way you can change and scale the problem is pretty huge compared with LibreOffice. Another minor drawback was MathProg's syntax coloring, we did not get our hands on the correct highlighter until the final stages of the coding section. Overall, once you get your mind used to working with sets and summations, there is no real challenge with it, although we would love more verbose syntax errors and more meaningful outputs.

Chapter 4

Conclusions

4.1 Difficulties with the assignment

With the first part, there were not real problems, we could learn the basic linear programming concepts without leaving a, sort of, known environment as the spreadsheet software is. However, the fact that once you obtain a solution you cannot check if it is the optimal one (you can check its correctness but the grade of the assignment depends on the model and the computation of the optimal solution) gave us some mixed feelings as we discovered a couple of times that our solution was not the optimal and therefore correct one.

When we started with MathProg's section, the situation changed, because the form seemed familiar (it looks quite like a regular programming language, you execute it and get syntax errors, etc) but the concept was not even close to what we knew. The introduction of sets, parameters and variables in the way that GLPK/MathProg handles them was a new thing to us, so we required some time to adapt our minds to the way they work. Once you get your brain to think in this way, the real problem remains in the modelling part, as translating the LP task from a mathematical standpoint to MathProg's syntax is trivial (once you already completed the assignment, of course). Once again, the syntax errors obtained in the output of GLPK's execution were unmeaningful most of the times and hard to fix, it might be due to the fact that we are used to program in more common languages with a huge development behind and a maybe "long" trajectory.

As a final thought, this report took longer than expected. It was our first time working with LaTeX and we found it pretty useful and surprisingly helpful with certain tasks (as creating formulas, indexes, tables of content, formulae referencing ...) but pretty tedious to write in some cases (creating tables and spacing them seems like a nightmare to me).