Lab assignment 1: *Modeling Linear Programming Tasks*
Heuristics and Optimization
Computer Science, 2017–2018

1 Goal

The goal of this assignment is to learn to model linear programming (LP) and mixed integer programming tasks, and to solve them with two different tools: The LibreOffice spreadsheet and a modeling language, MathProg.

2 Problem statement

An important airline with premises in Madrid has contacted the students of *Heuristics and Optimization* of the UC3M. After a number of conversations, you and your partner have been selected to solve different problems using Linear Programming.

2.1 Part 1: Basic Model in LibreOffice Calc

The airline has introduced us to an important problem they have when storing the luggage in the upper compartment of the cabin. Passengers usually take much more space than necessary for putting there their belongings. Thus, some bags do not fit and they have to be moved to the hold which results in an unnecesary waste of time and money. Therefore, it has been decided to automate the arrangement of bags and suitcases in the upper compartments so that the movement to the hold is minimized.

To this end, the luggage is categorized in three different groups: M1, M2 and M3, shown in Table 1.

Category	Number	Weight (kg)	Size (cm)	Cost (€)
M1	22	7	30×20×10	10
M2	18	8	40×20×10	20
M3	11	10	50×30×20	30

Table 1: Categories of luggage

The second column of Table 1 shows the number of bags of each category that have to be loaded into the plane. The third and fourth columns show the weight and dimensions (*height*×*length*×*width*) of each suitcase within each category. The last column shows the cost per bag in case it is moved to the hold.

In addition, the plane is equipped with 6 different compartments to store all the luggage, which are shown in Figure 1. Each compartment can hold up to the size and volume shown in Table 2. Each compartment can accomodate bags of different categories if necessary.

Compartment	Weight (kg)	Volume (m^3)	Compartment	Weight (kg)	Volume (m^3)
C1	50	0.1	C4	50	0.1
C2	60	0.15	C5	60	0.15
C3	40	0.07	C6	40	0.07

Table 2: Maximum allowed weight and volume of each compartment

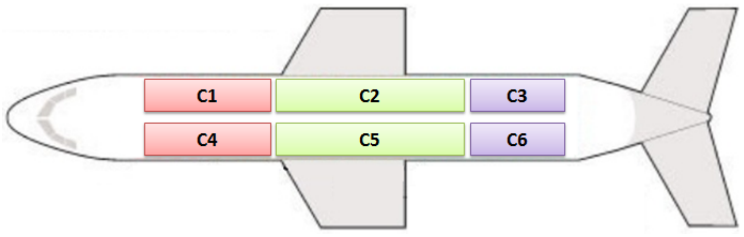

Figure 1: Arrangment of compartments in the plane

Finally, the company has reported that the gravity center of the plane should be shifted forward to improve stability and recovery in case of stall. To this end, the weight in compartments C1 and C4 should be at least 10% larger than the weight of all luggage in compartments C3 and C6.

The goal is to determine the number of suitcases of each category that shall be stored in each compartment of the plane, so that the waste for moving luggage to the hold is minimized:

1. Model the problem as a Linear Integer Programming task.
2. Implement the model in a spreadsheet (LibreOffice Calc).

2.2 Part 2: Advanced modeling with GLPK

The airline is fully satisfied with the results obtained in the first part, so that they decided to hire you again to solve another problem: the assignment of crew members (pilots and flight attendants) to various flights that will take place today. Table 3 shows the origin, destination, and the departure and arrival times of each flight.

Flight	Route	Departure	Arrival
V1	Madrid-Valencia	8:30	9:45
V2	Valencia-Madrid	10:30	12:00
V3	Madrid-Valencia	13:00	14:45
V4	Valencia-Madrid	15:00	16:05
V5	Madrid-Valencia	17:45	19:00
V6	Valencia-Madrid	19:50	21:00

Table 3: Flight information

The first flight, V1, is the one considered in the first part of this lab assignment, so that you are requested to still take into account the waste for moving luggage into the hold. All the other flights are cargo flights with no passengers on board, so that it is not required to load any luggage into the plane.

There are three pilots and three flight attendants. Tables 4 and 5 show the money that each pilot and flight attendant want to earn per hour.

Pilot	V1	V2	V3	V4	V5	V6
P1 (€/h)	20	17	25	34	31	22
P2 (€/h)	28	15	23	35	37	21
P3 (€/h)	27	16	24	31	35	29

Table 4: Earnings per flight hour for pilots in each different flight

Regarding the assignment of crew members to flights, the company set up a number of constraints. Clearly, each flight shall have assigned at least one pilot, and at least one flight attendant. Besides, the airline has recently signed an agreement with the crew members which guarantees that the number of flight hours of flight attendants will be larger than the number of flight hours of pilots. Each pilot has to agree with the airline the minimum duration of a break between consecutive flights, i.e., the minimum duration between the arrival time

Attendant	V1	V2	V3	V4	V5	V6
A1 (€/h)	17	16	14	14	11	12
A2 (€/h)	16	14	12	15	17	11
A3 (€/h)	15	15	14	11	15	19

Table 5: Earnings per flight hour for pilots in each different flight

of one flight, and the departure time of the next one. After a number of discussions, it has been decided that P1 will have at least a 60 minutes break; P2 will be allowed 25 minutes at least, and P3 will be given no less than 30. Finally, no crew member can be assigned to a flight if she/he is not available in the same airport from which the flight departs, and all crew members are initially located in Madrid —so that none can be immediately assigned to a flight departing from Valencia.

The goal is to minimize all wastes of the airline considering the move of luggage into the hold for V1, and the assignment of crew members to all flights. Make sure to consider all constraints.

1. Model the new problem as a MIP task.
2. Implement the new model in MathProg.

2.3 Part 3: Analysis of results

All results obtained in the previous parts have to be analyzed. The solutions obtained have to be properly described (verifying that all contraints are verified). It is also requested to check what constraints are more relevant.

Analyze the problem complexity: how many variables and constraints have you defined? Modify the problem, varying the parameters and adding/removing suitcases, compartments, flights, crew members, and explain if these changes affect the difficulty for solving the problem.

Also, address the following question: what is the number of pilots and flight attendants assigned to each flight? Why does it happen that more than one pilot and flight attendant are assigned to the same flight even if that results in more costly operations?

Likewise, discuss the pros and cons of using the proposed tools: LibreOffice and GLPK.

2.4 Optional part: Dynamic Programming

Given the success of this project, the airline contacted you and your partner again to solve a third problem. They are considering to open a new commercial route visiting Vigo, San Sebastián, Sevilla, Córdoba and Barcelona. They want to use just one plane, since that is enough to move the expected cargo among these cities. The flight should start in Madrid, it will visit all cities just once and will finally return to Madrid. The airline has requested us to use *Dynamic Programming* to decide the itinerary with the minimum total length. The program shall receive as an argument the location of a file that depicts the distance between cities. An example is shown below:

```
     M    V    Ss   Sv   C    B
M    0    602  450  434  394  624
V    602  0    755  741  796  1152
Ss   450  755  0    916  841  573
Sv   434  741  916  0    140  998
C    394  796  841  140  0    865
B    624  998  573  998  865  0
```

To test your solution, it is recommend to randomly generate files with a different number of cities. Take into account that the distance from one city to itself is obviously 0 and thus, the main diagonal of the matrix shall be null. You can use any programming language of your choice, along with a bash script that starts it called `ruta.sh`. This script should acknowledge the file with the distances between cities. As a result, two lines should be printed out on the console: the first line shall show the total length of the solution, and the second one should show the itinerary. An example is shown below:

```
Distancia total:    3338
Ruta:   M,Sv,C,V,Ss,B,M
```

Document all the recurrence relationships used in the memory, and also other additional solutions that might enhance your implementation. You should also address the following questions:

- If it is requested to visit 100 cities, what is the number of combinations that should be considered by a brute-force search approach? What is the number of combinations considered by your solution?
- Is it better for your algorithm if the cities are sorted according to some criteria? Yes, no? If so, which?

3 Requirements for the Report

The report must be delivered in PDF format and it should consist of a maximum of 15 pages, including the cover, table of contents and back cover. It should contain, at least:

1. Brief introduction explaining the contents of the document.
2. Description of the models, discussing the decisions carried out.
3. Analysis of results.
4. Conclusions.

The report **should not include source code** in any case.

4 Grading

This lab assignment will be graded over 10 points. Nevertheless, it is possible to get an additional point by submitting the extra part—hence, scoring for a maximum of 11 points.

To assure that the assignment is graded you must do at least the first part and the report.

Points are distributed as follows:

1. Part 1 (3 points)
    - Problem Modeling (1 point)
    - Model Implementation (2 points)
2. Part 2 (5 points)
    - Problem Modeling (3 points)
    - Model Implementation (2 points)
3. Part 3 (2 points)
4. Optional part: Dynamic Programming (1 point)

When grading the model proposed, a correct model will make half of the points. To get all points, the model must:

- Be correctly formalized in the report.
- Be simple and concise.
- Be properly explained (it should remain clear what is the reason for each variable/constraint).
- Justify in the report all the design decisions taken.

When grading the model implementation, a correct model will make half of the points. To get all points, the implementation must:

- Make use (in the implementation) of the features that the tools provide to facilitate that implementing/updating the model is as easy as possible (concretely, use `sumproduct` in the case of the spreadsheet or use *sets* in MathProg).
- Keep the code (spreadsheet or Mathprog files) correctly organized and commented. The names should be descriptive. You should also add comments in those cases that are needed to improve readability.

When grading the results analysis, it will be positively valued to include in the report personal conclusions about the assignment difficulty and about what you learnt while carrying it out.

Important: The models in the spreadsheet and GLPK must be correct. That is, *they must run and obtain optimal solutions to the problem provided.* Otherwise, the maximum achievable score for the implementation is 1 point. This is, if part 1 is not correctly implemented, the maximum score is 1; if part 2 is not correctly solved, the maximum score is 4 points.

5 Submission

The deadline for submitting the assignment is October, 29 at 23:55. This is a hard deadline. Only one student from each team must submit a single `.zip` file to the assignment section of 'Aula Global'. This file must be named `p1-NIA1-NIA2.zip`, where `NIA1` and `NIA2` are the last 6 digits of each student's NIA padding with `0`s if necessary. For example, `p1-054000-671342.zip`.

Uncompressing the `.zip` file must generate, at least, two directories called "`part-1`" and "`part-2`". If the optional part is submitted also, then a third directory should be generated with the name "`part-3`".

The report, in .pdf format, shall be located at the root of these directories and should be called `NIA1-NIA2.pdf` (after properly substituting the NIA of each student, e.g., `054000-671342.pdf`). The solutions (either source code or the spreadsheet) should be included in their corresponding directory.

Important: ignoring these guidelines one way or another might result in a loss of up to 1 point in the final score.