Lab assignment 1: *Modeling Linear Programming Tasks*
Heuristics and Optimization
Computer Science, 2020–2021

1. Goal

The goal of this assignment is to learn to model Linear Programming (LP) and Mixed Integer Linear Programming (MIP) tasks, and to solve them with two different tools: The LibreOffice spreadsheet and a modeling language, MathProg.

2. Problem statement

An important airline company has contacted the students of "*Heuristics and Optimization*". After a number of meetings you and your partner should solve a number of different optimization problems using Linear Programming.

2.1. Part 1: Basic Modelling with LibreOffice Calc

The first problem to deal with is the number of tickets that have to be offered for the different fares the company offers. Indeed, the airline company has reported they offer the following fares: standard, leisure plus and business plus, each one with different advantages to the passengers as shown in Table 1.

Fair	Allowed baggage	Price
Standard	1 kg.	19€
Leisure plus	20 kg.	49€
Business plus	40 kg.	69€

Tabla 1: Description of the fares offered by the airline company

Table 1 shows for each fare the allowed baggage and the price of the corresponding airline ticket. So far, the standard fare allows passengers to travel with no more than 1 kg for a price of 19€. In addition, the airline company consists of 5 airplanes whose description is provided in Table 2.

Airplane	Seats	Capacity
AV1	90	1,700 kg.
AV2	120	2,700 kg.
AV3	200	1,300 kg.
AV4	150	1,700 kg.
AV5	190	2,000 kg.

Tabla 2: Number of seats and capacity of each plane of the company

The number of available seats and the capacity of each airplain is shown in Table 2.

It is therefore requested to determine the number of airline tickets of each fare to offer for each airplane so that the overall profit is maximized considering the following observations:

1. It is not allowed to sell more tickets for an airline than its number of available seats.

2. It is strictly forbidden to exceed the maximum capacity of each airplane.

3. At least, 20 leisure plus airplane tickets should be offered for each airplane, and at least 10 business plus airline tickets for each airplane as well.

4. Because this is a low-cost company, the number of standard air tickets should be at least the 60 % of the overall number of airline tickets offered.

It is requested to:

1. Model the problem as an Integer Linear Programming task.

2. Implement and solve the model in a spreadsheet (LibreOffice Calc).

2.2. Part 2: Advanced modeling with GLPK

Another of the important optimization problems faced by this airline company is the assignment of landing runways to airplanes. The airport is known to have four runways, each one with the availability shown in Figure 1. Each row represents the availability of each runway, so that each cell is the availability of the runway for a specific timeslot. Green slots mean that the runway is free during that slot so that it can be assigned for an airplane to land; black cells instead stand for unavailable slots, so that they have been already assigned to another plane.

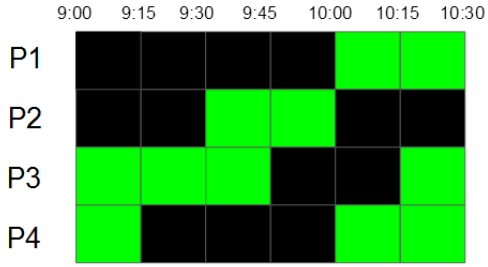


Figura 1: Availability of the landing runways of the airport. Each cell stands for a different runway so that each cell represents the availability of a runway in a specific timeslot which can be either available (in green) or unavailable, in black.

The airline company has reported that its 5 airplanes are about to land in the airport. The scheduled landing times of each airplane along with the maximum allotted landing time are shown in Table 3. The landing operation involves assigning a landing runway and also a specific timeslot whose starting time has to be either equal or greater than the scheduled landing time, and also less or equal than the maximum allotted landing time. For instance, airplane AV1 has a scheduled landing time equal to 9:10, so that it can not be assigned to the landing runway P3 and the timeslot corresponding at 9:00. However, it can be assigned to the timeslot at 10:15, even if this means that passengers would be delayed 65 minutes. Instead, airplane AV2 can be assigned to the slot at 9:00 and the runway P4, but not to other slots after 9:30, as the airplane might run out of fuel. Finally, every minute of delay with respect to the scheduled landing time involves an additional cost as more fuel has to be used, as shown in the last column of Table 3.

Thus, it is required to solve the assignment of planes to slots and runways while minimizing the overall cost due to the delays and taking into account the following observations:

1. Every airplane should be assigned one slot for landing.

2. No more than one slot has to be assigned to each airplane.

3. Assigned slots should be available, i. e., shown in green in Figure 1.

Airplane	Scheduled landing time	Maximum allotted landing time	Additional cost
AV1	9:10	10:15	100€
AV2	8:55	9:30	200€
AV3	9:40	10:00	150€
AV4	9:55	10:15	250€
AV5	10:10	10:30	200€

Tabla 3: Tabla con la hora de llegada de los vuelos.

4. The starting time of an assigned slot shall be either equal or greater than the scheduled landing time.
5. The starting time of an assigned slot shall be either less or equal than the maximum allotted landing time.
6. For safety reasons it is not allowed to assign to consecutive slots in the same track.

It is requested to:

- Model the new problem as an Integer Linear Programming task.
- Implement the model of the first part in MathProg. That problem is easier than this one so that this is intended to exercise with the syntax of MathProg before implementing the second model.
- Once the model of the first part has been implemented in MathProg, it is requested to perform the necessary modifications to include also the optimization problem discussed in the second part. For this, it is highly recommended to implement the model for the second part separately, and then to carefully determine how to combine both problems into a single one. In the end, a single optimization problem in MathProg must be delivered whose solution considers both parts simultaneously.

2.3. Part 3: Analysis of results

All results obtained in the previous parts have to be analyzed in this one. 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 planes/runways and explain if these changes affect the difficulty for solving the resulting problem.

Note that when considering exclusively the second part of the lab assignment, flight arrivals might be delayed and this might require re-computing the assignment of planes to take-off runways. How do you propose to consider this type of delays? Assume that flight AV1 has a delay of 20 minutes, is it possible to find a solution? If so, what is the new assignment of flights to runways and time slots?

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

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. It is mandatory to perform at least the first part and to write the report.

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)

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 be clear what is the purpose and meaning of each variable/constraint).
- Justify in the report all your design decisions.

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 November, 8 at 23:55. This is a hard deadline and it will not be extended.

Only one student from each team must submit:

- A single `.zip` file to the assignment section of 'Aula Global' through the submission entry point named "*First lab assignment*".

  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.

  Example: `p1-054000-671342.zip`.

- The report in pdf format shall be submitted through the Turnitin submission entry point enabled in Aula Global named "*First lab assignment (only pdf)*".

  The report in pdf format and shall be named `NIA1-NIA2.pdf` —after properly substituting the NIA of each student.

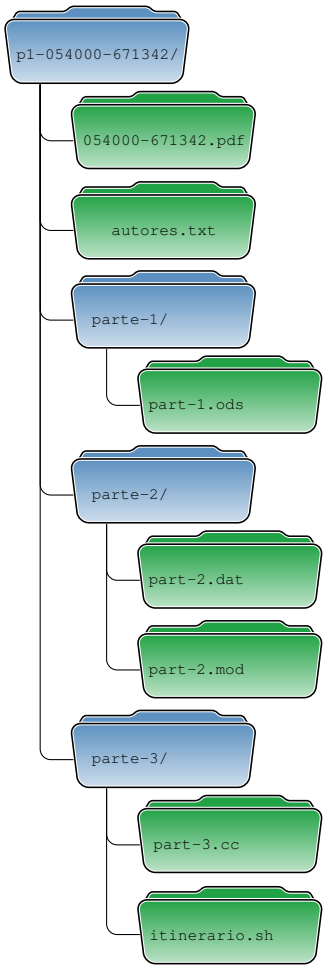
  Example: `054000-671342.pdf`.

Uncompressing the `.zip` file submitted through the first entry point must generate a directory named `p1-NIA1-NIA2`, where `NIA1` and `NIA2` are the last 6 digits of each student's NIA padding with `0`s if necessary. This directory shall contain: first, the same pdf submitted through Turnitin, and shall be named `NIA1-NIA2.pdf` —after properly substituting the NIA of each student; second, a file named `autores.txt` that identifies the members of them team, one per line with the format: `NIA Surname, Name`. For example:

```
054000 Von Neumann, John
671342 Turing, Alan
```

In addition, this directory shall contain also the directories called "`parte-1`" and "`parte-2`". If the optional part is submitted also, then a third directory should be generated with the name "`parte-3`". The solutions (either source code or the spreadsheet) should be included in their corresponding directories.

The following figure shows a plausible distribution of files after decompressing the `.zip` file:



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