Lab 1: Modelling Linear Programming Tasks

Heuristics and Optimization

100362610 - MANYA KHANNA

100386238 - LUCÍA CARRERA

Table of Contents

100362610 - MANYA KHANNA
100386238 - LUCÍA CARRERA

Introduction

For this first lab, we solved two different problems and modeled them as integer linear programming tasks. The second problem was an extension of the first, in this document we will be describing the models for part 1 and part 2 and explaining the decisions we carried out. Apart from modeling both parts, we implemented the first part in libreoffice and intended to implement both parts 1 and 2 using Mathprog and then their combination. In this document we will be going over most of these steps and the rest of the code has been attached along with this document.

Additionally we will further analyze the steps that we have taken in this project and look into the optimality of each decision along with their complexity.

Description of the Models

In this section we will be describing the models for the first and second part of this project. We will also go into further detail to explain our design decisions.

Problem 1

Our decision variables:

- x_1 : Number of Standard Tickets
- x_2 : Number of Leisure Plus Tickets
- x_3 : Number of Business Tickets

Our Resources:

- c_i : Capacity of Airplane i
- s_i : Seats available in Airplane i

The model that we came up with for the part 1 is as follows:

1	max $z = 19x_1 + 49x_2 + 69x_3$	Maximization function for overall profit
2	$x_1 + 20x_2 + 40x_3 \leq w_i$	Air planes can't exceed their capacity
3	$x_1 + x_2 + x_3 \leq s_i$	The airline can't sell more tickets than the total seats
4	$x_1 \geq 0.6\ (x_1 + x_2 + x_3\ )$	At least 60% of tickets are standard tickets per airplane
5	$x_2 \geq 20$	At least 20 leisure plus tickets are offered per
6	$x_3 \geq 10$	At least 10 business plus tickets are offered per
7	$x_{1,}\ x_{2,}\ x_3 \geq 0$	All the variables must be greater or equal to 0

In our model we implemented all the restrictions that was required, as they were all important to attain the most optimal solution. For example, we specified that the variables that are to be outputted must be in integer form. This was to get a whole number to see how many tickets can be sold of each. Furthermore, it was made sure that the capacity of each airline as well as the total number of seats was not surpassed that what was specified. Apart from that, in order to optimize the profit for the airline, we put in the required restrictions for how many tickets can be sold for each type of seat.

We further implemented this model into libreoffice in order to get the most optimal solutions for the data that was provided to us for different airplanes for the maximum capacity and the total seats.

100362610 - MANYA KHANNA
100386238 - LUCÍA CARRERA

Problem 2

Our decision variables:

- x_{aij} : Assigned track j in slot i for airplane a
- D_{ai} : Additional cost of airplane a in slot i
- T_i : Starting time of slot i
- S_a : Scheduled landing time of airplane a
- M_a : Maximum slotted time of airplane a
- A_{ij} : Availability of track j in slot i

1	$\min z = \sum_{a}^{a=1} \sum_{i}^{i=1} \sum_{j}^{j=1} D_{ai} x_{aij}$	Objective function to find the minimum total cost.
2	$\sum_{i}^{i=1} \sum_{j}^{j=1} x_{aij} = 1$	Every airplane a should be assigned one slot i and one track j for landing and no more than one slot i and track j has to be assigned to each airplane.
3	$x_{aij} + x_{a(i+1)j} \leq 1, \quad 1 \leq i \leq 5$	For safety reasons it is not allowed to assign to consecutive slots in the same track.
4	$\sum_{a}^{a=1} x_{aij} \leq 1$	Only one airplane a per track j in time slot i.
5	$\sum_{i}^{i=1} \sum_{j}^{j=1} T_i x_{aij} \geq S_a$	The starting time of an assigned slot i shall be either equal or greater than the scheduled landing time.

6	$\sum_{i}^{i=1}\sum_{j}^{j=1} T_i x_{aij} \leq M_a$	The starting time of an assigned slot shall be either less or equal than the maximum allotted landing time.
7	$\sum_{i}^{i=1}\sum_{j}^{j=1} x_{aij} A_{aij} = 1$	Assigned slots i should be available for airplane a to land.
8	$x_{aij} \in 0,1$	x should take binary values

In our design we decided to make the variable x binary because it is used to assign a slot and track to each airplane, thus only needing to be a zero or a one. The minimization function is obviously used because we want to minimize the overall costs. In the case of row 2 in our table, you can see that we decided to implement two restrictions in one, because it was possible and it made sense. In row 3, our constraint has the slot i bounded from 1 to 5 because $x_{a(6+1)j}$ would be out of bounds.

Combination of Problem 1 & 2

The only logical solution that we found to combine both parts of this practical assignment was to find the overall profit of the airline company by calculating profit = benefits - costs. Being benefits part 1 and costs part 2.

$$max\, z = \sum_{a}^{a=1}\sum_{f}^{f=1} C_f y_{af} - \sum_{a}^{a=1}\sum_{i}^{i=1}\sum_{j}^{j=1} D_{ai} x_{aij}$$

In order to get an optimal solution for this lab, we joined our solutions for problem 1 and 2 on mathprog. If we were to model the objective function it would be the one above and we would add the C_f which represents the cost according to the fare f and y_{af} represents the amount of tickets sold in airplane a of fare f . For the combination of this problem we use the maximization function and place the problem 2's function subtracting as then the maximization will work to minimize the cost.

Analysis of results

In this section we will be analyzing different results that we obtained for the two different parts and their optimality as well as their complexity.

Problem 1

For the first exercise, we analyzed the results using the same data but with different methods. We processed the data in both libreoffice and in mathprog. The results received were obviously the same as we were inputting the same functions and data. The results obtained were:

Decision Variables	Standard Tickets	Leisure Plus Tickets	Business Plus Tickets
Value AV1	54	20	16
Value AV2	72	20	28
Value AV3	153	37	10
Value AV4	90	40	20
Value AV5	114	58	18

It is clear that all the constraints that have been met that were stated in the previous section.

From all these constraints that have been set, they are all mostly equally important. For example it is crucial that the total number of occupied seats and the total capacity of the airlines are not surpassed. Hence, in order to make sure of this, the second and the third equation are very important.

Furthermore, in order to have a reasonable profit, the fifth and the sixt constraints are also very important to make sure that enough leisure plus and business plus tickets are sold.

Similarly it is relevant to have the constraint that the number of tickets sold should be greater than or equal, to avoid unreasonable data.

In total we solved this problem using 7 constraints and three variables one for each type of ticket; standard, leisure plus and business plus.

In order to further analyze our findings we will make few changes to the model and the data and see how it changes the difficulty to solve the problems. We will do this using mathprog, by running the functions and seeing the time and the memory it uses up in order to solve the problem.

Initially, the program takes 0.0secs to run the program and uses 0.2 MB (174072 bytes) of memory.

In order to see how this changes we will try to experiment with the size of the data and see if that makes a difference.

When we doubled the number of airplanes, we noted that although the time it took remained at 0.0sec, the memory used went up to 218475 bytes. It is clear that as the amount of data is increased or decreased, the memory used will also increase or decrease accordingly.

We feel that from this problem it was clear the pros and cons of using both LibreOffice and GLPK, although the modeling part was much easier to implement in libreOffice, GLPK makes is far better when trying to add more variables or when analyzing the complexity and it also gives an insight into how the program is working by showing how much memory is used each time, it helps u understand the complexity of your program and change it accordingly.

Problem 2

When we put our model into mathprog, the result we have obtained in 4500 euros of overall cost. We have defined 120 variables and 145 constraints. It has taken double the memory of just part 1, which is justifiable as it has many more variables and constraints. In our results we have that AV1 is assigned to slot 2 in track 2, AV2 to slot 1 in track 1, AV3 to slot 4 in track 3, AV4 to slot 5 in track 1 and AV5 to slot 6 in track 4.

We believe that one of the most important constraints is the following one as it is a matter of safety that an airplane lands on a track that is available.

$$\sum_{i}^{i=1} \sum_{j}^{j=1} x_{aij} A_{aij} = 1$$

In the case that airplanes were delayed we propose to aggregate to the constraint below after Ma: 15 multiplicated by a variable with a range from 0 to n, being 0 that there is no delay and n being the maximum amount of delay possible without the plane being cancelled. This would make later slots available. For 20 minutes of delay the var would be 2.

$$\sum_{i}^{i=1} \sum_{j}^{j=1} T_i x_{aij} \leq M_a$$

Combination of Problem 1 & 2

When combining part 1 and part 2 we obtain 19600 which if we sum the total results of benefits in part 1 we get 24100 euros and in part 2 the overall costs are 4500, if we subtract them, we can clearly see that our results are correct. Furthermore, another thing that we have noticed is that this combination, logically, takes a bigger amount of memory, using 0.5 Mb (528528 bytes). As the other problems, it also takes 0.0secs to execute. We have defined 135 variables and 170 constraints. If we were to remove runways this problem would be much easier as the matrix would only be bidimensional.

Conclusions

Overall this has been a very interesting project to work on, this project gave us an insight into how some everyday problems can be modelled using something as simple as a linear programming task, even though the complexity of a linear programming task can greatly vary.

We were able to solve part 1 but we greatly struggled when working on part 2 of this project, we found the modelling part to be pretty complicated. For part 1, although we were able to complete the modelling part and the libreoffice easily, we struggled a little bit when implementing the model into MathProg, especially since it was a little new to us. However, in the end it worked and we attained the same solutions as we did in libreoffice.

After finishing the implementation for part 1 in mathprog we implemented part 2 in mathprog which we found to be a little difficult as we were unsure how to implement the matrices into the system. In the end we were able to solve it and put the results together with part 1.

Overall through this project we have learnt better how to implement a linear programming task in a real setting and how it can be used to solve everyday problems. We further learnt different ways in which we can implement it using different programs, namely libreoffice and mathprog.