# Modeling Linear Programming Tasks

Fernando García Sanz & Antonio de Padua Díaz Martínez

November 1, 2017

# Contents


**1 Introduction** **3**

**2 Problem 1** **3**

2.1 Linear Programming Model . . . . . 4

2.2 Decision Variables . . . . . 4

2.3 Constraints . . . . . 5

2.3.1 Weight . . . . . 5

2.3.2 Volume . . . . . 5

2.3.3 Gravity Center . . . . . 6

2.4 Objective Function . . . . . 6

2.5 Results . . . . . 6

**3 Problem 2** **7**

3.1 Decision Variables . . . . . 8

3.2 Constraints . . . . . 8

3.2.1 Minimum of one pilot and one attendant in every flight . . . . . 9

3.2.2 Number of hours of Attendants larger than Number of hours of Pilots . . . . . 9

3.2.3 Breaks of Pilots . . . . . 9

3.2.4 Location of Crew . . . . . 10

3.3 Objective Function . . . . . 11

**4 Analysis of the results** **11**

4.1 Statement 1 . . . . . 11

4.2 Statement 2 . . . . . 13

**5 Conclusions** **14**

# 1 Introduction

The purpose of this document is to explain the applications of linear programming to different kinds of problems which happen in an airline.

By means of linear programming, we will be able to solve them, finding the optimal solution for the airline, allowing them to save the biggest amount of money possible.

The document is split into three blocks: The model and solution by means of MathProg and Open Office of the allocation of different types of suitcases into the compartments of a plane; the model and solution making use of MathProg of the airline flight crew distribution, saving the maximum amount of money available; the comparison of the solutions obtained in the two previous blocks, comparing the results we got, the tools we have used, their complexity and the way of modelling: variables used, constraints...

# 2 Problem 1

This problem consists of modelling and solving with Open Office the following statement:

An airline has decided to optimize the cost of storing the luggage into one of its planes, storing as many as they can into the vehicle compartments. Three kinds of suitcases can be stored, each one with different dimensions, weight and storage cost. Inside the plane, there are six compartments, three per side of the plane and each one equal to its opposite in terms of maximum volume available and maximum weight the compartment can afford. It is also known that the weight in the first compartments of the plane must be a ten percent bigger than the one in the compartments which conform the tail of the aircraft.

Also we know the following information about the suitcases and compartments:

- Suitcases:

| Category | Number | Weight(kg) | Size(cm) | Cost(€) |
|---|---|---|---|---|
| M1 | 22 | 7 | $30 \times 20 \times 10$ | 10 |
| M2 | 18 | 8 | $40 \times 20 \times 10$ | 20 |
| M3 | 11 | 10 | $50 \times 30 \times 20$ | 30 |

- Compatments:

| Compartment | Weight(kg) | Volume(m3) |
|---|---|---|
| C1 | 50 | 0.1 |
| C2 | 60 | 0.15 |
| C3 | 40 | 0.007 |
| C4 | 50 | 0.1 |
| C5 | 60 | 0.15 |
| C6 | 40 | 0.007 |

## 2.1 Linear Programming Model

This statement is solved by means of linear programming, crafting a model where all variables to take into account, parameters of these variables and constraints to respect are represented.

## 2.2 Decision Variables

The objective is to minimize cost, which is determined by the number of suitcases moved to the hold. There is a different cost, weight and dimensions for every type of suitcase. Every compartment has a volume and an allowed weight, as we can see the compartments C1, C2 and C3 present the same features as compartments C4, C5 and C6 respectively, despite this fact we treat them as different compartments in the modelling due to some further constraints. Every compartment can fit all three types of suitcases, so the decision variables are the number of suitcases of each type in each compartment.

In order to represent them we use:

$$X_{ij}$$

$$\forall \quad i \in M1, M2, M3 \quad ; \quad j \in C1, C2, C3, C4, C5, C6$$

being:

- i: type of suitcase
- j: number of compartment

### 2.3 Constraints

The limitations set for the suitcase placing belong to three types: the weight of the compartments is not exceeded, the volume of the compartments is not exceeded and the gravity center is shifted forward. Having these in mind, the constraints regarding the compartment limitations are modelled for every compartment so we have six of each type. We will go over type by type as most of the constraints follow a common pattern.

#### 2.3.1 Weight

Having a look at the tables we can see the maximum weight for every compartment. Therefore the number of suitcases of every type must be multiplied by their corresponding weight and then summed in order to obtain the total weight that the compartment will be provided. If such weight is higher than the maximum allowed weight for every compartment then decision variables cannot take the current value. We must check this for every compartment. Where:

Weight $\in M1 = 7, M2 = 8, M3 = 10$

MAXweight $\in C1 = 50, C2 = 60, C3 = 40, C4 = 50, C5 = 60, C6 = 40$

$$\sum_{j=C1}^{j} X_{1j} * Weight_{M1} + X_{2j} * Weight_{M2} + X_{3j} * Weight_{M3} \leq MAXweight_j$$

#### 2.3.2 Volume

In the same tables the volume is specified as well and we follow the same logic than in Weight with the difference that first we calculate the volume of the suitcases as it comes specified in dimensions. Therefore:

Volume $\in M1 = 0.006, M2 = 0.008, M3 = 0.03$

MAXvolume $\in C1 = 0.1, C2 = 0.15, C3 = 0.07, C4 = 0.1, C5 = 0.15, C6 = 0.07$

$$\sum_{j=C1}^{j} X_{1j} * Volume_{M1} + X_{2j} * Volume_{M2} + X_{3j} * Volume_{M3} \leq MAXvolume_j$$

#### 2.3.3 Gravity Center

The statement also says that we have to take into account that the weight in the front of the plane has to be a ten percent bigger than the one stored in the tail of the aircraft. Due to this constraint we consider the Compartments with the same characteristics as individual Compartments in the calculations as previously mentioned. To express it, we have modelled the following condition:

$$1.1*(\sum_{i=M1}^{i} X_{iC1} * Weight_i + X_{iC4} * Weight_i) \geq \sum_{i=M1}^{i} X_{iC3} * Weight_i + X_{iC6} * Weight_i$$

i $\in M1, M2, M3$

## 2.4 Objective Function

Now that we have defined all the constraints, we need to define the objective function, the one that will provide us the optimal solution for the statement, according to the decision variables and the constraints we previously added:

$$\text{Minimum(Z)} = (22 - \sum_{i=C1}^{i} X_{M1i}) * 10 + (18 - \sum_{i=C1}^{i} X_{M2i}) * 20 + (11 - \sum_{i=C1}^{i} X_{M3i}) * 30$$

i $\in C1, C2, C3, C4, C5, C6$

## 2.5 Results

By means of Open Office, we have calculated the results of this problem, obtaining the following results:

Result of the objective function Z: €160 is the minimum cost to assume

According to the values of the decision variables:

| Compartment | M1 | M2 | M3 |
|---|---|---|---|
| C1 | 2 | 2 | 2 |
| C2 | 0 | 2 | 4 |
| C3 | 0 | 5 | 0 |
| C4 | 2 | 2 | 2 |
| C5 | 0 | 5 | 2 |
| C6 | 2 | 2 | 1 |

# 3 Problem 2

The problem to optimize in this case, is the assignment of the shifts of a flight crew among the flights of a given day. All flights are either from Madrid to Valencia or vice versa. The crew consists of three pilots and three flight attendants. Each member of the crew earns a different amount of money depending on the flight. Every pilot has a different break time which they have to take in between consecutive flights before being allowed to perform another flight. The final objective is to reduce costs and we are told that the first flight carries passengers and their luggage so we must apply the same optimization of the first problem as well. For simplicity's sake we will not explain again the Decision Variables or the Constraints of the first problem.

The following tables contain such information:

- Flight Information:

| Flight | Route | Departure | Arrival |
|---|---|---|---|
| F1 | Madrid-Valencia | 8:30 | 9:45 |
| F2 | Valencia-Madrid | 10:30 | 12:00 |
| F3 | Madrid-Valencia | 13:00 | 14:45 |
| F4 | Valencia-Madrid | 15:00 | 16:05 |
| F5 | Madrid-Valencia | 17:45 | 19:00 |
| F6 | Valencia-Madrid | 19:50 | 21:00 |

- Earnings per flight hour (PILOTS):

| Pilot | F1 | F2 | F3 | F4 | F5 | F6 |
|---|---|---|---|---|---|---|
| P1(€/h) | 20 | 17 | 25 | 34 | 31 | 22 |
| P2(€/h) | 28 | 15 | 23 | 35 | 37 | 21 |
| P3(€/h) | 27 | 16 | 24 | 31 | 35 | 29 |

- Earnings per flight hour (ATTENDANTS):

| Pilot | F1 | F2 | F3 | F4 | F5 | F6 |
|---|---|---|---|---|---|---|
| A1(€/h) | 17 | 16 | 14 | 14 | 11 | 12 |
| A2(€/h) | 16 | 14 | 12 | 15 | 17 | 11 |
| A3(€/h) | 15 | 15 | 14 | 11 | 15 | 19 |

- Breaks for the pilots(mins):

| Pilot | Break |
|---|---|
| P1 | 60 |
| P2 | 25 |
| P3 | 30 |

### 3.1 Decision Variables

The objective is to minimize the wastes of the airline by adjusting the shifts of the crew in the most efficient way. What we have to decide in order to achieve our objective is which pilot will fly each flight as well as which attendant will accompany them. Therefore, we need to represent whether a crew member takes a flight or not and then be able to calculate the cost of such flight. To fulfill our purpose we use binary decision variables taking either 0 or 1 as a value. As for the decision variables itself, there is one per every crew member and flight.

In order to represent them we use:

$$X_{ij}$$

$$\forall \quad i \in F1, F2, F3, F4, F5, F6 \quad ; \quad j \in P1, P2, P3, A1, A2, A3$$

being:

- i: Flight number
- j: Crew member (Pn: pilots, An: attendants)

### 3.2 Constraints

In this problem we are told to follow the flight schedule and respect the breaks (in tables above). But we are also told that in every flight there must be at least one pilot and one flight attendant and the flight hours of the flight attendants must be larger than the flight hours of the pilots. Also, the whole crew starts in Madrid and they must be in the departure airport in order to be available for a flight.

### 3.2.1 Minimum of one pilot and one attendant in every flight

At least one attendant must be in every flight as well as (obviously) a pilot. The modelling of this constraint is as easy as set the number of pilots and attendants both larger or equal than 1. The constraint is splitted into Attendants and into Pilots.

Flight: i $\in F1, F2, F3, F4, F5, F6$

$$\sum_{i=F1}^{i} X_{iA1} + X_{iA2} + X_{iA3} \geq 1$$

$$\sum_{i=F1}^{i} X_{iP1} + X_{iP2} + X_{iP3} \geq 1$$

### 3.2.2 Number of hours of Attendants larger than Number of hours of Pilots

We need to check that the total number of hours of flight taken by the flight attendants is larger than the hours that the pilots have flown. Therefore we have to calculate the duration of each flight (DUR) based on the departure and arrival times given in the tables.

Duration(D) $\in F1 = 1.25, F2 = 1.5, F3 = 1.75.F4 = 1.083, F5 = 1.25, F6 = 1.167$

Flight $\in F1, F2, F3, F4, F5, F6$

Pilots $\in P1, P2, P3$

Attendants $\in A1, A2, A3$

$$\sum_{i=F1}^{i} [\sum_{k=A1}^{k} X_{ik}] * D_i \geq \sum_{i=F1}^{i} [\sum_{j=P1}^{j} X_{ij}] * D_i$$

### 3.2.3 Breaks of Pilots

The minimum time between the arrival and departure flight of the next consecutive flight must be respected as agreed with the airline. In order to check the breaks are correct we have to compare the time in-between flights of a given pilot with the break associated to such pilot (BR), so we check it for every pilot. Once we have calculated the time in between every flight we just need to perform the comparison. Regarding the decision variables, we have to consider that the pilot will work in the two flights that we are currently comparing, so we sum the decision variables corresponding to the pilot and flights we are evaluating. There are three types of outcomes: pilot does not take any flight , pilot takes the first but not the second one, the pilot wants to take both flights. We have to check the break only in the last type (since for the rest of cases there is not a departure time to compute the break). We multiply the break that the pilot must take by the result of summing both decision variables evaluated minus one. This allows us to discard the outcomes belonging to the first two types explained (the break will be multiplied by 0).

Breaks(BR) $\in B1 = 60, B2 = 25, B3 = 30$

Flight $\in F1, F2, F3, F4, F5, F6$

Pilots $\in P1, P2, P3$

$$\sum_{k=P1}^{k} \left[ \sum_{j=F1}^{j} \left( \sum_{i=F1}^{i} X_{ik} + X_{jk} - 1 \leq BR_k \right) \right]$$

#### 3.2.4 Location of Crew

The whole crew starts the day in Madrid. A crew member must be in the departure airport to be able to work in a flight. In order to model this constraint we divide it in two: one for the intended flights from MAD to VAL and another constraint for the intended flights from VAL to MAD. The way we build this constraint is by cancelling pairs of flights.

For a given flight, we take the flights up to the one being currently studied. Then we sum the decision variables corresponding to the pilots that have taken a flight from Madrid. We do the same with the pilots doing so from Valencia. Now we subtract both results in order to compare pairs. If every flight cancels another then we are making sure the location of the crew is the correct for the flights. The previous result shall be 1 as maximum since the whole crew starts in Madrid and it has no pair to cancel it. If not, then a pair has not been cancelled so the crew member cannot take the flight being compared.

Flight (j) $\in F1, F2, F3, F4, F5, F6$

Crew (i)$\in P1, P2, P3, A1, A2, A3$

MAD-VAL (m) $\in F1, F3, F5$

VAL-MAD (v) $\in F2, F4, F6$

$$\sum_{i=P1}^{i} [\sum_{j=F1}^{j} [(\sum_{m\leq j}^{m} X_{mi} - \sum_{v\leq j}^{v} X_{vi}) \leq 1]]$$

The second constraint of VAL to MAD follows the same procedure with one difference: the explained calculations must be lower or equal than 0 since in this case we don't have to take into account the extra flight.

$$\sum_{i=P1}^{i} [\sum_{j=F1}^{j} [(\sum_{v\leq j}^{v} X_{vi} - \sum_{m\leq j}^{m} X_{mi}) \leq 0]]$$

### 3.3 Objective Function

Now that the decision variables have been stablished and the constraints defined (remember that the decision variables and constraints explained in the first exercise are also applied in this exercise) we have to define the objective function which will give us the optimal solution.

$$\text{Minimum(Z)} = (\sum_{i=F1}^{i} \sum_{j=P1}^{j} costs[i,j]/60 * flightTime[i] * x[i,j])$$

Taking into account that the first problem must be included in this one the final Objective Function results in:

$$\text{Minimum(Z)} = (\sum_{i=F1}^{i} \sum_{j=P1}^{j} costs[i,j]/60 * flightTime[i] * x[i,j]) + (\sum_{i=T1}^{i} costLUG[i] * (NumberM[i] - \sum_{j=C1}^{j} unitsPerComp[i,j])$$

## 4 Analysis of the results

Arrived this point, it is time of discussing the results posted on the previous sections of this document.

### 4.1 Statement 1

We have calculated the optimal solution by means of two tools, Open Office and MathProg, obtaining with both of them the same result, which is that the minimum cost possible for those conditions and parameters is €160. In both models, constraints are satisfied and parameters are the same.

Now, talking about the complexity of each model, each one has its advantages and drawbacks:

- Open Office

  The model of the problem in Open Office is bigger in terms of data quantity. As we cannot express constraints as a set, we need to define each constraint individually, supposing this a greater quantity of time spent and a bigger possibility of commit a mistake. Each datum has to be defined individually in a cell, and each formula has to make reference individually to each cell with data related with it.

  Otherwise, the use of Open Office provides us a more visual and interactive way to see the problem solution. As the data are written individually, looking for them and their modification is simple. Changing one cell will provoke that all the information will be automatically updated. For the user, is simple to see and understand the information which appears in the sheet.

  In this case, due to the individual definition of any element, we have defined eighteen decision variables, one per compartment and suitcase type, and sixteen constraints, according to maximum volume and weight per compartment, total number of suitcases and weight relation between aircraft nose and tail.

- MathProg

  MathProg is a very useful problem to model problems of this kind. As the possibility of creating sets is defined, constraints can be easily written as operation between sets, or their corresponding subsets, making possible to write in one line several constraints as iterative calculations with sets/subsets (constraint for subset 1, subset 2...). This makes easier to write conditions when the same operation is performed in different elements.

  Nevertheless, understanding the language used in MathProg requires a bigger learning curve than in Open Office, and the use of any symbol or element which is not contemplated by the tool makes that all the program will be unable to work.

  With MathProg, we had to define four constraints: one, the maximum weight allowed; next one, the maximum volume allowed, the third one, the weight difference between nose and tail and the last one, the maximum number of suitcases of each kind.

  Due to the possibility of defining sets and subsets, all individual constraints for weight, volume and number of suitcases can be each one written in only one constraint, which provides us only three constraints, and the weight difference is the fourth one.

  Again, returning to the sets, they allow us to express all our decision variables in only one line, relating them with the set of TYPE and the set of COMP (compartments). Thus, we have our eighteen decision variables expressed in only one.

  We have defined also seven parameters, but as they are all written in the .dat file, we have exactly the same data in terms of parameters as in the Open Office sheet (logic, we cannot remove information of our model).

As a global resume, MathProg allows to compress the data, making the definition of constraints and variables easier than Open Office, especially in big statements, but Open Office is more user-friendly and any change over the things already written is automatically reflected in the global problem.

## 4.2 Statement 2

For the second problem, we have calculated the optimal solution by mean of MathProg software, modeling the parameters, constraints... as explained before.

We have obtained as result €342.083, but to this, it is necessary to add to the objective funtion the optimal result of the previous point, which calculated in the point one was €160, because it is known that one of the six flights admits passengers, so the optimal cost calculated before for the suitcases needs to be added to the final cost. Then, our optimal value for this second statement is €502.083, as result of adding the first optimal solution to the second one.

The time used by MathProg for the resolution of the exercise is practically zero, the answer is almost automatic. The program checks one hundred and fifty conditions and the model fifty four variables, the thirty six corresponding to the second statement plus the eighteen that were used in the first one. The system also provides the conditions which show that the program is feasible in the output file.

For the first flight, the number of pilots and attendants is two, and that is because pilot one, who is the cheaper for the first flight, needs to have a break of at least sixty minutes. Then, to achieve the objectives of the airline, another pilot needs to go in the flight, so his corresponding attendant, to take the second flight, because pilot one is still resting. For the remaining flights, due to the time of taking off and landing of the planes, it is no necessary to repeat this procedure again.

The complexity of the problem mainly depends on the number of flights, because it provokes that all relations between resting time and pilots available change and it also creates new constraints because of it, so adding flights or removing them will change de complexity of the statement more that any other variable or parameter.

# 5 Conclusions

This work has served to us to go deeper into the contents of linear programming, understanding a wide set of possibilities in which we may apply it.

The use of different tools such Open Office or MathProg has been adequate to get a better approach to know which software is more suitable for each case, each statement, each problem.

Linear programming has demonstrated to be an excellent way to solve problems of maximization and minimization, due to its speed, relative little steep learning curve and syntax in the software used to solve the statements presented in this work.

The use of binary variables has been maybe the most difficult concept to implement, not because of their complexity, but the idea of modelling the problem using this kind of variables and making them to affect the objective function avoiding to break the linear relations built, generating operations which are not under the linear programming criteria.

The contents acquired with this practice have a real practical application, because as shown, they can be used in problems of adjusting volume, weight, price... and the environment and context can also vary a lot; in this case it is supposed that we are performing this tasks for an airline, but this kind of problems happen everyday in lots of different enterprises dedicated to varied fields.

To sum up, this practice has been useful to understand the importance and all the applications of linear programming and learning how to use these two important tools which serve to solve this kind of problems in an efficient way.