A multi-objective approach for the aircraft seat assignment at check-in

Germán Pardo González [a], Alejandra Tabares [b], David Álvarez [c]

[a] Dept. of Industrial Engineering, Bogotá, Colombia, gr.pardo@uniandes.edu.co

[b] Dept of Industrial Engineering, Bogotá, Colombia, a.tabaresp@uniandes.edu.co

[c] Dept of Industrial Engineering, Bogotá, Colombia, d.alvarezm@uniandes.edu.co

Abstract

The number of flights is significantly increasing all over the world and the logistic and operational management must be optimised. This work presents a multi-objective approach to assign seats to each passenger at check-in selecting the cheapest available seat and maximising the distance between people in the same booking. This problem is currently solved by airlines using generic rules and using different approaches (like minimising distance between people in the same booking). The requirement for this problem to be solved online (in real time) makes it difficult to solve in a short time when the number of people in the same booking is over 6. Therefore, different acceleration techniques were implemented to reduce the computational time from 133 minutes to 63 seconds in one of the worst cases. Finally, The two main objectives are conflictive between each other, that's why the pareto front was built together with a fuzzy function to assist decision makers.

Keywords

Network flows optimisation; Integer programming; OR in airlines; Multi-objective optimisation; Seat assignment.

1. Introduction

Low-cost airlines are facing a decrease in their revenue due to the passengers that prefer not to buy a specific seat before, during or after check-in and are assigned for free in the aircraft. Even some passengers that travel with their families or friends would rather to be assigned in the airplane for free. Now that air traffic is experiencing a significant growth and nowadays air travelling is more accessible for more people than before, comparing its prices to bus transport and even trains in Europe, it is necessary to optimise the operations of airlines.

In this work we want to optimally assign seats at the exact moment of check-in to every passenger that has not paid for them. We will focus the study on a plane with 32 rows and 6 seats per row which is very common for low-cost airplanes and short flights, using historical data from a flight with origin in Bogotá and destination in San Andrés.

There are many applications of operations research in airlines such as the overbooking problem to determine the optimal amount of seats to overbook given the number of seats already sold and the time before departure (Rothstein, 1971), online seat assignment (Castro & Fernando, 2020), fleet assignment (Sherali, Bish, & Zhu, 2006), simultaneous aircraft routing and crew scheduling (Cordeau, Stojkovic, Soumis, & Desrosiers, 2001), air traffic management (Agustín, Alonso-Ayuso, Escudero, & Pizarro, On air traffic flow management with rerouting. Part I: Deterministic case, 2012) (Agustín, Alonso-Ayuso, Escudero, & Pizarro, On air traffic flow management with rerouting. Part II: Stochastic case, 2012), boarding strategies (Fonseca i Casas, A., Angel, & Mas, 2013) and even social distancing in airplane seat assignments due to the COVID pandemic (Salari, Milne, Delcea, Kattan, & Cotfas, 2020).

Most of the available literature on airlines deals with the problems mentioned above and just a few cover the seat assignment problem. A relevant approach similar to our problem is in (Castro & Fernando, 2020) which focuses on deciding where to seat the passenger of different online purchases. They have a deterministic and stochastic model which considers various scenarios of future demand. The difference of that model is that they do not decide in the exact moment of check-in and their second objective is to minimise the distance between people in the same booking. Furthermore, Given the recent COVID pandemic, airlines where interested to assign seats as furthest as possible in order to prevent people spreading the virus to each other. The problem in (Salari, Milne, Delcea, Kattan, & Cotfas, 2020) is also very relevant because the maximise distance between passenger, as we do. Another important work is in (Cordeau, Stojkovic, Soumis, & Desrosiers, 2001) where the main relevant aspect is that they use binary variables to solve the simultaneous aircraft routing and crew scheduling and to accelerate the model they use the Benders' decomposition and relax the variables, obtaining great computational times and results. Additionally, To obtain an overbooking policy in an airline, (Rothstein, 1971) uses dynamic programming to obtain the optimal number of seats to overbook given the mean of ticket purchases per day. Now regarding network flows, an interesting work is in (Tabares, Muñoz-Delgado, Franco, Arroyo, & Contreras, 2019) where they use all the concepts in network flows to calculate the standard network-dependent reliability indices of distribution systems solving linear equations.

Regarding multi-objective optimisation problems and the Pareto front, the work in (Dias de Lima, Tabares, Baños Arias, & Franco, 2021) was used as guide to build the Pareto front and the fuzzy function to help decision-makers.

The structure of this document is as follows: Section 2 provides an introduction of the overall ticket sales procedure in an airline and a contextualisation of the company Viva Air. The objectives of the model we are solving are presented in section 3. The mathematical optimisation model including the formulation and explanation of the parameters is in section 4. The computational results and experiments made are presented in the section 5. And the conclusions are presented in section 6.

2. Ticket sales procedure and contextualisation

In this work, we are only concerned about the seat's assignment during the check-in (if the customer has not bought a seat yet). The customer can buy the seat at any moment, but if at the moment of check-in, the customer finally decides to not buy the seat, the airline needs to assign a seat for free. Preferably that seat is one of the cheapest and not historically preferred by customers to buy it. Additionally, if the check-in is for more than 1 person the seats assigned must be the furthest as possible, to persuade the customer to buy the seats the next time. The overall ticket sale and seat assignment procedure can be seen in figure 1:

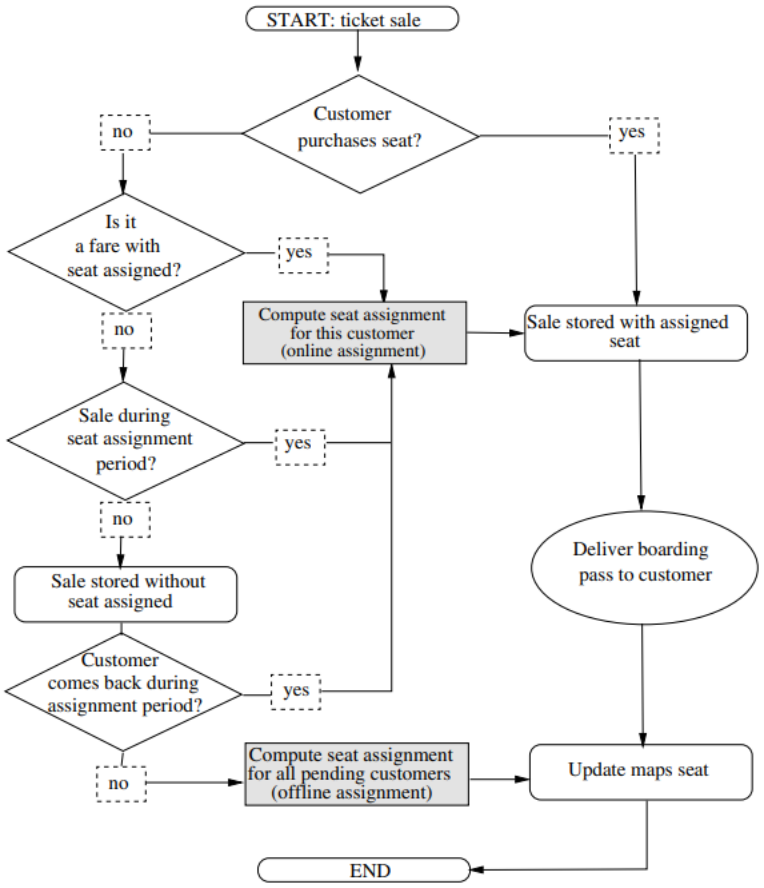


Fig 1. Overall tickets and seat assignment procedure

If we go straight with "no" on figure 1 we get to the point of seat assignment for pending customers (customers that decided to not buy a seat) and that moment is the check-in. It is important to notice that seats can also be bought at the counter desk just before the flight departure, therefore the model must leave the historically bought or preferred seats empty.

The company we are working with is Viva Air. It is a low-cost airline funded in Rionegro, Colombia in 2009 by the name of Viva Colombia, which later in 2017 was changed to the actual one. This airline offers a variety of national and international low-cost flights in the American continent. Since Viva Air was some of the first low-cost airlines in Colombia and Perú, it was able to reduce 66% and 30% respectively the flight's prices of their competitors in those countries. It has over 20 aircrafts and 35 routes. It flights around 40 thousand times per year and their mission is to offer accessible prices to their customers.

3. Research objectives

There are three main objectives in this work. Firstly, we want to minimise the cost associated with assigning a person that has not paid for the seat, taking into account that some seats are usually booked for its particular attributes, such as the window, more space or that historically people tend to purchase. Secondly, we want to maximise the distance between the people in the same booking, and logically it only applies when the bookings have more than one person ($q > 1$). The second objective's intention is to pursue customers to buy seats in the future since they prefer to flight as close as possible. The last objective concerns about the computational time to solve the model, so we want to accelerate the solution process without losing optimality in the solutions.

4. The mathematical optimisation model

4.1. Nomenclature

Sets

I: Set of seats in the airplane.

T: Time where a seat was assigned

Undirected Complete Graph

$G = (N, A)$

where $N = \{i\} \in I$ represents the nodes of the network, which in this case are the seats and $A = \{(i, j)\} \in I | i \neq j$ represents the flow through the arcs.

Parameters

a_i^t: $\begin{cases} 1 \text{ if the seat } i \in I \text{ is not occupied.} \\ 0 \quad \text{on the contrary.} \end{cases}$

c_i: Price of the seat $i \in I$.

d_{ij}: Distance from the seat $i \in I$ to the seat $j \in I | j \neq i$.

β_{ij}: 1 if the seats $i \in I$ and $j \in I | j \neq I$ are at minimum δ units of distance, 0 on the contrary.

b: Bonus price for the most purchased seats based on historical data.

q: Number of people in the booking.

δ: Minimum distance between seats in the same booking.

ω_1: Weight for the cost in the objective function.

ω_2: Weight for the distance in the objective function.

p: Iteration counter

Variables

x_i^t: $\begin{cases} 1 \text{ if the seat } i \in I \text{ is selected in the time } t \in T \mid t = p. \\ 0 \qquad\qquad\qquad\quad \text{on the contrary.} \end{cases}$

y_{ij}^t: $\begin{cases} 1 \text{ if the pair of seats } i \in I \text{ and } j \in I | j \neq i \text{ are selected in the time } t \in T \mid t = p. \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad\quad \text{on the contrary.} \end{cases}$

4.2. Formulation

$$min \sum_{i \in I} \omega_1 c_i x_i^t - \sum_{(i,j) \in G} \omega_2 d_{ij} y_{ij}^t \quad (1)$$

$$s.t.$$

$$\sum_{i \in I} x_i^t = q \; \forall t \in T \mid t = p \quad (2)$$

$$\sum_{(i,j) \in G} y_{ij}^t = qP2 \;\; \forall t \in T \mid t = p, q > 1 \quad (3)$$

$$x_i^t \leq a_i^t \;\; \forall i \in I, t \in T \mid t = p \quad (4)$$

$$(q-1)x_i^t = \sum_{j \in I} y_{ij}^t \;\; \forall i \in I, t \in T \mid t = p \, , j \neq i, q > 1 \quad (5)$$

$$(q-1)x_i^t = \sum_{j \in I} y_{ji}^t \;\; \forall i \in I, t \in T \mid t = p, j \neq i, q > 1 \quad (6)$$

$$y_{ij}^t \leq \beta_{ij} \; \forall (i,j) \in G, t \in T \mid t = p, q > 1 \quad (7)$$

$$x_i^t \leq 1 \; \forall i \in I, t \in T \mid t = p \quad (8)$$

$$x_i \in \{0,1\} \; \forall i \in I, t \in T \mid t = p \quad (9)$$

$$y_{ij} \in \{0,1\} \; \forall (i,j) \in G, t \in \; T \mid t = p \quad (10)$$

The multi-objective function (1) minimises the cost of the seat assignment at check in, selecting the cheapest seat available in the aircraft and at the same it maximises the distance between each pair of seats selected (when $q > 1$). The constraint (2) guarantees that the number of seats selected equals the number of people in the booking. Constraint (3) guarantees that there is only flow through $q - 1$ arcs for each seat selected (q$P2$ stands for q permuted 2). Constraint (4) prevents using previously assigned seats. Constraints (5) and (6) connect each node selected and guarantees flow through the connecting arcs. Constraint (7) is very important because it guarantees that the pair of seats $(i, j) \in G$ are at the minimum distance δ required by the decision-maker. It must be clarified that when the aircraft is mostly occupied the model becomes infeasible. Therefore, the parameter δ is relaxed until the solution is

feasible again. Constraint (8) imposes maximum one passenger per seat. Ultimately, (9) and (10) represent the nature of the variables, which in this case, every variable is binary. All the variables in the model are indexed by the iteration $t \in T$, for instance, if t takes the value of 3 it means that the variables were selected in the third iteration and the model is run until all the seats are assigned.

The distribution of seats is as following:

Seats distribution

Row	A	B	C	D	E	F	Row
1	1	33	65	96	127	158	1
2	2	34	66	97	128	159	2
3	3	35	67	98	129	160	3
4	4	36	68	99	130	161	4
5	5	37	69	100	131	162	5
6	6	38	70	101	132	163	6
7	7	39	71	102	133	164	7
8	8	40	72	103	134	165	8
9	9	41	73	104	135	166	9
10	10	42	74	105	136	167	10
11	11	43	75	106	137	168	11
12	12	44	76	107	138	169	12
13	13	45	77	108	139	170	13
14	14	46	78	109	140	171	14
15	15	47	79	110	141	172	15
16	16	48	80	111	142	173	16
17	17	49	81	112	143	174	17
18	18	50	82	113	144	175	18
19	19	51	83	114	145	176	19
20	20	52	84	115	146	177	20
21	21	53	85	116	147	178	21
22	22	54	86	117	148	179	22
23	23	55	87	118	149	180	23
24	24	56	88	119	150	181	24
25	25	57	89	120	151	182	25
26	26	58	90	121	152	183	26
27	27	59	91	122	153	184	27
28	28	60	92	123	154	185	28
29	29	61	93	124	155	186	29
30	30	62	94	125	156	187	30
31	31	63	95	126	157	188	31
32	32	64	-	-	-	-	32

Fig 2. Distribution of seats in the aircraft.

Now, the cost of each seat was defined using the original price in thousands COP plus a bonus that is added to the seats that are usually booked for their preferred attributes as mentioned above:

Base cost of each seat

Row	A	B	C	D	E	F	Row
1	39	34	39	39	34	39	1
2	34	29	34	34	29	34	2
3	34	29	34	34	29	34	3
4	34	29	34	34	29	34	4
5	34	29	34	34	29	34	5
6	27	22	27	27	22	27	6
7	27	22	27	27	22	27	7
8	27	22	27	27	22	27	8
9	27	22	27	27	22	27	9
10	27	22	27	27	22	27	10
11	27	22	27	27	22	27	11
12	29	24	29	29	24	29	12
13	29	24	29	29	24	29	13
14	18	12	18	18	12	18	14
15	18	12	18	18	12	18	15
16	18	12	18	18	12	18	16
17	18	12	18	18	12	18	17
18	18	12	18	18	12	18	18
19	18	12	18	18	12	18	19
20	18	12	18	18	12	18	20
21	18	12	18	18	12	18	21
22	18	12	18	18	12	18	22
23	18	12	18	18	12	18	23
24	14	9	14	14	9	14	24
25	14	9	14	14	9	14	25
26	14	9	14	14	9	14	26
27	14	9	14	14	9	14	27
28	14	9	14	14	9	14	28
29	14	9	14	14	9	14	29
30	14	9	14	14	9	14	30
31	14	9	14	14	9	14	31
32	14	9	-	-	-	-	32

Fig 3. Base cost of each seat in thousands COP

Figure 3 shows the cost for each seat before adding the bonus for the most purchased seats.

In order to determine what percentage of the bonus should be added to the price of each seat, an exploratory data analysis was performed based on real data for Viva Air flights from Bogotá to San Andrés. In the figure 4 it is possible to see the 10 most purchased seats:

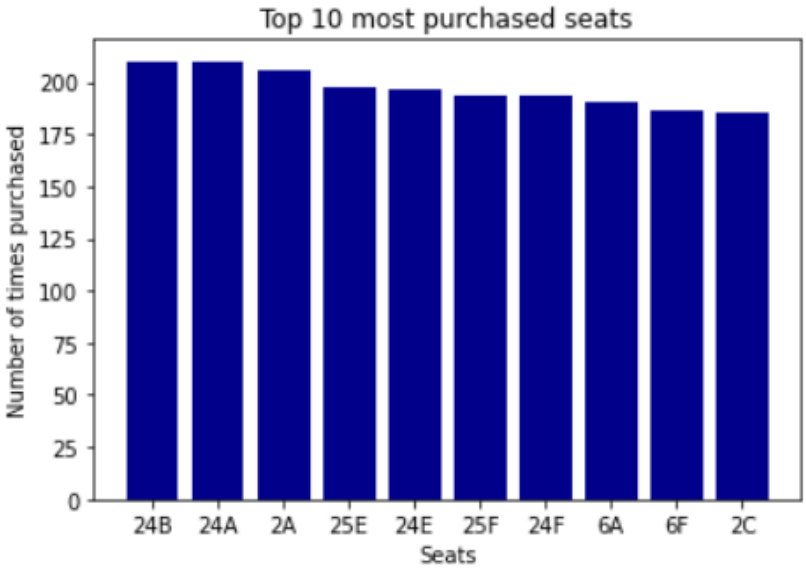


Fig 4. Top 10 most purchased seats

It is important to notice that every seat has been purchased at least one time, so we ranked each seat as a percentage of the two most purchased seats. For example, the seat 24A has a rank of 1 since it is one of the most purchased seats, whereas the seat 2C has a rank 0,88:

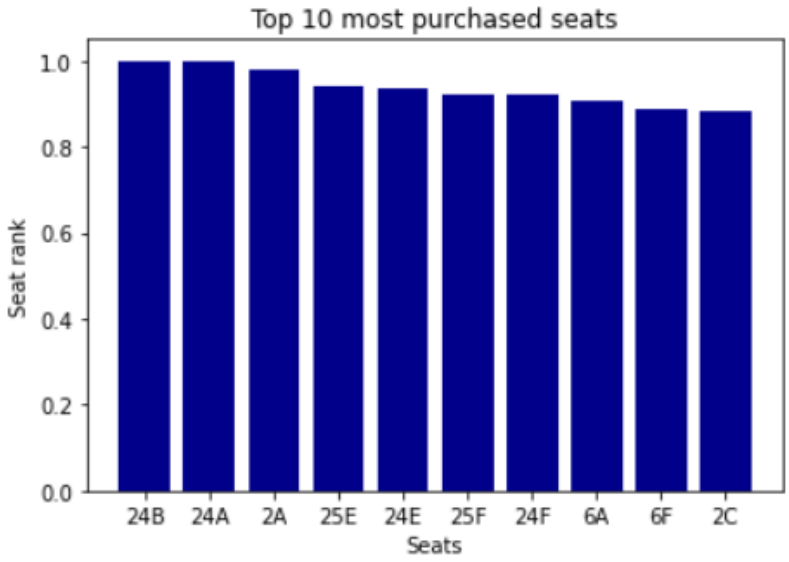


Fig 5. Rank for the top 10 most purchased seats

In order to determine the final cost of each seat the following formula was applied:

$$c_i = base_i + rank_i * b \quad \forall i \in I$$

As can be seen, the cost of seat $i \in I$ is determined by its base price plus its respective rank times the bonus, therefore the most famous seat have an additional cost of b. Exceptionally, the last three rows have a predetermined cost of 500 because these seats must be booked last and reserved for operational use and they do not have a window.

Regarding the distance between each seat, it was calculated using the Manhattan approach:

$$d_{ij} = |x_i - x_j| + |y_i - y_j| \ \forall (i,j) \in G \tag{11}$$

It is important to notice that in (11) the x_i stands for the x-coordinate of seat $i \in I$ and not for the variable. The x and y coordinates were assigned accordingly to each seat and for the corridor one extra unit was added.

An example of the graph that represents the solution to one iteration when $q = 3$ can be seen next:

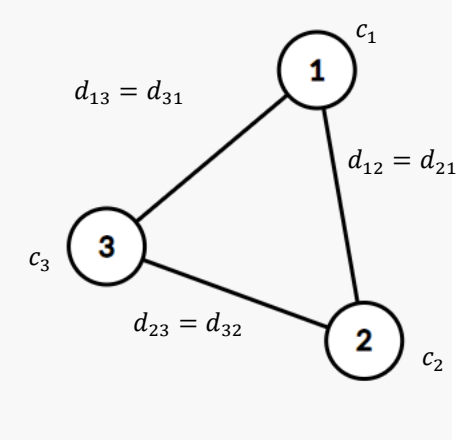


Fig 6. Example of a solution graph with three seats

In figure 6 one can see that every node in the solution is connected to each other by an undirected arc and that there are $qP2 = 6$ arcs (counting twice each arc since they are undirected). Each node in the solution represents a cost c_i and each arc stands for a distance between each node d_{ij}, thus representing the flow which in this case we want to

maximise. This solution would represent that the seats 1A, 2A and 3A (1, 2, 3 in the model) have been selected for a booking whose number of people q was 3.

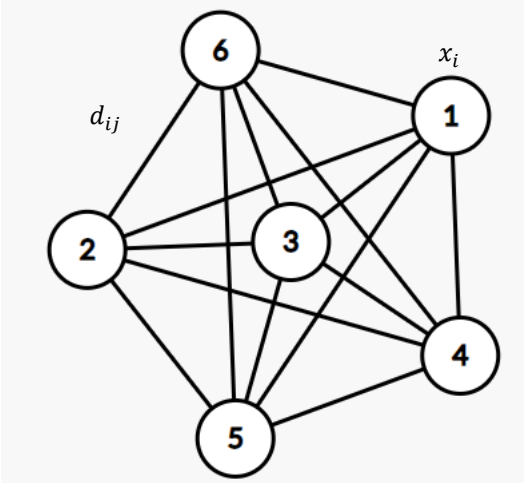


Fig 7. Example of a solution graph with six seats

In figure 7 it is noticeable how the solution gets way more complicated when the number of nodes start to increase. All the nodes in the solution need to be connected to each other to fulfil constraints (5) and (6) and to represent the flow through the nodes that are chosen in the same iteration. Moreover, the nodes that are not part of the solution are not connected and there is no flow through them.

Regarding the number of people per booking, from a dataset of 345 flights it was possible to obtain the histogram in figure 8:

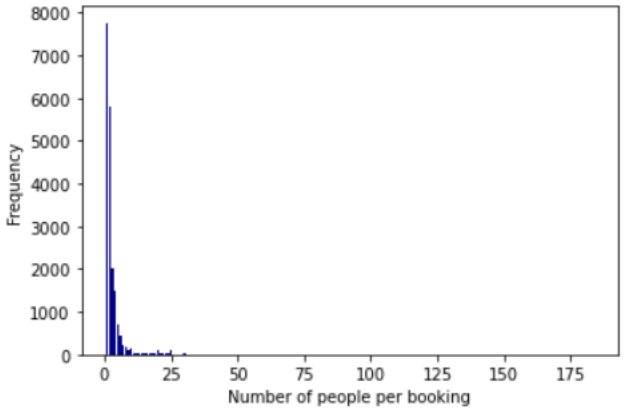


Fig 8. Histogram of the number of people per booking

It is possible to observe that the most common number of people in a booking is one, following by 2, 3 and so on. It is also interesting to see that there are bookings where the number of persons is 100 or even more. This is due to charter sales and in this special case where there are more than 10 people or so, the model needs to be accelerated.

5. Computational results

5.1. Results for Phase I

All the results were run in processor Intel(R) Core (TM) i5-8250U CPU @ 1.60GHz 1.80 GHz, RAM 8,00 GB (7,86 GB usable) in type 64-bit operating system, x64-based processor in Gurobi 9,5 student version optimiser. Also, in each run the heuristics were turned off and the 8 threads were used to get the best computational results.

The model was simulated using different scenarios and parameters which are shown as following:

$\omega_1 = 1{,}8$

$\omega_2 = 1{,}5$

$\delta = 6$

$cost_{bonus} = 100$

The actual selection of the weights demonstrated an improvement on the computational time for the model to get the optimal solution. It is possible to say that the gradient of the objective function, significantly affect the computational time. Also, the parameter δ starts with a value of 6 so that at least, the seats assigned in the same booking are not in the same row. As the airplane gets fuller, the problem can become infeasible since there is no combination of seats to satisfy the minimum distance δ, so this parameter is relaxed by one unit in such cases until the problem becomes feasible again.

For the first case, the model was tested for a $q = 1$, i.e., there was only one person in each booking in every iteration until the aircraft is full. In figures 9 and 10 it is possible to see that the time for each iteration is just two seconds maximum, so that does not represent a problem at all. Also, one can observe how the optimal value of the objective function changes and get worse throughout the iterations. That occurs because in the last iterations, the aircraft is almost full and the model needs to choose the worst seats. In the end the models chooses the seats without a window

which have the highest cost defined. This shows how the model prioritise and correctly choose the seats in every iteration.

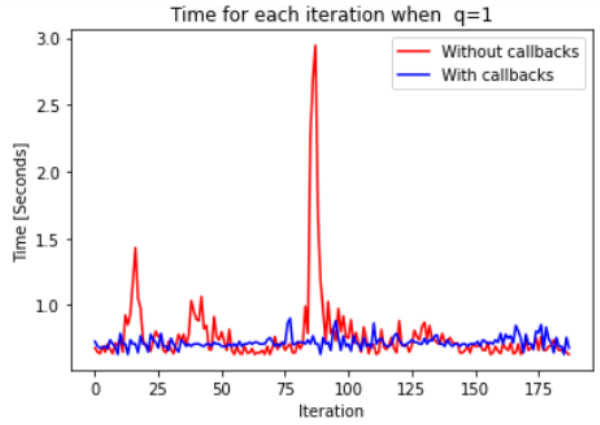


Fig 9. Time vs. iteration

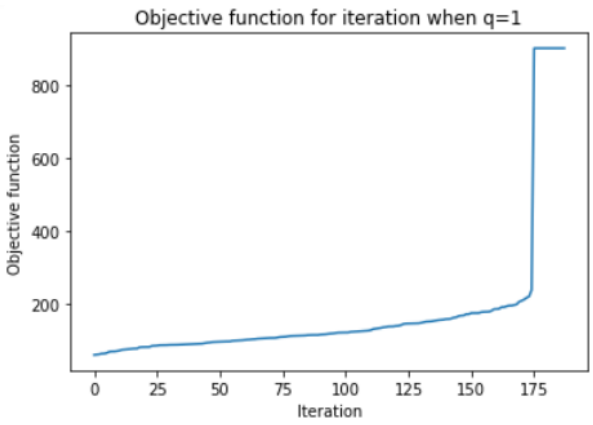


Fig 10. Objective function value vs. iteration

One can see that when $q = 1$ the model only takes at maximum around 3,0 seconds to give the optimal solution, this does not represent a problem for the model and can be used online. Also, when callbacks are implemented, the time seems to be more constant and reduced.

Now, the results for when $q = 2$ are shown:

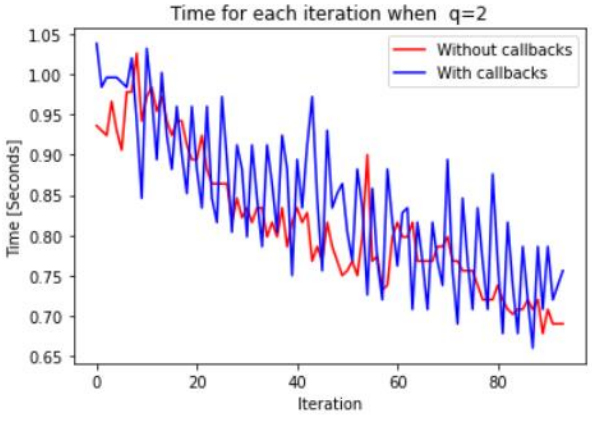


Fig 11. Time vs. iteration

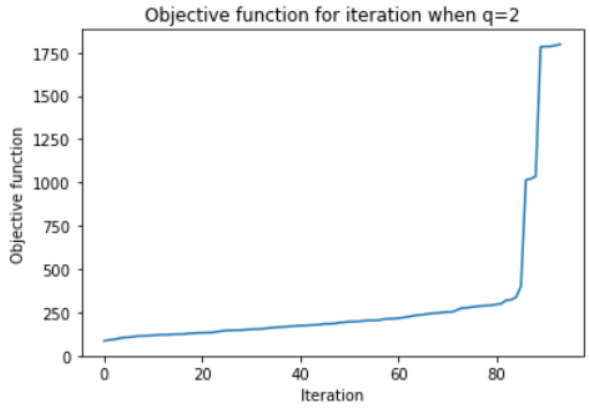


Fig 12. Objective function value vs. iteration

Row	A	B	C	D	E	F	Row
1	84	74	54	51	69	73	1
2	89	76	88	87	66	79	2
3	85	53	56	63	57	72	3
4	78	64	43	28	36	42	4
5	9	21	14	6	10	4	5
6	86	83	58	49	75	86	6
7	81	67	39	25	48	68	7
8	82	70	29	24	40	44	8
9	60	46	19	32	16	41	9
10	34	31	13	3	11	27	10
11	26	23	1	2	15	22	11
12	38	30	17	12	35	45	12
13	8	5	20	7	18	33	13
14	77	80	62	37	61	52	14
15	71	82	51	47	65	64	15
16	83	79	50	21	39	55	16
17	63	61	16	8	34	29	17
18	40	27	12	31	62	67	18
19	47	59	36	30	53	43	19
20	57	55	11	20	50	56	20
21	68	66	2	1	38	46	21
22	75	74	45	23	17	5	22
23	48	32	7	13	24	14	23
24	85	84	35	33	80	81	24
25	69	76	42	44	77	78	25
26	65	52	18	37	60	54	26
27	73	72	28	26	51	58	27
28	10	15	3	4	19	9	28
29	49	41	6	25	70	71	29
30	22	92	93	91	90	89	30
31	91	94	94	93	92	88	31
32	87	90	-	-	-	-	32

Fig 13. Seat assignment

In figures 11 and 12 one can see that throughout the iterations the computation time seems to decrease. Nevertheless, the maximum computation time is around 1,05 seconds which is not a problem and with callbacks it appears to be less constant, thus it can be preferred to run the model without callbacks when $q = 2$. As expected, the objective function value increases throughout the iterations because the airplane get fuller in every iteration. In figure 13 it is possible to see the seat assignment, for instance the seat 21D and 11C were the first to be assigned and correspond to the same booking. Since at least the 60% of the seats must be occupied in order for the airplane to depart, the balance rule is automatically in the model.

The results for when $q = 3$ are shown below:

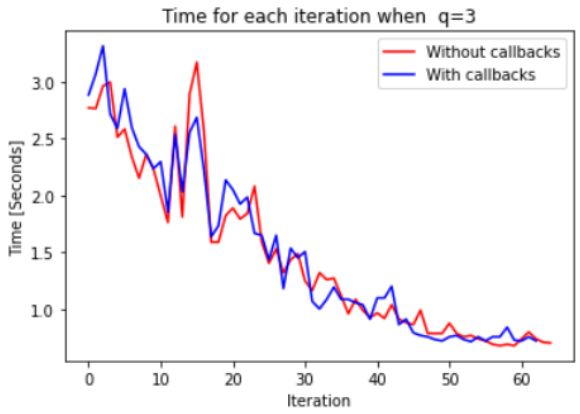


Fig 14. Time vs. iteration

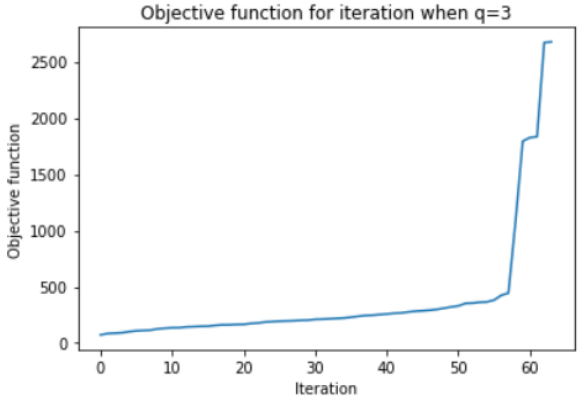


Fig 15. Objective function value vs. iteration

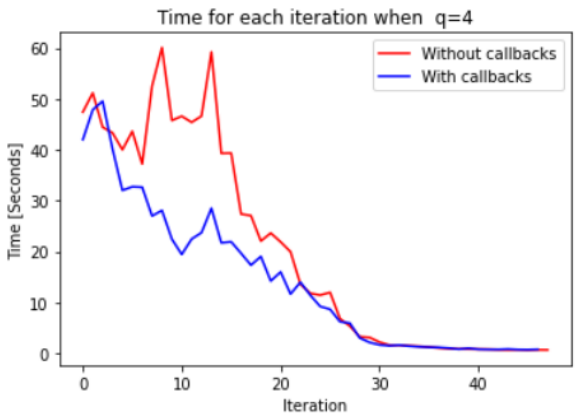
Row	A	B	C	D	E	F	Row
1	55	45	27	31	43	47	1
2	62	48	59	57	42	51	2
3	56	32	36	39	35	46	3
4	53	38	26	8	12	19	4
5	3	7	5	2	4	1	5
6	61	60	41	34	54	58	6
7	59	49	30	13	37	50	7
8	58	52	18	11	29	28	8
9	44	40	15	23	10	33	9
10	25	24	14	6	9	21	10
11	22	20	2	5	17	16	11
12	34	28	19	18	32	37	12
13	9	10	21	8	20	23	13
14	52	53	44	25	46	41	14
15	51	54	40	33	49	45	15
16	57	55	38	16	26	39	16
17	48	42	12	3	24	22	17
18	31	27	13	29	48	50	18
19	37	47	36	30	43	35	19
20	41	46	17	25	38	42	20
21	50	49	35	1	31	34	21
22	54	56	25	19	15	6	22
23	33	23	10	13	18	11	23
24	58	57	26	22	56	55	24
25	47	51	28	24	53	52	25
26	39	29	12	20	36	30	26
27	43	45	16	14	27	32	27
28	5	8	1	2	7	3	28
29	21	17	4	9	44	40	29
30	6	62	63	64	61	60	30
31	63	64	64	63	62	59	31
32	60	61	-	-	-	-	32

Fig 16. Seat assignment

When there are 3 people per booking, it is more noticeable in figure 14 how the computational time decreases significantly throughout the iterations and it seems to be indifferent from the use callbacks. Also, that means that when the airplane is fuller, the model runs and get the optimal solution faster. In this case the maximum time is

around 6 seconds and again it is not a problem for the moment of check in. The objective function appears to be increasing over the iterations as expected. And the seat assignment can be seen in figure 16.

Results when $q = 4$:



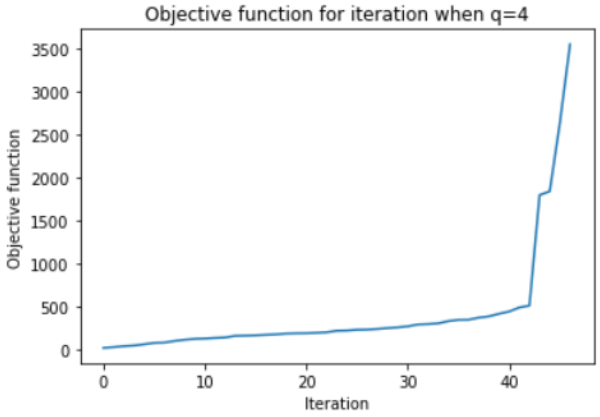


Fig 17. Time vs. iteration

Fig 18. Objective function value vs. iteration

Row	A	B	C	D	E	F	Row
1	33	25	10	16	24	32	1
2	45	29	43	41	26	35	2
3	38	12	17	27	15	34	3
4	36	20	13	6	8	9	4
5	2	7	5	3	4	1	5
6	43	44	31	28	40	42	6
7	41	39	22	11	30	37	7
8	42	40	18	14	29	23	8
9	34	30	17	19	12	27	9
10	21	23	8	4	9	18	10
11	13	15	1	2	16	6	11
12	24	22	14	10	25	28	12
13	3	7	20	5	11	21	13
14	41	44	37	26	35	33	14
15	39	43	32	31	38	36	15
16	45	42	30	15	29	34	16
17	35	33	11	7	23	13	17
18	18	14	9	22	37	39	18
19	27	36	24	21	32	20	19
20	32	34	8	17	33	31	20
21	37	38	3	1	25	26	21
22	40	41	28	16	10	2	22
23	25	19	6	12	14	5	23
24	43	42	23	18	39	40	24
25	31	35	21	20	36	38	25
26	26	16	10	15	27	22	26
27	28	29	12	13	17	19	27
28	5	6	1	4	7	3	28
29	11	9	2	8	30	24	29
30	4	46	47	46	47	45	30
31	46	47	Not	47	46	44	31
32	44	45	-	-	-	-	32

Fig 19. Seat assignment

Figures 17 and 18 show a decreasing of time throughout the iterations and it appears to become constant from the iteration 30. There is an important peak when callbacks are not used of approximately 120 seconds and it is possible that this happens because the problem becomes infeasible due to the minimum of distance between seats δ. When this occurs, the parameter δ must be relaxed by one unit until the problem is feasible again. The parameter is relaxed 0, 5, 5 and 6 times when q is 1, 2, 3 and 4 respectively. This means that in the last iteration the parameter took the value of 0 and just the last remaining seats were chosen. Additionally, it can be seen how when the callback function is implemented, the time for the solution decrease significantly in most of the cases.

Figure 20 shows how the seat where assigned in each iteration when the bookings consist of four people. For instance, one can see that seats 11C, 21D, 28C and 5F where selected in the first iteration i.e., when the plane was empty. In addition, in figure 20 it is easy to notice that the safety balance restrictions for the aircraft are not necessary. That is because the airplane will only depart if at least the 60% of the seats are occupied (from iteration1 to 28) and from figure 20 one can see that the balance of the airplane is accomplished without the restrictions. The orange colour represents the iterations of the 60% of the seats and it is obvious that the allocation is symmetric in most cases. It is interesting how the model is maximising the distance between seats in each booking and at the same time we want to choose the cheapest seat and in figure 20 one can notice how both objectives are satisfied.

As can be noticed, when the number of people in the bookings q increase, the number of iterations decrease, since more seats are selected per iteration.

Results when q is dynamic:

In this part the results when q is gathered from a real historical flight, exactly the flight on 2022-06-08 and number 59421 which has qs with two bookings starting with 5 people, then 4 and so on.

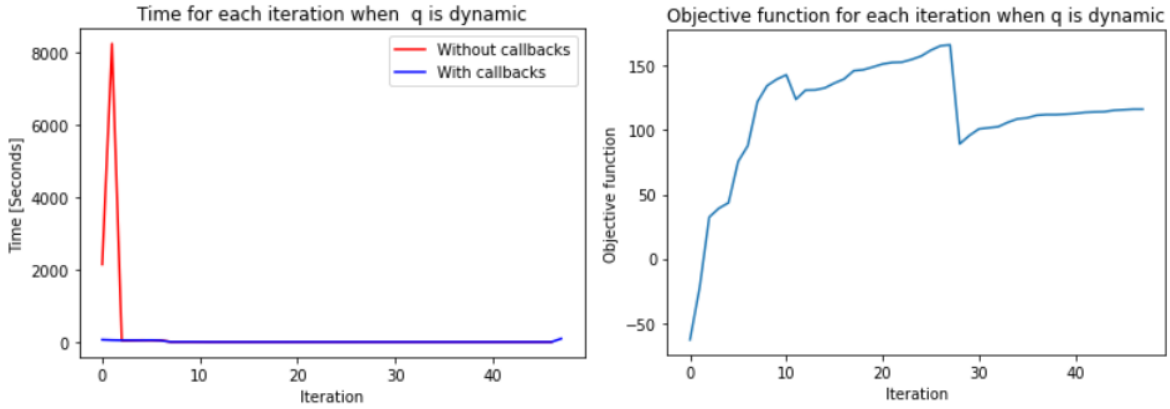


Fig 21. Time vs. iteration **Fig 22. Objective function value vs. iteration**

As can be seen, the objective function per iteration improves at some points, different for when is constant. That is because the model will find a better objective comparing for example between $q = 5$ and $q = 4$.

Also, it can be seen how the callback function appears to be way significantly important for values of $q \geq 5$. The model without callbacks lasts around 8000 seconds and with callbacks this value is decreased to 63 seconds.

Row	A	B	C	D	E	F	Row
1							1
2							2
3							3
4				4	7		4
5	1	6	5	2	3	1	5
6							6
7				20			7
8			22	11	44		8
9			14	25	8	46	9
10	28	26	10	5	9	23	10
11	21	17	2	4	16	19	11
12	34	24	12	13	30	36	12
13	3	7	15	6	18	27	13
14				29		43	14
15			35	31			15
16			32	10	28	37	16
17		47	8	2	22	17	17
18	27	16	9	19			18
19	39		25	18	38	33	19
20			7	13	41		20
21			4	1	26	40	21
22			42	14	12	3	22
23		20	6	9	15	5	23
24			23	21			24
25				45			25
26			8	24			26
27			11	10			27
28	2	5	3	1	6	2	28
29			4	7			29
30	1						30
31							31
32							32

Fig 23. Seat assignment

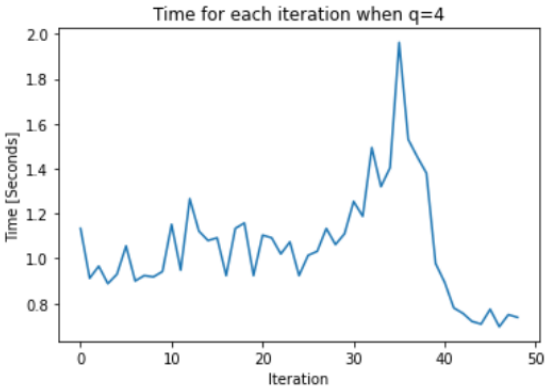
In figure 23 it can be seen that for this flight the airplane was not full and even though, the model seems to assure the balance of the plane the first two iterations are a group of five people each and the model tries to assure the maximum distance between them and the cheapest and less famous seats.

5.2. Results for Phase II

For the phase II of the project the main objective was to accelerate the solution. We modified some parameters in Gurobi in order to accelerate the solution to the most. The firs parameter modified was “Heuristics”, we turned off the heuristics because it demonstrated a higher computational time. The second parameter is “Threads” we are using the 8 available threads in the computer. The third parameter was “MIPGap” we are using this parameter so that Gurobi stops optimising when a 10% optimality gap is achieved. The last parameter modified was “NodeLimit” Gurobi will stop searching for the optimal solution if 10000 nodes in the branch and bound have been already explored. Additionally, one technique was used to accelerate the process even more. In phase I we were considering every available seat to be empty at the beginning of the iterations, but given the time vs. iteration graph it is noticeable that throughout the iterations the computational time tends to be constants and short. That means that when the airplane is fuller, the model takes less time to give the optimal solution. This technique consisted in marking as occupied the seats that are the most expensive with the previously defined cost for its attributes and operating cost. Also, throughout the iterations we started to release the cheapest seats that were marked as busy in the first place so that the model starts to consider them again. That means that in the first iterations we had a constant number of available/busy seats, it was a process of marking to busy (for the optimal solution in a iteration) and releasing the other ones that were marked busy without being physically busy.

Since the previous results just showed problems when $q \geq 4$ it is only interesting to compare the results regarding this condition. We are going to compare the computational time but also the objective function throughout the iterations because we are interested in not damaging the optimality of the problem. We started with the 80% of ghost seats busy.

The results are shown below:



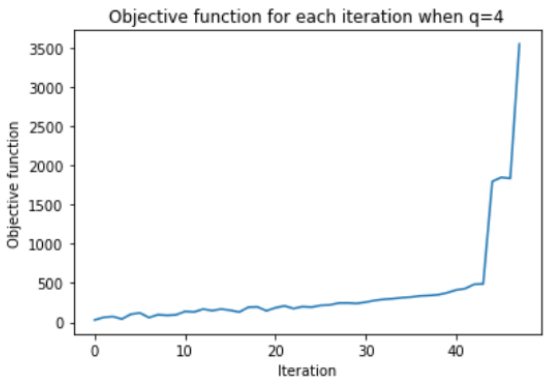
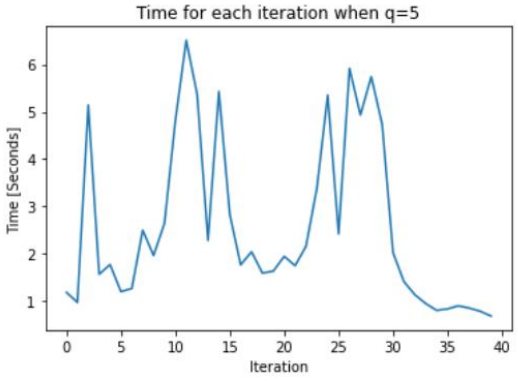


Fig 24. Time vs. iteration **Fig 25. Objective function value vs. iteration**

In the previous results we didn't compare when q was fixed to be 5 because it took around 8000 seconds for the first iterations to solve the model, but in this case and with the technique being used we were able to reduce the computational time to at most 6 seconds:



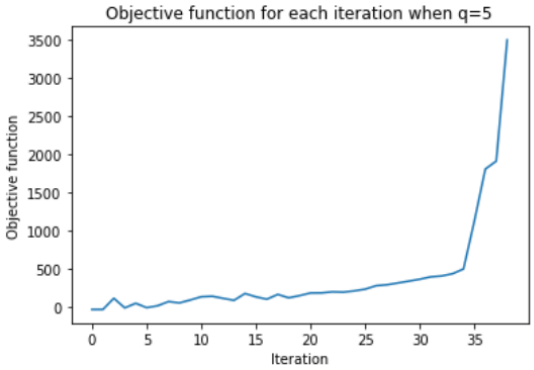
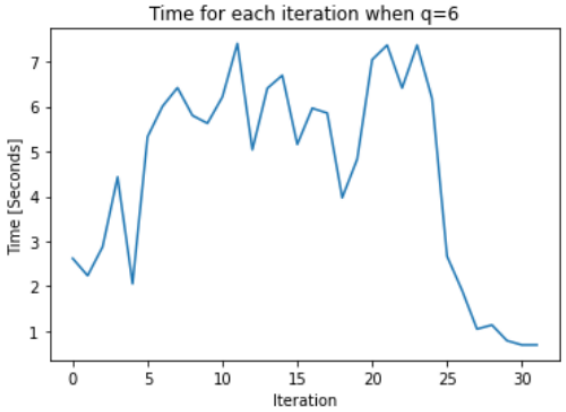


Fig 26. Time vs. iteration **Fig 27. Objective function value vs. iteration**

Finally, it is interesting to observe the limits of this model, so when $q = 6$:



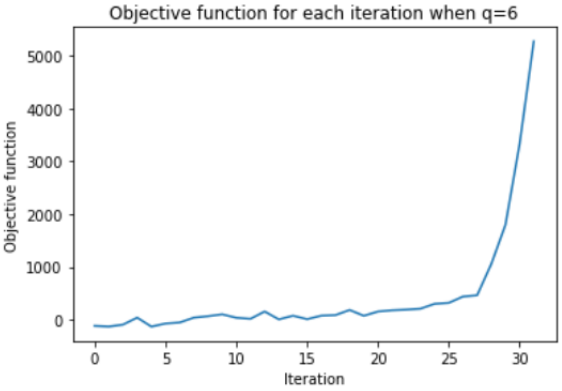
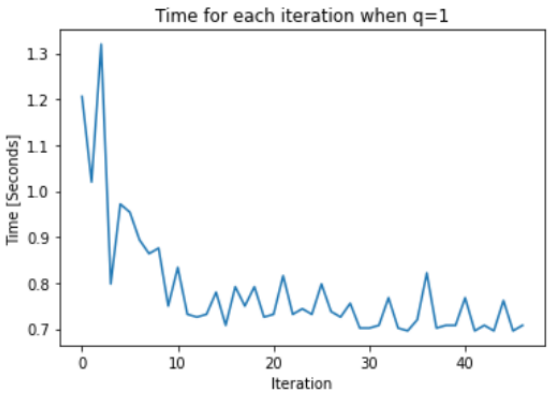


Fig 28. Time vs. iteration **Fig 29. Objective function value vs. iteration**

It is interesting to notice that the model performs very good even when the person per booking is increased by one unit, and the greatest number of people per booking are already performing very well to give results in a real application.

Now, trying with real data we can compare this instance with figures 21 and 22:



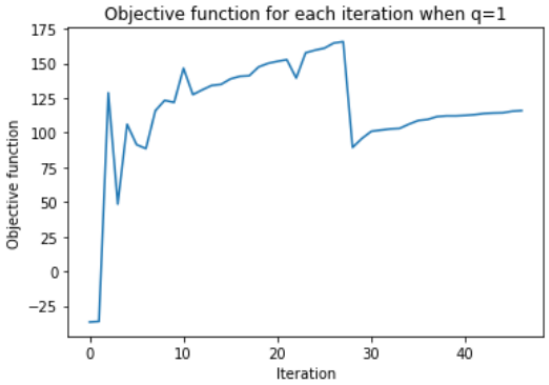


Fig 30. Time vs. iteration

Fig 31. Objective function value vs. iteration

Comparing with figures 21 and 22 we can appreciate how well the model is now performing for real instances of the airline. It gives a solution in a really small time, making this model suitable for real applications.

Row	A	B	C	D	E	F	Row
1							1
2							2
3							3
4				10	26		4
5	7	8	5	1	4	2	5
6							6
7				6			7
8			19	18	44		8
9			14	23	9	46	9
10	24	22	11	2	5	20	10
11	6	17	3	1	16	7	11
12	28	21	12	13	25	36	12
13	1	2	15	4	11	3	13
14				27		43	14
15			35	30			15
16			31	8	29	37	16
17		47	9	3	24	19	17
18	33	16	10	21			18
19	39		27	20	38	32	19
20			6	12	41		20
21			2	4	34	40	21
22			42	13	7	1	22
23		25	3	11	14	5	23
24			26	22			24
25				45			25
26			7	28			26
27			18	17			27
28	8	9	1	2	10	6	28
29			4	15			29
30	23						30
31							31
32							32

Fig 32. Seat assignment

It can be seen that the results are not exactly the same, but the model performs very good in both computational time and objective function.

6. Pareto front and decision-making approach

Since this is a multi-objective optimisation problem, decision makers can struggle to decide the appropriate weights for each part of the objective function, it would be easier to see how much it is improved one part of the objective function relative to the other part. Since in this problem we want to choose the cheapest seats, but at the same time to choose the furthest seats, these two objectives result conflictive. That's why the Pareto front was built using the augmented ϵ constraint method in (Mavrotas, 2009). The first step done to build the Pareto front was to calculate the payoff table, i.e., to calculate the maximum and minimum value of each objective. This was done by computing the model with one weight equal to 0 to find the minimum of the other part. To find the maximum, the optimisation problem was run using both weights for each objective:

Optimisation	$OF_1[\$\ COP]$	OF_2
$min\ OF_1$	1107,10	48
$max\ OF_2$	107,76	148

Table 1. Payoff table

Where $OF_1 = \sum_{i \in I} c_i x_i$ and $OF_2 = \sum_{(i,j) \in G} d_{ij} y_{ij}$.

Now, with the payoff table computed, the optimisation problem to build the Pareto front consisted of making one of the objectives as constraints:

$$\max OF_2 - \beta(SV_2/r) \tag{11}$$

$$s.t.$$

$$OF_1 + SV_2 = \varepsilon \tag{12}$$

$(2) - (10)$

Where β is a small number between 10^{-3} and 10^{-6}, SV_2 is a slack variable, r is the range of OF_1 and ϵ is calculated each iteration by:

$$\epsilon = \overline{OF_1} - \delta \tag{13}$$

Being $\delta = 1$ that represents like a step and $\overline{OF_1}$ the maximum value for OF_1. This way the following Pareto front was obtained:

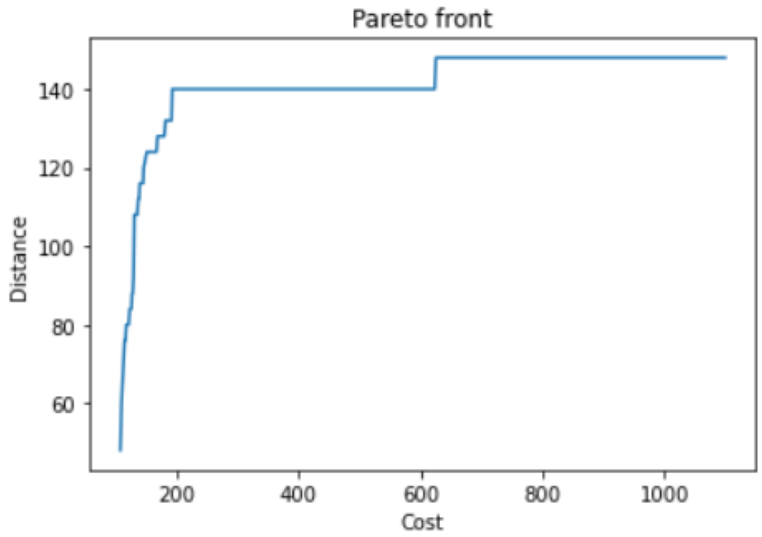


Fig 24. Pareto front

From figure 24 it can be noticed how the two objectives are conflictive between each other. The more distance between seats, we obtain a higher cost. Additionally, it is obvious that from at around the cost 300 the distance stops to increase, this is due because the maximum distance was reached and thus these are dominated points.

Now regarding the decision-making approach, the following functions are proposed:

$$\mu_p^{cost} = \begin{cases} 1 & OF_p^{cost} \leq OF_1 \quad \forall p \\ \dfrac{\overline{OF_1} - OF_p^{cost}}{\overline{OF_1} - OF_{1-}} & OF_{1-} \leq OF_p^{cost} \leq OF_1 \forall p \\ 0 & OF_p^{cost} \geq \overline{OF_1} \ \forall p \end{cases} \tag{14}$$

$$\mu_p^{distance} = \begin{cases} 1 & OF_p^{distance} \leq OF_2 \quad \forall p \\ \dfrac{\overline{OF_2} - OF_p^{distance}}{\overline{OF_2} - OF_{2-}} & OF_{2-} \leq OF_p^{distance} \leq OF_2 \ \forall p \\ 0 & OF_p^{distance} \geq \overline{OF_2} \ \forall p \end{cases} \tag{15}$$

$$\mu_p = \frac{\omega^{cost}\mu_p^{cost} + \omega^{distance}\mu_p^{distance}}{\omega^{cost} + \omega^{distance}} \quad \forall p \tag{16}$$

Where ω represents the weights for each objective and the goal is to find the highest compromise ratio μ_p given arbitrary weights chosen by the decision-maker.

7. Conclusion

We proposed an innovative multi-objective approach to assign seats at check-in. The two main objectives were satisfied in every iteration and the computational time starts to get problematic when there are four or more people in the same booking. The model was simulated using constant number of persons in a booking until the airplane was full and using a real instance given by Viva Air. The model proposed consisted of a network where every node was one seat with a fixed cost and every arc corresponded to the distance between seats, measured using the Manhattan approach and the model was accelerated using callback functions in Gurobi optimser. Finally, we proposed a decision-making approach based on the Pareto front and a fuzzy function.

Having used the technique in Phase II we were able to outperform with the expectations of the model, giving us a quality solution in a short and reasonable time to be applied in the industry. The code used for this project can be found in Github. (Pardo González, 2022)

Future work includes tuning the parameters δ. Also, δ can start with a higher value, but not high enough to significantly increase the computational time and not to bound the selection of the last seats. Additionally, a stochastic model that takes into account the future expected sales of seats would be relevant. Furthermore, it is important to fit a distribution to the number of people in a booking in order to make a more reliable simulation of the system. Finally, we want to accelerate the optimisation model as much as possible, there are many options like the use of heuristics to start with a better initial feasible solution or adding efficient cuts to reduce the computing time.

8. Acknowledgments

Thanks to Professors Alejandra Tabares and David Álvarez for always be willing to solve my doubts and helping me throughout the process.

9. References

Agustín, A., Alonso-Ayuso, A., Escudero, L. F., & Pizarro, C. (2012). On air traffic flow management with rerouting. Part II: Stochastic case. *European Journal of Operational Research*, 167-177.

Agustín, A., Alonso-Ayuso, A., Escudero, L. F., & Pizarro, C. a. (2012). On air traffic flow management with rerouting. Part I: Deterministic case. *European Jounral of Operational Research*, 156-166.

Castro, J., & Fernando, S. (2020). An online optimization-based procedure for the assignment of airplane seats. *Sociedad de Estadística e Investigación Operativa*.

Cordeau, J.-F., Stojkovic, G., Soumis, F., & Desrosiers, J. (2001). Benders Decomposition for Simultaneous Aircraft Routing and Crew Scheduling. *Transportation Science*, 375-388.

Dias de Lima, T., Tabares, A., Baños Arias, N., & Franco, J. F. (2021). Investment & generation costs vs CO2 emissions in the distribution system expansion planning: A multi-objective stochastic programming approach. *Electrical Power and Energy Systems* .

Fonseca i Casas, P., A., J., Angel, & Mas, S. (2013). Using Simulation to compare Aircraft Borading Strategies. *Simulation in Produktion und Logistik*, 237-246.

Mavrotas, G. (2009). Effective implementation of the e-constraint method in Multi-Objective Mathematical Programming problems. *Applied Mathematics and Computation*, 455-465.

Pardo González, G. (2022, 12 14). *Github*. Retrieved from https://github.com/germanrpardo1/Proyecto-Flujo-en-Redes.git

Rothstein, M. (1971). An Airline Overbooking Model. *Transportation Science*, 180-192.

Salari, M., Milne, J. R., Delcea, C., Kattan, L., & Cotfas, L.-A. (2020). Social distancing in airplane seat assignments. *Journal of Air Transport Management* .

Sherali, H. D., Bish, E. K., & Zhu, X. (2006). Airline fleet assignment concepts, models and algorithms. *European Journal of Operational Research*, 1-30.

Tabares, A., Muñoz-Delgado, G., Franco, J. F., Arroyo, J. M., & Contreras, J. (2019). An Enhanced Algebraic Approach for the Analytical Reliability Assessment of Distribution Systems. *IEEE Transactions on Power Systems*.