Airport Gate Assignment

AE4441-16 Operations Optimisation

by

Student Name	Student Number
Aleksandar Shokolarov	4836839

Place: Delft, the Netherlands
Institution: Faculty of Aerospace Engineering, TU Delft
Instructors: Ir. P.C. Roling, A. Bombelli
Date: Sunday 21st April, 2024

TUDelft

Contents

Abbreviations

Abbreviation	Definition
ICAO	International Civil Aviation Organization
MIP	Mixed-Integer Programming
MILP	Mixed-Integer Linear Programming
ATCO	Air Trafic Control Officer
TWD	Total Walking Distance from Gate to Baggage Carousels
TFC	Total Aircraft Fuel Consumption During Taxi Operations
DF	Domestic Flights
IF	International Flights
NB	Narrow Body Aircraft
WB	Wide Body Aircraft

Symbols

Symbol	Definition	Unit
M	Big Constant	[-]

List of Figures

1 Introduction

The air transportation industry has witnessed a remarkable surge in growth over recent years, marked by an exponential increase in the number of passengers and flights traversing the skies. According to The International Civil Aviation Organization (ICAO), the aviation industry is currently responsible for transporting more than 10 million passengers on over 100,000 daily flights[1]. This substantial growth is attributed to various factors, including globalization, technological advancements, and an ever-expanding demand for air travel. As the industry continues to bloom, airports face a critical challenge in efficiently managing their resources to accommodate the escalating influx of flights.

One of the fundamental aspects of the seamless operation of an airport is the assignment of aircraft to gates, which plays a pivotal role in determining the overall efficiency of airport operations. Several factors underscore the paramount importance of optimizing gate assignments in airports. Firstly, effective gate allocation directly impacts operational costs. An inadequately planned assignment can lead to underutilization of gates, requiring unnecessary maintenance and staffing expenses. Moreover, gate assignment optimization is integral to enhancing passenger experience. Efficient gate assignments facilitate streamlined boarding processes, minimizing wait times and congestion.

Formally, the problem of gate assignment can be formulated using the framework of mathematical optimization [2]. This allows for the definition of airport constraints, such as which gate can service what type of aircraft. It also allows Air Traffic Control Officers (ATCOs) to set a clear objective that this assignment process should achieve, for example, minimizing the distance walked by the passengers. Shan et al. [3] utilize a genetic algorithm to minimize the carbon emissions released by the aircraft at an airport, resulting in a more environmentally friendly schedule. Neuman et al. [4] center on modeling the gate allocation issue while considering potential conflicts occurring at taxiways adjacent to gates using a Mixed-Integer Programming (MIP) formulation. This report follows the work created by Cecen et al. [5], which utilize a Mixed-Integer Linear Programming (MILP) method to jointly minimize the total walking distance from gate to baggage carousels (TWD), and the total aircraft fuel consumption during taxi operations (TFC). A few simplifications and additions were made to the initial model to make solving it feasible and expand its applicability to real-life scenarios even further.

The following structure will be used in this paper: Chapter 2 will give an overview of the problem background, detailing how this model could be applied to a real airport, and what the potential benefits could be. Chapter 3 will present the mathematical formulation behind the gate assignment problem and the implementation details. Chapter 4 will present the verification procedures used to assure the correctness of the model, the sensitive analysis conducted to assess its robustness, and the final results obtained from using it on a real test case. Finally, Chapter 5 will present the conclusion to this report, summarizing all important findings.

2

Problem Description

In this chapter, background information about the importance of optimal gate assignment will be presented. Along with that, details about its practical application to one of Europe's biggest airports - Ankara Esenboga, will be given.

2.1. Problem Background

Balancing the objectives of minimizing fuel consumption for aircraft and ensuring easy access to luggage for passengers presents a challenge for air traffic control officers (ATCOs). This challenge arises because some gates may offer fuel-efficient routes but are distant from the baggage reclaim area, and addressing it requires a mathematical model capable of simultaneously optimizing both objectives. Before delving into the mathematical formulation of the model, it is important first to consider how and where such a model could be used.

This report focuses on apron 4 of Ankara Esenboga airport, whose layout is presented in Figure 2.1. According to [6], in 2023, the airport served about 12 million passengers on over 80 thousand flights, out of which 75% were domestic and 25% were international, ranking within the top 50 airports in Europe in terms of passenger traffic. As such, it functions as a central point for domestic flight connections, catering primarily to budget airlines. Since these low-budget airlines aim to minimize costs, balancing between passenger comfort (smaller walking distances) and reduced taxi distances (smaller fuel costs) becomes a problem, which is exactly where a mathematical model could make a difference.

As seen in Figure 2.1, apron 4 of Ankara Esenboga airport provides facilities for 27 gates in total, with 9 open park positions (201-207) and 18 gates at the terminal building (101-120). The airport services both domestic (DF) and international (IF) flights, which are separated between the gates as follows. Domestic flights can utilize gates 106-116 and 201-207, while international flights can be facilitated by gates 117-120 and 201-207. The gates also distinguish aircraft performance categories: narrow-body (NB) and wide-body (WB). The former is assumed to receive service from gates 101–103, 105, 107, 109, 111, 113, 115–118, 120, and 201–207, while the latter can utilize all gates. This difference in categories is reflected mainly in the time spent at the gate after landing, which is set to 81 minutes for NB and 100 minutes for WB as per [5]. The terminal building also houses nine different baggage carousels (0-9). Carousels 7 and 8 are allocated for international flights, while the remaining service domestic flights.

It is important to note here that each gate has two important parameters related to it: its distance to the runway and its distance to each baggage carousel. The distance from the runway to each gate is measured starting from point B as denoted in Figure 2.1. For the open park positions, it starts with $20[m]$ for gate 201 and increases with $20[m]$ for each subsequent gate, concluding with gate 207 at $140[m]$. As for the rest, gate 113 is positioned closest to point B with a distance

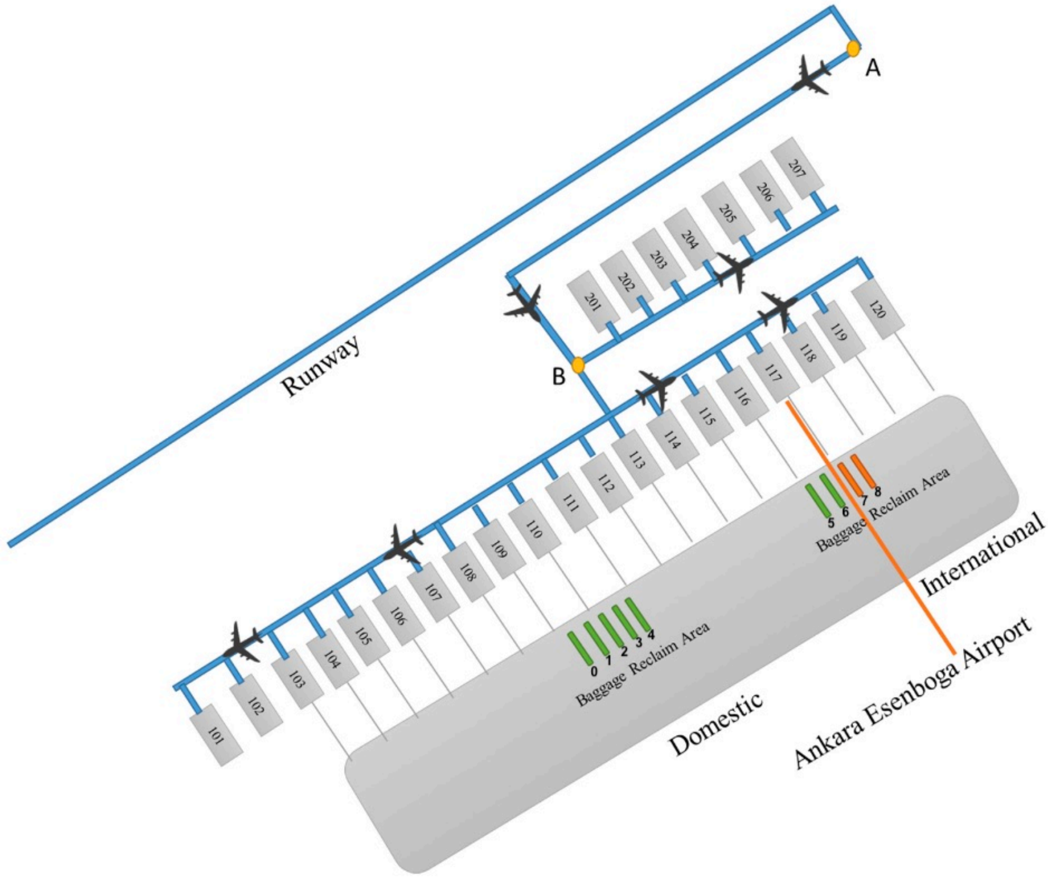


Figure 2.1: Gate layout of apron 4 at Asenboga Aiport.

of 40[m], while the rest are evenly spaced around it with a 20[m] separation, leaving gate 101 at 280[m] and gate 120 at 180[m]. Considering the distances from the gates to each baggage carousel, they are computed in a similar fashion but are omitted for the sake of brevity.

3 Methodology

In this chapter, an introduction to the mathematical model used to solve the gate assignment problem will be given. The mathematical model in this report is based on the work by Cecen et al. [5], with some extra additions, as will be explained later. Finally, this chapter will conclude with a section elaborating on the details of the implementation.

3.1. Mathematical Formulation

The gate assignment problem aims to map a set of flights to a set of gates and baggage carousels. These sets can be formally described as follows:

$$\mathcal{I} \coloneqq \{1, ..., \alpha\} \tag{3.1}$$
$$\mathcal{J} \coloneqq \{1, ..., \beta\} \tag{3.2}$$
$$\mathcal{K} \coloneqq \{1, ..., \gamma\} \tag{3.3}$$

where α, β, and γ are the total number of flights, gates, and baggage carousels respectively. There are a few associated parameters for each flight, gate, and baggage carousel. Starting with the flights, these parameters are:

1. g_i: expected arrival time for flight i.
2. p_i: performance category of flight i, which takes a value of 0 for NB and 1 for WB.
3. o_i: operational type of flight i, which takes a value of 0 for DF and 1 for IF.
4. v: aircraft taxi speed, which is constant for all flights.
5. y_{p_i}: fuel consumption per second for flight i, which is based on its performance category p_i.
6. gt_{p_i}: average gate occupancy time for aircraft i, which is based on its performance category p_i.
7. tb: buffer time that accounts for flight delays and is constant for all flights. Total buffer time is split equally between arrival and departure time of a flight.

For the different gates, the parameters associated are:

1. td_j: distance from gate j to the runway.
2. $ga_{j_{p_i}}$: binary value that is set to 1 if gate j can service aircraft i of performance category p_i, otherwise it is 0.
3. $fa_{j_{o_i}}$: binary value that is set to 1 if gate j can service aircraft i of operational type o_i, otherwise it is 0.

Finally, the baggage carousels contain the following parameters:

1. kd_{jk}: binary value that is set to 1 if baggage carousel k can service gate j, otherwise it is 0.
2. bd_{jk}: distance from baggage carousel k to gate j.

Now, more important to the mathematical formulation are the decision variables, which will be optimized to deliver a solution to the problem. These decision variables are:

1. q_{ij}: time at which aircraft i enters gate j.
2. x_{ijk}: binary variable that is set to 1 if aircraft i is assigned to gate j and baggage carousel k and is set to 0 otherwise.
3. f_i: fuel consumption of aircraft i during taxi.
4. d_i: distance from gate to baggage carousel for flight i.

Having listed them, a set of constraints defining the problem's boundaries is also needed. Most naturally, the first constraint that can be defined is:

$$\sum_{j=1}^{\beta}\sum_{k=1}^{\gamma} x_{ijk} = 1 \quad \forall i \tag{3.4}$$

which limits each flight to be allocated to only one gate and only one carousel.

$$x_{ijk} \leq ga_{j_{p_i}} \quad \forall i, j, k \tag{3.5}$$

$$x_{ijk} \leq fa_{j_{o_i}} \quad \forall i, j, k \tag{3.6}$$

Equation 3.5 and Equation 3.6 limit that flights can only be assigned to gates that can service them based on their operational type and performance category.

$$x_{ijk} \leq kd_{jk} \quad \forall i, j, k \tag{3.7}$$

Equation 3.7 assures that flights can reach their baggage carousel using their assigned gate.

$$q_{ij} = g_i - \frac{tb}{2} + \sum_k x_{ijk} \cdot \frac{td_j}{v} \quad \forall i, j \tag{3.8}$$

Equation 3.8 is used to compute the gate entrance time of aircraft i for gate j, taking into consideration the flight's arrival time and the taxi distance. Compared to the original model by [5], here, an extra buffer time, tb, that accounts for delays is added since real-world flights usually do not arrive and depart as expected. This buffer time is split equally and added to each flight's arrival and departure times.

$$q_{i_2 j} - q_{i_1 j} \geq gt_{p_i} - tb - \left(2 - x_{i_1 j k_1} - x_{i_2 j k_2}\right) M \quad \forall i_1, i_2, j, k_1, k_2 \mid i_1 < i_2 \tag{3.9}$$

Equation 3.9 enforces that two flights cannot be allocated to the same gate during the same timeframe, with the added extra buffer time. M, in this case, is a big constant.

$$f_i = \sum_j \sum_x x_{ijk} \cdot \left(\frac{td_j}{v}\right) \cdot y_{p_i} \quad \forall i \tag{3.10}$$

$$d_i = \sum_i \sum_k x_{ijk} \cdot bd_{jk} \quad \forall i \tag{3.11}$$

Equation 3.10 and Equation 3.11 are used to compute the total fuel consumption of aircraft i and the walking distance from the gate to the baggage carousel.

Having defined the constraints, the final step is devising an objective function, which will define the goal of the gate assignment procedure. The most obvious choices for that are Equation 3.12 and Equation 3.13, which respectively aim to minimize the fuel spent during the taxi maneuver and the total walking distance for the passengers.

$$\sum_{i=1}^{n} f_i \tag{3.12}$$

$$\sum_{i=1}^{n} d_i \tag{3.13}$$

However, minimizing both of these functions separately would invariably lead to undesired results. For example, minimizing the fuel spent on its own would reduce the cost for the airliner company, but that would simultaneously increase the distance between the assigned gate and baggage carousel, leading to a negative experience for the passengers. Therefore, striking a balance between both objectives is the ultimate goal. On the surface, this could be achieved by minimizing the weighted sum of both Equation 3.12 and Equation 3.13, but due to the difference in units, in the current formulation, it is not possible. Therefore, [5] propose scaling both terms by the so-called ”nadir” points to remove this unit difference as in:

$$w \cdot \left(\sum_i f_i \right) \cdot (f_{\max})^{-1} + (1 - w) \cdot \left(\sum_i d_i \right) \cdot (d_{\max})^{-1} \tag{3.14}$$

To calculate their values, four separate optimization problems need to be solved.

Firstly, the minimum total walking distance from gate to baggage carousel for all flights - d_{min} - needs to be calculated. This is done by simply optimizing the problem using Equation 3.12 as an objective function. Following that, an optimization problem for the minimum total fuel consumption for all flights - f_{min} - needs to be solved. That again is done by using Equation 3.13 as an objective function. Having obtained these two values, the nadir points d_{max} and f_{max} can be calculated. In order to solve for d_{max}, Equation 3.12 is optimized with an additional constraint of the form:

$$\sum_{i=1}^{n} f_i \leq f_{min} \tag{3.15}$$

This essentially poses the question of what the total walking distance for the minimum fuel consumption is, which is exactly what d_{max} is. As for f_{max}, Equation 3.13 is used as an objective, again with an additional constraint of the form:

$$\sum_{i=1}^{n} d_i \leq d_{min} \tag{3.16}$$

The intuition behind this is that f_{max} would represent the total fuel consumption for the minimum total walking distance. With these values at hand, the original gate assignment problem can be solved.

3.2. Implementation Details

In order to solve all the optimization problems mentioned, the mathematical models and utility functions behind them were implemented in Python, utilizing the Gurobi[7] optimization package on a Macbook Pro with an M1 chip.

To test the model, sets of different flight schedules were created using a dedicated function in the $data\_processing.py$ file. The flight arrival times were computed by sampling a uniform distribution set between a start and an end time of the day. The fuel consumption and taxi speed were set as per [5]. The operational type for each flight was decided by sampling a binomial distribution biased towards domestic flights with an 80% probability as per [5]. The performance category was again chosen by sampling a binomial distribution biased towards narrow-body aircraft with a probability of 90% as per [5]. All the data regarding flights, gates, and baggage carousels is stored in a single Excel file named $data.xlsx$. The file $model.py$ defines the optimization model and solves all of the problems in this report. The plotting utilities are also defined in that file. $utils.py$ includes the definitions of some helper functions. Finally, all of the parameters governing the problem, such as the number of flights, are defined in $config.yaml$, which is first parsed by $data\_processing.py$, generating the flight set, and then by $model.py$ to extract the model parameters.

4

Verification, Sensitivity Analysis and Results

In this chapter, the procedures that were used to verify the mathematical model will be presented. After that, a sensitive analysis that assesses the robustness of the model will be presented. Having done so, the results obtained from running the model on a realistic full-scale scenario will be shown.

4.1. Verification Procedures

The goal of the verification procedures that will be presented in this chapter is to confirm the functionality of the model and the correctness of its implementation. To do so, firstly, the validity of the constraints will be assessed. After that, manufactured problems with known solutions will be used to confirm that the model optimizes what it is meant to. To simplify these procedures, a schedule consisting of only 10 flights with arrival times spread between 9:00 and 11:00 with a buffer time of 15 minutes was generated and used as the test case.

Constraint Correctness

The constraints in the case of this gate assignment problem mostly prohibit non-physical behavior, such as a flight occupying two gates at the same time, but also enforce airport requirements in terms of which gate can service which aircraft. To verify these constraints, the model was run on the test flight schedule with no objective function, which removes the need for optimality and solely focuses on the enforcement of the constraints. Unfortunately, this does not allow us to check for the correctness of Equation 3.10, Equation 3.11, Equation 3.15, Equation 3.16, which means that additional test cases will be needed to verify them.

Running the aforementioned test case resulted in the following flight schedule, as seen in Figure 4.1, allowing us to check the constraints individually. Before doing so, it would be useful to explain all the numbers in the flight schedule. Firstly, on the x-axis, the time of arrival is set. On the left y-axis, the gate number is presented, while on the right y-axis, the performance type and aircraft category that this gate can service are shown. As for the flights, the flight number, as well as the performance type and operational category of the aircraft, are presented. The gray area surrounding each flight represents the added buffer time.

To verify Equation 3.4, it suffices to note that there are no two flights with the same number present in the schedule. Equation 3.5 and Equation 3.6 are also maintained, which can be noted by comparing the flight type and category with the allowed types and categories for each gate on the right y-axis.

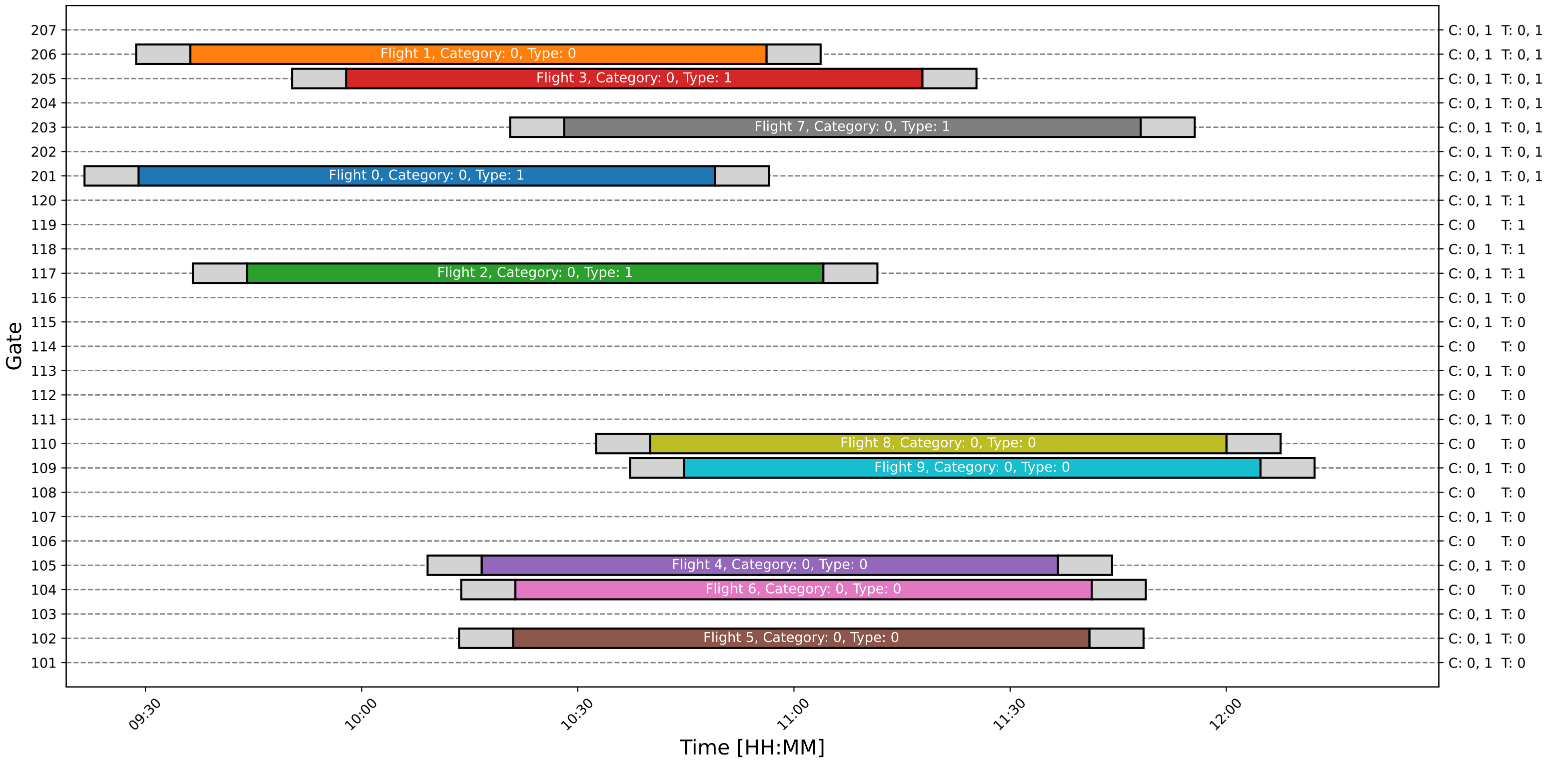


Figure 4.1: Resulting flight schedule after optimizing the first verification case.

Equation 3.7 can be verified by first introducing Figure 4.2, which represents the number of allocations per each carousel. Looking back at Figure 4.1, the only flight allocated to an international gate is flight 2, which is delegated to gate 117. Therefore, the expected number of allocations for the international carousels 6, 7, and 8 is one, which is indeed the case, verifying the constraint.

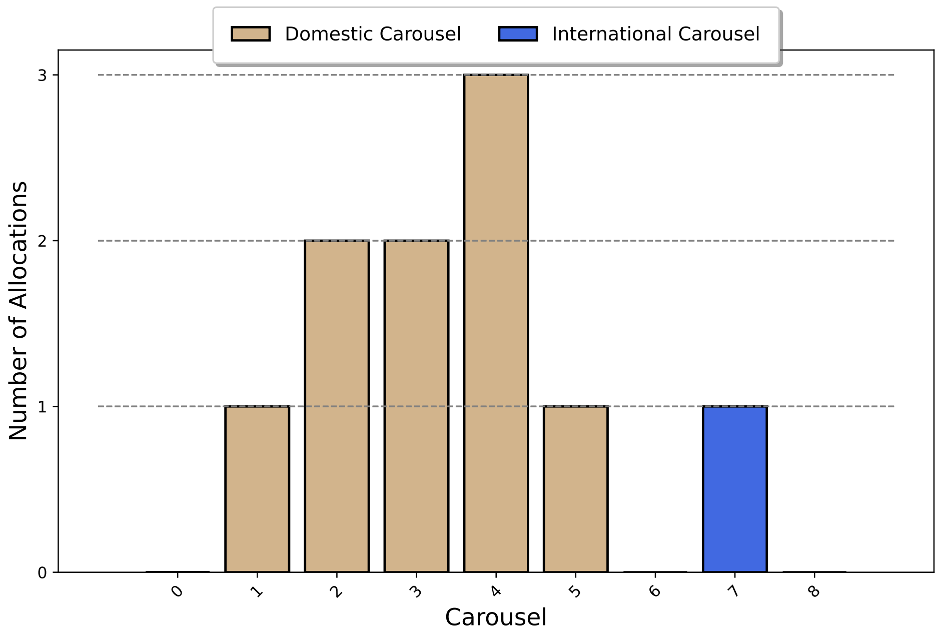


Figure 4.2: Number of allocations per carousel for the specific flight schedule.

To check the correctness of Equation 3.8 and Equation 3.9, referring to Figure 4.1 is again the way to go, showing that no two flights are allocated to the same gate in the same timeframe.

Minimum Walking Distance

To test the first objective, which aims at minimizing the walking distance of the passengers, the same test case as previously is used along with the objective function described in Equation 3.12. This also allows for the verification of Equation 3.11, which enforces that the total walking distance is the sum of the distances between the gate and the carousel for each individual flight. The expected result of this procedure is that most flights will be mapped to gates that are the closest to the terminal building and the carousels while still adhering to all constraints. Instead of presenting the flight schedule as done previously, to verify this case, Figure 4.3 is shown, which presents the number of flight allocations (on the y-axis) per gate (on the x-axis).

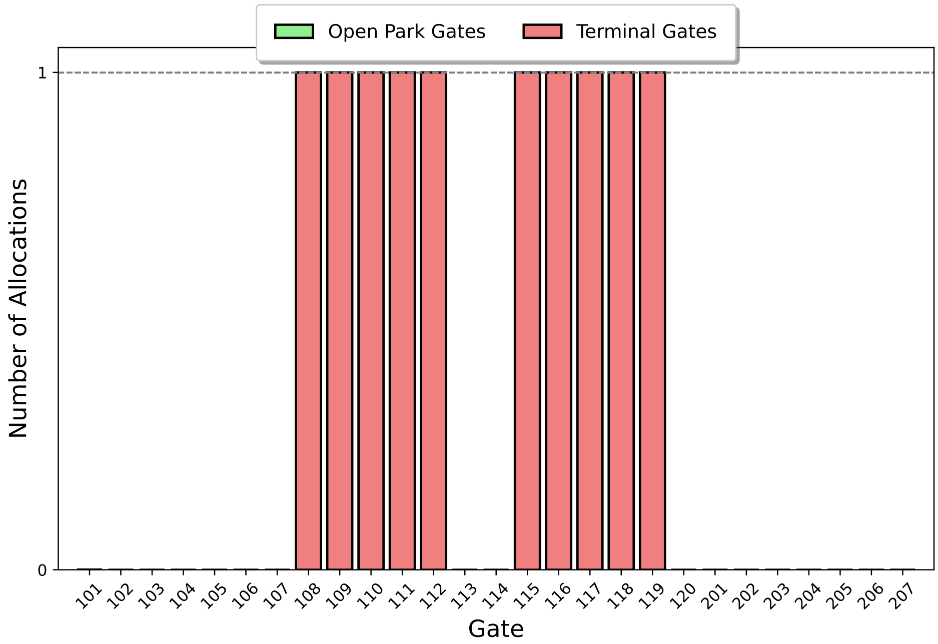


Figure 4.3: Number of flight allocations per gate for verification case 2.

Surely, as expected, all of the flights are allocated to the gates closest to the terminal building - gates 108-119, with 0 flights getting allocated to the open park gates, which are positioned furthest away. This test ensures that the objective function, as well as the constraint used to compute the total walking distance, work as intended.

Minimum Fuel Consumption

Similarly to the previous case, to verify that the minimum fuel consumption objective function and the constraint that is used to compute the total fuel consumption work, Equation 3.13 is set as an objective function with Equation 3.11 set as an additional constraint. The expected outcome in this case is that most flights will be mapped to the open park gates, which are positioned closest to the runaway, thus saving the most fuel and incurred costs.

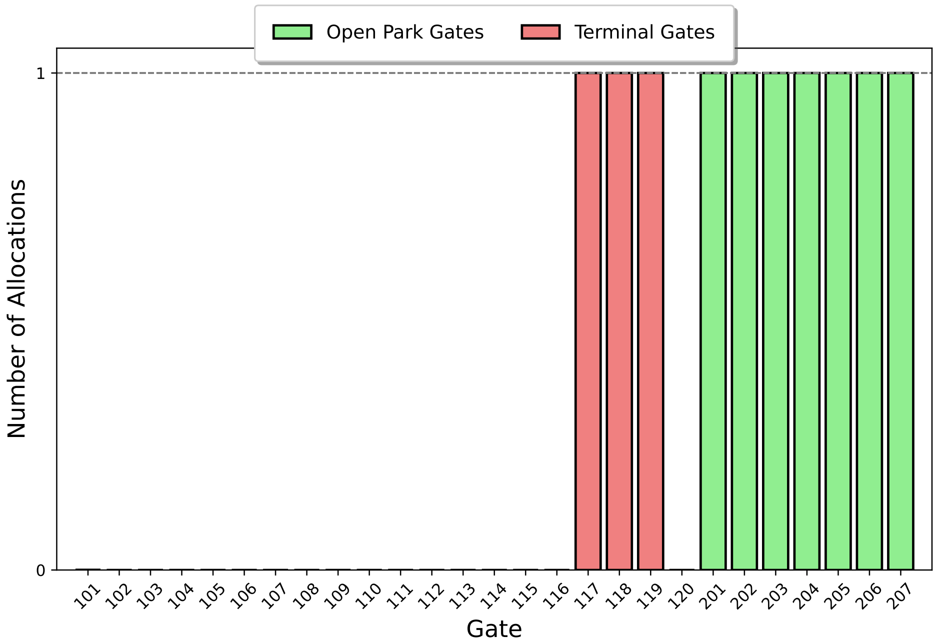


Figure 4.4: Number of flight allocations per gate for verification case 3.

Looking at Figure 4.4, it can be seen that this is exactly the effect that occurs with most flights being placed at the open gates 201-207 with the remaining flights being placed on some of the closest to the runway terminal gates thus verifying Equation 3.13 and Equation 3.10.

Ideal Point Calculation

The only remaining constraints that still need to be verified are Equation 3.15 and Equation 3.16, which are used to calculate the "nadir" points used during the normalization process explained in Chapter 3. The verification process this time was done purely numerically. In order to do so, firstly, the minimum walking distance and fuel consumption problems were solved to obtain d_{min} and f_{min} as described in Chapter 3. These values were set as constraints in Equation 3.15 and Equation 3.16, and the two problems were optimized anew. After that, the total fuel consumption and walking distances were computed and compared to the values of f_{min} and d_{min}, showing that they indeed had lower values and thus, satisfying the constraints.

4.2. Sensitivity Analysis

With that, all of the constraints and objective functions that define the optimization model were verified, making the model valid for real-problem usage. But before doing so, it is also important to investigate how sensitive the model is to the parameters that define it. During real-world operations, the values of such parameters are not clearly defined and can vary greatly. Therefore, having a model that is robust to such variations is of great importance.

To test the sensitivity of the model, a new test case consisting of 20 flights spread between 9:00 and 11:00 was considered. The number of flights was increased in order to make the problem more realistic but still computationally feasible to solve in a short time span.

Buffer Time

The first parameter that was investigated is the buffer time that is added in order to compensate for any delays in the arrival or departure of each flight as described in Equation 3.9. As an objective function, Equation 3.14 was used, which tries to jointly minimize the total distance walked by the passengers as well as the total fuel consumption, with an equal weight set for both contributions.

Three different buffer times of 0, 30, and 60 minutes were used to conduct the sensitivity analysis. These buffer times were equally split between the arrival and departure times as mentioned in Chapter 3. The problems were then re-optimized, and the value of the objective function was tracked. This progression is more clearly seen in Figure 4.5.

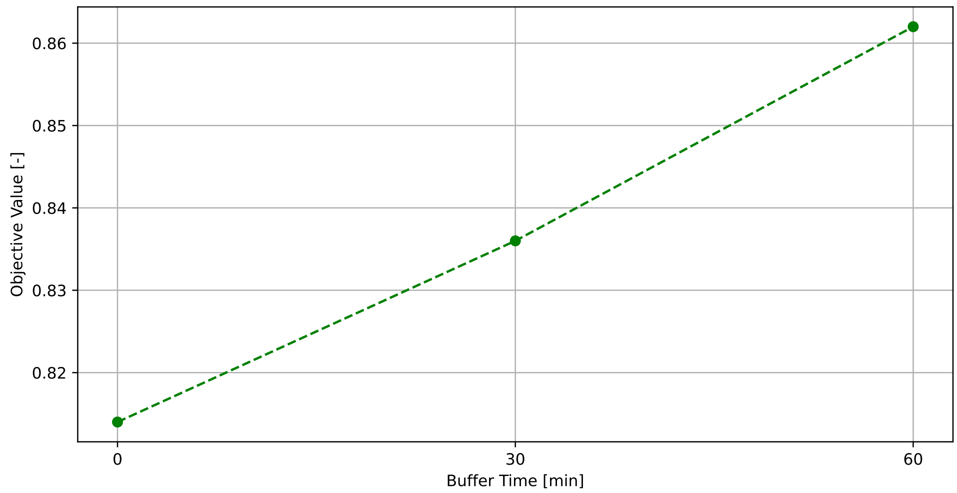


Figure 4.5: Value of the objective function plotted against different buffer times.

An almost linear progression can be seen, indicating that the increasing buffer time causes the model to converge to ”worse” solutions. Indeed, that is expected since, before trying to optimize the solution, the model first needs to adhere to the constraints set. By increasing the buffer time, these constraints get tighter and tighter, forcing the model to find new solutions that still obey them. Still, only a 5% difference in the objective value is observed from using no buffer time to using a buffer time of 60 minutes, which is a minimal change, indicating that the model possesses robustness against big changes in the buffer time.

Taxi Speed

An important parameter that directly impacts the fuel consumption of aircraft during taxi is their taxi speed, which is used in Equation 3.8 and Equation 3.10. Therefore, analyzing how this parameter affects the minimization problem could give potential insights on how airliners could minimize their operating costs on-ground. To do so, the gate assignment problem was solved for three different taxi speeds - 4, 8, and 12 [m/s]. For each of these speeds, the normalization procedure had to be done anew in order to compute the nadir values, making the total number of problems that needed to be solved 12.

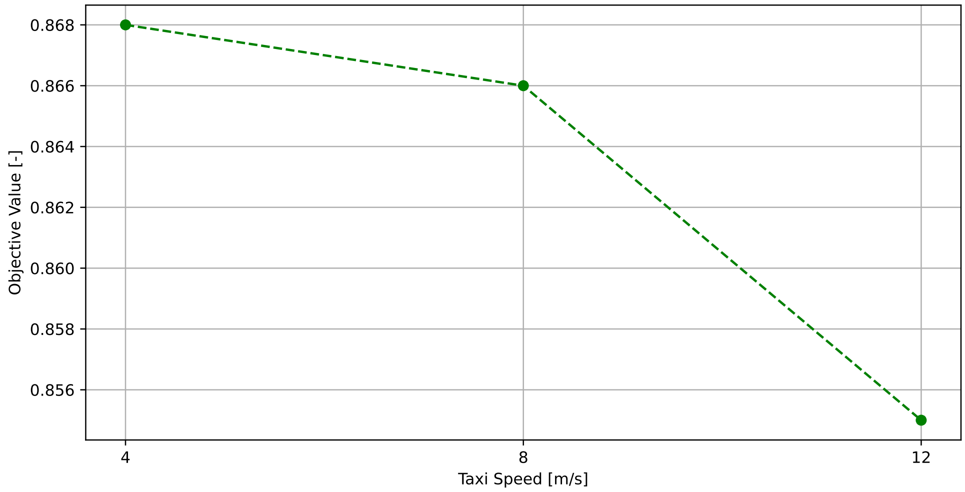


Figure 4.6: Value of the objective function plotted against taxi speeds.

Figure 4.6 showcases the variation of the objective function with respect to the chosen taxi speed. Immediately, a negative correlation can be observed, meaning that as the taxi speed increases, the value of the objective function decreases. This can be explained by the fact that, with higher taxi speeds, less time is spent on the runway and, consequently, less fuel. Yet, the relative change in the objective is less than 1% when comparing a taxi speed of 4 [m/s] and 12 [m/s], affirming the model's robustness to change in taxi speed.

Objective Weight Factor

The final parameter that was investigated is the weighting factor in Equation 3.14, which influences how the total walking distance and the total fuel consumption influence the value of the objective.

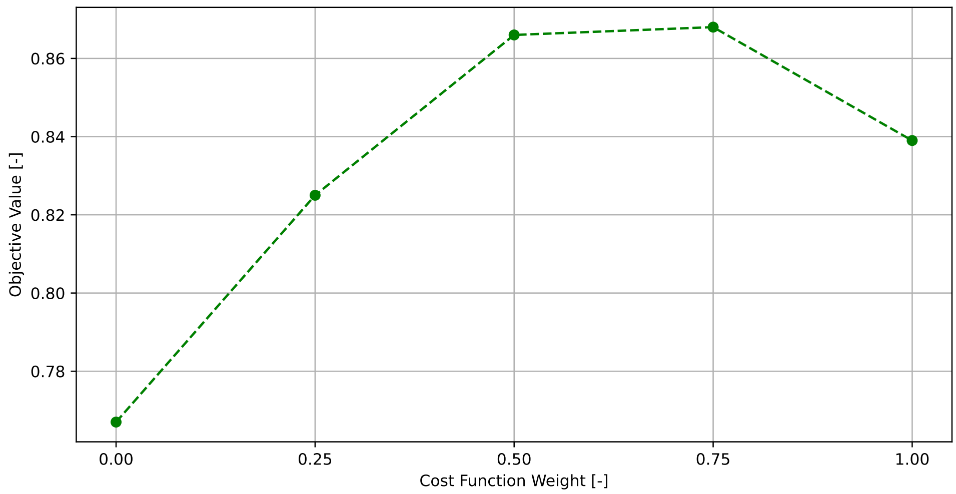


Figure 4.7: Value of the objective function plotted against different weight factors.

What Figure 4.7 shows is the effect of trying to jointly minimize two objectives simultaneously. For values of $w = 0$ and $w = 1$, the model is able to reach minima for the total walking distance and the total fuel consumption, respectively. However, when trying to minimize both functions at the same time, the cost function slowly increases, peaking in the middle, which signifies that in order to keep a balance between the two objectives, a higher cost has to be paid, which is again expected.

4.3. Final Results

Having verified the constraints of the model and performed a sensitivity analysis to analyze the importance of the individual parameters in the model, it can finally be used to solve real-world problems. For this case, a set of 25 flights that were spread over a 3-hour period was considered with an added buffer time of 30 minutes. The weight term in the cost function was set to 0.5, placing equal importance on both the fuel consumption and passenger walking distance. This resulted in the following flight schedule as presented in Figure 4.8 as well as gate and carousel allocation presented in Figure 4.9(a) and Figure 4.9(b) respectively.

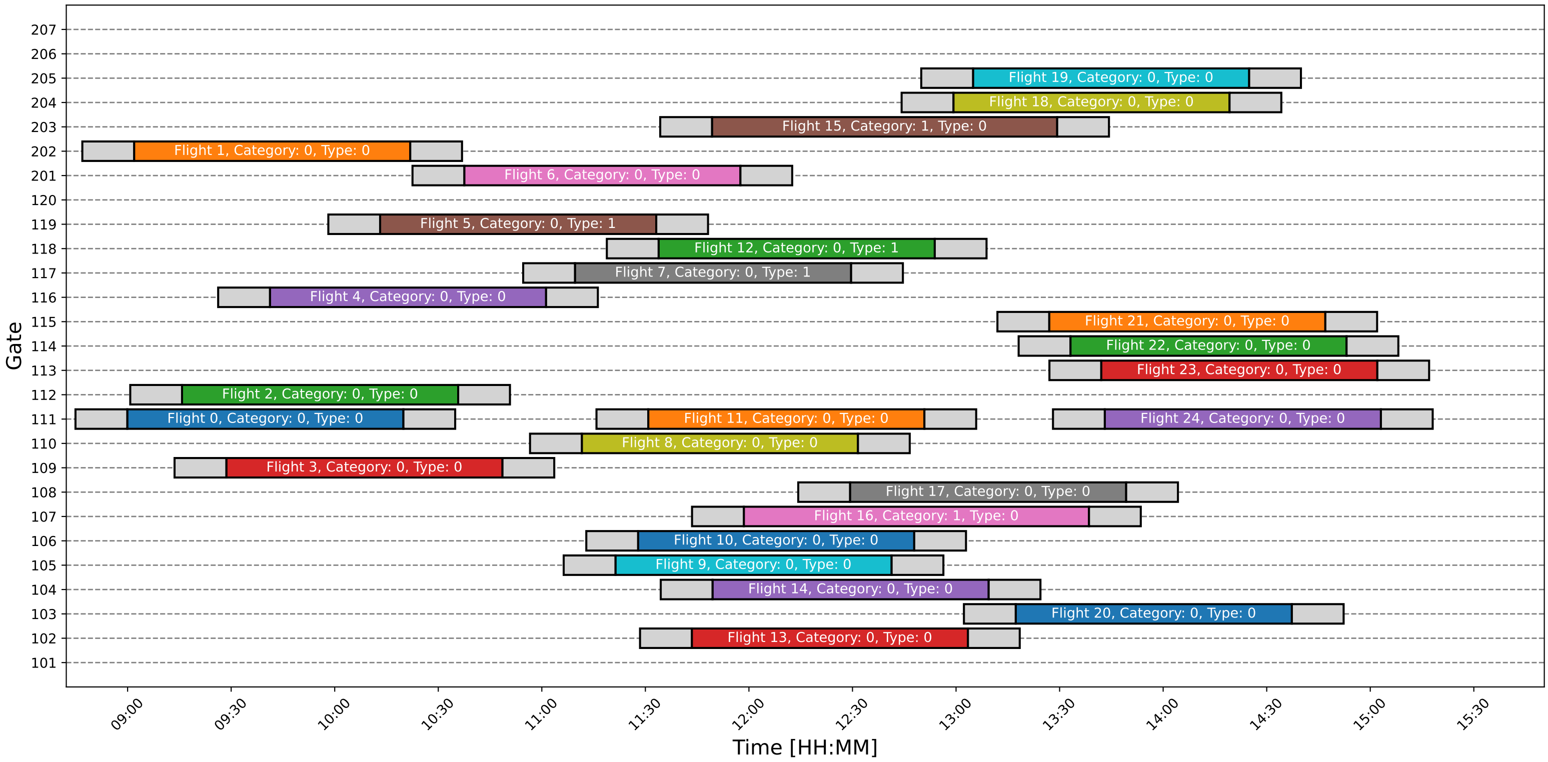
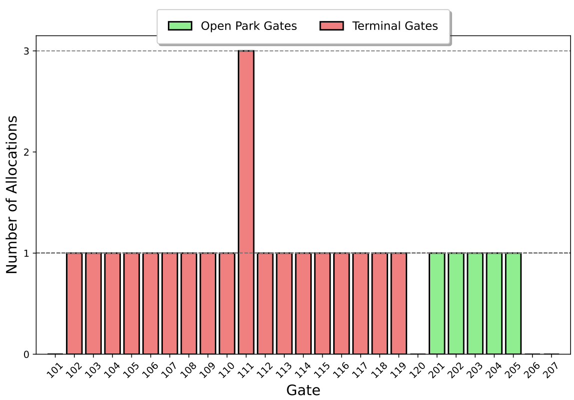
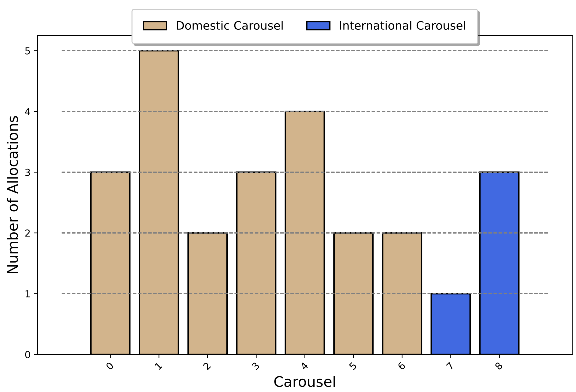


Figure 4.8: Optimized flight schedule for the real-world test case.



((a)) Optimized gate allocation for the real-world test case.



((b)) Optimized carousel allocation for the real-world test case.

5

Conclusion

With the never-ending growth of the air transport sector, efficiently allocating gates at airports becomes more and more important. In this report, the MILP model developed [5], which aims at jointly minimizing the total fuel consumption of aircraft as well as the total walking distance of passengers, was implemented and verified. To do so, apron 4 of Ankara Esenboga was chosen as a test case with a simulated but realistic flight schedule. The verification process ensured that all the constraints and objectives functions were correctly working. A sensitivity analysis was also carried out to assess the robustness of the model to some of its parameters since, in real-world conditions, these parameters can vary greatly. In this case, the three parameters that were chosen were - the arrival/departure buffer time, the aircraft taxi speed, and the weight factor in the cost function. It was concluded that none of these parameters had a substantial impact on the model results when undergoing a change, which established the robustness of the model. Finally, a real-world test case consisting of 25 flights was performed, showcasing the usefulness that the model could present to airports.

Bibliography

[1] ICAO, "Future of Aviation." `https://www.icao.int/Meetings/FutureOfAviation/Pages/default.aspx`.

[2] A. Bouras, M. A. Ghaleb, U. S. Suryahatmaja, and A. M. Salem, "The Airport Gate Assignment Problem: A Survey," *The Scientific World Journal*, vol. 2014, p. 923859, Nov. 2014.

[3] Y. Shan, Y. Shao, Q. Yuan, and Y. Jiang, "Multiobjective Gate Assignment Model Considering Carbon Emissions," *International Journal of Environmental Research and Public Health*, vol. 20, p. 3952, Feb. 2023.

[4] U. M. Neuman and J. A. D. Atkin, "Airport Gate Assignment Considering Ground Movement," in *Computational Logistics* (D. Pacino, S. Voß, and R. M. Jensen, eds.), (Berlin, Heidelberg), pp. 184–198, Springer, 2013.

[5] R. K. Cecen, "Multi-objective optimization model for airport gate assignment problem," *Aircraft Engineering and Aerospace Technology*, vol. 93, pp. 311–318, Jan. 2021. Publisher: Emerald Publishing Limited.

[6] I. Esb, I. Ltac, AJet, and P. Airlines, "Wikiwand - Ankara Esenboğa Airport."

[7] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2023.