33rd European Modeling & Simulation Symposium
18th International Multidisciplinary Modeling & Simulation Multiconference

ISSN 2724-0029 ISBN 978-88-85741-57-7 © 2021 The Authors.
doi: 10.46354/i3m.2021.emss.024

Opt-Sim approach for the gate allocation problem in covid-19 times

Paolo Scala[1,*], Margarita Bagamanova[1] and Miguel Mujica Mota[2]

[1]Amsterdam School of International Business – Amsterdam University of Applied Sciences, Fraijlemaborg 133, Amsterdam, 1102CV, The Netherlands
[2]Aviation Academy – Amsterdam University of Applied Sciences, Weesperzijde 190, Amsterdam, 1097 DZ, The Netherlands

[*]Corresponding author. Email address: p.m.scala@hva.nl

Abstract

This study tackles the gate allocation problem (GAP) at the airport terminal, considering the current covid-19 pandemic restrictions. The GAP has been extensively studied by the research community in the last decades, as it represents a critical factor that determines an airport's capacity. Currently, the airport passenger terminal operations have been redesigned to be aligned and respect the covid-19 regulation worldwide. This provides operators with new challenges on how to handle the passengers inside the terminal. The purpose of this study is to come up with an efficient gate allocator that considers potential issues derived by the current pandemic, i.e., avoid overcrowded areas. A sim-opt approach has been developed where an evolutionary algorithm (EA) is used in combination with a dynamic passenger flow simulation model to find a feasible solution. The EA aims to find a (sub)optimal solution for the GAP, while the simulation model evaluates its efficiency and feasibility in a real-life scenario. To evaluate the potential of the Opt-Sim approach, it has been applied to a real airport case study.

Keywords: simulation, optimization, gate allocation problem, covid-19

1. Introduction

The covid-19 pandemic has hugely impacted the aviation industry. The latest traffic figures show that the air traffic in March 2021 is down 65% compared to the traffic in the same month of 2019 (pre-covid-19) (EUROCONTROL, 2021). The pandemic has brought new habits to people, see social distancing, and face masks, affecting our lives. The behavior of people in an airport terminal is affected by the previously mentioned measures as well. A social distancing of 1.5 (2 m in some countries) (IATA, 2020; FAA, 2020)) has already cut down the capacity of existing facilities.

Moreover, airport operators had to adapt to the new situation with the terminal design, as previously used areas for passengers' comfort are now used for conducting covid-19 tests. New procedures have been implemented from an operational level, such as checkpoints for controlling the temperature and checking required documentation (e.g., negative covid-19 test certificate), new boarding procedures (EASA, 2020; IATA, 2020; TSA, 2021), and others. Although the traffic demand has gone down, existing facilities at airports have been used differently, which brought new challenges for airport operators as they need to make sure that the existing traffic of passengers will experience a comfortable and, above all, safe stay within the terminals.

In this context, it becomes clear that the gate allocation will affect the passenger flow within the airport terminal, as it will drive the passenger position within the airport, affecting the density in the different areas of the terminal. The gate allocation problem (GAP) is defined as allocating a flight to a specific gate at a specific time, considering constraints such as aircraft size, passenger walking distance, and many others (Daş et al., 2020). The researchers extensively studied this problem by applying different techniques and focusing on different objectives (Guépet et al., 2015; Dorndorf et al., 2008).

The purpose of this paper is to develop a gate allocation solution considering the covid-19 situation. Covid-19 is considered to avoid allocating gates close to each other and gathering many people from different flights in nearby areas. In the developed gate allocation framework, given a set of scheduled flights, it is intended to reallocate the flight assignment only to avoid passenger crowds and, consequently, the risk of infection. This scenario can be named covid-19-friendly gate allocation.

This paper's objective has been pursued by implementing an Opt-Sim approach (Scala et al., 2021), where an EA provides a (sub)optimal solution to the Gate Allocation Problem (GAP), and a dynamic passenger flow simulation model evaluates the efficiency and feasibility of the given solution in a real-life environment. The main contributions of the paper are twofold: from the operational point of view, considering covid restrictions and obtaining a covid-friendly solution to the GAP; from a methodological point of view, considering the gate position as a constraint in the EA, and applying a sim-opt approach to the problem mentioned above.

2. Opt-Sim approach

Simulation combined with optimization is an approach that has been applied in many different areas (Mujica and Flores, 2017). The main advantage brought by this approach is that these techniques complement each other in the sense that one compensates for the other's limitations. Optimization and various metaheuristics are very good at finding a (sub)optimal solution in a short time. However, still, they do not consider the stochastic nature of the problem under study. On the other hand, simulation can recreate a virtual environment where the variability of the system is considered, and therefore, is an excellent tool for evaluating the feasibility of an optimal solution in a close-to-reality scenario. One of the main drawbacks of the Opt-Sim approach is the computational time, an issue to be solved in the future (Scala et al., 2021). The Opt-Sim approach is shown in **Figure 1**.

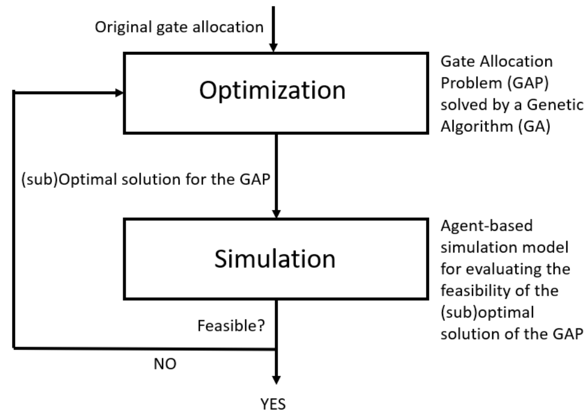

Figure 1. Opt-Sim approach implemented for the GAP

The following subsections describe both the optimization and the simulation components of the Opt-Sim approach developed in this study.

2.1. Evolutionary algorithm (EA) for solving the gate allocation problem (GAP)

This section briefly describes an evolutionary algorithm that was used to solve the specified GAP. The algorithm previously used in (Bagamanova and Mujica Mota, 2020a, 2020b) has been adapted to tackle the problem of gate assignment during the covid pandemic in this study. Evolutionary algorithms have been successfully applied to many air transport problems (Ghazouani et al., 2015; Mujica Mota, 2015; Abdelghany et al., 2017).

In this paper, the gate assignment schedule is coded as an *FxC* dimensional array, where *F* refers to the number of flights to be assigned to the gates and *C* is the number of various flight characteristics relevant for the assignment. Each cell (flight) has an array of characteristics that are considered by the problem's constraints. A complete chromosome that represents a potential solution is illustrated in **Figure 2**.

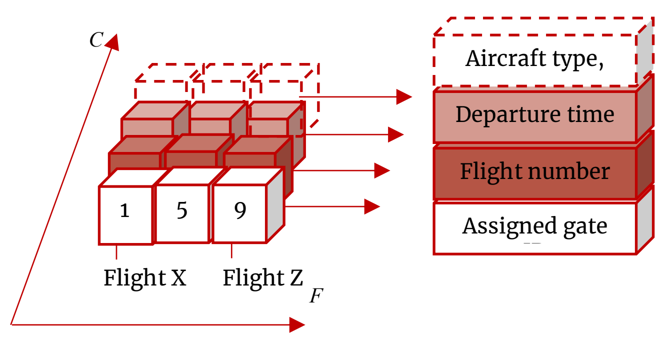

Figure 2. Gate assignment schedule coded for the optimization algorithm

Gate assignment algorithm uses target flight schedule, list of available gates, and their technical and operational characteristics as input data for generation of allocations. Then, the generated assignments are optimized with an evolutionary algorithm according to the following objective function:

$$\text{minimize} \rightarrow R_{hold} + R_{tabu} \tag{1}$$

This bi-objective function consists of the following objectives:

- Objective to minimize the number of unassigned flights:

$$R_{hold} = N_{hold} / N_{flights} \tag{2}$$

- Objective to minimize the number of flights assigned to the neighboring gates and therefore to minimize the density of passengers in the departure lounge:

$$R_{tabu} = N_{tabu} / N_{fligths} \tag{3}$$

Where:

- N_{hold} is the number of unassigned flights;
- $N_{flights}$ is the total number of flights in the schedule;
- N_{tabu} is the number of flights assigned to the taboo-listed gates at the same time.

The different operations and selection of the different chromosomes of the algorithm are presented in Pseudocode 1.

The general flow of the algorithm starts with importing the target flight schedule, terminal gate characteristics, and a list of gates that are not preferred to have simultaneously assigned flights (*Gate_Taboo_List*). In this paper, the assignment considered the size of aircraft and the airline preferences of assigning to a specific terminal module. An initial gate assignment solution (referred to as *Adam chromosome* in Pseudocode 1) is created with these data. This *Adam chromosome* is then copied and randomly modified by changing assigned gates into different ones. After that, the quality of the generated solutions (chromosomes) is evaluated by the objective function (1) for each of the chromosomes; the one with the smallest value is saved and marked as the best chromosome. Next, some chromosomes are subjected to random *Crossover* (where two randomly selected chromosomes exchange their assigned gates). As the next step, some chromosomes are subjected to *Mutation*. The chance of mutation is calculated randomly for each chromosome, and in case it is bigger than 10%, the chromosome gets mutated (assigned gates are shuffled within a chromosome). After that, all chromosomes are re-evaluated on their value of the objective function, and the best chromosome marking is updated if needed. This is followed by evaluating if the algorithm has reached the stopping criteria defined by the user. If so, the algorithm stops and exports the best solution into the data file.

```
GET Stop_Criteria
IMPORT
   Flight_Schedule,
   Constraints,
   Gate_Taboo_List
CREATE
   Adam chromosome, A
GENERATE
   Set(chromosomes), S = RandomChange(A)
WHILE CurrentSituation < > Stop_Criteria
   REPEAT
      FOREACH X IN S
       DO
         Calculate objective function F(X)
         IF value F(X) > Best_Value
          THEN
            Best_Value = value F(X)
            Best_Chromosome = X
         DO Crossover(Xi, Xj)
         IF MutationChance > 0.1
            Mutation (X)
EXPORT Best_Chromosome
```

Pseudocode 1. Evolutionary algorithm

Different stopping criteria can be defined, such as the number of iterations, total running time, or specific objective function value. For this paper, the total running time of 30 minutes was defined as a stopping criteria.

2.2. Dynamic passenger flow simulation model of the airport terminal operations

The simulation model is built based on a dynamic passenger flow simulation software (ARC, 2020), which can recreate the entire airport terminal, including the layout, processes, and the passengers' behavior (Mujica et al., 2020).

The airport terminal is the area that facilitates passengers in their journey. The processes to be undergone within an airport terminal differ depending on passengers, i.e., arriving, departing, and transferring. **Figure 3** depicts the departure passenger flows, as this will be the focus of this paper.

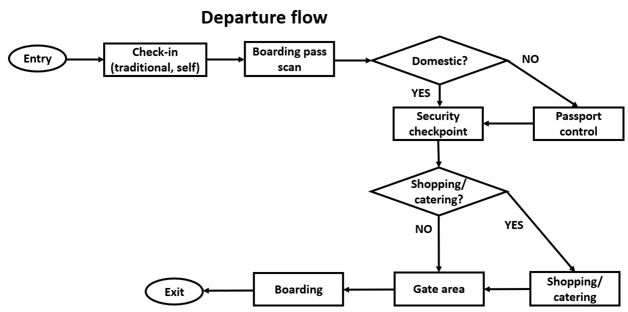

Figure 3. Passenger departure flow at the airport terminal

Departing passengers will be forced to pass through the scan of boarding passes and security checkpoints before going to the assigned gate. Even before that, passengers have the option to stop by at check-in counters (manned/self-service) for checking in to their flights and dropping off their luggage. The previous process is optional as nowadays airlines allow passengers to do the check-in online. Depending on the flight destination, passengers must undergo the process of passport control. Moreover, shopping and catering areas are often visited by passengers who spend their idle time there.

The COVID-19 pandemic has twisted all these predefined processes, as now airport operators require passengers to follow new safety measures (EASA, 2020; TSA, 2021). For instance, in the context of physical distance, the IATA (2020) demands a distance of at least one meter and the FAA (2020) a minimum of six feet (two meters). Additional processes such as temperature check, COVID-19 test certificate check, health self-declaration check have become standard procedures at most airport terminals. These processes are fitted into the existing airport terminal operations in different moments of the passenger trajectory (arrivals, departures, transfers).

In this work, the following processes have been implemented in the model:

- Check-in
- Boarding pass scan
- Security check
- Passport control
- Shopping/catering dwelling areas
- Gate lounge dwelling areas
- Gate boarding

The model recreates the actual facilities of an existing airport that, for privacy issues, kept anonymous. **Table 1** displays the type of facilities, the number for each facility type, and the processing time used to model the operations at each facility. Regarding the processing time, as there was no direct input data from the airport, these are chosen based on the authors' experience in the field.

The model of the airport terminal was built in accordance with the real layout of the airport. The airport has six access points in the departure hall, seven in the arrival hall, and 86 boarding gates. **Figure 4** shows a top view of the simulated airport terminal. As it can be seen, the terminal has gate areas (in green) distributed among four modules, namely Module A, B, C, and D.

Table 1. Characteristics of the airport terminal processes.

Process	Number of facilities	Processing time [sec.]
Check-in	204	[60-120][1]
Boarding pass scan	45	Uniform(6, 10) 1
Security check	19	[120-200]]
Passport control	18 passport control manual desks for departing and arriving passengers (6), respectively (12).	Unifor(25, 30) for Schengen passengers. Uniform(40, 60) for non-Schengen passengers
	80 passport control automated gates for departing (40) and arriving (40) passengers.	Uniform(25, 30)[2] for Schengen and non-Schengen passengers
Gate boarding	86	Normal(10,1)
Baggage claim	19	53

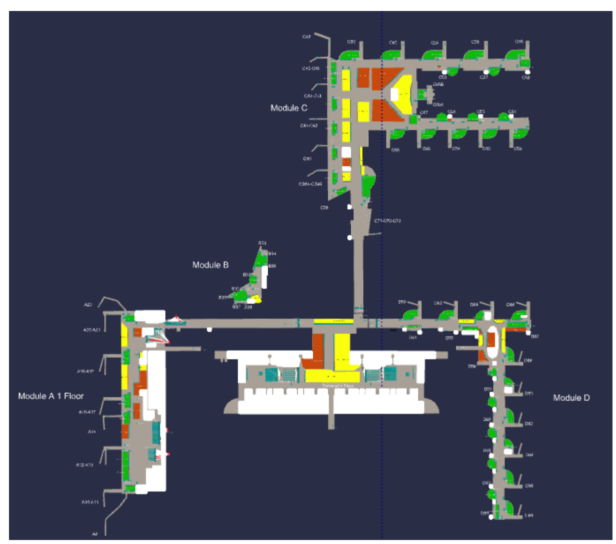

Figure 4. Top view of the airport terminal.

The airport terminal simulation model considers the processes displayed in **Figure 3**. Moreover, extra logic is implemented in order to make the model as

[1] Depending on the number of baggage to check-in.
[2] Automated passport reader scanners are used only by passengers who are holding a passport issued by a country of the Schengen area.
[3] This processing time refers to the baggage pick-up action. Passengers will need to wait until the bags will be available on the baggage belt.

close-to-reality as possible. These extra logics are described below:

- Opening/closing check-in desks: In the model, each check-in desk is assigned to a specific airline according to the real data provided by the airport operator. The check-in desks will be available for passengers of specific flights according to defined opening and closing times. Check-in desks will be open 2 or 3 hours before the respective scheduled departure time (SDT), depending on the flights' international or domestic status. The closing time is 40 minutes before SDT.
- Waiting at the departure hall: passengers that arrive at the airport terminal (departure hall) too early for check-in to their flights (using check-in desks) will wait in predetermined areas until the check-in desks assigned to their flight will open.
- Shopping/catering services: passengers that pass the security control will have the chance to visit shopping/catering services. The choice is based on the time left before the boarding gate opens. The choice of type of service, shopping or catering, is based on a probability value.
- Missing a flight: passengers that arrive too late at the airport terminal or that spend too much time undergoing the terminal processes (e.g., check-in, security control, passport control) will miss their flight if: the check-in desk is already closed; the boarding gate is already closed.
- Resources management: resources such as check-in desks, boarding pass scans, security checkpoints, and passport controls use by default a minimum number of resources (number of facilities/equipment). The utilization of these resources varies (increase or decrease), according to rules based on threshold values of the queuing waiting times and/or queue length. An example of this rule is the following: add (remove) service if the queuing waiting time is higher (lower) than 10 minutes.

To fulfill the goal of this paper, the simulation model computes the gate area occupancy, as the objective is to minimize the instances where flights are scheduled to gates that are close to each other. In this way, the risk of high passenger density is minimized, which minimizes the covid-19 risk of transmission.

3. Results and Discussion

The methodology developed in this work was applied to a real airport case study to test its validity. The case study refers to a large European airport that carries up to 25 million passengers per year. Due to a non-disclosure agreement with the airport operators, the authors will not mention the airport's name. The study is based on a full-day flight schedule that considers the high season traffic. In total, 768 air traffic movements were considered, 384 departures, and 384 arrivals. In these movements, 115579 passengers were transported. **Figure 5** and **Figure 6** show the daily trend of the traffic and passengers considered in the model. In **Figure 5**, there are three peaks during the day, between 6:00 and 7:00, 11:00 and 12:00, 17:00 and 18:00. If only the passengers flow is considered, **Figure 6** shows three peaks of 6604 at 7:00, 4995 at 12:00, 4888 at 17:00. This paper evaluates the number of passengers waiting to board the aircraft at each gate area of the airport terminal modules. The objective is to avoid scheduling flights too close to each other and avoid crowds of passengers in close-by gates. The results are presented by showing the original gate assignment compared with the optimized solution. By simulating the optimized solution, its feasibility is evaluated by considering the variability of the system. Due to the variability inherent in the system, the simulation outcome of the optimized solution might be worse than the simulation outcome of the original one.

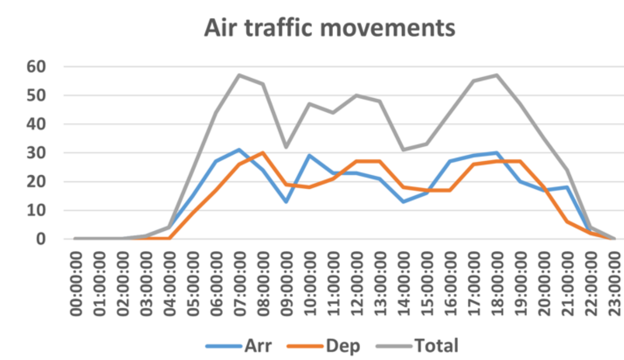

Figure 5. Air traffic movements in one day.

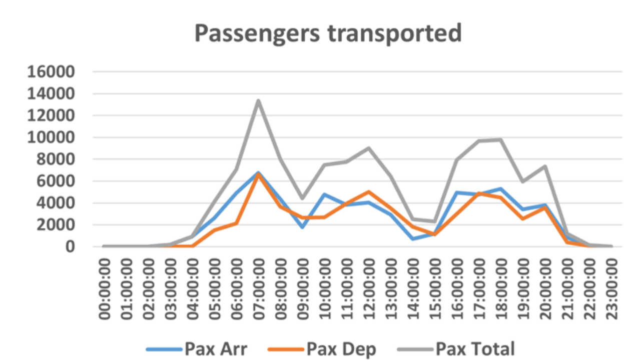

Figure 6. Passengers transported in one day.

3.1. Results from the EA optimization

After running the EA algorithm on the target flight schedule, which contained an original gate assignment, an optimized gate assignment solution has been obtained. The comparison of the quality of the original gate assignment and the obtained optimized solution is given in **Table 2**.

Table 2. Objective function value (fitness) comparison

	Fitness value
Original gate assignment	0.6484
EA optimized assignment	0.2370
Difference, %	36.5

As shown in Table 2, the EA improved the assignment by almost 37%, which means that 37% fewer flights were assigned simultaneously to the nearby gates. The following section compares two assignments in simulated stochastic conditions.

3.2. Results from the simulation

This section presents the results obtained by simulating the optimized solution. The section is divided according to the different airport terminal modules analyzed, namely, modules B, C, and D. Module A has been omitted from the results section as no conflicts were found in both the original and optimized schedule.

3.2.1. Module B

Module B of the airport terminal consists of eight boarding gates and three different gate areas. In this context, some gate areas accommodate multiple gates. The first gate area accommodates gates B30, B31, B36, and B37; the second gate area accommodates gate B32; the third gate area accommodates gates B33, B34, and B35. In **Table 3**, the Gate Taboo List used as input in the EA related to the gates of module B is shown.

Table 3. Gate Taboo List for the gates of module B

	B30	B31	B32	B33	B34	B35	B36	B37
B30	x	x					x	x
B31	x	x					x	x
B32			x	x	x	x		
B33			x	x	x	x		
B34			x	x	x	x		
B35			x	x	x	x		
B36	x	x					x	x
B37	x	x					x	x

Figure 7 shows the passenger occupancy of module B in the time interval between 4:00 and 9:00 related to the original gate assignment. In this graph, three conflicts have been detected. The first conflict appears between 4:30 and 5:45, involving gates B33 (green) and B34 (yellow). The maximum peak is reached when around 100 and 65 passengers occupy gates B33 and B34, respectively. The other two conflicts involve gates B30 (red) and B33 (magenta) conflicting with B32 (blue) between 6:30 and 7:15. In this time interval, the maximum peak is seen when the B32 area reaches around 60 passengers and B30, and B33 reach around 20 passengers.

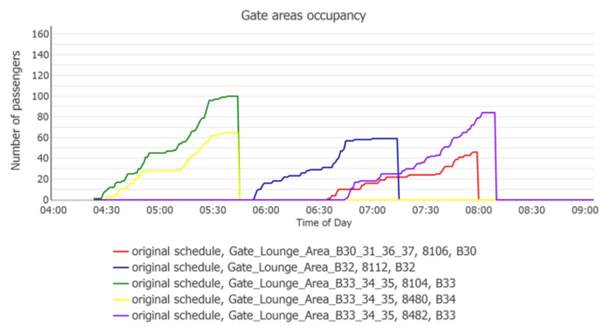

Figure 7. Original gate assignment schedule in module B (4:00 – 9:00)

Figure 8 shows the results obtained by implementing the optimized gate assignment schedule in the time interval 4:00 – 9:00. In this solution, three conflicts have been detected as well. The first involves gates B30 (red) and B32 (magenta) between 4:30 and 5:45. This conflict reaches its peak when B30 and B32 have around 100 and 65 passengers, respectively. The other two conflicts involve gates B31 (green) and B37 (yellow) conflicting with B36 (blue). The peak can be found when B36 has around 65 passengers, B37 30 and B31 25.

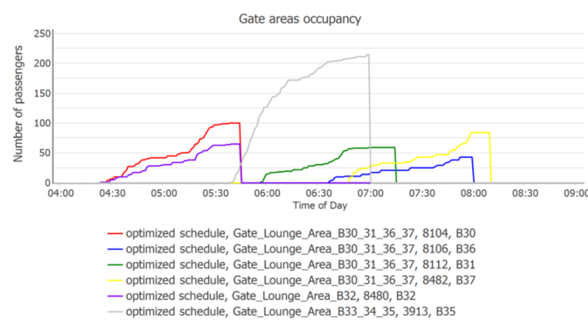

Figure 8. Optimized solution in module B (4:00 – 9:00)

To summarize, between 4:00 and 9:00, the EA could not effectively improve the original schedule, as the same amount of conflicts with similar values of gate area occupancy was found.

Figures **Figure 9Figure 10** show the gate area occupancy of module B within the time interval 9:00-12:00 for the original and optimized schedule, respectively. In the original schedule (**Figure 9**), there is one conflict between B33 (green) and B34 (blue) in the time interval between 9:30 and 10:15. The peak value is around 70 and 40 for B34 and B33, respectively.

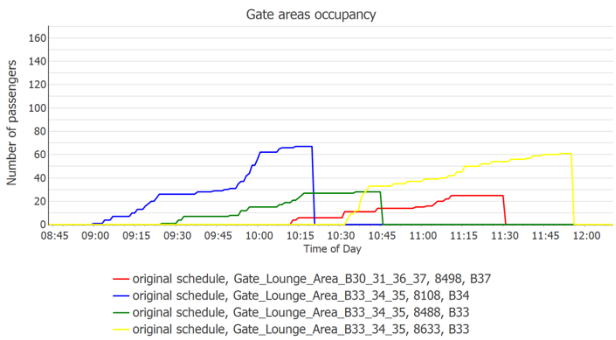

Figure 9. Original gate assignment schedule in module B (9:00 – 12:00)

In **Figure 10**, the gate area occupancy between 9:00 and 12:00 of the optimized gate assignment schedule is shown. In this schedule, there is one conflict between 9:30 and 10:30, involving gates B30 (green), B32 (yellow), and B36 (blue). The passenger occupancy peak value was around 150, 70, and 25 for B32, B36, and B30, respectively. In this instance, the optimized schedule does not improve the original assignment, as they both generate the same amount of conflicts. Considering the number of gates involved and the gate occupancy peak values, the optimized solution obtains even worse values than the original schedule.

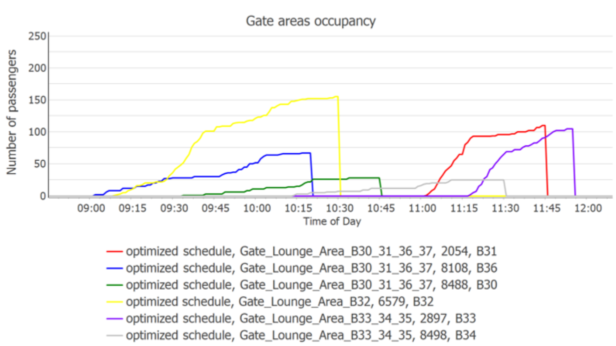

Figure 10. Optimized solution in module B (9:00 – 12:00)

Figures **Figure 11** and **Figure 12** show the gate area occupancy between 12:00 and 16:00 for the original and optimized schedule, respectively. In **Figure 11**, four conflicts can be detected. In the first conflict, gate B33 (red and blue) has two flights assigned next to each other, creating a passengers' overlap in the gate area. Moreover, B34 (yellow) enters into conflicts with B33 as well. The maximum value of the gate area occupancy is around 170, and it is found at 13:30. Another conflict is found between 13:30 and 14:00 involving the gates B33 (red) and B34 (yellow and magenta). The maximum value of gate area occupancy is around 200. Gate B34 (yellow and magenta) keeps a passenger overlap until 14:30, reaching a peak of 110 passengers in the gate area. One last conflict is found between 14:45 and 15:00 involving B33 (green) and B34 (magenta). In this case, the gate area occupancy value is smaller than the previous conflicts, approximately 90 passengers.

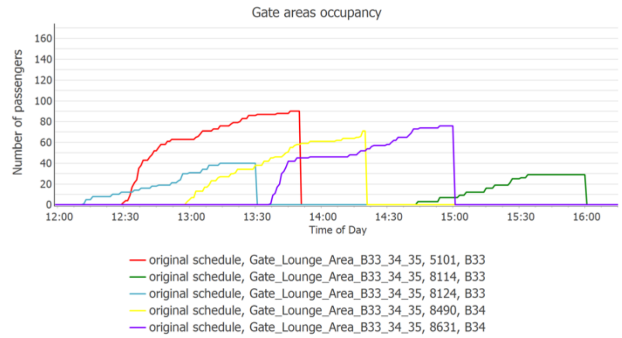

Figure 11. Original gate assignment schedule in module B (12:00 – 16:00)

In **Figure 12**, the optimized schedule in the time interval between 11:30 and 16:00 is shown. Here, the presence of three conflicts can be noticed. The first involves gates B31 (red) and B36 (green), and it is detected in the time interval between 12:15 and 12:30. The second conflict involves gates B30 (yellow) and B36 (green) in the time interval between 13:00 and 13:30. The third conflict involves B34 (magenta) and B35 (grey) in the time interval between 13:45 and 14:30. The maximum values of the gate area occupancy for these three conflicts are approximately 175, 70, and 125, respectively. In this instance, the optimized solution improved the original schedule as there were fewer conflicted gates with smaller values of gate area occupancy (570 in the original schedule and 370 in the optimized one).

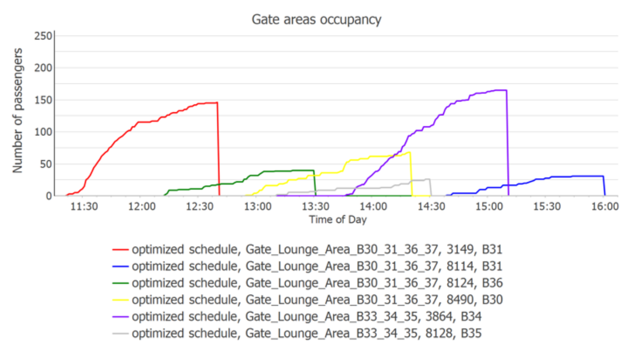

Figure 12. Optimized solution in module B (11:30 – 16:00)

Figures **Figure 13** and **Figure 14** show the gate area occupancy between 16:00 and 20:00 for the original and optimized gate assignment schedule, respectively. By looking at **Figure 13**, one conflict can be detected, involving gates B33 (green) and B34 (red) within the time interval 17:45 – 18:30. The maximum value of gate area occupancy is 80. The optimized schedule, as shown in **Figure 14**, also has one conflict. This conflict involves gates B30 (blue) and B31 (red) between 17:45 and 18:00, reaching the maximum value of gate area occupancy of 195. However, the conflict duration was not too long, lasting around 15 minutes.

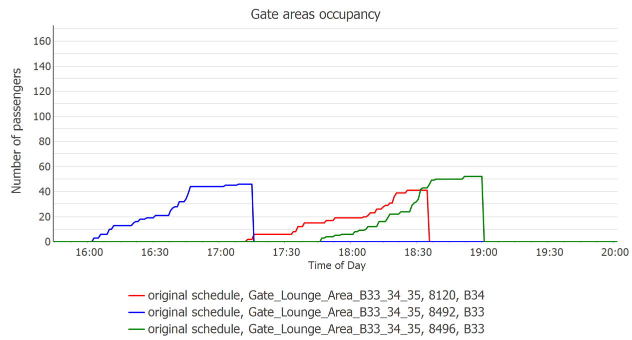

Figure 13. Original gate assignment schedule in module B (16:00 – 20:00)

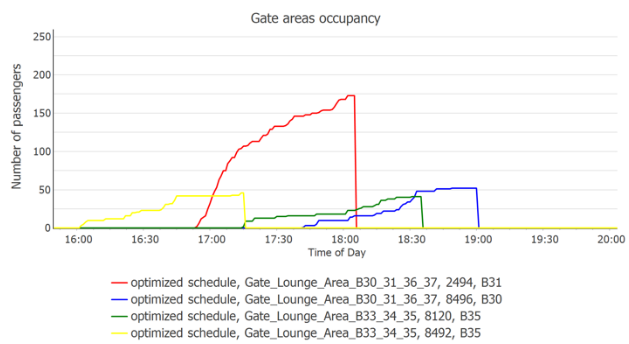

Figure 14. Optimized solution in module B (16:00 – 20:00)

Overall, in module B, the optimization algorithm found it very difficult to improve the original schedule as in each time interval, a similar number of conflicts was found. Due to the peculiarity of the layout of Module B, having eight gates sharing only three gate areas, it was very difficult for the optimization to find a free-conflicts solution. However, by considering all the conflicts found and comparing the value of gate area occupancy between the original schedule and the optimized one, the optimized schedule still provides better results than the original one.

3.2.2. Module C

In this section, the results related to module C are presented. In Table 4, the Gate Taboo List for the gates of module C is shown. Due to space limitations, only the gates involved in conflicts are shown so that the graphs with the results can be better interpreted. Figures **Figure 15** and **Figure 16** show the gate area occupancy of gates C45, C46, and C48 for the original and optimized gate assignment schedules, respectively. Only one conflict has been found in the original schedule, which involves gates C46 and C48 between 4:00 and 4:20. The maximum value of the gate area occupancy is approximately 60. Regarding the optimized schedule, no conflicts have been detected in module C. **Figure 16** shows that in the same time interval (2:15 – 6:00), only one gate (C46) is scheduled, therefore, improving the original schedule.

Table 4. Gate Taboo List for the gates of module C (45, 46, 48)

	C45	C46	C48
C45	x	x	x
C46	x	x	x
C48	x	x	x

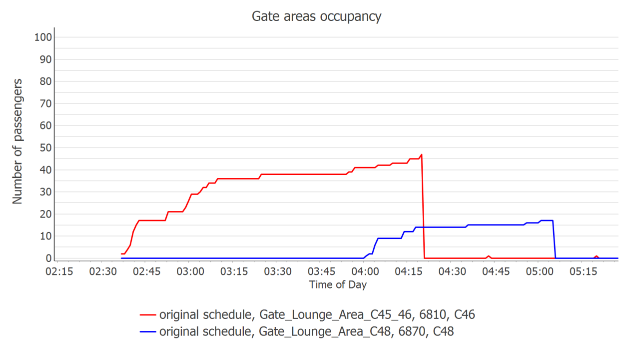

Figure 15. Original gate assignment schedule in module C (gates 46, 48)

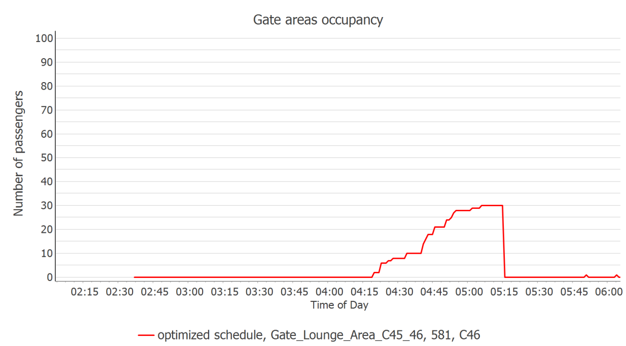

Figure 16. Optimized solution in module C (gates 46, 48)

3.2.3. Module D

In this section, the results related to module D are presented. In 5, the Gate Taboo List for the gates of module D involved in conflicts is shown.

Table 5. Gate Taboo List for the gates of module D (88, 90, 96, 97, 98)

	D88	D90	D96	D97	D98
D88	x	x			
D90	x	x			
D96			x	x	x
D97			x	x	x
D98			x	x	x

Figures **Figure 17** and **Figure 18** show the gate area occupancy for gates D88 and D90, for the original and optimized gate assignment schedules, respectively. In the original schedule, there is one conflict between gates B88 and B90. This conflict can be seen in **Figure 17**, as can be noticed an overlap of passengers from 3:30 until 4:45, with a maximum value of gate area occupancy of 90. The EA was able to solve this conflict, as shown in **Figure 18**. In this schedule, for this specific time interval (2:00-6:00), only gate B88 was assigned with a flight; therefore, any possible conflict was avoided.

Figures **Figure 19** and **Figure 20** show the gates D96, D97, and D98 area occupancy. **Figure 19** shows the original gate assignment schedule, where the presence of one conflict can be detected. This conflict involves gates D96 and D98, between 4:00 and 5:00, with a maximum gate area occupancy value of 70. The optimized schedule was able to avoid conflicts, as **Figure 20** shows. In this instance, there are no overlaps between flights assigned to gates D97 and D98. As shown in these graphs, the EA effectively solved conflicts and improved the original assignment for module D.

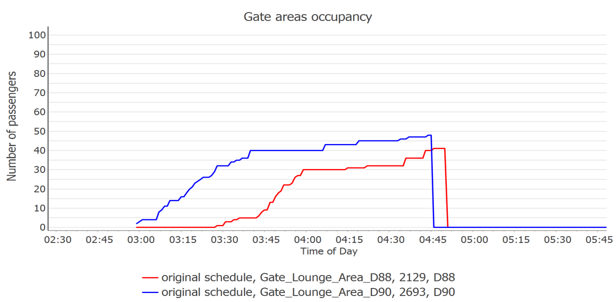

Figure 17. Original gate assignment schedule in module D (gates 88, 90)

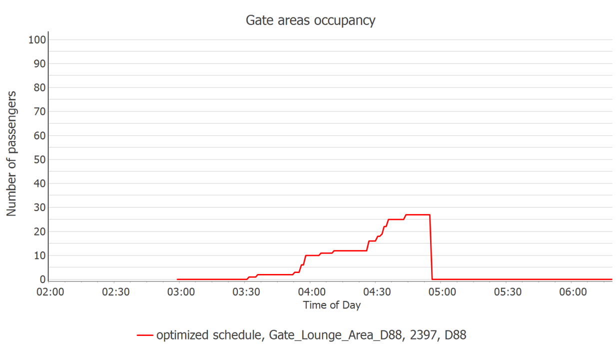

Figure 18. Optimized solution in module D (gates 88, 90)

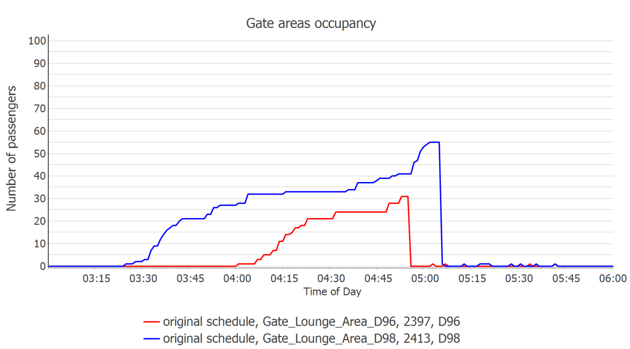

Figure 19. Original gate assignment schedule in module D (gates 96, 97, 98)

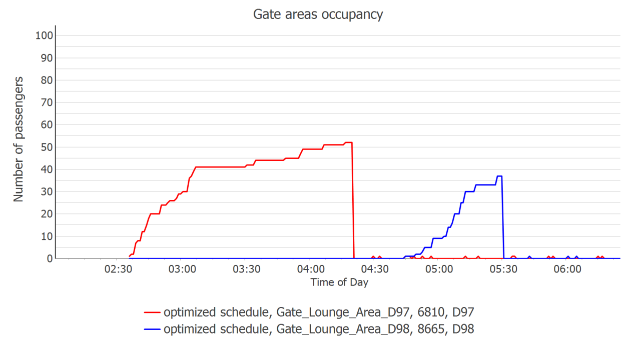

Figure 20. Optimized solution in module D (gates 96, 97, 98)

4. Conclusions

In this paper, an Opt-sim approach was applied to the GAP. The Opt-sim approach took advantage of an EA to find a (sub)optimal solution, which was then simulated using a passenger dynamic simulation model. The output from the simulation model gave information about the evolution of the gate occupancy through time, which was used as the main indicator for evaluating the effectiveness of the optimized solution. The GAP solver applied in this study was adapted to include covid-19 measures in airport terminals and reduce the covid-19 risk of transmission.

The simulation results highlight that the layout and traffic schedules affect the performance significantly, as it was found that conflicted situations in specific parts of the terminal (e.g., Module B) were not easily improved by the optimized schedule. However, in modules C and D, the optimized schedule provided a better solution than the original schedule. Overall, the optimized schedule reduced the risk of transmission by avoiding concentrations of passengers in close-by gates at the same time of the day.

In future work, the GAP-solving approach can be improved by considering the number of passengers expected at the gates in the objective function. In this way, more density-aware results could be obtained. Moreover, a more detailed output from the Opt-sim approach could be obtained by considering gate area occupancy peaks overlap and the duration of such overlaps. Finally, different policies could be implemented to influence the passengers' behavior in the terminal, which would avoid passengers' concentration in specific gate areas.

References

Abdelghany, A., Abdelghany, K., Azadian, F., 2017. Airline flight schedule planning under competition. Computers and Operations Research 87, 20–39. https://doi.org/10.1016/j.cor.2017.05.013

ARC, 2021. CAST. https://arc.de/cast-simulation-software/. Accessed on: March 2021.

Bagamanova, M., Mujica Mota, M., 2020a. A multi-objective optimization with a delay-aware component for airport stand allocation. Journal of Air Transport Management 83, 101757. https://doi.org/10.1016/j.jairtraman.2019.101757

Bagamanova, M., Mujica Mota, M., 2020b. Reducing airport environmental footprint using a disruption-aware stand assignment approach. Transportation Research Part D: Transport and Environment 89, 102634. https://doi.org/10.1016/j.trd.2020.102634

Das, G.S., Gzara, F., Stutzle, T, 2020. A review on airport gate assignment problems: single versus multi objective approaches. Omega. Vol. 92, pp. 102/146. Doi: https://doi.org/10.1016/j.omega.2019.102146

EASA. 2020. COVID-19 Aviation Health Safety Protocol : Operational Guidelines for the management of air passengers and aviation personnel in relation to the COVID-19 pandemic.

EUROCONTROL, 2021. EUROCONTROL Comprehensive Assessment of COVID-19's Impact on European Air Traffic. https://www.eurocontrol.int/publication/eurocontrol-comprehensive-assessment-covid-19s-impact-european-air-traffic. Accessed on: April 2021.

Federal Aviation Administration (FAA), 2020. Covid-19: Updated interim occupational health and safety guidance for air carriers and crews.

Ghazouani, H., Hammami, M., Korbaa, O., 2015. Solving airport gate assignment problem using Genetic Algorithms approach, in: 2015 4th IEEE International Conference on Advanced Logistics and Transport, IEEE ICALT 2015. IEEE, pp. 175–180. https://doi.org/10.1109/ICAdLT.2015.7136615

Guepet, J., Acuna-Agost, R., Briant, O., Gayon, J.P.P., 2015. Exact and heuristic approaches to the airport stand allocation problem. Eur. J. Oper. Res. 246, 597–608. Doi: https://doi.org/10.1016/j.ejor.2015.04.040

International Aviation Transport Association (IATA), 2020. Guidance for cabin operations during and post pandemic, edt. 4.

Mujica Mota, M., 2015. Check-in allocation improvements through the use of a simulation–optimization approach. Transportation Research Part A: Policy and Practice 77, 320–335. https://doi.org/10.1016/J.TRA.2015.04.016

Mujica Mota, M., Flores de la Mota, I., 2017. Applied Simulation and Optimization, Vol. 2 (first ed.), Springer (2017)

Mujica Mota, M., Scala, P., Herranz, R., Schultz, M., Jimenez, E. 2020. Creating the future airport passenger experience: IMHOTEP. Proceedings of the 32nd European Modeling & Simulation Symposium, 17th International Multidisciplinary Modeling & Simulation Multiconference. Doi: https://doi.org/10.46354/i3m.2020.emss.024

Dorndorf, U., Jaehn, F., Pesch, E., 2008. Modelling Robust Flight-Gate Scheduling as a Clique Partitioning Problem, Transportation Science, vol. 42, no. 3, pp. 292–301. Doi: https://doi.org/10.1287/trsc.1070.0211.

Scala, P., Mujica Mota, M., Wu, C.L., Delahaye, D., 2021. An optimization-simulation closed-loop feedback framework for modeling the airport capacity management problem under uncertainty. Transportation Research Part C: Emerging Technologies, Vol. 124, March 2021, 102937. Doi: https://doi.org/10.1016/j.trc.2020.102937

TSA, 2021. Coronavirus (COVID-19) information. https://www.tsa.gov/coronavirus. Accessed in April 2021.