

Allocation of Airport Check-in Counters Using a Simulation-Optimization Approach

Miguel Mujica Mota and Catya Zuniga Alcaraz

Abstract The aviation industry is expected to grow at a pace of 4% per annum in the coming years, therefore it is necessary to have techniques that support the management of the resources at hand in the best possible way so that facility expansion is delayed as much as possible with the corresponding capital savings. This chapter presents a methodology that combines evolutionary algorithms and simulation for performing the allocation of the check-in desks in such a way that the different stochastic and deterministic variables are taken into account for a more robust solution. The evolutionary algorithm is developed to satisfy the different mandatory restrictions for the allocation problem such as minimum and maximum number of check-in desks per flight, load balance at the counters, opening times of check-in desks, and other restrictions imposed by the level of service agreement. Once the solutions are obtained, a second evaluation is performed using a simulation model of the terminal that takes into account the stochastic aspects of the problem such as passenger arrival profiles, passenger profile, layout of the facility, among others, with the purpose of determining an airport terminal's check-in area which allocation is the most efficient in real situations to keep the quality indicators at the desired level. The example presented is for an airport terminal's check-in area, but the methodology can be used for similar allocation problems in the aviation industry and in other industries such as logistics or manufacturing.

1 Introduction

The aeronautical industry is still under expansion, in spite of the rise in oil prices, limited capacity, and new regulations. Different scenarios have been forecasted to explore the future of the Air Transportation System, the most likely scenario predicts

M. Mujica Mota (✉)

Aviation Academy, Amsterdam University of Applied Sciences, 1097, DZ, Amsterdam, The Netherlands

e-mail: m.mujica.mota@hva.nl

C.Z. Alcaraz (✉)

Logistics & Supply Chain Management Department, Universidad Popular Autonoma del Estado de Puebla, 17 Sur 901, Barrio de Santiago, 72410 Puebla, Puebla, Mexico

e-mail: catyaatziry.zuniga@upaep.mx

© Springer International Publishing Switzerland 2015

M. Mujica Mota et al. (eds.), *Applied Simulation and Optimization*,

DOI 10.1007/978-3-319-15033-8_7

that air traffic will double in the next 15 years. In the forecast of IFR flight movements in Europe up to 2035, the most likely scenario predicts 14.4 million flights, 50 % more than in 2012. Even under relatively conservative assumptions a steady 4–5 % annual growth will lead to a near doubling of total air travel during this period [1–3].

Increased air traffic makes the efficient management of available resources on both the airside and the landside of an airport even more complex. On the airside, it is even more evident on the runways and in the airspace surrounding airports, where the arrivals and departures serve a large number of aircraft that are subject to many logistical problems, which must continuously be solved to make sure every flight and passenger travels safely and efficiently. Besides the increased number of flights, there has also been an increase in the size of aircraft which in turn augments the number of passengers. These conditions could generate potential bottlenecks or congestion in the terminal buildings if the available resources are not efficiently managed. Inside the terminal buildings, they manifest as huge queues at the security filters, baggage handling systems, and check-in counters, to name but a few. In addition, they also cause excessive waiting times that the customers see as bad service levels.

For the sustainability of the Air Transportation System (ATS) all over the world, various ideas have been proposed to alleviate traffic growth and its implications such as the construction of additional runways or terminal buildings, or improved sequencing of operations in both air and land side, etc. However, more fundamental and innovative changes are required to improve the use of available air capacity.

In the case of airport terminals, analysts need to take into account not only the typical operational restrictions inherent in the system but also other measures that will make it possible to evaluate the perception of passengers, who are the main customers of the terminal and will drive the economic and social development of the system.

The International Air Transport Association (IATA) has published some guidelines for what is called as Level of Service (LOS) indicators. These measures evaluate characteristics associated with perceived comfort inside the terminal, such as available area per passenger, the speed at which the passengers can move inside the terminal, waiting times, queue lengths, etc., in different situations. Those metrics are of particular interest to airport planners which is why scientific community has focused on determining the factors that influence perception [4]. Table 1 illustrates some of these typical values suggested by IATA [5].

Optimization techniques are able to give optimal or close-to-optimal solutions to problems that are deterministic in nature; on the other hand, simulation approaches can consider the stochastic nature of the processes that participate in the system under study, while at the same time being able to describe the studied systems at different levels of abstraction. This chapter presents a methodology that combines the two approaches to generate better solutions than the ones that could be achieved by applying each technique independently. The methodology presented is applied to the check-in allocation problem for illustrative purposes but the methodology itself can be used in a wide range of problems from diverse fields to generate more robust solutions.

Table 1 Level of service indicators

m ² /Pax	A	B	C	D	E
1. Few trolleys and passengers with check-in baggage (row width 1.2m)	1.7	1.4	1.2	1.1	0.9
2. Few trolleys and 1 or 2 pieces of baggage per passenger (row width 1.2m)	1.8	1.5	1.3	1.2	1.1
3. High percentage of passengers using trolleys (row width 1.4m)	2.3	1.9	1.7	1.6	1.5
4. Heavy flights with 2 or more items per passenger and high percentage of passengers using trolleys (row width 1.4m)	2.6	2.3	2.0	1.9	1.8

The problem is tackled in the following way. First a brute-force approach is implemented to obtain initial feasible solutions taking into account quantitative restrictions such as minimum and maximum number of check-in desks per flight, load balance in the check-in islands, opening times of check-in desks, and other restrictions imposed by the LOS.

The initial solutions are then encoded as chromosome-like data structures and operations are performed in order to obtain the most promising solutions under a particular cost function. Once the initial solutions are obtained, they are in turn evaluated using a simulation model of the particular terminal under study including in the model stochastic variables that count for the passenger arrival profiles, opening times, layout of the facility, interactions between passengers, efficiencies of processes, etc. With these elements it is possible to determine which allocation is the most quality-efficient in a close-to-real scenario in order to maintain the LOS indicators at the desired level. The proposed methodology has been put into practice using information from a real terminal but it can be easily adapted to a different one with different restrictions imposed by the corresponding LOS agreements between the airport operator and the corresponding airlines.

The remainder of the chapter continues in the following way: Sect. 2 is a brief review of the principles of both approaches, simulation, and evolutionary algorithms are presented. Section 3 introduces the proposed methodology, Sect. 4 presents and discusses the check-in problem in detail, and the different steps of the methodology are described. Finally, Sect. 5 wraps up the discussion of the chapter.

2 Evolutionary Algorithms and Simulation

Evolutionary algorithms are a group of so-called metaheuristics. The authors selected evolutionary algorithms for tackling these problems because they have been widely used by scientific community and the implementation is rather easy. In the

optimization process, the cost function can be designed using different metrics which makes the model flexible. However, the solutions obtained do not represent real-life systems where random factors play an essential role to get optimal results. The drawbacks of evolutionary algorithms are overcome with the integration of simulation in the methodology. This integration leads to a more robust approach that can be easily adapted for problems in other fields by the reader.

There have been several theoretical and practical contributions to evolutionary algorithms field, as evidenced by the books, papers, and workshops proceedings published in the last years such as [6–8] or [9] among others. The use of metaheuristics such as evolutionary algorithms to solve problems of this nature has been motivated mainly because the population-based nature allows the generation of several elements of the Pareto optimal set in a single run. Evolutionary algorithms can be very useful for the selection of parameters to optimize the performance of a system. Furthermore, The choice of any decision parameters can cause the system to perform better or worse, which can be measured by some relevant objective or fitness function, as in real systems, where the interactions between the parameters are not generally amenable for analytical treatment.

In this section, a review of both approaches, Evolutionary Algorithms and Simulation are presented so that the reader can have a clear understanding of the differences and advantages between them.

2.1 Evolutionary Algorithms

The Evolutionary Algorithms (EAs) fall within the so-called population-based metaheuristics [10]. These techniques, which are considered as a general class of stochastic optimization algorithms, are employed to find optimal (or as optimal as possible) solutions to hard problems in a very wide range of areas.

Evolutionary algorithms are a group of methods inspired by the evolutionary processes found in nature, they borrow some concepts from population biology, genetics, and evolution such as inheritance, mutation, natural selection, and recombination (or crossover) to guide the search within the solution space. Detailed information can be found in [7, 10–14] to mention just some of the literatures.

The general idea behind an evolutionary algorithm is the representation of a solution to the problem in the form of a vector of decision variables. Using biological terms, the *genotype* is the encoded representation of the variables, and the *phenotype*, *chromosome* or *genome*, the set of variables themselves. In other words, a genotype or individual, represent a solution to the problem to be solved, and is represented by a list of parameters, also called chromosomes.

In most cases the transformation or modeling task is not simple but rather a complex one and depends on the perception of the modeler. Thus the transformation of the decision variables into a vector-like representation is per se an interesting and challenging problem. Once the decision variables have been represented in the form of a vector, the optimization problem can be specified.

Let us assume that we have a discrete search space and a function that assigns a value to each of the elements in the search space.

$$f : X \rightarrow \mathbb{R} \tag{1}$$

The general problem is to find:

$$\min f, x \in X \tag{2}$$

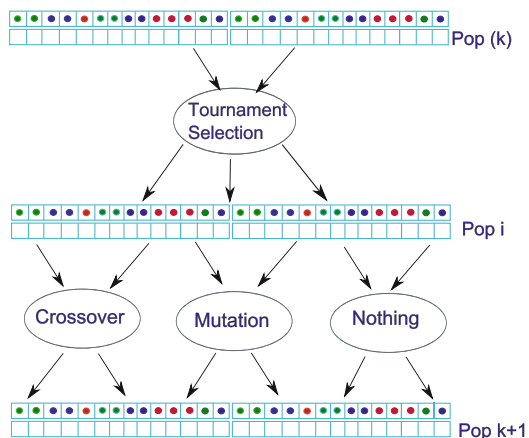
Here x is a vector of decision variables, and represents the objective function. Such a problem is commonly called discrete or combinatorial optimization problem [7].

Basic evolutionary algorithms follow the next steps. First, an initial population is constructed where several individuals are randomly generated to form the first initial population $POP(k)$. Then each individual is evaluated, and a value of fitness is returned by a fitness function. The initial population undergo a selection, mutation, and recombination process to identify the best adapted individual. Figure 1 illustrates this transformation process.

There are different ways of selecting individuals and a very popular one is the deterministic (λ, μ) -tournament selection. This selection begins by randomly selecting λ individuals from the current population $POP(k)$. Together with the selection, a fitness measure f is performed to evaluate each individual to keep the μ best ones. Using this evaluation, solutions that have a higher value of the so-called fitness function are identified and better opportunities for further evolution are given to those solutions. These steps are repeated until a new intermediate population (POP_i) is completed. Following selection, the evolutionary operations, mutation, and recombination (crossover) can be applied to improve the original problem.

The chromosomes of the parents are mixed during crossover, hence crossover results in two new individual child, which are added to the next generation population $POP(k + 1)$. Parents and children are joined in some fashion to form a new

Fig. 1 Evolutionary algorithm process



next-generation population that is different from the initial generation, and the cycle continues. This generational process is repeated until a termination condition, imposed by the developer, has been reached.

2.2 Simulation

Simulation is the imitation of the operation of a real system or process over time. It is used to generate artificial history and data of a system, and for the observation and analysis of that artificial history to draw inferences concerning the operating characteristics of the real system [11].

The model usually represents a set of parameters and assumptions concerning the operation of the system. These assumptions are expressed in mathematical, logical, and symbolic relationships between the entities, or objects of interest of the system. Once developed, verified and validated, a model can be used to investigate a wide variety of “what-if” scenarios about the real-world system [15]. Potential changes to the system can then be simulated in order to predict their impact on the system’s performance. Furthermore simulation can also be used to study systems in the design stage, before such systems are built based on relationships taken from other fields. Thus, simulation techniques can be used both, as an analysis tool for predicting the effect of changes to existing systems, and as a design tool to predict the performance of new systems under varying sets of circumstances. Nowadays, with the evolution of computer capacities, computer simulation is also able to develop very accurate and graphically appealing models that can represent a system at different levels of abstraction, depending on the objective of the study.

A simulation model can be developed using different tools, for example, several studies have been performed using modeling formalisms such as Coloured Petri Nets [16] or using commercial software such as SIMIO [17] or ARENA [18] in which the modeler makes use of a library of objects and just has to put them together.

Simulation alone has been proven to give good solutions in different fields such as the transport industry [19], manufacturing [20], airport operations [21], etc. However when it is used as a decision support tool it cannot ensure the best outcome since the experiments only explore a subset of the whole different configurations of the system under study and depending on the size of the model and the characteristics of the computer the number of experiments is limited to the time window for the decision to be taken.

Simulation recently has been used in combination with other techniques in order to overcome the aforementioned drawbacks. It has been used to explore scenarios more efficiently in combination with Petri nets [20] or for the evaluation of disturbances with the use of constraint programming techniques [15], just to mention a couple of examples. Thus the decision-making process supported by simulation experiments always has a certain level of uncertainty that can be minimized as more experiments are performed, however, this activity is time consuming and penalizes its potential for timely decisions over the real system.

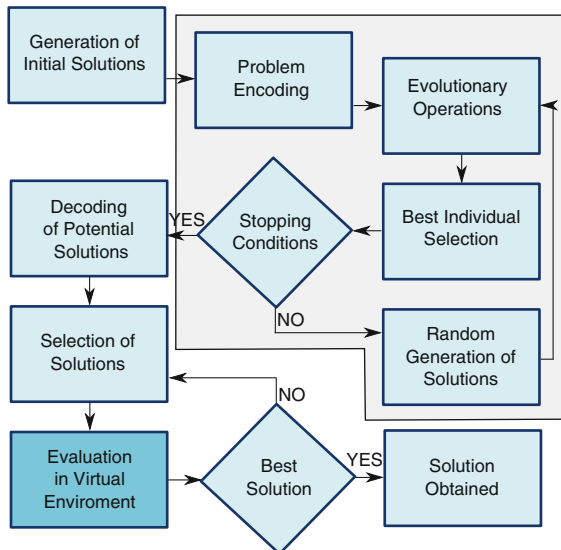
3 Methodological Approach OPT-SIM

The methodology presented has been applied to the case study of an airport terminal with good results. Nevertheless, the methodology can be implemented in a different set of problems from the one presented in this chapter. Some typical problems that can be tackled with this approach are the analysis of counter areas such as those of banks, service areas, ports, security filters and, in general, in situations where allocation of resources must be performed and the stochasticity presented in the system cannot be included in the analytical model, which would otherwise be quite easily implemented using some mathematical programming techniques. Allocation problems in particular can benefit from a mathematical programming formulation, especially when operational or sample size constraints lead away from straightforward or closed-form solutions.

Figure 2 gives the diagram that illustrates the different steps in the methodology. At the beginning a brute-force approach or constraint satisfaction problem generates feasible solutions. The feasible solutions are encoded and are improved by the evolutionary algorithm. Then some solutions are selected and evaluated in the simulated environment, where the stochasticity is integrated to come up with a more robust solution.

The Evolutionary Approach starts by representing the problem variables in a chromosome-like structure to produce some initial feasible set of solutions for the allocation problem to feed the evolutionary algorithm. As briefly described in Sect. 2, an initial population is constructed, where several solutions are randomly generated to conform the initial population $POP(k)$. Then, the evaluation process is performed, and a value of the fitness function is returned to measure the quality of each solution to end up with n best solutions. The fitness function is designed in such a way that it

Fig. 2 OPT-SIM methodology



measures the solutions depending on the objectives of the study. The new population undergoes a mutation and recombination process. After k iterations, a pool of efficient solutions is obtained. Figure 1 illustrates this transformation process.

The next phase of the methodology integrates the stochastic factors using a simulation model, where the solutions obtained in the previous phase are further improved through experiments with the simulation model. With this approach the simulation model starts in an improved configuration, as the solutions obtained using the evolutionary algorithm are cost efficient, therefore the improvement with the simulation approach is less time consuming.

With the previous implementation, the problem is approached by taking as many characteristics as possible into account instead of restricting only to either the ones that are limited to the perception of the modeler with an analytical technique or a time-consuming analysis performed with a simulation approach.

The best solution or solutions are selected depending on how the dynamic model has performed, thus ending up with a more robust solution than the one that could have been obtained through the sole use of one technique or the other.

4 Case Study: The Check-In Allocation Problem

An airport terminal is a facility where passengers start on their journey through the air transport service. In order to board the aircraft, the passenger must undergo different processes involving management resource. First the passengers must arrive to the terminal by any mean of transport that, depending on the location, could be public transport (bus, metro, train) or private transport (private car, taxi, shuttle). Once they enter the terminal, a registration process starts.

The check-in service consists of passenger registration, commonly known as the check-in, and handing over their baggage; moreover, this is the moment when passengers get a first impression of the airport and the airline. Although this first impression is very important, there are other issues involved in the management of resources. Baggage management is also a vital issue that it must be considered to successfully monitor the proper operation of the airport and the airline. The baggage has to be transported to the right airplane using conveyor belts and trolleys.

Passengers have to go to a check-in desk or a common check-in island to get their boarding passes and to drop off their baggage. The check-in process can be either manual or automatic. Even though automatic check-in (self check-in) is growing in popularity, there are still a lot of passengers who prefer the manual process, which is normally performed by personnel provided by the airline. In some cases, personnel provided by the airport may perform this process, depending on the agreement between the parties involved. The facility and resources needed to perform the check-in activity are normally provided by the airport through an agreement that sets up the conditions and also the performance indicators (PI) that will measure the quality of service provided.

At this point, there are two opposing objectives: on the one hand, the airlines want to provide the best possible service to their customers at the least cost while, on the other hand the airport needs to provide this service with limited resources that, in the case of a check-in area, means the available check-in desks and personnel. Furthermore, due to increased traffic and the consequent large inflow of passengers and baggage heading for many different flights, the check-in allocation problem has been gaining importance in the relevant literature.

An inefficient management of resources such as ground services, personnel, desks, filters, etc., are appreciated as congestion in terminals. The congestion in turn can be appreciated in several points throughout the passenger boarding process; for example, in the check-in desks, the security filters, passport control, and sometimes at the boarding gate. These problems have been traditionally faced by the aviation industry through the increase of physical resources (e.g., increase in the check-in desks, increase in the number of security filters, etc.). Furthermore the increase in competition between airlines and airports has forced both actors to optimize their resources at hand in order to reduce their costs and keep competitive. On the other hand, the increase of passenger traffic in airports makes necessary the development of novel decision support tools that take into consideration all the different elements that are involved in the system and influence the correct allocation of resources.

Furthermore the increase in traffic caused mainly by the competition between airlines, market liberalization, and the increasing number of low-cost airlines will force the need of efficient strategies and procedures for allocating the resources inside terminals if the LOS are to be maintained.

The general check-in allocation problem consists of allocating the available desks of a terminal in such a way that the allocation satisfies a series of restrictions imposed by the airport and the companies through a service contract. These restrictions may change depending on the airport, the airlines, the region it serves, and the type of terminal [22].

The check-in allocation problem is a well-known problem in airport terminals that has been studied by some authors using evolutionary approaches or mathematical formulations [23–25]. These techniques have the drawbacks that do not take into account all the different elements in the terminal's check-in area or the interactions of passengers inside the terminal with each other or with other elements of the facility.

The methodology presented in this chapter deals with the problem of performing the allocation of check-in desks in a terminal, considering not only the internal policies and quality indicators but also the interactions between passengers and the physical facilities of a terminal. The various rules and information data were provided by a terminal in the Middle East under a confidentiality agreement. We shall refer to this airport, when applicable, as “the airport.”

4.1 Literature Review

The check-in allocation problem has received little attention in the literature over the last few decades, but in recent years, because of the increasing traffic demand, special attention has been placed on this process due to its economic and time importance. It has been defined and studied using different modeling techniques and methods for its resolution.

Parlar and Sharafali [26] use a dynamic programming approach for the check-in allocation problem assuming that it is possible to close or open counters depending on the demand. This practice is efficient from the passengers' point of view as the perceived quality of service is high in comparison to a static way of managing the counters but it is difficult for some companies and the airport to count on having extra counters when the demand is at its peak.

For these kinds of resolution, the parties have different objectives; the airlines want to minimize their use, while the airports want to maximize their use. On the other hand, although the trend in airports now is to use paperless tickets and self check-in kiosks, this is not the case in many of the airports around the world, especially in the growing Asia-Pacific region where they are used to more personal treatment. Besides, some developing regions, such as Latin America or Africa, still depend strongly on the manual check-in process.

Another similar paper by Littler and Whitaker [27], provides a procedure for estimating staffing requirements to meet a preset processing time target. It uses a stochastic simulation of passenger arrivals at the terminal. This paper mainly focuses on the design phase of a managerial schema for the use of check-in counters.

Another paper on this problem in the Hong Kong Airport was presented by Chun and Mak [28]. Their work combines simulation with an allocation system, taking restrictions and desires of the companies at the airport into account. They use a simulation-based optimization approach to determine the best check-in allocation, considering the stochasticity of some of the processes involved, such as service and arrival rates, thus evaluating the fitness of the solutions by analyzing the efficiency of processing passenger by predicting queue lengths.

More recently, Park and Ahn [29], revisited the problem of passenger arrivals at Gimpo airport to determine the most appropriate number of check-in counters, again this work focuses on the sizing of the resources with a view to a particular objective.

With a different scope, Yan et al. [30], provide an integer programming approach to the assignment of common-use check-in counters for a model of Taipei International Airport. However, due to some size limitations they had to come up with a heuristic method to solve the model. This work aims at assigning the allocation of common-use counters in a close-to-optimal way.

van Dijk and van der Sluis [31] present a paper that is connected with the work presented here; the authors use simulation to determine the minimum numbers of desks and then an integer programming approach to optimize, as much as possible, the resources spent on allocating passengers. The work presented by these authors analyzes the problem from the passengers' point of view, with an eye to shortening

service times. It only considers some of the restrictions that are presented in this chapter.

On the other hand, the problem presented by Hsu et al. [24] deals with the dynamic allocation of the check-in services required by passengers aiming at reducing the total time the passengers spend on the check-in procedure. The assumptions made by the authors are from the standpoint of the passengers, so the model could be used for assigning the steps the passengers need to follow in order to minimize their processing time; some assumptions, such as an average processing time, are made, thus discarding the inherent stochasticity of this operation. Their approach uses the average values and clustering the passengers.

Very recently Castillo-Manzano and Lopez-Valpuesta [32] analyze the check-in problem from the sociodemographic factors that influence passengers the decision of using one type of check-in facility over another (e.g., check-in counters or common-use self-service desks).

The work presented in this chapter differs from the aforementioned review in the sense that it performs the allocation of check-in counters through an approach that combines an analytical approach using evolutionary algorithms with simulation that allows modeling the dynamic and stochastic characteristics of the system under study. The combination of a deterministic solution with the stochastic elements stressed in the simulation model provides a more robust and reliable solution than the one that could be achieved with the sole use of the evolutionary or the simulated approach. The applicability of this approach is validated by using information from a real airport terminal provided under a confidential agreement between the authors and the airport.

4.2 Technical Approach

The problem consists of performing the check-in desks allocation satisfying a series of rules provided through a contract between the airlines that use the counters and the airport. These rules are classified as hard or soft.

The *hard rules*, i.e., rules that cannot be broken under any circumstances must be satisfied when the allocation is performed. On the other hand, it is desirable for *soft rules* to be satisfied but they can be broken when there is no other available option. It is important to mention that the violation of soft rules would impact on the perception of quality by the passengers inside the terminal, so it is an important factor to be taken into account by the company that performs the allocation.

4.2.1 Hard Rules

The following are the mandatory rules:

1. **Overlap Verification:** The current allocation must always be aligned with the allocation of the previous month in order to avoid having allocated the same check-in desks or nearby for the last flights of the previous month and for the first flights of the current month.
2. **Balanced Loads:** Allocating flights to check-in counters will consider an aspirational usage of 20 % for each of areas A, B, C, D, and E. The acceptable deviance is 1 % on a daily basis and 5 % on each 2-hours window. This restriction means that the terminal is used in a balanced way, therefore the usage of the areas is maximized with an increase of the passengers perception of quality. Figure 3 illustrates the different zones of the terminal area under study.
3. **Number of Desks:** Allocating flights to check-in counters will consider a minimum standard of one check-in counter per 45 passengers. This restriction has been defined to provide a good quality of service to the passengers at the check-in process, but a higher limit of 5 counters per airline will also be established.
4. **Sorting Hall:** Since there are two different sorting areas, the airport authority has established that the allocation of check-in desks should be done in such a way that the allocation of check-in desks should be done in such a way that no desks for the same flight use different sorting lines, as there is a high risk of lowering the efficiency of the baggage transport process and also of ending on a different flight. Figure 4 is a diagram of the two baggage conveyors systems. In this system

Area A	Baggage - Hall D (1-4)	QR/KU/AH/CY/TK/WY/KC																				BK SPSVCS			
		101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121			
Area B	Option Hall-C	Area - A																							
		142	141	140	139	138	137	136	135	134	133	132	131	130	129	128	127	126	125	124	123	122	XBAG	Cash	
Area C	Baggage - Hall C	GF/SQ/CZ/W5/AZ/TD/D3																							
		UA/9W/MH/JU/QS/BG/OA/CA/KWRB/TU/BI/JJ																				DN SPHL			
Area D	Baggage - Hall C / D	Area - B																							
		242	241	240	239	238	237	236	235	234	233	232	231	230	229	228	227	226	225	224	223	222	Emerg C	XBAG	Cash
Area E	Baggage - Hall C / D	AF/KL/UN/D9/KE/8U/BT/MS/TG/B8/7D/KQ when one flight operates.																				DN SPS / P50			
		DL/BA/CX/RJ/S7/RO																				DN MAS			
Area A	Option Hall-C	Area - C																							
		301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	Emerg C	343	
Area B	Baggage - Hall C	Area - D																							
		342	341	340	339	338	337	336	335	334	333	332	331	330	329	328	327	326	325	324	323	322	Emerg C	XBAG	Cash
Area C	Baggage - Hall C / D	LH/O S/LX/SK/ME/SV/U6/UB/ZB																				DN FS			
		A1/PAK/IR/KQ when two flights operates																				DN ABC			
Area D	Baggage - Hall C / D	Area - E																							
		401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	OLFT	XBAG	Cash
Area E	Baggage - Hall C / D	Area - E																							
		442	441	440	439	438	437	436	435	434	433	432	431	430	429	428	427	426	425	424	423	422	XBAG	Cash	
Area A	Baggage - Hall C / D	IC/HY/ET/UU/SU/N9/UE																				DN Stores			
		V S/J9/J2/RA/XY/AS/LN/NL/PS/IY/SD/VV/IT/4Q/GW																							
Area B	Option Hall-C	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	OLR	MHB	Mercator
																						543			

Fig. 3 The check-in zones in the terminal under study

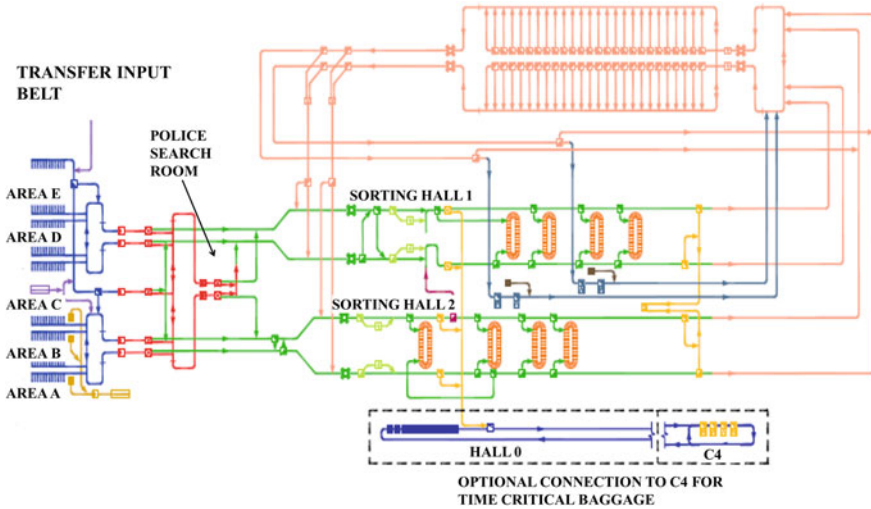


Fig. 4 Layout of the baggage handling system inside the terminal

areas A and B are processed in a different hall from areas C, D and E, therefore allocation must consider the boundary between these areas in order to avoid the allocation of a flight that could use both baggage systems.

4.2.2 Soft Rules

These rules are directly associated with the quality perceived by passengers; therefore in order to get a good evaluation of the quality of service, these rules should be largely satisfied.

1. Optimized Queuing/Circulation Areas for heavy flights: The allocation will avoid placing more than 3 heavy flights on the same row island during any given one-hour window. If this restriction is not satisfied, there is a high risk of congestion inside the island or row with the corresponding perception of poor quality on the part of the passengers.
2. Optimized Queuing/Circulation Areas for any flights: The allocation will avoid placing 2 flights on consecutive counters in any given one-hour window. The recommended practice is to leave at least one counter free between two flights for reasons of redundancy and flexibility. This practice is common in the terminal but the airport authority claims that better ways of allocation should be explored in order to make more efficient use of all the available resources, as this practice reduces the capacity of the check-in resources.
3. Airline Preferences: The preferences of the airline for their flights to be allocated to specific rows or fixed desks are officially taken note of during meetings and

the solution will consider their requests, but only after complying with all the allocation rules. It is desirable and common practice for companies from the same alliance to be allocated in the same area most of the time, therefore an efficient allocation will give priority of use to those airlines belonging to alliances inside the terminal islands.

4.3 Constraint Satisfaction

The first step of the methodology is the constraint satisfaction problem. This problem performs a static allocation based on the flight plan provided by the airport. The allocation algorithm performs an allocation of the planned flights taking into account the following constraints:

1. There is no overlap between flights
2. Counters are opened 3 h in advance
3. It calculates the number of counters needed in a base of 45 pax/counter
4. It leaves one check-in desk in between flights
5. The flights are allocated in the corresponding sections of the check-in area, so that the baggage does not end up in a different baggage hall
6. It randomly allocates check-in desks, trying to distribute the flights uniformly (load balance).

The allocation for the flights is performed sequentially in time slots, taking into account the aforementioned constraints as it is done in common practice. It takes every flight at a time and looks for the corresponding available time slot that satisfies the restrictions and, once allocated, it continues with the next flight on the allocation list. After all the flights are allocated an initial solution is obtained.

In order to get a variety of solutions for the evolutionary algorithm, a random selection of flights is performed every time the allocation algorithm is run. Using this approach a population of initial solutions is generated.

Once the initial solutions are generated, the next challenging task is the transformation of the solutions into vectors with the information that will be used by the evolutionary algorithm for improving the initial solutions.

4.4 Chromosome Representation

One of the key tasks in this approach is the proper representation of the solutions in the form of a vector of information. The representation will significantly influence the performance of the evolutionary algorithm. Every field of information holds key information that is useful for the performance of the operations of the evolutionary algorithm. In this chapter, the vectors have been defined as follows:

1. ID: Flight Identifier (string)
2. CI_OT: check-in Desk Opening time (min)
3. CI_CT: check-in Desk Closing time (min)
4. I_C: Initial check-in Counter (integer)
5. F_C: Final check-in Counter (integer)
6. Soft: check-in Allocation Soft Rules (Integer)
7. Hard: check-in Allocation Hard Rules (Integer)

Field 1 holds the Identifier of the corresponding flight. This field is used for keeping track of the corresponding flight. Fields 2 and 3 refer to the time the check-in counters are open and closed, respectively. Fields 4 and 5 provide the information about to which desk numbers are used. The last two fields (Fields 6 and 7) hold the information of the number of check-in desks needed to satisfy the hard and soft rules. They are necessary to identify if a counter is left or not in between flights.

4.5 Crossover Operations

Crossover is the main operation used for improving the current solutions. Crossover is performed in such a way that the feasibility of the new generated solution is maintained.

The crossing will be performed between chromosomes or elements of two current solutions (SolA and SolB) and it will perform the crossing between pairs. Figure 5 illustrates the crossing process.

The crossing is performed in the locus associated with the check-in desks being used, I_C and F_C , see Fig. 5. The reason is that the timeslots where the check-in is performed cannot vary, therefore only the counters will be the ones that will give variability to the generated solutions. The light blue color elements of the solutions (offspring) are the ones that have been changed by the crossover operators.

In order to maintain consistency in the generated solutions, the algorithm will verify three aspects of the new solution:

1. The crossing is performed between pairs that must use the same timeslot:

$$CI_{OT_A} = CI_{OT_B} \quad (3)$$

$$CI_{CT_A} = CI_{CT_B} \quad (4)$$

In order to choose a proper candidate for the crossing, the algorithm will take one element of the *SolA* and randomly choose another one from *SolB*. After the selection of the element of *SolB* it will verify that its timeslot corresponds to the same one as the element from *SolA*. If that is not the case it will take another one until a feasible one is found.

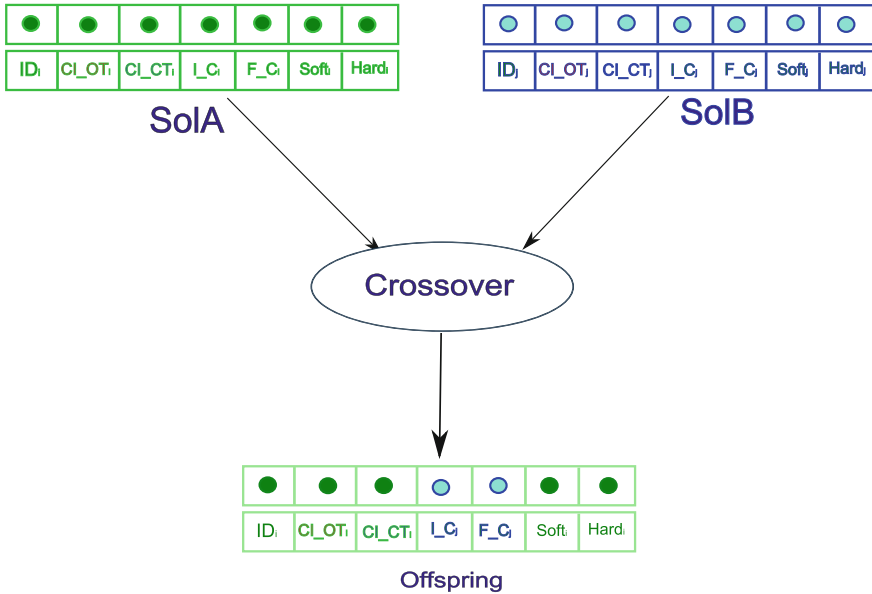


Fig. 5 Crossover between solutions

2. The crossing does not hinder previous or future flights.

It may be the case that the previous restriction is satisfied but the time and desks used for the allocation overlap previous or future allocations. So in order to avoid this situation the algorithm checks that this situation is avoided. Figure 6 illustrates the potential conflicts that may be encountered when the crossover is performed.

This example is schematized in Fig. 6. The counters of *Flight1* from *SolA* are swapped with the ones from *Flight4* of *SolB* as both flights have the same check-in time window. If we focus on the new *SolB* generated (right-hand side of Fig. 6), we can see that *Flight4* is in conflict with *Flight2*, while in the case of *SolA* there are no conflicts.

In order to avoid these types of conflicts, a procedure has been coded for the crossover operation. It compares the allocation performed against all the different elements of the current solution that fall within a time window of $[open_time - 180, open_time + 180]$. This comparison ensures that the previous allocations do not conflict with the current allocation, and at the same time that the timeslot of the present allocation does not conflict with a future one. If the allocation is conflict free, then the allocation is allowed and performed. However, if the time windows overlaps (a potential conflict) with some allocation, then the used check-in desks are verified for overlapping, if there is no conflict then the allocation is allowed. On the other hand when both conditions occur (time conflict and desks used overlapped), the crossing is not performed.

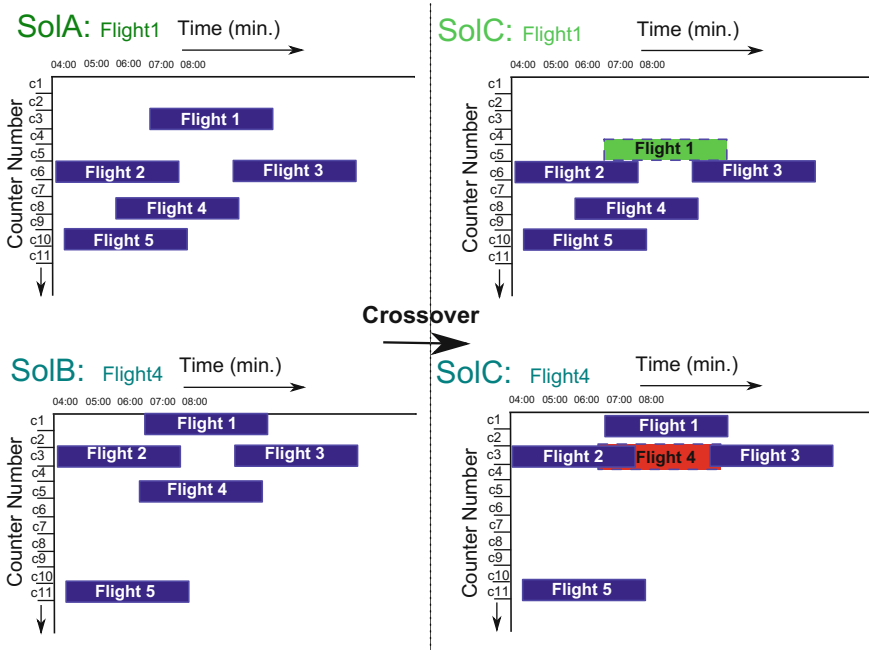


Fig. 6 Potential conflicts

3. Hard rules consistency

Finally, the crossover will evaluate if the allocation does not violate the hard rules concerning the number of desks needed (*Field7*). If the new allocation does not violate the minimum number of desks needed it will be kept as a feasible allocation. If the solution does not leave a desk in between flights, it will be kept as a feasible solution and later on this situation will be evaluated to see whether or not it affects the fitness of the new solution.

4.6 Objective Function Evaluation

During the performance of the evolutionary algorithm, once a feasible solution is generated, it is evaluated on a static basis using an objective function. This function evaluates the fitness of the solution by calculating several factors that make up the final value assigned by the function. These factors must have a direct impact on the LOS indicators.

The function used by this approach uses four parameters but it is not only restricted to those values. The analyst could extend the formula to include more parameters depending on the particular case of the airport in question. For the case presented

in this chapter, the function $F(v1, v2, v3, v4)$ is made up by a linear form of the 4 parameters:

$$F = a_1v1 + a_2v2 + a_3v3 + a_4v4 \quad (5)$$

where:

v1: is a factor that measures the number of flights in the solution that does not respect the 1-check-in desk in between flights.

v2: is a factor that evaluates the balance loads for the solution in accordance with the policies imposed by the airport.

v3: evaluates the number of heavy flights in the same island during 1-hour window.

v4: evaluates the distribution of flights in the islands of the check-in area.

a_n : is the weight of the corresponding factor.

In this case study of the different weights were kept at the value of 1 for illustrative purposes but the parameters can be changed to assign a different priority to one or several parameters over other ones. These priorities would drive the selection of the different feasible solutions, depending on the airport's requirements.

The evolutionary process is performed using the values of the objective function to calculate the goodness-of-fit of the different allocations and the selection process is carried out based on those values. Using this approach, the solutions are incrementally improved until a stop condition is satisfied. This condition is determined arbitrarily.

4.7 Simulation-Based Improvement

A pool of potential solutions for the allocation problem is obtained from the evolutionary algorithm to be later tested using a simulation model of the facility under study. This evaluation will provide a better estimation of the quality levels that can be achieved in the real system.

It is important to mention that certain requirements are desirable for the simulator in order to have the best evaluation of the quality indicators, i.e.:

- Agent-based so that the interaction between entities is more approximated to reality.
- High-description level; the more accurate the better is the evaluation.
- The model must allow interaction between agent-agent and agent-objects, so that it is possible to determine differences between relative positions among agents and objects.

There are some simulators in the market that satisfy these requirements [17], thus the methodology can be implemented making use of the one that suits best the objective of the study.

The use of the simulated scenario makes it possible to test the potential best solutions in a close-to-real environment. Sometimes it happens that solutions do

Table 2 Departure flight schedule

Airline	Passenger	Departure flight	Departure time
AIC	90	AI 0975	7:10
UAL	220	UA 0807	8:00
CHH	100	CH 9999	8:20
RNA	110	RN 0604	8:00
PIA	120	PI 0414	7:30
AFR	130	AF 8866	10:00
CSN	140	CS 0582	7:00
DLH	90	DL 0853	11:00
KLM	90	KL 0815	10:00
ABQ	90	AB 7777	12:00
SAI	90	SA 0570	14:00
NAX	90	NA 0835	14:00
SWR	285	SW 0847	10:20
ROT	306	RO 0705	18:00
BBC	120	BB 8888	22:00
AUA	116	UA 0221	21:10
KQA	314	KQ 0432	18:30
BAW	206	BA 0530	15:40
AFL	337	AF 0650	21:20
KLM	120	KL 0814	15:20

not perform well in the real system once they are implemented. The latter could be caused by some obstacles present in the facilities (e.g., big columns, trolley stations, etc.) that cause a potential good solution not to be such in reality because of congestion generated by pax–pax or pax–object interaction. Other causes are the emergent dynamics due to interactions of the entities and these can be easily observed in a terminal during congestion or during a disruption.

This methodology has been used to develop an initial solver for the check-in desk allocation for the airport. The initial approach will be used to evaluate the feasibility of the approach and, once it has been validated as a decision support tool, it will be extended to an operational level.

An initial flight plan has been used for testing the approach and its implementation in the simulated environment. Table 2 presents the flight plan used for the example presented here.

The titles of the columns are self-explanatory. Although Table 2 was not the actual flight plan, it is sufficient to test and validate the approach presented in the work.

4.8 Initial Solution

The set of initial solutions is generated for hundred desks as it has been explained in Sects. 3 and 2.1. A graphic representation of the solution on one airplane is depicted

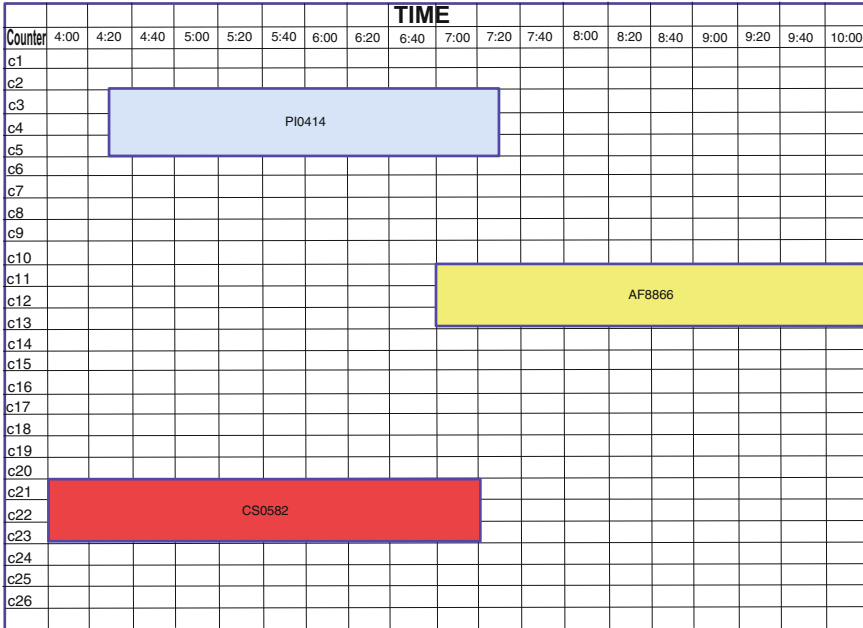


Fig. 7 Partial representation of an initial solution

in Fig. 7. The horizontal axis represents time and the vertical one represents the check-in desks used. Figure 7 exemplifies the initial solutions for 3 flights. The main outcome from the initial phase is a population of feasible solutions that are generated considering the different requirements for the allocation problem in question.

4.9 Chromosome Encoding and Evolution

The solutions generated in the initial phase are encoded as explained in Sect. 4.9 to start the evolutionary algorithm.

Table 3 gives an example of one encoded solution. The first column entitled *Flight* gives information about the Flight number to be allocated. The *PAX* column provides the information about the number of passengers registered for the flight. The following two columns, called *DeskIN* and *DeskEND*, store the information about desks that have to be allocated for the corresponding flight. The last two columns, *OpenTime* and *CloseTime*, give the information for the opening and closing time of the corresponding flight. For example, the first row means that flight *CS0582*, which has 140 passengers, will use the desks from 21 to 24 and they will be open from 4:00 am until 7:00 am.

Table 3 Encoded solution

Flight	PAX	DeskIN	DeskEND	OpenTime	CloseTime
CS 0582	140	21	24	4.00	7.00
AI 0975	90	61	63	4.17	7.17
PI 0415	120	1	3	4.50	7.50
UA 0807	220	71	75	5.00	8.00
RN 0604	110	91	93	5.00	8.00
CH 9999	100	81	83	5.33	8.33
AF 8866	130	11	13	7.00	10.00
KL 0815	90	41	43	7.00	10.00
SW 0847	285	84	88	7.33	10.33
DL 0853	90	31	33	8.00	11.00
AB 7777	90	51	53	9.00	12.00
SA 0570	90	61	63	11.00	14.00
NA 0835	90	71	73	11.00	14.00
KL 0814	120	51	53	12.33	15.33
BA 0530	206	31	35	12.67	15.67
RO 0705	306	91	95	15.00	18.00
KQ 0432	314	21	25	15.50	18.50
UA 0221	116	11	13	18.17	21.17
AF 0650	337	41	45	18.33	21.33
BB 8888	120	1	3	19.00	22.00

For each flight, whether or not they respect the check-in desk in between flights is evaluated. The number of check-in desks open is calculated based on the rule that one check-in desk must be open for every 45 passengers. As an example, see the initial flight on Table 3, for flight CS0582 with 140 passengers, for this rule to be respected three check-in desks should be opened.

The balance load in each area A, B, C, D, and E is also calculated for the solutions. This balance load depends on the layout of the airport and the internal policy of the airport authority. The computation divides the check-in desks into zones and then calculates how many flights have been allocated to which zones.

The third parameter needed to compute the objective function is the one for heavy flights. In this case the zoning is performed based on the time of the allocation and the correspondent zone, this way it is possible to penalize the allocation of heavy flights in the same island during a time period.

The fourth value is calculated taking into account the number of flights in each island, so that the flights are evenly distributed. Thus, the space in between flights is optimized and the level of service improved.

Using the objective function, the selection among the siblings solutions is made in such a way that the value of the objective function is progressively improved.

Table 4 Values of the cost function

Iterations	Avg. cost	Value
1	1.36	
10	1.012	25.58823529
20	0.6714	50.63235294
30	0.468	65.58823529
40	0.4228	68.91176471
45	0.3542	73.95588235
60	0.15068	88.92058824
624	0.145276	89.31794118
891	0.13397	90.14926471
3858	0.126434	90.70338235
6605	0.125325	90.78492647
6914	0.112363	91.73801471
31,600	0.11074	91.85735294

Table 4 illustrates how the objective function is improved as the algorithm evolves. Meanwhile, Fig. 8 illustrates the evolution of the cost function versus the number of iterations.

4.10 Performance Evaluation in a Virtual Environment

The model of the terminal area has been developed using a general-purpose simulation software called SIMIO [17]. The simulator has been selected for these studies as it possesses most of the necessary characteristics previously mentioned. There are

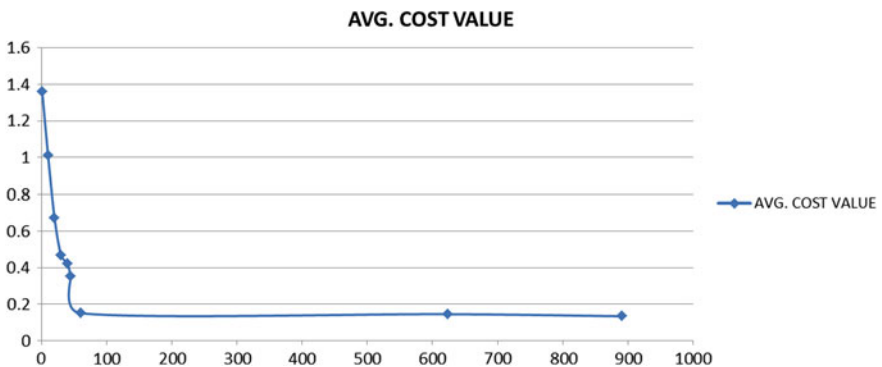


Fig. 8 Convergence of the cost function

other tools that possess better characteristics of agent interaction but for the sake of illustration the authors consider that SIMIO is good enough for evaluating different interactions. However, if the reader wants to emphasize in the interactions, other tools could be more suitable.

The simulation model represents the layout of the area under study, which comprises 100 check-in desks in an area of 170×70 square meters (see Fig. 9). With the help of the virtual environment, the LOS can be evaluated alongside other performance indicators that are important for assessing the correct management of the area under study.

Figure 9 presents the simulated layout of the terminal area together with some data and notation below the model. These notations illustrate how the different sections of the terminal have been identified in order to dynamically calculate the LOS over time. In the terminal area there are 5 sections, namely areas A, B, C, D and E; with 5 subsections each. Therefore, a total of 25 areas were monitored in the study.

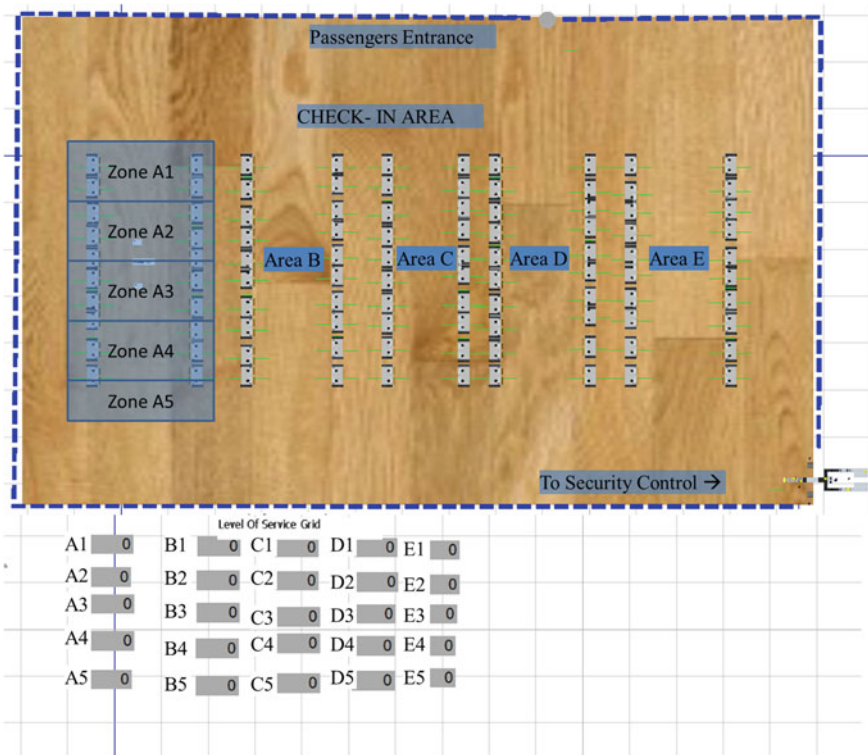


Fig. 9 The simulated environment

4.10.1 Stochastic Parameters

The advantage of using a simulation model for assessing the different solution, is that it is possible not only to test the proposed allocations in a close-to-real scenario but also different characteristics of the passengers can be added to the model, thus making the evaluation more reliable than the evolutionary algorithm by itself. Some examples of the dynamic parameters that have been added to the model are presented in Table 5. These parameters are just an example of common values, however, the reader should perform a data acquisition exercise to come up with the right values for its particular study.

Different configurations provided by the allocation algorithm have been evaluated using the simulation model. Table 6 is useful to illustrate the differences between the initial allocation and the final one.

It was considered that zones with congestion problems where those areas with values under $3 \text{ m}^2/\text{pax}$ to be considered as critical. As presented in Table 6, initially, there were several areas where congestion could be perceived as the LOS indicators illustrate; namely areas E1, C1, B1, and E2. This configuration presents the worst LOS value in area E1, with a value of $1.1113 \text{ m}^2/\text{pax}$. When the allocation provided by the evolutionary algorithm is evaluated in the simulation environment, the LOS

Table 5 Parameters of the simulation model

Concept	Type
Check-in desks processing time	Lognormal (0.709, 0.154) with a minimum of 1
Arrival profile	Triangular (−180, −90, −40) min
Passenger speed	Uniform (0.4, 0.9) m/s
Check-in area	$170 \times 70 \text{ m}^2$
Passengers desk	1 person/desk
Passengers do not show earlier than 3 h prior to departure	

Table 6 Level of service indicators

	Critical zones	Minimum value m^2/pax
Initial	Zone E1	1.1113
	Zone C1	1.6585
	Zone B1	1.8214
	Zone E2	1.9767
Final	Zone E1	2.3415
	Zone D1	2.1223
	Zone C1	2.5617

indicators show a significant improvement, see Table 6. For the new allocation, there were only three critical zones and the minimum value was perceived in zone D1, with a value of $2.1223 \text{ m}^2/\text{pax}$. In conclusion, with the use of the evolutionary approach an optimized allocation for desks was found. Due to this, the congested areas in the terminal could be reduced, thus providing a better allocation than the one that can be achieved by manual allocation.

5 Discussion

The present work introduces a new methodology that combines an evolutionary approach with simulation to perform the check-in desk allocation for optimizing the LOS indicators in an airport terminal. The strength of the methodology lies in tackling the problem in such a way that it is possible to take into account deterministic and stochastic characteristics resulting in a more robust and reliable solution. The algorithm uses an evolutionary approach to improve the initial allocation of check-in desks taking into account the policy restrictions imposed by the airport. Once good solutions from the mathematical standpoint are obtained, they are further improved using a simulated environment that takes into account other elements of the problem such as physical locations, queue policies, passenger arrival profiles, efficiency of the personnel, etc.

The results show that the methodology is robust enough to provide good solutions with few iterations and the reliability of the solutions is improved with the simulated model. In addition the methodology is flexible enough to include more constraints either in the evolutionary algorithm or the simulation model in order to provide solutions that are in line with the objectives of the airport. The methodology that has been presented can be easily implemented in other terminals or in other industries following the guidelines and suggestions presented here but the simulation model must be developed for the corresponding terminal. In future implementations a metamodel can be integrated into the evolutionary algorithm in order to develop a stand-alone tool for decision-making. This methodology would be recommendable for the planning phases of new passenger terminals or for assessing the current performance and to evaluate future implementations that involve policy restrictions.

Acknowledgments The authors would like to thank the Aviation Academy of the HvA, the Mexican Council for Science and Research (CONACYT) and the Popular Autonomous University of Puebla for their support for this work.

References

1. BOEING. Current market outlook 2013 2032. commercial airplanes market analysis, 2013.
2. EUROCONTROL. Challenges of growth: summary report, 2013.
3. AIRBUS S.A.S. Global market forecast. future journeys 2013 2032, 2013.

4. A.R. Correia, S.C. Wirasinghe, and A.G. Barros. Overall level of service measures for airport passenger terminals. *Transportation Research Part A: Policy and Practice*, 42(2):330–346, 2008.
5. R. Neufville, A.R. Odoni, Belobaba P.P., and T.G. Reynolds, editors. *Airport Systems: Planning, Design, and Management*. McGraw-Hill Education: New York, Chicago, San Francisco, Lisbon, London, Madrid, Mexico City, Milan, New Delhi, San Juan, Seoul, Singapore, Sydney, Toronto, 2013.
6. S. Delahaye, D., Puechmorel. *Modeling and Optimization of Air Traffic*. Wiley-ISTE, 2013.
7. F. Glover and G.A. Kochenberger. *Handbook of Metaheuristics*. Kluwer Academic Publishers, 2003.
8. KingLoong Shiu and K.Y. Szeto. *Self-adaptive Mutation Only Genetic Algorithm: An Application on the Optimization of Airport Capacity Utilization*, volume 5326 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2008.
9. Thomas Black. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, Oxford, UK, 1996.
10. D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.
11. M. Affenzeller, S. Winkler, A. Beham, and S. Wagner. On the influence of selection schemes on the genetic diversity in genetic algorithms. In Roberto Moreno-Daz, Franz Pichler, and Alexis Quesada-Arencibia, editors, *Computer Aided Systems Theory - EUROCAST 2009*, volume 5717 of *Lecture Notes in Computer Science*, pages 777–784. Springer Berlin Heidelberg, 2009.
12. J.R. Koza. *Genetic Programming*. MIT press, 1992.
13. Z. Michalewicz. *Genetic algorithms + Data Structures = Evolution Programs*. Springer-verlag, 1992.
14. H.P. Schwefel. *Evolution and Optimum Seeking*. Wiley, New York, 1995.
15. J. Banks, Carson J.S., and B.L. Nelson. *Discrete Event System Simulation*. Prentice Hall, Upper Saddle River, NJ, 1996.
16. K. Jensen and L.M. Kristensen. *Coloured Petri Nets: Modeling and Validation of Concurrent Systems*. Kluwer Academic Publishers, 2009.
17. SIMIO. Simio homepage, 2002.
18. ARENA. Arena homepage, 2002.
19. F. Longo. Advances of modeling and simulation in supply chain and industry. *Simulation*, 87(8):651–656, aug 2011.
20. Juan Ignacio Latorre, Emilio Jiménez, and Mercedes Pérez. The optimization problem based on alternatives aggregation petri nets as models for industrial discrete event systems. *Simulation*, 89(3):346–361, March 2013.
21. S. Yan, C.Y. Shieh, and M. Chen. A simulation framework for evaluating airport gate assignments. *Transportation Research Part A: Policy and Practice*, 36(10):885–898, 2002.
22. R.E. Kazda, A.; Caves. *Airport Design and Operation*. Emerald, Inc., 2007.
23. B. Giuseppe and A. Genovese. A mathematical model for the optimization of the airport check-in service problem. *Electronic Notes in Discrete Mathematics*, 36(0):703–710, 2010. {ISCO} 2010 - International Symposium on Combinatorial Optimization.
24. C.I. Hsu, C.C. Chao, and K.Y. Shih. Dynamic allocation of check-in facilities and dynamic assignment of passengers at air terminals. *Computers & Industrial Engineering*, 63(2):410–417, 2012.
25. M. Parlar, B. Rodrigues, and M. Sharafali. On the allocation of exclusive-use counters for airport check-in queues: static vs. dynamic policies. *OPSEARCH*, 50(3):433–453, 2013.
26. M. Parlar and M. Sharafali. Dynamic allocation of airline check-in counters: A queueing optimization approach. *Management Science*, 54(8):1410–1424, 2008.
27. R.A. Littler and D. Whitaker. Estimating staffing requirements at an airport terminal. *Journal of the Operational Research Society*, 48(2):124–131, 1997.
28. H.W. Chun and R. W.T. Mak. Intelligent resource simulation for an airport check-in counter allocation system. *Trans. Sys. Man Cyber Part C*, 29(3):325–335, August 1999.

29. Y. Park and S.B. Ahn. Optimal assignment for check-in counters based on passenger arrival behaviour at an airport. *Transportation Planning and Technology*, 26(5):397–416, 2003.
30. S. Yan, C.H. Tang, and M. Chen. A model and a solution algorithm for airport common use check-in counter assignments. *Transportation Research Part A: Policy and Practice*, 38(2): 101–125, 2004.
31. N.M. van Dijk and E. van der Sluis. Check-in computation and optimization by simulation and ip in combination. *European Journal of Operational Research*, 171(3):1152–1168, 2006.
32. J.I. Castillo-Manzano and L. Lopez.-Valpuesta. Check-in services and passenger behaviour: Self service technologies in airport systems. *Computers in Human Behavior*, 29(6):2431–2437, 2013.