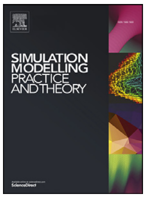
Simulation Modelling Practice and Theory 75 (2017) 146–164

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat



A simulation approach to modelling baggage handling systems at an international airport

Juan Pablo Cavada[a], Cristián E. Cortés[a,*], Pablo A. Rey[b]

[a] Civil Engineering Department, FCFM, Universidad de Chile, Chile
[b] Industrial Engineering Department, FCFM, Universidad de Chile, Chile

ARTICLE INFO

Article history:
Received 16 December 2015
Revised 4 January 2017
Accepted 5 January 2017
Available online 25 April 2017

Keywords:
Airport operations
Baggage handling
Microscopic simulation

ABSTRACT

The baggage handling system is a critical component of an airport's operations, coordinating many different elements and agents in different areas of the facility. Due to the complexity of their interrelationships, an effective analysis of the impact of different operating strategies on the system must consider these elements and agents not in isolation from each other but rather as an integrated whole. This paper presents a microscopic simulation model for a baggage handling system that fully integrates all baggage-related subsystems. These include passenger arrival to check-in queues, baggage check-in, security screening, sorting, transport to the aircraft and loading. Under this approach, not only the individual subsystems but also their interactions can be simulated and studied. The proposed simulator is applied to the case of Santiago International Airport in Chile where passenger demand has grown beyond the existing baggage handling system's operating capacity. The principal contributions of this study are the extension and adaptation of a vehicle traffic simulator software to the baggage handling problem and the development of a platform that models baggage handling as an integrated unit. The tool is applied to the analysis of the overall system and its components under a number of different real-world scenarios. In this application, the movement of bags and their interactions with the rest of the system are simulated in great detail for a given period. This level of granularity permits the simulator to analyse accurately the effects of different scenarios and how they are propagated through the system.

1. Introduction

The baggage handling system (BHS) at an airport is a key process in an airport terminal's landside activities, impacting both the airlines and their passengers. Poor management of the system may result in luggage arriving damaged, late or never, with consequent harm to the airlines' public image. This is especially true during the high season at peak hours when the number of flights a BHS must process at the same time rises dramatically, in some cases to levels beyond the terminal's design capacity.

Most major world airports today have automated baggage handling systems. At the international airport terminal in Santiago, Chile (SCL by its airport code), the BHS, though relatively advanced, is not completely automatic. Movement of baggage

---

* Corresponding author.
E-mail addresses: ccortes@ing.uchile.cl, jucavada@ing.uchile.cl (J.P. Cavada), ccortes@ing.uchile.cl (C.E. Cortés), pablo.rey.cl@gmail.coml (P.A. Rey).

http://dx.doi.org/10.1016/j.simpat.2017.01.006

within the terminal is automated but the sorting, distribution and aircraft loading operations are carried out manually. Baggage handling services at the airport are provided by a number of ground handler companies, one of which accounts for more than 80% of supply.

SCL's capacity was last expanded in 2001 and is currently estimated at 9.5 million passengers per year. However, with the explosion over the last 10 years in Chile's domestic air travel market, this figure was already surpassed in 2010 when more than 10 million passengers were carried. As of 2014, the annual number had risen to 16 million [1].

In this scenario the pressure on all of the airport's systems has reached unprecedented levels, leading to long passenger queues, delayed flights and a growing potential for damage or loss in the baggage handling process. The BHS in particular, which handles all of the terminal's luggage, is suffering under the strain. Yet no significant infrastructure investment is planned to remedy the situation in the short term, in part because priorities are focussed on the construction of a new terminal to handle 30 million passengers that is slated to open in 2019.

In light of this situation, there is a clear need to find ways of improving baggage handling at SCL in the short and medium terms through the optimization of operations, processes and usage of available resources in order to support the increased intensity of demand in an infrastructure scenario that has little prospect of upgrade at least for the next two years.

The present article presents a simulation platform for modelling the whole SCL baggage handling process from the moment a passenger joins the check-in counter queue to the loading of the baggage onto the aircraft. As will be seen in the following sections, the BHS is made up of multiple subprocesses that are strongly interrelated. This results in an overall process that is highly complex to model and calls naturally for the use of simulation as an approach to its analysis. The proposed platform will enable us to make a thorough study of BHS operations for both the domestic and international terminals, and identify its critical processes and their impact on the rest of the system.

The contributions of this work are twofold: 1) a simulation platform is developed that integrates various airport operation systems (counters, baggage conveyor network, baggage loading area, security screening, etc.) so that both the systems themselves as separate entities and their interactions can be studied, and 2) the simulation of the conveyor belts is built by extending a vehicle traffic microsimulation software.

From a practical application standpoint, this simulation tool generates evaluations of hypothetical operating scenarios based on real data whose results can be used to design operating protocols for both normal conditions and contingencies. The arrival baggage operation was not included in this study but it should be considerably simpler to address than the departure baggage operation that is the focus of our analysis.

The remainder of this paper is organized into six sections. Section 2 briefly reviews the state of the art in baggage handling system modelling for airports; Section 3 describes the workings of SCL's departure baggage handling area, focussing on the movement of bags from the check-in counters to the departing aircraft, as well as the adaptation of a traffic simulation software for simulating conveyor belts; Section 4 details the structure of the simulator; Section 5 sets out the results of a number of experiments in modelling realistic BHS operating scenarios; and finally, Section 6 presents our conclusions, the current state of the simulation development and some tasks for the future.

2. Literature review

Although many published works have attempted to model the different systems for airport terminal processes as separate entities, relatively few have tried to construct an integrated formulation of a BHS. Among the earliest efforts are [2] and [3]. The latter study analyzed various configurations for the design of baggage conveyor systems and tested a number of operating schemes, but due to the technological limitations of the time it was not possible to experiment with real scenarios.

Integrated models have, however, been developed for facilitating the decision-making process in airport planning and design [4–7]. Other models diagnose the global performance of an airport terminal or parts of one [8–10]. These formulations concentrate on passenger flow (check-in, customs, security screening, departure lounges, etc.), leaving baggage handling in the background.

As regards modern baggage handling systems, the consensus is that they are complex and difficult to model and thus require elaborate tools of analysis to be managed correctly [11–14].

Broadly speaking, the available models and tools for airport process analysis may be divided into three categories: macroscopic, mesoscopic and microscopic, depending on whether the level of detail is low, medium or high, respectively.

Macroscopic models are used primarily to support strategic decisions. They operate at a high level of aggregation and therefore omit much important detail [15]. Prominent among these is SLAM [8], an analytical model for estimating the approximate capacity of passenger terminal systems. It consists of a set of modules, one for each terminal area, that estimates capacity and expected service levels, and was used as the model base for the development of OPAL (Optimisation Platform for Airport including Landside) [16]. SLAM cannot, however, model the possible interactions between the different terminal elements, nor can it handle stochastic effects that might impact general system performance, and therefore fails to capture the full complexity of a terminal's processes.

Microscopic models, on the other hand, are designed to incorporate a large number of operating considerations, require much information to be implemented and tend to be expensive to develop. They are usually based on simulation and have been developed for specific airports or limited to certain terminal processes [17].

A well-known general microscopic model was developed by Abdelghany et al. [18], who propose an algorithm for planning which flights should be processed in each zone of an airport. They analyzed the impact of grouping the departures of

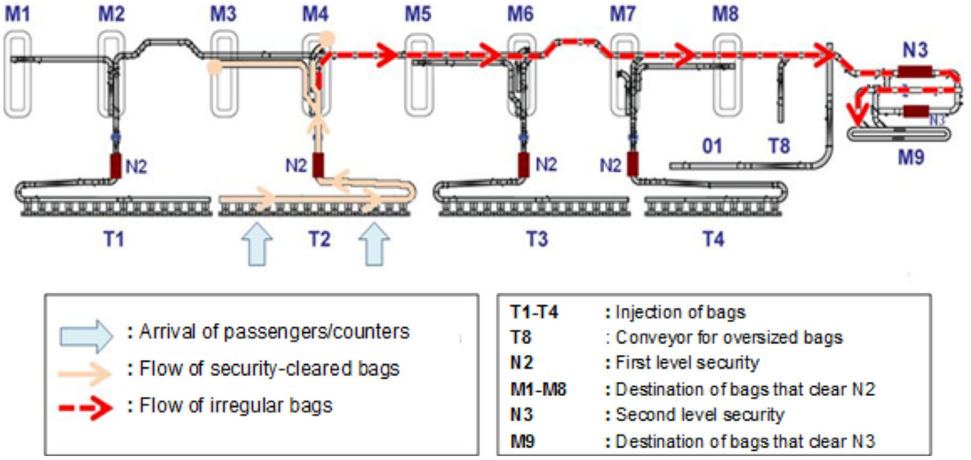


Fig. 1. Schematic diagram of BHS conveyor network, Santiago International Airport.

various similar flights within short time periods, generating peak (high load) periods alternating with non-peak (low load) periods. Although this creates better passenger transfer opportunities, it also results in congested baggage handling periods that alternate with periods of low BHS utilization. This variability causes additional difficulties for the planning problem. The approach was later extended by Ascó et al. [19], who addressed the airport baggage sorting station allocation problem in terms of airport geometry, flight service time, station capacity and distance to the aircraft gate.

Finally, the literature on mesoscopic modelling of baggage systems presents specific applications for particular airports. Savrasovs et al. [20] used a discrete-event simulation approach to model the check-in and baggage screening areas and the conveyors connecting them at Latvia's Riga Airport. Their objective was to determine the expected behaviour of the terminal in the face of possible growth in passenger flows. Lazzaroni [21] also employed simulation to create a detailed model of passenger flows at various terminal points (check-in, security screening, departure lounges and baggage claim) at Vancouver International Airport.

The approach developed in the present article incorporates various terminal systems as in the macroscopic models while also capturing a level of detail only found in the existing literature in microscopic models.

3. The Santiago's airport BHS and its modelling using traffic micro simulation

The baggage handling system at Santiago International Airport consists of a set of conveyors, diverters and carrousels that move passengers' bags from the check-in counters to the baggage carts, which are then loaded onto the aircraft. In this study we will consider the entire process from the arrival of a passenger at the airport terminal to the loading of the airplane, including related systems. The main components of the BHS are shown in Fig. 1.

Passengers enter the terminal and join the check-in queue for their respective flights. Each queue is associated with a set of check-in counters and thus may have passengers taking different flights. As each passenger is checked-in, their bags are individually deposited in special trays on the conveyor belt by the counter agent, who inputs the bags to the BHS controller system using a barcode gun. From that moment, the system knows the position of the bag and also which loading area carrousel it is destined for since this is determined by the counter it was checked in at.

At SCL there are 104 check-in counters grouped into four "T" subsystems. The counters in each T are connected to a common conveyor known as the *collector*. When a bag is inputted to the BHS controller, a bag injection order is sent to the collector conveyor. The bag then waits at the entrance to this conveyor until a space appears. The movement of a bag from the counter to the collector conveyor is automatic.

Upon entering the collector, a bag is carried along to the security screening scanner (N2) and then on to the baggage loading area. If anything suspicious is detected the bag is sent directly to a higher security area (carrousel M9); otherwise, it is sent to the loading carrousel associated with the counter where it originated.

The baggage loading area (BLA) is the place where all bags arriving at or departing from the terminal are handled. Each of the T counter subsystems feeds two carrousels where the bags are separated and deposited into carts for loading onto the aircraft. Running the length of the baggage loading area is a common conveyor, which collects suspicious bags from all T's and transport them to carrousel M9.

Once the bags for a given flight arrive at the appropriate loading carrousel, the rest of the process is essentially manual. A baggage handler assigned to the preparation of the flight locates the bags for that departure on the carrousel, loads them into the corresponding containers and records their final destination. The last container is taken out to the aircraft between 20 and 14 min before departure time, and any bag not loaded during that period remains behind. How many handlers are tasked to a particular flight and how long before departure the preparation process starts both depend on the flight's complexity.

Bags that for any reason exhibit some irregularity are redirected to carrousel M9 at the end of the baggage handling area. This will occur if either the system has lost the bag's identification (a tracking error) or the N2 security screening has flagged it as suspicious. In the latter case, it is checked by a more powerful scanner (N3). Once it has cleared all security checks it is returned to carrousel M9. There, a handler registers its arrival and places it on a cart to be sent to the appropriate loading carrousel (M1-M8) where the flight is being prepared or, if the flight is about to leave, directly to the aircraft. In some cases, bags sent to the high security zone may not be cleared in time to be loaded on the plane and will remain behind.

The normal flow of baggage through the system may be interrupted by an event or incident. The most common example is a bag jam at some point along the conveyors. A conveyor may also stop if a bag is "forced" onto it at the check-in counter without being correctly checked in or if the bag is wrongly positioned. Whatever the cause of the incident, the conveyor will remain stopped until system personnel resolve the problem. This may take several minutes depending on the accessibility of the location where the event occurred. The bag triggering the incident will automatically be diverted to M9 for examination.

Our simulation for modelling the baggage moving processes at SCL was developed using the traffic microsimulation software package Quadstone Paramics v6 (www.paramics-online.org) through its API (application programming interface) functionatities. This platform supports continuous-time simulation with a half-second time step for all conditions.

In a microscopic vehicle traffic simulation, a fixed road network is coded with a set of zones representing the origins and destinations of trips made by the different entities travelling to their desired destinations.

This dynamic is similar to a BHS, which is made up of bags that are transported by conveyors to a predefined destination. The fundamental equivalences in our model are then: "vehicles = bags" and "streets = conveyors". Also, the same mechanisms normally used by the simulator to assign vehicle destinations will be used here to assign those of the bags. The bags are created from the plugin (API) and not from an origin-destination matrix as often in microscopic traffic simulation applications. Once the bag was created starting from a specific zone (representing a specific counter entrance) it follows a fixed route towards a destination assigned also in the plugin corresponding the carrousel associated with the flight that bag belongs to.

There are, however, certain significant differences between the two systems that must be taken into account. First, to maintain the correct proportions in the Paramics simulator with the vehicles representing the bags, we had to scale the distances and speeds up by a factor of 10. Also, whereas vehicles on streets may at any moment be accelerating, stopping, bunching up, overtaking, or falling back or pulling away from other vehicles, bags in a BHS maintain a constant distance from each other and thus never either get closer together or further apart. The behaviour of the vehicles must therefore be modified so that they always move at the same speed as the conveyor over each section. We describe the details in Section 4.2.

In cases where conveyors join or divide, the Santiago BHS has well-defined rules regarding which conveyor branch has priority at the junctions. The arrangement was modelled using *traffic-actuated signal lights* that define priority in response to traffic conditions. For example, the injection of bags to the controller conveyor at the check-in counters is regulated by a light that constantly checks for approaching bags on the conveyor, switching to red if one is detected and not returning to green until it has passed.

Carrousels were modeled by defining streets with multiple (three) lanes and rewriting the lane-changing rules so that each vehicle takes the outermost lane possible, thus simulating bags piling up at the bottom of the carrousel. To ensure the capacity of the carrousel (where the bags are very close together) is the same as in the simulator (where the vehicles maintain a certain minimum separation) the carrousel is scaled up and the vehicle speed adjusted such that the numbers of vehicles and bags in the simulator are equal and the amount of time they take to go around the carrousel is the same. The way we model the carrousel operation by changing some aspects of the network topology, allows representing quite well the phenomenon observed in reality. Also, when the time comes for a bag to be removed from the conveyor, it is directly eliminated. This resembles the observed operation, in which handlers remove any bag that is passing through, regardless of the lane the bag is cycling on.

Finally, conveyor stoppages and bag jams were the easiest processes to implement. Paramics incorporates a wide selection of "incidents" that can affect vehicles. This functionality is called by defining, for each stoppage type, the probability of its occurrence by vehicle type and place it occurs.

In Fig. 2, two 3D animation snapshots of the simulation are shown, highlighting the counter area and one carrousel. The color of the bags represent different categories described later in Section 4.1.4.

One positive feature of the Paramics API's is its flexibility. In fact, through it we can control the simulation parameters and variables at any moment in order to achieve the desired representation. By using the microscopic simulator, we were able to obtain a great level of details in the representation of the physics behind the movement of bags.

Since Paramics controls each of the vehicles independently at all times, their respective speeds can be set instantaneously to the appropriate section speed without accelerating and can be stopped instantaneously without braking. This allows to directly model the movement of bags passing through a section comprising several subsections of different speed. The physics behind the movement of bags can be represented very accurately and based on real distances, features that were very useful for modelling semiautomatic systems such as the one coded in this work. Indicators associated with the use of capacity in carrousels are also easy to obtain from the available API functions.

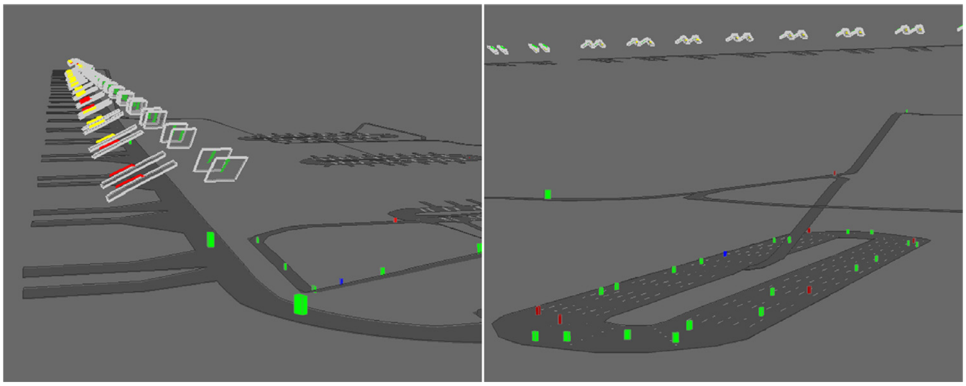


Fig. 2. Paramics 3D animation of the check-in counter area and one of the carrousels.

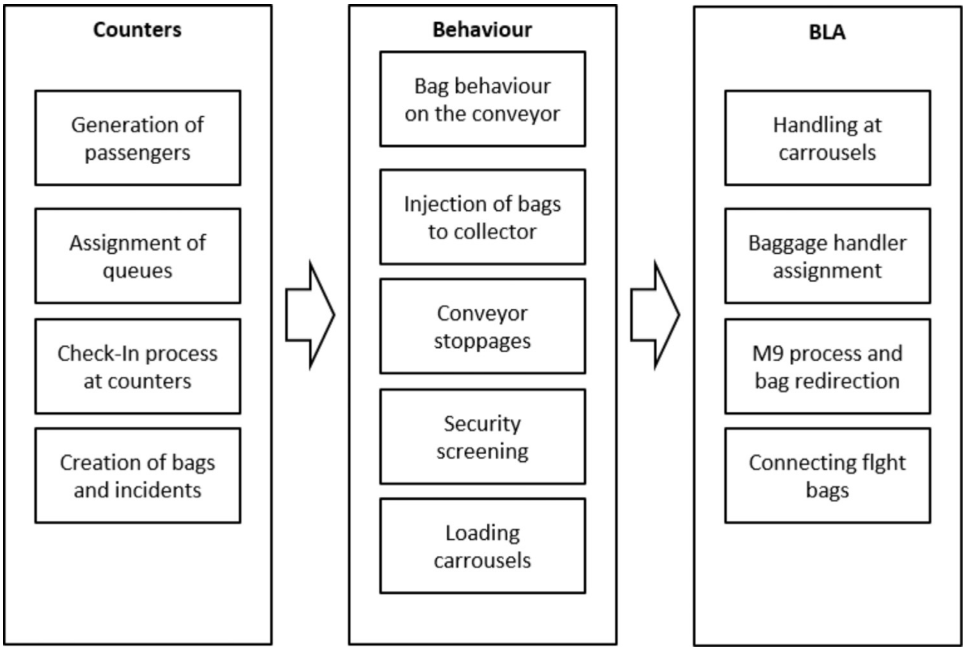


Fig. 3. Schematic diagram of simulator operation.

4. Simulation model elements

The simulator consists of three basic interconnected modules: check-in counters, behaviour and baggage loading area (hereafter simply “Counters”, “Behaviour” and “BLA”). The three are interconnected and operate in sync with each other. Each one consists of a number of routines as shown in Fig. 3. The modules consist in various routines developed in C++ and interact with Paramics through its API functions. The routines execute in response to events occurring during the simulation such as the arrival of a bag at a scanner, or at certain defined moments like the departure of a flight or the arrival of passengers at the terminal. In the subsections that follow, these modules and their routines are described in more detail.

4.1. Counters

The Counters module routines represent the generation of passengers and their arrival at the airport, the assignment of passengers to check-in counters, passenger at the counters and the injection of bags to the BHS. The objective of this process, flow-charted in Fig. 4, is the creation of bags in the simulator network. The routines in the module are described below.

4.1.1. Generation of passengers

In most cases we are studying specific days, so the number of passengers for a flight is determined directly from the passenger reservation forecast for that day. If this information is not available, historical data for the date and time are used.

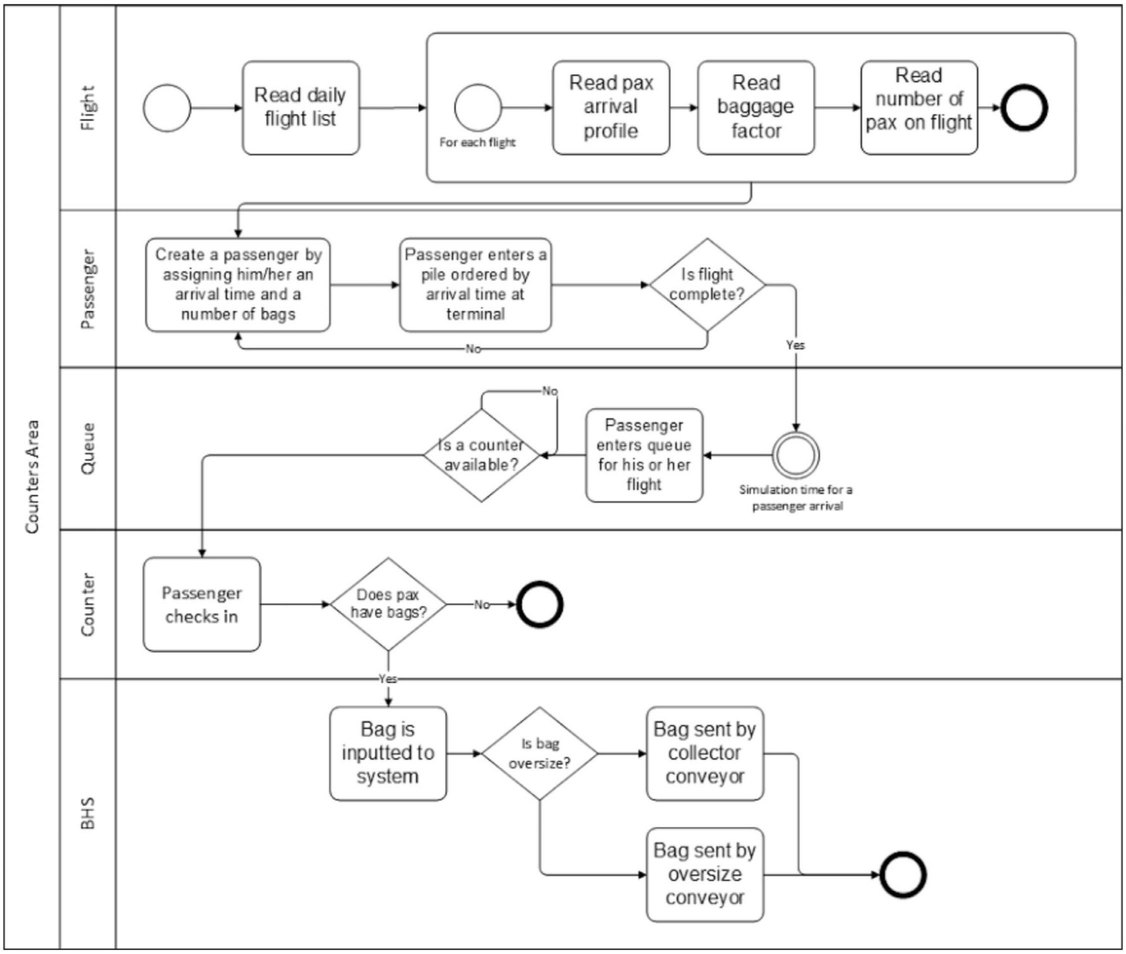


Fig. 4. Flow diagram of simulator counters module.

The arrival distribution of passengers varies considerably among different flights, days and time of the day. We build empirical distributions derived from real data provided by the company, corresponding to a histogram of passengers' terminal arrival times. The airlines build their own profile curves for representing the passengers' arrival distributions, which are updated month by month in a very rigorous way to be used for many other purposes. For flights where this data is not available estimates are made based on flights whose destinations and aircraft type are similar. The main reason for not using a theoretical distribution is that we should have fitted a different distribution for each flight, as passenger arrival behavior varies greatly with destination and departure time. This is illustrated by the example in Fig. 5, which shows four SCL passenger arrival profiles, two for a domestic destination (Punta Arenas) and two for an international one (Buenos Aires, Argentina).

For each passenger in a flight f we compute his arrival time with the following procedure: Let I be the number of time-intervals, $(t_i, t_{i+1}]$ be the ith interval and $p_i(f)$ the proportion of passengers historically arriving in the interval i. Let us call a_i the accumulated proportion of passengers arriving until the end of the interval i, that is $a_i = \sum_{j \le i} p_j(f)$. To compute the arrival time for each passenger we generate a random number u uniformly distributed between 0 and 1. The passenger will arrive in the jth interval if $a_j \le u < a_{j+1}$. Then we generate a second random number v uniformly distributed between 0 and $t_{j+1} - t_j$. The exact instant of arrival is interval $\tau = t_j + v$. That moment is identified at the start of the simulation for each passenger and the arrivals for the entire experiment can then be scheduled.

4.1.2. Assignment of queues

The BHS technology allows defining the destination carrousel of bags entering a specific counter. Given this, airlines define groups of counters that will serve a set of flights and passengers for these flights wait in a single queue to be checked-in. As the passengers arrive at the terminal they join the queue for the counters handling their flight. As mentioned above, once a passenger is generated, the information on his(her) corresponding flight, number of bags and arrival time is set.

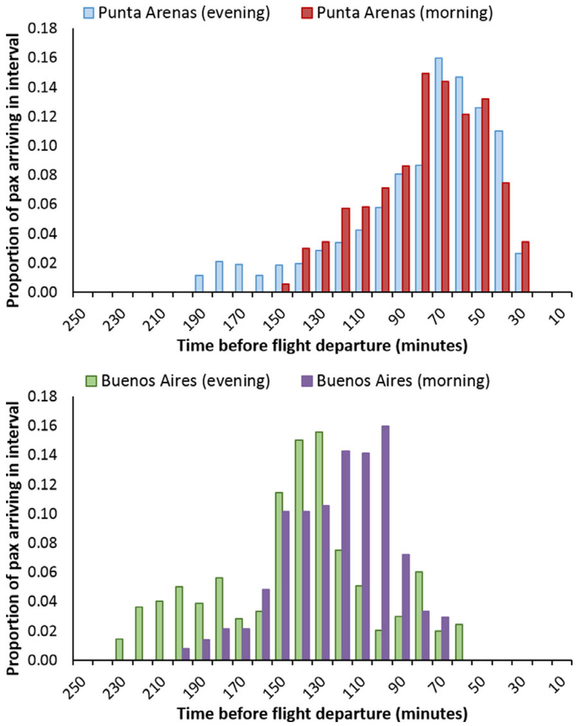


Fig. 5. Distribution of passenger arrivals at SCL check-in queues (January 21, 2011). *Source:* LAN Airlines.

By default, passengers in a given queue are sort by order of arrival. However, to speed up the process for flights whose check-in is about to close, a procedure mimicking what is done in practice was implemented. When a flight is taking off in less than a fixed time, this procedure search for passengers on that flight and still in queue, and make then skip positions to the first places of the queue. The queuing process continues from this point making these passengers the next ones to be checked-in.

4.1.3. Check-in process at counters

The simulator was designed to be flexible in handling the check-in process. To model it with both realism and rigour, the simulator is inputted with the operating times, flight queues served, check-in time (i.e., duration) and associated destination carrousels for each counter [22].

The total check-in time of a passenger comprises a system service time and an input delay for each checked bag. For each passenger p, his system service time S_p and his number of bags B_p are modelled as random variables following distributions calculated from historical data. The delay produced by each bag is assumed constant and denoted by K. Therefore, passenger's p total check-in time is $T_p = S_p + B_pK$. In our experiments, we generate separate distributions for three types of passengers classified by the airline: domestic, to-USA and other international destinations. Each distribution was obtained from several field measurements.

The number of counters open at any given moment must also be managed. Each airline has an initial daily assignment but as the day progresses this is adjusted so that counter agents can be redistributed to maintain quality of service as expressed in terms of passenger waiting time. In practice (and in the simulator), service managers responsible for these assignments execute the following algorithm:

1. Calculate queue waiting time (QWT) for the last passenger at each queue q. Let $\bar{T}_q$ be the expected check-in time of a passenger in queue q. Let $C_q(t)$ be the number of counters servicing q and $L_q(t)$ the number of passengers in q at time t. Then QWT at time t is $W_q(t) = L_q(t)\bar{T}_q/C_q(t)$.
2. Let W^* be the maximum QWT allowed (a service quality standard). We determine $q' = \arg\max_q W_q(t)$. If $W_{q'} \geq W^*$, proceed to step 3. Otherwise, the process continues unchanged and the algorithm terminates.
3. For each $q \neq q'$ check whether one counter assigned to it can be reassigned to q' without exceeding the maximum expected QWT, W^*. Choose the option that will least effect the queue whose check-in staff would be thus reduced.
4. Reassign the counters accordingly and continue the check-in process.

Note that whereas the simulator makes these checks every 10 min, in the real operation their frequency will depend on the experience and attentiveness of the manager responsible.

Table 1
Bag categories.

Category (k)	Fraction (p_k)	Stoppage chance	Stoppage complexity	Example
Type 1	0.63	Low	Low	Properly injected bags
Type 2	0.27	Low	High	Soft (small or medium) items (backpacks, gym bags, etc.) injected without a tray. Typically, the incident they cause is to catch on something at some point along the conveyor.
Type 3	0.09	High	Low	Bags with wheels standing up or with the wheels down or in front, They tend to slide around on the belt colliding with other bags or blocking their advance.
Type 4	0.01	High	High	Large items that should go into the Oversized Belt or big soft items injected without a tray.

4.1.4. Generation of bags and incidents

Before creating a bag in the system, the likelihood that it will generate a conveyor incident must be determined. Each bag will have associated with it a probability that it will generate such an incident, the type of incident generated and the resulting conveyor stoppage time.

Based on field measurements the bags were divided into four categories designated Type 1, 2, 3 or 4 according to the frequency and seriousness of the incidents they might originate. Thus, when a bag is created in the system it is assigned the type number corresponding to its category. The probability that a bag is in category k, namely p_k, is equal to the observed fraction of bags falling in category k in the field. The definitions of the categories and the associated probabilities are summarized in Table 1.

We estimate the probability that a bag of certain type causes an stoppage on a given section of the conveyor from historical data provided by the company in charge of the conveyor operation. We also fit a uniform distribution for the time needed to fix the stoppage and resume normal operation. The minimum and maximum stoppage times for the uniform distribution were computed using the same database. Some measurements on the field were conducted to verify the quality of these estimations. The details of the simulation of stoppages are described in Section 4.2.3).

Finally, around 3% of the bags to be inputted in the system are oversized items that are not permitted on the conveyor serving the counters. In such cases those bags will be created for the oversized conveyor (T8) and the extra time needed to transfer each of these bags to it from the counter will also be inputted. We assumed a constant delay for each counter based on the distance between the counter and the oversized conveyor. We estimated the time a handler needs to reach the counter and return with the oversized bag. The handlers use a hand pushed cart and the travel times are very consistent. They were all obtained from real data.

4.2. Behaviour

The BHS Behaviour module routines are the nucleus of the simulator, incorporating all of the rules governing the operation of the baggage handling system. The module also defines the modifications made to Paramics to adapt it to the BHS.

4.2.1. Bag behaviour on the conveyor

The module's first routine adapts the behaviour of the entities to that of the conveyor. At each time step of the simulation clock, the speed of each bag is defined by the following rules:

1. If the bag is the first in its section of the conveyor (the leader bag), the last bag in the section immediately ahead is checked, and if the latter bag is stopped it means the conveyor is jammed (see "conveyor stoppages" below) and its speed is therefore zero. If it is not stopped, the speed of the leader is set at the same speed as was determined for its conveyor section.
2. If the bag is not the leader, its speed is set at the same speed as the bag immediately ahead of it.

The speed of each conveyor section is defined by the airport authority and is set by a configurable system parameter.

4.2.2. Injection of bags into collector

When bags are created in the system, they appear on the conveyor that connects the counter where they were checked in to the collector conveyor for that counter's T. Before a bag being inputted is accepted by the system, however, it must wait until there is a space large enough for it to enter.

To implement the conveyor priority for the counters in the simulator, an actuated signal light (red or green) is inserted at the intersection of the counter conveyor with the collector conveyor. Let δ^+, δ^- be the minimum upstream and downstream gaps required for a bag to cross the intersection. Then the corresponding actuated signal light only permits injection of a bag if the following conditions are satisfied:

1. The gap between the first upstream bag and the intersection is equal to or greater than δ^+ seconds; and

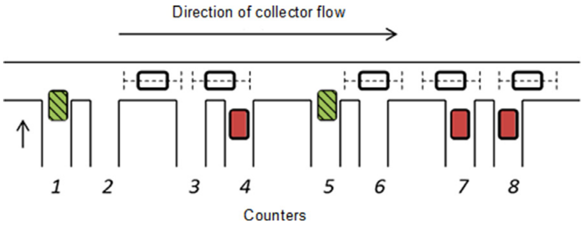


Fig. 6. Entry of bags into collector conveyor. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

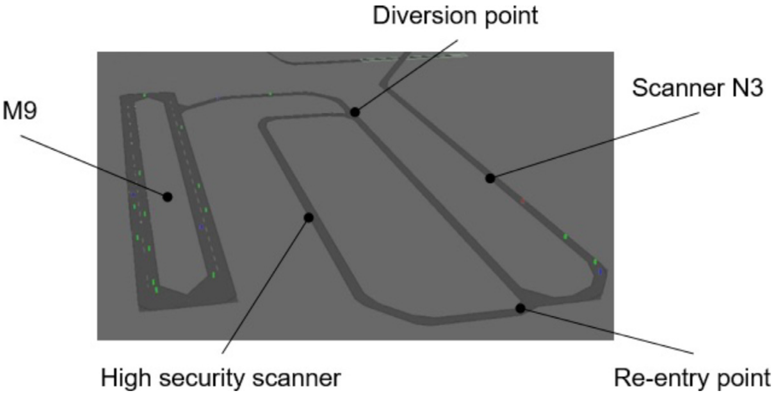


Fig. 7. High security scanner and M9 carrousel.

2. The gap between the first downstream bag and the intersection is equal to or greater than δ^- seconds.

In simple terms, the signal light prevents the entry of a bag into the collector conveyor whenever a bag already in the system is passing along it. In Fig. 6, for example, because of the flow of bags along the collector, bags at counters 4, 7 and 8 (all marked in red) cannot enter. This mechanism was implemented at all of the intersections subject to priorities, with appropriate δ^+ and δ^-.

The counter can have up to two bags waiting for a gap. When these two slots are occupied, the counter stops checking new passengers until one of them has entered the collector. There is an option that can be activated to completely bypass the collector and send all bags directly to the oversized conveyor. This option allows a counter to check passengers when the collector is offline.

4.2.3. Conveyor stoppages

This routine implements conveyor stoppages caused by bags. When an incident occurs somewhere along a conveyor section, the entire section and all the upstream sections back to the collector conveyor also stop. Sections downstream of the incident continue to function normally.

For the sake of simplicity we define certain “stoppage points” as the only places where a conveyor incident can occur. These points are located at places where conveyor bag jams are most frequent. According to Section 4.1.4, for each type of bag k and stoppage point s we specify the probability of occurrence of a stoppage p_k^s, the minimum m_k^s and the maximum M_k^s stoppage times. Each time a bag b passes through a stoppage point s the following routine is executed:

1. The occurrence of a stoppage is determined by generating a random number u, if $u \le p_k^s$ a stoppage occurs at point s.
2. If a stoppage occurs, its duration is determined by drawing a variate of an $U(m_k^s, M_k^s)$ distribution.
3. A call is made to the “incident” API in Paramics for the bag in question. The bag speed will then be set to zero for the period of time determined in point 2.
4. The bag behaviour routine described above propagates the stoppage upstream.
5. A call is made to the bag redirection routine (see below) for all bags on the conveyor section where the incident occurred given that the system will have lost the bags’ positions.

4.2.4. Security screening (redirection to M9)

N2 and N3 scanners. Each T has an X-ray scanner machine that screens all bags. When a bag enters the N2 scanner section of the conveyor (see Fig. 1), its redirection to M9 is executed with a fixed probability that models whether or not a bag is found to be suspicious. In the case of the N3 scanner, the routine is the same except that the bag is redirected to the high security scanner (see Fig. 7).

High security screening. When a bag is redirected for screening to the high security scanner, it is stopped there for a random time period to be scanned and is then re-injected into the system at the re-entry point.

4.2.5. Loading carrousels

Carrousels consists of sections of conveyors with various lanes to simulate bags that pile up on top of each other as the carrousel fills up. The input of a bag into a carrousel is therefore different from the conveyor intersections modelled by the injection-of-bags routine above.

When a bag coming directly from a counter arrives at the entrance to a carrousel, the number of bags already in the carrousel's first section is checked. If the three lanes are occupied, the arriving bag must wait until there is a space in one of them, at which point one of the lanes is chosen randomly and the bag then enters.

If the bag has not come directly from a counter, it enters the carrousel by a special lane; the only requirement is that a space be available.

4.3. Baggage loading area (BLA)

The routines in the baggage loading area module control the exit of bags from the system and define the state of system resources.

4.3.1. Handling at carrousels

A bag arriving without incident at the loading carrousel its flight is assigned to will circulate on the carrousel continuously until a baggage handler assigned to that flight removes it for loading. From that point the rest of the process is primarily manual. The bag will be sorted and then deposited in a cart to be loaded onto the appropriate aircraft.

Bags that arrive before the flight preparation start time wait in the assigned carrousel while those that arrive when the flight is in the *closing* state have only a certain probability of being loaded. Those that arrive in the *closed* state will, as just noted, be left behind.

Each carrousel has a number of handlers assigned to it determined by the handler-flight assignment parameter (see Table 3). The value of the parameter must be newly defined at each time step. This flexibility allows the user to assign a handler to multiple flights for simultaneous preparation or exclusively to a single, relatively complex one.

More formally, each time a bag passes by the carrousel exit point, the following routine is executed:

1. If the bag is destined for a closed flight, it is recorded as a late arrival and eliminated from the system. What happens in reality is that handlers set them aside (which takes almost no time) so a supervisor can reschedule them for another flight.
2. The simulator checks whether the flight the bag is destined for is in preparation or closing. If neither, it goes around the carrousel once again.
3. The simulator checks the availability of baggage handlers. If none is available the bag goes around the carrousel once again.
4. If none of the above conditions is satisfied, a handler removes the bag for loading and the simulator records the time and place. The handler will be occupied in the loading process for a time period that is exponentially distributed (see baggage handler assignment below). The bag is then eliminated from the simulator and considered to have been successfully loaded onto the aircraft.

4.3.2. Baggage handler assignment

At each simulation time step the following routines are executed to define the assignment of baggage handlers:

Availability of handlers: The total number of available handlers for each position during the workday is defined as a simulation parameter. This set of routines constantly updates the maximum number of handlers for the different positions in accordance with personnel planning decisions. In the case of the loading carrousels, the flights they are assigned to are also defined.

Handler availability time: At each time step the simulator checks whether any of the handlers have become available. The time a handler takes to complete a loading and become newly available is distributed exponentially with mean $1/\mu$ min.

4.3.3. M9 process and bag redirection

Carrousel M9, where bags exhibiting some irregularity are handled, functions similarly to the 8 loading carrousels but with two differences. First, there is always a fixed number of baggage handlers for all of the flights; and second, once the bags are processed they are not eliminated from the system but rather are placed in special piles to be returned to their destination carrousel (assuming there is still time before departure). The routine executed when a bag arrives at the exit point of the M9 carrousel is as follows:

1. If no baggage handler is available, the bag goes around the carrousel once again.
2. If the flight the bag is destined for is already closed, the time is recorded and the bag is eliminated from the system.

Table 2
System parameters.

Module	Parameter	Description
Counters	Arrival profiles	Empirical distributions of terminal passenger arrivals. Periodically updated for each flight.
	Baggage factors	Historical average number of bags per passenger. Calculated separately for each flight and month. New flights are assigned a factor for a flight similar in time of day and destination.
	Check-in times (duration)	Distribution of time required to check in a passenger at the counter. Supports normal, exponential, uniform and triangular distributions.
	Bag drop percentage	Historical average percentage of passengers already checked in who queue only to drop off baggage. Calculated separately for each flight. Defines which queue they are assigned to and drop-off queuing time.
	Maximum QWT	Maximum queue waiting time considered by airlines to be acceptable. Defines the activation of contingency measures.
BHS behaviour	Conveyor speed	Speed of each conveyor section.
	Incident probability	Probability that a given type of bag will cause an incident at some point along the conveyor.
	Post-incident recovery	Distribution of time required to return the system to normal operation, measured from the moment the incident occurs. Includes time to detect the incident, resolve it (this component varies by incident type) and reinitiate the system.
	High security screening rate	Probability (calculated from historical averages) that a bag is flagged as suspicious by first security screening (N2) and diverted to M9 screening. Historically, 8% is considered to be the “natural” rate of such diversions.
Baggage handling area	Bag processing times per baggage handler	Average time taken by a handler to remove a bag from the conveyor, place it in the baggage cart, record it on the loading sheet and be ready to receive another bag. Studies have found that generically, handlers average 30 s/bag.
	Carrousel flight preparation capacity	Maximum number of flights that can be prepared simultaneously at a carrousel. Determined by the space availability around the carrousel and whether carts or DCV’s are used.

3. If a handler is available, the exit time from M9 is recorded and the bag is eliminated from the system. If the state of the bag’s flight is “closing”, go to step 4; otherwise, go to step 5.
4. If the flight state is “closing”, it is too late to return the bag to its loading carrousel so it is sent directly to the aircraft. The loading time is recorded as the time the bag exited M9 and the bag is eliminated.
5. The identification number of the bag is recorded and placed in a cart with all the other bags going to the same loading carrousel.

The bags exiting M9 are sent back to their respective loading carrousels once either (i) the cart destined for a given loading carrousel reaches a certain number (the cart’s capacity), (ii) the time limit since the last return of bags from M9 has been reached (usually 10 min), or (iii) the flight a bag is destined for has entered the “closing” state. If either of these conditions is satisfied, all bags in M9 are returned to the loading carrousels. This routine creates (in a number of time units depending on the distance between M9 and the carrousel in question) new bags, each with the identification number assigned to it when it was placed in the pile. These bags are created in the special lane for entry into their respective loading carrousels.

4.3.4. Connecting flight bags

A second source of bags is the arriving passengers taking connecting flights. At SCL, however, there is no direct link between the arrival and departure infrastructures. Connecting flight baggage is therefore modelled using a known distribution for each arrival. These bags arrive at the carrousels for the connecting flights at a uniform rate during the flight preparation period.

4.4. Input parameters

The simulator has a large number of input parameters that can be adjusted to suit the needs of the experiment. To simplify their development we have divided them into two types: low variability or *system parameters* and high variability or *scenario parameters*.

The system parameters set the general configuration of the BHS and other characteristics that do not vary from one instance to the next, such as counter check-in time. Any modification to these would only be required in the case of some major change in actual airport operations. The scenario parameters, on the other hand, tend to vary from experiment to experiment and express scenario data such as daily flights or planning strategies that can be easily modified in reality. A complete list of the system and scenario parameters is set out in Tables 2 and 3.

Table 3
Scenario parameters.

Module	Parameter	Description
Counters	Itinerary	Departure (arrival) time and destination (origin) of each flight
	Expected passengers	Expected number of passengers for each flight. Either the historical load factor for the date, reservation data or the actual load is used, depending on the scenario.
	Bag type input error	Probabilities that a bag will be assigned each bag type.
	Counter planning	Daily planning for each check-in counter; specifies for each 10 min interval whether counter is open or closed.
	Destination carrousel	Each counter is associated with a destination loading carrousel for the bags it originates. This assignment is modifiable every 10 min.
	Queue formation	Grouping of flights into check-in queues and the list of counters at which each queue is checked in.
Baggage handling area	Number of baggage handlers in area per work position	Number of baggage handlers available for each position. Defined for each 5 min interval.
	Pre-departure preparation time	Time before departure of flight at which baggage handlers begin to prepare it. Bags arriving before this time wait in the assigned loading carrousel.
	Handler-flight assignment	Defines the flights each baggage handler is preparing at any moment. Definitions are for 5 min intervals.

5. Experiments and results

The baggage system simulation described above was applied in a number of practical experiments. In what follows, three of the most interesting applications are described in detail to demonstrate the capabilities of the proposed simulator tool.

The experiments include exploratory analyses to improve our understanding of the workings of the system and testing of specific operating policies. They entailed the creation of 120 different scenarios for which 100 replications each were performed, amounted to 12,000 replications in all. As each simulation run took on average around 30 min, total computer resource use amounted to some 6000 CPU hours. Each scenario simulates almost a full day of operation, from 4:00 a.m. to 3:00 p.m. We leave a one hour warm time and one hour cool- down. This times coincide with the airport's lowest activity moments. A common random number variance reduction technique was used in comparative experiments; specifically, the same random numbers stream was used for generation of passengers and bags. For the rest of the simulation processes the same random seeds were used.

5.1. Determining baggage handling system capacity

5.1.1. Introduction

Santiago International Airport has experienced a steady increase in demand in recent years, registering an annual growth rate between 2010 and 2013 of almost 5%. In this context, and with no new investment in airport expansion planned for the medium term, there is an obvious need for an effective means of analyzing the facility's capacity to absorb the rising demand. Our first experiment was therefore aimed at explaining the discrepancy between the theoretical capacity of the BHS conveyor, rated by the manufacturer at 1000 bags/h, and the X-ray scanner (bottleneck) processing rates observed in actual operation.

An analysis of the average check-in time for domestic flights at SCL found that for a T subsystem of 28 counters, the expected rate was 1160 bags/h (assuming an average of 1 bag per passenger). This implies that under normal conditions the observed rate should be close to the theoretical one, at least over short intervals. In reality, however, the observed rate rarely rose above 500 bags/h.

For non-domestic flights we would expect the bottleneck to be the check-in process rather than the conveyor. The objective of this exploratory experiment was therefore to discover whether under ideal operating conditions the theoretical processing rates could be reached or whether there was some other, undetected bottleneck in the BHS system.

Applying our simulator it was demonstrated that the true capacity of the system, attainable only under ideal operating conditions, was 800 bags/h, considerably below the theoretical level. The key factor in the discrepancy turned out to be the impediments to a more rapid injection of bags from the counters to the collector conveyor.

5.1.2. The experiment

We began the experiment by simulating ideal conditions in which there were no conveyor stoppages or bag jams. Demand was assumed to be constant but large enough that the counters were never idle. In other words, as soon as one passenger's check-in process was completed, another one stepped up to the counter. Check-in time (duration) was set at a constant 50 s, the best time recorded in field observations.

The highest numbers of bags entering the system during a one-hour moving window in the simulation with no stoppages or counter handling errors is shown in Table 4 for different numbers of check-in counters. As can be seen, the maximum capacity was found to be only 775 bags/h. Also evident is that the marginal benefit of opening an additional counter steadily decreased as the number of counters increased. This was the case because fewer and fewer bags were injected at the

Table 4
Bag injection capacity of a T (simulation).

No. of opened counters	28	27	26	25	24	23	22	21	20	19	18
Bags/hour	775	774	773	771	768	764	759	753	738	710	682
Marginal counter benefit	+1	+1	+2	+2	+4	+5	+6	+15	+28	+28	–

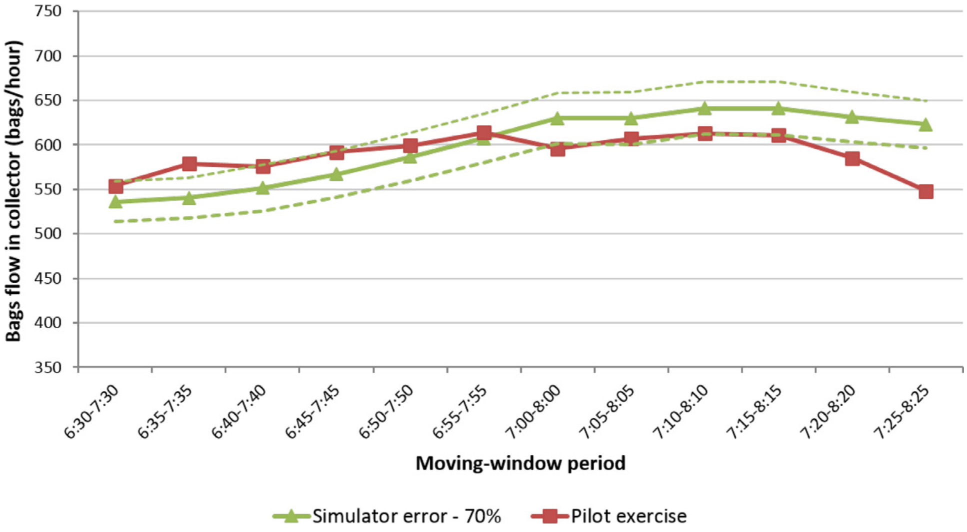


Fig. 8. Simulated and observed baggage flows in collector conveyor (T3, LAN domestic).

counters added “downstream” as the traffic-actuated signal light system described earlier gave priority to bags already on the conveyor, having been injected at the “upstream” counters. This result affords an interesting insight into the current problems at SCL, where the system was originally designed for only 20 counters per T. The other 8 were added later in the apparent hope of expanding baggage injection capacity, to little avail as we can now appreciate.

This simulation showed that the maximum capacity of each T is given by neither the scanner nor the number of counters but rather the injection of bags into the collector, and is indeed much less than the manufacturer’s capacity rating.

5.2. Impact of counter handling errors

5.2.1. Introduction

Our second experiment was aimed at quantifying the impact of bag handling errors at the counters on system operation. Previously recorded observations revealed that an average of 30 min of operating time were lost every day due to stoppages. Not surprisingly, these incidents occurred more frequently during periods of higher bag loads.

In the previous experiment we saw that under optimal conditions, conveyor capacity should be approximately 750 bags/h (see Table 4), but that this fell to less than 500 bags/h under the same demand conditions if the observed error rates were applied. What remained to be determined was the effect of a significant reduction in errors under conditions of normal operation.

5.2.2. The experiment

The experimentation consisted in simulating several days of real operation using observed itineraries and baggage loads in which the handling errors were gradually reduced in each successive scenario. Although each of the scenarios covered an entire day’s operations, attention was focussed on the hours of 6:30 a.m.–8:30 a.m., the peak demand period for domestic flights when the impact of errors is greatest. Based on the results for these scenarios, we set up a one-week pilot exercise in which operations would be closely monitored to avoid handling errors.

The main results of this experiment, in which a 70% reduction in handling errors was observed, was an increase in effective collector capacity, attaining levels approaching those predicted by the simulation. In Fig. 8 the peak-hour simulated collector baggage flows (average of scenarios) are compared with the observed flows during the one-week pilot exercise (daily average) for one-hour moving-window periods. The dotted lines correspond to the 95% pointwise confidence band for the simulated baggage flows Given that during the pilot exercise the airport’s BHS recorded its best ever performance and the baggage handlers are unfortunately not always so diligent, it is unlikely the system will ever achieve a better error rate than it did during this experiment. We may therefore conclude that the SCL system’s maximum baggage capacity is 2500 bags/h (625 per T) and that any effort to improve this figure must focus on the collector conveyor.

Table 5
Time carrousel operated at different bag load levels: base vs. scenarios A and B.

Time (in minutes)											
Load level	Base			Scenario A				Scenario B			
	M3	M4	Total	M3	M4	Total	Δ	M3	M4	Total	Δ
70 or +	0	10	10	5	10	15	(-50.0%)	20	15	35	(-133.3%)
60 or +	0	80	80	30	45	75	(6.3%)	50	65	115	(-53.3%)
50 or +	55	350	405	165	110	275	(21.4%)	110	185	295	(-7.3%)
40 or +	95	495	590	300	220	520	(-5.1%)	215	270	485	(6.7%)
30 or +	320	635	955	495	565	1060	(-66.9%)	590	460	1050	(0.9%)

5.3. Carrousel load balance

It was noted earlier that each check-in counter sends bags to a single assigned loading carrousel. This implies that deciding which counters handle which flights also determines which carrousels will prepare which flights for loading. The objective in this third experiment was to determine configurations of passenger queues at counters that will prevent carrousel overload. Priority in this case was given to BLA operations.

5.3.1. Introduction

The experiment was inspired by the situation that arose at SCL during February 2011 when 75% of the baggage to be loaded on international flights of Chile's main air carrier (checked in through T2) were handled at carrousel M4 and the remaining 25% at M3. This resulted in overloading of M4 during peak periods, generating stoppages throughout the BHS. The root of the problem was that the carrousel assignments were decided by the airline's counter managers, who were not sufficiently informed of the conditions at any given moment in the baggage handling area and thus based their decisions exclusively on ensuring adequate check-in counter service to passengers. Furthermore, since each counter has only one assigned carrousel, changing the assignments would have necessarily meant stopping the system for several minutes.

We began by analyzing the distribution of the assignments of international flights to counters, in each case evaluating three factors: *passenger impact*, represented by the queue waiting time (QWT); *carrousel load*, that is, the extent to which the distribution avoids bag overload at the carrousels; and *ease of implementation*, or the extent to which multiple flights checked in at a single counter are grouped by similarities such as the same destination or type of flight (regional, long-haul).

International flights are particularly complex from a BLA operating standpoint. Since they are generally served by wide-body aircraft carrying 160 to 260 passengers, or more than double the 100 passengers averaged by domestic flights, bags must be loaded in ULD's (special containers known as unit load devices) that take up twice the space of traditional carts and can only be moved by tractor, complicating manoeuvrability in the carrousel area where space is already limited.

The base scenario (see Fig. 9) is the situation actually observed at SCL on a typical day, in which only 9 flights are processed at M3 compared to 55 at M4. As a result, there are periods in which M3 is not being used while M4 is almost saturated.

Based on observations in the BLA it was estimated that once the number of bags in a carrousel exceeds 50, the loading operation starts to be affected, and with more than 70, special measures must be resorted to. The worst BLA situations are prolonged overloads (more than 2 h), which occur when various flights have to be prepared simultaneously. On the other hand, short overload spikes caused by 1 or 2 simultaneous large flights are manageable.

5.3.2. The experiment

Simulations were conducted for two alternative scenarios denoted A and B in which flights for a particular destination were transferred from M3 to M4. In A, the transfers were flights destined for Argentina and in B, they were flights headed for Brazil. The results in terms of time the carrousels operated at different bag load levels are displayed in Table 5.

Since in our experiment priority was given to BLA operation, both scenarios diverted a significant part of demand from one carrousel to the other. An orderly counter assignment configuration was maintained, however.

Although at first it would seem that either scenario met the objective of improving bag load distribution, in B the carrousel overload times were slightly greater. Both scenarios solved the problem for the worst part of the day, between 6:00 a.m. and 11:00 a.m. (Fig. 9). Whereas in the base case carrousel M4 was constantly operating at more than 50 expected bags, A as well as B replaced this with short high load intervals. B generated an excessively high spike in M4 when some flights normally processed in M3 were removed (8:30–10:00 a.m.) and aggravated demand on M3 during the morning peak demand period (6:30–8:00 a.m.). In operating terms, therefore, the segregation in A produced a substantial improvement in quality of operation.

These simulations can also be used to evaluate the hourly average passenger queue waiting time. Scenario A was found to generate an improvement for the segregated passengers (those going to Argentina) and a marginal impact on the other flights. This was so because the more complex flights (for example, those to the United States) have longer average check-in times, slowing down passengers for other destinations (such as Argentina) with shorter times who are in the same queue (Fig. 10).

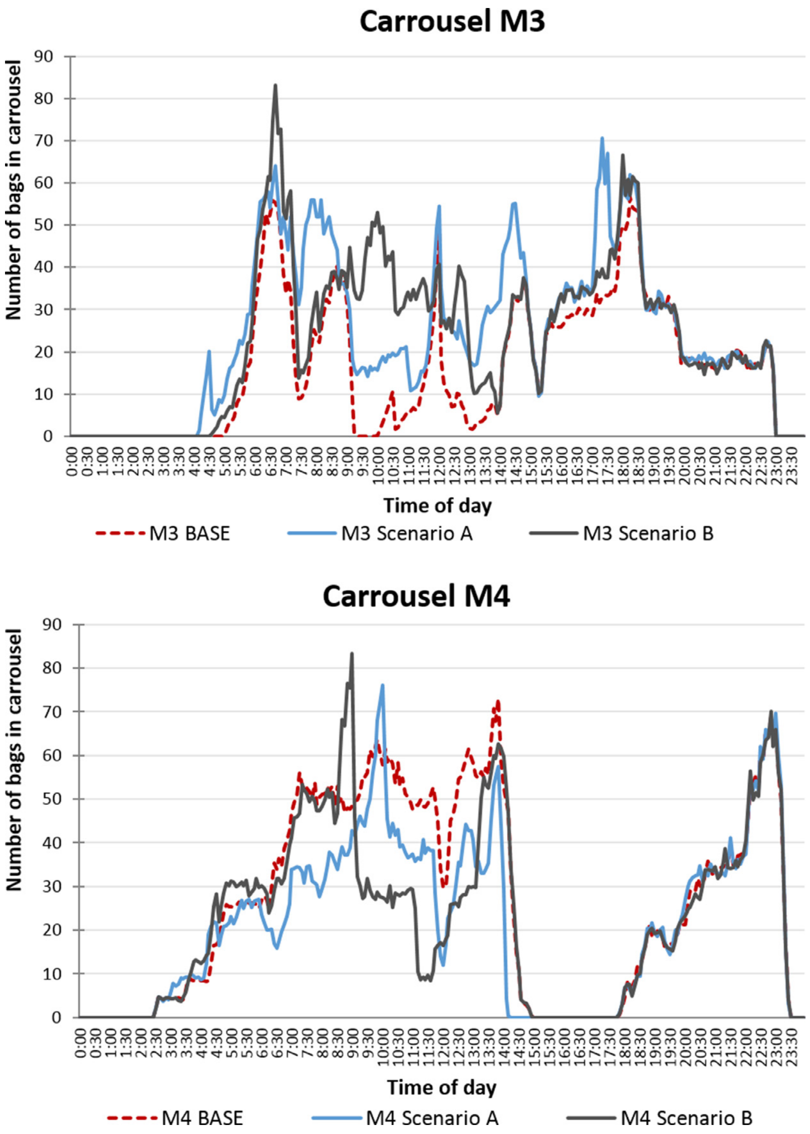


Fig. 9. Bags on carrousel by time of day: base vs. scenarios A and B.

The analysis of the scenarios in which international flights processed in T4 were segregated showed that there is significant potential for operating improvements by incorporating expected carrousel bag loads as a consideration in the grouping of flight queues at the counters. Although both alternatives initially appeared to be similarly attractive in improving the operation, the simulator was able to determine that one of them (Scenario B) actually exacerbated it.

Finally, note that the QWT for A improved considerably for the segregated passengers. Given that flights to Argentina accounted for about 30% of passengers checked in through T4, the global reduction in waiting time was substantial.

5.4. Evaluation of contingency plans

At peak hours, SCL has no spare capacity for dealing with breakdowns in the BHS. Any fault in the system can therefore have major consequences for the entire operation. Our last experiment tested alternative operating schemes that would permit the system to handle the most critical system faults.

5.4.1. Introduction

System breakdowns are inevitable. If a conveyor fault is brief, the operation can be stopped for a few minutes with no real repercussions, but in a more serious case, operating measures must be taken immediately to prevent delays in flight departures.

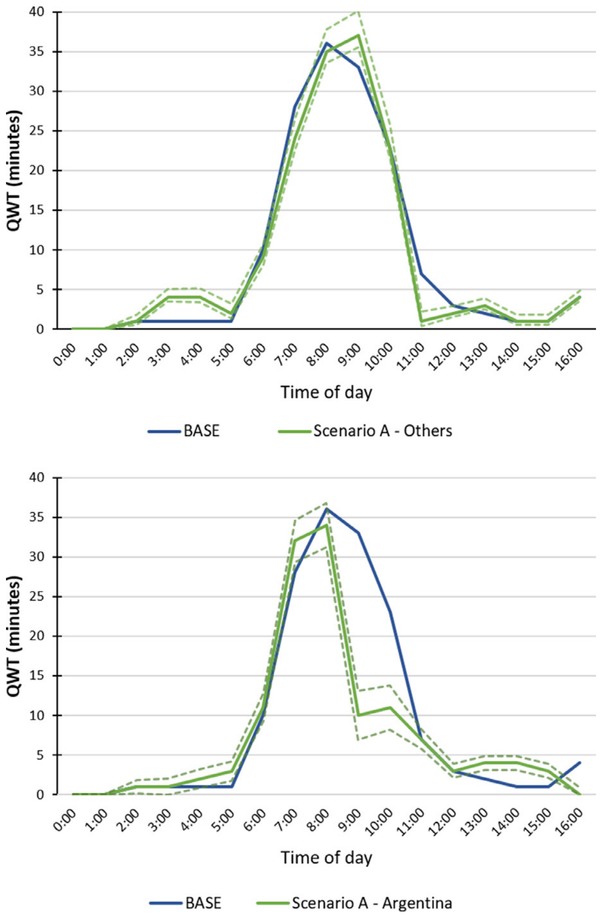


Fig. 10. Passenger queue waiting time: base vs. scenario A. Dotted lines correspond to the 95% pointwise confidence band of queue waiting time for scenario A.

Four types of faults were identified that require a significant alteration of normal operations to fix: (a) a single carrousel fault; (b) a fault in both carrousels of a T, preventing them from being used to prepare a flight but with counters still able to carry out check-ins; (c) inability to inject bags into a T, caused, for example, by a conveyor fault or a scanner breakdown; and (d) total loss of T functionality (see Fig. 11).

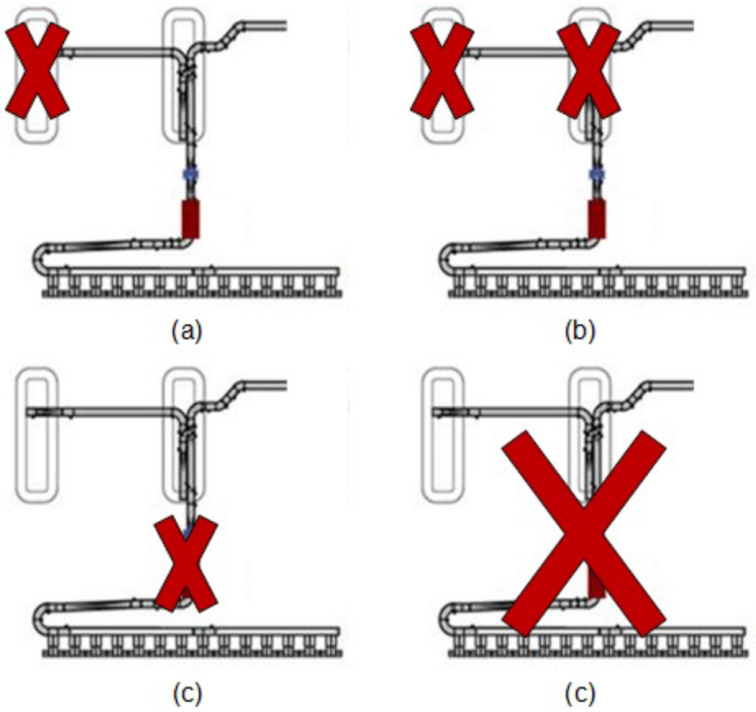
Our experiment examined four possible strategies for responding to these faults:

1. Redirect the bags injected at the check-in counter to the other carrousel for the same T. The natural solution when a carrousel breaks down is to prepare all of the flights in the remaining carrousel. This has the advantage of not having to switch passenger queues to different counters but care will have to be taken to ensure the functioning carrousel is not overloaded.
2. Move check-in to a different T. This implies shifting queues to different counters, meaning in turn that there must be counters available, and may affect the check-in operations of other airlines.
3. Send the bags to the loading area on the oversize conveyor, which is relatively little used and has its own security scanner. Passenger check-in operations will not be affected by this measure but transferring the bags to the oversize conveyor, which is handled manually, will require the use of airport personnel.
4. Prioritize flights and redirect passengers and baggage for flights that are departing shortly to the systems that are functioning. This will inevitably mean check-in delays for passengers with longer times to departure and thus will negatively impact the quality of service they experience.

Finally, the time of day at which faults occur was also taken into account given that a strategy that works well at a moment of low demand may not be effective in periods when demand is high.

5.4.2. The experiment

The objective of the experiment was to determine which strategy or mix of strategies is most effective for each fault type. The fault/strategy pairs that were tested are indicated in Table 6.


(a) (b)

(c) (c)

Fig. 11. Critical faults.

Table 6
Strategies tested for each type of fault.

Strategy	Critical fault			
	1 Carrousel	2 Carrousels	Injection	T
Carrousel redirect	✓	–	–	–
Change counter/T	–	✓	✓	✓
Use oversize conveyor (OS)	✓	✓	✓	✓
Prioritize flights	–	✓	✓	✓

Table 7
Contingency protocol for international flights: fault in T2 carrousel on weekday.

Time of day	Strategy
00:00–11:40	Redirect all bags to the functioning carrousel.
11:40–13:20	Use oversize conveyor for flights to following airports: -EZE -COR -MDZ -GRU Redirect all bags to the functioning carrousel.
13:21–15:30	Redirect all bags to the functioning carrousel.
15:31–17:30	Use oversize conveyor for flights to following airports: -LIM -GYE Redirect all bags to the functioning carrousel.
17:31–24:00	Redirect all bags to the functioning carrousel.

Since as noted above, the time of day is also an important consideration, the strategies were tested separately for the morning and afternoon peak periods, the critical moments for the BHS. Weekday and weekend operations were also tested separately given the significant differences between them as regards itineraries and especially the distribution of passenger arrivals at the terminal.

To determine which flights were affected and what resources were available (other counters, lightly loaded carrousels, etc.), it was necessary to identify the counter subsystem where the fault occurred. Policies were generated only for faults in T2 and T3 since they are the two most heavily used subsystems.

The test scenarios were defined in terms of 4 attributes: fault type, fault location, time of day, and strategy mix. There were a total of 192 scenarios, even after restricting the analysis to morning and afternoon peak periods and excluding aspects such as which flight queues would be moved to different counters. Scenarios that were completely dominated or were known to have little impact on operations were discarded.

To evaluate the quality of the solutions generated by the strategies, three quantitative factors were analyzed: on-time bag loading feasibility, queue waiting time, and impact on other systems (more specifically, avoiding system overloads). Qualitative conditions, mainly ease of implementation and number of agents involved, were also evaluated.

The results of this experiment were used to specify a series of contingency protocols based on the most effective strategy mix for each combination of fault type, fault location, time of day and weekday/weekend. An example of these protocols for international flights is set out in Table 7.

6. Conclusions

This article presented a novel extension of a commercial simulation platform to model the baggage handling system at Santiago International Airport in Chile. The proposed tool used microsimulation for the complete system including all of its processes and interactions from the arrival of passengers at the check-in counters to the loading of their bags onto the aircraft. This detailed treatment enabled us to evaluate highly specific operating policies and measure the impact on the system of minor changes (e.g., the order of opening of additional counters) or major ones (i.e., breakdowns or other contingencies).

A key contribution of this work was the development of an unconventional application to model the movement of baggage along a conveyor using the API of a software originally designed to simulate urban vehicle traffic. This adaptation provided a high degree of flexibility to represent unique elements in the baggage system that are not supported by the specific capabilities of commercially available baggage handling softwares.

Another important contribution is the extended simulation platform’s ability to model an integrated system that not only combines various airport operating subsystems but also takes account of their interactions. This latter feature is particularly significant because the complexity of the interactions is such that it would be very difficult to develop an analytical model for optimizing the system as a whole.

Experiments were conducted in which the proposed simulation tool was applied to four specific cases involving Santiago airport’s baggage handling system. The first application, focussing on the design aspect, was aimed at finding the system’s bottleneck. The tool located it at the interaction between the check-in counters and the conveyor, a discovery that would have been very difficult to make by separate analyses of the capacities and flows of each system element (conveyor speed, check-in time, scanner capacity). The second application evaluated the impact and propagation through the system of changes made to the assignments of passenger queues to check-in counters. In this case the tool demonstrated how an inappropriate grouping of different flight queues at counters is the determining factor in baggage overloads at the aircraft loading carrousels.

The third experiment studied options for balancing the baggage load at loading area carrousels and evaluated the impact of two alternatives on passenger queue waiting times. Finally, the fourth case used the tool to analyze multiple “what-if” scenarios representing possible major breakdowns in the system and evaluate the effectiveness of alternative practical responses that could be implemented.

The potential applications of the our simulation tool are much wider than these experiments suggest, however. One example among many would be the evaluation of the impact on airport operations of the expected increment in passenger demand over the coming years. The platform should be able to answer key questions such as how many flights the current baggage system could simultaneously handle, at what times of day or days of the week could additional demand be handled without overloading the system, and how many additional airlines could be accommodated with the current infrastructure. As for future extensions to the proposed platform, the incorporation of baggage handlers as simulation entities would enable check-in counter staffing policies and critical points in the baggage loading area to be analyzed. Finally, the tool could be employed to evaluate possible changes in airport infrastructure, with the results then used to support the design process for implementing those changes.

Acknowledgments

The authors wish to acknowledge the support of CONICYT/FONDECYT/ REGULAR/ N.° 1141313 and the Complex Engineering Systems Institute, ISCI (ICM: P-05-004-F, CONICYT: FB0816). The authors also thank Andes Airport Services for kindly providing the real problem that motivates this research as well as the data to develop and test the simulation platform.

References

[1] J. Cavada, Modelos de simulación y asignación de personal para manejo de equipajes en un aeropuerto internacional, Tesis de Magíster en Gestión de Operaciones, Universidad de Chile, 2013.

[2] J. Tanner, A Queuing Model for Departure Baggage Handling at Airports, Institute of Transportation and Traffic Engineering, University of California, Berkeley, 1966 Master's thesis.

[3] G. Robinson, Simulation models for evaluation of airport baggage-handling systems, in: Proceedings of the Third Conference on Applications of Simulation, 1969, pp. 226–235.

[4] R. de Neufville, A. Odoni, Passenger terminal design, Transp. Res. Part A 26 (1992) 27–35.

[5] V. Tošić, Review of airport passenger terminal operations analysis and modelling, Transp. Res. Part A 26 (1992) 3–26.

[6] R. de Neufville, The baggage system at denver: prospects and lessons, J. Air Traffic Manag. 1 (1994) 229–236.

[7] A. Law, Simulation, Modeling and Analysis, fourth ed., McGraw-Hill, 2007.

[8] L. Brunetta, L. Righi, G. Andreatta, An operations research model for the evaluation of an airport terminal: SLAM (simple landside aggregate model), J. Air Transport Manag. 5 (1999) 161–175.

[9] A. Correia, S. Wirasinghe, A. de Barro, Overall level of service measures for airport passenger terminals, Transp. Res. Part A 42 (2008) 330–346.

[10] A. Lemer, Measuring performance of airport passenger terminals, Transp. Res. Part A 26 (1992) 37–45.

[11] M. Johnstone, D. Creighton, S. Nahavandi, Status-based routing in baggage handling systems: Searching verses learning, IEEE Trans. Syst. Man Cybern. Part C 40 (2010) 189–200.

[12] V.T. Le, J. Zhang, M. Johnstone, S. Nahavandi, D. Creighton, A generalised data analysis approach for baggage handling systems simulation, in: 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2012, pp. 1681–1687.

[13] S. Nahavandi, D. Creighton, M. Johnstone, V. Le, Airport operations: a system-of-systems approach, in: M. Jamshidi (Ed.), Systems of systems engineering: principles and applications, CRC Press, 2009, pp. 403–419. Ch. 16.

[14] M. Johnstone, D. Creighton, S. Nahavandi, Simulation-based baggage handling system merge analysis, Simul. Modell. Pract. Theory 53 (2015) 45–59.

[15] R. de Neufville, A. Odoni, Airport Systems: Planning, Design, and Management, McGraw-Hill, 2003.

[16] G. Zografos, OPAL Optimisation Platform for Airports including Landside, in: Transportation Research E-Circular E-C 042, 2002, pp. 6–10. Transportation Research Board.

[17] I.E. Manataki, K.G. Zografos, A generic system dynamics based tool for airport terminal performance analysis, Transp. Res. Part C 17 (4) (2009) 428–443, doi:10.1016/j.trc.2009.02.001. http://www.sciencedirect.com/science/article/pii/S0968090X09000229.

[18] A. Abdelghany, K. Abdelghany, R. Narasimha, Scheduling baggage-handling facilities in congested airports, J. Air Transport Manag. 12 (2006) 76–81.

[19] A. Ascó, J. Atkin, E. Burke, The airport baggage sorting station allocation problem, in: Proceedings of the 5th Multidisciplinary International Scheduling Conference, MISTA, 2011.

[20] M. Savrasovs, A. Medvedev, E. Sincova, L.D. Pedrera, Riga airport baggage handling system simulation, in: J. Otamendi, A. Bargiela, J. Montes (Eds.), Proceedings 23rd European Conference on Modelling and Simulation, ECMS, 2009, pp. 384–390.

[21] M. Lazzaroni, Modeling passenger and baggage flow at Vancouver airport, Simio case studies, Last retrieved on 12 December 2015, 2012. http://www.simio.com/case-studies/Modeling-Passenger-and-Baggage-Flow-at-Vancouver-Airport/.

[22] R. Goepingger, Modelo de simulación y análisis de políticas de operación para el manejo de equipajes en el Aeropuerto de Santiago, Tesis de Magíster en Gestión de Operaciones, Universidad de Chile, 2012.