Simulation model for assessing passengers' dwelling patterns: case study of Lisbon T2

João Bernardo de Paiva

Dissertação para obtenção do Grau de Mestre em

Engenharia Civil

Orientador:

Doutor Vasco Domingos Moreira Lopes Miranda dos Reis

Júri:

Presidente: Professor Doutor João Torres de Quinhones Levy

Orientador: Doutor Vasco Domingos Moreira Lopes Miranda dos Reis

Vogais:

Professor Doutor Jorge Miguel dos Reis Silva

Professor Doutor Filipe Manuel Mercier Vilaça e Moura

Outubro de 2015

Acknowledgements

I would like to thank, upon completion of this work, several people that somehow were important in order for this achievement to be possible.

First of all, I would like to express my sincere gratitude to my supervisor, Dr. Vasco Reis, whose guidance and suggestions have taught me a lot along the way and were crucial at several major points of this work's development process.

This research would not have been possible either without the precious and continuous support from Duarte Cunha regarding the simulation software, or the insightful advices from Liliana Magalhães, so my sincere thanks also go to these two researchers from Instituto Superior Técnico.

Last but not least, I want to thank my parents, sister and friends for always being there throughout the entire process and providing me all the necessary support and patience in order to surpass the various obstacles that inevitably arose at several different moments in time.

Resumo

A presente dissertação visa avaliar e eventualmente propor, através de um modelo de simulação, alternativas para a configuração espacial da área restrita do terminal 2 do aeroporto de Lisboa. De modo a atingir estes objectivos, uma revisão da literatura foi realizada antes do mais, de modo a recolher as principais abordagens e conclusões de outros investigadores relativamente à modelação de pedestres, processos aeroportuários de passageiros, modelos de escolha discreta, retalho em aeroportos e comportamento de consumo de um modo geral.

Uma vez anotadas as características individuais mais determinantes para as actividades não-aeronáuticas dentro de um aeroporto e destacadas as ferramentas computacionais mais utilizadas, um modelo de simulação genérico foi desenvolvido, através do *software* AnyLogic, visando representar o processo de chegada de passageiros vindos da verificação de segurança, o modo de funcionamento do processo de escolha discreta dentro de um conjunto finito de alternativas e, por fim, a etapa do embarque.

Este modelo genérico foi de seguida aplicado a um caso de estudo específico (o terminal 2 do aeroporto de Lisboa) e a situação actual foi analisada através de parâmetros de desempenho definidos pelo autor. Esta análise permitiu constatar que a configuração do espaço comercial poderia ser melhorada, tanto do ponto de vista dos passageiros como das entidades gestoras do aeroporto. Seis alternativas foram então propostas e simuladas – quatro de entre elas revelaram-se interessantes e geradoras de melhores resultados globalmente, sendo a melhor das quatro uma combinação de três modificações distintas no posicionamento relativo das instalações dentro do actual *layout*.

Palavras-chave

Aeroporto, simulação, modelação de pedestres, tomada de decisão, *layout* comercial, aeroporto de Lisboa

Abstract

The present dissertation aims to evaluate and eventually propose, through a simulation model, some alternatives regarding the commercial layout of the terminal 2 in Lisbon International Airport. In order to achieve these objectives, a literature review was made before anything else, in order to better understand what approaches previous researchers had used and what main conclusions had been reached regarding pedestrian modelling, air passenger processes, discrete choice procedures, airport retailing and shopping behaviour in general.

Once the most determining individual characteristics for discretionary purposes inside an airport were noted and the most commonly used software tools were highlighted, a generic simulation model was developed in AnyLogic software in order to represent the passenger arrival distribution from security to the restricted area of the terminal, the way of functioning for their discrete choice process among a finite set of discretionary activities and, finally, the boarding stage.

That general model was then applied to a specific case study (Lisbon T2) and the current situation was analysed through some specific performance indicators defined by the author. These parameters' values revealed that the commercial layout of the terminal could be improved both from the passengers' and the managing entities' perspectives, so six different alternatives were proposed and simulated. Four of these proved themselves as leading to better results overall, being the best one a combination of three different modifications in the relative positioning between different facilities inside the terminal layout.

Keywords

Airport, simulation, pedestrian modelling, decision-making, commercial layout, Lisbon airport.

Table of Contents

List of figures

List of tables

Glossary

ANA – Aeroportos de Portugal

ATM – Automated Teller Machine (cash machine)

CA – Cellular Automata

CDM – Consumer Decision-Making

GA-Ped – Group Aware Pedestrian

GEV – Generalised Extreme Value

IATA – International Air Transport Association

MNL – Multinomial Logit

LOS – Level Of Service

SDL – Specification and Description Language

SEF – "Serviço de Estrangeiros e Fronteiras" (Immigration and Border Control)

SLAM – Simple Landside Aggregate Model

pax – abbreviation for "passengers"

T2 – Terminal 2

USA – United States of America

VI – Visibility Index

Chapter 1. Introduction

1.1. Motivation

Air transport has been considered, statistically speaking, the fastest and safest mean of transport of all. Nowadays, airports play a major role in the transportation of both people and goods all over the world. This role has been growing throughout the years and is becoming even more important in Europe than it used to, thanks to the increasing number of low-cost companies. In the city of Lisbon, Portugal, in particular, a globally positive trend has been being observed in terms of passenger flows, especially since 2009 (ANA, 2015b).

Furthermore, due to the market liberalisation and its consequences, airports have been putting a considerable amount of effort on non-aeronautical revenues, leading to a strong growth in airport retailing (Zenglein & Jürgen, 2007), namely because such sectors typically lead to higher profit margins than the merely aeronautical one (De Neufville & Amedeo, 2003, cited in Zenglein & Jürgen, 2007). This tendency has been being observed in the last decade more than ever before, in several places of the globe. In the USA, for example, non-aeronautical revenues accounted for 44.8 % of the total, in the year of 2013 (ACI, 2013).

Therefore, the quality of the services that are offered inside the airports is becoming more and more important. Being passengers the most important entities involved in an airport, their personal experiences and opinions are extremely relevant in order to measure that service quality, which airport managers aim to enhance as much as they can. Depending on several aspects (economic, technologic or safety-related), managerial measures are taken and passenger flows inside terminals need to adapt and change. Since building new spaces ends up becoming impractical at some point, both for spatial and economic reasons, these improvements should take advantage of what has already been built, whenever possible.

According to the annual report of 2014 by ANA (2014, p. 15), one of the main objectives for the management entities of the Portuguese airports for 2015 is to «ensure that the shopping area layout and offer continues to best serve [their] passengers' interests. [They] expect that, over the next three years, this effort will result in significant growth in the commercial areas in general and in [their] retail offer in particular». Therefore, and keeping in mind that making physical experiments with the spatial layout and waiting for people to react to it before drawing important conclusions would be way too consuming both in terms of time and money, the creation of a simulation model seems like an interesting way of approaching the matter.

Some research has been done throughout the years in order to try to improve airport terminals all around the world. However, not too many of these documents focus on the commercial space of the airport. In fact, most focus on the mandatory (processing) activities passengers have to undertake (such as check-in, security, immigration and boarding), which is not sufficient nor realistic when the scope is to improve the passengers' experience as a whole; other scientific documents focus on commercial spaces, either inside or outside airports; but no work was found in which a model for people's shopping behaviours was created and applied, through a simulation process, to the context of a retail space inside an airport terminal with the aim of finding a better relative position of the different facilities. Such model should be able to represent the wayfinding process in a realistic way based on people's characteristics, namely psychological. Some surveys have been made and some good practices have been discovered, which need to be kept in mind.

1.2. Dissertation objectives and research questions

As mentioned above, this dissertation aims to develop a simulation model for pedestrian movements inside an airport terminal in order to improve its retail space. In an airport context, dwell time can be defined as the time during which a given passenger remains in a specific state between two mandatory activities. This work will focus on departing passengers inside the restricted area only – that means, in physical terms, that pedestrian movements will be studied before the boarding process, but after the security mandatory checkpoint.

Putting it differently: one of the main challenges for such a model consists of creating a logical representation of the pedestrians' discretionary behaviour during their individual dwell times, harmonised with their obligations as air passengers after security. Since they are humans, they tend to behave in a rational way based on their perception of the environment they are in. When inside an airport terminal, the way they find their paths is an autonomous process, based on their cognition and basic psychological traits. This rationality, however, is not too evident from someone else's point of view, making it more defying to model it through mathematic variables and formulas. Several scientists have studied human behaviour and different ways to model the way individuals choose from a finite set of alternatives, leading to several ways of modelling individual choice and decision processes.

All in all, the present work aims to fulfil the following objectives:

1. To develop a modelling structure for passenger discretionary processes inside an airport terminal between security and boarding mandatory processes;
2. To implement that structure in a simulation software and apply it to a case study (which is the terminal 2 of Lisbon airport) and conclude about the current situation regarding the terminal's architectural layout, both from an average passenger's and a manager's points of view;
3. To apply that same model to other specific alternative layouts of the terminal and conclude whether the current situation can be improved or not through the rearrangement of the facilities inside the terminal, always from both these perspectives.

In order to achieve these goals, the following questions are expected to be answered by the end of the present work:

1. How can an airport be approached in the aim of this research? What are its characteristics, who are the agents that make part of it and what main processes do departing passengers need to go through during their journeys inside an airport terminal?
2. What should be taken into consideration regarding the layout of a retail space, and inside an airport terminal specifically? How do individual basic characteristics relate to their shopping behaviours? How can pedestrian modelling be approached?
3. How can pedestrian behaviour be modelled during passengers' dwell time between the security and boarding processes? How can the system's performance be measured in such a model?
4. Is the current situation for the case study satisfactory overall or does it need any improvements regarding relative positions for the different facilities? If so, how can these improvements be achieved?

1.3. Methodological approach and dissertation structure

In order to attain the goals and answer the questions mentioned in the previous section, the present work will be sequentially developed along the following guiding line:

- First of all, it is important to get into the airport concept with a certain level of detail, in order to know what its general structure and main agents are, so a research will be made in order to gather some of that information;
- Secondly, a literature review will be done in order to highlight relevant scientific documents published in the past that can somehow be useful in the scope of the previously outlined objectives and questions. It is also relevant to focus on what approaches and modelling tools have been used by those or other authors throughout the years, so that, subsequently, the choices of a modelling approach and a simulation software for this specific case study become easier to make;
- Thirdly, the case study needs to be analysed, in order for the author and the reader to become familiarised with its main characteristics and the most relevant ones that should be included in the newly developed model. These features will hence be described after the literature review;
- A generic model will then be created from scratch, so that it can later be applied to any specific case study. The entire reasoning behind it will be explained at this point;
- Once the model is working as supposed to, it will be applied to the present case study and some important conclusions will be drawn regarding the current situation. A sensitivity analysis procedure will then be made in order to observe how the facilities' relative positions influence the system's performance.

That will be achieved through seven different chapters:

- Chapter 1 is the current chapter. It introduces the rest of the work, explains what led the author to approach this problem, what the main objectives are and how the whole document is organised;

- Chapters 2 and 3 represent the State of the Art. In the beginning of Chapter 2, airports are defined and described. The rest of the chapter is dedicated to what some other authors have done in the past. Chapter 3, on the other hand, focuses exclusively on the approaches and tools that other authors have been using;
- Chapter 4 is exclusively dedicated to the presentation of the case study. After this chapter, the reader will feel more familiarised with Lisbon airport and its recent statistics. The general plan and facilities of terminal 2 are mentioned there, as well as the most important results of a recent survey that focused on the passengers' characteristics;
- Chapter 5 explains the whole concept of the generic model. It starts with the choice of the modelling approach and tool and continues with a detailed description of the model: its individual objects, its functions and its variables;
- Chapter 6 is where the generic model described in Chapter 5 becomes specific, through its application to the case study described in Chapter 4. Performance indicators are defined, the current situation of the terminal is analysed and other alternatives for layout are tested;
- Chapter 7 is where the most important conclusions of the work are drawn. General remarks about the simulation results are highlighted, considerations about limitations of the model are made and suggestions for future work are mentioned.

After those seven chapters, a set of appendices is presented. It contains important data that was used throughout the work but, since space for writing is limited and some elements are too long, could not make it to the core text of the document.

Chapter 2. Airport structure, airport retailing, pedestrian behaviour and shopping preferences

2.1. Airports, airport processes and airport passengers

First of all, it is important to define what an airport exactly is. Doganis (2005) defined airports as complex industrial enterprises designed in order to facilitate freight and passenger transport between air and ground transportation. In fact, an airport is often said to be an interface between ground and air transportation. As such, it is commonly divided into two parts: the landside and the airside. The latter includes all the aircraft-related areas, as well as areas dedicated to people who have already been somehow "checked" before. Only staff and passengers holding a valid boarding pass for imminent travel are allowed in the airside, whereas the landside areas are basically open to the public. In practical terms, this means that the airside covers the areas for security check, immigration and boarding processes, as well as the service areas in between and every area inaccessible to non-authorised people; the landside, on the other hand, includes parking lots, check-in areas, services before check-in and every other public area of the terminal.

Secondly, it is necessary to recall who the main intervening people are inside an airport. Kalakou (2012) distinguishes the following entities:

- The most important agents in the aim of this dissertation are the passengers. These are the people who use the airport for travelling reasons rather than seeing it as a working place, constituting one of the airport's main clients. There are three types of passengers: the ones departing from the airport, the ones arriving to the city (as their destination) through the airport and the ones that find themselves in a transferring process between two different airports. In the aim of this dissertation, arriving passengers will not be considered. As described below, all departing passenger need to go through the same common steps before boarding into their respective aircrafts. However, they are seen from a different perspective for the airport management depending on their individual characteristics, namely travel-related ones. In fact, it is important to know whether their destination is domestic or international, whether they are business or leisure passengers, whether they are transferring or not, whether they have any baggage to be checked in and what type of flight they are taking (regular, low-cost or charter);
- Airline employees, performing several possible air-related actions;
- Business people, who are there in order to gather passengers from different origins for a specific meeting at the airport, without having to reach the city centre in between their arrival and departure;
- Security agents and policemen, who assure that the airport is a safe place and take special actions in case of emergency;
- Non-aviation users and leisure visitors, who go there in order to entertain themselves with the available retail activities.

Livingstone et al. (2012) also mention another category of people called “wavers”, who go to the airport as companions of departing passengers but leave the building once they reach security.

It is now important to recall the different steps departing passengers have to go through from the moment they enter an airport. Liu et al. (2014) distinguish three key moments for them – completing check-in, passing the security checkpoint and boarding. Kalakou (2012) goes slightly more into detail and highlights the following main steps:

1) The first one is passenger arrival. Passengers usually get to the airport by road or railway, either by public or private transport. In terms of road transport, this can either consist of private cars, taxis or buses; the most usual option in terms of railway transport, on the other hand, is the metro. Once again, the airport works as an interface between different means of transport;
2) Once passengers have arrived at the airport, their objective is to catch their flights on time. Depending on the type of flight and destination, they might be up to three hours ahead of their flight’s departure time or more. During that time, some mandatory steps need to be undertaken. The first one, in case the passenger has not done so already by himself, is checking in. In fact, the check-in process nowadays does not always require the help of an assistant at the airport – in many cases, passengers have the option of doing it outside the airport building, being an internet connection all that is needed in order to do so (for some flight companies, it is even the recommended option for their passengers); otherwise, passengers can check in once at the airport entrance hall, either through self-service machines or by heading to one of the available check-in counters for their flights. The check-in process itself consists of verifying the passengers’ tickets, their personal data and their hold baggage, if there is to be any. In that case, the bag(s) always need to be checked in at a drop-off counter, being usually necessary the help of an employee for that. Some airports in the world, however, already have automated machines through which passengers can do all of these operations by themselves (Kalakou, 2012). Other (less common) alternatives consist of checking in at hotels that are somehow associated to the airport, or even some railway stations. Inside the terminal, business travellers might have their own dedicated areas and/or counters;
3) After checking in, passengers can directly head to the next mandatory step, which is security check. Alternatively, they might spend some more time in the entrance hall of the airport, where several services and facilities are usually available in order for them to spend time and money, if they are willing to do so. People can obviously use those services before checking in as well but, statistically speaking, they prefer to save them for the moments after that first step of their journeys has been accomplished;
4) At the security control step, each passenger is individually checked, as well as all of their hand luggage. Personal belongings and hang luggage are placed on a belt in order to be scanned, while the passenger goes through a metal detector. Any object or container considered to be somehow dangerous is retained at this stage. During this process, no distinction is made between passenger categories – they all wait in the same queues and the process is the same for every single person.

5) In European terminals, if the passenger is flying between two Schengen destinations, they are ready to spend the rest of their time until boarding on diverse activities; if not, they have to add one more step in between, which is passing through the Immigration and Border Control;
6) Between security control and the boarding process, passengers are invited to enjoy the various services offered by the airport terminal. These include retail shops, restrooms, Wi-Fi… among other activities that will be highlighted in the following sections of this document;
7) Once the aircraft is ready, the boarding process begins, which is the third and last mandatory step before take-off. Before actually entering the aircraft, passengers queue in order to have their passports/I.D. documents and boarding cards checked by the staff – this is called gate control. A distinction in the queues can be made according to the passengers' types of tickets, since some types might have priority over others.

While inside the airport, and throughout these steps, a passenger has the possibility of undertaking several different activities, as long as some temporal constraints, defined by the key moments specified above, are accomplished. Each one of these three moments works as a transition between phases (Liu et al., 2014). These "extra" activities will be described in the following subchapter. Figure 1 was taken from Kalakou (2012) and illustrates, through a diagram, the global path and different possibilities one has at each moment when departing from an airport.

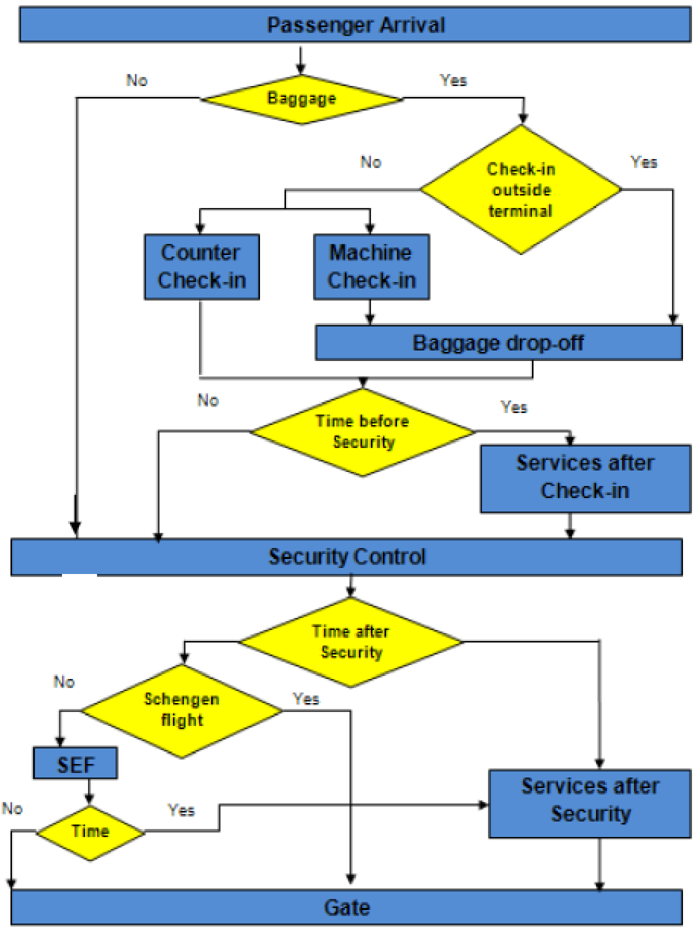


Figure 1. Conceptual diagram of the different steps departing passengers need to go through once inside an airport (Kalakou, 2012)

Having described the general definitions of airport spaces, agents and passenger processes, it is now time to focus on research that has been done in previous years that might somehow be useful in the aim of this work.

In the following subchapters, a general description of the factors that influence people's shopping behaviour and preferences is presented, as well as one possible theory about how the individual choice process is made in mathematical terms. The objective is to collect the necessary data one should understand and take into account in order to implement certain improvements in a given retail space layout. Also, important numerical data regarding useful statistic parameters will be highlighted here. Results will be drawn from other authors' previous studies and conclusions, and will be presented through a logical guideline. At the end of the chapter, some general comments and conclusions will be made.

2.2. Available activities for airport passengers

It is worthwhile for this work to make a list of possible activities that might be undertaken by pedestrians when inside an airport, including the ones somehow related to retailing and alternative sources of income.

As previously mentioned, Liu et al. (2014) divided the entire process in three phases. In each one of these phases, five activity categories are considered in their article: wait, inquiry, dine, shop and facility use. This accounts for fifteen (three times five) possible alternative activity scheduling patterns, considering only one activity per phase. The authors analysed their model through a nested logit model, which used a tree structure based on econometric error variances among alternatives, rather than using a more intuitive behavioural logic. Simulation and results were obtained thanks to a web-based preference survey that revealed how frequent each one of these activities was in each stage. Thanks to those results, three new tree structures were proposed – two intuitively and one through the Heteroskedastic Extreme Value (HEV) model, which proved to be the most plausible according to the survey results. This final model divided the activities into four nests:

- The "more frequent" nest included the following activities: inquiry before check-in, inquiry before security, inquiry before boarding, waiting before boarding, facility use before boarding and dining before boarding;
- The "less frequent" nest included dining before security and waiting before boarding;
- The "shop" nest only included shopping before security;
- The "time killing" nest included facility use before check-in, facility use before security and waiting before security.

Although the authors think that further studies and improvements should be made in their models, this tree structure and survey results are relevant in that they illustrate the frequency of facility use as taking place mainly before boarding (but also before check-in).

Livingstone et al. (2012) divided activities into processing and discretionary: the first category stands for activities that need to be done in order for passengers to get to boarding (in their article, which focuses on Australian international airports, there are four of those – check-in, security check, customs and boarding), whereas the latter type of activities regards the ones undertaken by passengers during non-processing times, including retailing and others (which are not directly specified). Pedestrians were found to spend 55 % of their time in processing activities and 45 % in discretionary ones. Several other authors have also used this type of distinction.

Smits et al. (2013) focused on the first area passengers get into contact with when entering an airport, which is the one comprising the check-in counters and/or machines. In their article, the authors distinguish and simulate four different possible activities at this stage – going to the toilet, leaning against a pillar, standing in the queue for the check-in desk and actually checking in.

Ma (2013), on the other hand, created a simulation model in order to represent and study people's complex behaviour inside an airport and the associated pedestrian flows, taking into account both processing and discretionary activities. In his work, each passenger is assumed to have basic and advanced traits. Basic traits are the ones that are typically static whilst passengers are at the airport and they are not specific to any activity in particular (gender, age, baggage, travel class, frequency of travel, travel group size and nationality), while advanced traits might vary throughout the process and strongly influence the decision-making process of an individual at a given point in time and space, as well as the route choice, being therefore very relevant in order to detect and explain more complicated behaviours. The author made sure that each passenger was an individual agent who both perceived and reacted to its environment. Different times and locations (and combinations of both) lead to different activities. Some examples mentioned by the author are shopping, eating and drinking, interacting with other people, going to a restroom, praying, making a phone call, using a cash machine, using social connectivity, waiting… Among others.

As a matter of conclusion: this subchapter showed that, among the cited bibliography, authors have been adopting quite different approaches regarding the available options for a passenger to undertake inside an airport. In some cases, a "list" of activities was not a main objective for the authors to determine, but just something they had to take into consideration at some point. Livingstone et al. (2012), in particular, simply defined discretionary activities as all the activities that are not mandatory. However, among the highlighted possibilities above, it can be said that authors ended up reaching similar conclusions: both Livingstone et al. (2012) and Ma (2013) make a distinction between processing and discretionary activities; Ma (2013), in turn, included in his list all of the five activities highlighted by Liu et al. (2014), being the latter's division more general, rather than exhaustive. Smits et al. (2013), on the other hand, considered some more specific activities, such as leaning against a pillar or a more precise division inside the check-in processing activity, but all in all they can be compared to the other author's options.

It is important to keep in mind that each terminal has its specific set of options for discretionary activities to be undertaken by its passengers, and each case should be studied individually. Processing activities are the ones that can be more easily generalizable regarding different locations in the world.

2.3. Airport retailing, individual characteristics and shopping preferences in general

2.3.1. Analysis of previous research

Having specified some of the activities that can be undertaken by the passengers throughout the different phases of their journey, it is now important to know what their preferences are regarding several aspects, as well as what causes them and what derives from them. Such relationships have been studied by a considerable number of authors throughout the last decades.

It is common to divide people into different categories and define the main characteristics for each one of them. This segmentation allows researchers to determine what variables have an actual influence on people's behaviours and is, according to Bamberger et al. (2009), one of the five key success factors for airport retailing.

Kalakou et al. (2014), for instance, mentioned five different scopes for segmenting passengers:

- Destination: international or domestic;
- Trip purpose: business or not;
- Role of the airport in the trip: final destination or transfer;
- Type of flight: regular, low cost or charter;
- Baggage: whether there is baggage to be checked in or not.

Geuens et al. (2004), on the other hand, developed a whole typology for airport shoppers. Their analysis was made through a hierarchical and a K-means cluster analysis on four main shopping motivations revealed by a questionnaire. Indeed, there were four dimensions that explained more than half of the variance in the results: one airport-infrastructure related, one airport-atmosphere related, one experiential and one functional dimension. In order to draw the passengers' shopping profiles, chi-square and one-way ANOVA analysis with cluster membership, as well as sociodemographics, were used. Three types of shoppers were defined: mood shoppers, apathetic/indifferent shoppers and shopping lovers. People from the first category showed high and moderate levels on atmosphere-related and experiential components, respectively, while they did not seem to value the other two components that much. They were basically travel-related buyers; shopping lovers, on the other hand, valued all four components, even though not the atmosphere-related one as much as the other three; finally, apathetic shoppers revealed low levels on all four components. The surveys included questions about personal airport shopping preferences, as well as socio-demographic data (gender, age and nationality), travel behaviour (frequency and capacity) and purchase behaviour at the airport (frequency, shop location at the airport, preferential type of shop). In terms of individual characteristics, age did not appear as a very relevant factor; neither did travel frequency and capacity. Apart from those: mood shoppers were essentially male, impulsive shoppers who would rather have centralised stores; apathetic shoppers were also mainly male and regarded airports as the "poles" of their journeys, rather than shopping places. If they decided to buy something there, they did it in a pre-planned way; shopping lovers, however, were mostly female. They felt that the travel itself puts some pressure on them and tended to buy impulsively. They usually preferred less but bigger shops over more but smaller ones, and ideally these should be located near the departure gates.

Some years later, Chung et al. (2013) made a distinction between four different types of consumers: three of them are the same as Geuens et al. (2004) (mood shoppers, apathetic shoppers and shopping lovers); the forth one was named "traditional shoppers" and included people who always pre-planned their purchases and restricted them within a given budget. In their article about information seeking behaviour of passengers, they concluded that apathetic shoppers were the ones who exhibited the lowest information seeking effort, both before their trips and on-site. The authors separated two types of convenience: decision and access convenience. The latter is the one related to the airport atmosphere and layout, so it is the most relevant one for this dissertation. It is defined by the service environment, the consumer information and the service system design. The higher the access convenience for the passengers, the more comfortable they will feel inside the terminal, which is one of the main objectives of this study.

It is noteworthy, however, that Geuens et al. (2004) regarded their results as being valid for Belgium (since the surveys took place in Brussels airport) but considered it important to generalise them for other countries by adopting the same kind of procedures and studies elsewhere. Indeed, within the bibliography that is here cited, other authors reached different results and/or conclusions. Lin & Chen (2012), for example, found out, through surveys made inside Taiwan Taoyuan International Airport, that business travellers were usually submitted to higher levels of time pressure and therefore ended up spending less time in the terminal, when compared to leisure travellers. Lu (2014) also concluded that people travelling for work shopped less than the ones travelling for leisure purposes, showing once again that shopping capacity was indeed a relevant aspect to keep in mind. Also, results from Denver International Airport (2009) detected different spending patterns between business and leisure travellers that should be taken into account. Bamberger et al.'s (2009) study about the best practices regarding airport retailing also mentions the same conclusion.

Freathy & O'Connell (2012), also, made some surveys and interviews and concluded that business travellers had some clear spending patterns. In other words, they found that travel capacity (flying for tourism or business) was actually relevant for shopping behaviour, since business passengers had, in general, higher levels of income, and they only had limited time to shop in the country they are visiting. Indeed, they were found to travel more frequently, to spend less time shopping, less time in the commercial area in general, and to buy less items (in case they buy any), since they travelled more frequently and usually did not really need anything. However, if they did need something, they were found to often end up spending above the average value, even though they did not need to spend too much time browsing since they were more familiar with the terminal's layout.

In their article, the same authors registered that a passenger spent, on average, 33 minutes in the commercial space, although individuals were found to behave in very different ways – some might go through the commercial area without even stopping, while others had the opposite attitude and ended up spending a long period of time there and undertook several different activities. Previous research had demonstrated that, the longer the dwell period, the more likely it was for an individual to make a purchase. Three groups of activities were distinguished: shopping, eating/drinking and spending time on non-commercial pursuits. Table 1 illustrates the results as measured by the authors, by giving the percentages of passengers that spent different time intervals undertaking each one of these types of activities.

Table 1. Activities undertaken while in the commercial area of the airport in Dublin (Freathy & O'Connell, 2012)

	Activity minutes						
	0	1-10	11-20	21-30	31-40	41-50	51+
Total time shopping (%)	12.3	22.9	21.9	19.9	13.3	5.6	4
Total time eating and drinking (%)	56.5	9.6	20.3	7.6	3	1.7	1.3
Total time on non-commercial activities	27.2	69.4	3	0.3	0	0	0

The table shows that 12.3 % of the 301 respondents did not spend any time at all undertaking shopping activities. Globally, the ratio between the number of passengers who purchased and the total number of travellers (which is defined as the penetration rate) was 85.7 %, with an average of 1.75 transactions/passenger. In terms of shopping behaviour, people were found to visit, on average, three stores, and the average time per visit was found to be over six and a half minutes. However, only 12 respondents out of 301 decided to go to the same store more than once, showing that the most natural tendency is not to deviate too much from the main route that leads to the departure gate; in other words, preferably not to go back on their steps at any point.

But what are the relevant variables in order to define the different categories for shopping behaviours? Several studies have been made about this throughout the years, either regarding specifically airports or just commercial spaces in general (shopping malls, for instance).

Freathy & O'Connell (2012) defined that the main explanatory variables regarding differences in shopping behaviour were gender, the composition of the group, the flight duration, the frequency of travel, the purpose of travel, the duration of the visit and the number of flights taken in a one-year period. Indeed, female passengers were found to spend considerably more time shopping; also, passengers who travelled alone spent less time in the retail space than people travelling in groups, and passengers with children tended to dedicate shorter periods of time to shopping activities than groups without any child; finally, people who were travelling out and returning within the same day usually spent less time and money shopping.

Livingstone et al. (2012) found out, through video footage in two Australian airports, that passengers travelling alone did not spend indeed much time in retail but were the ones that most probably would purchase something; passengers with travel companions or wavers (people who accompany the traveller but do not go on board) were found to be more likely to browse without actually buying. Their research only focused on the landside retail experience, though.

Wesley et al. (2006) focused on how consumer decision-making (CDM) styles of individuals were related to their global evaluations of shopping malls, by creating a theory for antecedents and consequences of CDM styles. They did a study through heterogeneous personal face-to-face interviews (heterogeneous in terms of mall locations and dimensions, interviewed population, time of the day/week…) and concluded that different CDM styles existed among the consumers and that they strongly related to their satisfaction with the malls. Gender was said to be the most influential demographic characteristic on CDM styles – age, education and income were found not to be that relevant in their study, after all. One very influential factor on the customers' activities was the individual planned expenditure before going inside the mall, which was, amongst other factors, defined by the mall type and the CDM style. Demographics were found to be relevant for the planned expenditure, especially income, and males were found to be more sensible about it than females. The higher the planned expenditure, the higher the satisfaction, which positively influences the intention of visiting the mall again in the future. Authors also concluded that CDM styles were not dependent on the mall type and that they were not as relevant as expected to justify the activities undertaken by the customers once inside the mall (which was against one of the authors' hypothesis), but they clearly affected global satisfaction; global satisfaction, overall, was also determined by the CDM style, but not by the activities undertaken inside the commercial space. It is however important to mention that one does not follow the same CDM style in all shopping decisions.

El Hedhli et al. (2013) also focused on the well-being of the customers inside a shopping centre, which contributed to their global satisfaction and therefore a higher success for the mall, namely higher loyalty and word of mouth. Shopping well-being was said to be the same as consumers' well-being, which was defined as the perceived impact of the shopping mall on the users' life quality, that is, how much it influenced some important life domains (consumer life, social life, leisure life, community life). In fact, several factors were considered as being determinant for one's overall perception of quality while visiting a mall: besides the previously mentioned life domains, some sub-domains were taken into account as well, such as product acquisition, assembly, ownership, consumption, maintenance or disposal, but also factors like entertainment and socialization with the retail personnel. The authors defined six factors as being influential for the customers' well-being: functionality, safety, leisure, atmospherics, self-identification and convenience. The latter predictive factor is the one in which the space layout plays an important role. The necessary input data for this study were obtained through intercept surveys made in two urban Canadian shopping centres at different days of the week, different times of the day and different mall entrances. In the end, all of these six predictive factors were statistically proven to influence well-being in a positive way, which was in line with the authors' hypothesis. Shopping well-being, in turn, led to higher levels of loyalty and word of mouth. However, the authors did not assure the validity of their results in general terms and thought that similar studies should be made in other types of commercial spaces.

In their (already mentioned previously) study, Chung et al. (2013) also took several variables into account. A questionnaire was made in order to obtain the necessary input data for analysis and, within the questions that were included in the survey, three main sections were defined: social demographic data (age, income, travel frequency, recent airport purchases and frequent flyer program membership), information about the passenger's last trip using Taoyuan International Airport (trip purpose, travel company size, goods purchased, purchased amounts, waiting time before boarding, flight time, pre-trip and on-site information seeking, decision convenience and repurchase intentions) and non-airport specific information (shopping characteristics, most recent purchase and purchase decision attitude). Fourteen explanatory motivation factors were considered as statistically relevant and were divided into three categories:

- The functional dimension included motivation items such as convenience, local goods/specialties, attractive prices, service inside the shops, width of the product range, speed of the check-out service, product quality, possibility of paying with different currencies and multilingual communication in the shops;
- The experiential dimension was based on impulse, surrounding atmosphere and boredom of the passengers;
- Finally, the rational dimension consisted of pre-planned purchases, as well as considerations based on cost.

Lu (2014) studied, in his article, how socio-demographic characteristics of passengers, as well as their trip characteristics and perceptions of airport shopping, influenced their shopping motivations inside the airport. In order to do that, he made some surveys and created six different factors, as well as two representative types of intentions, and created a model in which the factors were independent variables and the types of intentions were dependent. The six factors considered as being relevant and reliable were named Enjoyment, Values, Environment, Famous, Services and Conveniences; the two types of intentions were Impulse shopping and Pre-planned shopping intentions. After analysis of the results, all six variables were considered to be statistically relevant, being Enjoyment, Values and Environment the three most relevant ones for pre-planned shopping and Values, Environment and Famous the top three for impulse-buying tendencies. Male passengers were the ones that shopped the most impulsively; older passengers, as well as customers travelling with friends and relatives, tended to pre-plan their shopping ideas rather than having an impulsive shopping attitude; higher incomes, on the other hand, led to more spontaneous shopping in general; as previously mentioned, there was a distinction between passengers travelling for leisure or for work, since the ones travelling for leisure ended up showing more significant shopping tendencies overall in this study; also, the more frequently a passenger travelled, the more spontaneously he tended to buy at the airport; finally, passengers who had never shopped in an airport did not tend to do so.

As previously mentioned, Lin & Chen (2012) studied, through the analysis of the surveys they made at Taiwan Taoyuan International Airport, the effects of time pressure on passengers' shopping behaviour, while analysing the relationship between shopping motivations and activities. They distinguished three different categories for shopping motivations – "favourable price and quality", "environment and communication" and "culture and atmosphere". The two latter ones would be the most relevant ones in the aim of this dissertation, since the authors found that they were very relevant for food, drink and leisure activities at the airport. Time pressure, on the other hand, was found to be a moderating variable between luxury and travel products and the "environment and communication" motivation. Impulse buying, however, had a significant negative moderating effect between those two elements. Neither the impulse buying nor time pressure had any moderating effects on dining and recreation activities. However, once again, the authors recognised that their results might not be fully generalised worldwide, since they were focused on Taiwanese citizens' responses only.

Freathy & O'Connell (2012) also mentioned the distinction between pre-planned and impulse purchasing. No relationship was found between passenger segmentation and the likelihood of making an impulse purchase. Actually, most passengers were found to pre-plan their shopping activities, even though some of them would not discard the possibility of buying larger and more specific items impulsively. Table 2 illustrates the percentages of pre-planned and impulse purchases by different categories.

Table 2. Percentages of pre-planned and impulse purchases depending on shopping categories (Freathy & O'Connell, 2012)

	Planned	Impulse	Total
Personal purchase	255 (74 %)	88 (26 %)	343 (100 %)
Gift purchase	104 (68 %)	50 (32 %)	154 (100 %)
Total	359	138	497

Crawford & Melewar (2003) focused on the importance of impulse tendencies on airport passengers, which have been mentioned in the previous paragraphs. It has been quite a largely investigated topic over the past few decades by several researchers and the authors defended that it could considerably increase the productivity of a commercial space, if properly understood and explored by retailers. They pointed out ten explanatory factors for impulse shopping in airports: holidays, gift giving, occasion driven (Christmas, Easter, etc.), forgotten items, airport exclusivity by retailers, disposal of foreign currency, confusion (driven by too much information by the retailers), self-indulgence, guilt (most applicable to business travellers who want to compensate their families for being too absent) and value driven by retailers through advertising. The authors defined an "airport impulse maximiser" as being constituted by four main elements: stress and anxiety reduction, browsing induction, normative traits reduction and pure impulse. All of them would lead to a higher propensity for the consumer to spend money. Table 3 illustrates a more detailed formulation of the necessary measures in order to accomplish such objectives, according to the authors.

Table 3. Impulse strategy formulation (Crawford & Melewar, 2003)

Airport impulse maximiser elements	
Reduce stress and anxiety	Locate flight screens in visible locations. Provide customer service representatives in central area to assist passengers. Provide clear signage for departure gates and zones.
Induce browsing	Maintain the level of inherent excitement. Minimise onset of boredom. Increase store penetration. Provide good store layout, positioning of outlets in highest volume areas. Concentrate on providing an open and attractive retail environment. Use of psychological tactics to improve penetration.
Reduce normative traits	Reinforcement of 'value' through visible price comparisons. Stress the rationality of impulse buying in advertising efforts. Stress the non-economic rewards of impulse buying. Risk-free purchasing — guarantees, warranties and money back promises. Well-trained sales staff. Maximise the effects of the happy hour. Create a more complex environment, straining customers' abilities to process information accurately. Minimise transaction times. Maximise convenience.
Pure impulse	Cater for the '10 impulse stimuli'. Innovative location of identified impulse lines. Utilisation of trained promotional staff to maximise in-store spends.

To increase impulse buying tendencies and to induce a more intuitive browsing for passengers should therefore be priorities for airport managers. These concepts, especially the ones related to the induction of browsing, are extremely interesting in the aim of this research. Excitement levels, however, should remain at their natural levels.

Perng et al. (2010) also studied passengers' shopping processes at Taiwan Taoyuan International Airport. They focused, more specifically, on their responses to retail product categories for shopping purposes and satisfaction by designing a questionnaire for a survey, which took place at the entrance hall of the airport. The available terminal retail products were classified into seven groups: “utility”, “entertaining”, “brand-name”, “low-cost”, “service”, “souvenir” and “café products”. The analysis and review of the shopping requirement led to the determination of product priority. Shopping behaviour and satisfaction were analysed sequentially. Results showed that leisure passengers tended to choose their travel-related items at the airport, rather than pre-plan such purchases. Statistic tests evidenced that age, gender and the companion of the passengers were relevant factors for shopping decisions. For all the interviewed people, useful items and souvenirs were preferred over other options (either for a specific occasion or for recreational purposes). Figure 2 illustrates the rankings (from 1 to 7) for product preference and satisfaction for each one of the seven groups.

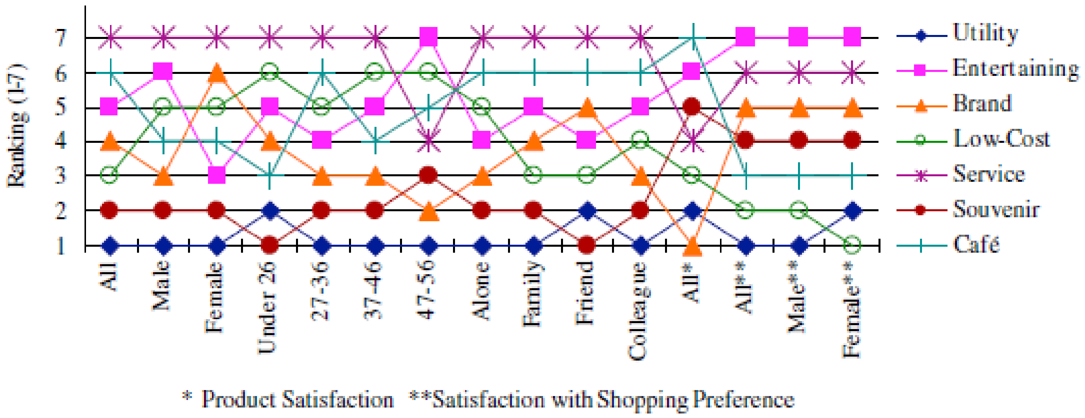


Figure 2. Rankings for product preference and satisfaction (Perng et al., 2010)

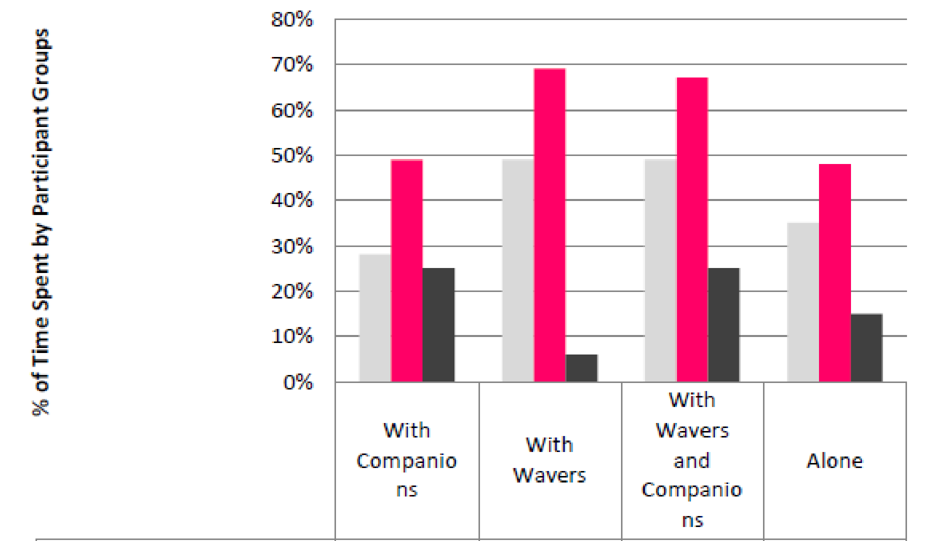
In the same study, passengers who spent money at cafés generally did so impulsively, but it was a weakly represented shopping category overall. Men were found to be more attracted to brands than women, while women were more likely to buy entertaining products than men. Passengers travelling alone were not very attracted to low-cost products and preferred brand-named items, contrarily to the ones travelling in groups, even though colleague-accompanied passengers also had a tendency to purchase brand-named products.

Concepts other than the merely commercial ones are important when studying pedestrian behaviour. Van Oel & Van den Berkhof (2013), for example, investigated how architectural design characteristics influenced airport passengers’ perception and satisfaction regarding that space. Data were collected in departure and transfer areas at the Amsterdam Airport Schiphol. In that study, questions were asked to the passengers by showing them visualizations of several hypothetical passenger areas and the influences of emotional states over the appreciation of these design characteristics were studied. A multinomial logit (MNL) model was used to estimate the design factors (attributes) from the choices the respondents made within the alternatives. Interviewed people were asked about their travelling reasons, their demographics, their education levels and their gross annual incomes, but also, for the options proposed, about the materials, the exterior/interior architectural designs, the layout, the point-of-purchase and decoration and the human factors. In general, passengers preferred an area with a curvilinear roof, a curved hallway, the presence of greenery, no decoration emphasizing the distinctiveness of Holland (since the airport was in Amsterdam), the use of warm lighting, a wide dimensioning and an emphasis on white materials. Signage was not involved in passengers' decision-making process. The authors concluded that it was important in logistic terms, but not that much in terms of space perception by passengers. Overall, the results were in line with previous studies. The authors highlighted the importance of taking into account some eventual cultural differences within the passengers as well.

Denver International Airport (2009), however, found through surveys that 42 % of the people figured out what shops and services were available at the airport thanks to airport signage and directories, which is still a significant percentage. However, the predominant answer in that study was "just by walking around", selected by 88 % of the participants. 35 % also mentioned past experience as a factor, and 10 % asked people around for directions.

Two studies which illustrate, through real numbers and statistics, what airport passengers' preferences are, will now be highlighted, as well as their main conclusions. Both have already been mentioned throughout this text.

The first one was written by Livingstone et al. (2012) and focused on landside retailing only, i.e., from the entry of the airport to the Liquids, Aerosols and Gases security checkpoint. Overall, people were found to have spent 14 % of their time in retail activities. If only the ones who entered the retail area were considered, they spent 24 % of their time in such activities. The authors focused particularly on the passengers' companions, as shown in Figure 3.



	With Companions	With Wavers	With Wavers and Companions	Alone
% of overall dwell time on landside	28%	49%	49%	35%
% of landside dwell time in discretionary	49%	69%	67%	48%
% of landside dwell time in retail	25%	6%	25%	15%

Figure 3. Different time statistics depending on different passenger groups (Livingstone et al., 2012)

Figure 4, on the other hand, shows which groups ended up actually purchasing and the respective percentages for each one of them.

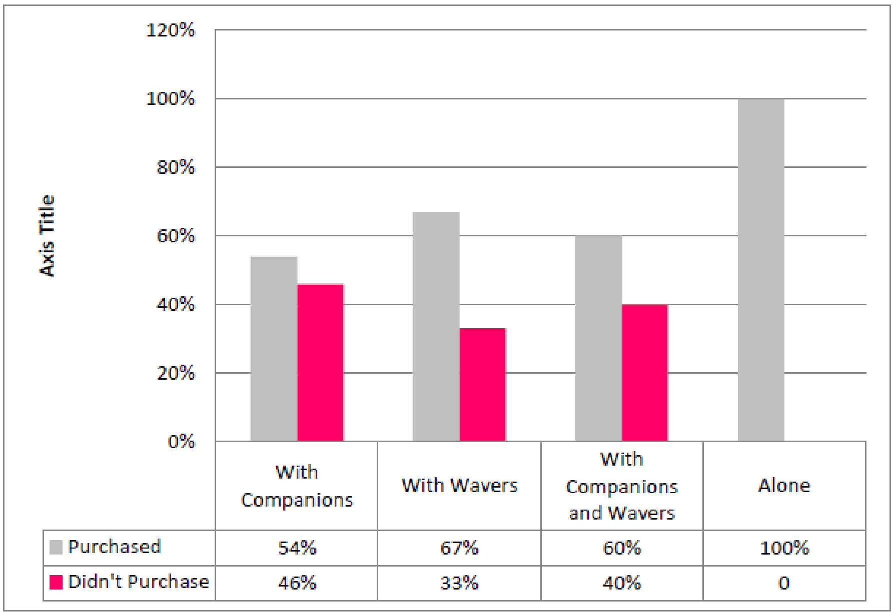


Figure 4. Passenger groups who made landside retail purchases (Livingstone et al., 2012)

The second study was made by Denver International Airport (2009) and shows all the results obtained through the 463 surveys that took place at the airport in January 2009. The surveys covered both weekdays and weekends, as well as the three concourses (big open areas) in the airport, besides the main terminal. All respondents were above 18 years old and had a plane to catch the day they were interviewed. Only 5 % were arriving passengers – all the others were either departing or transferring passengers. 56 % were travelling for leisure, 31 % for business and 13 % for both. Some of their main conclusions were the following:

- 49 % of the passengers visited the shops while looking for something to do; 19 % were specifically looking for something to buy; 32 % visited the shops for both reasons at the same time;
- 90 % of the people stopped and looked at one or more retail stores; 72 % of the total number of people (80 % of those 90 %) ended up purchasing something and spent, on average, $ 13.03;
- Passengers with more than three hours of waiting time (16 % of the population) were more likely to both stop and look (68 % versus 43 % of other passengers) and purchase something (47 % versus 17 % of other passengers) at a sit-down/dine-in restaurant. And while they were also more likely to stop and look at a newsstand or bookstore (70 % versus 48 %), they were not significantly more likely to purchase something from one of these concessionaires compared to other passengers (20 % versus 17 %);
- Passengers who purchased something in the central area of their concourse were more likely to be transferring passengers (46 % vs. 38 %) and were more likely to be travelling with one other person (30 % vs. 22 %);
- Passengers who made purchases near their gates were also more likely to be transferring, to be travelling with two other people, to be females and not single;

- Restaurants generated the highest revenue per passenger. Two thirds of the respondents at least stopped to look at one take-out restaurant and nearly half ended up purchasing something in there, spending an average of $ 9.70. Regarding dine-in restaurants, on the other hand, nearly half of the respondents stopped and looked, and one quarter decided to eat something in one of them;
- Half of the passengers stopped and looked at newsstands/bookstores but only 17 % actually bought something in there;
- Similarly, 41 % stopped and looked at gift shops but only 7 % purchased something;
- Leisure passengers spent around 3 dollars more than business passengers and tended to stop, look and purchase more, except for gift shops, where business travellers went more frequently;
- 66 % of the respondents considered it to be likely or very likely for them to purchase something near their departing gates, whereas 39 % gave that answer regarding the central part of the concourse, 13 % for the main terminal and only 5 % considered going to a different concourse;
- Similarly, 56 % of the people would only consider looking for a place to shop or dine within sight of their gates or a few minutes from it; 27 % would be willing to walk until the centre of the concourse, 12 % until the end of the concourse, and only 5 % would possibly look into a different concourse.

Figure 5 and Figure 6 illustrate, through bar graphs, the overall probabilities of someone purchasing something, depending on the shop type, shop location and previously undertaken activities. For example, 26 % of the people who ate at a dine-in restaurant also purchased something at a newsstand or bookstore. Similarly, 44 % of the people who purchased in the centre of the concourse also did so near their gates.

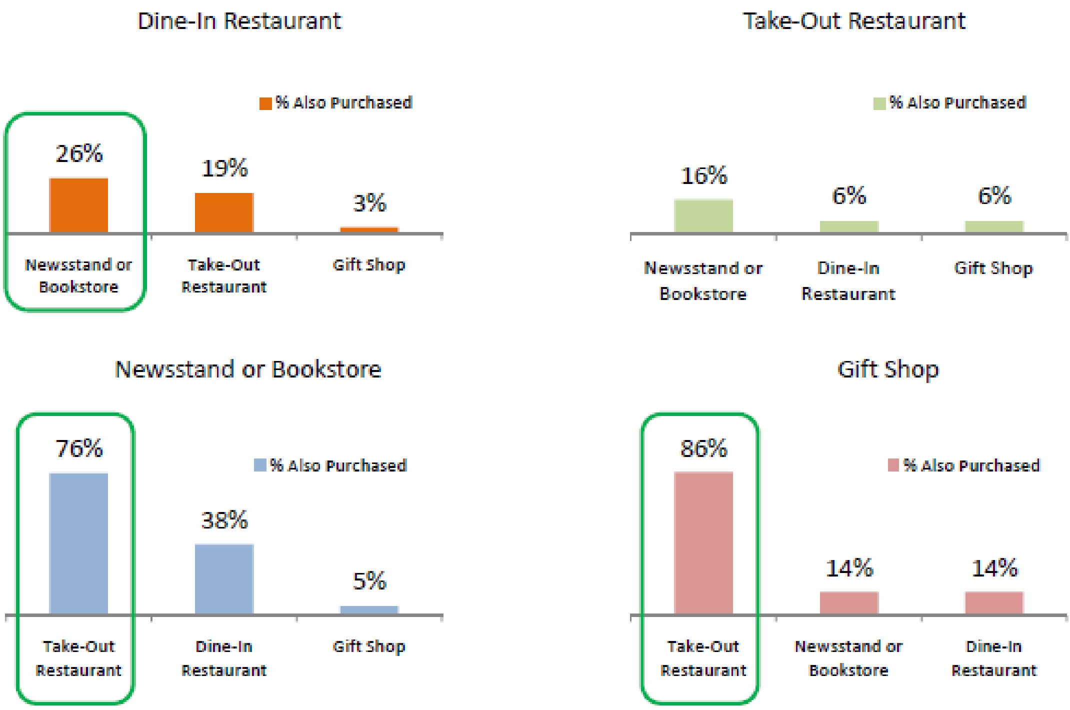


Figure 5. Purchasing probabilities depending on the shop category (Denver International Airport, 2009)

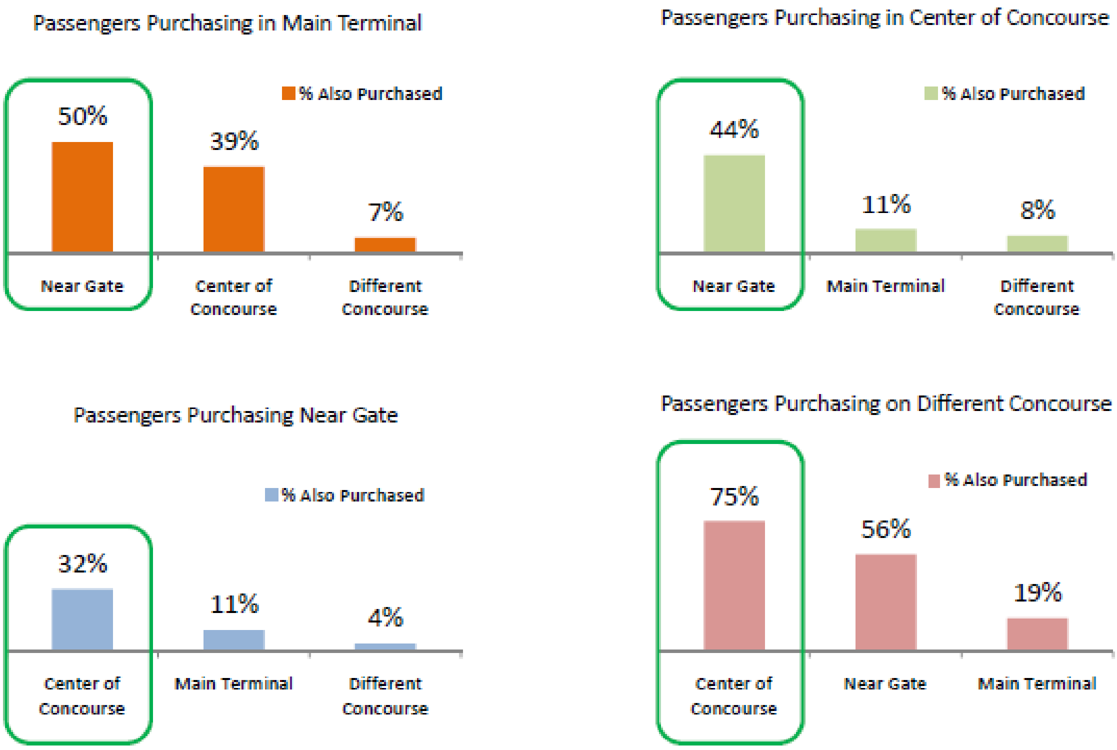


Figure 6. Purchasing probabilities depending on the shop location (Denver International Airport, 2009)

As previously mentioned, Ma (2013) wrote a dissertation specifically about an agent-based model of passenger flows in airport terminals. By studying the retailing statistics in four Australian airports, he concluded that passengers spent at least 12.5 % of their total time in the duty-free stores. Later on, in the chapter concerning simulation outputs and analysis, he used a set of data from an Australian airport regarding each possible activity to be undertaken by pedestrians.

Table 4 illustrates the different ancillary activities considered by the author during passenger dwell time, as well as their probability distributions and intervals.

Table 4. Planned passenger dwell time depending on the ancillary facility (Ma, 2013)

Ancillary facility	Dwell time (s)	Distribution
Shop	300-450 landside 600-750 airside	Normal, alpha = 371 s Normal, alpha = 685 s
Take-away	30-120	Uniform
Café	1650-1750 landside 1300-1400 airside	Normal, alpha = 1709 s Normal, alpha = 1333 s
Internet	1600-1700	Uniform
Restroom/Baby care	160-230	Uniform
ATM (cash withdraw)	60-70	Uniform
Money exchange	140-190	Uniform
Information kiosk	5-60	Uniform
Phone booth	60-300	Uniform
Prayer room	780-900	Uniform

Such figures will be important in order to model passengers' behaviours, as pretended in this dissertation. The methodology used by the author will be mentioned in the next chapter as well.

Other scientific documents try to summarize conclusions from previous research and put them all together in what the authors call "best practices". Bamberger et al. (2009) had a document published aiming to help the reader understand and know how to improve airport retailing. Five key success factors and measures were listed:

- Retail surface density should be developed until maturity, which was estimated to be at around 1000-1200 m^2 per million departing passengers (including both airside and landside, but excluding food and beverages) and would guarantee sales around 18 € per passenger;
- Available offer should be adapted to the local passenger profile, including prices, brands and products;
- Passenger footfall should be optimised as well, by positioning stores in a strategic way: core business categories should be placed immediately after security checkpoints in order for passengers to rapidly buy their eventually pre-planned purchases and hence be more receptive and comfortable for other (impulse-originated) purchases; last-minute offers for top-selling brands in core categories should exist near the departure gates, so that business travellers (or other people who do not spend that much time in the airside) are captivated and induced to buy; also, walk-through shops help passengers look and become aware of the products. However, there should always be enough communication to ensure that the passengers know how much time they have left before boarding and to reduce their pressure;
- To purchase items inside the airport, rather than in the city, should always be advantageous in terms of pricing of the products;
- A proper passenger segmentation should be made, leading to adequate sales channels.

Perfumes and cosmetics, as well as confectionary and fine food, were the two categories which the authors predicted would have the highest increase of sales between 2008 and 2012 (above 10 % per year); alcohol, fashion and accessories, on the other hand, were predicted to increase their sales between 5 and 10 % per year, while tobacco would have the lowest annual increase rate, no higher than 5 %. Overall, duty-free airport retail was predicted to increase around 9 % per year during that period of time. However, the authors were still uncertain about the effects of the economic slowdown during that period of time, so the actual results in the meantime might have been different.

Finally, and mentioning global economics, Hampson & McGoldrick (2013) decided to investigate about the effects of an economic crisis on shopping behaviour. Their objective was actually to focus on both sides: to study the consequences for retailers and other marketers and to propose a whole new typology for shoppers in such a context. In order to do so, they adopted a literature review and qualitative research with both managers and consumers. A 1211-people sample was used, representative in terms of gender, age and major regions. Four clusters of people were created, depending on how they reacted and adapted themselves to the economic crisis – the maximum adaptors, the caring thrifties, the minimum changers and the eco-crunchers. Two of them (minimum changers and caring thrifties), representing approximately 50 % of the sample, did not appear to change their habits considerably and therefore had an important effect over retailers and marketers, since they contradicted the common tendency of lowering the prices and adapting their loyalty programmes. In fact, minimum changers simply were not open to changing their habits in general, and caring thrifties maintained ethical spend and store loyalty, although they did care about the economic recession and became more concerned about their money spending in general, since that was the global tendency; maximum adapters, on the other hand, were the ones who adapted their habits the most and actually started shopping less frequently. They were mainly females and represented 20.1 % of the respondents; finally, eco-crunchers were older consumers who deliberately adopted much higher cuts regarding ethical and charitable spending. Overall, the authors confirmed results from previous studies and concluded that, during an economic crisis, shoppers become more knowledgeable and worried about the value of money than in "normal" situations. Such tendencies are relevant to be kept in mind when such an economical context has important effects on a given case study.

2.3.2. Conclusions

Several conclusions can be taken from this analysis. Authors, overall, have agreed with each other and have reached similar conclusions. In terms of passenger segmentation, the distinction between groups is usually similar. For example, Geuens et al. (2004) and Chung et al. (2013) had people divided into three types of shoppers in common: mood shoppers, apathetic/indifferent shoppers and shopping lovers. Chung et al. (2013) added traditional shoppers to that list as well. In terms of commercial space, mood shoppers are the ones who are most sensible to atmosphere-related and experiential factors and would prefer to have centralised stores, while shopping lovers would rather have fewer but bigger shops, closer to the departure gates. Hampson & McGoldrick (2013), however, defined a different shopper typology for a context of recession: the maximum adaptors, the caring thrifties, the minimum changers and the eco-crunchers. The general tendency for shoppers in such context is to become more knowledgeable and worried about the value of money than in "normal" situations.

Also, the relevant variables that researchers have considered as being explanatory are often the same (even though some authors go further into detail than others). The variables that commonly appear as possibly relevant for shopping behaviours are gender (Freathy & O'Connell, 2012; Geuens et al., 2004; Ma, 2013; Perng et al., 2010; Wesley et al., 2006), age (Geuens et al., 2004; Ma, 2013; Perng et al., 2010; Wesley et al., 2006), nationality/culture (Geuens et al., 2004; Ma, 2013; Van Oel & Van den Berkhof, 2013), income (Perng et al., 2010; Wesley et al., 2006), group composition (Freathy & O'Connell, 2012; Livingstone et al., 2012; Ma, 2013; Perng et al., 2010), flight/trip characteristics (Freathy & O'Connell, 2012; Ma, 2013; Perng et al., 2010), travel frequency/capacity (Denver International Airport, 2009; Freathy & O'Connell, 2012; Geuens et al., 2004; Lin & Chen, 2013; Lu, 2014; Ma, 2013) and previous purchase behaviour inside an airport (Geuens et al., 2004; Perng et al., 2010). However, there is a disagreement regarding travel capacity, in that some authors found it not to be important, whereas the rest of the authors concluded it was actually relevant, which makes more intuitive sense as well. The utility of signage is not unanimous between the authors either, in that Denver

International Airport (2009) found it to be more relevant than Van Oel & Van den Berkhof (2013). Finally, Wesley et al. (2006) highlighted the pre-planned expenditure as well as a very relevant factor on the customers' satisfaction and activities in a commercial space.

Researchers have also defined different sets of motivation factors throughout the years and put them into different categories. El Hedhli et al. (2013), for example, found six determining factors for customers' well-being - functionality, safety, leisure, atmospherics, self-identification and convenience; Lu (2014) also thought of six factors, which he named Enjoyment, Values, Environment, Famous, Services and Conveniences; Lin & Chen (2013), on the other hand, considered three different motivations, while Chung et al. (2013) distinguished three categories for motivation factors – a functional dimension, an experiential dimension and a rational one. Overall, it can be said that the authors have taken into account factors in common and have sometimes divided them into similar categories.

Several researchers have made a distinction between pre-planned and impulse shopping and focused on that difference (Crawford & Melewar, 2003; Freathy & O'Connell, 2012; Lin & Chen, 2013; Lu, 2014). Perng et al. (2010) also concluded, in their work, that passengers who spent money at cafés generally did so on impulse. It would be interesting, from an economic point of view, to maximise impulse shopping in airports. Crawford & Melewar's (2003) research might work as an important reference for airport retailing, stating that stress, anxiety and normative traits should always be reduced and browsing induction should always exist, as well as pure impulse. Important measures in order to achieve such objectives are exemplified in Table 3. According to Lu (2014), Enjoyment, Values and Environment are the three most relevant variables for pre-planned shopping, while Values, Environment and Famous the top three for more impulsive shopping.

Finally, in terms of architectural layout, it is important to retain the idea from Freathy & O'Connell (2012) and keep in mind that people, in general, do not like to go back on their steps at any point and would rather avoid big detours from their way to the departure gate, which is their final destination. The personal preferences revealed in the study made at Denver International Airport (2009) are interesting as well. Van Oel & Van den Berkhof (2013) concluded that other architectural factors are also important, since pedestrians were found to prefer warm lighting, wide spaces, some greenery, some white presence, curvilinear roofs, curved hallways and no decoration related to the distinctiveness of the place they are visiting (in that case, Holland). However, Chebat et al. (2014) concluded that visitors' spending attitude is not too influenced by mall renovation, which was the main focus of their research.

Finally, the work developed by Ma (2013) will be important in the aim of this dissertation in terms of numerical input data and concepts, since one of his main objectives was to model pedestrian behaviour in a realistic way.

2.4. Individual choice modelling theories

Regarding the way each individual chooses his or her next task, several discrete choice models have been developed throughout the years: multinomial logit (MNL), mixed logit, generalised extreme value (GEV) models, Probit models... They can all be applied when the objective is to model a finite set of choices and get to know the probabilities for each individual to choose each alternative. This set of alternatives needs to be finite and exhaustive (cover all possible alternatives), while the alternatives inside it need to be mutually exclusive (no more than one option can be chosen by an individual at a time).

These are all based on the Utility Theory, which defines a function that can be calculated for each and every individual n, associating it to the actual "interest" for him or her to choose alternative i among the set. The general formula is written in Equation (1) (Train, 2003):

$$U_{ni} = V_{ni} + \varepsilon_{ni} \tag{1}$$

Where V is often called the representative (deterministic) part of the utility and ε is the random/unobserved part, which cannot be easily determined. Differences between the previously mentioned types of models mostly concern the way this unobserved part is modelled.

The deterministic part of the utility, in turn, is defined by the general equation (2) (Train, 2003):

$$V_{ni} = \beta' \times x_{ni} + k_i \tag{2}$$

Where x_{ni} is a vector 1 x N containing all the N explanatory variables and β' is a vector N x 1 with the values of the weights for each explanatory variable. For instance, if the utility of an alternative for a given person depends on time and money, then these are the two explanatory variables. Each one of them has a different weight in the final utility, so the values of their respective β' will necessarily be different. k_i is called an alternative specific constant, which is a constant that is specific to alternative i. In the end, the probability of choosing one activity over the others is the probability for that activity to have a higher utility than any of the others, as shown in Equation (3) (Train, 2003):

$$P_{ni} = P(U_{ni} > U_{nj}\ \forall j \neq i) \tag{3}$$

The simplest one amongst the previously mentioned models, and the most widely used, is the multinomial logit (MNL). This type of model dictates that, given a finite set of alternatives, the closed-form for an individual n to choose alternative i is given by Equation (4) (Train, 2003):

$$P_{ni} = \frac{e^{V_{ni}}}{\sum_j e^{V_{nj}}} \tag{4}$$

2.5. Conclusions and discussion

Several scientific documents, which were considered to be somehow useful in the aim of this dissertation, have now been mentioned in this chapter. The first impression is that, overall, research has been made from several different points of view, but no studies were found regarding exclusively the terminal layout and shop positioning. Indeed, some authors have mentioned such factors but none of them has got too much into detail about them. Most of the above referred researchers focused mainly on shopper typologies or product preferences, which are not the most relevant aspects for this dissertation, even though they might be helpful.

The subchapter 2.4 was also useful to understand that another important document for the development of the present work is the book written by Train (2003) – the chapter about the multinomial logit model and its formulas, in particular, will be interesting to keep in mind while creating a model about the individual choice process among a finite set of alternatives.

Some further general remarks can be made about these bibliographic references and results. Firstly, results should always be analysed through a logical scope and should preferably make some intuitive sense to the reader. However, in some of the previously referred works, some conclusions seem to be slightly misleading at first sight. For example, it does not make intuitive sense that variables such as age, education and income are not really relevant in CDM styles (Wesley et al., 2006). Similarly, the same authors concluded that the activities one undertakes inside a mall do not affect global satisfaction, which seems to be controversial: the fact that browsing around without purchasing anything leads to the same satisfaction level as actually shopping does not seem to make entire intuitive sense. Also, the nested model presented by Liu et al. (2014) does not appear to make total sense, since the same variable ("waiting before boarding") appears in two different nests (the "more frequent" and the "less frequent" ones). Even though such results are conceivable to happen when using a nested logit model, they should be considered with precaution.

Secondly, it is important to always wonder about the relevance of certain parameters in the aim of this dissertation. Focusing once again on a more individual level within the articles: one of the main objectives for El Hedhli et al. (2013) was to increase loyalty and word of mouth among the customers, but how important would that be for an airport? Even though more general articles about commercial spaces were here cited, it is very important not to forget an important and distinctive difference to airport shopping: while people inside a shopping centre decide to go inside a commercial space wittingly (either because they want to purchase something or for any other reason), air passengers are "forced" (in a way) to find themselves in the middle of it because it is part of the general, mandatory process to fly. Therefore, concepts as loyalty and word of mouth are not that relevant in airport management. The article itself is still interesting though, since it tries to maximise people's satisfaction while inside a commercial space.

Finally, another important remark concerns the generalisation of the results. In fact, most results should be carefully considered before applying them to other situations. It is noteworthy that, within the articles cited in this chapter, four were written by Taiwanese authors and focused on Taoyuan International Airport. Recalling one of the final comments made by Van Oel & Van den Berkhof (2013), it is important to always keep cultural differences in mind. As such, it would be expectable to obtain different results if the same studies were applied to Lisbon International Airport. In other words, as several authors have highlighted before, results cannot always be generalised outside of their original scopes. Yuksel (2004), for instance, was not mentioned in this chapter, but highlighted the fact that international visitors might have very different perceptions of the products when compared to local visitors (from Turkey, in that case). Actually, not only cultural differences were observed in his study, which focused on a shopping centre – the evaluation of the prices was also very different, which could be explained by the higher or lower price sensibility depending on the nationalities, as well as by the fact that local people might have other references in mind for comparison, which foreign visitors do not have.

Chapter 3. Modelling tools and approaches for airports, commercial spaces and pedestrians in general

3.1. Introduction

In this second chapter of the State of the Art, bibliography will be analysed in order to gather the most important studies and conclusions of the past decades regarding pedestrian modelling problems. For each one of the mentioned models, the tools that have been used throughout the years for creating and running them are detailed. The same kind of procedure and textual structure as the ones applied in the previous chapter will be used here.

Several research has been made throughout the last decades. Tošić (1992) decided to collect, back then, all the research that had been made in passenger terminal modelling before that year. He concluded that several efforts had already been made by that time, in several different aspects of pedestrian processes inside an airport, from the passenger demand to the different types of counters, waiting areas, baggage processes and terminal dimensioning (both in terms of number of gates and configuration). In the last part of the article, some general models that had been created for airport terminal buildings as a whole are mentioned as well. In the end, the author concludes that a lot of research had already been done by then but that there should be even more, and better organized. Odoni & de Neufville (1992), by the same time, also made a review about passenger terminal design. They concluded that more professional effort should be put into the uncertainty of the system rather than trying to forecast a single future value, which was considered to have a very low cost-effectiveness relationship. They also concluded that, until 1992, the use of simulation models had not led, in general, to successful results. Since that time, fortunately, considerable progress has taken place in this field and computer simulation has been being used for several decades now.

3.2. Types of modelling and State of the Art regarding previously applied tools/approaches

3.2.1. Types of modelling

The first important distinction that needs to be kept in mind when treating modelling problems is between simulation and analytical models. The latter type of modelling is purely based on mathematics: input and output data are directly connected through formulas and/or scripts (Grigoryev, 2015), allowing the user to obtain the mathematical solutions as long as these relationships are well established in the model. Typically, it is run in an Excel spreadsheet. It is not an adequate type of modelling for problems where uncertainty is important, where there are time and causal dependencies, where behaviour is non-linear or when variables influence each other in a non-intuitive way. Analytical solutions thus become difficult or impossible to determine when a dynamic system is being analysed (Grigoryev, 2015); simulation models, on the other hand, are always executable, which means that they need to be run by the user in order to produce results. They allow the user to choose an abstraction level and, depending on that, decide on how to build the model, which generally has a similar structure to the studied problem's. Because of that, added to the fact that simulation models are often associated to animation, they end up becoming more convincing (Grigoryev, 2015), namely for someone who does not know the model well. The temporal component is always present in a simulation model, and can be easily set according to the user's preferences. Because of these advantages, simulation processes have been used by several researchers in diverse fields of knowledge throughout the years.

Grigoryev (2015) defines, in general terms, the three types of simulation modelling that are recurrently distinguished by scientists. Each one of them is associated to a given set of assumptions and concepts that lead to a conceptual model of the studied system (Behdani, 2012):

- System dynamics are typically used to model highly abstract problems with nearly no details and end up being applied to strategic modelling on a macroscopic level (Grigoryev, 2015). Entities tend to be homogenised and have similar features, leading to a low adaptivity at an individual level – it is qualified as a system-orientated type of simulation (Behdani, 2012);
- Discrete-event modelling, on the other hand, is based on the idea that the system punctually evolves from one state to the other through well-defined events, considering that no change occurs in between. Time, at a smaller scale, is not too relevant (Grigoryev, 2015). Entities at a micro-level are heterogeneous but passive objects that follow a pre-determined fixed process by the user – that is why it is called a process-orientated simulation paradigm (Behdani, 2012). This type of approach approximates continuous processes in the real world with non-continuous events on the software. Entities, although passive, are capable of participating in the events and might have specific attributes that have an influence on them (Kalakou, 2012);
- Finally, there is agent-based modelling, which can cover either very detailed models or considerably more abstract ones, as well as everything in between. In practical terms, it can be applied to a broader set of problems than any of the two other types of modelling (Grigoryev, 2015), which is another interesting advantage, since it allows the model to be more flexible and adapt the level of detail (or, oppositely, abstraction) according to what suits the system the best. It is an individual-orientated type of simulation, where heterogeneous entities are active agents that are able to analyse the environment, interact with other agents and make their own decisions based on all that. The system structure is typically not fixed (Behdani, 2012).

3.2.2. Applied tools and approaches by previous authors

Some research has been made in order to model the airport as a whole. Kalakou (2012) wrote a dissertation in which the main objective was to reproduce passengers' behaviour from the moment they entered the airport building until they finished going through the security check, in order to detect underperforming areas inside Lisbon airport and define performance indicators. In order to do so, passengers were segmented into categories and then two different models – one discrete model and one pedestrian dynamics model – were tested in the simulation software, which the author decided to be AnyLogic. It is a simulation software based on an object-orientated language that can support systems dynamics, discrete-event and agent-based modelling, which are the three most common methodologies applied to simulation nowadays (AnyLogic, 2015) and were previously described. Pedestrian movements can be animated, making it easier to present the results. It has been applied in several different fields throughout the years, such as logistics, transportation, airports, shopping malls, markets and competition, healthcare, among others. In Kalakou's dissertation (2012), in particular, a specific tool called AnyLogic Pedestrian Library was used in order to better simulate flows of people inside a given physical space, with its own obstacles and pedestrians interfering inside. Figure 7 illustrates the check-in process in the modelling process for Lisbon airport made by Kalakou (2012).

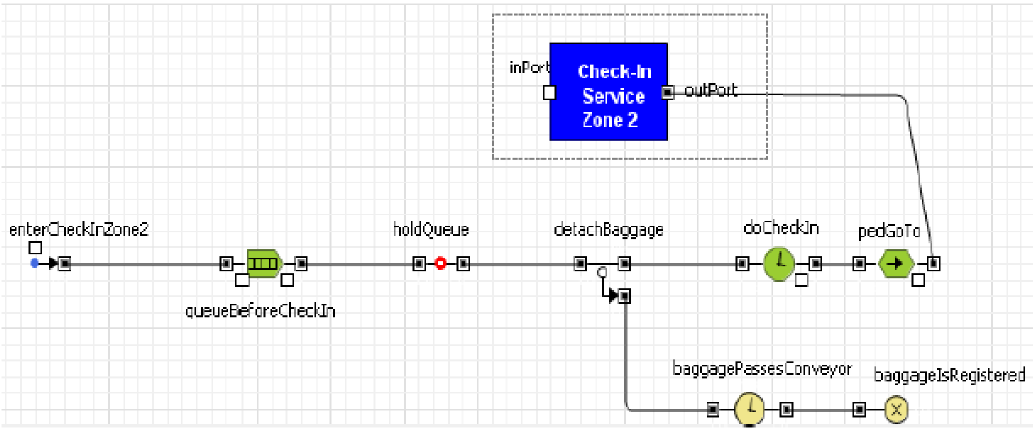


Figure 7. Simulation model for the check-in event process in Lisbon airport, as developed on AnyLogic software (Kalakou, 2012)

Figure 8, on the other hand, illustrates the visual representation that AnyLogic provided for the security control area and the pedestrian movements inside.

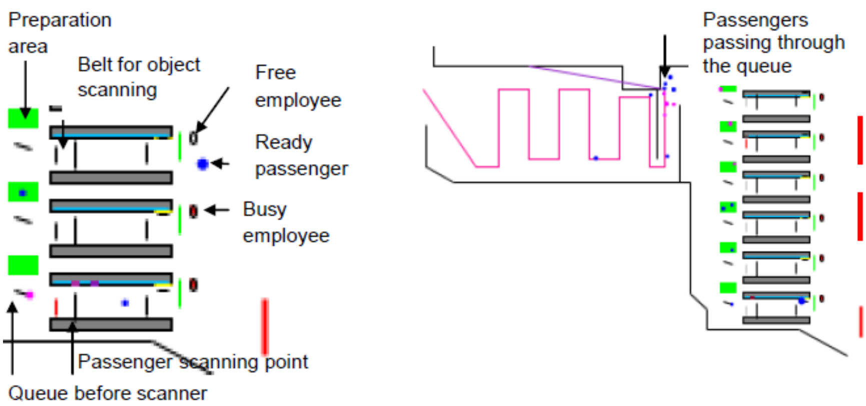


Figure 8. Simulation model for the security control area in Lisbon airport, as developed on AnyLogic software (Kalakou, 2012)

In her model, the software required several topics about the flights (flight ID number, time, destination, airline, operator, check-in zone, number of check-in counters…), the passengers (their flights, times of arrival to the airport, flying companies, whether check-in is made in or out of the airport, their baggage, their categories as passengers and their entire “schedules” once inside the airport) and the processes in the airport (check-in, security check and available services); most of this data, as well as the passenger arrival distributions, were obtained through ANA’s historical records. The travellers’ behaviour was studied through a survey conducted by the author and the speeds for the pedestrians were taken from IATA (International Air Transport Association)’s recommendations. In the end, the model was considered to be interesting and to give relevant results, since it highlighted some problems in the functioning of the airport. The software also proved to be effective and efficient, by allowing the user to test different scenarios and determine what improvements should be made. However, there is always room for improvement. The author wrote that the model could become more realistic if some parameters (regarding destinations and delays, for example) were also available and included in it. Also, the simulation results could be more complete if the simulation period and space was extended (both in time and further into the terminal steps).

As stated on AnyLogic official website (2015), other authors have used that software for airport modelling as well. Curcio et al. (2007), for example, used AnyLogic to simulate passenger flows and detect security problems in the airport of Lamezia Terme, in Calabria, Italy. Passenger flows, aircraft movements and baggage flows were all modelled using AnyLogic, and simulation results were then compared to real data in order to be validated. Passengers’ waiting time before reaching the gate area was used as a performance indicator of the system. The authors mention that several objects off AnyLogic Enterprise Library were important in the modelling process as to simulate queues, delays, paths, etc. In terms of input data, their model required all the information regarding both departing and arriving flights (such as the departure and arrival times, the origins and destinations, the flight reference numbers, the names of the airlines, the frequency of the connection…). All this information was put into an Excel file, which needed to be compatible with AnyLogic in terms of units. Each simulation run represented 130 days and was evaluated by a mean square pure error analysis. In the end, the passenger flows obtained by the model differed, on average, 4.64 % from reality, which revealed a satisfactory accuracy and led the authors to test the model under other scenarios, in which they changed the passenger arrival times at the airport, the available check-in points and the available security control lines. In the end, the developed model was considered to be valid, accurate and applicable to other similar airport terminals.

Cheng (2014) also chose AnyLogic as the software to apply to her model, which aimed to represent group dynamics inside an airport terminal through agent-based modelling. Several scenarios were tested (in terms of group composition) at different stages of the passengers’ journeys and some parameters were measured, such as dwell times, discretionary activity preference and the levels of service (LOS) at the processing areas. The model was then applied to a case study and the impact of group dynamics in an airport evacuation process was analysed. In the end, the agent-based simulation model was considered to be interesting and to lead to important conclusions, even though some other details could be taken into account as well, such as the existence of staff or alternative evacuation paths. Figure 9 illustrates one of the animations that were obtained in her work.

Figure 9. Landside of the departure terminal, as seen on AnyLogic software - 3D animation (Cheng, 2014)

Similarly, Ma (2013) chose AnyLogic as the most adequate software for his simulation work about pedestrian modelling in Australian airports as well. Two separate models were tested: one with one flight only, in order to test the model in a less complex situation, and one for an entire day, with several departing flights throughout the simulation period and the additional complexity that arose from that. The author highlighted the importance of considering the existence of discretionary activities, and one of the main objectives of his dissertation was to model passengers properly in terms of their activity scheduling. He tested and compared, for each model, two different scenarios – one where only processing activities were undertaken by pedestrians, and the other with discretionary activities included as well. His model was made through agent-based simulation. As previously mentioned, agents are autonomous decision-making units able to interact in order to achieve common goals. The author saw them as the best option to explore space-time dynamics and described them as being the most adequate to capture emergent phenomena and complex human behaviours. Grigoryev (2015, p. 20) defined it the following way: «You may not know how a system behaves, be able to identify its key variables and their dependencies, or recognize a process flow, but you may have insights into how the system's objects behave. If that's the case, you can start building your model by identifying the objects (agents) and defining their behaviours.

Afterwards, you may connect the agents you've created and allow them to interact or put them in an environment which has its own dynamics. The system's global behaviour emerges from many (tens, hundreds, thousands, millions) concurrent individual behaviours».

In his work, in order to model people's behaviour, Ma (2013) used Bayesian networks. These are based on Bayes' theorem and are used to model joint distributions of probabilistic properties. They might have a causal structure or not (in other words, the arrows' directions do not need to be meaningful – although a good network is generally formed by causal relationships between elements). Visually speaking, Bayesian networks are pretty intuitive and easy to understand. Influence diagrams were used as well, which are basically a way of expanding Bayesian networks through utility and decision nodes, through which utility values are evaluated and decisions are made at specific decision points in space. Bayesian networks were defined by the passengers' basic and advanced traits, while influence diagrams were defined by devised utility tables.

Two types of stress were considered as well in the model by Ma (2013): time and distance. Time stress is related to the fact that no passenger is willing to miss their flight, whilst distance stress accounts for the distance a passenger can walk before needing a rest. Uncertain dynamic facilities were also taken into account, in case some facilities (such as information desks) would not be available at some point, for some reason.

He took into account ten advanced traits, leading to ten different passenger categories, associated to ten different activities. At each decision point, probabilities were calculated for each individual through Bayes' theorem, thanks to the basic traits and some environmental factors, and one decision was made. It is noteworthy, however, that these ten categories are not exclusive, so a given passenger might belong to several categories at the same time. Figure 10 illustrates the overall logic of this behaviour modelling. Each passenger agent tries to choose the category C and the action A that maximise their expected profit, based on a prior state S.

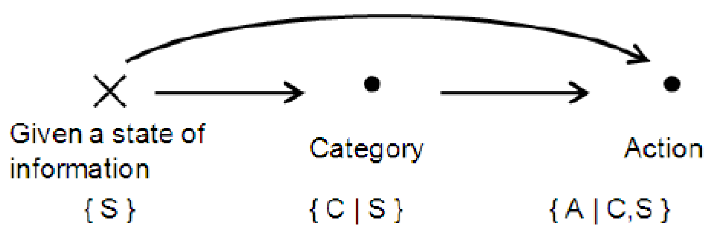


Figure 10. Overall behavioural logic of an individual (Ma, 2013)

Figure 11, on the other hand, illustrates the general logic of an influence diagram and interactions within its nodes, by Ma (2013). At each point, the activity with the highest utility is chosen by the agent; however, time stress is never put aside, since accomplishing the next checkpoint always becomes the activity with the highest priority when temporal stress sets in. Priorities in such cases are defined in specific tables created by the author (when there is time stress, priority for the action "proceed checkpoint" switches from 0 to + 100, while all of the others switch from positive to negative values).

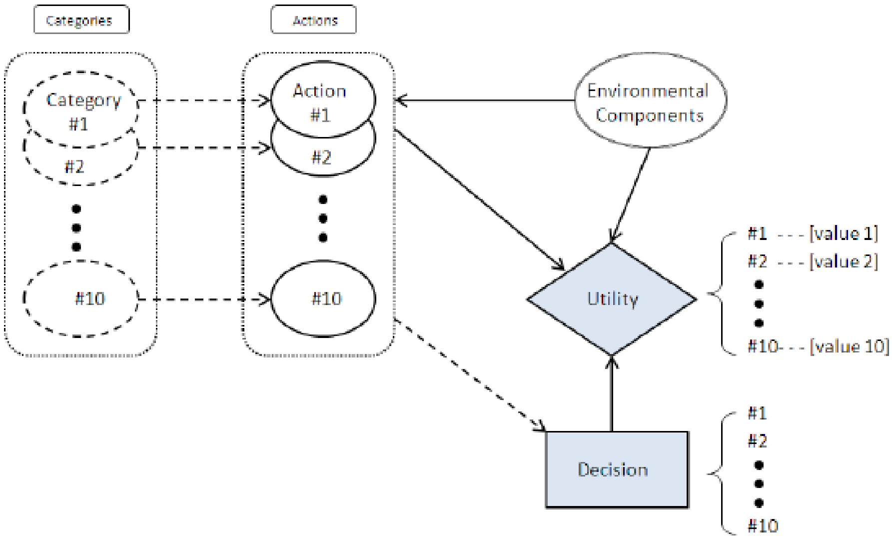


Figure 11. Generic influence diagram (Ma, 2013)

Each passenger was generated with random attributes regarding the basic traits (age, gender, frequency of travel, travel class and nationality) and the probabilities for the advanced traits were calculated through the Bayesian rules. The conditional probability tables he used can be found on his work. Processing activities had the highest priority and the others had varying degrees of priority, depending on special needs or urgent circumstances. Table 5 gathers the priority values for five different actions, depending on whether time stress was activated or not. After dedicating some time to a given activity which was on top of the priority list, the value of the associated advanced trait would immediately switch to “false” and the new probabilities would be calculated. In the end, the model proved to be working and to generate valid results, even though the author thinks that a more precise validation should be done.

Table 5. Priority values for five activities, with or without time stress (Ma, 2013)

Actions	Priority values (no time stress)	Priority values (time stress)
Shopping	+ 60	- 100
Emails, phone calls and social media	+ 40	- 100
Withdraw money	+ 30	- 100
Buy a drink or meal	+ 100	- 84 to - 50 (depending on how long the service lasts)
Proceed checkpoint	0	+ 100

More recently, Fonseca i Casas et al. (2014) also created a simulation model for an airport, aiming to assess the design and construction of a new terminal and allow a dynamic management in the future, using Barcelona International Airport as their case study. However, the tools that were used were not the same as previously – the model was defined in Specification and Description Language (SDL) and the software that was used was Witness, in order to allow a visual interpretation of the simulation runs and results. The airport geometry was the existing one in the case study and all the data regarding passengers and aircrafts came from historical records from the airport. All of the entities and areas were defined and put into categories. In the end of the simulation, all the main flows and processes of the airport were obtained. The developed model had an actual utility both for the design and management of Barcelona International Airport and can be applied to other airports with similar configurations. Figure 12 illustrates the type of vision one can have while running a simulation in Witness software.

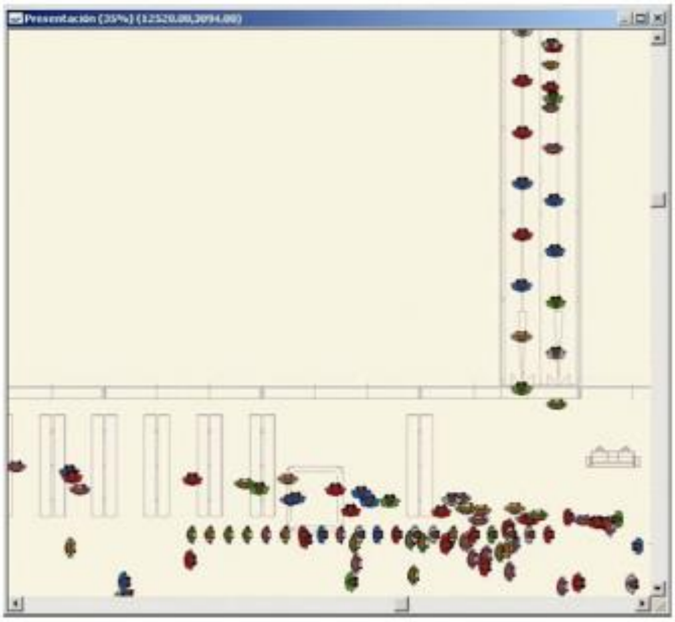


Figure 12. Boarding process simulation at Barcelona International Airport, as developed on Witness software (Fonseca i Casas et al., 2014)

Andreatta et al. (2007), on the other hand, used SLAM II language in order to model the Athens International Airport under different scenarios. SLAM II is a Fortran-based simulation language that combines network, discrete-event and continuous modelling concepts all in the same language. Their model gives back to the user graphs regarding performances of all the passenger facilities, as functions of time. Three different scenarios were tested: a baseline scenario, a hubbing scenario and the Olympic Games scenario (since Athens hosted the Olympic Games in 2004). In order to run the program, the number of departing flights during the studied period of time, their times of departure, the types of aircrafts and flights, the number of passengers, the passenger arrival profiles, as well as the number of counters and their service time needed to be used as input data. The developed model managed to point out the main problems in the terminal facilities, and results were considered to be reasonable both by the airport experts and the companies.

Ballis (2002) wanted to develop a simulation model that could be used to make a proper design for two Greek airports (Kavala and Alexandroupolis), which were said to have some particularities in their passenger flows at the time. A model that had already been developed at the National Technical

University of Athens (NTUA) in 2001 was adapted to these airports and applied, even though it is not technically specified and described in the article. The model required the arrival patterns of the passengers, the service rates and the service point allocation rules. It ran the simulation and gave quantitative information to the user regarding critical areas such as the check-in area, the passport control or the departure lounges, which enabled the measurement of the Level of Service (LOS). The model proved to be very useful, but the authors highlighted the fact that an Engineer's opinion on the results should always exist before totally trusting the simulation results.

Roanes-Lozano et al. (2004), later, decided to focus on the passengers movements inside the departures terminal of the airport in Malaga, Spain. Their simulation model was implemented in the computer algebra system Maple 8 and calculations were repeated for every minute in the simulation. Passengers were treated individually in the simulation and were temporarily guided by one individual variable that represented the time left until the next step. In terms of input data, the model needed to know the list of companies, the average check-in times, passenger delays, and the areas comprising check-in counters, control points and gates needed to be specified in detail. A full day simulation took approximately 5-6 minutes to run at the time and the output results were essentially graphs representing variables such as queue length or number of passengers as functions of simulation time. Maple 8 was used because it was considered to be a simple programming language that would fulfil all the authors' needs. In fact, they were more interested in running time-efficient simulations with plotting, rather than longer and more visual simulations. The authors even mentioned that software like Witness were neither necessary nor possible to apply in the aim of their project. The final result was considered to be very useful both for the design and the exploitation phases, since it allowed the users to compare different solutions. Some improvements, regarding the interaction with the package, have however been made by the authors since the time they published this article.

Sourd et al. (2011), differently, did not focus on an airport problem – their objective was to model pedestrian flows during dwell period, but within the railway context. In fact, the idea was to create a tool which would help modelling the design of passenger coaches. Simultaneously but independently, they modelled four types of behaviour: passengers who stayed in the train, passengers who wanted to get off the train, passengers who wanted to get in the train and passengers who stood on the platform. The simulator they used was developed by Goleam and required some programming commands as well in order to model human behaviours. It allowed both 2D or 3D animation runs of the system and used as input data the population size, distribution and characteristics such as age, comfort speed, mobility capacities and diameter of the dots that represent individuals. Figure 13 illustrates the type of animation that can be obtained with this model. In order to model pedestrian behaviour, a tool called Goleam Activity was used.

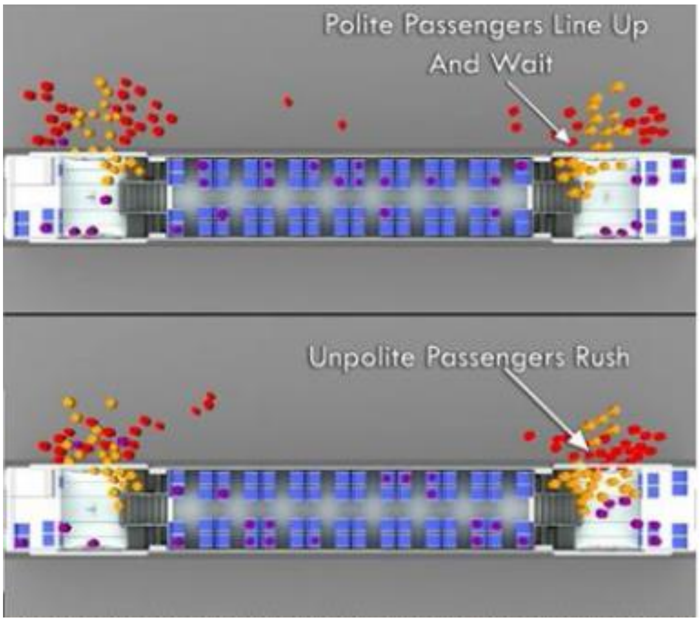


Figure 13. Pedestrian simulation for passengers wanting to enter a train coach (Sourd et al., 2011)

An important aspect that needs to be taken into account is the specific parameters and concepts that have been used to study this type of problems throughout the years. Authors have been applying different passenger-related indicators in order to optimise airport terminals from a pedestrian point of view. A common concept that can be found in the bibliography is wayfinding, which is illustrative of the passengers' orientation in space. Fewings' (2001), cited in Tam (2011, p. 74), defines it as «the process of finding one's way in the geographical or built environment; that is, being able to identify one's present location and knowing how to get to the required location». In his article, Tam (2011) attempted to create a model that would help designing the signage system in an enclosed environment (either an airport, a train station or a shopping centre – not necessarily an airport) and tested it in an airport environment. In order to do so, he focused on one important parameter called the Visibility Index (VI), which is the ratio of the number of available sight lines at a given point and moment, and the total number of sight lines that could potentially exist there. A binary linear program, that tried to maximize the VI in order to have a good allocation for the directional signs, was proposed in the article. The node network in the airport was modelled as a binary quadratic matrix that included all the connectivity factors. Facilities were divided into primary or secondary and each one of them had a weight value associated to it. Both components were used in the formula for the VI calculation, which was meant to be maximised so that passengers' wayfinding capacity would become higher as well. Data were obtained through the site map of the terminal building, where all the nodes and arcs, as well as all the necessary distances and parameters, were measured/determined. The formulation made in this article was considered to be interesting, but was restricted to one level of signage only. The author suggested that further investigation should be made in that the model should be extended to multi-level signage indication (even though the number of variables would increase considerably, as well as computing time).

Other authors focused on the same type of concepts and modelling. Churchill et al. (2008), however, proposed a new model for the Visibility Index and compared it to the one most usually used at the time. Survey methods made in the Calgary International Airport were used in order to define the new VI, which showed the importance of creating a sound signage program. The differences between the old and the new definitions of VI were said to be largely due to signage, in general.

Beyond maximising the VI, Tam (2011) had the walking distance minimisation as a main objective as well. For each one of the links in the physical system, the associated walking distance was defined and the signage system was designed in order to make it as short as possible, since that was considered to be the most convenient for pedestrians inside the terminal.

Such concept and objective had already been previously used by other authors as well. Bandara & Wirasinghe (1992) used a different methodology while having the same objective overall. They investigated the standard geometries for airport terminals at the time and tried to define the optimal configuration for a given number of gates, as well as to develop a procedure to determine the best geometry aiming to minimise the walking distance for passengers. In their work, Equation (5) was defined for the walking distance:

$$\bar{W} = (1-P)\overline{W_A} + P(1-Q)\overline{W_N} + PQ\overline{W_H} \tag{5}$$

Being:

- $\bar{W}$ the mean walking distance for all passengers;
- $\overline{W_A}$ the mean walking distance for arriving and departing passengers;
- $\overline{W_N}$ the mean walking distance for non-hub transfers;
- $\overline{W_H}$ the mean walking distance for hub transfers;
- P the fraction of transfers with respect to the total number of passengers;
- Q the fraction of hub transfers with respect to the total number of transfers.

The mean walking distance for hub transfers, in turn, was defined by the authors as being a function of the mean walking distance for hub transfers whose departure gate was pre-determined to be in the arrival pier, and the walking distance for hub transfers who are equally likely to depart from any gate in the terminal. The terminal’s geometric characteristics were then analysed and, for each one of them, formulas were determined for the previously mentioned walking distances. A simulation was then run for each geometry in order to compare them. In each one of these simulation runs, the simulation software generated the number of passengers based on the number of gates and distributed them uniformly. The best geometry for a given terminal configuration was found to depend on the number of gates, the distances defined by the geometry of the terminal and the selection criteria. The presented method was considered to be usable for studies regarding the expansion of new terminals. However, it did not take into account elements such as construction, maintenance, baggage movements, automated movement of people and taxiing of aircrafts. Once again, the software that was used by the researchers is not mentioned in the article.

Smits et al. (2013), in turn, used a different concept to model passengers in the transit hall of an airport, which they validated by applying it to Rotterdam Airport: they used time-augmented Petri nets, which are a mathematical modelling language for distributed systems. There, an extension of Petri nets (called generalized stochastic Petri nets) was used. As mentioned in the previous chapter, the authors defined and studied four particular actions: going to the toilet, standing in the queue for the check-in desk, actually checking in and leaning against a pillar. The input data for the model was basically the geometry of the Rotterdam airport and the relative time of the pedestrians (the extra time they had left before reaching their temporal deadline, which was used for other types of behaviour, namely shopping). However, and although the model was considered by the authors to be realistic, they also wrote that further research should be made in order to study more types of behaviours passengers might adopt when they have too much free time in their hands. In terms of software, the Mason multiagent simulation toolkit was used, in order to create a 2-dimensional environment for pedestrians. According to the developers' official website (Mason, 2015), this software is a fast discrete-event multiagent simulation library core in Java, and is more adequate to "light", not too time-consuming, simulations.

Finally, Vizzari et al. (2014) developed an agent-based simulation model aiming to study how pedestrian group worked on the system and tested it on a simple real-life example. A GA-Ped (Group Aware Pedestrian) model was created based on several physical experiments, where the group configurations and densities varied and movements were recorded on video as opponent passenger groups made their way through a 2.5 x 10 meter corridor. Such model was based on cellular automata (CA) approaches for pedestrian simulation, which means that the environment spaces were divided into regular grids of cells. In this case, each cell had 50 x 50 cm and constituted a part of the previously mentioned corridor. An example of the visual results that could be obtained with this method is illustrated in Figure 14, where the formation of a queue is visible by the black entities, who try to make their way past the grey entities, which are perceived as obstacles.

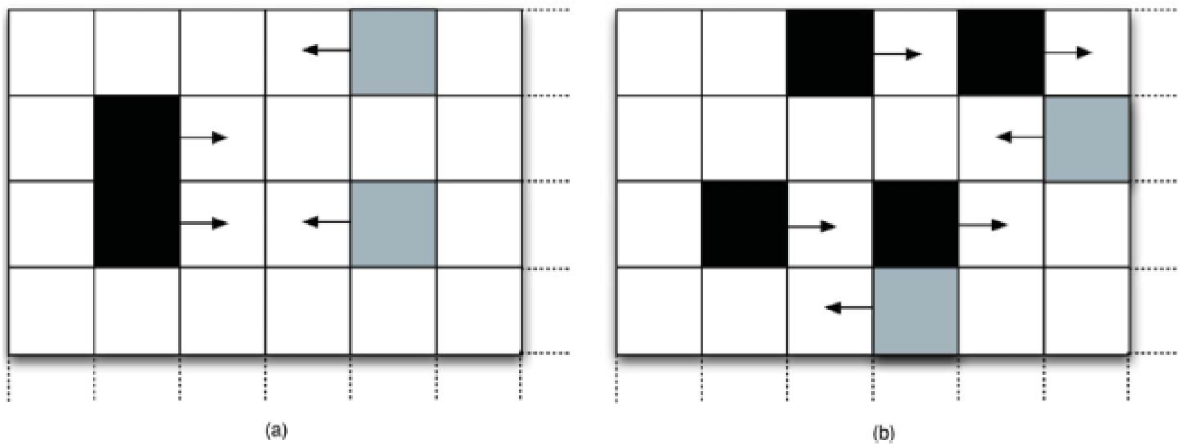


Figure 14. Queue formation by the black entities from (a) to (b) in a GA-Ped model (Vizzari et al., 2014)

After running the simulation model, graphs could be obtained where the specific passenger flows (number of pedestrians/ms) were shown as functions of pedestrian densities (number of pedestrians/m^2). This model was validated and applied to a real, newly constructed railway station in Makkah (Saudi Arabia), where very high levels of flows were originated by the yearly pilgrimage to Mecca. Visual results as the ones shown in Figure 15 were obtained. The two figures on the left illustrate two groups walking simultaneously towards the ramp, while the two of the right illustrate three of those groups.

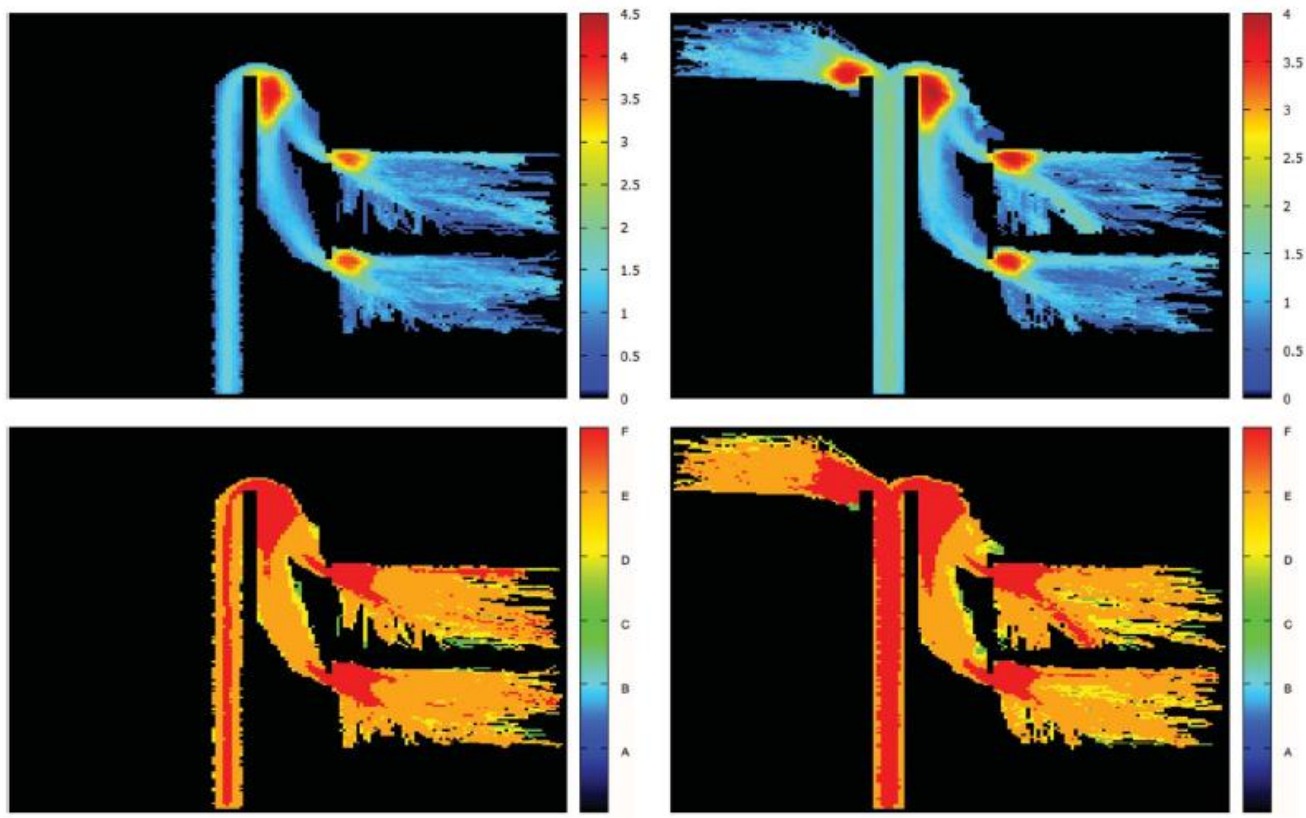


Figure 15. Cumulative mean density (upper figures) and levels of service (lower figures) in the modelling of a railway station in Makkah (Vizzari et al., 2014)

The higher the number of people or groups, the higher the density and the lower the LOS. However, the model does not consider a different walking speed on the ramp, while in reality the speed should be lower there (here, the speed is always considered to be 1.5 meters/second). Simulation results were close to observations on the field, but the model could still be improved if some factors such as group cohesion or hierarchy were also represented and preserved in the results. Also, more live experiments should have been made in the calibration phase in order to have a more accurate model. No software tool is specified in the article, though.

3.3. Chapter conclusions

The whole bibliography for the State of the Art of this dissertation has now been fully covered in these two chapters.

Overall, it can be said that a lot of research has been done regarding airport modelling in general, as well as pedestrian modelling in contexts other than the air transport ones. However, no researcher has focused on the retail space of an airport in the same way as this dissertation is aiming to. In fact, in terms of space and passenger modelling, none of the cited authors has dedicated their research to the layout of the commercial space or the way different shop types are distributed inside the general layout.

However, some useful information can always be extracted from what has been previously done. In the aim of this chapter, several different methodologies and tools have been mentioned and the type of results each one has originated is well observable. Such analysis is important when a software needs to be chosen for the simulation of a new problem. Nevertheless, it is noteworthy that, in some scientific documents, there is no tool or software specification. Indeed, some authors preferred to present a way of thinking and structuring the problems, rather than recommend a given tool or software.

Table 6 summarizes the software tools that were used by most of the authors cited above.

Table 6. Modelling tools used by the different authors

Authors	Modelling tools/approaches
Andreatta et al. (2007)	SLAM (Simple Landside Aggregate Model) II simulation language
Ballis (2002)	Model developed by NTUA LOS measure in some critical areas
Bandara & Wirasinghe (1992)	Walking distance minimisation (no software specification, though)
Cheng (2014)	AnyLogic
Curcio et al. (2007)	AnyLogic, AnyLogic Enterprise Library
Fonseca i Casas et al. (2014)	Specification and Description Language (SDL), Witness software
Kalakou (2012)	AnyLogic, AnyLogic Pedestrian Library
Ma (2013)	AnyLogic, Bayesian networks, influence diagrams
Roanes-Lozano et al. (2004)	Computer algebra system Maple 8
Smits et al. (2013)	Time-augmented Petri nets, Mason multiagent simulation toolkit
Sourd et al. (2011)	Goleam, Goleam Activity
Tam (2011)	VI maximisation / walking distance minimisation
Vizzari et al. (2014)	A GA-Ped (Group Aware Pedestrian) model based on cellular automata (CA) approaches for pedestrian simulation

It is possible to notice that some tools and concepts have been used more frequently than others throughout the years, showing that authors are tendentiously in tune with each other and the software choice often depends on what the desired output results are (whether the authors want to obtain graphs, animations, numbers, or something else).

Finally, this chapter also proved that it is important to keep in mind the idea that walking distance should be minimised overall (Bandara & Wirasinghe, 1992; Tam, 2011), which complies with Freathy & O'Connell's (2012) conclusion that big detours should be avoided since pedestrians do not enjoy walking too much, showing once again coherence between authors.

Chapter 4. Presentation of the case study – Lisbon airport and Terminal 2

4.1. Introduction

Being the case study a specific terminal of Lisbon airport, a general description of the airport, and then the terminal, will be done in this chapter. Some statistics from the last few years will be given, as well as the most recent information about the passenger-restricted area of the terminal regarding its geometry and available commercial options.

Most of the data are taken from ANA's official website (2015), from a section called RouteLab, where all the available statistics are. Also, a physical visit was done to the terminal on the 24th of June 2015 and some general ideas about the terminal layout were outlined. A phone call was made and an email was sent in order to try to obtain some specific figures that were not available on the website, but answer was given that the only available information was the one online. Therefore, in some cases, the presented numbers are not from 2014, but from previous years instead.

4.2. Overall description of the airport and traffic statistics

The three main airports in the mainland of Portugal that operate civilian flights are located in the cities of Lisbon, Oporto and Faro.

The one in Lisbon, studied in this research, is the biggest one in the country. People can easily reach it through several means of transport (both public and private), since it is closely located to the city centre. Connection between the two can be easily established by metro, bus, taxi or private car.

Air traffic to and from Lisbon airport has been increasing in the last few years: nearly 152.4 thousand movements took place there in 2014 (which is 7.0 % more than in the previous year). In terms of passenger flows, it served nearly 18.2 million passengers in 2014, which accounts for a 13.3 % increase when compared to the same parameter in 2013 (ANA, 2014). Figure 16 illustrates, graphically, the evolution of the annual air traffic since 2006. A noticeable increasing trend can be observed in the last years, since 2009.

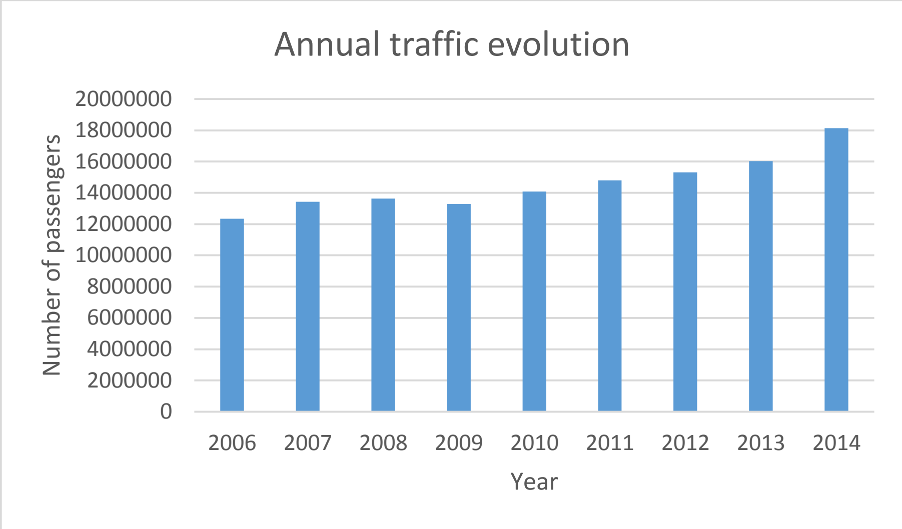


Figure 16. Annual traffic evolution for Lisbon airport between 2006 and 2014 (numbers from ANA, 2015b)

Regarding the monthly distribution and the average passenger distribution throughout the day, no data were found for 2014, so Figure 17 and Figure 18 were taken from the monthly report of December 2013 (ANA, 2015c) in order to illustrate these numbers. Figure 17 shows that July, August and September were the busiest months of the years of 2012 and 2013, as opposed to January and February, when the measured traffic was the lowest; Figure 18, on the other hand, shows that the periods from 8h00 to 9h00 and from 13h00 to 14h00 were, on average, the busiest hours of the day, while the period from 1h00 to 5h00 was the quietest.

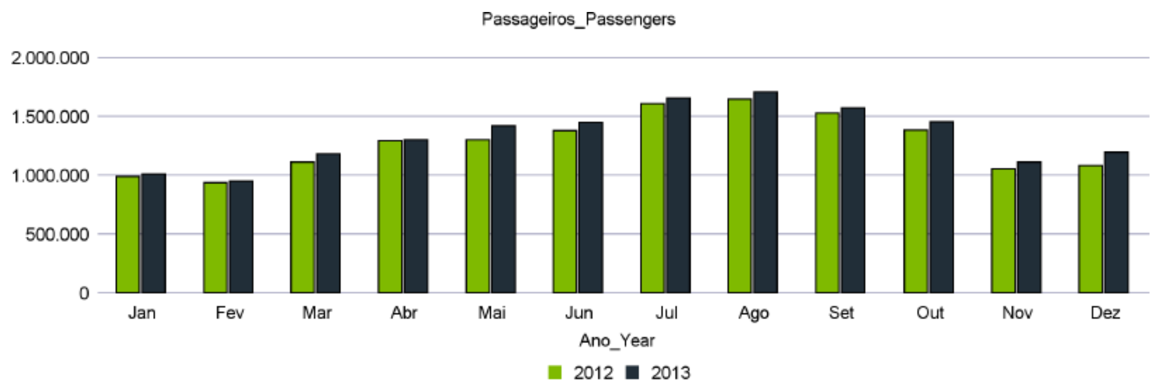


Figure 17. Monthly traffic in Lisbon airport in 2012 and 2013 (ANA, 2015b)

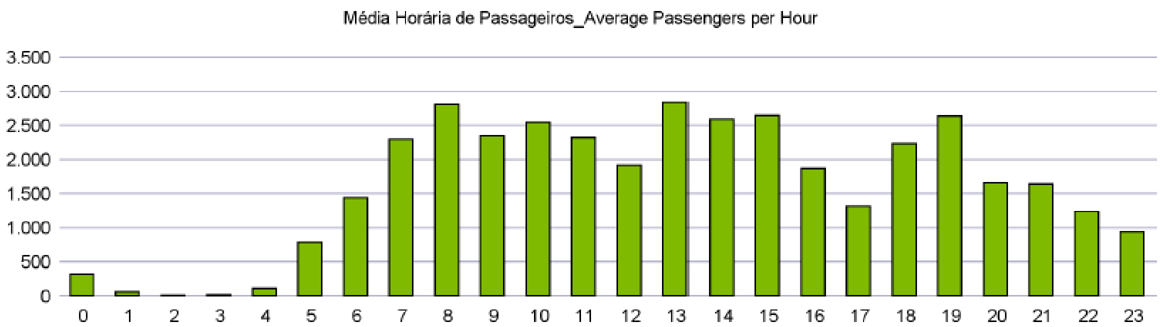


Figure 18. Average traffic distribution throughout the day in 2013 for Lisbon airport (ANA, 2015c)

In terms of destination categories (Schengen, non-Schengen or International) among the commercial traffic in the airport, the ones from 2013 are represented in Figure 19. Commercial movements represent 99.9 % of the total number of movements (the remaining 0.1 % regards general aviation). Figure 20, on the other hand, shows that the local company TAP was by far the most used in 2013, followed by easyJet and then the others.

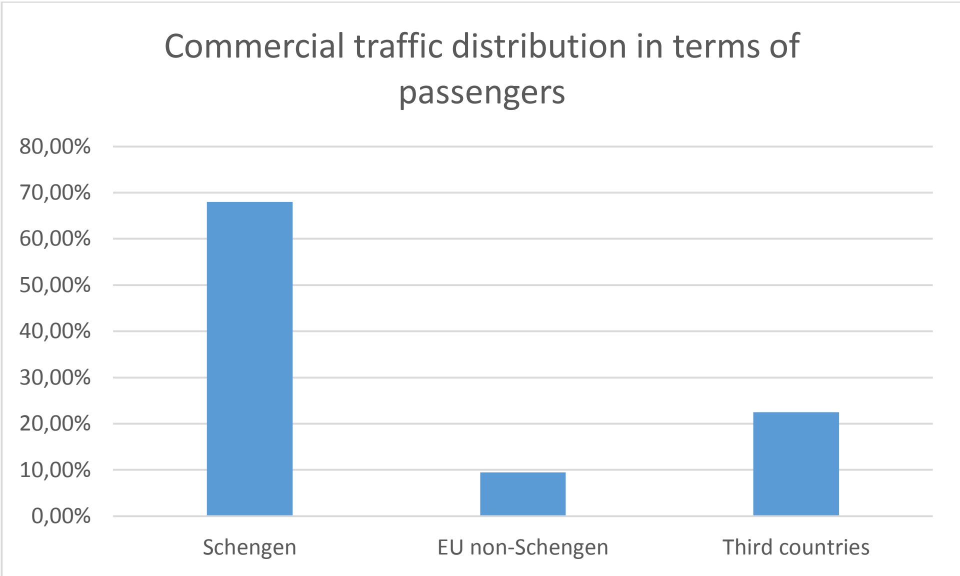


Figure 19. Commercial traffic distribution for Lisbon airport in 2013 (numbers from ANA, 2015b)

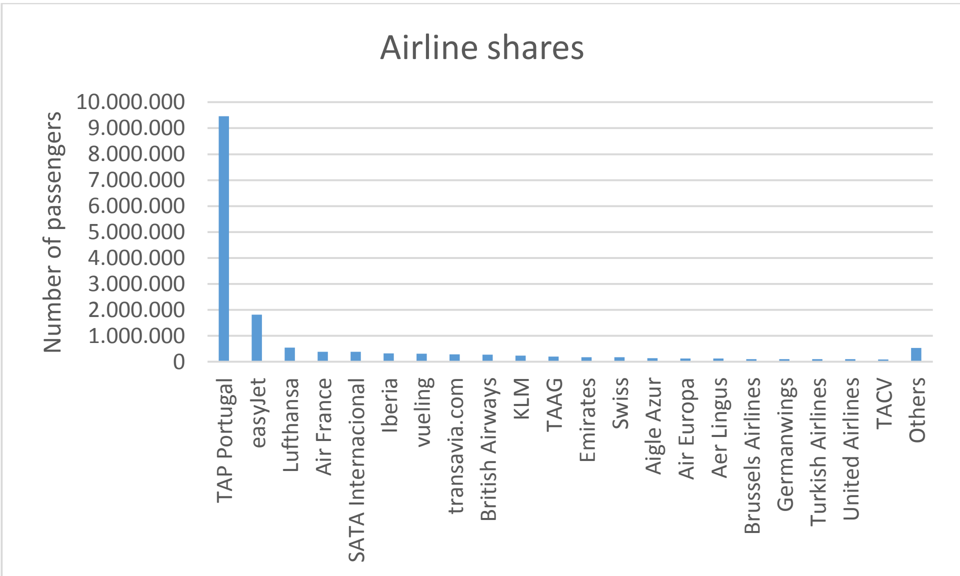


Figure 20. Airline shares for Lisbon airport in 2013 (numbers from ANA, 2015b)

The airport has one military terminal (Figo Maduro Airport) and two civil terminals (known as T1 and T2) – the work on this dissertation will be focused on T2, which is described in the following section.

4.3. Terminal 2 (T2) characteristics

This terminal is more recent than T1: built and inaugurated in 2007, its dimensions are smaller and passenger flows are less significant. In fact, T2 is a departure-only terminal – all the arrivals to the airport take place inside T1. Also, it does not have any parking lot (the only one is the airport is part of T1). The two terminals are however connected by road and there is a shuttle service between the two that passengers can take for free. The main difference between the two is that the companies that operate inside T2 have different needs from the other companies, since the flights that are operated there do not require a specific infrastructure for the transfer passengers' processing (ANA, 2015b). In practical terms, only low-cost companies operate there: EasyJet, Norwegian, Ryanair, Transavia and Wizz Air.

Figure 21 illustrates the general layout of the terminal 2. It has two main entrances, one public area and one restricted area. The public area includes all the check-in facilities, as well as a restroom, a scale (for people to weight their luggage, if needed) and an information desk; the restricted area, which can only be accessed by the passengers after going through the security check process, includes several discretionary activities and, obviously, the boarding gates and a main waiting area. Passengers who are travelling to a non-Schengen destination need to go through Immigration processes (SEF) as well and have a dedicated area for that. The terminal has eight boarding gates that are usually active and three that can be open if necessary.

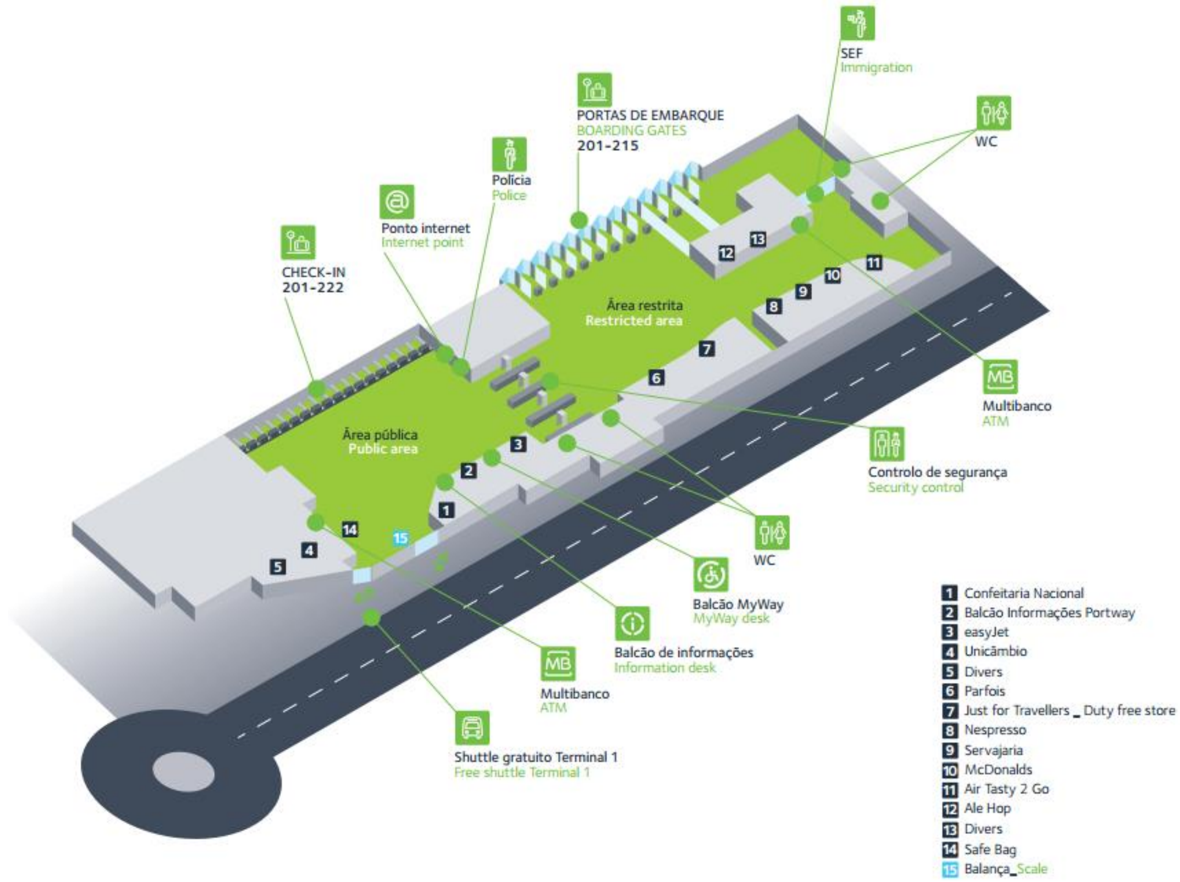


Figure 21. Drawing of the terminal 2 (ANA, 2015b)

This research will focus on the restricted area of the terminal only. There, several different facilities for discretionary activities can be spotted: two restrooms, four different shops (Ale-Hop, Parfois, Divers and the big duty-free store called Just for Travellers), four different options to eat or drink (Nespresso, Servejaria, McDonald's and Air Tasty 2 Go) and one cash machine.

This terminal's geometry is particular in terms of the boarding process: passengers wait for gate control in the main waiting area, and then wait for the doors to open in specific waiting areas for each gate. When the gate control process is over, those doors open and passengers start actually boarding the aircraft.

4.4. Passenger characteristics

In the work developed by Turismo de Lisboa (2014), the general profile of the low-cost passenger is drawn for Lisbon airport. Keeping in mind that only low-cost companies work inside T2, it becomes a very interesting document in the aim of this dissertation.

Two different seasons were analysed, in two separate documents: one winter season (between November and March) and one summer season (from May to October). Given the fact that, as stated previously, summer is the busiest time of the year for this airport, it is chosen to focus on the analysis for the summer period of 2014. Such data were collected through several interviews that took place in the check-in areas. Although the study took into account some routes and companies others than the four aforementioned ones, it is assumed that results are just as valid. In fact, this study did not consider Wizz Air, whereas other companies, such as Aer Lingus, Air Canada Rouge, GermanWings and Vueling, were taken into account.

According to that study:

- 71.4 % of the interviewed people were male;
- 13.3 % of the respondents were travelling for business purposes. It is thus assumed that 13.3 % of the people travel in business class (even though Transavia does not have such a category);
- The age distribution is illustrated in Figure 22 – only 3.6 % of the interviewed people was above 65 years old; the prevailing age group was 45-54 years old, accounting for 30.6 % of the interviewed population;
- The distribution for the annual frequency of travel is represented in Figure 23 – it is strongly dependent on the passenger's purpose for travelling (whether it is leisure or business). The values that originated the graph can be checked in Appendix 1;
- Table 7 indicates the different percentages for the reasons why the interviewed people were at Lisbon airport at that moment. The total sum of the percentages is above 100 %, since some people were travelling for more than one single reason. In fact, the total sum equals 118.1 %.

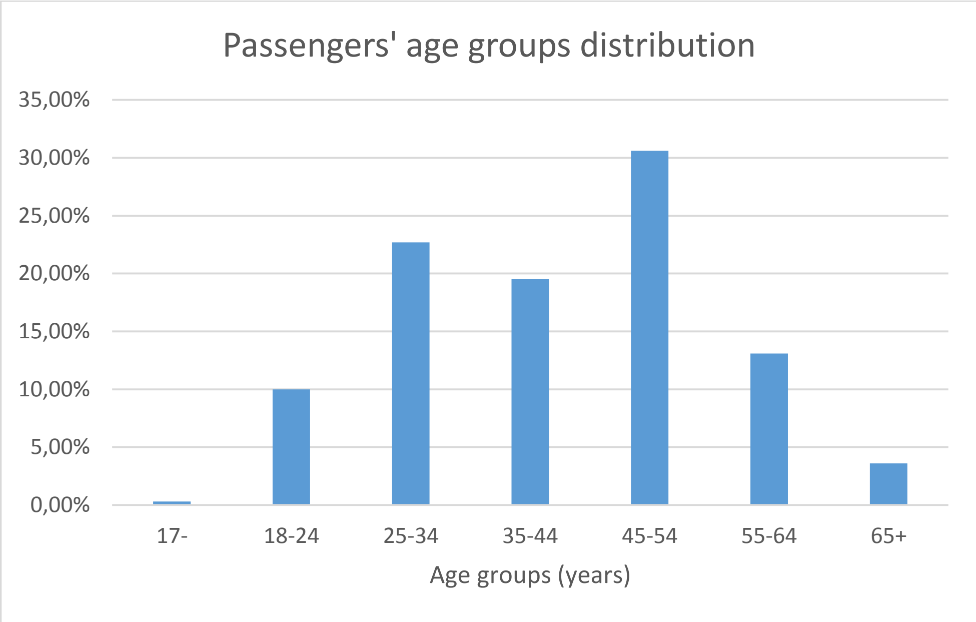


Figure 22. Age groups distribution for low-cost passengers in Lisbon airport (numbers taken from Turismo de Lisboa, 2014)

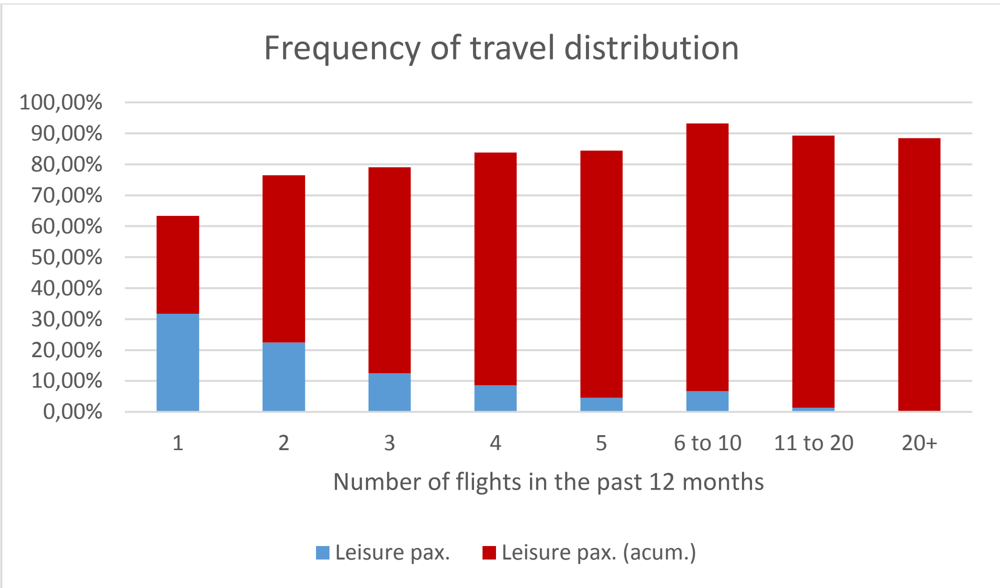


Figure 23. Annual frequency of travel distribution for low-cost passengers in Lisbon airport (numbers taken from Turismo de Lisboa, 2014)

Table 7. Reasons for low-cost passengers to be in Lisbon airport (numbers taken from Turismo de Lisboa, 2014)

Foreigner living in Portugal	2.1 %
Visiting someone in Portugal but living in another country	18.2 %
Visiting Portugal for work purposes but living in another country	2.9 %
Spending holidays in Portugal but living in another country	53.2 %
Native Portuguese living in another country	16.5 %
Living in Portugal but visiting someone in another country	8.4 %
Living in Portugal but visiting another country for work purposes	6.5 %
Living in Portugal but spending holidays in another country	10.3 %

Chapter 5. Simulation processes, agent-based modelling and structure of the model

5.1. Introduction

The current chapter is where the whole developed simulation model will be conceptually explained. Before developing the actual model, it is important to recall what its main objectives are in order to better choose amongst the different available options regarding types of modelling, simulation processes and software tools, accordingly to the model's main goals; once those decisions are taken, the general structure for the model will be explained and subsequently applied by means of the chosen software.

5.2. Objectives of the model

The model this research is aiming to develop should be able to represent passenger flows in a realistic way inside an airport terminal, from the moment they leave security until the moment they leave the building in order to board their aircrafts. In order to achieve this target, the proposed model should fulfil the following objectives:

- To generate the passenger arrival distribution from security in a logical way according to the specified flight schedule;
- To generate and characterise each air passenger on an individual level, taking into account the most relevant individual features in the scope of a discretionary context inside an airport. Such characteristics will be taken from Chapter 2 of this dissertation;
- To establish an individual choice model among a finite set of alternatives, in order for each passenger to undertake the activity he or she wishes the most at a given moment during dwell time after security. That model should include motivational differences between different options;
- To model each possible alternative in a realistic way, both in terms of architectural location/positioning and duration;
- To accurately represent, for every passenger, the boarding process of each generated flight on the specified schedule. As soon as the passenger notices that boarding has started, he or she should immediately integrate the process and give up on any activity other than that one.

5.3. Choices regarding types of modelling and software tools

In the aim of the present work, the impossibility of physically changing the layout of the terminal and study the passengers' reactions led to the decision of creating a whole new model, running it and then drawing important conclusions from it. Taking into account the fact that an airport is a dynamic system in which people do not necessarily follow a linear behaviour, where there are time and causal dependencies and where uncertainty is often present, an analytical model would not be adequate. A system that would somehow reproduce the structure of the problem itself was considered to be necessary, and that can be more easily achieved through simulation.

Having taken that decision, it becomes necessary to choose the type of modelling. In the context of this dissertation, individual modelling is extremely relevant, since one of the main challenges here is to model pedestrian behaviour in a realistic way. Hence, the possibility of creating an agent for passenger modelling and define all the necessary variables part of that agent becomes a very interesting characteristic to have in mind when considering agent-based modelling, since it allows the model to better represent more complex human behaviours. Furthermore, according to Grigoryev (2015), airports are placed in a medium abstraction level, which means that they can also be studied through agent-based modelling. In fact, this type of approach has proven to be an interesting option for modelling problems like the one treated by Ma (2013) and the one this research focuses on, namely because time-space dynamics are extremely important in an airport from a passenger's point of view, and those dynamics become easier to model in an agent-based model. A passenger is meant to be an autonomous and proactive entity who can analyse its environment, make a decision based on that and its personal characteristics and then create its own route towards the following target/facility. Each one of them has its individual variables but, in the end, all of these independent objects interact in order to achieve common objectives. Agent-based modelling thus seems like a more attractive option in this work than the other two, at first sight.

Some particular processes, however, need to be pre-defined by the user and undertaken by some of the entities (the passengers). In fact, the non-aeronautical activities mentioned throughout this text necessarily occur within the departing process of the passengers: they take place, more specifically, between the end of security check and the beginning of the boarding process for each individual. As previously mentioned, such processes happen at specific moments in time and are the same for every single entity (passenger), who integrates them with a passive attitude. They can easily be modelled through punctual events and processes in the simulation system, making the discrete-event approach adequate to represent them.

A multi-paradigm is thus needed in the aim of this work. This mix will be mainly made out of agent-based components but will include some discrete-event elements as well.

Having decided that the airport system in this case study will be modelled through a mix of agent-based and discrete-event approaches, it is now time to decide on which simulation software shall be applied. Among the software tools mentioned in Chapter 3, AnyLogic seems to be an interesting one in the scope of this work. It allows the user to choose from the three types of simulation modelling (namely agent-based and discrete-event behaviour) and it has a very strong and intuitive graphic component, which is very valuable when trying to improve an architectural layout. Animation, in particular, allows the user to have a good idea of how functional and logical the model is and how the entities move around the designed space, which can be very interesting when testing different space-related solutions. Each agent can be devised with as much detail as desired and every parameter can be statistically defined or measured in the software.

Several items and functions are available among the several libraries included in the software (as previously mentioned in Chapter 3, there is a specific library for pedestrian modelling, among others). However, the software also includes a programming interface in Java language for most of the elements – therefore, and because individual choice processes are very complex problems to approach, the present work will end up containing a considerable amount of code lines in Java too.

5.4. The simulation model

5.4.1. General concept

The whole model for people behaviour will be based on a task list: each pedestrian, when generated, will have a list of tasks they wish to accomplish before boarding. The number and types of tasks each person has in mind will depend on their personal characteristics. The general structure of the model, common to every single person who is generated by the simulation software, is the one represented in Figure 24 (the different steps are described with detail immediately after the figure itself).

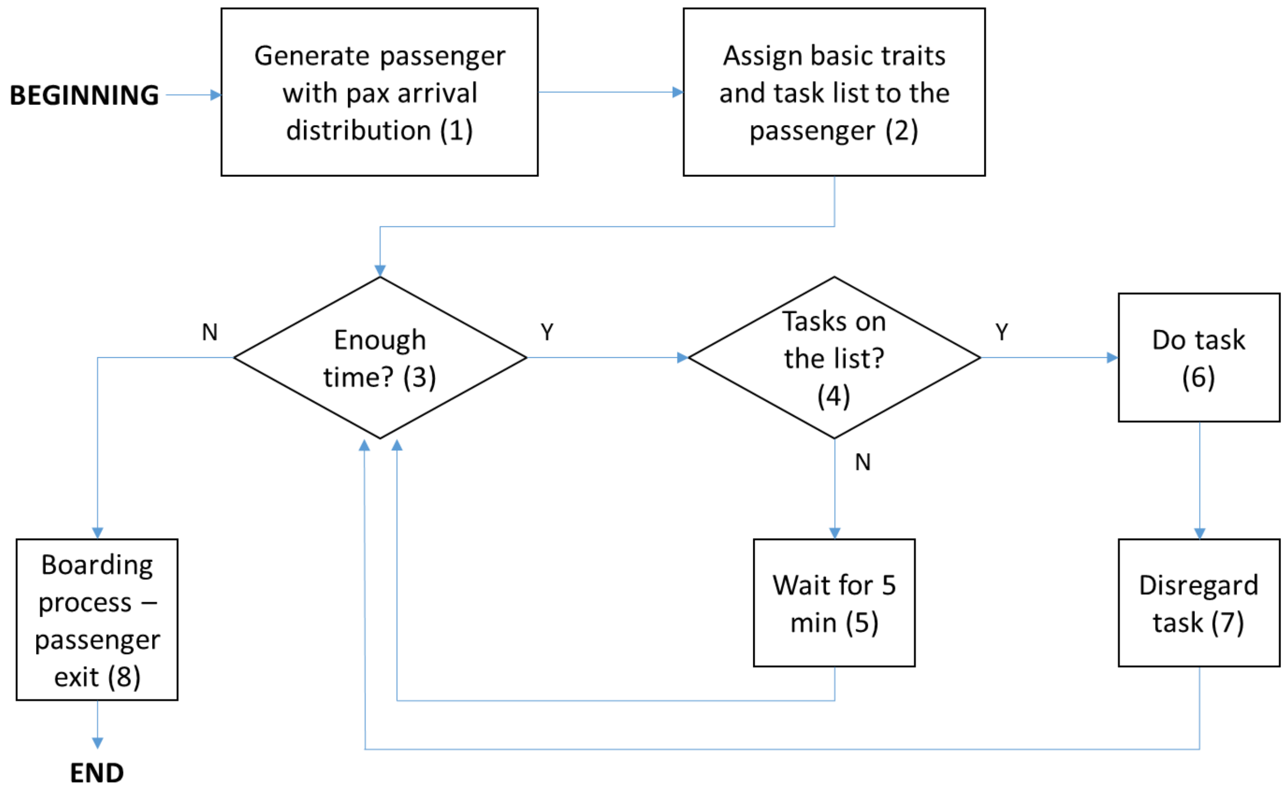


Figure 24. Diagram of the model's general structure for people's behaviours inside the terminal

The general reasoning, put into words and with a higher level of detail, is the following:

1) Firstly, the individual is generated, according to a certain arrival pattern that will be detailed further on in this chapter. The pedestrian enters the terminal at a specified location, defined by a target line inside the terminal. At this point, the pedestrian has no specific characteristics or tasks in mind;
2) It is then necessary to assign that person their specific features and, subsequently, the tasks they wish to accomplish while inside the terminal. Two specific functions are called in the software – one that attributes specific characteristics to the passenger (travel class, sex, age, frequency of travel and whether they are foreigners or not) and another one that generates a task list from those specific characteristics and calculates the probabilities for each one of them to be undertaken. Obviously, the sum of all the individual probabilities for each activity on the list needs to equal 1 (unless the list is empty). These two functions will be mentioned again and further into detail later on in this work;
3) Remaining time until boarding is then calculated. If boarding has not started yet, the passenger considers that they have enough time to undertake at least one of the activities they might have in mind; otherwise, they start boarding straight away (which is point 8 of this sequence);
4) If there is enough time, the pedestrian checks whether they have any tasks left to fulfil or not. If they do not, they go on to point 5 of this list; if they do, they go on to points 6 and 7;
5) If the person is generated with no tasks at all, or in case they have already done them all, they go to the main waiting area. Every five minutes from that moment on, and as long as the boarding does not start, they check again whether they have something to do in their list or not. The reason why this constant verification is necessary is because people's hunger/thirst and need of using a restroom naturally increase with time (this is modelled thanks to an event that will be detailed later on in this chapter);
6) If the passenger has something to do, however, they need to select which activity they will undertake. That choice process will be described hereinafter;
7) After accomplishing a task, it is not considered as being part of the list anymore. A function is called in order to calculate the new probabilities and, after that, the passenger goes back to point 3 of this sequence. The sum of the new probabilities (excluding the task which has just been done) is 1 and a new task is chosen inside this function. If there are any on the list and if boarding has not started yet by that time, that task will be undertaken next;
8) This general cyclic reasoning goes on until the passenger reaches the conclusion that boarding has already started and goes to the assigned gate in order to have their documents checked and board. The model's attainability does not go any further from this point.

Regarding the way each individual chooses his or her next task, a discrete choice model will be used. It is inspired, more precisely, by the Utility Theory and the MNL formulas in order to determine what the next task will be at each moment. The most important formulas for this type of model have already been specified in Chapter 2, in the form of equations (1), (2), (3) and (4).

In this case, instead of a typical utility function, another function will be used. It was decided that this function would be called "propensity function", noted Prop, and would be defined by the formula in Equation (6).

$$Prop_{ni} = \frac{Priority_i}{distance_{ni}} \tag{6}$$

Being:

- $Prop_{ni}$ the propensity for an individual n to choose alternative i;
- $Priority_i$ the priority value of alternative i: a value between 0 and 100 which is illustrative of the dominance of some activities over others, in the sense that some might be, intuitively or statistically speaking, more important for an individual to undertake in the first place than others. This concept was taken from Ma (2013);
- $distance_{ni}$ the distance (in meters) between the pedestrian location at a given moment and the point where activity i takes place. It makes intuitive sense that, the further away a given facility is, the less interesting it will be for an individual to undertake it, thus constituting the denominator of the fraction for the propensity calculation. This distance is obtained through the coordinates of the points and the classical formula taken from the Pythagorean theorem: given two points A and B with coordinates (x^A, y^A) and (x^B, y^B), the distance between them is given by: $d_{AB} = \sqrt{(x^B - x^A)^2 + (y^B - y^A)^2}$.

It is here assumed that equation (4) is still valid, but with propensities rather than utilities. Therefore, the passengers' choice process will still be based on it. Probabilities for each alternative i to be picked by an individual n will thus be given by the formula in equation (7).

$$P_{ni} = \frac{e^{Prop_{ni}}}{\sum_j e^{Prop_{nj}}} \tag{7}$$

Having these general ideas and steps in mind, it is now important to define the main general elements that intervene. Taking into account the fact that the model only focuses on the post-security period of time, it is important to define the passenger arrival pattern from security check, as well as the boarding process. All discretionary activities will necessarily take place between these two moments in time.

5.4.2. Model implementation in the software

In order to implement this conceptual idea in AnyLogic, several different "families" need to be created in order to model different stages or elements and, of course, an agent needs to be implemented in order to represent passengers. In the end, all these types of objects will interact together in the main working space of the software interface, called Main.

For each "family" of elements that will be described in this subchapter, the necessary parameters and variables will be summarily described and will have their names specified in Appendix 2 in alphabetical order, in order for the reader to better understand the code lines in Appendix 3. It can be said that Appendix 2 works like a “glossary” for Appendix 3.

The Flight type of object

First of all, it is necessary to create a type of object called Flight that will represent each and every flight that will be generated.

Logically, each flight has some main characteristics that define it as a whole and make it different from any other flight occurring on a given day. Four properties are distinguished – which means, in practical terms, that four different parameters are created inside the agent Flight: one to designate the flight reference, one for the departure time, one for the number of passengers and one for the boarding gate. These elements are specified in Appendix 2.

The Task type of object

As previously mentioned, one of the main ideas behind the described model relies on the fact that each individual generates and maintains their own list of tasks. Therefore, an object specifically designed to represent these tasks is necessary.

The different available possibilities for tasks are defined inside a list (“option list” in the software) called TaskType, where each line correspond to one possible activity/task. The different options and probabilities are taken from Ma's work (2013). A given person might be willing to undertake any of the following types of tasks:

1. Use a restroom;
2. Visit one or more shops and, eventually, purchase something inside them;
3. Eat or drink;
4. Ask someone for some kind of help or assistance along the way (ask for directions, for example);
5. Use technology in order to socially connect to the internet through an electronic device;
6. Withdraw money from a cash machine.

With the existence of this list, it is possible to create the object Task, which is actually a very simple one, with two parameters only: one for the type of task and one for the index of the facility, in case there is more than one possible option in order to fulfil a given type of task. For example, if there are X different shops, a given passenger might wish to go to shop 1, shop 2, …, or shop X. Therefore, this parameter will have its value between 0 and X-1. These two parameters and their names can be seen in Appendix 2.

Thanks to this new type of object, it will be possible to create parameters belonging to the type Task in the following steps.

The Passenger agent

As previously mentioned, an agent is an entity that analyses its environment and makes decisions by its own, based on what it observes and its individual characteristics. In the model that is here being developed, the only proper agent that needs to be defined regards the passengers.

This is the agent where all the necessary individual elements are defined in order to accurately model the way people behave. The elements with a fixed value are created as "parameters" in the software, whereas the ones that might vary with time are created as "variables"; when necessary, "collections" are created, in order to gather several parameters or variables of the same type. A table that puts together all of these elements can be found in Appendix 2.

Firstly, each person who arrives from security logically has one associated flight they need to catch at the moment they leave the terminal, so one parameter is needed for that.

Also, as previously mentioned, each person will have a task list associated. Therefore, a collection is necessary in order to represent that list, and all the elements inside it belong to the type Task. It has no initial contents, but this is where tasks are created when the passenger is generated.

Several parameters are necessary for passengers to be different between each other and, therefore, originate different lists of tasks. These are called the basic traits. There are five of those in the developed model: one for the travel class (business or economy), one for gender, one for the frequency of travel, one for the nationality (either native or foreigner) and one for the age class.

Once these characteristics are specified for a given person, the list of tasks is generated from a set of probabilities, depending on the different possible combinations for the basic traits listed above. These probabilities do not have any default values, since they strongly depend on each person's characteristics and only have a specified value inside the function that generates the tasks for an individual. Values for these probabilities, taken from Ma (2013), can be seen in Appendix 1.

It is now important to get into more detail regarding the way the individual choice process is modelled:

When the passenger is generated and assigned his task list, that same function calculates the propensity values for each one of the available options, as well as the probabilities. If the passenger has got something to do, he needs to select which activity he will undertake. That selection works through the probabilities that were previously calculated. A random number is generated between 0 and 1; depending on its value, it will fall into a given interval of probability sum, which will define the chosen task to be done: supposing that there are N activities in the list, with probabilities P_1, P_2, …, P_N, the pedestrian will choose activity 1 if the random number is greater than 0 but smaller than P_1; if it is greater than P_1 but smaller than $P_1 + P_2$, activity 2 will be the chosen one; that same reasoning applies until the last interval – if the random number is greater than $P_1 + P_2 + \ldots + P_{N-1}$ but lower than $P_1 + P_2 + \ldots + P_{N-1} + P_N$ (which will necessarily equal 1), activity N will be chosen. The higher the propensity value of an option, the higher its probability will be, making the associated interval broader and hence increasing the probability for the pedestrian to choose it. If there is more than one task, however, there is always a chance that the passenger ends up choosing another activity with smaller values of propensity and probability. This is how the model includes people’s less linear or predictable behaviour.

After accomplishing a task, another function is called in order to calculate the new propensities and probabilities. In order for the software to calculate the propensities and probabilities for an individual to undertake each possible activity, four sets of variables need to be created:

- As mentioned above, each propensity value will necessarily need to be calculated more than once in the model. Therefore, given the need of developing general functions that calculate (or recalculate) these values at the same time for all types of tasks, a new set of integer variables is needed, with 0 as their default values. Their names all start with the word *task*. If a type of activity becomes part of the list, its correspondent variable *task* passes from 0 to 1. From this moment on, the software will only calculate propensities for tasks that have this variable equal to 1; otherwise, they will keep their initial values;
- In order to calculate propensities, variables for priority values need to be defined. Their default values are gathered in Table 8. Four of these values were decided to be the same as the ones in the work developed by Ma (2013), so four of the rows if the table are essentially the same as the ones in Table 5. It is also assumed that using the restroom is equal to eating or drinking in terms of priority, and that asking someone for help is less of a priority than that but still more than the rest, being therefore given the intermediate value of 65. There is no column for “time stress”, since the passenger goes directly to the boarding process if there is not enough time to undertake any discretionary activity;
- Once propensities are calculated, their values are assigned to a specific set of variables that change over time. Their initial values are set to - 100 000, which is extremely low and leads to null weights in the calculation of the probabilities, as well as null probabilities;
- Once propensity values are calculated in the Main agent, probabilities for each possible alternative on the list are calculated through Equation (7) and their values are put into a specific set of variables too. The number that will be randomly generated between 0 and 1 will be put into an independent variable at all times.

Table 8. Priority values in the developed model

Actions	Priority values
Buy a drink or meal	+ 100
Use the restroom	+ 100
Ask for help	+ 65
Shopping	+ 60
Emails, phone calls and social media	+ 40
Withdraw money	+ 30

Finally, one variable is necessary to determine which task is chosen at each cycle. The number of possible values for such variable, apart from its initial value zero, is the same as the number of alternatives in the set of options. After an activity is done, its value is set to zero again and only changes when the next activity is chosen by the entity.

It is important to mention that, in the particular case of the activities related to eating/drinking or using a restroom, their priority values are set to zero after the activity is done, but they start gradually increasing with time from the moment they become null, and therefore the associated propensities will take positive values again at some point. When their values become important enough, they might be selected as the activity to do next by the pedestrian. In other words: in those particular cases, the task is only deleted from the list until the priority value becomes greater than a certain value again; from that moment on, the task is back on the list again, until that necessity is satisfied. This increase of the priority values can be achieved through an event that increase their values periodically, as well as two other parameters that establish what the periodic increment values are. All of these elements are listed and can be seen in Appendix 2.

Passenger arrival pattern (the PaxArr object)

A distribution of the passenger arrival process will be assigned to each and every single flight on the schedule, so the only necessary parameter in order for this to work is a reference for each flight. Every created element that is here mentioned is named and listed in Appendix 2.

What this object does is to define the time intervals between arrivals for three different periods of time for each flight, thanks to a function inside it: it considers that 25 % of the people arrive to the airport on the first third of the time, then 50 % of the people arrive on the following third and, finally, the remaining 25 % arrive on the last third of the period of time before the end of that arrival process. In order to do this, the function starts by defining an exponential arrival pattern for each one of the intervals of people and puts the values into a collection; it then uses a simple rule of three in order to calculate a time scale, which then multiplies by the previously determined values in order to convert them to an actual time dimension. This scale is calculated for the three arrival patterns and put into an array with size 3, by simply dividing the correspondent time dimension by the last arrival in that interval. In the end, the function calls an event that injects the correspondent number of entities into the passenger generator in the agent with the desired cadence. This distribution was adapted from the *Airport* example available in AnyLogic software; it was then discussed and approved by some researchers before being applied to this work. This function is called immediately on start-up of the object. Its code lines, as well as others that will be mentioned throughout this chapter and the next, can be seen in Appendix 3. It is important to mention that two of the variables used in this code belong to the Main agent, which will be explained further on in this chapter.

However, that procedure only defines a passenger arrival pattern to the airport. Keeping in mind that the model that is here being developed only focuses on post-security events, it is necessary to somehow represent the time people spend checking in, dropping off their baggage, queueing for security and passing through it. This is done through a “delay” element that introduces a certain temporal delay right after the passenger generation. Figure 25 represents the visual aspect of this agent. The lower right icon on the figure is nothing more than that – an icon for this object, in order to determine the way this process appears in the main working space of the interface.

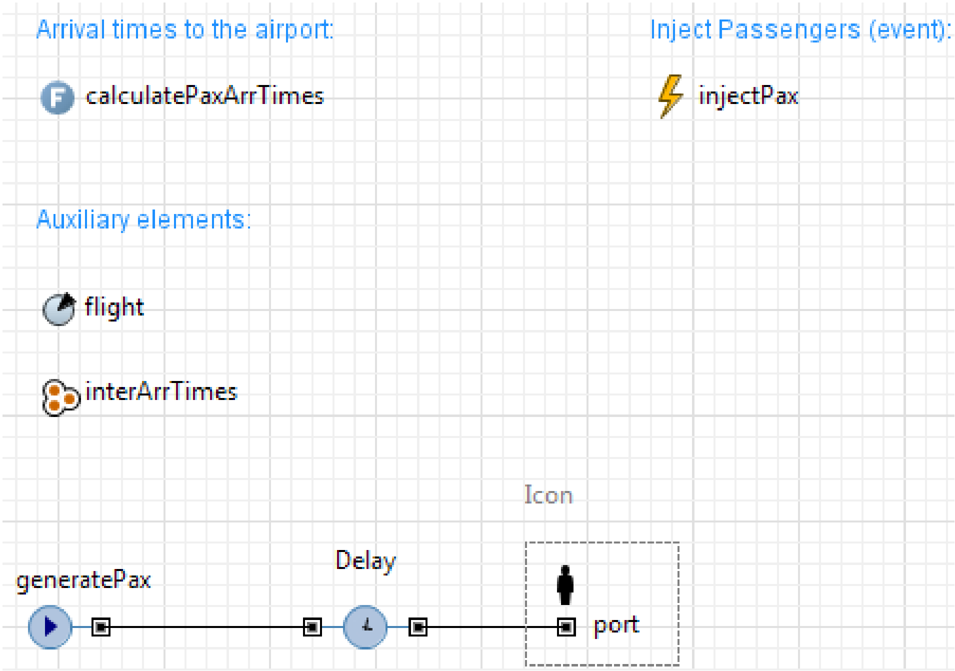


Figure 25. Structure of the PaxArr object on AnyLogic

The Gate type of object

The whole boarding process is complex and it is necessary to make it generic and applicable to several different gates. Therefore, a new object type is created in order to model it. Figure 26 illustrates the visual representation of the object in the software interface, while the different elements that constitute it can be found in Appendix 2.

Passengers only enter this object when the boarding process has started. However, they do not all go immediately to the gate control process – some of them prefer to wait before joining the queue. This is established by a facility that is associated to a specific waiting area, definable by the user.

Once that waiting period comes to an end, the passenger goes through gate control, which is modelled by a service for pedestrians. This stage is cancelled when the boarding process is over – at that point, the passenger is late and misses the flight. If everything goes well, on the other hand, the passenger moves on to the final step of this process, which is to actually board and disappear from the terminal. Before that, however, and depending on the terminal geometry, the passenger might have to wait in a specific area of the gate until the moment the boarding time has come to an end (as mentioned in Chapter 4, this situation happens on the case study of this dissertation), so another waiting area is necessary between the service and that last stage. Many of these steps come from the pedestrian library available in the palette of the software.

For each one of them, some spaces and values need to be specified, so new parameters and variables need to be created inside this Gate agent, as seen on Figure 26:

- The pedestrian will wait at a waiting area, then queue and go through gate control, then wait at the specific waiting area for the gate and, finally, disappear from the building;
- Three parameters are necessary in order to assign values for the duration of the gate control process – one minimum, one maximum and one mean value. It is assumed that the process follows a triangular distribution depending on these values;
- Logically, each gate will have a number, so a parameter needs to be created for that;
- Finally, every flight will have a gate associated to it. Therefore, a variable is necessary for that relationship as well. It will be used in the boarding event, which will be described with more detail further on.

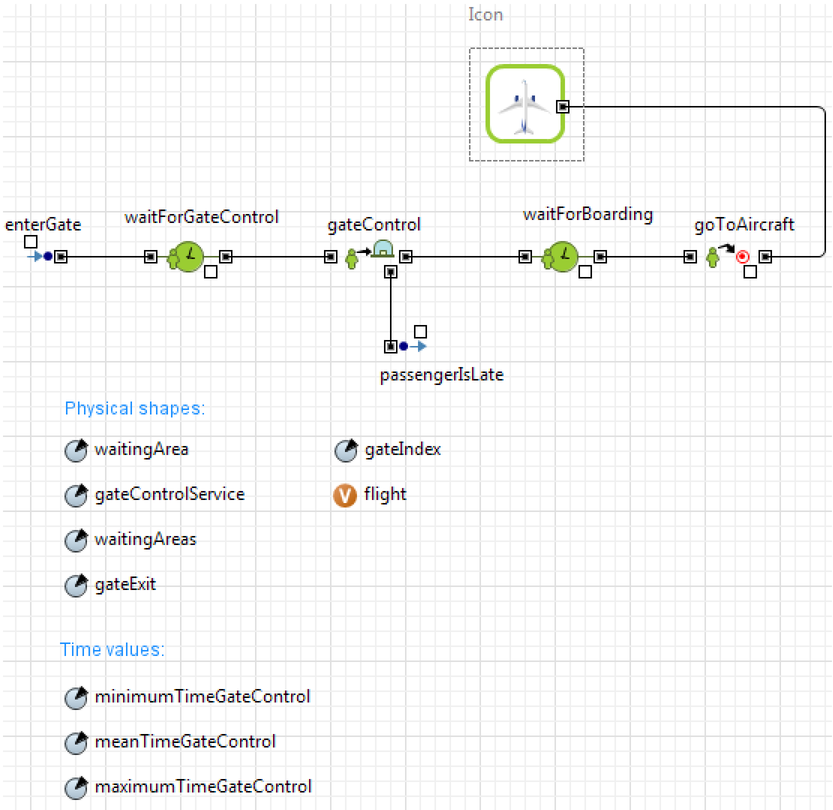


Figure 26. Structure of the "Gate" object type in the model and its icon on AnyLogic

It is assumed that the time people spend waiting before moving on to gate control is given by a triangular distribution between:

- the remaining time until boarding starts;
- the remaining time until 5 minutes before boarding finishes and;
- a quarter of the sum of the remaining time until boarding starts and the remaining time until there are only 5 minutes left before it finishes.

In other words: it is considered that, from the moment when the boarding process starts, and while it lasts (except in the last 5 min), people follow a triangular distribution with its peak at one quarter of the boarding period, in order to reach the gate control process. In case the result of that triangular distribution is negative, they do not wait at all and immediately proceed to gate control.

The Main object – working space of the user interface

Once all the other objects are individually defined, it is possible to gather them in one single space and make them interact – this is the Main object. Here, the general logic of the model is implemented: everything fits and works together as a whole. It is, however, necessary to build some new elements in the software as well.

The general concept described at the beginning of this chapter is implemented through a scheme for pedestrians, which Figure 27 illustrates. Every single element in that scheme is exclusively dedicated to entities that belong to the agent Passenger. Each numbered step in the figure is described in detail in the following sequence:

1) The individuals are generated thanks to the PaxArr object, which is located at the beginning of the scheme; they are then physically introduced in the model;
2) Two functions are called and applied to the entity (the passenger): the first one of them assigns the basic traits to the passenger, while the second one generates the task list from those basic traits and then calculates all the propensities and probabilities. The first activity to be undertaken (if any) is also defined in this latter function. This is achieved inside a horizontal object called "plain transfer" that can be seen on the figure;
3) The system then checks whether boarding has already started or not. In mathematical terms, it calculates the difference between the departure time and the current time and compares it to a parameter that accounts for how much time in advance the boarding starts, compared to the departure time. In case boarding has already started, the passenger moves on to point 8 of this list;
4) If the passenger has got enough time to undertake discretionary activities, the model checks whether there are any more tasks that have not been done yet or not. If all of the propensity variables equal - 100 000, the passenger is considered either not to have any tasks on the list (the variables maintained their initial values) or not to have any more tasks to fulfil, since a given activity's propensity changes to - 100 000 after the passenger undertakes it (as seen in point 7 of this list);
5) If tasks have already been accomplished, the pedestrian goes to the main waiting area (associated to a specific waiting area) and has its remaining time and his presence or absence of tasks checked every five minutes until the remaining time is insufficient or a task is on the list, as previously described;
6) However, if the passenger has a task on the list (with the corresponding *task* variable equal to 1) and has enough time to do it, he will. The task selection is made through the previously described process;

7) Every time a task is fulfilled, the passenger's variable that indicates their next task becomes null. Also, the task's propensity value is set to - 100 000 again and a function is called. All this function does is calculate the new distances, propensities and probabilities, taking into account the current position of the pedestrian in space and the tasks that are still on the list (the ones with the *task* variable equal to 1). In the case of the food- and WC-related tasks, priority variables are set to zero at that moment as well; in the case of the other tasks, the correspondent variables *task* are set to 0, so that they will not be considered anymore in that function (there are some conditions in the code lines in order not to consider them). This is the way for the model to "delete" a task from the list, even though it does not actually delete it – in fact, it just turns it into an insignificant alternative, with null weights and probabilities. After that, the passenger goes back to point 3 of this sequence;
8) The moment the passenger enters the boarding process is represented by the object Gate and its icon. The pedestrian can either board on time or not; in both cases, they disappear from the system at that moment, through a "sink" for pedestrians.

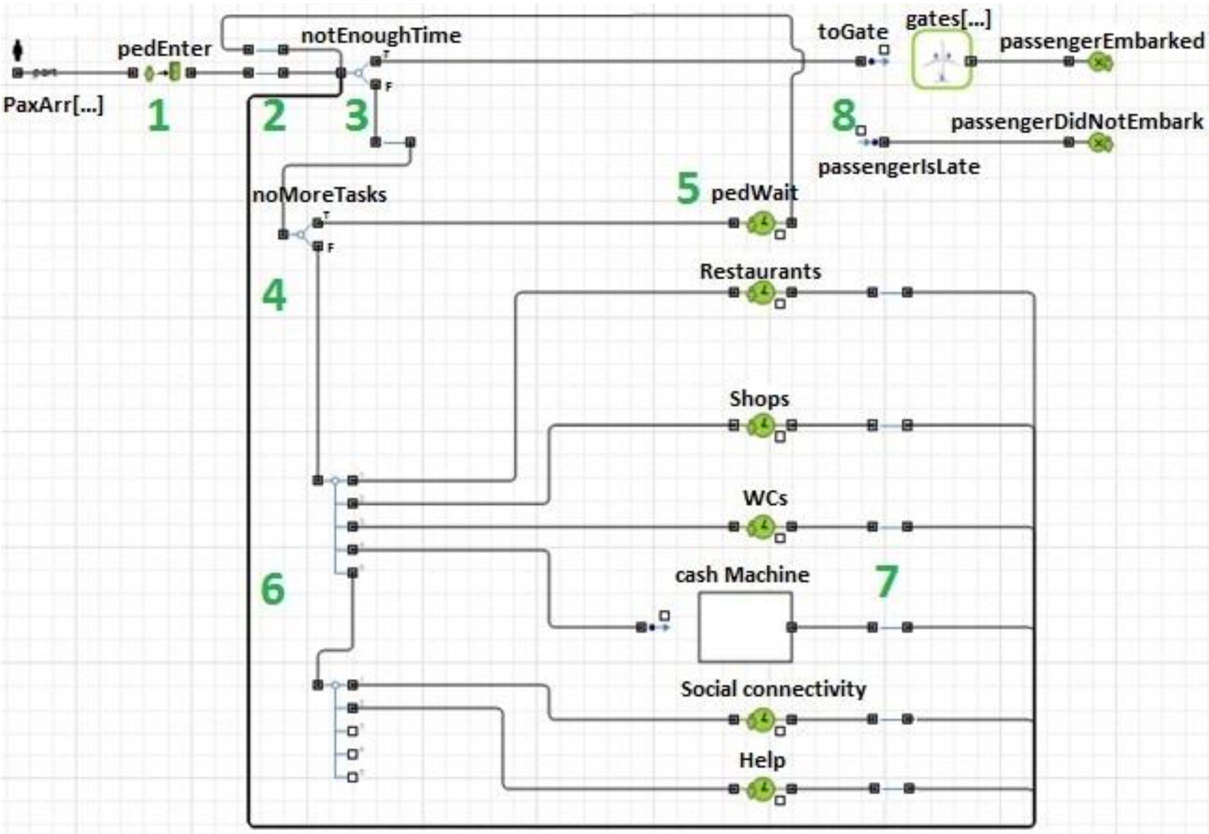


Figure 27. General pedestrian scheme of the model on AnyLogic

The code lines for the mentioned functions in this chapter, as well as others used inside their bodies, can be found in Appendix 3, but can only be fully understood after reading Chapter 6 of this dissertation, since they are already applied to the terminal 2 of Lisbon airport.

In terms of flight schedule, it is based on a modifiable text document that needs to be inside the same folder as the AnyLogic file. This file is read by a function that is called immediately on start-up of the Main agent. This function also puts all of these departures into a collection called *flights*.

Also on start-up, two other functions are called: one defines, for each flight, three periods of time for passenger arrival patterns (which will correspond to three different intensities, as already explained in the description of the PaxArr object) and puts each one of them into an array of size 3. Five temporal parameters of type double are needed in those simple calculations and can be seen in Table 9, as well as in Appendix 2.

Table 9. Time parameters in the Main object of the model

Name	Description	Type
timeFromCheckInOpenToDep	Time between the opening of the check-in process and the flight departure time	Double
timeFromStartPaxArrToDep	Time between the arrival of the first person and the flight departure	Double
timeFromFinishPaxArrToDep	Time interval between the arrival of the last person and the flight departure	Double
timeFromHighDensArrStartToCheckInStart	Time between the beginning of the high density arrival period and the moment when check-in starts	Double
timeFromCheckInStartToHighDensArrFinish	Period of time that the open state of the check-in and the high density period have in common	Double

The other function creates, for each flight in the schedule, boarding and departure (dynamic) events. In order for it to work, another parameter is needed, which defines the time period between the end of the boarding process and the departure of the aircraft. The boarding event has in its parameters a Boolean named *begin* – when it is true (which happens when the boarding starts and while it lasts), the flight is added to a collection that contains all the active flights at a given moment; when it becomes false (when the boarding process is over), the flight is removed from that same collection because it is not active anymore and the gate control process is cancelled, causing the passenger to be late for the flight. The departure event takes place at the flight's departure time and all it does is remove it from the collection of flights. Code lines for both these events can be seen in Appendix 3.

The other very important part about the Main object is the geometry of the problem – it is fully defined inside it. Hence, it is expectable that the last set of elements that needs to be mentioned regarding this agent solely regards the geometry of the problem. These can be seen in Appendix 2.

5.5. Model verification

As described by Reis (2010), no model can remain untested if its creator wants it to be used by other people dealing with similar matters. Therefore, every model should, at least, be verified and, eventually, validated. In fact, these are two different concepts: according to the same author, verification consists of testing how much the model is working in line with what its creator has previously specified and assumed; validation, on the other hand, is more linked to reality and refers to the accuracy of the model results when compared to the ones originated by the real world system that is being modelled.

Being one of the main objectives of this work the modelling of individual choice behaviour, it makes sense that passengers should be studied individually before considering the model as being valid for a whole population of agents. Hence, in order for the model to be verified, this simple simulation is made repeatedly and extensively at every important step of the model construction. That can be done by changing the schedule file so that only one flight, with one passenger alone, is listed and generated. That one person can then be followed and studied by printing some variables' values on the software's console and checking whether everything is making sense and working as predicted at each step of the cycle or not.

Commands are thus given to the software so that the passenger's basic traits are printed, as well as the list of tasks and the initial values for propensities and probabilities. After that, values for propensities and probabilities are checked at every important step of the process, that is, whenever they are recalculated. At each point, the sum of the probabilities for the tasks that have not already been fulfilled has to equal 1. Values for the random number between 0 and 1 are also printed every time it is generated and the chosen task has to be in accordance with the probability interval where that number falls into.

This was done several times, for a considerable number of individuals, with different basic traits and different task lists. The model was also shown to researchers other than the author (namely the author's supervisor) in order to receive exterior opinions and suggestions, since the modeller often ends up disregarding some of their own inaccuracies (Reis, 2010). It is noteworthy, however, that one can seldom verify their model with the certainty that it is working 100 % as expected – the modeller can, at best, build and grow a personal trust in the model and its output values, but it will always be, by definition, a simpler version of reality and can never represent it perfectly (Reis, 2010).

Once the model is considered to be trustworthy, it can be run with many more people, more flights and a longer simulation period. The visual aspect of one of those runs, with a random schedule applied to the specific case study of this dissertation, is shown in Figure 28. People on the figure can be seen boarding at gate 6 while the others are undertaking other, discretionary, activities (or moving between them).

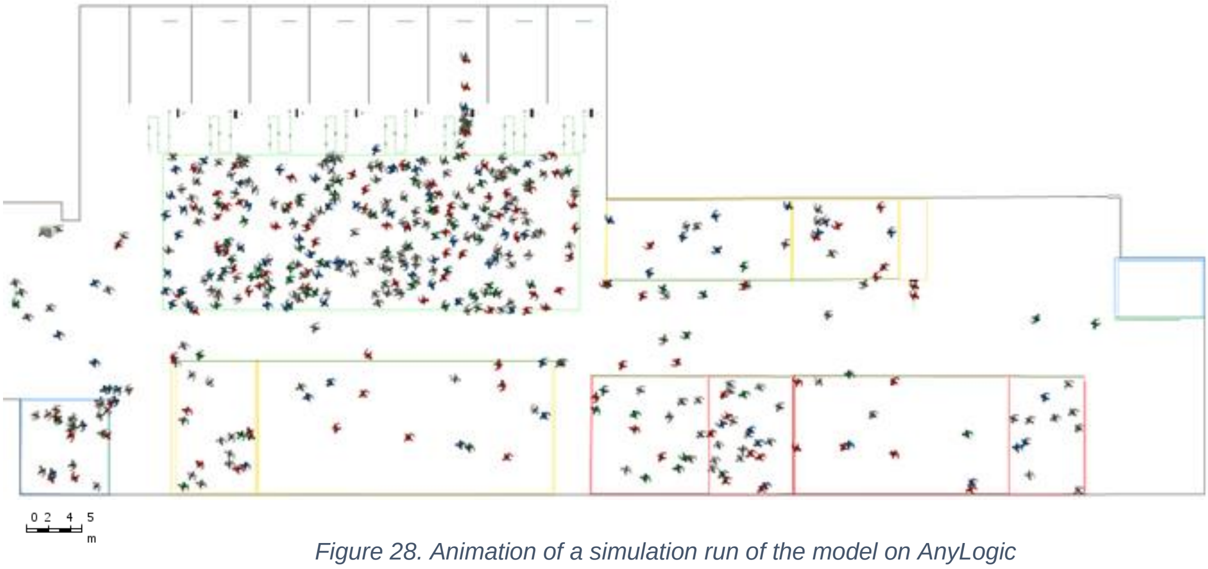


Figure 28. Animation of a simulation run of the model on AnyLogic

5.6. Performance indicators

Given the model that has been being described throughout the present chapter, some indicators are needed in order to study the current situation and draw conclusions about how much room for improvement there is (and, when studying other possible alternatives, whether those improvements have been met or not). These indicators represent, in other words, the system's performance from the scope of this work. They aim to represent important factors both from the passengers' and the managers' points of view.

Five principles and types of indicators are here taken into account:

- Following the same reasoning as some authors mentioned in Chapter 3 (Bandara & Wirasinghe, 1992; Tam, 2011), it is here considered that, the less the people are required to walk, the better it is from their points of view. Therefore, walking distance minimisation also becomes an important objective of this research, applied to the described case study. This indicator is checked through the final value of a specific variable defined inside the Passenger agent: each time the pedestrian decides to undertake an activity (either one of the discretionary or simply wait for boarding), the distance (in meters) to the destination is added to that variable. This also happens when the passenger finally boards – at that moment, the variable reaches its maximum value and does not change anymore. Of course, this value will mainly depend on the initial list of tasks for each person and on the time they arrive to the terminal;
- Secondly, people should be able to fulfil all of the activities defined in the initial task list. This is defined through a variable specifically defined inside the Passenger agent for this purpose (when all the propensity values for an individual equal -100 000 for the first time, he is considered to have fulfilled all the tasks and that variable's value is set to 1). A counter is also defined inside the main working space, in order to measure the number of people that leave the system with 1 as that variable's final value;

- Thirdly, for each studied layout, the number of people who miss their flights shall be taken into account. This number should be as low as possible for an alternative to be considered interesting. It is automatically measured in the pedestrian sink called “passengerDidNotEmbark” in Figure 27;
- Also, for each individual, in case he or she leaves the system without having completed their initial list of tasks, it is interesting to know which tasks were left undone. In practical terms, these are the activities that appear on the initial task list but arrive to the end of the process with positive propensity values (the others all equal - 100 000). Therefore, the necessary variables are created in order to work as counters for this: each one of them counts the number of people that leave the system without having achieved one particular activity they still have in mind at the moment they leave (one variable per alternative). This is interesting both from the individuals’ and the managers’ points of view, since it indicates what the most underperforming areas are for each alternative layout;
- Finally, from a managerial point of view, it is important to measure facility utilisation since it is, in a way, proportional to the revenues: it is considered that people who have “food” in their lists of tasks and have the opportunity of going to a restaurant necessarily end up buying something to eat or drink there; regarding shopping activity, on the other hand, there is no guarantee that people purchase something inside the shops when they visit them, but it is intuitively assumed that, the more people enter the shops, the higher the probability is that they end up buying something inside (either by impulse or because they pre-planned it). The other discretionary activities (using the restrooms, asking someone for help, withdrawing cash or using the internet) do not bring the same additional income to the managing entities, so only counter variables for shops and restaurants are here created. These variables simply count the number of people that have used a given facility at some point in the simulation run. In the end, these variables indicate the total number of visitors for each task on that simulated day.

Most of these indicators are simple to measure, since their values automatically appear on the screen after running the model. Regarding the distance variable, however, it becomes necessary to create an element that collects the individual values for every passenger. It is thus necessary to create one “histogram data” element in the model. What this does is to gather these parameters for each and every passenger that leaves the system and then calculate statistical information about this set of numbers (such as minimum, maximum and mean values, standard deviations, etc.) as it collects them.

Note that, even though the code lines in Appendix 3 reveal most of the programming code that was written in the aim of this model, there are still some code lines that are executed inside the elements of the pedestrian scheme that are not here specified since it would not make sense from a practical point of view.

Chapter 6. Application of the model to the case study

6.1. Model adaptation to the case study

The most obvious consequences of adapting a general model like the one described in the previous chapter to a specific case study regard the geometry of the problem. As mentioned in Chapter 4, information regarding the geometry of the terminal was obtained through ANA's official website (2015) and a physical visit to the place was made.

As mentioned in that same chapter, the terminal has eight boarding gates (plus three "extra" ones), two restrooms, one cash machine, four shops and four options regarding food and drinks. In the developed model, the Immigration process (SEF) and the three extra gates will not be considered. There is a big waiting area next to the boarding gates, where pedestrians wait for their gate control and boarding processes to start. It is considered that people who wish to use their electronic devices for the social connectivity task do it there as well.

Therefore, in terms of locations, the following elements are created:

- Four areas for restaurant services;
- Four areas for shops;
- Two areas for restrooms;
- One service with line for the cash machine;
- One big area for the passengers who are waiting for their flight's boarding process to start, or for those who want to use their electronic devices for social connectivity at a given moment;
- One line for the information desk, although there is not really such object in this part of the terminal – an assumption is made that people who need to ask for help will ask their questions to the staff working right after getting out of the security check;
- Eight boarding gates, consisting of eight services with lines (for the gate control), eight specific waiting areas and eight target lines (where the passengers disappear).

The necessary walls in order to delimitate the physical space and barriers are drawn as well, so that passengers only move around spaces where they are (physically or legally speaking) allowed to. Figure 29 illustrates that architectural layout in the software interface.

This leads us to the geographical coordinates of each facility. Pixels are used in the software in order to know the X and Y coordinates of some specific points in space and their values are put into the already mentioned parameters. A relationship between lengths in pixels and meters needs to be established. It was decided, while making the architectural drawing, that 40 pixels on the interface would roughly equal 1 meter on the real world.

The following ones need to be created:

- Two pairs of parameters regarding the restrooms' coordinates;
- Four pairs of parameters for shops' coordinates;
- Four pairs of parameters for restaurants' coordinates;
- One pair of parameters for the cash machine's coordinates;
- One pair of parameters for people willing to ask for help and want to see how far away it is;
- Eight pairs of parameters for the gate coordinates;
- Four pairs of parameters for the social connectivity task, since the waiting area is considerably big. Therefore, before calculating the propensity for that task, four distances are calculated and, then, only the one with the minimum value is considered in the formula for the propensity determination. By doing this, the passenger will always evaluate how far the waiting area is through the smallest of the four distances, depending on where he is at a given moment.

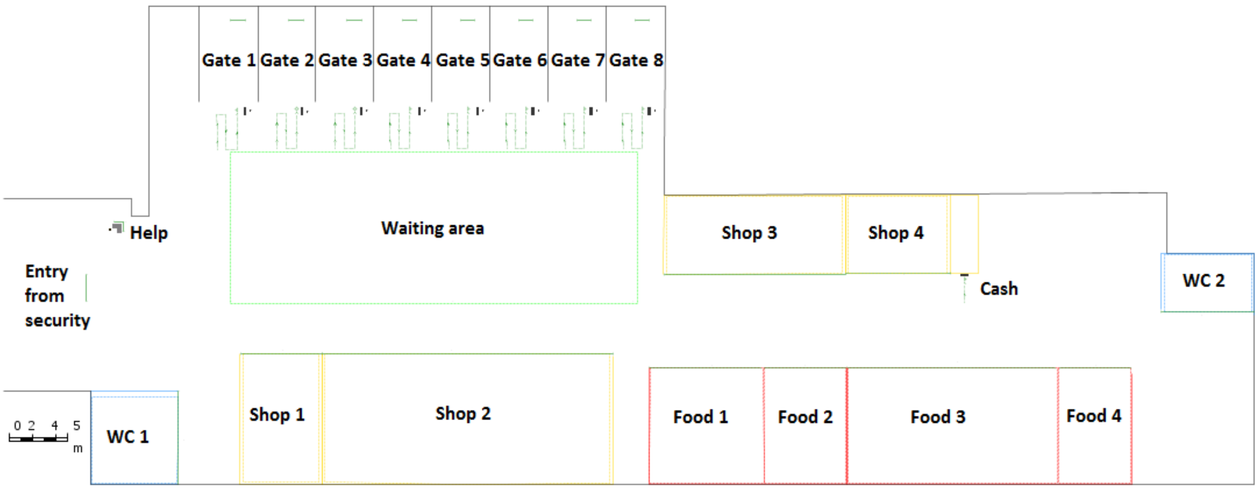


Figure 29. Plan of the terminal 2 of Lisbon airport in the developed model on AnyLogic

Regarding temporal elements, the previously described parameters take the following values:

- People are said to start arriving to the terminal 210 minutes before the scheduled departure time**;**
- Check-in is said to open 120 minutes before the scheduled departure time**;**
- People are said to finish arriving to the terminal 80 minutes before the scheduled departure time**;**
- The high intensity arrival period is said to start 15 minutes before check-in opens;
- The high intensity arrival period is said to finish 15 minutes after check-in opens;
- Boarding is set to start 50 minutes before the scheduled departure time
- Boarding is set to finish 25 minutes before the scheduled departure time;
- Gate control is set to follow a triangular distribution between 5 seconds (minimum), 10 seconds (maximum) and 7.5 seconds (mean);

These values were also taken from the *Airport* example available in AnyLogic software. Once again, they were commented and approved by the same previously mentioned researchers before being applied to this work. They can be seen in Table 10, as well as the parameters' names that are used in the model (specified in Appendix 2).

Table 10. Values for the time parameters in the applied model

Name	Value
timeFromStartPaxArrToDep	210 minutes
timeFromCheckInOpenToDep	120 minutes
timeFromFinishPaxArrToDep	80 minutes
timeFromHighDensArrStartToCheckInStart	15 minutes
timeFromCheckInStartToHighDensArrFinish	15 minutes
timeFromBoardingStartToDeparture	50 minutes
timeFromBoardingFinishToDeparture	25 minutes
gateControlTimeMin, *gateControlTimeMean*, *gateControlTimeMax*	5 seconds, 7.5 seconds, 10 seconds

It is assumed that, if a given person is willing to go shopping, they will visit either 1, 2 or 3 of the available shops. People who are willing to eat or drink, on the other hand, are considered to choose only one among the four available options. In practical terms, these assumptions are defined in the function that generates the task list for each individual (called *generateTasks*), through the creation of the integers *shopCount* and *foodCount* (these functions' bodies can be found in Appendix 3). Regarding the restrooms, it is assumed that people do not have any preference between the two, and therefore propensities are always calculated for both as long as that task is on the list.

Regarding the Passenger agent:

- None of the parameters that define basic traits has got a specified default value - they are all defined when the function that assigns them to the passengers (named *setupPassenger*) function is called, with the values taken from Turismo de Lisboa's work (2014), cited in Chapter 4. In this function (which can be seen in Appendix 3), it is considered that a "foreigner" is someone who lives outside of Portugal, while a "native" is someone who lives in the country, since no real data is available about the passengers' nationalities. Hence, from the values in Table 7, it is considered that 76.9 % ((18.2+2.9+53.2+16.5) / 118.1) of the people there are foreigners, while the remaining 23.1 % ((2.1+8.4+6.5+10.3) / 118.1) are native Portuguese. In what concerns the frequency of travel, exact values that led to Figure 23 can be found in Appendix 1. They justify the values in the code lines of the function;
- The age groups probabilities collection, however, needs to have its specified values inside. Two age classes are considered: people are either under 65 years old or above. Taking the values from the data mentioned in Chapter 4, the initial contents of the collection are {0.964, 0.036}, since only 3.6 % of the people covered by the survey are above 65 years old;
- All probabilities have their values set to 0 as a default value;
- All of the propensities equal - 100 000 and all of the *task* variables equal 0;
- Priority variables have their debut values equal as the ones in Table 8. The food and WC priority increment values are assumed to be 5.56 / 10 min and 8.33 / 10 min, respectively. These values were obtained through an intuitive reasoning where it was assumed that one would not consider using the restroom less than one hour after going there, and that one would not buy something to eat or drink again after less than one hour and a half after their previous purchase in a restaurant. In terms of priority values, this represents minimum values of 50 and 45, approximately (for WC and Food, respectively), before being considered as relevant again after having been set to zero before.

These conditions can be seen in the code lines for the *recalculatePropensities* function in Appendix 3, since the values for these activities' propensities are only recalculated if their priority values are above these minimum values; otherwise, they remain the same as their value after accomplishing the activities, which is - 100 000.

The tables with the probability values from which tasks are generated (Appendix 1) were taken from Appendix C of Ma's work (2013) and their values are the ones used in this model as well, since the data for Lisbon passengers was not available. Most of the time distributions for the activities are taken from the work developed by Ma (2013) as well:

- Dwell time (shop) = normal distribution with alpha = 685 s (as seen in Table 4);
- Dwell time (food) = normal distribution with alpha = 1333 s (as seen in Table 4);
- Dwell time (cash) = uniform distribution between 60 and 70 s (as seen in Table 4);
- Dwell time (WC) = uniform distribution between 160 and 230 s (as seen in Table 4);
- Dwell time (Social Connectivity) = uniform distribution between 1600 and 1700 s (as seen in Table 4);
- Dwell time (Help) = uniform distribution between 5 and 60 s (as seen in Table 4);
- Probabilities that lead to the task list of each individual can be found in Appendix 1. As previously stated, they were taken from the last appendix in Ma (2013). The only activity that is needed in the studied terminal that is not mentioned in his tables is the use of a restroom. Therefore, it is assumed that 70 % of the people wish to use a restroom inside the terminal at some point.

Regarding the passenger arrival pattern, it is defined in the schedule file. In the PaxArr object, the "delay" element is set to a uniform distribution between 5 and 40 minutes. Such values were assumed from the fact that the analysed terminal is exclusively run low-cost companies that usually ask their passengers to check-in before arriving to the airport. Therefore, it is considered that one should not take more than 20 minutes dropping off their baggage (and checking in, if it is the case) nor more than another 20 minutes queuing for security and passing through it. Oppositely, it is assumed that one does not take less than 5 minutes between the moment they arrive to the airport and the moment they get out of security and enter the restricted area of the terminal.

At the time the pedestrian enters the model, two values for the walking speed need to be defined by the user – one for the initial speed and another one for the comfortable speed. Several authors have made various proposals throughout the years, but the chosen one for this research work is taken from Schultz et al. (2010), who are mentioned by Cheng (2014) in her work. These authors recorded data from Dresden International Airport through video surveillance of an area located between check-in and security and analysed it statistically. According to the values cited by Cheng (2014), they concluded that the walking speed for an individual alone follows a normal distribution with mean 1.36 m/s. This will be the adopted value for the comfortable speed in the model described in this dissertation. Assuming here that the initial speed is 0.3 m/s lower than the comfortable speed, its value is said to follow a normal distribution with mean 1.06 m/s.

6.2. The simulation period

The simulation period is set to one entire day. In order to use real data from the case study, ANA official website (2015b) was consulted. The chosen day in terms of flight schedule was September 2nd, 2015. This schedule, as well as the flight reference numbers, companies and destinations, can be seen in Appendix 4.

However, this schedule does not include any information about the number of passengers. Therefore, in order to have an idea about these figures, the following procedure is followed:

1) The departure times, flight reference numbers, companies and destinations are taken from ANA official website (2015b), as mentioned above;
2) The respective aircraft models are taken from FlightAware website (2015), through research of the flights' reference numbers, one by one;
3) Once the aircraft type is known, its configuration can be checked on PlaneSpotters website (2015);
4) The aircrafts are seldom entirely full, however. Hence, and keeping in mind that the companies that operate in Lisbon T2 are low-cost, an occupation rate of 90 % is assumed.

These data and calculations can be found in Appendix 4. It is noteworthy that two of the flights from the list (both operated by EasyJet) had no aircraft specified on FlightAware website (2015). An assumption thus needs to be made: taking into account that all the remaining aircrafts by easyJet that operate throughout the day are either Airbus A319 or Airbus A320, but mainly A319, it does not seem illogical to assume that both these aircrafts missing are A319 as well. These two particular cases appear on the table with bold characters and a question mark before the corresponding name of the aircraft model. This accounts for a total of 6140 passengers in a day.

It is also important to mention that, even though some models on the list are the same, their capacities might be different, depending on the companies that operate them. In what concerns the boarding gates, values are not available online, so their values are fictional.

6.3. Application of the model to the current situation and associated values for the performance indicators

In order to try to eliminate (or, at least, reduce) the effects of randomness in the system, several simulation runs shall be made for each tested alternative. For each one of the measured parameters, the minimum, maximum and mean values will be highlighted, as well as the standard deviation. After some experiments with the model, it was decided that each alternative layout would be run 50 times, through the "Parameters Variation" type of simulation available on AnyLogic, which is capable of running more than one simulation at the same time, without showing any type of animation. At the end of each iteration, a set of values is printed on an external text file. Table 11 indicates the minimum, maximum and mean obtained values for the performance indicators applied to the current layout, as well as the standard deviation values, for that set of 50 simulation runs. The percentages for the number of people (out of the 6140 total) are indicated between brackets. Some interesting observations and conclusions can already be drawn from these figures.

Table 11. Values for the performance indicators after 50 simulation runs with the current layout

	Min	Max	Mean	Standard deviation
Number of people who have achieved their initial task list:	3754 (61.1 %)	3870 (63.0 %)	3805 (62.0 %)	32.67
Number of passengers who did not embark:	108 (1.8 %)	153 (2.5 %)	135 (2.2 %)	10.80
Number of people who did not achieve Food 1:	172 (2.8 %)	221 (3.6 %)	196 (3.2 %)	12.50
Number of people who did not achieve Food 2:	189 (3.1 %)	243 (4.0 %)	213 (3.5 %)	13.09
Number of people who did not achieve Food 3:	207 (3.4 %)	285 (4.6 %)	245 (4.0 %)	17.49
Number of people who did not achieve Food 4:	230 (3.7 %)	290 (4.7 %)	261 (4.3 %)	13.58
Number of people who did not achieve Shop 1:	283 (4.6 %)	370 (6.0 %)	319 (5.2 %)	16.25
Number of people who did not achieve Shop 2:	476 (7.8 %)	557 (9.1 %)	520 (8.5 %)	18.73
Number of people who did not achieve Shop 3:	651 (10.6 %)	770 (12.5 %)	708 (11.5 %)	25.08
Number of people who did not achieve Shop 4:	731 (11.9 %)	827 (13.5 %)	780 (12.7 %)	23.29
Number of people who did not achieve WC:	551 (9.0 %)	639 (10.4 %)	593 (9.7 %)	23.22
Number of people who did not achieve Cash:	397 (6.5 %)	475 (7.7 %)	439 (7.1 %)	19.16
Number of people who did not achieve Social:	414 (6.7 %)	518 (8.4 %)	458 (7.5 %)	23.79
Number of people who did not achieve Help:	50 (0.8 %)	84 (1.4 %)	69 (1.1 %)	7.64
Number of people who used Restaurant 1:	563 (9.2 %)	664 (10.8 %)	608 (9.9 %)	22.05
Number of people who used Restaurant 2:	544 (8.9 %)	627 (10.2 %)	588 (9.6 %)	22.16
Number of people who used Restaurant 3:	503 (8.2 %)	598 (9.7 %)	553 (9.0 %)	25.09
Number of people who used Restaurant 4:	497 (8.1 %)	579 (9.4 %)	533 (8.7 %)	20.45
Number of people who used Shop 1:	1265 (20.6 %)	1431 (23.3 %)	1360 (22.1 %)	38.04
Number of people who used Shop 2:	1085 (17.7 %)	1241 (20.2 %)	1165 (19.0 %)	27.90
Number of people who used Shop 3:	915 (14.9 %)	1040 (16.9 %)	989 (16.1 %)	30.82
Number of people who used Shop 4:	861 (14.0 %)	989 (16.1 %)	917 (14.9 %)	25.97
Walking distance - minimum value (m):	7.2	16.1	12.1	2.45
Walking distance - maximum value (m):	344.2	486.3	386.5	31.51
Walking distance - mean value (m):	98.8	101.4	100.0	0.54

An average of 3805 people out of the 6140 in total managed to achieve their initial list of tasks, which accounts for 62.0 % of the people. An average of 2.2 % of the people, however, did not embark in their flights.

It is interesting to notice that Restaurant 1 was more used than Restaurant 2, which was more used that Restaurant 3, which was more used than Restaurant 4. The same happens with the number of people who wanted to use these facilities but ended up not having enough time for it: the number of passengers who wanted to eat in Restaurant 1 but did not have the time for it is lower than for Restaurant 2, which is lower than for Restaurant 3, which is lower than for Restaurant 4. The exact same thing happens to the shops: Shop 1 was more used than Shop 2, which was more used than Shop 3, which was more used than Shop 4. It is therefore clear that Restaurant and Shop 1 are somehow more "accessible" than Restaurant and Shop 2, which are more accessible than restaurants and shops 3 and 4, respectively. Visually speaking, these facilities are numbered from left to right, as seen in Figure 29. So, in other words, the further away a restaurant or a shop is from the entry point in the system (after security), the less pedestrians actually use them. The fact that shopping activities tended to remain more undone than restaurant-related tasks is due to the difference in the priority values.

Given the relatively high priority for asking for help, the short period of time it takes for someone to do it and the activity's position on the plan (right after security), it is not surprising that only 69 out of 6140 people (1.12 %) could not manage to fulfil that task. The activities of withdrawing cash and using social connectivity, on the other hand, have the lowest priorities and therefore tended to remain undone, since time is limited. Food and restroom-related activities are the ones with the highest priority. However, more people left the system without having used a restroom (when they needed to) than the ones who wanted to eat/drink but did not manage to. This can probably be explained by two main factors:

- Statistically speaking, more people have the "wc" task in their lists than the "food" task, because the probability for someone to use the restroom was set to be higher than for eating/drinking, so there necessarily is a higher number of "unsatisfied" people regarding that activity;
- As stated previously, the restroom task becomes relevant to the individual approximately one hour after they used the facility for the last time, while that only happens in 1h30 for eating/drinking. Therefore, it is more likely that one ends up boarding with the "wc" task on the list than the "food" task.

Distance-wise:

- The absolute minimum walked distance was 7.2 meters. This was probably a passenger who needed to board at gate 1 and went directly there, either because of a late arrival to the airport or because they had no tasks at all on the list;
- The absolute maximum walked distance, on the other hand, was 486.3 m. This, on the contrary, was probably associated to someone with a long list of tasks and enough time to spend on all (or most) of them;
- The average walked distance was exactly 100.0 m.

Now that the current situation is known, it is time to apply the model to other possible layouts and see how these performance indicators evolve. In other words, a sensitivity analysis of the model needs to be made in the following subchapter, where the changing values will be the facilities' X and Y coordinates.

6.4. Application of the model to other possible layouts of the terminal – Sensitivity analysis

In this section, some experiments will be made with the model, by testing different layouts than the current one. Some of the bullet points highlighted by Crawford & Melewar (2003) need to be kept in mind, in order to provide a good store layout and thus induce browsing. One element that Lisbon T2 has got to its advantage is that its current architecture basically consists of a big open space, which is one of the important factors mentioned by these authors. Their measures that focus on stress and normative traits reduction are not part of the aim of this work.

The general geometry of the terminal shall be maintained, both in terms of external walls and gate positions. Therefore, the position of the main waiting area needs to remain the same. Taking into account that this study focuses on the restricted part of the terminal only, its relative position to the public area shall be maintained as well, which means that passengers will exit security from the same place as previously in the plan of the terminal, that is, from the left side in Figure 29. This implies that the information point, which is basically represented by the staff working just after security, will have its position maintained too, since it also leads to a good performance, as seen previously. What will change in the analysed alternatives are the coordinates of all the other facilities. However, nothing can be put on the right side of the actual position for Shop 4, since that is where the Immigration processes take place.

The first restroom is considered to be well located, since it is very close to security and to the main waiting area as well. Taking into account that 70 % of the people in this model have that need in their task lists and it has the highest priority value, it is good to have it there, so that people go directly there after security in case they need to. Hence, it makes sense that the second restroom is diametrically opposed to the first, so both restrooms will keep their current positions in this chapter. Also, it is considered that shops 1 and 2 shall maintain their current positions, since they have a lower priority than restrooms and restaurants and, therefore, it is predictable that they would be less visited if they were to be moved away from that "main" area of the terminal. Furthermore, their current location allows people who want to eat, withdraw money or use the second restroom to get past these shops and, eventually, see something there that grabs their attention.

Some of the main individual ideas that were tested will be firstly described one by one in the following sections and then some of their combinations will be run as well. Each one of them will be fully detailed in the following subsections:

1. Alternative layout nº 1 consists of an individual change in one of the restaurants' position, so that it would be more closely located to the boarding gates and main waiting area (see Figure 30);
2. Alternative layouts nº 2 and 3 regard the cash machine's position. Both would place it closer to the main waiting area as well, but in different positions (see Figure 31 and Figure 32);
3. Alternative layout nº 4 would change one of the shops' position and put it in the middle of the main waiting area (see Figure 33);
4. The fifth alternative consists of a combination between alternatives nº 3 and 4. In other words, it would place one of the shops in the centre of the main waiting area and pull the cash machine's position closer to that area as well (see Figure 34);
5. The last described alternative is a combination of alternative 5 with another (previously untested) move, which would only become possible thanks to the fact that there would be a free space in one of the shops' current location (due to alternative 4), which could therefore be occupied by one of the restaurants (see Figure 35).

Figure 30. Alternative layout nº 1 – change in Restaurant 4's position

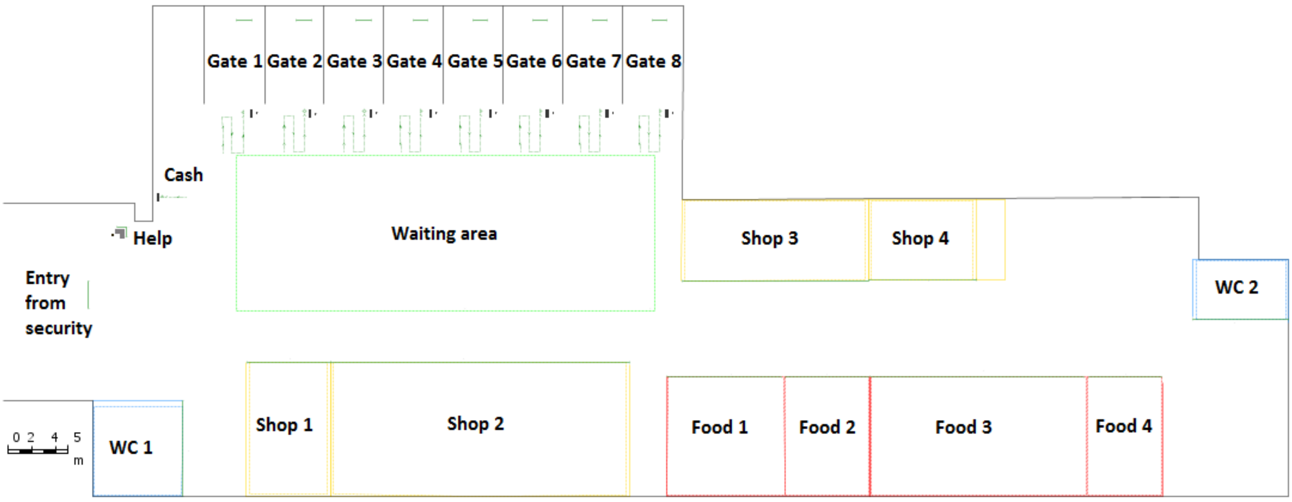


Figure 31. Alternative layout nº 2 – change in the ATM's position

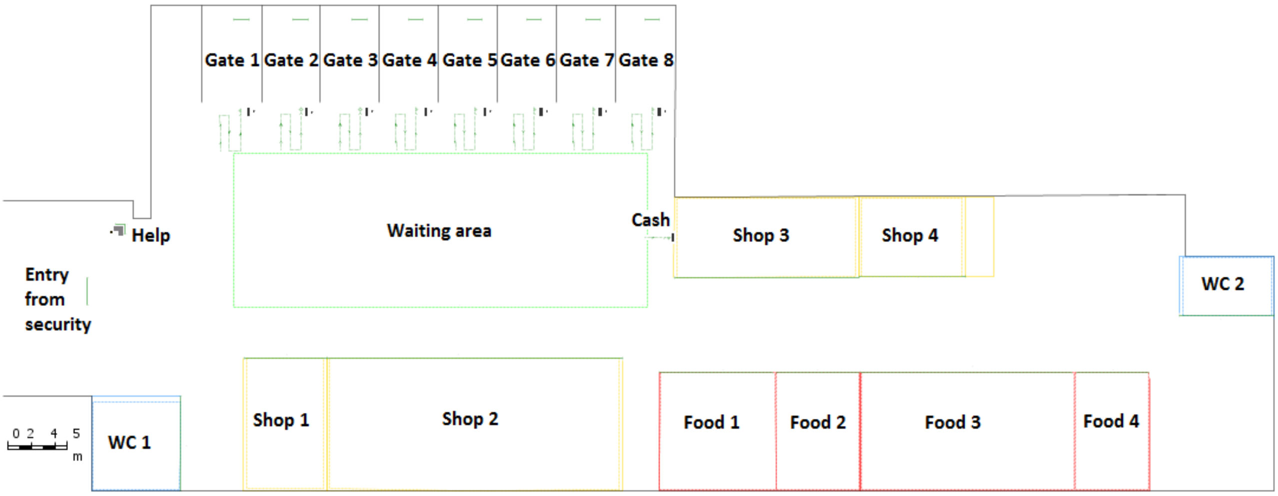


Figure 32. Alternative layout nº 3 – change in the ATM's position

Figure 33. Alternative layout nº 4 – change in Shop 4's position

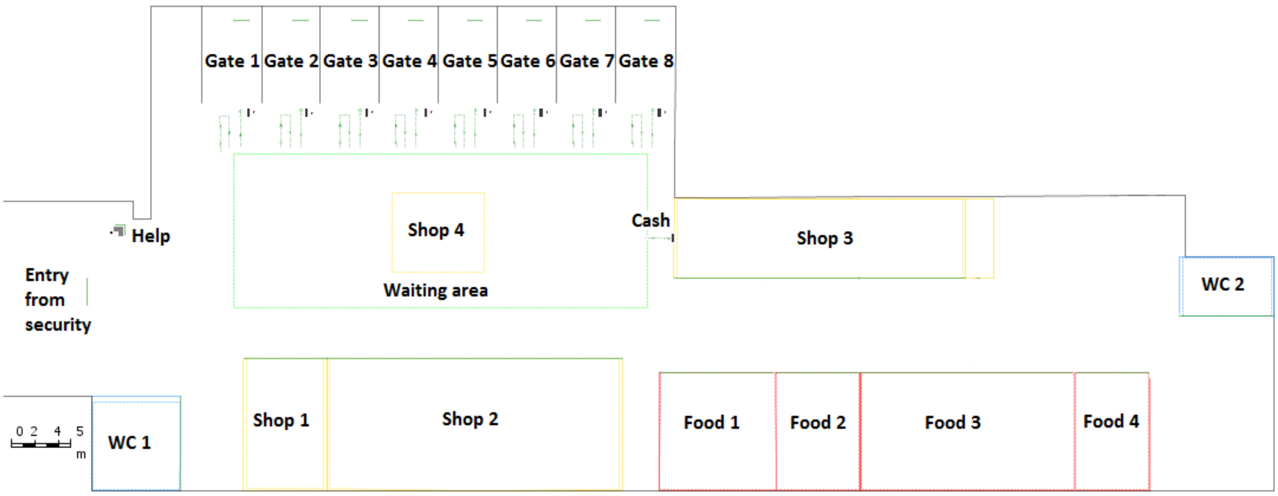


Figure 34. Alternative layout nº 5 – combination of alternatives 3 and 4

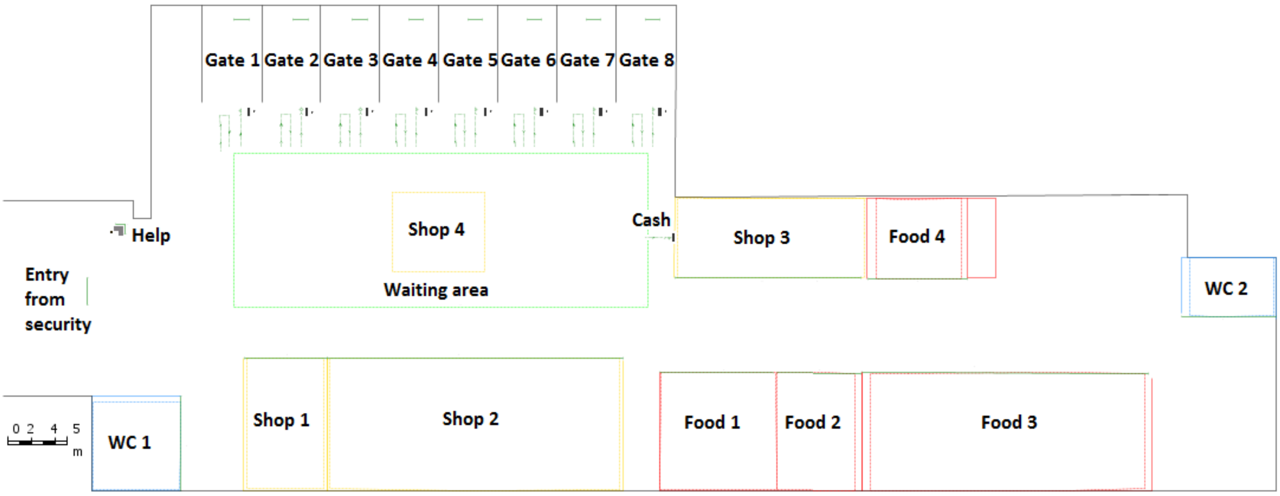


Figure 35. Alternative layout nº 6 – combination of alternative 5 and a new change in Restaurant 4's position

6.4.1. Alternative layout nº 1

Considering the observations that were made for the current layout of the terminal, it was decided that the first thing that could be changed in the layout would be the position of one of the available restaurants, since eating/drinking is a high-priority activity that often remains undone for people who wish to use this type of facility.

After analysing the terminal plan and its physical restrictions, it was decided that Restaurant 4 (Air Tasty 2 Go) could be placed near the boarding gates, as illustrated in Figure 30. Motivation for this choice was that Air Tasty 2 Go is the smallest of the four restaurant spaces and thus the one that could more easily fit in that position, even if its area would need to be (slightly) reduced. By doing this, Restaurant 3 would see its area slightly increased, since it would extend to the space that is currently occupied by Restaurant 4.

Table 12 illustrates the obtained average values after 50 simulation runs for this alternative and compares them to the ones for the current layout. Variation values were calculated for each indicator: positive variations from this research's point of view (that turn the option into a more interesting one according to the reasoning described in chapter 5.6) appear in green and bold characters, while negative variations (from the same scope) appear in blue. Standard deviations and values that would remain the same appear in black, as well as the values for the current layout.

Table 12. Comparison table between the current layout and alternative nº 1

	Mean (current)	Mean (1)	Variation values	Standard Dev. (current)	Standard Dev. (1)
Achieved their initial task list:	3805	3787	- 2.93 ‰*	32.67	35.88
Did not embark:	135	145	+ 1.63 ‰*	10.80	11.80
Did not achieve Food 1:	196	**193**	**- 0.49 ‰***	12.50	12.62
Did not achieve Food 2:	213	215	+ 0.33 ‰*	13.09	17.61
Did not achieve Food 3:	245	**240**	**- 0.81 ‰***	17.49	16.23
Did not achieve Food 4:	261	**42**	**- 35.67 ‰***	13.58	7.56
Did not achieve Shop 1:	319	379	+ 9.77 ‰*	16.25	22.75
Did not achieve Shop 2:	520	570	+ 8.14 ‰*	18.73	24.39
Did not achieve Shop 3:	708	747	+ 6.35 ‰*	25.08	32.60
Did not achieve Shop 4:	780	820	+ 6.51 ‰*	23.29	24.45
Did not achieve WC:	593	738	+ 23.62 ‰*	23.22	25.75
Did not achieve Cash:	439	454	+ 2.44 ‰*	19.16	23.63
Did not achieve Social:	458	514	+ 9.12 ‰*	23.79	20.85
Did not achieve Help:	69	73	+ 0.65 ‰*	7.64	8.91
Used Restaurant 1:	608	**611**	**+ 0.49 ‰***	22.05	23.74
Used Restaurant 2:	588	586	- 0.33 ‰*	22.16	21.85
Used Restaurant 3:	553	**556**	**+ 0.49 ‰***	25.09	22.11
Used Restaurant 4:	533	**804**	**+ 44.14 ‰***	20.45	27.32
Used Shop 1:	1360	1308	- 8.47 ‰*	38.04	30.10
Used Shop 2:	1165	1123	- 6.84 ‰*	27.90	27.51
Used Shop 3:	989	952	- 6.03 ‰*	30.82	26.75
Used Shop 4:	917	883	- 5.54 ‰*	25.97	22.00
Min. walking distance (m):	12.1	**11.8**	**- 2.48 %****	2.45	2.26
Max. walking distance (m):	386.5	**345.0**	**- 10.74 %****	31.51	24.11
Mean walking distance (m):	100.0	**93.2**	**- 6.80 %****	0.54	0.46
* variation values for the proportions out of the 6140 people in total ** variation values from the current state to the proposed alternative					

The most important conclusions that can be taken from this table is that the performances for most of the restaurants basically remain the same (it slightly increases for Restaurants 1 and 3 and it slightly decreases for Restaurant 2, but these variations affect no more than 5 people out of 6140 per day, so they are not considered to be relevant), while for Restaurant 4, which is the most underperforming one in the current situation, would become the most visited and would only have 42 out of 6140 people (less than 0.7 %) without managing to go there.

In terms of total individually walked distance, this option would also be more attractive, since the mean value for it would decrease almost 7 % overall, which makes sense, since a higher number of facilities would be “concentrated” in the same area, closer to the arrival from security, the main waiting area and the boarding gates.

However, this modification would mostly lead to negative aspects: being the food-related activities one of the most time-consuming and the ones with the highest priority value, they would end up delaying the rest of the tasks: all of the other options would become less performant from a passenger’s point of view, in that more people would embark with a lower level of satisfaction because of not having completed them. Actually, the total number of people having achieved their initial tasks would slightly decrease as well. Shops, which have a lower priority value than restaurants, would have less visitors throughout the day because people willing to go to Air Tasty 2 Go would mostly go there before dedicating their times to the rest of the tasks. Finally, the number of people who would miss their flights would increase by 10 people each day (to 3.49 % of the total number of people), which is a non-negligible increase. This alternative is thus rejected as an interesting one.

6.4.2. Alternative layouts nº 2 and 3

The easiest facility to be moved around the terminal is the cash machine. Therefore, two different options are tested regarding its position, both closer to the main waiting area and the boarding gates than in the current layout: in alternative nº 2, the ATM would be placed on the left of the main waiting area, whereas on alternative nº 3 it would be located on the right. These are represented in Figure 31 and Figure 32, respectively.

After running each one of them 50 times, the performance indicators were analysed, in order to define what the best of the options was the best overall. That analysis showed that alternative nº 2 was worse than the current situation in almost every aspect: averagely, slightly less people would not achieve their initial tasks, more people would not embark in time (2.6 % instead of 2.2 %), more people would leave the system with unfinished tasks (except the task which consisted of withdrawing money) and less people would use the available shops and restaurants. The walking distance, however, was lower, with a mean value of 97.7 m instead of 100.0 m, and the utilisation rate for the ATM would be higher.

Alternative nº 3, on the other hand, showed up as a more interesting option: not only more performing than alternative nº 2, but more than the current layout as well, in general. A comparison of results can be seen in Table 13, showing that this alternative for the cash machine’s position would actually lead to better results overall: more people would achieve their task lists, more people would undertake the activity of withdrawing cash and the walking distances would be slightly smaller. The rest of the parameters would not vary in a significant way.

Table 13. Comparison table between the current layout and alternative nº 3

	Mean (current)	Mean (Alt. 3)	Variation values	Standard Dev. (current)	Standard Dev. (Alt. 3)
Achieved their initial task list:	3805	**3815**	**+ 1.63 ‰***	32.67	38.75
Did not embark:	135	**134**	**- 0.16 ‰***	10.80	13.99
Did not achieve Food 1:	196	**194**	**- 0.33 ‰***	12.50	14.52
Did not achieve Food 2:	213	215	+ 0.33 %	13.09	14.32
Did not achieve Food 3:	245	**241**	**- 0.65 ‰***	17.49	14.51
Did not achieve Food 4:	261	**260**	**- 0.16 ‰***	13.58	14.81
Did not achieve Shop 1:	319	**316**	**- 0.49 ‰***	16.25	15.77
Did not achieve Shop 2:	520	**517**	**- 0.49 ‰***	18.73	21.48
Did not achieve Shop 3:	708	**704**	**- 0.65 ‰***	25.08	18.58
Did not achieve Shop 4:	780	789	+ 1.47 ‰*	23.29	23.13
Did not achieve WC:	593	602	+ 1.47 ‰*	23.22	24.32
Did not achieve Cash:	439	**411**	**- 4.56 ‰***	19.16	21.16
Did not achieve Social:	458	461	+ 0.49 ‰*	23.79	18.41
Did not achieve Help:	69	73	+ 0.65 ‰*	7.64	7.35
Used Restaurant 1:	608	**613**	**+ 0.81 ‰***	22.05	24.81
Used Restaurant 2:	588	584	- 0.65 ‰*	22.16	24.46
Used Restaurant 3:	553	553	0.00 ‰	25.09	27.47
Used Restaurant 4:	533	**536**	**+ 0.49‰***	20.45	25.18
Used Shop 1:	1360	**1366**	**+ 0.98 ‰***	38.04	32.52
Used Shop 2:	1165	1165	0.00 ‰	27.90	35.29
Used Shop 3:	989	989	0.00 ‰	30.82	32.67
Used Shop 4:	917	902	- 2.44 ‰*	25.97	26.68
Min. walking distance (m):	12.1	**11.1**	**- 8.26 %****	2.45	2.49
Max. walking distance (m):	386.5	395.1	+ 2.23%**	31.51	24.37
Mean walking distance (m):	100.0	**99.0**	**- 1.00 %****	0.54	0.68

* variation values for the proportions out of the 6140 people in total
** variation values from the current state to the proposed alternative

6.4.3. Alternative layout nº 4

In this alternative layout, the shop that was the furthest away from security would be "pulled" to the middle of the main waiting area, in order to increase space and store penetration, as highlighted by Crawford & Melewar in their work (2003). The idea would be to build a kind of "kiosk" for people in the area and somehow "force" pedestrians to get past that point. This kiosk would be Shop 4, which is the one called Divers in reality (various items are sold there, such as newspapers, magazines, some souvenirs...). Figure 33 illustrates this plan.

As a consequence of this, there would be an empty space next to Shop 3, so it was decided that its area would be increased so that no empty space would be left at the current location for Shop 4. In order to apply this in the model, two reference points are considered for the new Shop 3 rather than one (since its entrance line becomes considerably longer), as it happens for the four referenced points for the main waiting area. Table 14 shows the obtained results.

It is possible to notice, thanks to the figures in that table, that Shop 4 would become much more popular. The number of passengers who would not complete the "social connectivity" task would be reduced as well, probably due to the people who have both tasks on the list ("shop 4" and "social connectivity"), who basically would not need to walk to go from one area to the other. This alternative would also lead to lower values for the walked distances, a higher number of people who would achieve their initial task lists and a slightly inferior number of passengers who would miss their flights (even though it is too small to be considered significant). All the other facilities would get less users, but on a small scale. The most "harmed" facility would be Shop 3, since it would become the one furthest away from security, the main waiting area and the boarding gates.

Table 14. Comparison table between the current layout and alternative nº 4

	Mean (current)	Mean (Alt. 4)	Variation values	Standard Dev. (current)	Standard Dev. (Alt. 4)
Achieved their initial task list:	3805	**3823**	**+ 2.93 ‰***	32.67	36.29
Did not embark:	135	**133**	**- 0.33 ‰***	10.80	12.02
Did not achieve Food 1:	196	213	+ 2.77 ‰*	12.50	15.11
Did not achieve Food 2:	213	235	+ 3.58 ‰*	13.09	14.82
Did not achieve Food 3:	245	261	+ 2.61 ‰*	17.49	16.01
Did not achieve Food 4:	261	276	+ 2.44 ‰*	13.58	15.20
Did not achieve Shop 1:	319	339	+ 3.26 ‰*	16.25	18.50
Did not achieve Shop 2:	520	539	+ 3.09 ‰*	18.73	22.43
Did not achieve Shop 3:	708	757	+ 7.98 ‰*	25.08	23.90
Did not achieve Shop 4:	780	**534**	**- 40.07 ‰***	23.29	24.24
Did not achieve WC:	593	602	+ 1.47 ‰*	23.22	23.09
Did not achieve Cash:	439	453	+ 2.28 ‰*	19.16	20.52
Did not achieve Social:	458	**450**	**- 1.30 ‰***	23.79	21.05
Did not achieve Help:	69	74	+ 0.81 ‰*	7.64	8.75
Used Restaurant 1:	608	595	- 2.12 ‰*	22.05	25.01
Used Restaurant 2:	588	566	- 3.58 ‰*	22.16	21.62
Used Restaurant 3:	553	534	- 3.09 ‰*	25.09	25.50
Used Restaurant 4:	533	521	- 1.95 ‰*	20.45	17.06
Used Shop 1:	1360	1346	- 2.28 ‰*	38.04	32.01
Used Shop 2:	1165	1160	- 0.81 ‰*	27.90	32.27
Used Shop 3:	989	934	- 8.96 ‰*	30.82	22.94
Used Shop 4:	917	**1162**	**+ 39.90 ‰***	25.97	27.58
Min. walking distance (m):	12.1	**12.0**	**- 0.83 %****	2.45	2.34
Max. walking distance (m):	386.5	387.5	+ 0.26%**	31.51	35.76
Mean walking distance (m):	100.0	**96.6**	**- 3.40 %****	0.54	0.51

* variation values for the proportions out of the 6140 people in total
** variation values from the current state to the proposed alternative

This forth alternative is considered to be interesting, since it would bring positive effects on several performance indicators, while it would bring a lot more activity to Shop 4 and it would create a whole new dynamism around the main central area, where passengers necessarily walk through or past at some point of their journeys. Apart from Shop 3, which clearly would become less relevant, it would only affect the other facilities to some small extent.

6.4.4. Alternative layout nº 5

In this proposed alternative, a combination of alternatives 3 and 4 is done in the model in order to see what would happen in that situation, since those were the two most interesting options so far. The tested layout can be seen in Figure 34.

After analysis of the results, it is possible to conclude that this would be an interesting option – overall the best one so far, actually: in average terms, more people would achieve their tasks than in any of the other options, the walking distance would be even more reduced, Shop 4 would be much more visited than it is nowadays and less people would be unable to fulfil the “cash”, “social connectivity” and “shop 4” tasks.

Table 15 shows the mean values for this alternative, as well as the variations for these parameters when compared not only to the current layout, but to the other two best alternatives so far (3 and 4) as well. Green indicates a supporting variation for alternative nº 5, whereas blue indicates less positive aspects for this alternative when compared to the others. In this case, it is noticeable that, apart from the lower performance for Shop 3, the other negative variations are not significant enough to turn this alternative into a worse one overall.

This alternative is thus considered to be interesting as well, since it is the one that brings the higher benefits overall so far, both compared to the current layout and the other two best tested alternatives until now through the developed model.

Table 15. Comparison between alternative nº 5, the current situations and the two other best alternatives so far

	Mean (Alt. 5)	Current **vs.** Alt. 5	Alt. 3 **vs.** Alt. 5	Alt. 4 **vs.** Alt. 5
Achieved their initial task list:	3831	**+ 4.23 ‰***	**+ 2.61 ‰***	**+ 1.30 ‰***
Did not embark:	134	**- 0.16 ‰***	0.00 **‰***	**+ 0.16 ‰***
Did not achieve Food 1:	214	+ 2.93 ‰*	+ 3.26 ‰*	+ 0.16 ‰*
Did not achieve Food 2:	240	+ 4.40 ‰*	+ 4.07 ‰*	+ 0.81 ‰*
Did not achieve Food 3:	260	+ 2.44 ‰*	+ 3.09 ‰*	**- 0.16 ‰***
Did not achieve Food 4:	282	+ 3.42 ‰*	+ 3.58 ‰*	+ 0.98 ‰*
Did not achieve Shop 1:	344	+ 4.07 ‰*	+ 4.56 ‰*	+ 0.81 ‰*
Did not achieve Shop 2:	538	+ 2.93 ‰*	+ 3.42 ‰*	**- 0.16 ‰***
Did not achieve Shop 3:	768	+ 9.77 ‰*	+ 10.42 ‰*	+ 1.79 ‰*
Did not achieve Shop 4:	539	**- 39.25 ‰***	**- 40.72 ‰***	+ 0.81 ‰*
Did not achieve WC:	603	+ 1.63 ‰*	+ 0.16 ‰*	+ 0.16 ‰*
Did not achieve Cash:	422	**- 2.77 ‰***	+ 1.79 ‰*	**- 5.05 ‰***
Did not achieve Social:	449	**- 1.47 ‰***	**- 1.95 ‰***	**- 0.16 ‰***
Did not achieve Help:	70	+ 0.16 ‰*	**- 0.49 ‰***	**- 0.65 ‰***
Used Restaurant 1:	586	- 3.58 ‰*	- 4.40 ‰*	- 1.47 ‰*
Used Restaurant 2:	564	- 3.91 ‰*	- 3.26 ‰*	- 0.33 ‰*
Used Restaurant 3:	538	- 2.44 ‰*	- 2.44 ‰*	**+ 0.65 ‰***
Used Restaurant 4:	513	- 3.26 ‰*	- 3.75 ‰*	- 1.30 ‰*
Used Shop 1:	1346	- 2.28 ‰*	- 3.26 ‰*	0.00 **‰***
Used Shop 2:	1158	- 1.14 ‰*	- 1.14 ‰*	- 0.33 ‰*
Used Shop 3:	938	- 8.31 ‰*	- 8.31 ‰*	**+ 0.65 ‰***
Used Shop 4:	1162	**+ 39.90 ‰***	**+ 42.35 ‰***	0.00 **‰***
Min. walking distance (m):	11.8	**- 2.48 %****	+ 6.31 %**	**- 1.67 %****
Max. walking distance (m):	383.4	**- 0.80 %****	**- 2.96 %****	**- 1.06 %****
Mean walking distance (m):	95.1	**- 4.90 %****	**- 3.94 %****	**- 1.55 %****
* variation values for the proportions out of the 6140 people in total ** variation values from the current state to the proposed alternative				

6.4.5. Alternative layout nº 6

This one is the last alternative layout that is going to be tested in the aim of this dissertation. It essentially consists of a combination of alternative nº 5 and a new change regarding Restaurant 4's current position: instead of having a bigger area for Shop 3, Restaurant 4 would be put at that place and Restaurant 3's area would be increased instead. This would allow to move Restaurant 4 to a slightly closer position to the waiting area, so that space would be more optimised in a way and more people would use it. The plan for this sixth alternative is illustrated in Figure 35.

Once again, the obtained results for this alternative are pretty satisfactory, since they represent not only considerable improvements when compared to the current situation, but also when compared to alternative nº 5, which was considered to be the best one so far. The minimum, maximum and mean values for each performance indicator, as well as the standard deviations, are shown in Table 16.

This table also shows the comparison between the mean values of the current layout, alternative nº 5 and this new alternative.

First of all, it is possible to notice that this alternative is better than any of the others in terms of walking distance reduction; secondly, it is the one where the most people, on average, would finish their initially defined task list; thirdly, it is the alternative with the most significant reduction of people who would not embark their flights; it also solves, in a way, the problem with Shop 3 that would exist in alternatives 4 and 5, probably because people who wish to eat at Restaurant 4 and enter Shop 3 would go more easily from one to the other and, furthermore, the total number of clients for Restaurant 4 would increase (because it would be closer to the entrance in the system, i.e., security).

If this option was to be applied to the terminal geometry, the main results according to the developed model would be:

- No less than 61.5 % of the passengers would fulfil their entire list of tasks at least once;
- No more than 2.7 % of the passengers would miss their flights;
- Less pedestrians would leave the system without having used social connectivity or achieved the activities they wished to at Restaurant 4, the cash machine or Shops 3 and 4;
- Restaurant 4 would have slightly more visitors, while Shop 4 would have many more, thanks to its new location. This would certainly be interesting from a managerial point of view;
- The minimum, maximum and mean values for the total walked distance by the pedestrians would be some meters below their current values.

The other facilities, however, would see their performance indicators get less attractive values, but it is considered that these variations would not be harmful enough to the system to the point of "compensating" all the beneficial improvements. In fact, more passengers would leave the system with a higher level of satisfaction, and that is extremely relevant for an airport terminal too.

Table 16. Results for alternative nº 6 + comparison with the current layout and alternative nº 5

	Min	Max	Mean	Std. Dev.	Current **vs.** Alt. 6	Alt. 5 **vs.** Alt. 6
Achieved their initial task list:	3777	3938	3852	31.86	**+ 7.65 ‰***	**+ 3.42 ‰***
Did not embark:	106	163	129	11.83	**- 0.98 ‰***	**- 0.81 ‰***
Did not achieve Food 1:	187	264	216	14.31	+ 3.26 ‰*	+ 0.33 ‰*
Did not achieve Food 2:	216	271	242	13.48	+ 4.72 ‰*	+ 0.33 ‰*
Did not achieve Food 3:	210	300	265	16.33	+ 3.26 ‰*	+ 0.81 ‰*
Did not achieve Food 4:	230	296	258	16.58	**- 0.49 ‰***	**- 3.91 ‰***
Did not achieve Shop 1:	313	387	351	18.11	+ 5.21 ‰*	+ 1.14 ‰*
Did not achieve Shop 2:	514	594	555	18.8	+ 5.70 ‰*	+ 2.77 ‰*
Did not achieve Shop 3:	648	753	706	26.38	**- 0.33 ‰***	**- 10.10 ‰***
Did not achieve Shop 4:	493	585	550	21.79	**- 37.46 ‰***	+ 1.79 ‰*
Did not achieve WC:	563	654	603	23.26	+ 1.63 ‰*	0.00 **‰***
Did not achieve Cash:	384	488	424	19.8	**- 2.44 ‰***	+ 0.33 ‰*
Did not achieve Social:	420	521	454	18.86	**- 0.65 ‰***	+ 0.81 ‰*
Did not achieve Help:	52	87	71	8.11	+ 0.33 ‰*	+ 0.16 ‰*
Used Restaurant 1:	537	644	587	23.92	- 3.42 ‰*	**+ 0.16 ‰***
Used Restaurant 2:	494	614	563	24.49	- 4.07 ‰*	- 0.16 ‰*
Used Restaurant 3:	465	573	524	22.08	- 4.72 ‰*	- 2.28 ‰*
Used Restaurant 4:	486	589	539	22.13	**+ 0.98 ‰***	**+ 4.23 ‰***
Used Shop 1:	1229	1403	1338	36.04	- 3.58 ‰*	- 1.30 ‰*
Used Shop 2:	1086	1178	1138	23.89	- 4.40 ‰*	- 3.26 ‰*
Used Shop 3:	916	1030	984	25.73	- 0.81 ‰*	**+ 7.49 ‰***
Used Shop 4:	1065	1215	1146	32.28	**+ 37.30 ‰***	- 2.61 ‰*
Min. walking distance (m):	4.6	16	11	2.59	**- 9.09 %****	**- 6.78 %****
Max. walking distance (m):	301.1	400.2	346.7	23.84	**- 10.30 %****	**- 9.57 %****
Mean walking distance (m):	90.5	93.3	91.8	0.55	**- 8.20 %****	**- 3.47 %****
* variation values for the proportions out of the 6140 people in total ** variation values from the current state to the proposed alternative						

6.5. Chapter conclusions

After six different alternatives to the current layout were run and compared, the main observations are the following:

- Alternative nº 1 (Figure 30) was considered not to be interesting, since it would be too harmful for the tasks other than Restaurant 4, less people would catch their flights on time and less people would achieve the tasks they would wish to, although the walking distance would be considerably reduced;
- Alternative nº 2 (Figure 31) was also considered not to be interesting, since it would be worse than the current situation in every aspect, except for the cash activity and the total walking distance;
- Alternative nº 3 (Figure 32), on the other hand, showed up as a positive option for the cash machine's position, since more people would achieve their tasks in general (especially for the task that consists of withdrawing cash), the walking distance would be slightly smaller and the rest of the parameters would not change in a significant way;
- Alternative nº 4 (Figure 33) would also be interesting, since it would lead to a new dynamism in the main waiting area and would improve most of the indicators (especially the ones related to Shop 4); the facilities that would get less users would not be affected in a significant way;
- The combination of alternatives nº 3 and 4 (alternative nº 5, in Figure 34) proved to be positive as well, since more people would achieve their tasks than in any of the previous options, the walking distance would be even more reduced, Shop 4 would still be much more visited than it is nowadays and less people would be unable to fulfil the activities related to cash, social connectivity and Shop 4. Once again, the other options and indicators would only be lightly harmed or changed;
- Finally, alternative nº 6 (Figure 35) would represent a substantial upgrade when compared to both the current situation and all of the other tested alternative layouts: the mean walking distance would be considerably reduced, more people would achieve their tasks, less people would miss their flights and Shop 4 and Restaurant 4 would get more visitors. These improvements are so important that the light reductions in the other facilities' performances are considered not to be relevant enough to the point of counterbalancing all of these benefits.

In fact, the sixth and last alternative appeared as a more performing alternative in almost every way, leading to lower walking distances, more performing activities / facilities, more passengers boarding in time for their flights and more satisfied pedestrians (with a higher number of tasks fulfilled at the end of the day). In order to achieve this, the following alternative modifications would need to be made:

- The cash machine would have to be changed from its current position to a closer one to the boarding gates and main waiting area, where most of the passengers end up spending a considerable amount of time and would rather not move away too much from there. The best tested position would be to put it left to Ale Hop (Shop 3 in the model), in a way that people would queue perpendicularly to that wall. This would lead to a higher utilisation rate for the ATM (as well as for most of the facilities), a higher number of people who would fulfil their task lists, a smaller total walking distance and a slightly lower number of people who would miss their flights;

- The shop Divers (Shop 4 in the model) could be entirely moved to the centre of the main waiting area, so that people would not need to walk from there in order to use it. This would not improve the facility utilisation in such a uniform way as the previous option but it would essentially be beneficial for the "social connectivity" task and, at a much higher scale, the shopping activities related to Divers. The total walking distances and number of people missing their flights would also be reduced, at the same time that the number of people having achieved their initial task list would be higher;
- The two previously described options could also be combined, leading to simultaneous overall improvements regarding these performance factors;

Air Tasty 2 Go (Restaurant 4 in the model) could then be changed from its current position so that it would occupy the area where Divers is at the moment. This, combined with the previous options, would improve the system even more than any of the other options. Comparing to the current situation, a significant increase regarding the number of completed task lists would be achieved, more people would catch their flights in time, the walking distance values would be reduced by 8 meters and no facility would see its utilisation being negatively affected in a significant way; on the contrary – the Air Tasty 2 Go restaurant, the cash machine and the Divers shop would increase their performances (very substantially in the case of the latter).

Table 17 gathers all the main pros and cons of each one of these alternatives, when compared to the current situation.

Table 17. Synthesis table for comparison between the 6 different alternatives and the current situation

Alternative layout nº:	Main Pros:	Main Cons:
1	- Much higher utilisation for Restaurant 4; - Reduction in the walking distance.	- Lower utilisation rates for the activities and facilities other than Restaurant 4; - Lower satisfaction of the passengers, due to a higher number of missed flights and more tasks that would remain unachieved.
2	- Considerable reduction in the number of people who would not withdraw cash; - Reduction in the walking distance.	- Lower utilisation rates for every restaurant and shop; - Increase in the number of missed flights; - Higher number of people without having achieved their tasks other than cash.
3	- Increase in the number of people who would achieve their initially defined task list; - Reduction in the number of people who would not withdraw cash; - Reduction in the walking distance; - More concentrated layout.	- Some facilities would be less used, but not in a considerable way.
4	- Increase in the number of people who would achieve their initially defined task list; - Very important reduction in the number of people who would not visit Shop 4; - Reduction in the walking distance; - More concentrated and interpenetrated layout.	- Non negligible reduction in the number of visitors for Shop 3; - Less users for all the other facilities, but not in a considerable way when compared to all of the important pros.
5	- Combination of the main pros from alternatives nº 3 and 4; - Even more concentrated and interpenetrated layout.	- Non negligible reduction in the number of visitors for Shop 3; - Less users for all the other facilities, but not in a considerable way when compared to all of the important pros.
6	- Main pros from alternative nº 5; - Reduction in the number of missed flights; - Slight reduction in the number of people who would not achieve social connectivity; - Slight increase in the number of people who would use Restaurant 4.	- Less users for all the other facilities, but not in a considerable way when compared to all of the important pros; - The problem with Shop 3 that would arise in options 4 and 5 would become considerably less important.

Chapter 7. Conclusions

The past five chapters aimed to accomplish the objectives and answer the questions that were defined in Chapter 1. It is now time to check what has emerged in discussing those matters and gather the most important conclusions in this final chapter, as well as some recommendations for further research about the subject.

At this point, it is considered that the objectives were met and all the research questions were answered throughout these chapters:

- Chapters 2 and 3 allowed the author to better understand what the airport concept was all about, as well as to get to know several different authors' works and approaches that could be useful in the aim of this work in some way. A better vision of what had already been done was gained and some important concepts that later played a crucial role in the model's conceptual choices and development were picked, such as the most influential individual traits for passenger behaviour and the main tools and approaches that had been used on previous research works. The list of available bibliography is long and a lot has already been done regarding airport modelling, on the one hand, and shopping behaviour, on the other hand, but no scientific documents were found that approached an airport terminal in the same way as the present work does, i.e., by developing a simulation model for passengers' discretionary behaviours in order to test alternative layouts and draw conclusions both from the individuals' and the managing entities' points of view. The used references in these chapters still allowed the author to improve his perception about these matters and understand what main factors and guidelines should be kept in mind when approaching these subjects;
- Chapter 5 thoroughly described the creation, from scratch, of a simulation model that aims to study passengers' discretionary behaviours when inside an airport terminal between security and boarding processes, which was the first main objective of this work. That model was then applied in Chapter 6 to a specific case study, which Chapter 4 had already introduced to the reader. Some performance indicators were defined and the current architectural layout was studied both from an average passenger's and a manager's perspectives (being the latter strongly related to the first), which allowed to fulfil the second main objective of this dissertation. A set of alternatives were subsequently tested through the same set of parameters and some specific observations and conclusions were taken from each of these simulation runs, which was the third major objective of this research.

This methodology allowed the author to observe that some improvements regarding several aspects of the restricted area of the Lisbon T2 could be made: some facilities are less performant than what would be most desirable and the way facilities are disposed in space leads passengers to walk an average of 100.0 m, while an average of 2335 people leave the area without fulfilling their initial list of tasks even once. Some activities tend to remain undone more than others, but a general observation was made that, the further away from security a facility is located, the less people visit it on a daily basis.

After several alternatives were tested and assessed through the same set of performance indicators, it was possible to determine several possible improvements for the current layout. In fact, two modifications in different facilities would already be beneficial on an individual level but, when combined with each other, would lead to even more interesting results. These modifications would be the change of the cash machine's position to a closer location to the main waiting area and the placing of Shop 4 (Divers in the real world) in the middle of that same waiting area. By combining these measures to an extra change in Restaurant 4's position in order to occupy the current area destined to Shop 4, the system would become even more attractive in nearly every aspect, both from a passenger's and a manager's points of view: more people would catch their flights in time, more people would fulfil the tasks they had previously defined, some facilities would increase their utilisation rates considerably and pedestrians would need to walk less in order to induce these effects in the terminal.

The described model could however be improved and lead to more accurate results for the described study. It is important to keep in mind what is probably its biggest limitation: it does not take into account the different brands or product categories that are sold inside each one of these facilities, due to lack of available data about it for the case study. At the moment of a pedestrian's generation, if their personal traits lead them to wanting to eat, the probabilities of generating a task called "food i" or "food j" (with $i \neq j$) are equal, and the same happens with shopping activities. Therefore, no distinction is made between facilities of the same category, apart from the spatial position of each. In the case of the alternative layout nº 1, for instance, such a distinction would be important, since Air Tasty 2 Go sells more "portable" edible and drinkable options than the other three, allowing the passengers to purchase something there a few minutes before boarding if they wish to do so, or in case they are waiting for gate control in the main waiting area and become hungry again. Impulse purchases would probably end up increasing, which is not really taken into account in the developed model. This type of strategic reasoning regarding the facilities' positioning was considered one of the five key success factors for airport retailing by Bamberger et al. (2009).

In fact, no impulse shopping tendencies are modelled, since people have their lists of tasks defined at the same moment they are generated and they do not change from that moment on. The only way "impulse" shopping is considered is through the different facilities' positions, leading to a higher utilisation rate for some spaces than others, since one might decide to go somewhere and leave the other options on the task list for later because of their relative geometrical positions. This impulse shopping concept was also included among the performance indicators behind the assumption that, the more people visit the shops, the more likely it is that the sales increase overall, because of the people who spontaneously decide to purchase something they had not previously planned. Still, it was not directly integrated in the model.

Also, some imprecisions need to be kept in mind regarding this model, even though they do not distort the results much: distances are calculated referring to specific isolated points in space and they are always calculated in a straight line. In this specific case study's geometry, these do not really make a difference in the end, since it is basically a wide and wall-free space, so the way pedestrians move between different facilities does not differ too much from a straight line. In other spaces, however, it would be necessary to consider different paths for each pedestrian (depending on their flight's boarding gate, for example). Also, from an airport logistics point of view, these simulations were made considering that there were no problems at all, i.e., no delays. Such elements could be included in future research in this field.

Another limitation of the model is the propensity function: this concept was invented by the author. In future research, it might be interesting to create a utility function and calibrate it through specific surveys, such as the ones done by Denver International Airport (2009) but applied to this specific case study. This calibration could be done by trying to include various combinations of parameters in the utility function and, for each combination, see how the mean and the t-statistic values for the coefficients of the explanatory variables would vary and whether the predicted results would be in accordance with the obtained results in the survey. Eventually, a mixed logit model could be used instead of a MNL model: that type of model, although more complicated that the MNL, allows the user to define some variables as random (by associating them to a chosen distribution) and predicts the possibility of introducing covariance between parameters. In this case, a mixed logit would make it possible to introduce correlation between, for instance, the way pedestrians would perceive a "distance" variable and another called "remaining time to departure", and would allow to transform a variable such as "cost" into random, since not every person reacts the same way to that variable. In other words, this problem could have been approached in several different ways in terms of individual choice modelling options, but that would have ended up deviating too much from the dissertation's main objectives. It could however be interesting to approach the problem differently in terms of individual discrete choice processes in the future.

More specific conclusions to the case study can also be drawn by future researchers if other aforementioned aspects are kept in mind, namely if differences between facilities of the same type are considered and changes in positions are made while taking them into account, depending on the type of products and overall success each facility has among the passengers in the current situation. Some surveys can be made in future works in order to understand what the passengers' specific preferences are, depending on their basic traits. New statistical information can also be obtained in order to integrate impulse behaviour in the model and, eventually, make some profit calculations (in monetary terms).

To summarize: results can become more accurate and complete in the future if more data are added. In spite of these limitations, the developed model is considered as being valid and interesting from a managerial perspective, since it leads to some important conclusions regarding spatial configuration of an airport terminal, while integrating a logical behavioural model for passengers that takes into account their (airport-specific or not) basic traits and relates them to the different discretionary activities inside such a space. The developed work allowed the author to detect some problems in the current state of the building and suggest alternative layouts that would somehow improve it as a whole. Thanks to the simulation capacities, all of this was tested and analysed through a screen, without having to make a single physical experiment on the field.

References

ACI, A. C. I.-N. A. (2013). *Airports Flying High on Non-Aeronautical Passenger Revenue - 2013 ACI-NA Concessions Benchmarking Survey Results for CY2012*.

ANA, A. de P. (2014). *Annual Report 2014*.

ANA, A. de P. (2015a). ANA Aeroportos de Portugal, Departures, Flight information; T2. Retrieved September 3, 2015, from http://www.ana.pt/en-US/Aeroportos/lisboa/Lisboa/Departures/FlightInformation/Pages/FlightInformation.aspx?Day=-1&sTerminal=Terminal+2

ANA, A. de P. (2015b). ANA Airports Official Website. Retrieved August 6, 2015, from http://www.ana.pt/en-US/Aeroportos/lisboa/Lisboa

ANA, A. de P. (2015c). ANA RouteLab. Retrieved August 12, 2015, from http://routelab.ana.pt/en-US/Airports/Lisbon/Statistics/Pages/default.aspx

Andreatta, G., Brunetta, L., & Righi, L. (2007). Evaluating terminal management performances using SLAM: The case of Athens International Airport. *Computers and Operations Research*, *34*(6), 1532–1550.

AnyLogic. (2015). AnyLogic Official Webpage. Retrieved April 5, 2015, from http://www.anylogic.com

Ballis, A. (2002). Simulation of airport terminal facilities in the greek airports of Kavala and Alexandroupolis. *Operational Research*.

Bamberger, V., Bettati, A., Hoeffinger, S., Kuruvilla, T., & Wille, V. (2009). *Mastering Airport Retail - Roadmap to new industry standards*.

Bandara, S., & Wirasinghe, S. C. (1992). Walking distance minimization for airport terminal configurations. *Transportation Research Part A: Policy and Practice*.

Behdani, B. (2012). *Evaluation of paradigms for modeling supply chains as complex socio-technical systems*.

Chebat, J. C., Michon, R., Haj-Salem, N., & Oliveira, S. (2014). The effects of mall renovation on shopping values, satisfaction and spending behaviour. *Journal of Retailing and Consumer Services*, *21*(4), 610–618.

Cheng, L. (2014). *Modelling airport passenger group dynamics using an agent-based method*. Queensland University of Technology.

Chung, Y. S., Wu, C. L., & Chiang, W. E. (2013). Air passengers' shopping motivation and information seeking behaviour. *Journal of Air Transport Management*, *27*, 25–28.

Churchill, A., Dada, E., de Barros, A. G., & Wirasinghe, S. C. (2008). Quantifying and validating measures of airport terminal wayfinding. *Journal of Air Transport Management*, *14*(3), 151–158.

Crawford, G., & Melewar, T. C. (2003). The importance of impulse purchasing behaviour in the international airport environment. *Journal of Consumer Behaviour*, *3*(1), 85–98.

Curcio, D., Longo, F., Mirabelli, G., & Pappoff, E. (2007). Passengers' flow analysis and security issues in airport terminals using modeling and simulation. *21st European Conference on Modelling and Simulation*, 1–7.

Denver International Airport. (2009). *Denver International Airport Travel Passenger Study Results*.

Doganis, R. (2005). *The airport business*. *Journal of Air Transport Management* (Vol. 1). Routledge.

El Hedhli, K., Chebat, J. C., & Sirgy, M. J. (2013). Shopping well-being at the mall: Construct, antecedents, and consequences. *Journal of Business Research*, *66*(7), 856–863.

FlightAware. (2015). FlightAware Website. Retrieved September 3, 2015, from http://flightaware.com

Fonseca i Casas, P., Casanovas, J., & Ferran, X. (2014). Passenger flow simulation in a hub airport: An application to the Barcelona International Airport. *Simulation Modelling Practice and Theory*, *44*, 78–94.

Freathy, P., & O'Connell, F. (2012). Spending time, spending money: passenger segmentation in an international airport. *The International Review of Retail, Distribution and Consumer Research*, *22*(4), 397–416.

Geuens, M., Vantomme, D., & Brengman, M. (2004). Developing a typology of airport shoppers. *Tourism Management*, *25*(5), 615–622.

Grigoryev, I. (2015). *AnyLogic 7 in three days - a quick course in simulation modeling* (Second edi).

Hampson, D. P., & McGoldrick, P. J. (2013). A typology of adaptive shopping patterns in recession. *Journal of Business Research*, *66*(7), 831–838.

Kalakou, S. (2012). *Performance Evaluation of Passenger-related Processes at an Airport with a Simulation Model*. Instituto Superior Técnico, Universidade de Lisboa.

Kalakou, S., Bierlaire, M., & Moura, F. (2014). Effects of terminal planning on passenger choices. In *14th Swiss Transport Research Conference (STRC)*.

Lin, Y. H., & Chen, C. F. (2013, June). Passengers' shopping motivations and commercial activities at airports - The moderating effects of time pressure and impulse buying tendency. *Tourism Management*, pp. 426–434. Elsevier.

Liu, X., Usher, J. M., & Strawderman, L. (2014). An analysis of activity scheduling behavior of airport travelers. *Computers and Industrial Engineering*, *74*(1), 208–218.

Livingstone, A., Popovic, V., Kraal, B., & Kirk, P. J. (2012). Understanding the Airport Passenger Landside Retail Experience. *DRS 2012 Bangkok–Research: Uncertainty, Contradiction and Value*, *1*.

Lu, J. L. (2014). Investigating factors that influence passengers' shopping intentions at airports - Evidence from Taiwan. *Journal of Air Transport Management*, *35*, 72–77.

Ma, W. (2013). *Agent-based model of passenger flows in airport terminals*. Queensland University of Technology.

Mason. (2015). Mason Official Webpage. Retrieved April 5, 2015, from http://cs.gmu.edu/~eclab/projects/mason/

Odoni, A. R., & de Neufville, R. (1992). Passenger terminal design. *Transportation Research Part A: Policy and Practice*.

Perng, S. W., Chow, C. C., & Liao, W. C. (2010). Analysis of shopping preference and satisfaction with airport retailing products. *Journal of Air Transport Management*, *16*(5), 279–283.

PlaneSpotters. (2015). PlaneSpotters Website. Retrieved September 3, 2015, from http://www.planespotters.net

Reis, V. (2010). *Development of cargo business in combination airlines: strategy and instrument*. Technical University of Lisbon.

Roanes-Lozano, E., Laita, L. M., & Roanes-Macías, E. (2004). An accelerated-time simulation of departing passengers' flow in airport terminals. In *Mathematics and Computers in Simulation* (Vol. 67, pp. 163–172).

Schultz, M., Schulz, C., & Fricke, H. (2010). Passenger dynamics at airport terminal environment. In *Pedestrian and Evacuation Dynamics 2008* (pp. 381–396). Springer Berlin Heidelberg.

Smits, D., Visser, A., & Groen, F. C. a. (2013). Modeling pedestrians in an airport scenario with a time-augmented Petri net. *Intelligent Autonomous Systems*, *12*, 543–551.

Sourd, F., Talotte, C., Constans-Brugeais, Y., Pillon, A., & Donikian, S. (2011). Modelling of pedestrian flows during dwelling: development of a simulator to evaluate rolling stock and platform flow performance. In *WCRR Lille 2011*.

Tam, M. L. (2011). An optimization model for wayfinding problems in terminal building. *Journal of Air Transport Management*, *17*(2), 74–79.

Tošić, V. (1992). A review of airport passenger terminal operations analysis and modelling. *Transportation Research Part A: Policy and Practice*, *26*(1), 3–26.

Train, K. E. (2003). *Discrete choice methods with simulation* (2nd ed.). Cambridge.

Turismo de Lisboa, O. (2014). *Perfil do passageiro low-cost de Lisboa - Verão 2014*.

Van Oel, C. J., & Van den Berkhof, F. W. D. (2013). Consumer preferences in the design of airport passenger areas. *Journal of Environmental Psychology*, *36*, 280–290.

Vizzari, G., Manenti, L., Ohtsuka, K., & Shimura, K. (2014). An Agent-Based Pedestrian and Group Dynamics Model Applied to Experimental and Real World Scenarios. *Journal of Intelligent Transportation Systems (ahead-of-Print)*, (March 2014), 1–13.

Wesley, S., LeHew, M., & Woodside, A. G. (2006). Consumer decision-making styles and mall shopping behavior: Building theory using exploratory data analysis and the comparative method. *Journal of Business Research*, *59*(5), 535–548.

Yuksel, A. (2004). Shopping experience evaluation: A case of domestic and international visitors. *Tourism Management*, *25*(6), 751–759.

Zenglein, M. J., & Jürgen, M. (2007). Non-aviation revenue in airport business - Evaluating performance measurement for a changing value proposition. *Performance Measurement Paper*.

Appendices

Appendix 1. Individual-related data

Tables taken from Appendix C of Ma (2013):

Willingness to ask for help				
Age	< 65		> 65	
Frequency of travel	< 3	> 2	< 3	> 2
Prob(true)	0.9	0.0	1.0	0.4

Desire to Shop								
Gender	Male				Female			
Nationality	Native		Foreigner		Native		Foreigner	
Frequency of travel	< 3	> 2	< 3	> 2	< 3	> 2	< 3	> 2
Prob(true)	0.4	0.5	0.2	0.5	1.0	0.9	1.0	1.0

Social connectivity				
Age	< 65		> 65	
Travel class	Economy	Business	Economy	Business
Prob(true)	0.8	1.0	0.0	0.2

Use a cash machine				
Nationality	Native		Foreigner	
Frequency of travel	< 3	> 2	< 3	> 2
Prob(true)	0.1	0.5	0.1	0.2

Hunger and desire of food	
Probability	0.5

Percentages for the frequency of travel associates to low-cost passengers inside Lisbon airport, depending on the purpose of the trip (leisure or business) (Turismo de Lisboa, 2014):

Nº of flights in the last 12 months	Leisure pax.	Leisure pax. (acum.)	Business pax.	Business pax. (accum.)
1	31.70%	31.70%	0.30%	0.30%
2	22.40%	54.10%	0.70%	1.00%
3	12.50%	66.60%	0.60%	1.60%
4	8.60%	75.20%	1.10%	2.70%
5	4.60%	79.80%	0.90%	3.60%
6 to 10	6.70%	86.50%	4.30%	7.90%
11 to 20	1.39%	87.89%	2.60%	10.50%
20+	0.30%	88.19%	1.50%	12.00%

Appendix 2. Elements created in the simulation model

Necessary parameters in order to define the Flight object in the model:

Name	Description	Type
gate	Gate where the boarding process will take place	Integer
notation	Flight reference (e.g.: AB1234)	String
passengersQuantity	Number of passengers that are predicted to board the flight	Integer
time	Scheduled departure time	Double

Necessary parameters in order to define the Task object in the model:

Name	Description	Type
id	Facility index	Integer
type	Type of task	TaskType

Necessary elements in order to define the PaxArr object distribution for a given flight in the model:

Name	Description	Type
calculatePaxArrTimes	Calculates an arrival distribution for each flight	Function
flight	Scheduled flight	Flight
injectPax	Injects passengers in the system with the distribution defined by *calculatePaxArrTimes*	Event
interArrTimes	Auxiliary collection for the *calculatePaxArrTimes* function to calculate time intervals	Double

Necessary elements in order to define the Passenger agent in the model:

Name	Description	Type
ageClass	Age class	Integer
ageGroupsBoolean	Collection of elements with "false" as their initial values. After the passenger is submitted to the function that assigns them to their characteristics, the index correspondent to their age class becomes "true" and the rest of the slots remain as "false"	Boolean
ageGroupsProbabilities	Collection of probabilities for a given individual to belong to a certain age class. Each slot corresponds to one age class and its probability	Double
business	Travel class	Boolean
chosenTask	Defines the chosen task at each moment	Integer
distanceWalked	Accounts for the sum of the distances each passenger has needed to walk since their arrival from security until a given moment. This counting lasts until the moment they reach the boarding gate	Double
finishedTasksAtLeastOnce	It has 0 as an initial value and becomes 1 when tasks are fulfilled for the first time. This can be seen among the code lines for the functions *generateTasks* and *recalculatePropensities* in Appendix 3	Integer
flight	Passenger's flight	Flight
foreigner	Nationality	Boolean
freqTravel	Frequency of travel in a year	Integer
goesToWaitingArea	Switches from 0 to 1 whenever the passenger has fulfilled all the tasks on the list and goes to the waiting area	Integer
male	Gender	Boolean
recalculatePriorities	Cyclic event that happens every 10 minutes from the moment when the pedestrian is generated and checks if the priority values are under their normal values. If one of them is, it increases it with the respective increment.	Event

Necessary elements in order to define the Passenger agent in the model (cont.):

Name	Description	Type
selectedIndex	Merely auxiliary variable in order to define the indexes of the shops that enter the task list of an individual, since there might be more than one in the terminal model and it would not make sense for a person to visit the same shop twice during dwell time	Integer
TaskChoice	Random number between 0 and 1 that determines which one is the chosen task	Double
tasks	Passenger's individual task list (collection)	Task
wcPriority *shopPriority* *foodPriority* *helpPriority* *socialPriority* *cashPriority*	Priority values for each type of task	Double
foodPriorityIncrement *wcPriorityIncrement*	Increment values for hunger/thirst and need to use a restroom	Double
WCProb *ShopProb* *FoodProb* *HelpProb* *SocialProb* *CashProb*	Probabilities for each possible task at each moment. In case there is more than one facility for a given type, the corresponding variable's name will end with an index number	Double
wcProbability *shopProbability* *foodProbability* *helpProbability* *socialProbability* *cashProbability*	Probabilities for each type of task to make it into the passenger's task list. Values defined in Appendix 1	Double
propensityWC *propensityShop* *propensityFood* *propensityHelp* *propensitySocial* *propensityCash*,	Propensity values for each possible task at each moment. In case there is more than one facility for a given type, the corresponding variable's name will end with an index number	Double
taskWC *taskShop* *taskFood* *taskHelp* *taskSocial* *taskCash*.	Auxiliary variables in order to know what types of tasks are on the initially generated task list for each passenger	Double

Necessary elements in order to define the Main working space in the model:

Name	Description	Type
activeFlights	Collection that contains the active flights at a given moment	Flight
arrIntensitiesTimeInterval	Collection that gathers the three values for different intensities' time periods	Double
calculateDistances	Calculates the distance between the pedestrian's current location and a given pair of coordinates X and Y	Function
Departure	Takes a departed flight out of the list	Dynamic event
FlightBoarding	Starts and finishes boarding events at the scheduled times	Dynamic event
gateControlShapes	Collection of services with lines that assembles the queue lines and services for gate control. It is needed because it gathers all the lines in the same entity and can be easily called at the population of agents "gates […]"	ServiceWLine

Necessary elements in order to define the Main working space in the model (cont.):

Name	Description	Type
gateLines	Collection of target lines that represents the lines across which passengers disappear at the moment they leave the terminal before boarding the aircraft. It is useful for the same reason as the *gateControlShapes* array list	TargetLine
generateShopIndex	Algorithm that uses the “case” formulation in its body in order not to repeat the same index twice when the pedestrian wishes to visit more than one shop	Function
generateTasks	Generate a task list from the passenger’s basic traits	Function
readScheduleFromFile	Reads schedule from file: associates, for each row, one flight. The first column corresponds to the flight references; the second one is the passenger quantities; the third one is the departure times and the last one indicates the corresponding boarding gates.	Function
recalculatePropensities	Recalculate propensity values after a given task is fulfilled	Function
setBoardingTimes	creates, for each flight in the schedule, boarding and departure events	Function
setIntensitiesInterval	Sets the time intervals for different passenger generation intensities	Function
setupPassenger	Assign basic traits to the passenger	Function
timeFromBoardingFinishToDeparture	Time period between the end of the boarding process and the scheduled departure time	Double
timeFromBoardingStartToDeparture	Period of time between boarding starts and scheduled departure time	Double
timeFromCheckInOpenToDep	Time between the opening of the check-in process and the flight departure time	Double
timeFromCheckInStartToHighDensArrFinish	Period of time that the open state of the check-in and the high density period have in common	Double
timeFromFinishPaxArrToDep	Time interval between the arrival of the last person and the flight departure	Double
timeFromHighDensArrStartToCheckInStart	Time between the beginning of the high density arrival period and the moment when check-in starts	Double
timeFromStartPaxArrToDep	Time between the arrival of the first person and the flight departure	Double
shopEntrances *foodEntrances*	Collections of entrances to the shops and restaurants. They are useful in the *generateShopIndex* and *generateTasks* functions in order to define the shop and restaurant indexes	TargetLine
WCX *WCY* *ShopX* *ShopY* *Etc.*	X and Y coordinates for each facility	Double

Appendix 3. Code lines in the simulation model

PaxArr agent:

- ***calculatePaxArrTimes* function:**

```
int lowIntPax = flight.passengersQuantity / 4; // 25 %
int highIntPax = flight.passengersQuantity - 2 * lowIntPax; // 50 %

double[] lastArrInInterval = new double [3];
double[] timeScaleForInterval = new double [3];

// generate random arrival times for three types of arrival intensity

for (int i = 0; i < lowIntPax; i++)
{
        double nextArr = exponential(1);
        interArrTimes.add(nextArr);
        lastArrInInterval[0] += nextArr;
}

for (int i = 0; i < highIntPax; i++)
{
        double nextArr = exponential(1);
        interArrTimes.add(nextArr);
        lastArrInInterval[1] += nextArr;
}

for (int i = 0; i < lowIntPax; i++)
{
        double nextArr = exponential(1);
        interArrTimes.add(nextArr);
        lastArrInInterval[2] += nextArr;
}

//Calculate necessary time scale

for (int i = 0; i < 3; i++)
{
        timeScaleForInterval[i] = main.arrIntensitiesTimeInterval[i] / lastArrInInterval[i];
}

//Scale arrival times

for (int i = 0; i < lowIntPax; i++)
{
        interArrTimes.set(i, interArrTimes.get(i) * timeScaleForInterval[0]);
}

for (int i = lowIntPax; i < highIntPax + lowIntPax; i++)
{
        interArrTimes.set(i, interArrTimes.get(i) * timeScaleForInterval[1]);
}



for (int i = highIntPax + lowIntPax; i < flight.passengersQuantity ; i++)
{
        interArrTimes.set(i, interArrTimes.get(i) * timeScaleForInterval[2]);
```

```
}

//Define arrival event
if (interArrTimes.size() > 0)
{
	injectPax.restart(flight.time - main.timeFromStartPaxArrToDep + interArrTimes.remove(0));
}
```

- ***injectPax* event:**

```
generatePax.inject(1);

if (interArrTimes.size() > 0)
{
	injectPax.restart(max(0, interArrTimes.remove(0)));
}
```

Main agent

- ***notEnoughTime* selectOutput – condition:**

```
entity.flight.time - time() < timeFromBoardingStartToDeparture
```

- ***noMoreTasks* selectOutput – condition:**

```
(entity.propensityWC1 == -100000
&& entity.propensityWC2 == -100000
&& entity.propensityShop1 == -100000
&& entity.propensityShop2 == -100000
&& entity.propensityShop3 == -100000
&& entity.propensityShop4 == -100000
&& entity.propensityFood1 == -100000
&& entity.propensityFood2 == -100000
&& entity.propensityFood3 == -100000
&& entity.propensityFood4 == -100000
&& entity.propensityHelp == -100000
&& entity.propensitySocial == -100000
&& entity.propensityCash == -100000)
```

- ***readScheduleFromFile* function:**

```
while(ScheduleFile.canReadMore())
{
	Flight flight = new Flight();
	flight.notation = ScheduleFile.readString();
	flight.passengersQuantity = ScheduleFile.readInt();
	flight.time = ScheduleFile.readDouble();
	flight.gate = ScheduleFile.readInt();
	flights.add(flight);

}
```

- ***setIntensitiesInterval* function:**

```
arrIntensitiesTimeInterval[0] = timeFromStartPaxArrToDep - timeFromCheckInOpenToDep -
timeFromHighDensArrStartToCheckInStart;

arrIntensitiesTimeInterval[1] = timeFromHighDensArrStartToCheckInStart +
timeFromCheckInStartToHighDensArrFinish;

arrIntensitiesTimeInterval[2] = timeFromCheckInOpenToDep - timeFromFinishPaxArrToDep -
timeFromCheckInStartToHighDensArrFinish;
```

- ***setBoardingTimes* function:**

```
for (Flight flight : flights)
{
        create_FlightBoarding(flight.time - timeFromBoardingStartToDeparture, flight, true);
        create_FlightBoarding(flight.time - timeFromBoardingFinishToDeparture, flight, false);
        create_Departure(flight.time, flight);
}
```

- ***FlightBoarding* event** – this event requires two parameters in order to work: one called *flight*, which type is Flight, and one boolean called *boardingIsTakingPlace*:

```
if (boardingIsTakingPlace)
{
        Gate gate = gates.get(flight.gate - 1);
        gate.flight = flight;
        activeFlights.remove(flight);
        activeFlights.add(0, flight);
}

else

{
        Gate gate = gates.get(flight.gate - 1);
        gate.gateControl.cancelAll();
        gate.flight = null;
        activeFlights.remove(flight);

}
```

- ***Departure* event** – this event requires one parameter in order to work. Its type is Flight and it is called *flight*:

```
flights.remove(flight);
```

- ***calculateDistances* function** – this function requires three arguments in order to work: one double called *destinationX*, one double called *destinationY* and one Passenger called *passenger*:

```
// Distance calculator:

class Location // creates a Location object with given coordinates
{
   Location( double xcoord, double ycoord )
   {
      x = xcoord;
      y = ycoord;
   }
```

```
    double x; // x coordinate of the location
    double y; // y coordinate of the location

    // calculates distance from this location to another one:
    double distanceTo( Location other )
    {
        double dx = other.x - x;
        double dy = other.y - y;
        return sqrt( dx*dx + dy*dy ) / 40; // dimensions were drawn so that
                                           // 1 m = 40 pixels (roughly)
    }

}

// Create first location (origin) == current location of the pedestrian

Location origin = new Location(passenger.getX(), passenger.getY());

// Calculate the desired distance to the destination location

Location destination = new Location (destinationX, destinationY);

double distance = origin.distanceTo(destination);

return distance;
```

- ***generateShopIndex* function** – this function requires one argument of type Passenger, called *passenger*, in order to work:

```
boolean indexFound = false;
int shopIndex = 0;

while (indexFound == false)
{
        shopIndex = uniform_discr(0, shopEntrances.size() - 1);
        switch (shopIndex)
                {
                case 0: if (passenger.propensityShop1 == -100000) {indexFound = true;}
                        break;
                case 1: if (passenger.propensityShop2 == -100000) {indexFound = true;}
                        break;
                case 2: if (passenger.propensityShop3 == -100000) {indexFound = true;}
                        break;
                case 3: if (passenger.propensityShop4 == -100000) {indexFound = true;}
                        break;

                }
}

return shopIndex;
```

- ***setupPassenger* function** – this function requires one argument of type Passenger, called *passenger*, in order to work. It uses data from Appendix 4:

```
// Business passenger?
passenger.business = randomTrue(0.133);
```

```
// Male Passenger?
passenger.male = randomTrue(0.714);

// Frequency of Travel?
if (passenger.business == false)
{
	double numberOfTravels = uniform(0, 0.8819);

	if (numberOfTravels <= 0.317)
	{
	passenger.freqTravel = 1;
	}

	else if (numberOfTravels <= 0.541)
	{
	passenger.freqTravel = 2;
	}

	else if (numberOfTravels <= 0.666)
	{
	passenger.freqTravel = 3;
	}

	else if (numberOfTravels <= 0.752)
	{
	passenger.freqTravel = 4;
	}

	else if (numberOfTravels <= 0.798)
	{
	passenger.freqTravel = 5;
	}

	else if (numberOfTravels <= 0.865)
	{
	passenger.freqTravel = 8; // it is assumed that "between 6 and 10" is 8
	}

	else if (numberOfTravels <= 0.8789)
	{
	passenger.freqTravel = 15; // it is assumed that "between 10 and 20" is 15
	}

	else if (numberOfTravels <= 0.8819)
	{
	passenger.freqTravel = 21; // it is assumed that "20+" is 21
	}
}

else if (passenger.business == true)
{
	double numberOfTravels = uniform(0, 0.12);

	if (numberOfTravels <= 0.003)
	{
	passenger.freqTravel = 1;
	}

	else if (numberOfTravels <= 0.01)
```

```
        {
        passenger.freqTravel = 2;
        }

        else if (numberOfTravels <= 0.016)
        {
        passenger.freqTravel = 3;
        }

        else if (numberOfTravels <= 0.027)
        {
        passenger.freqTravel = 4;
        }

        else if (numberOfTravels <= 0.036)
        {
        passenger.freqTravel = 5;
        }

        else if (numberOfTravels <= 0.079)
        {
        passenger.freqTravel = 8; // it is assumed that "between 6 and 10" is 8
        }

        else if (numberOfTravels <= 0.105)
        {
        passenger.freqTravel = 15; // it is assumed that "between 10 and 20" is 15
        }

        else if (numberOfTravels <= 0.12)
        {
        passenger.freqTravel = 21; // it is assumed that "20+" is 21
        }
}


// Foreigner passenger?
passenger.foreigner = randomTrue(0.769);

// Age Groups:

boolean classNotFound = true;
boolean belongsToClass = false;
int ageClassDetermination = 0;
while(classNotFound == true)
{
        belongsToClass =
randomTrue(passenger.ageGroupsProbabilities.get(ageClassDetermination));
        if (belongsToClass == true)
        {
                classNotFound = false;
        }
        else
        {
ageClassDetermination = (++ageClassDetermination) % passenger.ageGroupsBoolean.size();
// modular division = remainder of (ageClassDetermination + 1) by
passenger.ageGroupsBoolean.size(), which will necessarily be between 0 and
passenger.ageGroupsBoolean.size() - 1
        }
}
```

```
passenger.ageGroupsBoolean.set(ageClassDetermination, true);
// indicates that the person actually belongs to the age class ageClassDetermination
// all of the other classes remain equal to the initial value, which is "false"

passenger.ageClass = ageClassDetermination;
```

- ***generateTasks* function** – this function requires one argument of type Passenger, called *passenger*, in order to work:

```
// Generates the task list for each passenger, based on their basic traits.
// Also calculates the initial distances and propensities for each possible activity

// WC probabilities and propensities:

passenger.wcProbability = 0.7;
if (randomTrue(passenger.wcProbability))
{
	passenger.taskWC = 1;

	double distanceWC1 = calculateDistances(WC1X, WC1Y, passenger);
	passenger.propensityWC1 = passenger.wcPriority/distanceWC1;

	double distanceWC2 = calculateDistances(WC2X, WC2Y, passenger);
	passenger.propensityWC2 = passenger.wcPriority/distanceWC2;

	passenger.tasks.add(new Task(TaskType.wc, 0));

}


// Help probabilities and propensities:

if (passenger.ageClass == 0 && passenger.freqTravel < 3)
{
	passenger.helpProbability = 0.9;

	if (randomTrue(passenger.helpProbability))
	{
	passenger.taskHelp = 1;

	passenger.tasks.add(new Task(TaskType.help, 0));

	double distanceHelp = calculateDistances(HelpX, HelpY, passenger);

	passenger.propensityHelp = passenger.helpPriority/distanceHelp;
	}
}

else if (passenger.ageClass == 1 && passenger.freqTravel < 3)
{
	passenger.taskHelp = 1;

	passenger.tasks.add(new Task(TaskType.help, 0));

	double distanceHelp = calculateDistances(HelpX, HelpY, passenger);

	passenger.propensityHelp = passenger.helpPriority/distanceHelp;
}
```

```
else if (passenger.ageClass == 1 && passenger.freqTravel > 2)
{
        passenger.helpProbability = 0.4;
        if (randomTrue(passenger.helpProbability))
        {
        passenger.taskHelp = 1;

        passenger.tasks.add(new Task(TaskType.help, 0));

        double distanceHelp = calculateDistances(HelpX, HelpY, passenger);

        passenger.propensityHelp = passenger.helpPriority/distanceHelp;
        }
}

// Shop probabilities and propensities:

if (passenger.male == true && passenger.foreigner == false && passenger.freqTravel < 3)
{
        passenger.shopProbability = 0.4;
        if (randomTrue(passenger.shopProbability))
        {

                int shopCount = uniform_discr(1, 3);
                for (int i = 0; i < shopCount; i++)
                {
                        int shopIndex = generateShopIndex(passenger);
                        passenger.tasks.add(new Task(TaskType.shop, shopIndex));

                        if (shopIndex == 0)
                        {
                                passenger.taskShop1 = 1;
                                double distanceShop1 = calculateDistances(Shop1X, Shop1Y,
passenger);
                                passenger.propensityShop1 = passenger.shopPriority/distanceShop1;
                        }

                        else if (shopIndex == 1)
                        {
                                passenger.taskShop2 = 1;
                                double distanceShop2 = calculateDistances(Shop2X, Shop2Y,
passenger);
                                passenger.propensityShop2 = passenger.shopPriority/distanceShop2;
                        }

                        else if (shopIndex == 2)
                        {
                                passenger.taskShop3 = 1;
                                double distanceShop3 = calculateDistances(Shop3X, Shop3Y,
passenger);
                                passenger.propensityShop3 = passenger.shopPriority/distanceShop3;
                        }

                        else if (shopIndex == 3)
                        {
                                passenger.taskShop4 = 1;
                                        double distanceShop4 = calculateDistances(Shop4X,
Shop4Y, passenger);
                                passenger.propensityShop4 = passenger.shopPriority/distanceShop4;
```

```
                    }

                }


            }
}

else if (passenger.male == true && passenger.foreigner == false && passenger.freqTravel > 2)

{
            passenger.shopProbability = 0.5;
            if (randomTrue(passenger.shopProbability))
            {

                int shopCount = uniform_discr(1, 3);

                for (int i = 0; i < shopCount; i++)
                {
                    int shopIndex = generateShopIndex(passenger);
                    passenger.tasks.add(new Task(TaskType.shop, shopIndex));

                    if (shopIndex == 0)
                    {
                        passenger.taskShop1 = 1;
                        double distanceShop1 = calculateDistances(Shop1X, Shop1Y,
passenger);

                        passenger.propensityShop1 = passenger.shopPriority/distanceShop1;
                    }

                    else if (shopIndex == 1)
                    {
                        passenger.taskShop2 = 1;
                        double distanceShop2 = calculateDistances(Shop2X, Shop2Y,
passenger);

                        passenger.propensityShop2 = passenger.shopPriority/distanceShop2;
                    }

                    else if (shopIndex == 2)
                    {
                        passenger.taskShop3 = 1;
                        double distanceShop3 = calculateDistances(Shop3X, Shop3Y,
passenger);

                        passenger.propensityShop3 = passenger.shopPriority/distanceShop3;
                    }

                    else if (shopIndex == 3)
                    {
                        passenger.taskShop4 = 1;
                        double distanceShop4 = calculateDistances(Shop4X, Shop4Y,
passenger);

                        passenger.propensityShop4 = passenger.shopPriority/distanceShop4;
                    }

                }


            }
}
```

```
else if (passenger.male == true && passenger.foreigner == true && passenger.freqTravel < 3)
{
	passenger.shopProbability = 0.2;
	if (randomTrue(passenger.shopProbability))
	{

		int shopCount = uniform_discr(1, 3);
		for (int i = 0; i < shopCount; i++)
		{
			int shopIndex = generateShopIndex(passenger);
						passenger.tasks.add(new Task(TaskType.shop,
shopIndex));

			if (shopIndex == 0)
			{
				passenger.taskShop1 = 1;
				double distanceShop1 = calculateDistances(Shop1X, Shop1Y,
passenger);
				passenger.propensityShop1 = passenger.shopPriority/distanceShop1;
			}

			else if (shopIndex == 1)
			{
				passenger.taskShop2 = 1;
				double distanceShop2 = calculateDistances(Shop2X, Shop2Y,
passenger);
				passenger.propensityShop2 = passenger.shopPriority/distanceShop2;
			}

			else if (shopIndex == 2)
			{
				passenger.taskShop3 = 1;
				double distanceShop3 = calculateDistances(Shop3X, Shop3Y,
passenger);
				passenger.propensityShop3 = passenger.shopPriority/distanceShop3;
			}

			else if (shopIndex == 3)
			{
				passenger.taskShop4 = 1;
				double distanceShop4 = calculateDistances(Shop4X, Shop4Y,
passenger);
				passenger.propensityShop4 = passenger.shopPriority/distanceShop4;
			}

		}

	}
}

else if (passenger.male == true && passenger.foreigner == true && passenger.freqTravel > 2)
{
	passenger.shopProbability = 0.5;
	if (randomTrue(passenger.shopProbability))
	{
		int shopCount = uniform_discr(1, 3);
		for (int i = 0; i < shopCount; i++)
		{
			int shopIndex = generateShopIndex(passenger);
			passenger.tasks.add(new Task(TaskType.shop, shopIndex));
```

```
				if (shopIndex == 0)
				{
					passenger.taskShop1 = 1;
					double distanceShop1 = calculateDistances(Shop1X, Shop1Y, 
passenger);
					passenger.propensityShop1 = passenger.shopPriority/distanceShop1;
				}

				else if (shopIndex == 1)
					{
					passenger.taskShop2 = 1;
					double distanceShop2 = calculateDistances(Shop2X, Shop2Y, 
passenger);
					passenger.propensityShop2 = passenger.shopPriority/distanceShop2;
				}

				else if (shopIndex == 2)
				{
					passenger.taskShop3 = 1;
					double distanceShop3 = calculateDistances(Shop3X, Shop3Y, 
passenger);
					passenger.propensityShop3 = passenger.shopPriority/distanceShop3;
				}

				else if (shopIndex == 3)
				{
					passenger.taskShop4 = 1;
					double distanceShop4 = calculateDistances(Shop4X, Shop4Y, 
passenger);
					passenger.propensityShop4 = passenger.shopPriority/distanceShop4;
				}

			}
		}
}

else if (passenger.male == false && passenger.foreigner == false && passenger.freqTravel < 3)
{
		int shopCount = uniform_discr(1, 3);
		for (int i = 0; i < shopCount; i++)
		{
			int shopIndex = generateShopIndex(passenger);
			passenger.tasks.add(new Task(TaskType.shop, shopIndex));

			if (shopIndex == 0)
			{
				passenger.taskShop1 = 1;
				double distanceShop1 = calculateDistances(Shop1X, Shop1Y, passenger);
				passenger.propensityShop1 = passenger.shopPriority/distanceShop1;
			}

			else if (shopIndex == 1)
			{
				passenger.taskShop2 = 1;
				double distanceShop2 = calculateDistances(Shop2X, Shop2Y, passenger);
				passenger.propensityShop2 = passenger.shopPriority/distanceShop2;
			}

			else if (shopIndex == 2)
```

```
                {
                        passenger.taskShop3 = 1;
                        double distanceShop3 = calculateDistances(Shop3X, Shop3Y, passenger);
                        passenger.propensityShop3 = passenger.shopPriority/distanceShop3;
                }

                else if (shopIndex == 3)
                {
                        passenger.taskShop4 = 1;
                        double distanceShop4 = calculateDistances(Shop4X, Shop4Y, passenger);
                        passenger.propensityShop4 = passenger.shopPriority/distanceShop4;
                }

        }

}

else if (passenger.male == false && passenger.foreigner == false && passenger.freqTravel > 2)

{
        passenger.shopProbability = 0.9;
        boolean shopProb = randomTrue(passenger.shopProbability);
        if (shopProb = true)
        {
                int shopCount = uniform_discr(1, 3);
                for (int i = 0; i < shopCount; i++)
                {
                        int shopIndex = generateShopIndex(passenger);
                        passenger.tasks.add(new Task(TaskType.shop, shopIndex));

                        if (shopIndex == 0)
                        {
                                passenger.taskShop1 = 1;
                                double distanceShop1 = calculateDistances(Shop1X, Shop1Y,
passenger);
                                passenger.propensityShop1 = passenger.shopPriority/distanceShop1;
                        }

                        else if (shopIndex == 1)
                        {
                                passenger.taskShop2 = 1;
                                double distanceShop2 = calculateDistances(Shop2X, Shop2Y,
passenger);
                                passenger.propensityShop2 = passenger.shopPriority/distanceShop2;
                        }

                        else if (shopIndex == 2)
                        {
                                passenger.taskShop3 = 1;
                                double distanceShop3 = calculateDistances(Shop3X, Shop3Y,
passenger);
                                passenger.propensityShop3 = passenger.shopPriority/distanceShop3;
                        }

                        else if (shopIndex == 3)
                        {
                                passenger.taskShop4 = 1;
                                double distanceShop4 = calculateDistances(Shop4X, Shop4Y,
passenger);
                                passenger.propensityShop4 = passenger.shopPriority/distanceShop4;
```

```
                }

            }
        }
}

else if (passenger.male == false && passenger.foreigner == true)
{
        int shopCount = uniform_discr(1, 3);
        for (int i = 0; i < shopCount; i++)
        {
                int shopIndex = generateShopIndex(passenger);

                passenger.tasks.add(new Task(TaskType.shop, shopIndex));

                if (shopIndex == 0)
                {
                        passenger.taskShop1 = 1;
                        double distanceShop1 = calculateDistances(Shop1X, Shop1Y, passenger);
                        passenger.propensityShop1 = passenger.shopPriority/distanceShop1;
                }

                else if (shopIndex == 1)
                {
                        passenger.taskShop2 = 1;
                        double distanceShop2 = calculateDistances(Shop2X, Shop2Y, passenger);
                        passenger.propensityShop2 = passenger.shopPriority/distanceShop2;
                }

                else if (shopIndex == 2)
                {
                        passenger.taskShop3 = 1;
                        double distanceShop3 = calculateDistances(Shop3X, Shop3Y, passenger);
                        passenger.propensityShop3 = passenger.shopPriority/distanceShop3;
                }

                else if (shopIndex == 3)
                {
                        passenger.taskShop4 = 1;
                        double distanceShop4 = calculateDistances(Shop4X, Shop4Y, passenger);
                        passenger.propensityShop4 = passenger.shopPriority/distanceShop4;
                }

        }
}
// Social Connectivity probabilities and propensities:

if (passenger.ageClass == 0 && passenger.business == false)
{
        passenger.socialProbability = 0.8;
        if (randomTrue(passenger.socialProbability))
        {
                passenger.taskSocial = 1;

                passenger.tasks.add(new Task(TaskType.socialConnectivity, 0));

                double distanceSocial1 = calculateDistances(Social1X, Social1Y, passenger);
                double distanceSocial2 = calculateDistances(Social2X, Social2Y, passenger);
                double distanceSocial3 = calculateDistances(Social3X, Social3Y, passenger);
                double distanceSocial4 = calculateDistances(Social4X, Social4Y, passenger);
```

```
                double distanceSocial = min(distanceSocial1, min(distanceSocial2,
min(distanceSocial3, distanceSocial4)));

                passenger.propensitySocial = passenger.socialPriority/distanceSocial;
        }
}

else if (passenger.ageClass == 0 && passenger.business == true)
{
        passenger.taskSocial = 1;

        passenger.tasks.add(new Task(TaskType.socialConnectivity, 0));

        double distanceSocial1 = calculateDistances(Social1X, Social1Y, passenger);
        double distanceSocial2 = calculateDistances(Social2X, Social2Y, passenger);
        double distanceSocial3 = calculateDistances(Social3X, Social3Y, passenger);
        double distanceSocial4 = calculateDistances(Social4X, Social4Y, passenger);

        double distanceSocial = min(distanceSocial1, min(distanceSocial2, min(distanceSocial3,
distanceSocial4)));

        passenger.propensitySocial = passenger.socialPriority/distanceSocial;
}

else if (passenger.ageClass == 1 && passenger.business == true)
{
        passenger.socialProbability = 0.2;
        if (randomTrue(passenger.socialProbability))
        {
                passenger.taskSocial = 1;

                passenger.tasks.add(new Task(TaskType.socialConnectivity, 0));

                double distanceSocial1 = calculateDistances(Social1X, Social1Y, passenger);
                double distanceSocial2 = calculateDistances(Social2X, Social2Y, passenger);
                double distanceSocial3 = calculateDistances(Social3X, Social3Y, passenger);
                double distanceSocial4 = calculateDistances(Social4X, Social4Y, passenger);

                double distanceSocial = min(distanceSocial1, min(distanceSocial2,
min(distanceSocial3, distanceSocial4)));

                passenger.propensitySocial = passenger.socialPriority/distanceSocial;
        }
}

// Withdraw money probabilities and propensities:

if (passenger.foreigner == false && passenger.freqTravel < 3)
{
        passenger.cashProbability = 0.1;
        if (randomTrue(passenger.cashProbability))
        {
                passenger.taskCash = 1;

                passenger.tasks.add(new Task(TaskType.withdrawMoney, 0));

                double distanceCash = calculateDistances(CashX, CashY, passenger);

                passenger.propensityCash = passenger.cashPriority/distanceCash;
```

```
        }
}

else if (passenger.foreigner = false && passenger.freqTravel > 2)
{
        passenger.cashProbability = 0.5;
        if (randomTrue(passenger.cashProbability))
        {
                passenger.taskCash = 1;

                passenger.tasks.add(new Task(TaskType.withdrawMoney, 0));

                double distanceCash = calculateDistances(CashX, CashY, passenger);

                passenger.propensityCash = passenger.cashPriority/distanceCash;

        }
}

else if (passenger.foreigner = true && passenger.freqTravel < 3)
{
        passenger.cashProbability = 0.1;
        if (randomTrue(passenger.cashProbability))
        {
                passenger.taskCash = 1;

                passenger.tasks.add(new Task(TaskType.withdrawMoney, 0));

                double distanceCash = calculateDistances(CashX, CashY, passenger);

                passenger.propensityCash = passenger.cashPriority/distanceCash;
        }
}

else if (passenger.foreigner = true && passenger.freqTravel > 2)
{
        passenger.cashProbability = 0.2;
        if (randomTrue(passenger.cashProbability))
        {
                passenger.taskCash = 1;

                passenger.tasks.add(new Task(TaskType.withdrawMoney, 0));

                double distanceCash = calculateDistances(CashX, CashY, passenger);

                passenger.propensityCash = passenger.cashPriority/distanceCash;

        }
}

// Food probabilities and propensities:

passenger.foodProbability = 0.5;

if (randomTrue(passenger.foodProbability))
{
        int foodCount = 1;

        for (int i = 0; i < foodCount; i++)
```

```
	{
		int foodIndex = uniform_discr(0, foodEntrances.size() - 1);
		passenger.tasks.add(new Task(TaskType.food, foodIndex));

		if (foodIndex == 0)
		{
		passenger.taskFood1 = 1;
		double distanceFood1 = calculateDistances(Food1X, Food1Y, passenger);
		passenger.propensityFood1 = passenger.foodPriority/distanceFood1;
		}

		else if (foodIndex == 1)
		{
		passenger.taskFood2 = 1;
		double distanceFood2 = calculateDistances(Food2X, Food2Y, passenger);
		passenger.propensityFood2 = passenger.foodPriority/distanceFood2;
		}

		else if (foodIndex == 2)
		{
		passenger.taskFood3 = 1;
		double distanceFood3 = calculateDistances(Food3X, Food3Y, passenger);
		passenger.propensityFood3 = passenger.foodPriority/distanceFood3;
		}

		else if (foodIndex == 3)
		{
		passenger.taskFood4 = 1;
		double distanceFood4 = calculateDistances(Food4X, Food4Y, passenger);
		passenger.propensityFood4 = passenger.foodPriority/distanceFood4;
		}

	}
}
// Check if there is still something to do on the list and, if not, change the values of the two auxiliary variables:

if (passenger.propensityWC1 == -100000
&& passenger.propensityWC2 == -100000
&& passenger.propensityShop1 == -100000
&& passenger.propensityShop2 == -100000
&& passenger.propensityShop3 == -100000
&& passenger.propensityShop4 == -100000
&& passenger.propensityFood1 == -100000
&& passenger.propensityFood2 == -100000
&& passenger.propensityFood3 == -100000
&& passenger.propensityFood4 == -100000
&& passenger.propensityHelp == -100000
&& passenger.propensitySocial == -100000
&& passenger.propensityCash == -100000)
{
	passenger.goesToWaitingArea = 1;
	passenger.finishedTasksAtLeastOnce = 1;
}

// Calculte probabilities:

passenger.Food1Prob = exp(passenger.propensityFood1)
					/(exp(passenger.propensityFood1)
					+ exp(passenger.propensityFood2)
```

```
                                        + exp(passenger.propensityFood3)
                                        + exp(passenger.propensityFood4)
                                        + exp(passenger.propensityShop1)
                                        + exp(passenger.propensityShop2)
                                        + exp(passenger.propensityShop3)
                                        + exp(passenger.propensityShop4)
                                        + exp(passenger.propensityWC1)
                                        + exp(passenger.propensityWC2)
                                        + exp(passenger.propensityHelp)
                                        + exp(passenger.propensitySocial)
                                        + exp(passenger.propensityCash));

passenger.Food2Prob = exp(passenger.propensityFood2)
                                        /(exp(passenger.propensityFood1)
                                        + exp(passenger.propensityFood2)
                                        + exp(passenger.propensityFood3)
                                        + exp(passenger.propensityFood4)
                                        + exp(passenger.propensityShop1)
                                        + exp(passenger.propensityShop2)
                                        + exp(passenger.propensityShop3)
                                        + exp(passenger.propensityShop4)
                                        + exp(passenger.propensityWC1)
                                        + exp(passenger.propensityWC2)
                                        + exp(passenger.propensityHelp)
                                        + exp(passenger.propensitySocial)
                                        + exp(passenger.propensityCash));

passenger.Food3Prob = exp(passenger.propensityFood3)
                                        /(exp(passenger.propensityFood1)
                                        + exp(passenger.propensityFood2)
                                        + exp(passenger.propensityFood3)
                                        + exp(passenger.propensityFood4)
                                        + exp(passenger.propensityShop1)
                                        + exp(passenger.propensityShop2)
                                        + exp(passenger.propensityShop3)
                                        + exp(passenger.propensityShop4)
                                        + exp(passenger.propensityWC1)
                                        + exp(passenger.propensityWC2)
                                        + exp(passenger.propensityHelp)
                                        + exp(passenger.propensitySocial)
                                        + exp(passenger.propensityCash));

passenger.Food4Prob = exp(passenger.propensityFood4)
                                        /(exp(passenger.propensityFood1)
                                        + exp(passenger.propensityFood2)
                                        + exp(passenger.propensityFood3)
                                        + exp(passenger.propensityFood4)
                                        + exp(passenger.propensityShop1)
                                        + exp(passenger.propensityShop2)
                                        + exp(passenger.propensityShop3)
                                        + exp(passenger.propensityShop4)
                                        + exp(passenger.propensityWC1)
                                        + exp(passenger.propensityWC2)
                                        + exp(passenger.propensityHelp)
                                        + exp(passenger.propensitySocial)
                                        + exp(passenger.propensityCash));


passenger.Shop1Prob = exp(passenger.propensityShop1)
                                        /(exp(passenger.propensityFood1)
```

```
                                                + exp(passenger.propensityFood2)
                                                + exp(passenger.propensityFood3)
                                                + exp(passenger.propensityFood4)
                                                + exp(passenger.propensityShop1)
                                                + exp(passenger.propensityShop2)
                                                + exp(passenger.propensityShop3)
                                                + exp(passenger.propensityShop4)
                                                + exp(passenger.propensityWC1)
                                                + exp(passenger.propensityWC2)
                                                + exp(passenger.propensityHelp)
                                                + exp(passenger.propensitySocial)
                                                + exp(passenger.propensityCash));

passenger.Shop2Prob = exp(passenger.propensityShop2)
                                                /(exp(passenger.propensityFood1)
                                                + exp(passenger.propensityFood2)
                                                + exp(passenger.propensityFood3)
                                                + exp(passenger.propensityFood4)
                                                + exp(passenger.propensityShop1)
                                                + exp(passenger.propensityShop2)
                                                + exp(passenger.propensityShop3)
                                                + exp(passenger.propensityShop4)
                                                + exp(passenger.propensityWC1)
                                                + exp(passenger.propensityWC2)
                                                + exp(passenger.propensityHelp)
                                                + exp(passenger.propensitySocial)
                                                + exp(passenger.propensityCash));

passenger.Shop3Prob = exp(passenger.propensityShop3)
                                                /(exp(passenger.propensityFood1)
                                                + exp(passenger.propensityFood2)
                                                + exp(passenger.propensityFood3)
                                                + exp(passenger.propensityFood4)
                                                + exp(passenger.propensityShop1)
                                                + exp(passenger.propensityShop2)
                                                + exp(passenger.propensityShop3)
                                                + exp(passenger.propensityShop4)
                                                + exp(passenger.propensityWC1)
                                                + exp(passenger.propensityWC2)
                                                + exp(passenger.propensityHelp)
                                                + exp(passenger.propensitySocial)
                                                + exp(passenger.propensityCash));

passenger.Shop4Prob = exp(passenger.propensityShop4)
                                                /(exp(passenger.propensityFood1)
                                                + exp(passenger.propensityFood2)
                                                + exp(passenger.propensityFood3)
                                                + exp(passenger.propensityFood4)
                                                + exp(passenger.propensityShop1)
                                                + exp(passenger.propensityShop2)
                                                + exp(passenger.propensityShop3)
                                                + exp(passenger.propensityShop4)
                                                + exp(passenger.propensityWC1)
                                                + exp(passenger.propensityWC2)
                                                + exp(passenger.propensityHelp)
                                                + exp(passenger.propensitySocial)
                                                + exp(passenger.propensityCash));

passenger.WC1Prob = exp(passenger.propensityWC1)
                                                /(exp(passenger.propensityFood1)
```

```
                                                  + exp(passenger.propensityFood2)
                                                  + exp(passenger.propensityFood3)
                                                  + exp(passenger.propensityFood4)
                                                  + exp(passenger.propensityShop1)
                                                  + exp(passenger.propensityShop2)
                                                  + exp(passenger.propensityShop3)
                                                  + exp(passenger.propensityShop4)
                                                  + exp(passenger.propensityWC1)
                                                  + exp(passenger.propensityWC2)
                                                  + exp(passenger.propensityHelp)
                                                  + exp(passenger.propensitySocial)
                                                  + exp(passenger.propensityCash));

passenger.WC2Prob = exp(passenger.propensityWC2)
                                                  /(exp(passenger.propensityFood1)
                                                  + exp(passenger.propensityFood2)
                                                  + exp(passenger.propensityFood3)
                                                  + exp(passenger.propensityFood4)
                                                  + exp(passenger.propensityShop1)
                                                  + exp(passenger.propensityShop2)
                                                  + exp(passenger.propensityShop3)
                                                  + exp(passenger.propensityShop4)
                                                  + exp(passenger.propensityWC1)
                                                  + exp(passenger.propensityWC2)
                                                  + exp(passenger.propensityHelp)
                                                  + exp(passenger.propensitySocial)
                                                  + exp(passenger.propensityCash));

passenger.HelpProb = exp(passenger.propensityHelp)
                                                  /(exp(passenger.propensityFood1)
                                                  + exp(passenger.propensityFood2)
                                                  + exp(passenger.propensityFood3)
                                                  + exp(passenger.propensityFood4)
                                                  + exp(passenger.propensityShop1)
                                                  + exp(passenger.propensityShop2)
                                                  + exp(passenger.propensityShop3)
                                                  + exp(passenger.propensityShop4)
                                                  + exp(passenger.propensityWC1)
                                                  + exp(passenger.propensityWC2)
                                                  + exp(passenger.propensityHelp)
                                                  + exp(passenger.propensitySocial)
                                                  + exp(passenger.propensityCash));

passenger.SocialProb = exp(passenger.propensitySocial)
                                                  /(exp(passenger.propensityFood1)
                                                  + exp(passenger.propensityFood2)
                                                  + exp(passenger.propensityFood3)
                                                  + exp(passenger.propensityFood4)
                                                  + exp(passenger.propensityShop1)
                                                  + exp(passenger.propensityShop2)
                                                  + exp(passenger.propensityShop3)
                                                  + exp(passenger.propensityShop4)
                                                  + exp(passenger.propensityWC1)
                                                  + exp(passenger.propensityWC2)
                                                  + exp(passenger.propensityHelp)
                                                  + exp(passenger.propensitySocial)
                                                  + exp(passenger.propensityCash));


passenger.CashProb = exp(passenger.propensityCash)
```

```
                                          /(exp(passenger.propensityFood1)
                                          + exp(passenger.propensityFood2)
                                          + exp(passenger.propensityFood3)
                                          + exp(passenger.propensityFood4)
                                          + exp(passenger.propensityShop1)
                                          + exp(passenger.propensityShop2)
                                          + exp(passenger.propensityShop3)
                                          + exp(passenger.propensityShop4)
                                          + exp(passenger.propensityWC1)
                                          + exp(passenger.propensityWC2)
                                          + exp(passenger.propensityHelp)
                                          + exp(passenger.propensitySocial)
                                          + exp(passenger.propensityCash));

// Choice parameter:

passenger.TaskChoice = uniform(0,1);

// Task Choice:

if (passenger.TaskChoice <= passenger.Food1Prob && passenger.taskFood1 == 1)
{
passenger.chosenTask = 1;
}

else if (passenger.TaskChoice <= passenger.Food1Prob + passenger.Food2Prob &&
passenger.taskFood2 == 1)
{
passenger.chosenTask = 2;
}

else if (passenger.TaskChoice <= passenger.Food1Prob + passenger.Food2Prob +
passenger.Food3Prob && passenger.taskFood3 == 1)
{
passenger.chosenTask = 3;
}

else if (passenger.TaskChoice <= passenger.Food1Prob + passenger.Food2Prob +
passenger.Food3Prob + passenger.Food4Prob && passenger.taskFood4 == 1)
{
passenger.chosenTask = 4;
}

else if (passenger.TaskChoice <= passenger.Food1Prob + passenger.Food2Prob +
passenger.Food3Prob + passenger.Food4Prob
 + passenger.Shop1Prob && passenger.taskShop1 == 1)
{
passenger.chosenTask = 5;
}

else if (passenger.TaskChoice <= passenger.Food1Prob + passenger.Food2Prob +
passenger.Food3Prob + passenger.Food4Prob
 + passenger.Shop1Prob + passenger.Shop2Prob && passenger.taskShop2 == 1)
{
passenger.chosenTask = 6;
}

else if (passenger.TaskChoice <= passenger.Food1Prob + passenger.Food2Prob +
passenger.Food3Prob + passenger.Food4Prob
```

```
 + passenger.Shop1Prob + passenger.Shop2Prob + passenger.Shop3Prob && passenger.taskShop3
== 1)
{
passenger.chosenTask = 7;
}

else if (passenger.TaskChoice <= passenger.Food1Prob + passenger.Food2Prob +
passenger.Food3Prob + passenger.Food4Prob
 + passenger.Shop1Prob + passenger.Shop2Prob + passenger.Shop3Prob + passenger.Shop4Prob
&& passenger.taskShop4 == 1)
{
passenger.chosenTask = 8;
}

else if (passenger.TaskChoice <= passenger.Food1Prob + passenger.Food2Prob +
passenger.Food3Prob + passenger.Food4Prob
 + passenger.Shop1Prob + passenger.Shop2Prob + passenger.Shop3Prob + passenger.Shop4Prob
 + passenger.WC1Prob && passenger.taskWC == 1)
{
passenger.chosenTask = 9;
}

else if (passenger.TaskChoice <= passenger.Food1Prob + passenger.Food2Prob +
passenger.Food3Prob + passenger.Food4Prob
 + passenger.Shop1Prob + passenger.Shop2Prob + passenger.Shop3Prob + passenger.Shop4Prob
 + passenger.WC1Prob + passenger.WC2Prob && passenger.taskWC == 1)
{
passenger.chosenTask = 10;
}

else if (passenger.TaskChoice <= passenger.Food1Prob + passenger.Food2Prob +
passenger.Food3Prob + passenger.Food4Prob
 + passenger.Shop1Prob + passenger.Shop2Prob + passenger.Shop3Prob + passenger.Shop4Prob
 + passenger.WC1Prob + passenger.WC2Prob
 + passenger.CashProb && passenger.taskCash == 1)
{
passenger.chosenTask = 11;
}

else if (passenger.TaskChoice <= passenger.Food1Prob + passenger.Food2Prob +
passenger.Food3Prob + passenger.Food4Prob
 + passenger.Shop1Prob + passenger.Shop2Prob + passenger.Shop3Prob + passenger.Shop4Prob
 + passenger.WC1Prob + passenger.WC2Prob
 + passenger.CashProb
 + passenger.SocialProb && passenger.taskSocial == 1)
{
passenger.chosenTask = 12;
}

else if (passenger.TaskChoice <= 1 && passenger.taskHelp == 1)
{
passenger.chosenTask = 13;
}
```

- ***recalculatePropensities* function** – this function requires one argument of type Passenger, called *passenger*, in order to work. Some of the code lines are the same as in the generateTasks function, so are written again here. These parts are specified in blue:

```
// Calculates the different distances and propensities again
```

```
// WC propensities:

if (passenger.taskWC == 1 && passenger.wcPriority >= 50)
{

double distanceWC1 = calculateDistances(WC1X, WC1Y, passenger);
double distanceWC2 = calculateDistances(WC2X, WC2Y, passenger);

passenger.propensityWC1 = passenger.wcPriority/distanceWC1;
passenger.propensityWC2 = passenger.wcPriority/distanceWC2;

}

if (passenger.wcPriority < 50)
{
        passenger.propensityWC1 = -100000;
        passenger.propensityWC2 = -100000;

}


// Help propensity:

if (passenger.taskHelp == 1)
{
double distanceHelp = calculateDistances(HelpX, HelpY, passenger);

passenger.propensityHelp = passenger.helpPriority/distanceHelp;
}

// Shop propensities:

if (passenger.taskShop1 == 1)
{
double distanceShop1 = calculateDistances(Shop1X, Shop1Y, passenger);
passenger.propensityShop1 = passenger.shopPriority/distanceShop1;
}

if (passenger.taskShop2 == 1)
{
double distanceShop2 = calculateDistances(Shop2X, Shop2Y, passenger);
passenger.propensityShop2 = passenger.shopPriority/distanceShop2;
}

if (passenger.taskShop3 == 1)
{
double distanceShop3 = calculateDistances(Shop3X, Shop3Y, passenger);
passenger.propensityShop3 = passenger.shopPriority/distanceShop3;
}

if (passenger.taskShop4 == 1)
{
double distanceShop4 = calculateDistances(Shop4X, Shop4Y, passenger);
passenger.propensityShop4 = passenger.shopPriority/distanceShop4;
}

// Social Connectivity propensity:

if (passenger.taskSocial == 1)
```

```
{

        double distanceSocial1 = calculateDistances(Social1X, Social1Y, passenger);
        double distanceSocial2 = calculateDistances(Social2X, Social2Y, passenger);
        double distanceSocial3 = calculateDistances(Social3X, Social3Y, passenger);
        double distanceSocial4 = calculateDistances(Social4X, Social4Y, passenger);

        double distanceSocial = min(distanceSocial1, min(distanceSocial2, min(distanceSocial3,
distanceSocial4)));

        passenger.propensitySocial = passenger.socialPriority/distanceSocial;
}

// Withdraw money propensity:

if (passenger.taskCash == 1)
{
double distanceCash = calculateDistances(CashX, CashY, passenger);

passenger.propensityCash = passenger.cashPriority/distanceCash;
}



// Food propensities:

if (passenger.taskFood1 == 1 && passenger.foodPriority >= 45)
{
double distanceFood1 = calculateDistances(Food1X, Food1Y, passenger);
passenger.propensityFood1 = passenger.foodPriority/distanceFood1;
}

if (passenger.taskFood2 == 1 && passenger.foodPriority >= 45)
{
double distanceFood2 = calculateDistances(Food2X, Food2Y, passenger);
passenger.propensityFood2 = passenger.foodPriority/distanceFood2;
}

if (passenger.taskFood3 == 1 && passenger.foodPriority >= 45)
{
double distanceFood3 = calculateDistances(Food3X, Food3Y, passenger);
passenger.propensityFood3 = passenger.foodPriority/distanceFood3;
}

if (passenger.taskFood4 == 1 && passenger.foodPriority >= 45)
{
double distanceFood4 = calculateDistances(Food4X, Food4Y, passenger);
passenger.propensityFood4 = passenger.foodPriority/distanceFood4;
}

if (passenger.foodPriority < 45)
{
        passenger.propensityFood1 = -100000;
        passenger.propensityFood2 = -100000;
        passenger.propensityFood3 = -100000;
        passenger.propensityFood4 = -100000;
}

// Check if there is still something to do on the list and, if not, change the values of the two auxiliary
variables:
```

```
[same code lines as in the generateTasks function]

// Probabilities:
[same code lines as in the generateTasks function]
// Choice parameter:
passenger.TaskChoice = uniform(0,1);
// Task Choice:
[same code lines as in the generateTasks function]
```

Appendix 4. Flights and number of passengers leaving Lisbon T2 on September 2nd, 2015

Flights that departed from Lisbon T2 on September 2nd, 2015 (ANA, 2015a):

Hour	Flight No.	Airline	Destination
06:15	EZY7651	EasyJet Airlines	Berlin, Schoenefeld
06:30	FR 1885	Ryanair	London, Stansted
06:40	EZY8716	EasyJet Airlines	London, Gatwick
06:40	FR 1083	Ryanair	Paris, Beauvais
07:00	EZY7611	EasyJet Airlines	Madrid
07:10	EZY7603	EasyJet Airlines	Madeira
07:55	FR 2094	Ryanair	Porto
08:25	EZS1444	EasyJet Switzerland	Geneva
09:00	FR 2941	Ryanair	Hamburg
09:15	FR 2086	Ryanair	Milan/Bergamo
09:45	TO 3413	Transavia France	Paris, Orly
09:55	HV 5952	Transavia	Amsterdam
10:00	EZY2366	EasyJet Airlines	London, Luton
10:00	FR 7329	Ryanair	Dublin
10:05	FR 1787	Ryanair	Frankfurt, Hahn
10:45	EZY4434	EasyJet Airlines	Lyon, St. Exupery
10:45	EZY7084	EasyJet Airlines	Liverpool
11:00	FR 2926	Ryanair	Brussels
11:40	EZY7637	EasyJet Airlines	Bordeaux
12:35	FR 1671	Ryanair	Porto
13:30	EZY1448	EasyJet Airlines	Geneva
14:05	FR 2077	Ryanair	Marseille
14:45	TO 3415	Transavia France	Paris, Orly
14:45	TO 3929	Transavia France	Nantes
15:05	EZY7615	EasyJet Airlines	Madrid
15:50	EZY2368	EasyJet Airlines	London, Luton
16:35	FR 2097	Ryanair	Rome, Ciampino
16:40	EZY7663	EasyJet Airlines	Amsterdam
16:45	EZY7687	EasyJet Airlines	Basel
17:30	FR 2932	Ryanair	Brussels
17:40	FR 2253	Ryanair	Manchester
18:10	DY 1787	Norwegian	Oslo
18:50	EZY7607	EasyJet Airlines	Madeira
19:20	HV 6204	Transavia	Eindhoven
19:20	TO 3417	Transavia France	Paris, Orly
19:25	EZY3764	EasyJet Airlines	Paris, Ch. de Gaulle
20:20	EZY8720	EasyJet Airlines	London, Gatwick
20:25	EZY2716	EasyJet Airlines	Milan, Malpensa
20:35	FR 2096	Ryanair	Porto

Estimated number of passengers per aircraft departing Lisbon T2 on September 2nd, 2015, and boarding gates for each flight:

Hour	Flight No.	Aircraft (FlightAware, 2015)	Nº of seats (PlaneSpotters, 2015)	Nº of pax (90 % of nº of seats)	Boarding gates
06:15	EZY7651	Airbus A319	156	141	1
06:30	FR 1885	Boeing 737-800	189	171	2
06:40	EZY8716	Airbus A320	180	162	3
06:40	FR 1083	Boeing 737-800	189	171	4
07:00	EZY7611	Airbus A319	156	141	5
07:10	EZY7603	Airbus A319	156	141	6
07:55	FR 2094	Boeing 737-800	189	171	7
08:25	**EZS1444**	**(?) Airbus A319**	**156**	**141**	**8**
09:00	FR 2941	Boeing 737-800	189	171	1
09:15	FR 2086	Boeing 737-800	189	171	2
09:45	TO 3413	Airbus A320	178	161	3
09:55	HV 5952	Boeing 737-800	183	165	4
10:00	EZY2366	Airbus A319	156	141	5
10:00	FR 7329	Boeing 737-800	189	171	6
10:05	FR 1787	Boeing 737-800	189	171	7
10:45	EZY4434	Airbus A319	156	141	8
10:45	EZY7084	Airbus A319	156	141	1
11:00	FR 2926	Boeing 737-800	189	171	2
11:40	EZY7637	Airbus A319	156	141	3
12:35	FR 1671	Boeing 737-800	189	171	4
13:30	EZY1448	Airbus A319	156	141	5
14:05	FR 2077	Boeing 737-800	189	171	6
14:45	TO 3415	Boeing 737-800	183	165	7
14:45	TO 3929	Boeing 737-800	183	165	8
15:05	EZY7615	Airbus A320	180	162	1
15:50	EZY2368	Airbus A319	156	141	2
16:35	FR 2097	Boeing 737-800	189	171	3
16:40	EZY7663	Airbus A319	156	141	4
16:45	**EZY7687**	**(?) Airbus A319**	**156**	**141**	**5**
17:30	FR 2932	Boeing 737-800	189	171	6
17:40	FR 2253	Boeing 737-800	189	171	7
18:10	DY 1787	Boeing 737-800	186	168	8
18:50	EZY7607	Airbus A320	180	162	1
19:20	HV 6204	Boeing 737-700	149	135	2
19:20	TO 3417	Boeing 737-800	183	165	3
19:25	EZY3764	Airbus A319	156	141	4
20:20	EZY8720	Airbus A320	180	162	5
20:25	EZY2716	Airbus A319	156	141	6
20:35	FR 2096	Boeing 737-800	189	171	7