A SIMULATION STUDY ON BAGGAGE SCREENING AT SAN LUIS OBISPO COUNTY REGIONAL AIRPORT

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Industrial Engineering

by

Marian Ott

March 2021

COMMITTEE MEMBERSHIP

TITLE:	A Simulation Study on Baggage Screening at San Luis Obispo County Regional Airport
AUTHOR:	Marian Konstantin Ott
DATE SUBMITTED:	March 2021
COMMITTEE CHAIR:	Alessandro Hill, Ph.D. Assistant Professor of Industrial Engineering
COMMITTEE MEMBER:	Tali Freed, Ph.D. Professor of Industrial Engineering
COMMITTEE MEMBER:	Mohamed Awwad, Ph.D. Assistant Professor of Industrial Engineering

ABSTRACT

A Simulation Study on Baggage Screening at San Luis Obispo County Regional Airport

Marian Konstantin Ott

Efficient passenger flow is a crucial objective at both small and larger airports. One central part of this is the handling of checked luggage which is influenced by necessary security screening. Within this thesis, these processes are studied at San Luis Obispo County Regional Airport. The underlying problem of the airport is its outbound luggage system which was already suffering from delays prior to Covid-19. Delays were never measured and the bottleneck responsible for them was never identified. However, expected growth in passenger and flight volume necessitates to predict when customer dissatisfaction and extensive luggage delays are inevitable, given that the airport does not plan to change the baggage screening system in the near future.

In order to understand the dependencies within said system, process flowcharts for baggage-related activities are defined and translated into a simulation model. After model verification and validation, scenarios of expanding the flight schedule during different times of the day are tested while monitoring the number of bags failing to be loaded into the respective aircraft in time. Further scenarios of model adjustments are used to monitor how the number of missed bags changes while maintaining an expanded flight schedule. Model adjustments were made by changing single parameters such as the scan time or single resources each.

Simulation experiments have shown that the number of additional flights that can be added to the flight schedule of February 2020 depend on the time of the day. For instance, the current outbound luggage conveyor system's capacity is sufficient to cover 1 additional early morning flight, and up to 3 afternoon flights. Experiments with model parameter adjustments led to identifying the luggage scanner as the bottleneck of the luggage system, whereas other tested parameter adjustments showed to have minimal impact on the number of missed bags. Since the model's flight plan can be conveniently adjusted in the connected Excel database, the model could be used as a tactical decision tool for capacity analysis.

Keywords: Airport, Baggage screening, Simulation

ACKNOWLEDGMENTS

I want to express my deep appreciation to my advisor Dr. Alessandro Hill for his excellent guidance and advising. In addition, I would like to thank Dr. Tali Freed for her excellent guidance and support.

Further I want to thank Craig Piper, Deputy Director, and Kevin Bumen, former Airport Director for their patience in helping me completing this study.

Besides, I would like to thank Fulbright Germany for supporting me in going to graduate school in the United States of America.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

Airports operate as businesses that provide a service to commercial and private customers. In fact, they bring both parties - airlines and passengers - together. One component within their range of service is to provide the physical facility that acts as a knot in the global network of interconnected airports. Besides the rolling field where airplanes will take off and land they have one or more airport buildings with a single or multiple terminals. By bringing passengers and airlines together, they must also ensure that passengers' drop-off luggage arrives at the aircraft in time. Considering that travelling with an aircraft comes with the exception that luggage units are not with the passenger at all times, a system for luggage drop off and claim is required. Check-in counters are provided for the latter and allow the pieces of luggage to be introduced into the conveyor belt system. Following that, bags typically go through a TSA (Transportation Security Administration) scanning machine that ensures that no dangerous substances or forbidden items are in the bag. Depending on the size of the airport, there might be a complex conveyor system that sorts bags by flight and moves them to different locations within the airport. Eventually, they will arrive at a transshipment point where bags are loaded onto carts that move bags to the aircraft, where workers load them into the cargo space. It is the goal of the airport, airlines, and passengers that all of this is achieved up to the time when the plane is scheduled to roll to the takeoff runway. Otherwise, late luggage units that cannot be loaded into an aircraft in time, resulting in delayed takeoffs and related costs, cannot be avoided.

1.1 Motivation

A study conducted by (SITA, 2019) states that 4.27 billion bags were carried by airlines worldwide in 2018. 24.8 million of them were mishandled. Figure 1 shows the breakdown of these mishandled bags and reveals that a 77% share of them were delayed bags. The annual costs for the aviation industry that were associated with mishandled bags in 2018 were listed at \$2.4 billion. Delayed bags resulted in costs of \$1.85 billion, accordingly. Figure 2 shows the reasons for delayed bags in 2018. The reasons provided in the graphic that can be linked to baggage delays related to the airport's baggage handling are failure to load, security, and airport restrictions. These categories accounted for annual costs of \$759 million.

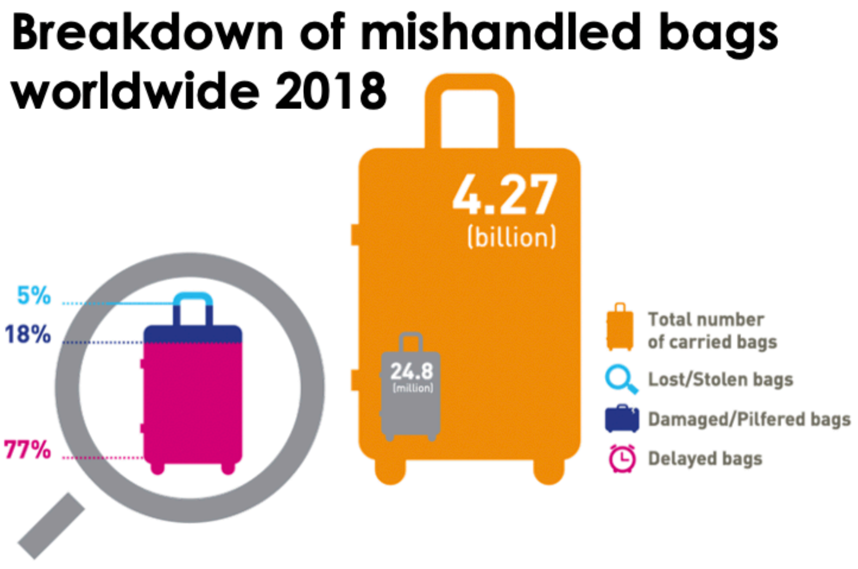


Figure 1: Breakdown of mishandled bags worldwide 2018 (SITA, 2019)

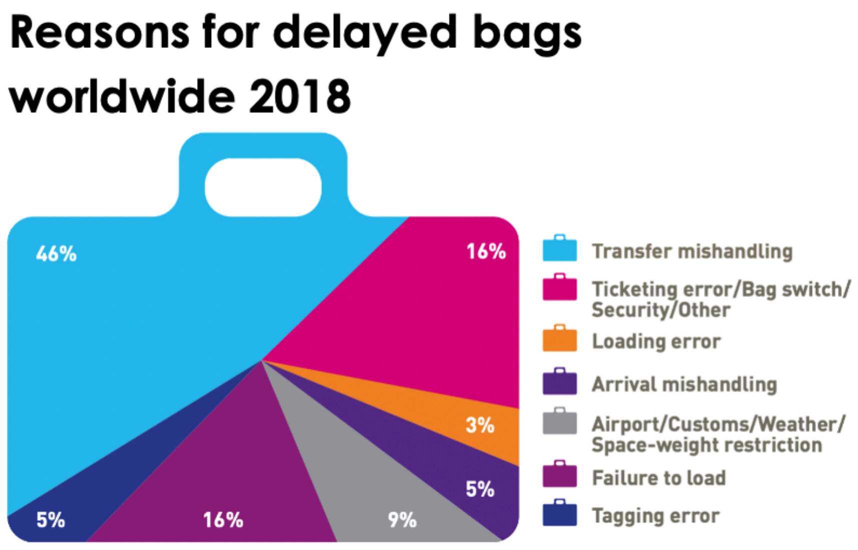


Figure 2: Reasons for delayed bags worldwide 2018 (SITA, 2019)

1.2 San Luis Obispo Regional Airport

San Luis Obispo Regional Airport is located in the central coast of California. The airport's single terminal which is shown in Figure 3 connects passengers to and from multiple destinations including Seattle, Portland, San Francisco, Los Angeles, San Diego, Phoenix, Denver and Dallas-Fort Worth as shown in Figure 4. These flights are offered by Alaska Airlines, American Airlines and United Airlines. The aircrafts in use are the Canadair CRJ-200, the Canadair CRJ-700 and the Embrear 175 with respective passenger capacities of 50, 68 and 78. Since the start of the ongoing global pandemic, the passenger volume, such as offered flights declined and continue to fluctuate depending on the demand that the airlines experience. However, prior to this, with 544,575 total passengers in 2019, SLO airport celebrated its busiest year to date. Compared to the amount of 485,911 passengers in 2018, this is a growth of 12.1%. At the end of 2019, close to 20 commercial flights departed from SLO airport on a daily basis. The expected annual growth of passenger traffic flow within North America will be 2.6% from 2019 to 2039 according to (Boeing, 2020).

Figure 3: San Luis Obispo Regional Airport terminal (San Luis Obispo Regional Airport, 2020)

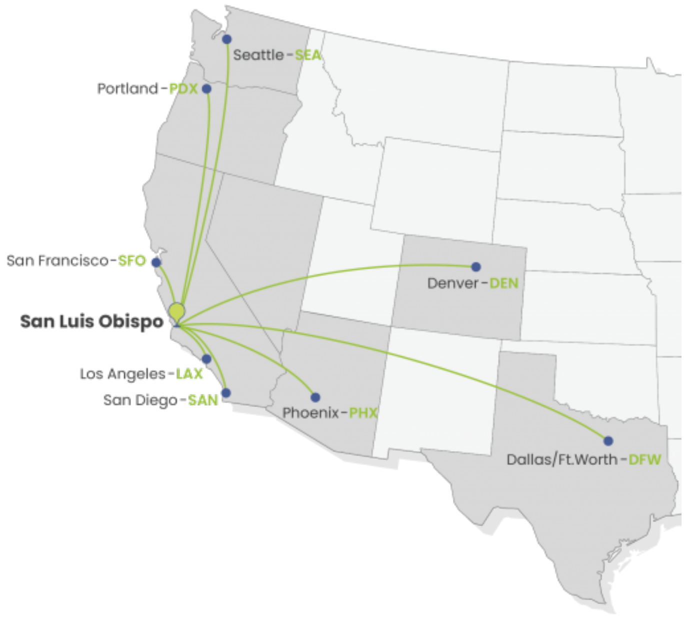


Figure 4: Direct connections of San Luis Obispo Regional Airport (San Luis Obispo Regional Airport, 2020)

1.3 Problem Statement

Despite Covid-19 effects SLO airport has been experiencing significant annual growth in the last decade and is expected to grow in number of passengers, flights and systems. Prior to Covid-19, the outbound luggage processing was already suffering from extensive delays leading to passenger dissatisfaction and an increased risk of takeoff delays. However, the airport aims to increase revenue by adding flights. Contradicting to this, they want to avoid to worsen the luggage issues by that. The delays have never been measured and the bottleneck of the luggage processing system has never been identified. Thus, this thesis focuses on developing a simulation model of the existing luggage handling system in order to determine capacity limits and potential capacity increasing methods. The following questions will be answered with the help of simulation:

- Which entity of the outbound luggage handling system is the bottleneck?
- How many additional flights are feasible to be included in daily operations until takeoff delays and customer dissatisfaction are unavoidable?
- How will the increased number of passengers impact the outbound luggage system?
- Which measures can compensate for the negative effects on the luggage system that come with significant growth of passenger and luggage numbers?

1.4 Literature Review

This chapter provides an overview of papers that handle comparable problems and analyses with simulations. It explains and provides justification for how general airport simulation is modeled with passenger arrival distributions, distributions of checked bags per passenger, counter passenger check-in, luggage scanning and which key performance indicators (KPIs) can be used.

1.4.1 Airport Simulation

Literature shows that for a lot of business cases where potential changes to a physical system are being considered, using simulations is often a good approach. For instance, simulation models are often used in evaluating system performance, for comparing alternatives and scenarios (Lin, Shih, Huang, & Chiu, 2015), (Cavada, Cortés, & Rey, 2017) and (Batta, Drury, Appelt, & Lin, 2007). The work of Savrasovs et al.

(Savrasovs, Medvedev, & Sincova, 2009) incorporates a simulation modelling approach of the baggage handling system of Riga International Airport in Latvia. This simulation model is used to compare several scenarios. Their motivation stems from the expected future growth of passenger numbers and the resulting effects on the baggage handling system, similar to the motivation for the work of (Cavada, Cortés, & Rey, 2017). Further, (Alodhaibia, Burdett, & Yarlagadda, 2019) use airport simulation to study how arrival patterns of passengers affect international terminal operations including check-in and security screening at Brisbane International Airport. They show that arrival patterns depend on the departure time of a flight and the passenger type. Postorino et al. point out that simulation models are within the most popular tools when designing airport terminals. This is because queueing related processes and human behavior can be modeled (Postorino, Mantecchini, Malandri, & Paganelli, 2019). (Van Dijk & Joustra, 2011) argue that simulation has the advantage of not relying on steady arrival rates as queueing theory does. It enables the use of random distributions and is therefore suitable to model arrival patterns with peaks, which occur, for instance, at airports. Further, the possibility to represent the simulation model logic in the form of animation helps to convey understanding on multiple levels. Based on two case studies at Amsterdam-Schipohl Airport in the Netherlands, they show that different check-in procedures affect average queue times. Further, they did a capacity analysis of the current system in order to identify the maximum possible growth in number of departing flights while maintaining the current check-in setup such as queueing time. (Cavada, Cortés, & Rey, 2017) have their work centered around their simulation of the baggage handling system at Santiago International Airport. General growth of passenger numbers in recent years lead to longer passenger queues, delayed flights and growing potential for the baggage handling process not being smooth. Within their simulation model, all baggage-related subsystems are incorporated. They argue that by looking at the system as a whole they are able to study impacts and interrelationships of different operating strategies. Further, they investigate ways of improving the baggage handling system of the international terminal of Santiago International Airport (SCL) as the system's infrastructure will not be upgraded in the near future. Wonkyu et al. studied how dwelling time distributions affect the number of passengers arriving at airport terminals in given times. Further, they give insights on how terminal operations are affected by significant flight schedule changes (Wonkyu, Yonghwa, & Byung Jong, 2004).

In conclusion, simulation models have several advantages such as allowing the gauging of effects resulting from different scenarios faster and cheaper compared to making changes to the respective real system – may it be a whole supply chain of a manufacturing company, or the outbound baggage handling system of an airport.

1.4.2 Passenger Arrival Distribution

This chapter covers methods and approaches for modeling realistic passenger arrivals used by researchers in different projects. Postoriono et al. describe the knowledge of passenger arrival rate functions as crucial for determining the correct amount of required resources at an airport (Postorino, Mantecchini, Malandri, & Paganelli, 2019).

Simulated arrivals at the airport terminal can be generated by statistical distributions that reflect real passengers' behavior (Alodhaibia, Burdett, & Yarlagadda, 2019). The amount of passenger's arrivals within a certain time interval in (Savrasovs, Medvedev, & Sincova, 2009) were determined by the Binomial distribution and the Bernoulli formula. While (Alodhaibia, Burdett, & Yarlagadda, 2019) list exponential, uniform, empirical, and normal distributions as appropriate for modeling passenger arrivals, they decided to use the normal distribution. The analysis on real passenger arrival data performed by (Postorino, Mantecchini, Malandri, & Paganelli, 2019) showed that the Weibull function was most accurate for modeling arrival processes. Cavada et al. model individual passengers' arrivals with the help of empirical distributions created from histograms of passengers' terminal arrivals (Cavada, Cortés, & Rey, 2017).

Passenger arrival patterns depend on a range of factors such as departure times, type of travelling (leisure or business), and flight destination (Alodhaibia, Burdett, & Yarlagadda, 2019). Similar information can be retrieved from (Cavada, Cortés, & Rey, 2017). Factors considered in (Postorino, Mantecchini, Malandri, & Paganelli, 2019) include the type of carrier, for instance, full carriers (FC), low-cost carriers (LLC) as well as the scheduled departure time.

The majority of passengers checks-in 40 to 60 minutes prior to the closing of check-in counters which is 20 minutes before scheduled aircraft departure (Savrasovs, Medvedev, & Sincova, 2009). According to (Alodhaibia, Burdett, & Yarlagadda, 2019), common arrival patterns at international terminals include 90% of travelers arriving 60 minutes prior to departure time and business travelers arriving later than leisure ones. They identified peak check-in times 100 to 120 minutes before departure, and state that peaks in the morning appear to be shorter but busier compared to the evening peaks. Postorio et al. present that most passengers arrive between 60 and 90 minutes prior to their flight (Postorino, Mantecchini, Malandri, & Paganelli, 2019).

Generally, passengers of early flights tend to arrive later than passengers of late flights, while the airport check-in rules significantly affect passenger arrival patterns (Alodhaibia, Burdett, & Yarlagadda, 2019), (Postorino, Mantecchini, Malandri, & Paganelli, 2019) and (Cavada, Cortés, & Rey, 2017). Figure 5 demonstrates historical arrival patterns that Cavada et al. collected depend on morning versus evening flights, such as domestic (Punta Arenas) versus international (Buenos Aires) destinations.

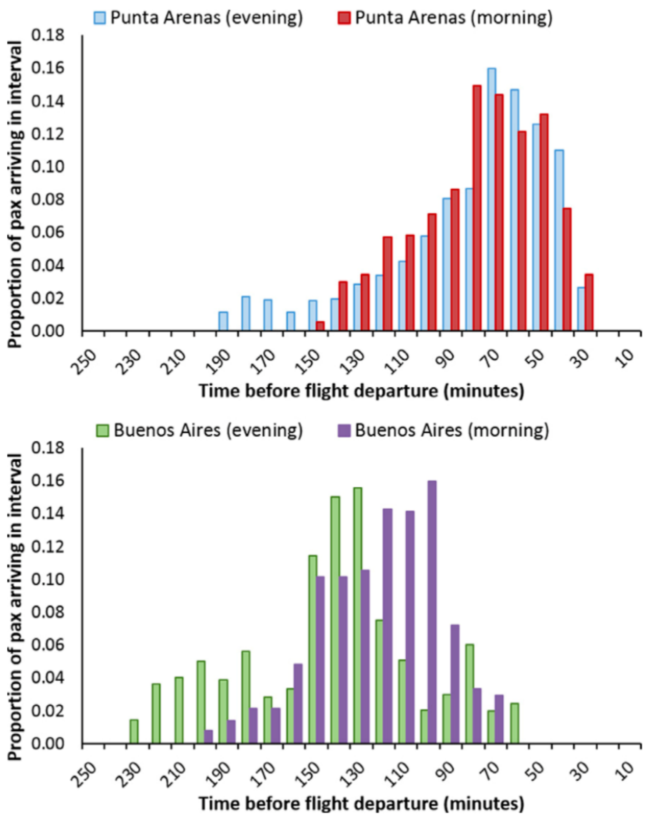


Figure 5: Histograms of passenger arrivals at check-in queues (Cavada, Cortés, & Rey, 2017)

Figure 6 visualizes the collected passenger arrival data in histograms while the estimated functions for passenger arrivals are shown as dashed and solid lines. Graphic a) represents the early morning, graphic b) the late afternoon. Travelers flying with full carriers (FC) are more likely to arrive closer to their departure compared to passengers travelling with low-cost carriers (LCC). This applies for the early morning and the late afternoon periods. However, a minor part of LCC passengers with late flights arrive significantly earlier than the FC passengers. (Postorino, Mantecchini, Malandri, & Paganelli, 2019)

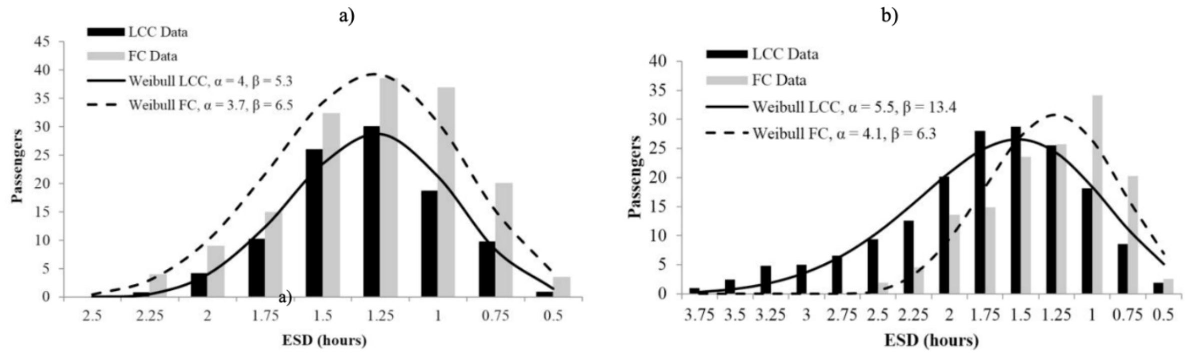


Figure 6: Histograms of passenger arrivals and functions (Postorino, Mantecchini, Malandri, & Paganelli, 2019)

The simulation model in (Cavada, Cortés, & Rey, 2017) determines individual passenger arrivals based on historical arrival patterns and are computed the following way: the cumulative density function over all time intervals prior to departure is built. This means that a_i is the cumulative share of passengers that usually arriving until the end of interval i. For determining which interval the passenger is assigned to, a uniform variable u between 0 and 1 is assigned. If $a_j \leq u < a_{j+1}$ the interval will arrive in interval j. A specific position within the respective interval is calculated by introducing another random variable called v which is between 0 and the length of the interval. Then, the arrival is computed by adding v to the start of the determined interval.

1.4.3 Checked Bags per Passenger

When studying aspects of a baggage handling system, the distribution for the amount of bags that passengers check before proceeding to the security area is just as crucial as the passengers' arrival patterns themselves. Especially if the capacity of the system is of interest.

An online survey performed by Ipsos (Ipsos, 2018) considered 4 traveler types – all combinations of personal and business travelers with ones that have domestic or international destinations. Most passengers of each type check one piece of luggage prior to going through security. It appears that the average number of checked bags for domestic personal and business trips is 1.2 each. For international personal trips the average number is 1.4 and for the business equivalent 1.7. The amount of luggage units Savranos and Medvedev assign per passenger was determined by empirical data: The likeliness for 1 bag per passenger was the highest with 65% while 0 and 2 bags have a likeliness of 15% each. 3, 4 and 5 bags only have shares of 3, 1 and 1%. This results in an average of 1.13 bags per passenger (Savrasovs, Medvedev, & Sincova, 2009).

1.4.4 Passenger Check-In

According to empirical data of (Savrasovs, Medvedev, & Sincova, 2009) a passenger's processing time at a check-in counter exclusively depends on the number of luggage units the passenger drops off. The processing time in seconds at check-in counters was described as $f(x) = \begin{cases} 30 + x * 15, x \leq 5 \\ 105, x > 5 \end{cases}$ where x represents the number of bags checked in. The work of (Batta, Drury, Appelt, & Lin, 2007) takes a more holistic view on different check-in types, for example, online check-in, curbside check-in, kiosk check-in and counter check-in. Statistical analysis was performed to identify parameters that have a significant impact on the duration of the different check-in types. For the indoor check-in counters it was found that the number of bags checked and group size are significant. Further, they state that there is a correlation of 0.595 with p-value < 0.001 between group size and number of bags checked. Both factors are therefore interchangeable, and data could be grouped by group sizes. The times they observed from the base model are as follows: a total average wait time in the queues of 24.3 seconds, which includes the averages for kiosk queues of 5.49 seconds and the average for counter queues of 27.74 seconds and more. The average total of waiting time and service time was 158.85 seconds.

1.4.5 Luggage Scanning

The share of bags that need additional checking after the automatic scan is 10%, while these bags have a probability of 3% for being restricted from being moved to the aircraft (Savrasovs, Medvedev, & Sincova, 2009).

1.4.6 Key Performance Indicators

(Savrasovs, Medvedev, & Sincova, 2009) investigates the utilization levels as their major KPI (key performance indicator). Specifically, they are looking at the mean utilization of check-in, additional security check and the sorting area. The average and maximum passenger queue length for all check-in counters combined was investigated as well. Another KPI that was studied is the baggage queue length in both the additional security check area, and the sorting area. As the simulation of (Batta, Drury, Appelt, & Lin, 2007) is dedicated to study variations of check-in procedures, they look at average times spent in queues by check-in type such as the total average time spent in the system.

2 Airport Infrastructure, Resources and Processes

As suggested by the problem statement, a specific part of daily operations at SLO airport, the outbound luggage system, is the center of this analysis and simulation project. The following sections provide an overview of the airport's infrastructure and resources and will lead to a detailed description of processes that are included in the outbound luggage process. Specific measures of resources are introduced in chapter 3.2

2.1 Infrastructure and Resources

A range of essential activities is required to allow passengers to depart from SLO airport. All of these activities and processes are made feasible by the airport's infrastructure and resources. The processes that are relevant to the outbound luggage system will be listed and described within this chapter.

Figure 7 provides an overview of the commercial airport section and indicates where the essential activities take place. Gray arrows visualize the typical passenger flow starting with them arriving in the terminal building and proceeding to the check-in counters which are displayed as orange rectangles. This is where they check-in and give their luggage to the airline employee. The passengers proceed to walk to the security and scanning area where their identity such as their carry-on luggage is checked. Following that, they proceed to the gates and boarding section where they wait to finally board their flight.

Red arrows indicate the path of checked luggage through the airport. After being dropped-off, the luggage unit will be introduced into the outbound luggage conveyor system which is sketched as connected gray rectangles in Figure 7. A detailed representation of the conveyor belt system is shown in Figure 9. Once baggage arrives in the TSA room it will be scanned in the TSA scanner which is indicated by the left rectangle that the luggage scanning arrow points to. Certain bags will be searched on the manual search table represented by the right rectangle indicated by the luggage sorting arrow. All bags found clear by the scanner or the manual search are conveyed further.

Upon arrival at the end of the conveyor belt, bags are sorted by flight in the light blue luggage sorting area at the end of the conveyor belt. Before getting loaded into an aircraft, bags are transported to one of the six aircraft parking positions around the gates, see Figure 8. Considering these positions, the transportation of luggage carts from the luggage sorting station to the aircraft greatly depend on the parking position of the latter.

The focus of this thesis is the outbound luggage system with passenger arrival. This means that passengers' activities after luggage drop-off, the security scanning of them, such as their movement to gates, and boarding are not considered as they are not impacting the luggage system.

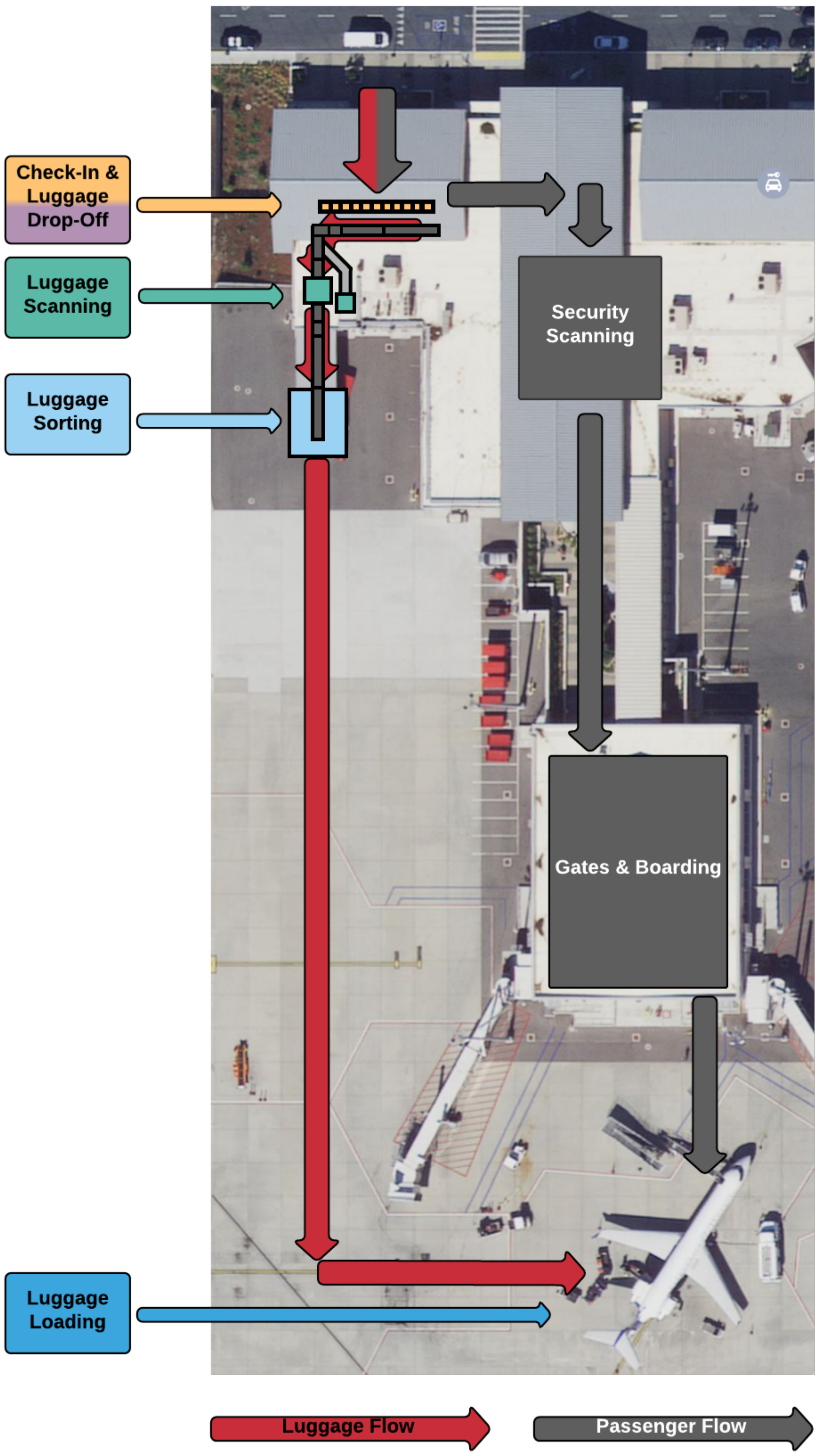


Figure 7: Overview of the commercial part of SLO airport

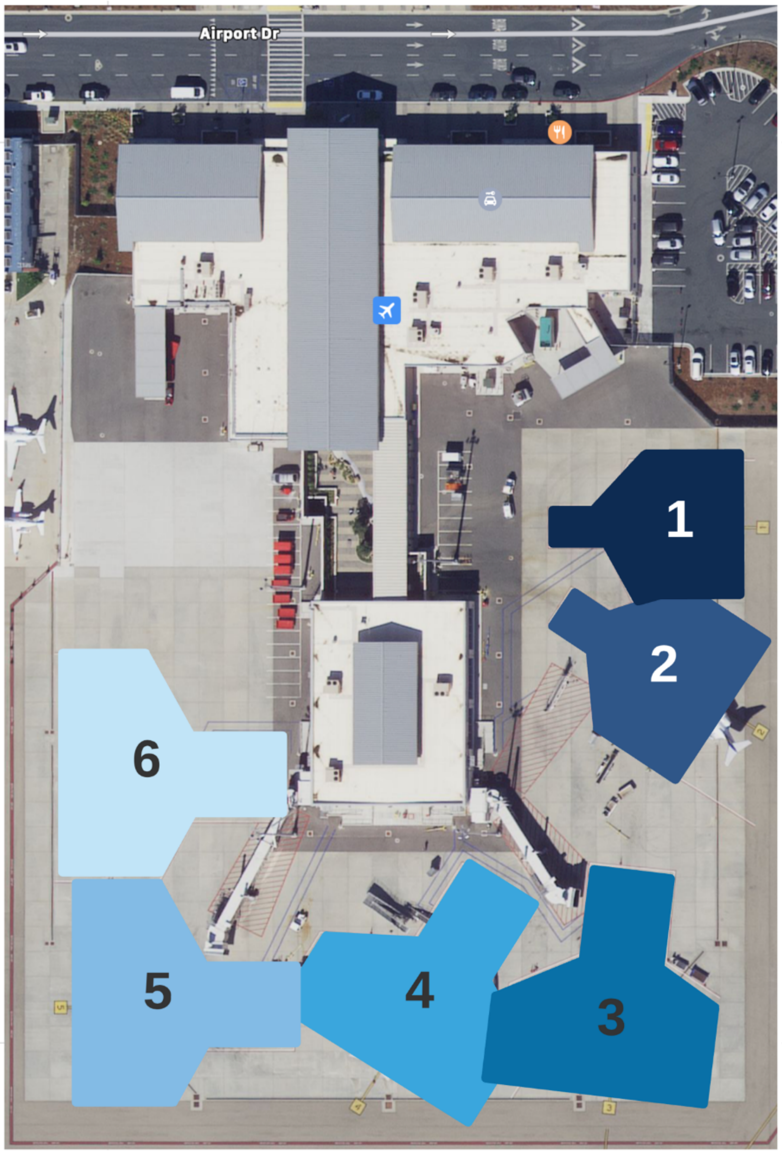


Figure 8: Aircraft parking positions at SLO airport

Figure 9 visualizes the outbound luggage conveyor system with more detail and specific dimensions. The used colors refer to the same processes as in Figure 7. Therefore, the orange-purple rectangles stand for check-in counters where passengers stay on the orange side, while airline employees work on the purple side. The automated TSA scanner such as the manual search table are displayed as the mint rectangles. As indicated before, the light blue rectangle stands for the sorting station. The light gray colored arm of the conveyor system intended to be used for big bags, however it is not in use and the airport does not plan to do so in the near future.

Two horizontal lines in black indicate how the system is separated by walls that each have a gate for the conveyor belt. Braces on the right side of Figure 9 show that the purple section is within the publicly accessible airport terminal. The mint-colored section is a separate TSA room while the blue area is outside of the airport.

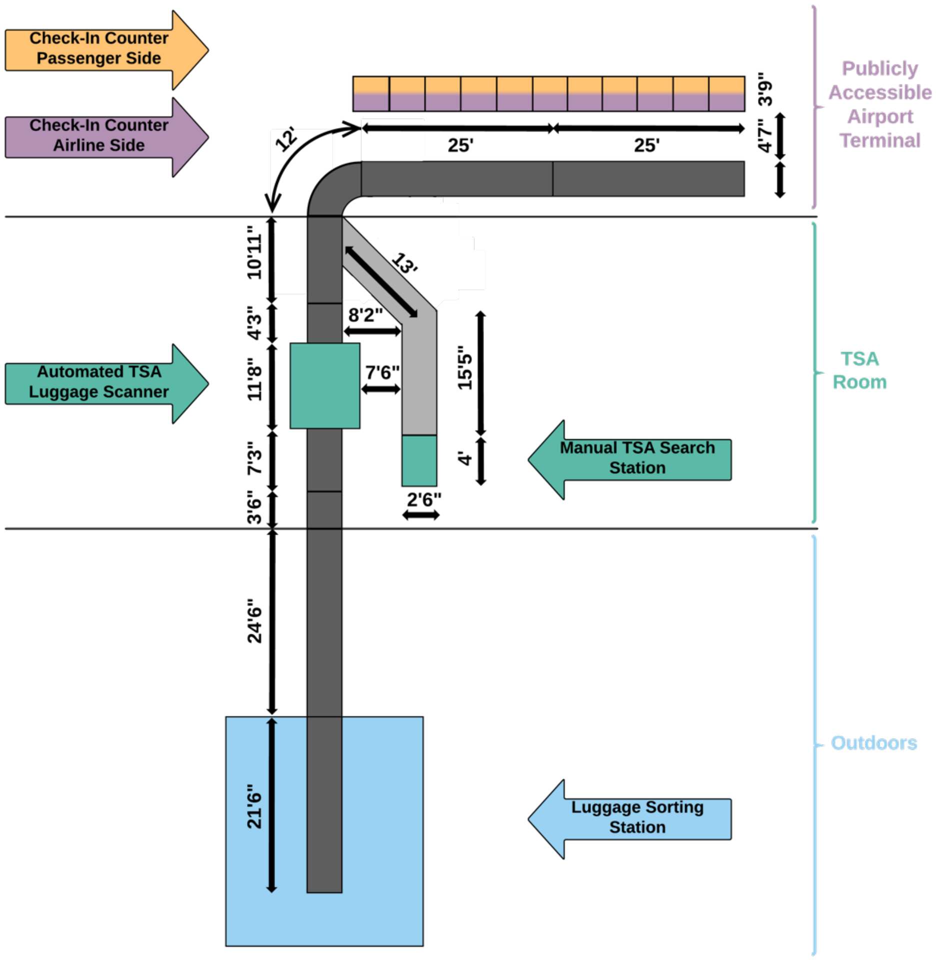

Check-In Counter Passenger Side
Check-In Counter Airline Side
Publicly Accessible Airport Terminal
3'9"
4'7"
12'
25'
25'
10'11"
13'
4'3"
8'2"
Automated TSA Luggage Scanner
11'8"
7'6"
15'5"
TSA Room
Manual TSA Search Station
7'3"
4'
3'6"
2'6"
24'6"
Outdoors
21'6"
Luggage Sorting Station

Figure 9: The outbound luggage conveyor system

2.2 Detailed Process Description

The following process flowcharts serve two major purposes: first, they are meant to help build a consensus about the underlying outbound baggage process for all readers. Second, they provide a base on which the simulation model can be built. Every passenger, including their luggage runs through this process chain, likely with varying sub-paths. The circular shapes indicate the start or the end of the total process chain, rectangles with vertical lines on the insides represent predefined processes. Predefined processes are defined on a detailed scale at a later point. Regular rectangles stand for processes. The Rhombuses indicate a decision with "Yes" and "No" as possible outcomes. Depending on the outcome, the respective arrow is followed. Processes may also vary from passenger to passenger or luggage unit to luggage unit.

Considering the chronological order of the outbound luggage process chain with a low level of detail, five essential predefined process blocks are included after passenger arrival at the airport as shown in Figure 10. The first predefined process takes the perspective of passengers who will check-in and possibly drop-off their luggage units. Following that, the check-in process from the airline employee's perspective is covered in the purple predefined process. Dropped-off luggage units will be scanned by the TSA which is covered in the third predefined process block. The final two predefined processes are from the perspective of the airline employees' that take care of luggage sorting and loading. Even though the processes have been separated for uncluttering reasons, they are both executed by the group of employees of the different airlines. Second to last, there is the predefined flight-specific bag sorting process. Finally, the last predefined process includes luggage loading into the aircraft. The following sub-chapters zoom into these five predefined processes.

All of the following flowcharts have been designed to be easy and consistent to read. Due to this, the decision blocks' "Yes" arrows always face downwards while "No" arrows go to the right with few exceptions where they go to the left. This can lead to decision blocks being phrased negatively. Besides, the flowcharts mostly go to downwards or to the right.

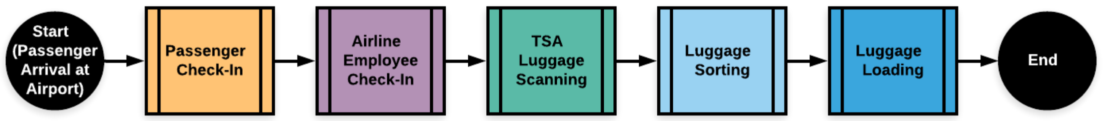


Figure 10: Process flow overview

2.2.1 Passenger Check-In

For the detailed level of the predefined passenger check-in process visualized in Figure 11, we look at the processes and decisions from the passenger's perspective.

The first thing determining a passenger's action after their arrival at the airport is if they did an online check-in. If yes, further actions are influenced by the passenger having drop-off luggage or not. In case of not having any, the passenger's process ends, as they do not need to go to a check-in counter. In case the passenger did not check-in online, the rest of their specific path in the flowchart depends on their decision of using a check-in kiosk or not.

Technically, only the passengers that do not have drop-off luggage should use a kiosk for check-in as the they do not provide a possibility for luggage drop-off. However, observations have shown that some individuals still check-in at kiosks when carrying drop-off luggage. This requires them to drop-off bags at a check-in counter nevertheless. If the traveler decides to go to a kiosk, he will check himself in. Self-check-in passengers that do not have drop-off luggage will leave the process as they are no longer impacting the outbound luggage system. Travelers that did a self-check-in who carry luggage to drop-off will go to the check-in queue of his airline check-in counters. Note that these travelers join the arm of the flowchart that passengers who have not checked-in online or at the kiosk go through.

Travelers like these need to go to the check-in queue. Airlines at SLO airport offer specific time windows for check-in, which begin two hours before departure of a flight and end 30 minutes prior to departure. Passengers who try to check-in later than this will be denied and asked to leave the counter. Thus, the process for them ends as well. Passengers at the counter during the window are separated into either already being checked-in or still needing to be checked-in. The travelers who do not need to receive check-in services from the airline employee will skip to the luggage drop-off. Passengers needing the check-in services will receive them from the airline agent. The specific services included in the check-in depends on the individual passenger. For example, he could possibly still require his boarding pass, or he could have obtained it already via online check-in. Besides, he might have to pay for a seat or service he did not select when booking his ticket. All of this will have an impact on the processing time at the check-in counter.

The travelers will then either drop off one or multiple bags to the check-in employee or leave the process for previously mentioned reason. One or multiple bags could be overweight and require the traveler to pay an additional fee, resulting in a longer processing time. After luggage drop-off the traveler exits the system and proceeds to go to the TSA security screening, where they no longer impact the outbound luggage system.

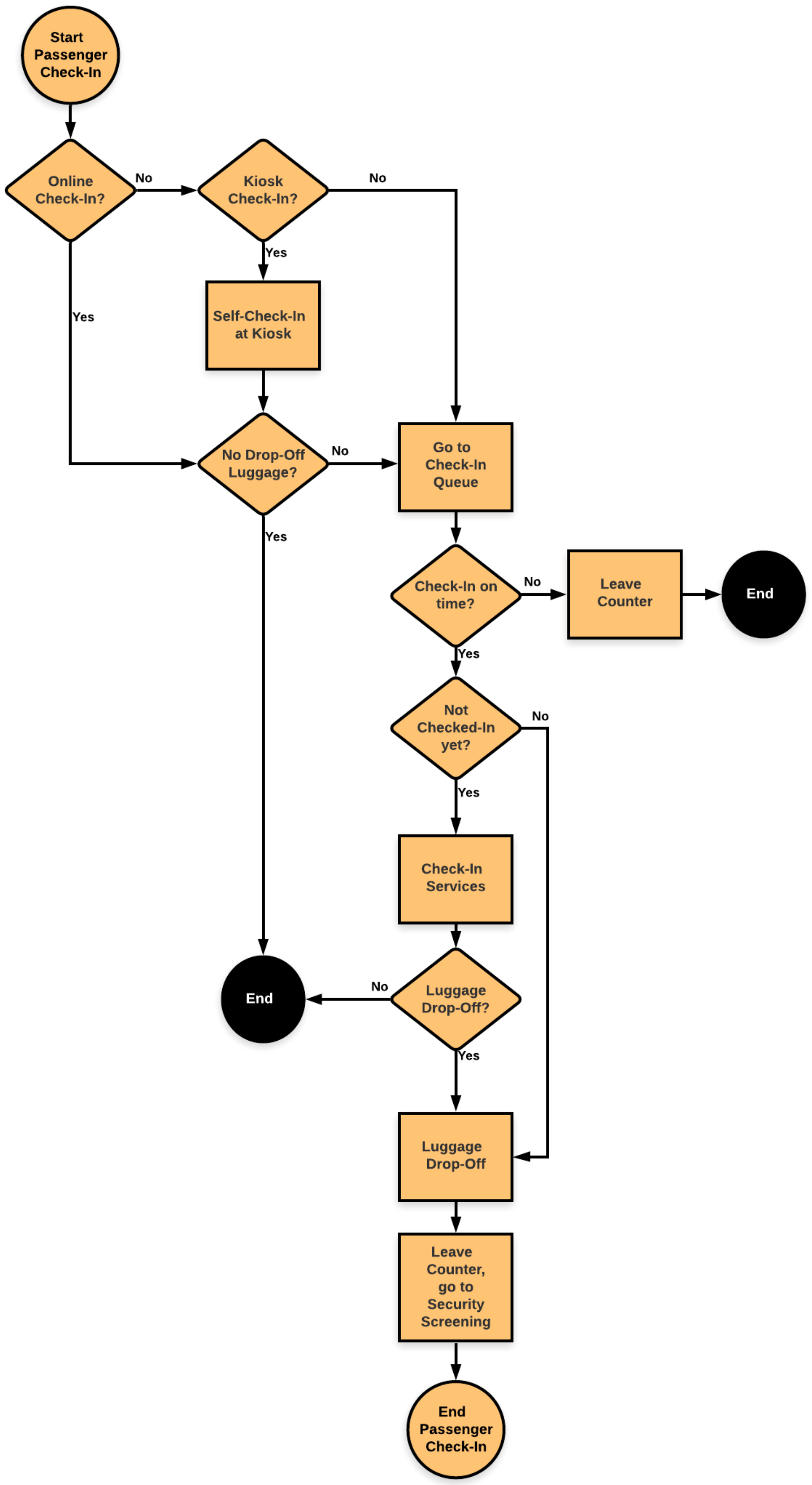


Figure 11: Passenger check-in process in detail

2.2.2 Airline Employee Check-In

Figure 12 visualizes a more detailed level of the purple predefined airline employee check-in process, in which we look at the processes and decisions from the airline employee's perspective.

An airline employee's actions will be determined by the counter queue. If no passengers are waiting for service, the airline employee will check for bags that have been temporarily stored on the floor between the check-in counter and the conveyor. An explanation for bags being stored on the floor is included in the process description at a later point. The airline employee will also check if there is a free spot on the conveyor. If both questions were answered with "Yes" the stored bag will be placed onto the conveyor and the airline agent will go back to his counter. Following that, actions will again be determined by the check-in queue.

If passengers wait for service, the airline employee will verify that the passenger is checking-in within the allowed time window. Passengers that are too late will be denied and leave the counter because the process ends. The airline employee will provide different services to in time travelers depending on if they still need to be checked-in or not. The ones being checked-in already will receive luggage drop-off services. If the passenger has not checked-in yet he will receive the appropriate services. Consequently, the process ends if the respective passenger does not have drop-off luggage. At the same time, travelers with luggage will receive the drop-off service.

At this point, the actions of the airline employee are related to luggage units, not the passengers any-more. The way the airline employee handles baggage depends on its size. Oversized luggage items cannot be transported on the conveyor belt or scanned in the TSA scanner. Thus, they are carried into the TSA room where TSA agents will receive them. Hereafter, the airline employee will go back to their counter and the process ends.

Regular sized bags will be carried to the conveyor behind the counter where the conveyor is checked for free space. The belt could already be packed with bags due to a congestion stemming from a point further down the belt. If this is the case, the airline employee will temporarily store the baggage on the floor and then go back to the counter and the process ends.

If the conveyor belt is free, the bag will be placed on the belt. Usually, when there are no bags on the conveyor belt, it is not moving. Thus, the airline employee will push a button in order to start it. Once a bag is placed and being moved on the conveyor, its remaining transportation is automated. Yet again, the airline employee’s actions depend on two decision blocks: are there temporarily stored bags on the floor and is the conveyor free? If both decision blocks are left though the “Yes” port, the stored bag will be put onto the conveyor. If either decision blocks’ outcome is “No”, the airline employee will go back to the counter and the process ends.

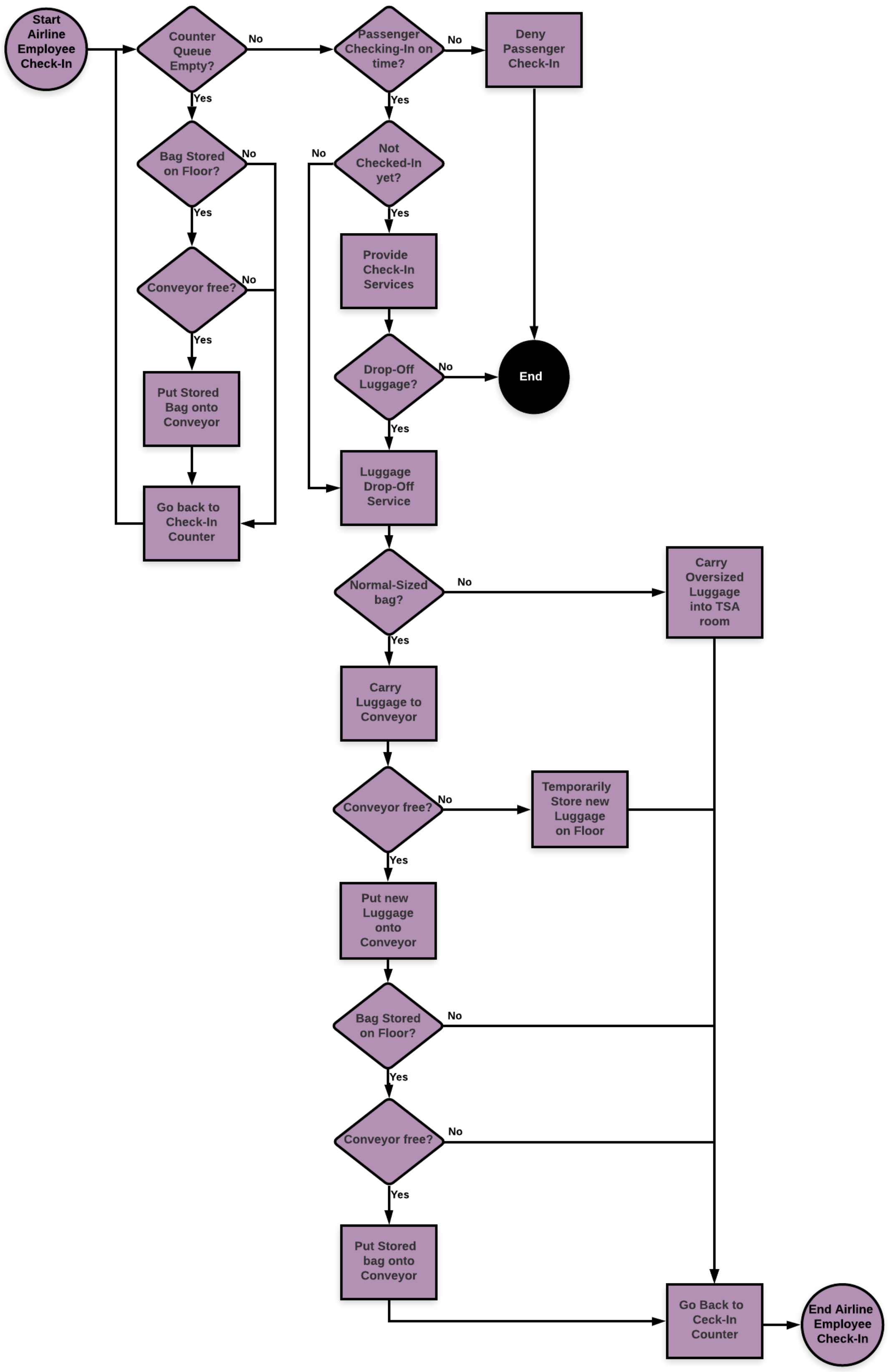

Start Airline Employee Check-In
Counter Queue Empty?
No
Yes
Passenger Checking-In on time?
No
Yes
Deny Passenger Check-In
Bag Stored on Floor?
No
Yes
Not Checked-In yet?
No
Yes
Conveyor free?
No
Yes
Provide Check-In Services
Put Stored Bag onto Conveyor
Drop-Off Luggage?
No
Yes
End
Go back to Check-In Counter
Luggage Drop-Off Service
Normal-Sized bag?
No
Yes
Carry Oversized Luggage into TSA room
Carry Luggage to Conveyor
Conveyor free?
No
Yes
Temporarily Store new Luggage on Floor
Put new Luggage onto Conveyor
Bag Stored on Floor?
No
Yes
Conveyor free?
No
Yes
Put Stored bag onto Conveyor
Go Back to Ceck-In Counter
End Airline Employee Check-In

Figure 12: Airline employee check-in process in detail

2.2.3 TSA Luggage Scanning

Since the end of the airline employee check-in process, bags have been transported into the TSA room by the conveyor belt. All of this predefined process of TSA luggage scanning is from the TSA agent's perspective.

As visualized in Figure 13, the working TSA agent checks if there are new bags arriving. If not, he will consider if there are temporarily stored bags next to the scanner and if the conveyor behind the Scanner is free. In the event of both questions being answered with "Yes", the stored bag will be put onto the conveyor and the TSA agent will check for arriving bags again. An explanation for bags being temporarily stored next to the scanner is included in the process description at a later point.

There is a significant difference in how regular sized and oversized bags are handled, provided that an arriving oversized bag is brought into the room by an airline employee instead of arriving via conveyor. The TSA agent will take and carry the oversized bag to the manual search station, perform a manual search and determine if the baggage is clear or not. Non-clear bags result in another series of events that have been summarized into the additional predefined process called "TSA non-clear bags", which is defined at the end of this sub-chapter. Clear bags are then carried back to the conveyor behind the scanner. These bags join the flow of regular sized ones.

Regular sized bags are transported into the TSA room on the conveyor. Each luggage item will stop on the conveyor next to where the non-used conveyor arm is attached to the regular conveyor, shown in Figure 9. Further conveyor transportation of each bag to the TSA scanner must be initialized by the TSA agent by pressing a foot pedal which is located where the non-used arm of the conveyor starts. Following this is the automated scan of the luggage item. The automatic scan determines if the bag is clear from security-threatening objects and gives the TSA agent an alert if the bag possibly contains such an object. Every bag is transported out of the scanner automatically, however, process steps afterwards depend on the clear status. Non-clear bags result in the bag having to be searched manually to clarify if the machine made a valid alert. That requires the TSA agent having to pick up the bag behind the scanner and carry the bag to the manual search station, where the process flow is similar to that of searching oversized bags.

Given that there is no congestion of bags on the conveyor, luggage that the scanner considered clear is further transported, eventually leading it to the outside sorting area. Bags that have been searched manually will be placed back onto the conveyor before getting transported outside. Considering a situation with a luggage congestion behind the scanner, finished bags will be temporarily stored behind the scanner next to the conveyor. The TSA agents have capacity to store a small number of bags, allowing them to process some additional arriving bags, even if a backlog is present. Those luggage items are put onto the conveyor after the newly processed bags are finished.

This paragraph elaborates on the additional predefined "TSA non-clear bags" process, which is visualized in Figure 14. A manual search that finds a bag containing an actual threat is rare. It is differentiated between non-severe cases where a passenger "accidently" left a threatening object, such as a loaded gun, in their luggage. Having a gun in a drop-off bag would require the gun to be stored in a safe case, to be unloaded, and to be announced to the airline. For the loaded gun example, the TSA agent would call a coworker who takes over the following processes in order to prevent the luggage handling system getting impacted with a major delay. Law enforcement will be called, the passenger will be pulled out and might miss his flight or be denied boarding his flight at all. This would result in the bag not being reintroduced into the conveyor system. If the TSA decides that the passenger is still allowed to board, his bag would be reintroduced into the conveyor system. An example for severe violation of rules scenario leads to the TSA shutting down the entire airport which would end the entire process chain for a significant time.

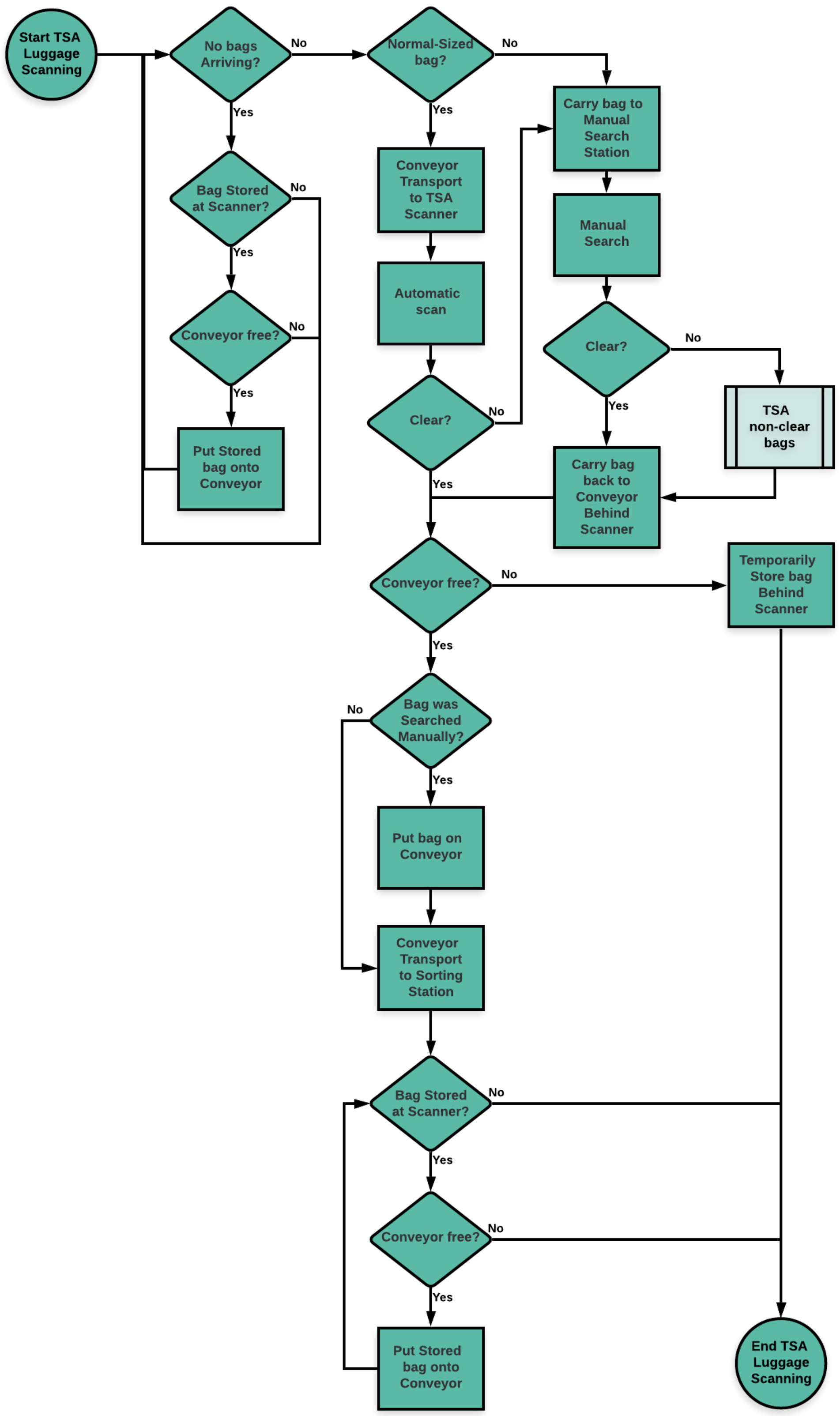

Start TSA Luggage Scanning
No bags Arriving?
No
Yes
Normal-Sized bag?
No
Yes
Carry bag to Manual Search Station
Manual Search
Bag Stored at Scanner?
No
Yes
Conveyor Transport to TSA Scanner
Automatic scan
Conveyor free?
No
Yes
Clear?
No
Yes
TSA non-clear bags
Put Stored bag onto Conveyor
Clear?
No
Yes
Carry bag back to Conveyor Behind Scanner
Conveyor free?
No
Yes
Temporarily Store bag Behind Scanner
Bag was Searched Manually?
No
Yes
Put bag on Conveyor
Conveyor Transport to Sorting Station
Bag Stored at Scanner?
No
Yes
Conveyor free?
No
Yes
Put Stored bag onto Conveyor
End TSA Luggage Scanning

Figure 13: TSA luggage scanning process in detail

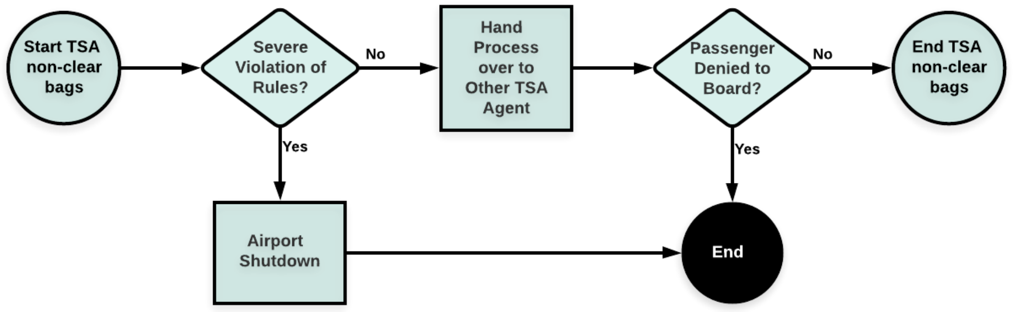

Start TSA non-clear bags
Severe Violation of Rules?
No
Yes
Hand Process over to Other TSA Agent
Passenger Denied to Board?
No
Yes
End TSA non-clear bags
Airport Shutdown
End

Figure 14: TSA non-clear bags process in detail

2.2.4 Luggage Sorting

Within the predefined luggage sorting process flowchart, the perspective of the airline employees that handle luggage movements outside is adopted. It is important to note that these airline employees also take care of luggage loading. The airline employee at the sorting station checks if bags arrive at the end of the conveyor belt.

As shown in Figure 15, a bag's label is checked upon arrival in order to determine which destination the bag has. Having this information in mind, the airline employee checks if a matching luggage cart is available. Each luggage cart will be loaded with baggage for one single flight in order to avoid bags ending up in the wrong aircraft. Depending on the availability of a matching luggage cart, the airline employee will either scan the bag's label for documentation and put the bag on the luggage cart or temporarily store the bag on the floor. Taking a bag off of the conveyor, even if no matching cart is available, prevents the laser sensor at the end of the conveyor belt from stopping the last section of the belt. Thus, baggage congestions that impact the rest of the conveyor system are avoided. Usually, however, a sufficient amount of matching luggage carts is available.

Supposing that no luggage is arriving, two things are considered. Is there a bag stored on the floor and is there a matching luggage cart available? Granted that both considerations apply, the airline employee will scan the bag's label and put it on the luggage cart.

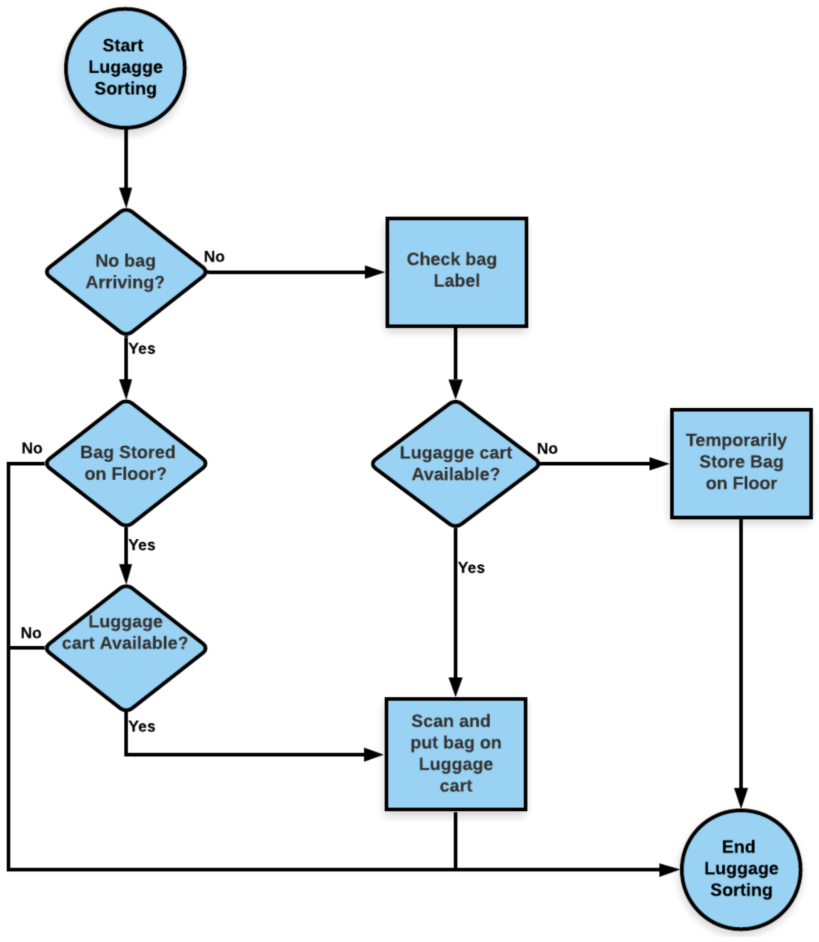


Figure 15: Luggage sorting process in detail

2.2.5 Luggage Loading

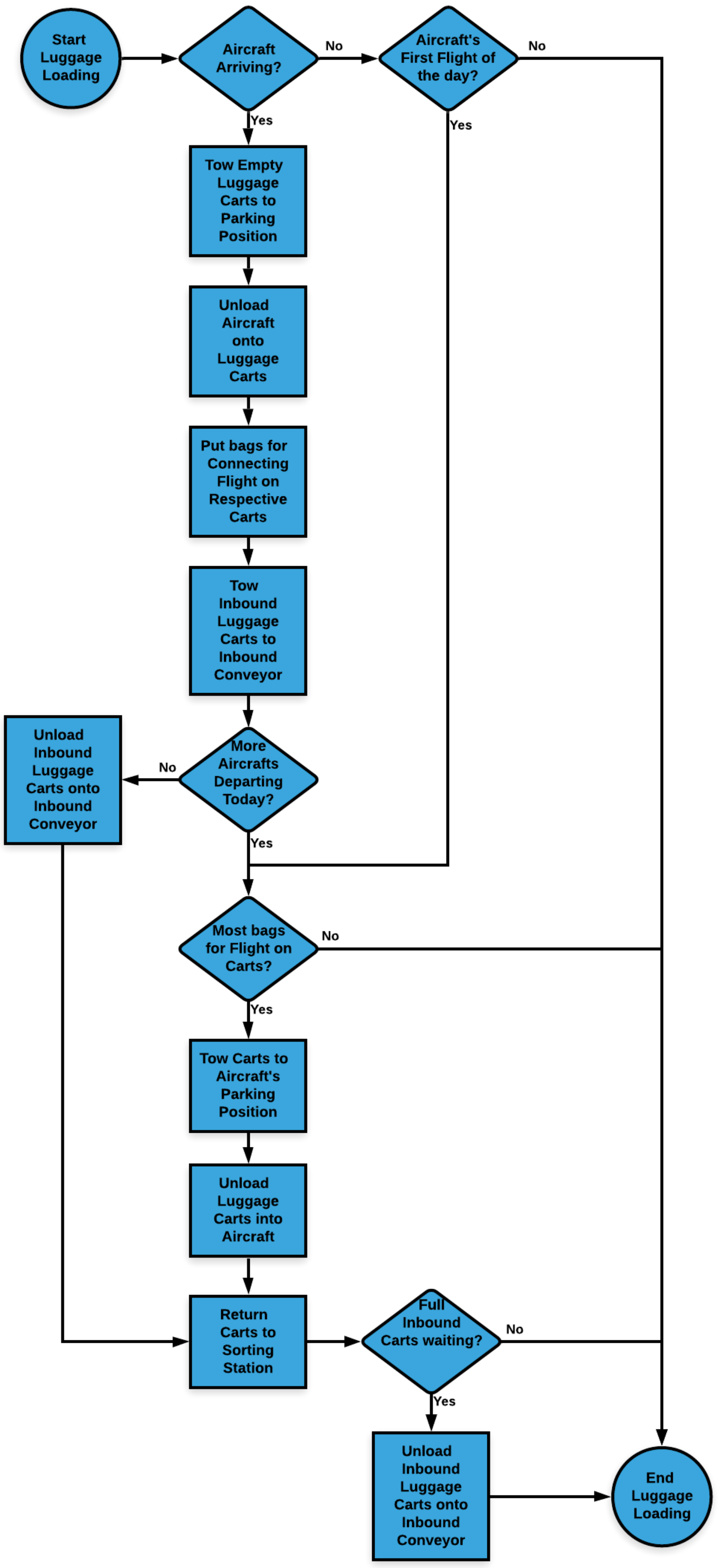
The luggage loading process is carried out by the same airline employees that handle the luggage sorting. Hence, their perspective is adopted for the luggage loading process flowchart as well. Generally, the airline employees that load departing aircrafts are also in charge of unloading incoming flights. In order to prevent departure delays, their priority is to unload arriving aircrafts first so that they can load them for their departure right afterwards. Figure 16 presents the process flowchart in detail.

Therefore, the first consideration is if an aircraft is arriving. Granted that this is not the case, it might be the early morning before the departure of aircrafts that spent the night at SLO airport. If this is true, the process flow for arriving aircrafts will be joined.

Suppose that there is an aircraft arriving, empty luggage carts are towed to the assigned parking position. Following that, the aircraft's inbound luggage is unloaded onto the luggage carts. Any bags that will go on a connecting flight will be put on the respective carts. Filled carts are then moved to the start of the inbound luggage conveyor.

Further processes depend on if more aircrafts have their departure scheduled today. In other words, if the last departure has already taken place, the luggage carts that were previously placed at the start of the inbound conveyor will be loaded onto the conveyor. This is where the process flow joins the arm of the flowchart that is used if more aircrafts are departing that day.

If more departures are scheduled, the airline employee will tow the carts to the aircraft’s parking position, given that most bags for this flight are on the carts. Then, the luggage carts are unloaded into the aircraft. Subsequently, luggage carts are returned to the sorting station. Luggage carts for flights that do not have most bags already will wait. Finally, if there are full inbound luggage carts waiting at the inbound conveyor as it was the priority to load a departing aircraft first, they will be unloaded onto the inbound conveyor.



Start Luggage Loading
Aircraft Arriving?
No
Aircraft's First Flight of the day?
No
Yes
Yes
Tow Empty Luggage Carts to Parking Position
Unload Aircraft onto Luggage Carts
Put bags for Connecting Flight on Respective Carts
Tow Inbound Luggage Carts to Inbound Conveyor
Unload Inbound Luggage Carts onto Inbound Conveyor
No
More Aircrafts Departing Today?
Yes
Most bags for Flight on Carts?
No
Yes
Tow Carts to Aircraft's Parking Position
Unload Luggage Carts into Aircraft
Return Carts to Sorting Station
Full Inbound Carts waiting?
No
Yes
Unload Inbound Luggage Carts onto Inbound Conveyor
End Luggage Loading

Figure 16: Luggage loading process in detail

3 Simulation Approach

The name of the software used to build the simulation model of the outbound luggage handling system is Anylogic by Anylogic Company. More specifically, the free trials of the Personal Learning and the University Edition were used. Anylogic was chosen because the author of this thesis has worked with Anylogic as part of a lecture. It is capable of modeling agent-based, discrete event, and system dynamics methodologies. Similar to a lot of simulation models, the one that has been built for this thesis is a hybrid of agent-based simulation and discrete event simulation. According to Anylogic, agent-based simulation focuses on active components of a system. These active entities are called agents and could represent people, households, vehicles, equipment, products, companies or other entities. Agents might be interconnected, and their behavior depends on common variables and parameters. Discrete event simulation is process-centric where systems are viewed as a chain of processes. Models are visually specified as process flowcharts (Egor Yakovlev, 2019).

3.1 Overview

The simulation model is based on the process descriptions that were introduced in chapter 2. However, the data that that was available to build the model upon was limited. As no data on the luggage sorting and loading processes was accessible, is has been decided to not include these aspects in modeling the system. This means that the simulation covers anything in between passengers arriving at the airport and luggage items arriving at the end of the conveyor belt. Due to the major question this thesis is trying to answer—what amount of additional flights is the conveyor system able to handle without causing major delays—one run of the simulation model simulates one whole day of operations.

3.2 Input Data

Simulation models are built to answer questions by obtaining data. This data can be determined by the model as output data. Any simulation model's output data greatly depends on the input data that the model is being provided with. The introduction of input data in the following sub-chapters distinguishes between data that is stored externally but accessed by Anylogic, such as data stored within the simulation model. Throughout the development of the simulation model, SLO airport kindly provided insightful data from February 2020. This data includes the scheduled flight plan corresponding with two datasets. One dataset contains the daily number of luggage items that passed though the TSA scanner by the hour. The other dataset contains daily number of persons that went through the security check by the hour. However, neither of the datasets is precise, as oversized bags are not counted. At the same time, the actual passenger throughput is smaller than what the data shows. This is because non-passengers passing through the security check are counted, too. These include cleaning staff, bistro staff, persons that assist passengers in the airport and more. Flight schedules usually deviate from day to day. Days with a large number of flights were selected from the flight schedules, on which the flight schedule is identical. This specific flight plan occurred on five days and is presented in the next chapter.

This results in five sets of time series of luggage and person throughput. Simultaneously to the model development, the model's input data was selected so that the simulated equivalent of these time series are matching on a sufficient level of similarity.

Input data presented in the following chapter is from the final simulation model that has been calibrated to create simulated luggage and person throughput time series that are sufficiently similar to the real time series.

3.2.1 Externally Stored

The flight schedule of departing flights is stored in an Excel-file and represents one of the most crucial inputs of the simulation model. It lists every departing flight per day. As seen in Table 1, information like departure time, destination, airline, the aircraft in use and its capacity, the average passenger load factor, and the start and end of the check-in time window are included. Time-related information is shown in the 24-hour format. The average passenger load factor values are destination-based averages from 2019. They represent the share of passengers that typically board the plane compared to available seats. Load factors had to be increased by 8.5% across the board in order to simulate a realistic amount of total passengers.

Table 1: Flight schedule of San Luis Obispo Regional Airport February 2020

Flight ID	Departure	Destination Code	Destination	Airline	Flight Code	Aircraft	Passenger Capacity	Average Passenger Load Factor	Start Check-In	End Check-In
1	05:15	DEN	Denver	United	5676	Canadair CRJ-200	50	87,60%	3:15	4:45
2	06:00	LAX	Los Angeles	United	5626	Embrear 175	78	65,40%	4:00	5:30
3	06:05	DFW	Dallas Fort Worth	American	3013	Canadair CRJ-700	68	85,20%	4:05	5:35
4	06:33	PHX	Phoenix	American	3041	Canadair CRJ-700	68	78,90%	4:33	6:03
5	06:41	SFO	San Francisco	United	5667	Canadair CRJ-700	68	70,70%	4:41	6:11
6	08:10	LAX	Los Angeles	United	5646	Canadair CRJ-200	50	65,40%	6:10	7:40
7	09:00	DEN	Denver	United	5830	Canadair CRJ-200	50	87,60%	7:00	8:30
8	10:19	SFO	San Francisco	United	5966	Canadair CRJ-200	50	70,70%	8:19	9:49
9	11:10	LAX	Los Angeles	United	5382	Canadair CRJ-200	50	65,40%	9:10	10:40
10	12:22	PHX	Phoenix	American	3220	Canadair CRJ-700	68	78,90%	10:22	11:52
11	12:50	DEN	Denver	United	5794	Canadair CRJ-200	50	87,60%	10:50	12:20
12	12:55	SEA	Seattle	Alaska	2106	Embrear 175	78	84,00%	10:55	12:25
13	14:05	SFO	San Francisco	United	5657	Canadair CRJ-700	68	70,70%	12:05	13:35
14	14:10	SAN	San Diego	Alaska	3498	Embrear 175	78	80,50%	12:10	13:40
15	16:03	SFO	San Francisco	United	5839	Canadair CRJ-200	50	70,70%	14:03	15:33
16	16:31	PHX	Phoenix	American	3056	Canadair CRJ-700	68	78,90%	14:31	16:01
17	18:10	LAX	Los Angeles	United	5644	Canadair CRJ-200	50	65,40%	16:10	17:40
18	19:14	PHX	Phoenix	American	3053	Canadair CRJ-700	68	78,90%	17:14	18:44

As seen in Figure 17, this schedule was used to create a first visual overview of open check-in time windows over the course of a day. Generally, departures are scheduled between 05:15am and 7:14pm. United Airlines operates the most flights, which are spread over the whole day. American Airlines have less flights, resulting in gaps of multiple hours in which their check-in counters are not staffed. Alaska Airlines operate only two flights. It can be concluded that there are likely passenger arrival peaks approximately between 04:00am and 06:00am, as well as between 11:00am and 13:30pm.

In order to determine which number of passengers arrive at which specific time within the check-in interval, a passenger arrival distribution is used. Such a function describes how much prior to departure passengers will arrive at the airport with a certain probability. This model's arrival distributions are based on the distributions that have been presented in the literature review. However, check-in time windows from papers in the literature review deviate significantly from the one at SLO airport. This is expected, as the airlines at SLO airport offer domestic flights only. It was decided to initially model passenger arrival with an "early" distribution, which is presented in Figure 18. The chart shows the shares of passengers that arrive in a certain time interval before departure in minutes. However, two additional distributions that are used to account for varying arrival behavior throughout the day are shown in Figures 19 and 20. These three distributions model early, balanced, and late arrival behavior, respectively.

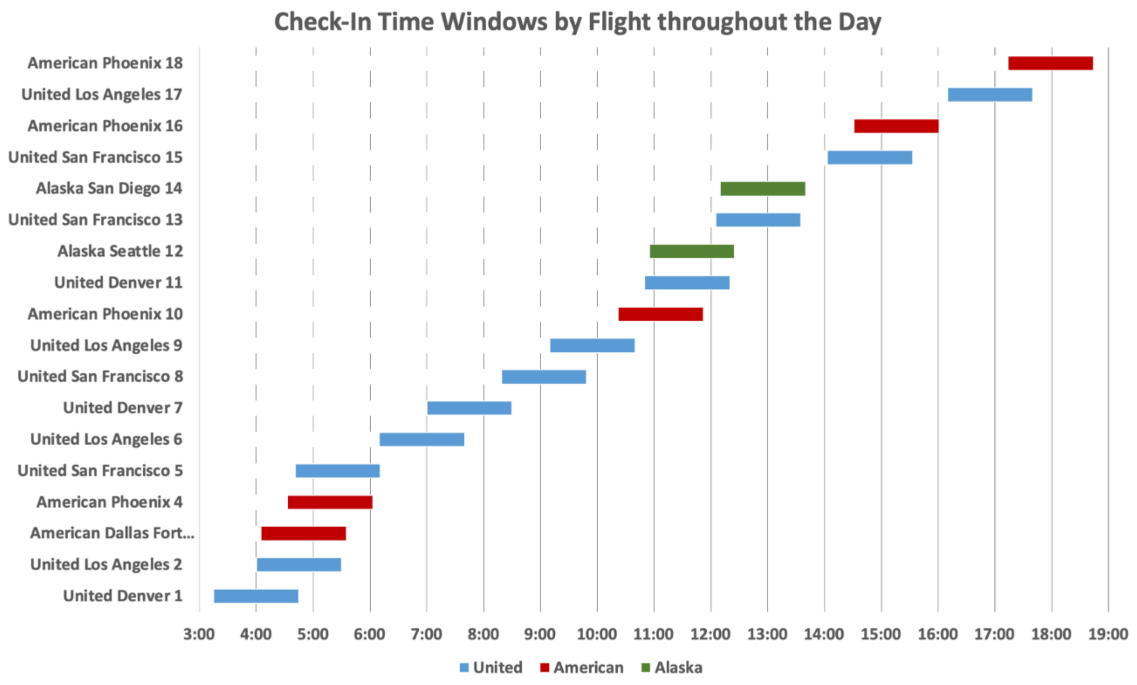


Figure 17: Check-In time windows by flight throughout the day

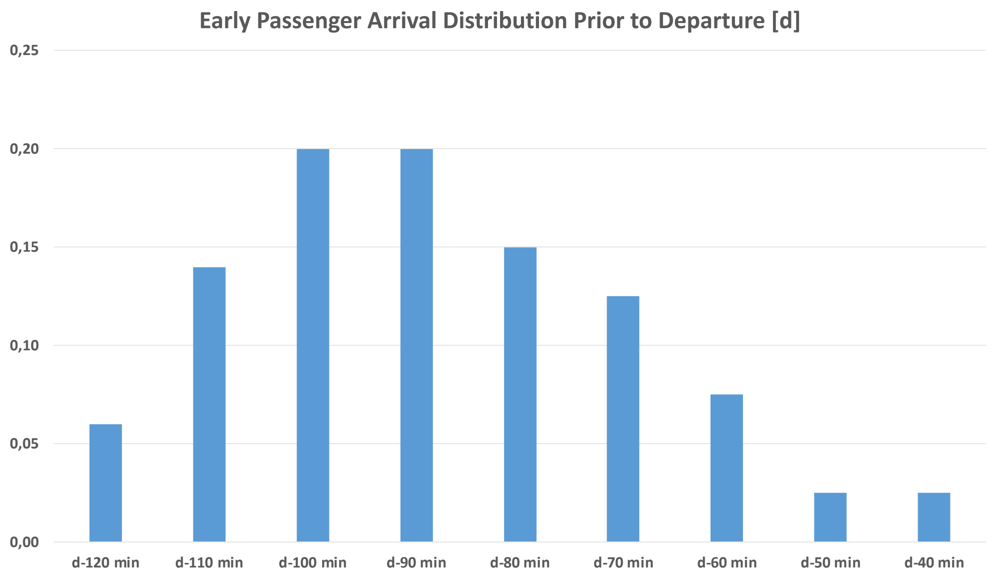


Figure 18: Early passenger arrival distribution prior to departure [d]

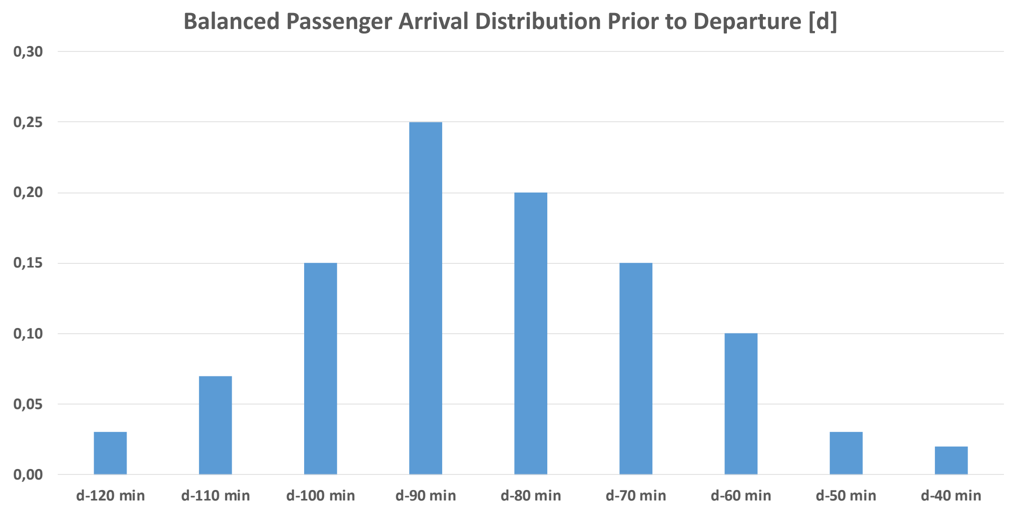


Figure 19: Balanced passenger arrival distribution prior to departure [d]

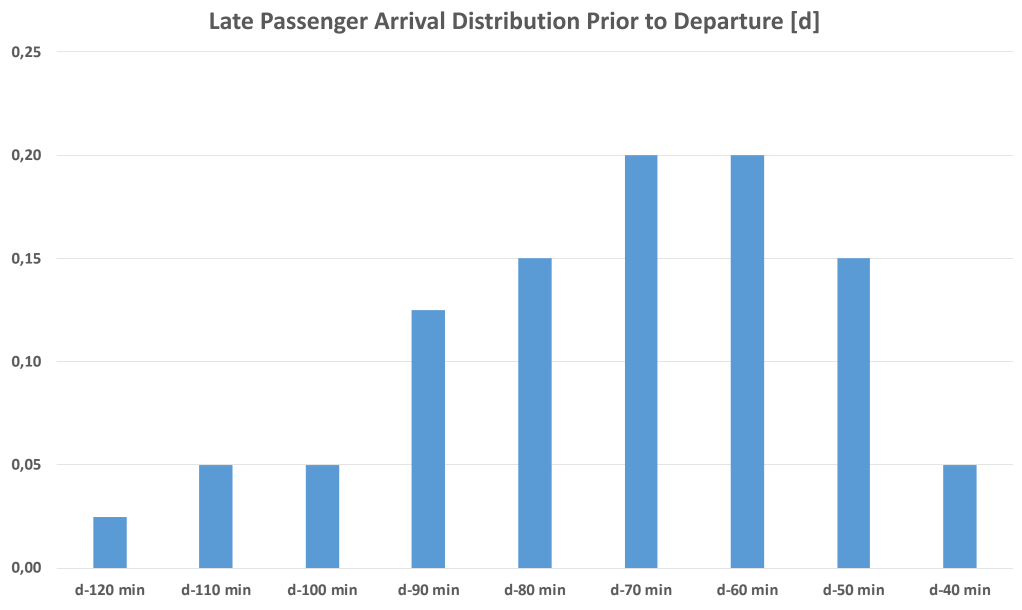


Figure 20: Late passenger arrival distribution prior to departure [d]

3.2.2 Internally Stored

Besides storing model data in logic blocks, Anylogic allows several other objects to store data that can be referenced by the model. These include objects such as parameters, variables, custom distributions, and schedules.

Table 2 provides an overview on parameters that store simulation input data. Parameter names, the data type, the initial value, the unit, and the usage within the model are listed. Those parameters that have "tbd" as their initial value will be used to model additional flights within the experimentation stage.

Variables that store simulation input data are listed in Table 3. The table is structed similarly as Table 2 and "tbd" entries in the column for initial values indicate that these variables will be used to model additional flights within the experimentation stage.

Table 2: Simulation input parameters overview

Parameter Name	**Data Type**	**Inital Value**	**Unit**	**Usage**
Acceleration	double	0.5	m/s^2	Acceleration of the conveyor belt
Airline19	int	tbd		Airline code of additional flight #19
Airline20	int	tbd		Airline code of additional flight #20
Airline21	int	tbd		Airline code of additional flight #21
AlaskaCheckPriority	int	1		American Airlines priority for check-in services
AlertShare	double	0.1		Share of bags that result in the TSA scanner giving an alert
AlsakaBagStorePriority	int	1		American Airlines priority for putting stored bags on the conveyor
AmericanBagStorePriority	int	1		American Airlines priority for putting stored bags on the conveyor
AmericanCheckPriority	int	1		American Airlines priority for check-in services
BagIsCritical	int	1200	s	Bags arriving at the end of the conveyor belt later than 20 minutes prior to departure are critically late
BagIsMissed	int	600	s	Bags arriving at the end of the conveyor belt later than 10 minites prior to departure are missed
BagLength	double	0.715	m	Length of bags
BagsWaitingOnFloor	int	3	bags	Function that dynamically detemines priority of each airline's priority for check-in vs. stored bags
BigBagsShare	double	0.05		Share of oversized bags
CheckInFullMax	double	65	s	Maximum check-in time at the counter for passengers that did not check-in previously
CheckInFullMed	double	55	s	Medium check-in time at the counter for passengers that did not check-in previously
CheckInFullMin	double	45	s	Minimal check-in time at the counter for passengers that did not check-in previously
CheckingTimePerBag	int	45	s	Time needed to check one bag at the counter
CheckInKioskMax	double	45	s	Maximum check-in time at the counter if the passenger used the kiosk
CheckInKioskMed	double	35	s	Medium check-in time at the counter if the passenger used the kiosk
CheckInKioskMin	double	25	s	Minimal check-in time at the counter if the passenger used the kiosk
ClearShare	double	1		Share of bags that are considered clear after being searched manually (1 as this is an exeptional case)
Deceleration	double	0.5	m/s^2	Deceleration of the conveyor belt
Departure19	int	tbd	s	Departure of additional flight #19
Departure20	int	tbd	s	Departure of additional flight #20
Departure21	int	tbd	s	Departure of additional flight #21
GapAlaska	double	0.3	m	Minimal gap between two bags on the conveyor section of Alaska Airlines
GapAmerican	double	1.315	m	Minimal gap between two bags on the conveyor section of American Airlines
GapUnited	double	0.3	m	Minimal gap between two bags on the conveyor section of United Airlines
IntSpeed	double	0.3	m/s	Initial speed of the conveyor belt
KioskShare	double	0.05		Share of passengers that check-in at a kiosk
ManualSearchMax	int	150	s	Maximum time the TSA agent takes to manually search a bag
ManualSearchMed	int	120	s	Medium time the TSA agent takes to manually search a bag
ManualSearchMin	int	90	s	Minimum time the TSA agent takes to manually search a bag
MaxSpeed	double	1	m/s	Maximum speed of the conveyor belt
MovingSpeed	double	1.2	m/s	Walking speed of humans
NrBagsFlexibleTSA	int	10	bags	Function that dynamically adds or subtracts the second TSA agent
NrBagsPermanentTSA	int	5	bags	Function that dynamically adds or subtracts the second TSA agent
NrBagsStorageFlexibleTSA	int	2	bags	Function that dynamically adds or subtracts the second TSA agent
NrBagsStoragePermanentTSA	int	1	bags	Function that dynamically adds or subtracts the second TSA agent
OffsetJump	double	0.05	m	Amount of meters a bag is moved on the conveyor when looking for a free spot
OnlineCheckInShare	double	0.1		Share of passengers that check-in online
PassengersWaitingInLine	int	4	passengers	Function that dynamically detemines priority of each airline's priority for check-in vs. stored bags
PressPedalTime	double	1	s	Time that the TSA agent needs to press the foot pedal
ScanConvSpeed	double	5	m/s	Maximum speed of the conveyor belt in the scanner
ScanGap	double	0.3	m	Minimal gap between two bags on the scanner section
ScanTime	double	30	s	Time the TSA Scanner needs to scan one bag
TimeLimitLate	int	1800	s	Late passengers trying to check in less than 30 minutes before departure are denied
TimeLimitSkipLine	int	2100	s	Late passengers trying to check in within the last 35 minutes before departure can skip the queue
UnitedBagStorePriority	int	1		American Airlines priority for putting stored bags on the conveyor
UnitedCheckPriority	int	1		American Airlines priority for check-in services

Table 3: Simulation input variables overview

Variable Name	Data Type	Inital Value	Unit	Usage
Airline22	int	tbd		Airline code of additional flight #22
Airline23	int	tbd		Airline code of additional flight #23
Airline24	int	tbd		Airline code of additional flight #24
Airline25	int	tbd		Airline code of additional flight #25
CapacityConveyorAlaska	int	3	bags	Determening if there is a free conveyor spot for Alaska Airlines (theoretically)
CapacityConveyorAmerican	int	3	bags	Determening if there is a free conveyor spot for American Airlines(theoretically)
CapacityConveyorUnited	int	7	bags	Determening if there is a free conveyor spot for United Airlines (theoretically)
ConvLengthAlaska	double	3142	m	Calculating the conveyor capacity of Alaska Airlines
ConvLengthAmerican	double	7.62	m	Calculating the conveyor capacity of American Airlines
ConvLengthUnited	double	7.62	m	Calculating the conveyor capacity of United Airlines
Departure22	int	tbd	s	Departure of additional flight #22
Departure23	int	tbd	s	Departure of additional flight #23
Departure24	int	tbd	s	Departure of additional flight #24
Departure25	int	tbd	s	Departure of additional flight #25

Custom distributions are used to model the amount of drop-off luggage units that each passenger has. Table 4 provides a summary of three different luggage distributions. In comparison to the distributions that have been introduced in the literature review, the average amount of bags per passenger at SLO airport is significantly smaller, according to data that the airport provided. Therefore, new distributions have been designed for the simulation model. Luggage distributions are assigned to passengers by flight.

Table 4: Distributions of luggage units per passenger

	Distribution		
Luggage Units per Passenger	**Regular**	**More**	**Less**
0	55,5%	48,0%	79,0%
1	39,5%	42,0%	21,0%
2	5,0%	10,0%	0,0%
# Average Bags per Passenger	**0,495**	**0,62**	**0,21**

Schedules used in the model contain staffing information for the check-in counters of each airline as seen in Table 5. The amount of staff members in each interval has been determined by considering the amount of flights that have their check-in time windows simultaneously. Roughly, one flight will have two airline employees serving passengers at the check-in counters. In case of a slight overlapping of check-in time windows for the same airline, three employees will staff the counters. Heavy overlapping results in four airline employees.

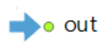
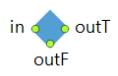
Table 5: Check-in counter shift schedule by airline

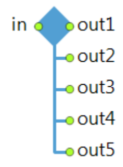
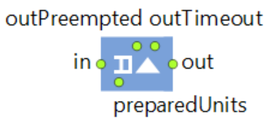
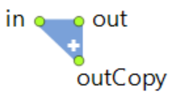
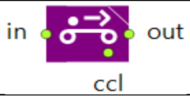
American Shift Capacity	**From**	**To**
4	4:05	6:03
2	10:22	11:52
2	14:31	16:01
2	17:14	18:44
United Shift Capacity	**From**	**To**
2	3:15	4:00
4	4:00	6:11
3	6:11	13:35
2	14:03	15:33
2	16:10	17:40
Alaska Shift Capacity	**From**	**To**
3	10:55	13:40

3.3 Logic Blocks Used in the Model

Anylogic offers multiple modeling libraries, such as the Process Modeling Library and the Material Handling Library, that each contain multiple logic blocks. Table 6 gives an overview of all logic blocks used in the model, their symbols, names, and basic functions. Green dots on the symbols indicate where the logic blocks can be connected with other blocks. The "in" ports can have multiple incoming connections while the "out" ports allow one connection. Some blocks, for instance, the Select Output logic blocks, have multiple outgoing ports. Here, "outT" stands for the logical true output port and "outF" stands for the logical false output port. These references shown on the symbols will be used in the model explanation. Blue symbols indicate a logic block is from the Process Modeling Library, purple ones are from the Material Handling Library.

Table 6: Overview of simulation modeling logic blocks

Symbol	**Name**	**Function**
	Enter	Introduces agents into flowchart
	Select Output	Leads agents to the true or false output depending on condition or probability

	Select Output 5	Leads agents to one of the 5 output ports depending on conditions or probabilities
	Delay	Delays the agent
outPreempted outTimeout in out	Queue	Keeps incoming agents in order or sorts them
	Seize	The incoming agent seizes a resource unit
in out wrapUp	Release	The agent releases the seized resource unit
	Split	Creates a copy of incoming agent, possibly other agent type
	Move to	Moves agent in animation
	Conveyor Enter	Puts agent on conveyor
	Convey	Conveys agent to defined location in conveyor network
	Conveyor Exit	Removes agent from conveyor
	Hold	Blocks agent flow
	Sink	Removes agents from flowchart

3.4 Detailed Simulation Model Design

For consistency and clarity, the documentation of the simulation model design follows the logical flow of processes, similar to the process descriptions presented before. However, sub-headings might incorporate additional sub-headings. One example for this case is the computational procedure used to model individual passenger arrival time stamps within the passenger check-in process. Figure 21 shows the complete logic flowchart of the simulation model that is built out of logic blocks that were introduced above. Its sections will be described in more detail in the following sub-chapters. Anylogic allows to incorporate java code in models. Code can be stored in functions that can be executed upon calling them. Further, code can be saved in the logic blocks and executed when an agent passes through the respective block. Next to the logic flowchart, multiple graphs that visualize key performance indicators are included.

The model works with several types of agents. Agents can be used as resources, such as the counter staff members of each airline or the TSA staff. Those are linked to resource pools, such as the staff of American Airlines. Further, two types of agents travel through the logic flowchart: passengers and bags. Many decisions within the model depend on characteristics of those two agent types. Characteristics are modeled as parameters. Every agent type has a graphical representation that is used in the animation.

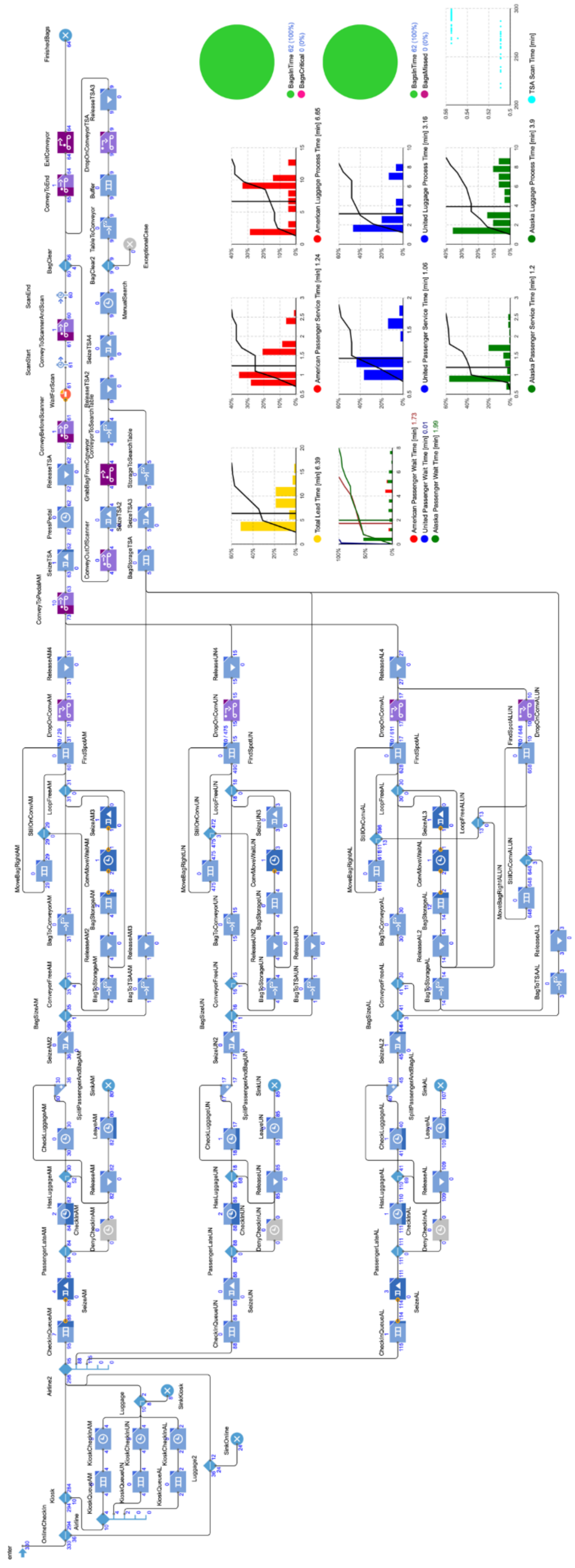


Figure 21: Overview of the simulation model's logic flowchart

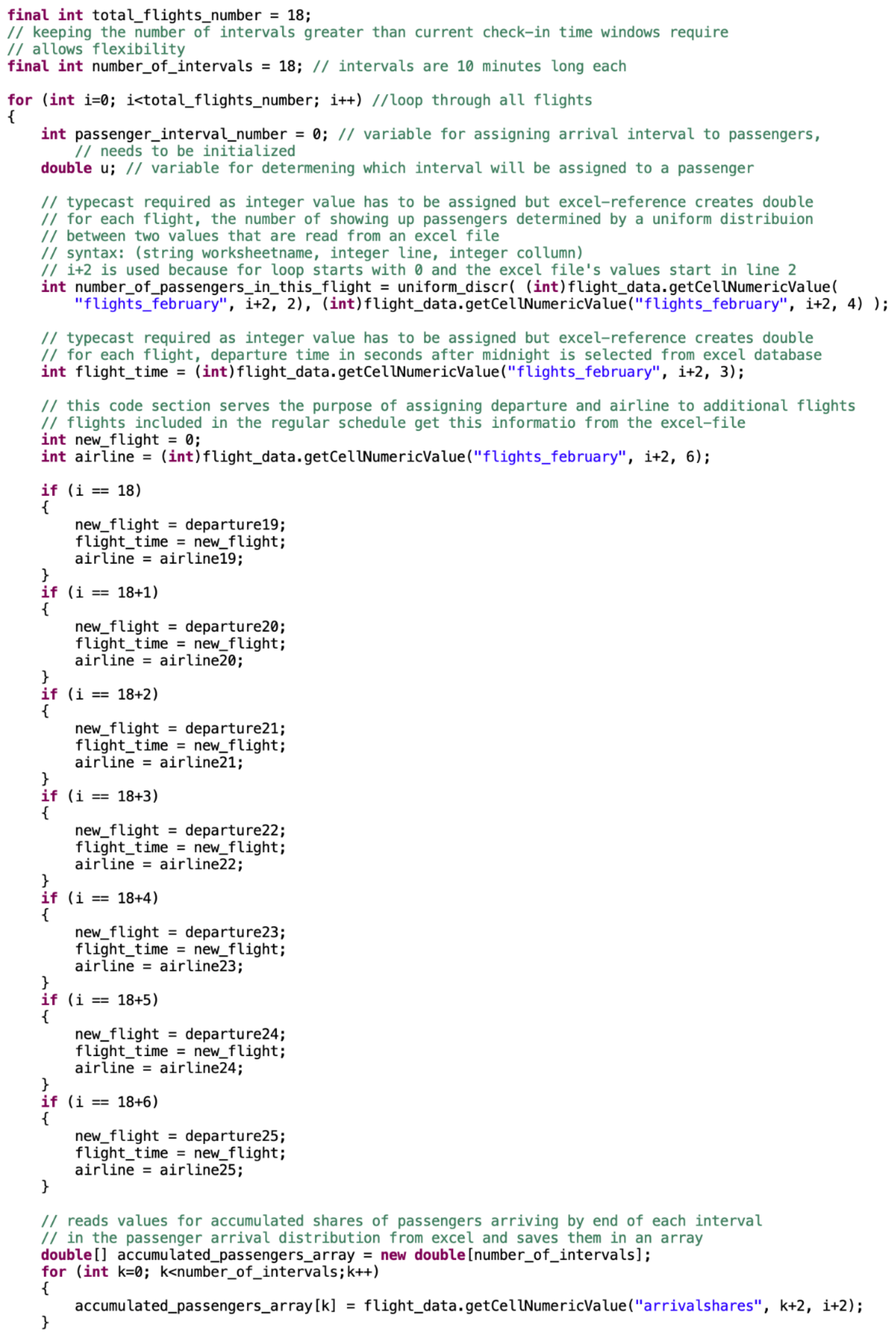
3.4.1 Passenger Creation

In some way, passengers have to be created in the simulation. The presented model uses a function named “initschedule” that was written to take care of this. It is automatically executed on model startup and incorporates several nested loops that go through every scheduled flight. Figure 22 to 24 show the java code of this function. Figure 22 includes the first snippet of the code that assigns the number of passengers, the flight’s departure, the operating airline, and the passenger arrival distribution for each flight. Additionally, the code recognizes if additional flights have been added to the schedule and assigns their departure and operating airline differently from how this is done for the 18 regular flights from the February 2020 schedule. While regular flights are fully assigned by the Excel database, the additional ones use parameters as this allows to try different scenarios in an automated way.

The Excel database that the function is referring to has been derived from the flight schedule that was presented in the simulation data input chapter. Table 7 includes the flight numbers, the possible range of passengers resulting from allowing fluctuations of 5%, a code for the luggage distribution, and an airline code for each flight. Passenger load factors are not visible here, however, they are used to compute the range of passengers possible.

Table 7: Excel database used for passenger creation

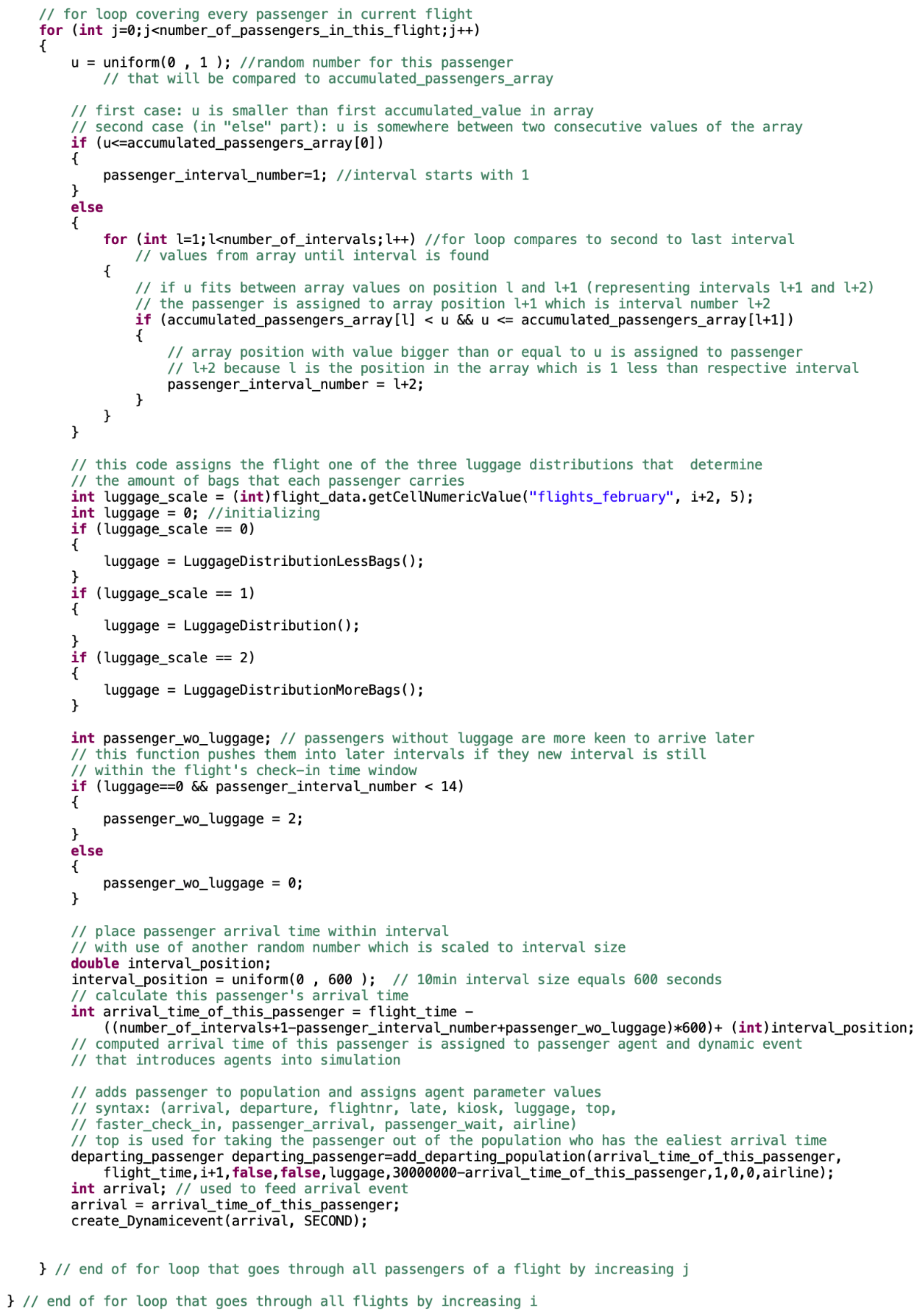
Flight ID	Passengers min	Departure [s]	Passengers max	Luggage	Airline Code
1	36	18900	40	2	2
2	55	21600	61	2	2
3	61	21900	67	2	1
4	56	23580	62	2	1
5	58	24060	64	0	2
6	35	29400	39	1	2
7	46	32400	50	1	2
8	38	37140	42	1	2
9	45	40200	49	2	2
10	63	44520	70	2	1
11	47	46200	52	2	2
12	61	46500	68	2	3
13	45	50700	49	1	2
14	62	51000	69	1	3
15	47	57780	52	0	2
16	53	59460	59	1	1
17	45	65400	49	0	2
18	50	69240	55	0	1



```java
final int total_flights_number = 18;
// keeping the number of intervals greater than current check-in time windows require
// allows flexibility
final int number_of_intervals = 18; // intervals are 10 minutes long each

for (int i=0; i<total_flights_number; i++) //loop through all flights
{
    int passenger_interval_number = 0; // variable for assigning arrival interval to passengers,
        // needs to be initialized
    double u; // variable for determening which interval will be assigned to a passenger

    // typecast required as integer value has to be assigned but excel-reference creates double
    // for each flight, the number of showing up passengers determined by a uniform distribuion
    // between two values that are read from an excel file
    // syntax: (string worksheetname, integer line, integer collumn)
    // i+2 is used because for loop starts with 0 and the excel file's values start in line 2
    int number_of_passengers_in_this_flight = uniform_discr( (int)flight_data.getCellNumericValue(
        "flights_february", i+2, 2), (int)flight_data.getCellNumericValue("flights_february", i+2, 4) );

    // typecast required as integer value has to be assigned but excel-reference creates double
    // for each flight, departure time in seconds after midnight is selected from excel database
    int flight_time = (int)flight_data.getCellNumericValue("flights_february", i+2, 3);

    // this code section serves the purpose of assigning departure and airline to additional flights
    // flights included in the regular schedule get this informatio from the excel-file
    int new_flight = 0;
    int airline = (int)flight_data.getCellNumericValue("flights_february", i+2, 6);

    if (i == 18)
    {
        new_flight = departure19;
        flight_time = new_flight;
        airline = airline19;
    }
    if (i == 18+1)
    {
        new_flight = departure20;
        flight_time = new_flight;
        airline = airline20;
    }
    if (i == 18+2)
    {
        new_flight = departure21;
        flight_time = new_flight;
        airline = airline21;
    }
    if (i == 18+3)
    {
        new_flight = departure22;
        flight_time = new_flight;
        airline = airline22;
    }
    if (i == 18+4)
    {
        new_flight = departure23;
        flight_time = new_flight;
        airline = airline23;
    }
    if (i == 18+5)
    {
        new_flight = departure24;
        flight_time = new_flight;
        airline = airline24;
    }
    if (i == 18+6)
    {
        new_flight = departure25;
        flight_time = new_flight;
        airline = airline25;
    }

    // reads values for accumulated shares of passengers arriving by end of each interval
    // in the passenger arrival distribution from excel and saves them in an array
    double[] accumulated_passengers_array = new double[number_of_intervals];
    for (int k=0; k<number_of_intervals;k++)
    {
        accumulated_passengers_array[k] = flight_data.getCellNumericValue("arrivalshares", k+2, i+2);
    }
```

Figure 22: Java code of the "initschedule" function part 1

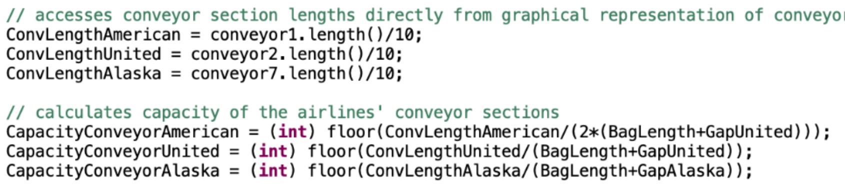
Figure 23 contains the code section that loops through every passenger of each flight. For every passenger, a random variable, “u,” is assigned. The variable, “u,” is uniformly distributed between 0 and 1 and is compared to the array values of the accumulated passengers array from the passenger arrival distribution that was assigned to the respective flight. This way, the passenger receives the interval number in which he arrives. This logic is based on the work of (Cavada, Cortés, & Rey, 2017). Towards the end of the code, this is one of the factors used to compute the exact arrival time. Depending on the luggage distribution of their flight, the passenger is assigned with a certain number of drop-off luggage. Given that a passenger is assigned with zero baggage items and is not arriving within the last two intervals in which the check-in time window is open, he will be pushed back 2 intervals, resulting in him arriving later. This is because travelers matching the criteria are likely going to plan with less slack time. In order to finally compute the passenger’s arrival time, another random variable, “v,” is assigned. It is uniformly distributed between 0 and 600, which includes every second within one 10-minute interval. At this point, the passengers are added to a population while feeding the passenger agent with all of these parameters. Following that, a dynamic event that will introduce the passenger into the simulation logic flowchart at the correct time is provided with this information.



```
        // for loop covering every passenger in current flight
        for (int j=0;j<number_of_passengers_in_this_flight;j++)
        {
            u = uniform(0 , 1 ); //random number for this passenger
                // that will be compared to accumulated_passengers_array

            // first case: u is smaller than first accumulated_value in array
            // second case (in "else" part): u is somewhere between two consecutive values of the array
            if (u<=accumulated_passengers_array[0])
            {
                passenger_interval_number=1; //interval starts with 1
            }
            else
            {
                for (int l=1;l<number_of_intervals;l++) //for loop compares to second to last interval
                    // values from array until interval is found
                {
                    // if u fits between array values on position l and l+1 (representing intervals l+1 and l+2)
                    // the passenger is assigned to array position l+1 which is interval number l+2
                    if (accumulated_passengers_array[l] < u && u <= accumulated_passengers_array[l+1])
                    {
                        // array position with value bigger than or equal to u is assigned to passenger
                        // l+2 because l is the position in the array which is 1 less than respective interval
                        passenger_interval_number = l+2;
                    }
                }
            }

            // this code assigns the flight one of the three luggage distributions that  determine
            // the amount of bags that each passenger carries
            int luggage_scale = (int)flight_data.getCellNumericValue("flights_february", i+2, 5);
            int luggage = 0; //initializing
            if (luggage_scale == 0)
            {
                luggage = LuggageDistributionLessBags();
            }
            if (luggage_scale == 1)
            {
                luggage = LuggageDistribution();
            }
            if (luggage_scale == 2)
            {
                luggage = LuggageDistributionMoreBags();
            }

            int passenger_wo_luggage; // passengers without luggage are more keen to arrive later
            // this function pushes them into later intervals if they new interval is still
            // within the flight's check-in time window
            if (luggage==0 && passenger_interval_number < 14)
            {
                passenger_wo_luggage = 2;
            }
            else
            {
                passenger_wo_luggage = 0;
            }

            // place passenger arrival time within interval
            // with use of another random number which is scaled to interval size
            double interval_position;
            interval_position = uniform(0 , 600 );  // 10min interval size equals 600 seconds
            // calculate this passenger's arrival time
            int arrival_time_of_this_passenger = flight_time -
                ((number_of_intervals+1-passenger_interval_number+passenger_wo_luggage)*600)+ (int)interval_position;
            // computed arrival time of this passenger is assigned to passenger agent and dynamic event
            // that introduces agents into simulation

            // adds passenger to population and assigns agent parameter values
            // syntax: (arrival, departure, flightnr, late, kiosk, luggage, top,
            // faster_check_in, passenger_arrival, passenger_wait, airline)
            // top is used for taking the passenger out of the population who has the ealiest arrival time
            departing_passenger departing_passenger=add_departing_population(arrival_time_of_this_passenger,
                flight_time,i+1,false,false,luggage,30000000-arrival_time_of_this_passenger,1,0,0,airline);
            int arrival; // used to feed arrival event
            arrival = arrival_time_of_this_passenger;
            create_Dynamicevent(arrival, SECOND);


        } // end of for loop that goes through all passengers of a flight by increasing j

    } // end of for loop that goes through all flights by increasing i
```

Figure 23: Java code of the "initschedule" function part 2

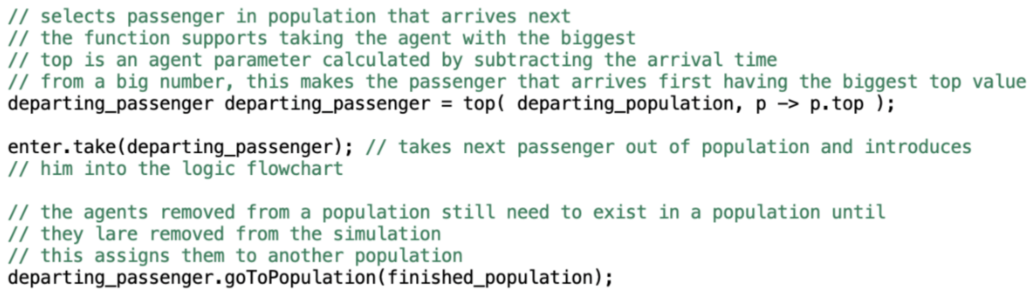
The last section of the code is displayed in Figure 24. Input variables of the conveyor section lengths and the respective bag capacities of these sections are computed.



```
// accesses conveyor section lengths directly from graphical representation of conveyor
ConvLengthAmerican = conveyor1.length()/10;
ConvLengthUnited = conveyor2.length()/10;
ConvLengthAlaska = conveyor7.length()/10;

// calculates capacity of the airlines' conveyor sections
CapacityConveyorAmerican = (int) floor(ConvLengthAmerican/(2*(BagLength+GapUnited)));
CapacityConveyorUnited = (int) floor(ConvLengthUnited/(BagLength+GapUnited));
CapacityConveyorAlaska = (int) floor(ConvLengthAlaska/(BagLength+GapAlaska));
```

Figure 24: Java code of the "initschedule" function part 3

The final step of passenger creation is their removal from the initial population and their introduction into the simulation. Figure 25 shows the java code of the dynamic event that covers this functionality.



```
// selects passenger in population that arrives next
// the function supports taking the agent with the biggest
// top is an agent parameter calculated by subtracting the arrival time
// from a big number, this makes the passenger that arrives first having the biggest top value
departing_passenger departing_passenger = top( departing_population, p -> p.top );

enter.take(departing_passenger); // takes next passenger out of population and introduces
// him into the logic flowchart

// the agents removed from a population still need to exist in a population until
// they lare removed from the simulation
// this assigns them to another population
departing_passenger.goToPopulation(finished_population);
```

Figure 25: Java code of the dynamic event

3.4.2 Before Check-In Counter

Compared to the process flowcharts in chapter 2, the perspective here is not changed from passenger, to airline employee, to TSA worker, et cetera. Rather, the perspective is on what passengers actively do, which services they passively receive from resources, and what is done with dropped-off luggage.

Figure 26 gives an overview of the first section of the logic flowchart which incorporates activities and processes that can take place before a traveler might go to a check-in counter. Now that passengers are introduced into the model via the enter-block, passengers are split depending on if they checked-in online or not. Travelers who have done this will take the “false” port of the “OnlineCheckIn” block. If they do not carry drop-off luggage, they are led to “SinkOnline” and are removed from the simulation. Online check-in passengers that do have luggage are led to the Airline2 block which leads them to the check-in counter of their airline.

Passengers who did not check in online can use their airline's kiosk for self-check in. These people take the "false" port of "Kiosk" and select the airline's queue before checking themselves in, which is modeled by delays. Names of delay and queue blocks end with AM, UN, or AL which stands for the airlines American, United, and Alaska, respectively. This name coding is used in the whole model. Kiosk passengers will also leave the simulation if they do not have baggage items with them. The rest proceeds to the check-in counters. For all passengers leaving by entering a sink, a timestamp of the current simulation time plus five minutes is saved. These timestamps are used for the creation of simulated person throughput time series by the hour.

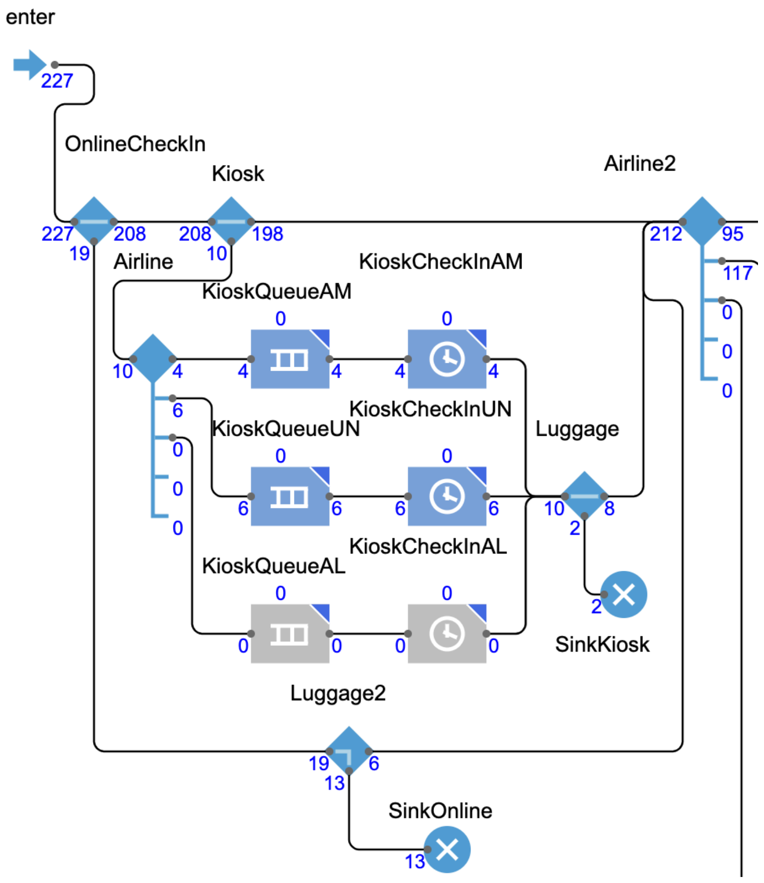


Figure 26: Before check-in counter model logic

3.4.3 Check-in Counter

Check-in counter procedures are underlying a similar structure of logic blocks. American Airline's logic is picked as an example that is shown in Figure 27. First, passengers go into the queue which is priority-based depending on the arrival time of passengers. In addition, implemented java code detects passengers that arrive rather late and assigns them a higher priority. This means that late passengers skip the line. Following that, passengers arrive in the "SeizeAM" block where they stay until they seized a resource, one of the American Airlines staff members. Java code recognizes late passengers and labels them accordingly. These passengers will be denied from check-in, resulting in releasing the seized resource, leaving the counter, and getting removed from the simulation. Travelers trying to check-in on time seize a staff member, are checked in, drop-off their luggage if they have any, and enter the "SplitPassengerAndBagAM" block. The passenger leaves the block via the "out" port, and luggage agents exit through the "outCopy" port. Passengers that are checked in and do not have luggage will behave similarly as passengers that dropped-off luggage: they release the airline employee and leave the counter.

If the counter queue is very long, the airline employee will "hurry" and the check-in process is faster. For all passengers leaving, a timestamp of the current simulation time plus five minutes is saved. These timestamps are used for the creation of simulated security check person throughput time series by the hour.

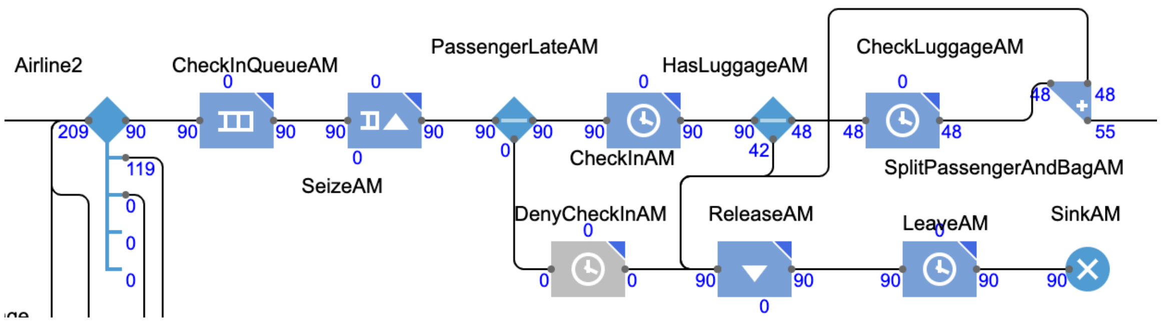


Figure 27: Check-in counter model logic

3.4.4 Airline Luggage Handling

From now on, luggage agents representing one bag each are passing through the flowchart. The structure is similar for American and United Airlines, Alaska however has a slightly different process that requires a slightly varying logic that will be explained towards the end of this chapter. Figure 28 presents the logic of the airline luggage handling logic with the example of American Airlines. Any bag being dropped-off seizes the airline agent. "BagSizeAM" leads oversized bags downwards. The airline agent will carry the oversized luggage item into the TSA room and is released afterwards. When released, he returns to the counter. Regular sized bags enter the "ConveyorFreeAM" block, which compares the bag capacity of the airline's conveyor section with the sum of bags that are about to be put onto the conveyor and the ones already on the conveyor.

Comparisons such as these are required as the simulated airline agent cannot just look at the conveyor belt and notice or register if there is a free spot like a real human.

If the conveyor's capacity is reached, the airline agent carries the bag next to the conveyor and is released. The luggage agent proceeds into the "BagStorageAM" block, which models bags being temporarily stored on the floor. This queue is priority-based depending on an agent parameter that is increased if an airline agent unsuccessfully tried to put the bag on the conveyor. Before a bag agent seizes an airline worker in the "SeizeAM3" block, they are delayed to compensate for conveyor movement based on the TSA scanner takt time.

In case of the conveyor theoretically having space, the airline worker carries the bag to the conveyor. This is where luggage items that were temporarily stored and the ones that have not both enter the "Loop-FreeAm" block. The purpose of this block is to prevent deadlock situations by only allowing one luggage agent to be within the loop that is explained in the following. Baggage agents that cannot enter the loop are sent back to the temporary storage between the counter and the conveyor.

The loop is formed between the "outTimeout" port of "FindSpotAM," "MoveBagRightAM," and "StilOnConvAM." Bags are introduced into the conveyor system by the purple block. As simulated airline agents do not have a natural intelligence, they will not place a bag on any empty spot on the conveyor. However, Anylogic does allow to assign an offset value for how far from the conveyor start a bag shall be put on the belt. If this specific spot is blocked by another bag, the luggage agent that wants to get into the "DropOnConvAM" block will remain in the queue before until the spot has cleared. This means that the flow of luggage items and the airline agent are blocked. The loops that have been described earlier solves this problem. Given that a luggage agent enters "FindSpotAM" for the first time, the agent's parameter, called "conveyorspot," which is used as the offset, will be set to the start of the conveyor. The "DropOnConvAM" block will not allow the agent to enter if this offset position is blocked. "FindSpotAm" has a timeout implemented that leads the luggage agent out of the "outTimeout" port after waiting in it for 2 milliseconds. The luggage agent's "conveyorspot" parameter value will be increased by 5 centimeters. The block "StillOnConvAM" checks if the new spot is still on the airline's conveyor section. If yes, the luggage agent enters "FindSpotAM" again. If the new "conveyorspot" is free, the luggage item will enter the "DropOnConv" block, which means that the bag is put onto the conveyor. Following that, the bag agent releases the airline worker.

If necessary, the loop will be passed multiple times until the added 5 centimeters exceed the conveyor section's length. These luggage agents are led to the temporary bag storage where they skip the queue as the loop assigns a higher priority.

Alaska Airline's conveyor section is significantly smaller than those of the other airlines. This means that they do not have the same chance of getting their bags onto the conveyor, especially when a continuous flow of bags from the other airlines blocks their belt. Alternatively, a backlog from the scanner might block their section, too. Therefore, Alaska workers will try to put bags on the conveyor section of United whenever their own section is blocked. Another loop, similar to the one that has been explained is implemented in Alaska Airline's logic.

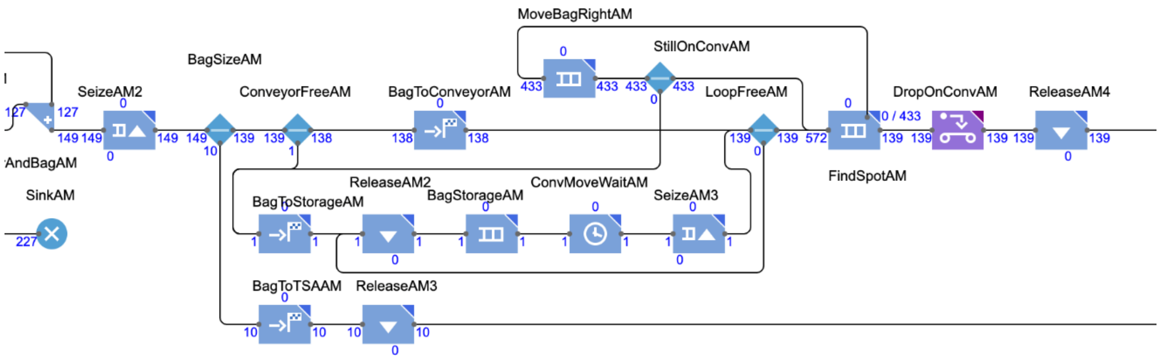


Figure 28: Airline luggage handling model logic

3.4.5 TSA Luggage Handling

The TSA luggage handling section of the model is where the sections separated by airline join. First, the most common process chain of regular sized bags is described. As seen in Figure 29, luggage agents arrive in the "ConveyToPedalAM" block which transports them into the TSA room next to the foot pedal. Upon arrival there, the bag agent seizes a TSA worker that walks to the pedal and presses it to initiate further conveyor transport into the TSA scanner. The TSA agent is then released and walks back to his home position next to the conveyor. "ConveyBeforeScanner" moves the bag agent on the last conveyor belt section before the scanner where bags will be held if another bag is scanned in this moment. When the bag scan is completed, "WaitForScan" releases the hold and allows the next bag to be moved into the scanner. During the scan, the current simulation time is saved in order to create a simulated time series of luggage throughput. If the scanner did not detect suspicious objects, the bag agent is transported to the end of the conveyor, exits the conveyor, and leaves the simulation.

Bags that create an alert are transported outside of the scanner. Upon arrival the bag seizes a TSA worker in "SeizeTSA2" who walks to its position on the conveyor and picks it up from the conveyor. Following that, the TSA agents bring the bag to the manual search table. This is where oversized bags join the logic stream of bags that created an alert. Temporary storage of oversized bags is modeled with the queue "BagStorageTSA." Those bags seize a TSA agent that walks to their location and brings them to the manual search table. Bags that are placed on the manual search table release the TSA agent at first. This serves the purpose of TSA agents being able to, for example, keep the scanner throughput going by pressing the foot pedal before searching a bag. Task priorities in the seize blocks were implemented for this. Most crucial for throughput is the removal of alert bags from the conveyor as these bags prevent the scanner from releasing scanner bags. After a side-task like this is done, the bag agent on the table will seize a TSA agent that is performing the manual search. As it would be an exceptional case that results in an airport shutdown, the model finds all manually searched luggage items clear. Consecutively, the bag is carried to and put onto the conveyor and the TSA agent is released. As the model does not include the baggage sorting, there cannot be backlogs from the station that would require to model temporary bag storage next to the conveyor as described in chapter 2.

For any bag that leaves the simulation, the current simulation time is checked and compared to the required slack time defined before departure for bags that are critically late and bags that are missed. In other words, if a bag arrives at the end of the conveyor less than 20 minutes prior to its departure it is labeled as critical. If there are less than 10 minutes of slack time until departure, the bag will either miss its flight or delay takeoff. These bags are labeled as missed.

Whenever oversized bags are brought into the TSA room and whenever regular-sized bags arrive in front of the scanner, java code can dynamically add and remove a second, flexible TSA agent. The flexible agent is added when many bags are waiting in front of the scanner and when some oversized bags wait to be searched manually. Given that the congestion has been processed, the second TSA agent is removed. Properties of the "SeizeTSA" blocks are designed so that both TSA agents work together effectively. It would not make sense to have both searching bags manually at the same time. This is why manual searching and carrying of oversized bags are only performed by the permanent TSA agent.

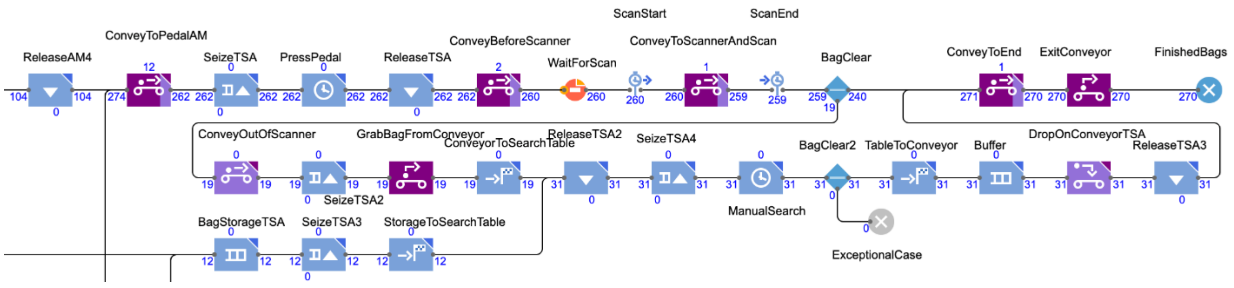


Figure 29: TSA luggage handling model logic

3.5 Animation

Besides the logical flowchart, a graphical representation of the model has been built. This requires linking logic blocks that contain movements and locations of agents and resource agents to be linked to graphical nodes within the graphical model. Figure 30 shows the two-dimensional layout of the conveyor system. The big rectangle filled with a pattern represents the publicly accessible section of the terminal. The yellow rectangle is where passengers arrive. Self-check-in kiosks are attached with their queues. Rectangles and lines that are dashed represent nodes. They define where agents and resources are located and move. American Airlines counters, employees, passengers, and their luggage are colored red. United Airlines is color-coded in blue and Alaska Airlines in green. The order of airline counters in Figure 30 from top to bottom is four counters of American Airlines, four counters for United Airlines, and three counters of Alaska Airlines.

The L-shaped conveyor leads into the TSA room which is marked by filling with a different pattern. The rectangle with solid filling on the conveyor represents the TSA scanner; the one above the scanner represents the table where bags are searched manually. Airline workers that carry oversized bags into the TSA room will walk on the dashed line that leads from the counters to the rectangle next to the scanner where these bags are dropped.

Figure 31 visualizes the same graphical representation in a three-dimensional way. Within that image, the TSA agents are easier to see. Besides, the rectangles between the counters and conveyor that mark where bags can be temporarily stored stand out.

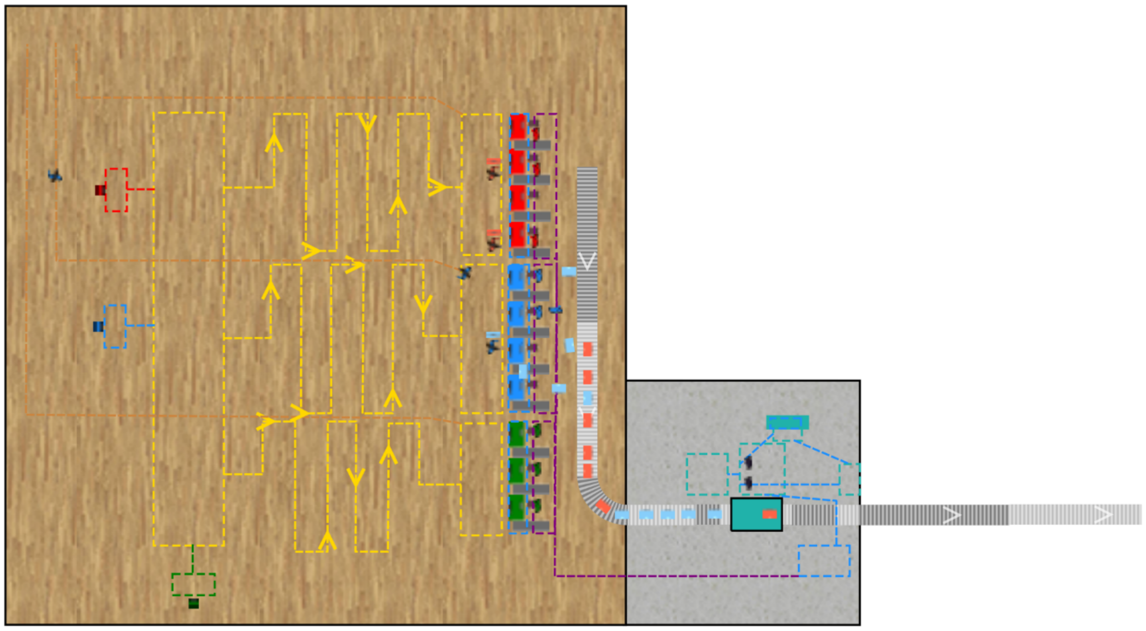


Figure 30: Two-dimensional simulation model animation

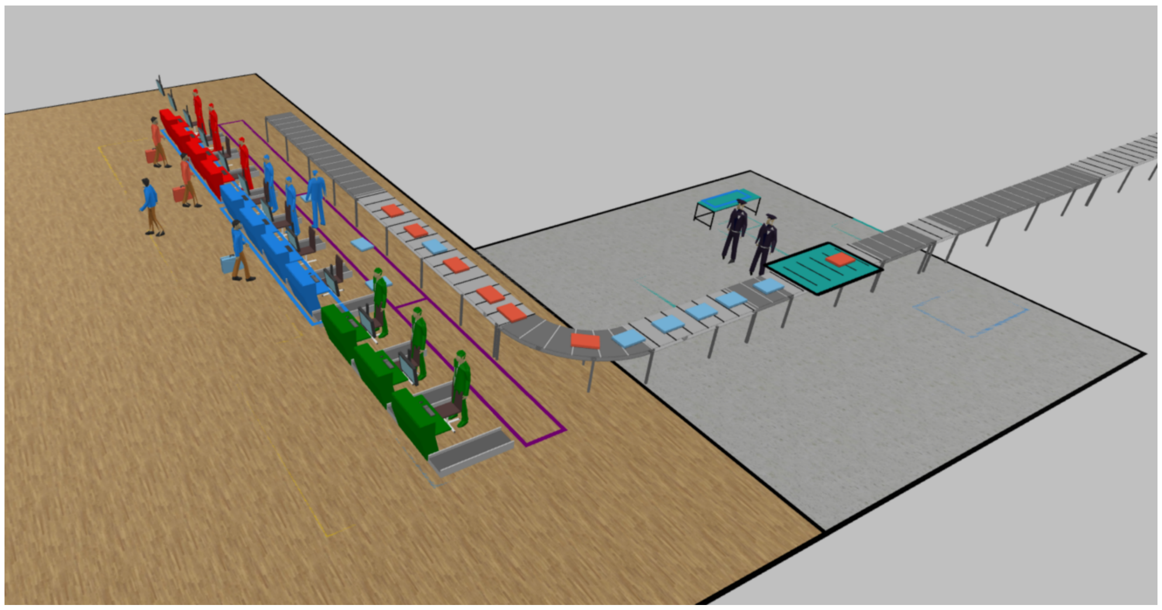
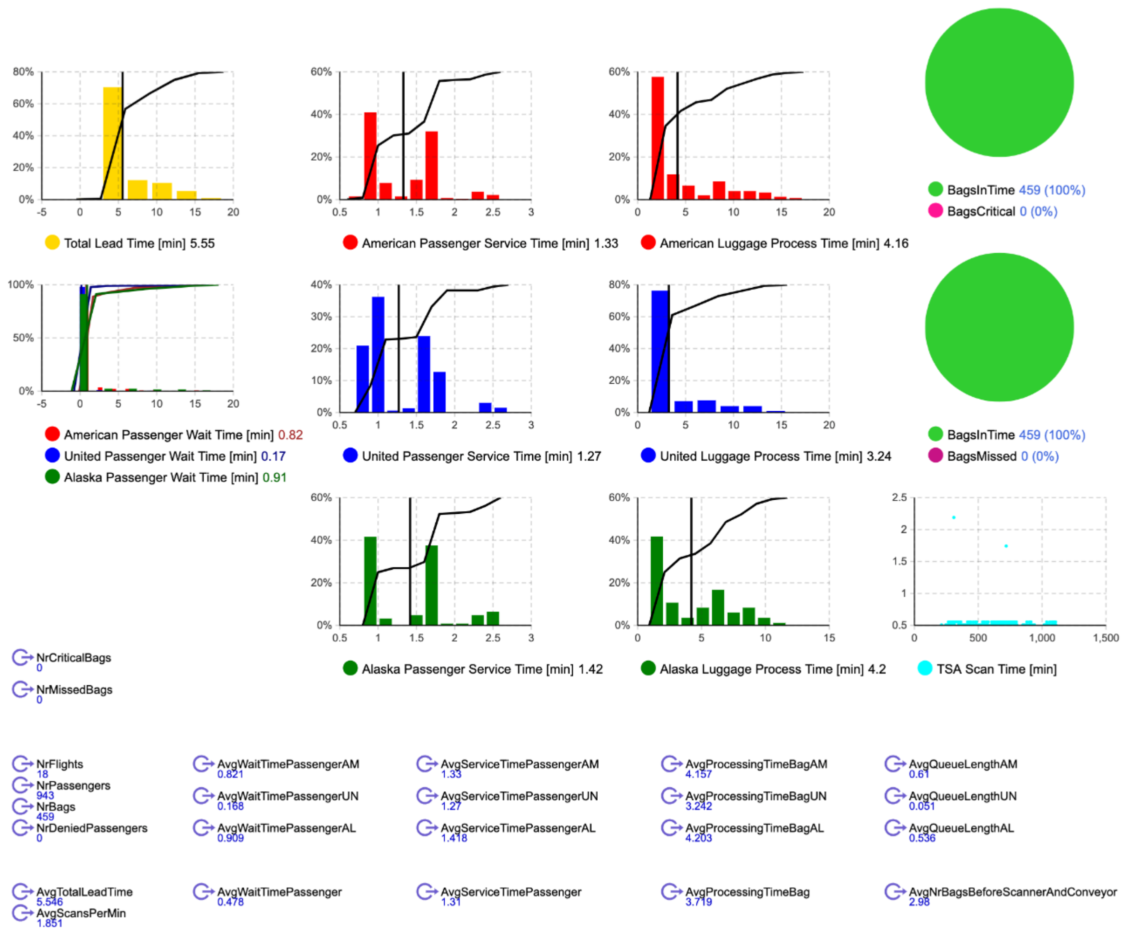


Figure 31: Three-dimensional simulation model animation

3.6 Key Performance Indicators Output

In order to evaluate the simulated system's behavior, java code within the model outputs a range of KPIs that are computed at the end of a simulation run. A part of them is also visualized in charts. Figure 32 displays the outputs, such as the respective charts that fills with data during the simulation run. Charts in the first three columns display histograms and their respective cumulative density function of the described measures. Numbers below the chart display the average value. Charts showing passenger service time and luggage processing time are featured for each airline separately. Pie charts present the share of bags that are critical and missed.



NrCriticalBags 0
NrMissedBags 0

NrFlights 18
NrPassengers 943
NrBags 459
NrDeniedPassengers 0
AvgWaitTimePassengerAM 0.821
AvgWaitTimePassengerUN 0.168
AvgWaitTimePassengerAL 0.909
AvgServiceTimePassengerAM 1.33
AvgServiceTimePassengerUN 1.27
AvgServiceTimePassengerAL 1.418
AvgProcessingTimeBagAM 4.157
AvgProcessingTimeBagUN 3.242
AvgProcessingTimeBagAL 4.203
AvgQueueLengthAM 0.61
AvgQueueLengthUN 0.051
AvgQueueLengthAL 0.536

AvgTotalLeadTime 5.546
AvgScansPerMin 1.851
AvgWaitTimePassenger 0.478
AvgServiceTimePassenger 1.31
AvgProcessingTimeBag 3.719
AvgNrBagsBeforeScannerAndConveyor 2.98

Figure 32: Simulation model KPI output values and visualization

A brief description of all outputs is provided in Table 8. All the time-related outputs are saved in minutes. Their values are computed by using references to datasets within the model that collect the respective data. For example, the current passenger queue length by the airline is saved to airline-specific datasets whenever a new passenger is receiving check-in services by airline staff. The outputs are linked to these datasets and compute the mean. This way, it is ensured that time intervals of idle counters with empty queues do not affect the averages.

Table 8: Overview of simulation outputs (KPIs)

Output Name	Description
NrCriticalBags	Number of Critical Bags
NrMissedBags	Number of Missed Bags
NrFlights	Number of Flights
NrPassengers	Number of Passengers
NrBags	Number of Bags
NrDeniedPassengers	Number of Denied Passengers
AvgTotalLeadTime	Average Total Lead Time
AvgScansPerMin	Average Number of Scans per Minute
AvgWaitTimePassengerAM [min]	Average Queue Wait Time American Passengers
AvgWaitTimePassengerUN [min]	Average Queue Wait Time United Passengers
AvgWaitTimePassengerAL [min]	Average Queue Wait Time Alaska Passengers
AvgWaitTimePassenger [min]	Average Queue Wait Time Passengers, Weighted
AvgServiceTimePassengerAM [min]	Average Service Time American Passengers
AvgServiceTimePassengerUN [min]	Average Service Time United Passengers
AvgServiceTimePassengerAL [min]	Average Service Time Alaska Passengers
AvgServiceTimePassenger [min]	Average Service Time Passengers, Weighted
AvgProcessingTimeBagAM [min]	Average Processing Time American Bags
AvgProcessingTimeBagUN [min]	Average Processing Time United Bags
AvgProcessingTimeBagAL [min]	Average Processing Time Alaska Bags
AvgProcessingTimeBag [min]	Average Processing Time Bags, Weighted
AvgQueueLengthAM	Average Passenger Queue Length American Counters
AvgQueueLengthUN	Average Passenger Queue Length United Counters
AvgQueueLengthAL	Average Passenger Queue Length Alaska Counters
AvgNrBagsBeforeScannerAndConveyor	Average Number of Bags Before Scanner and Conveyor

3.7 Verification

Verification is a crucial aspect of simulation model development. Verification is the act of confirming that the simulation is working properly. This can be done by observing if the model's behavior matches the real system's behavior. In addition, KPIs can be investigated to see if the model results in unexpected KPI values.

The first check of verifying the model's behavior is checking the distribution of passenger arrival times over the day that the java code results in. Data for this evaluation is saved within a dataset during model runtime. Figure 33 shows the number of passengers that arrive at the airport during 10-minute increments of the day by the flight number of the passenger. Multiple passenger arrival peaks are present; these are during the times at which check-in time windows of multiple flights overlap.

In order to achieve additional insights, the same data is presented in another way. Figure 34 shows how passenger arrivals at the airport are broken down by the respective airline the travelers are flying with. Looking at the first arrival peak between 4:10 and 5:20 am, it can be recognized that within these 10-minute intervals, roughly 10 United and 15 American passengers arrive. This is due to lower combined seat capacities of the first two United flights compared to the first two American flights.

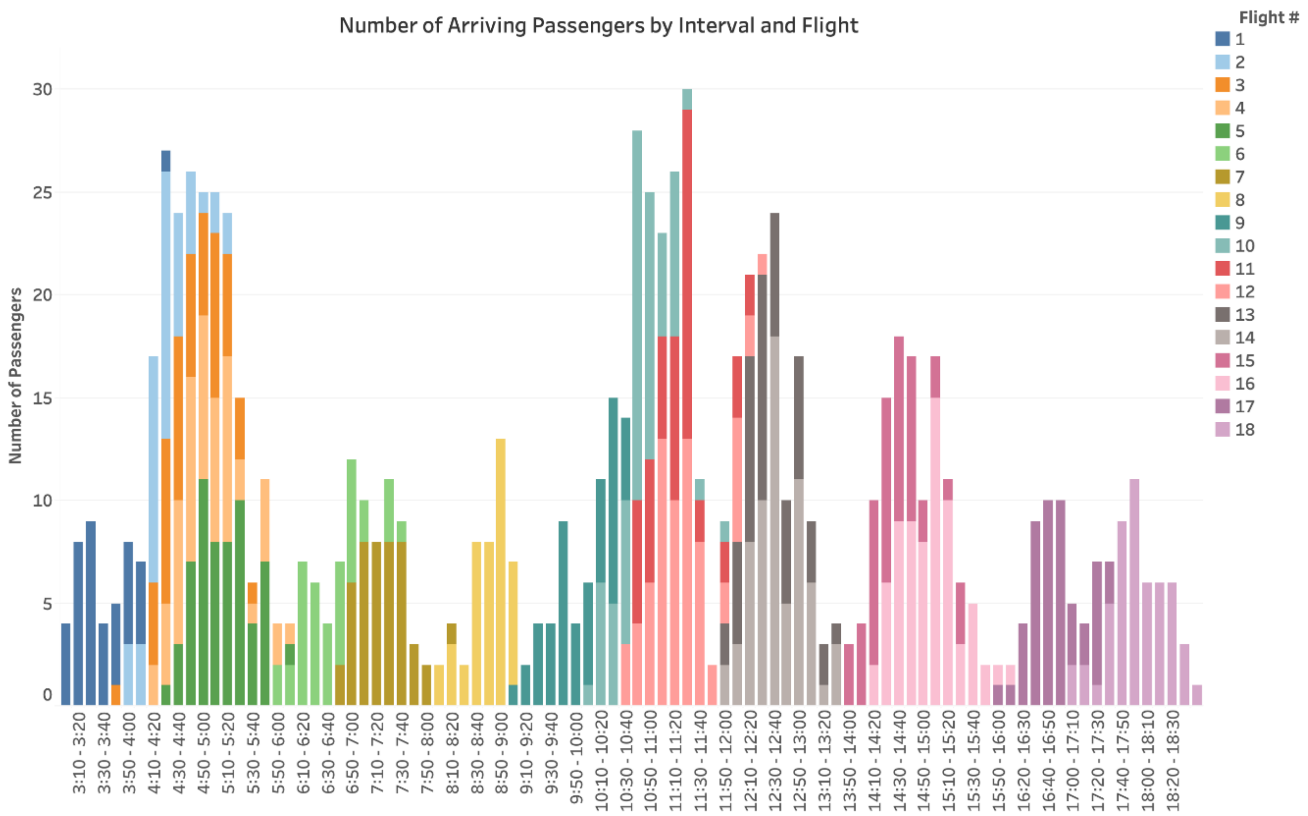


Figure 33: Simulated passenger arrivals at the airport by flight

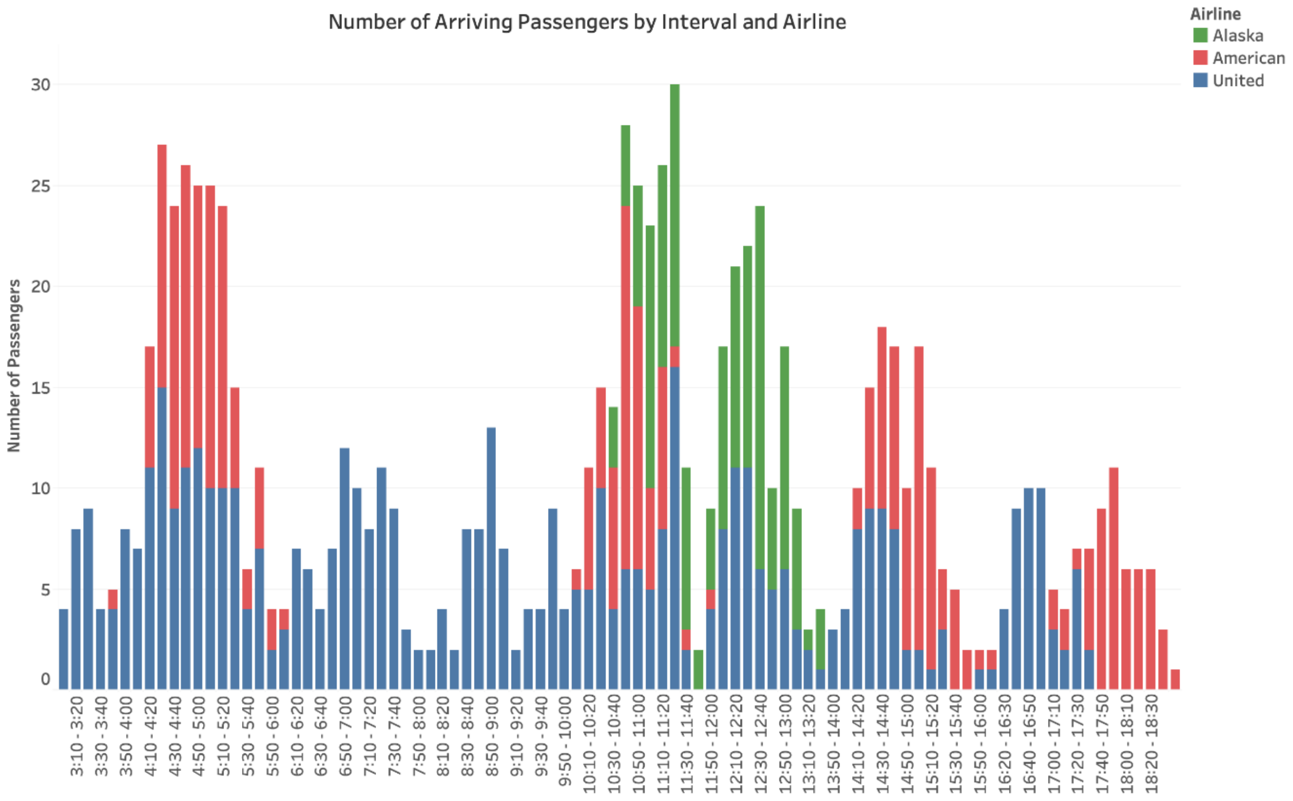


Figure 34: Simulated passenger arrivals at the airport by airline

In order to take variation into account, an experiment that automatically runs the model 20 times in a row while using different random seeds was created. Output values that are listed above are automatically saved into an Excel file for each simulation run. While executing the simulation runs, the logic flowchart and animations are not visible. Instead, histograms of the output values' distributions are presented. For instance, after 20 simulation runs, the histograms of averages for the total lead time and the processing time per bag looks as presented in Figure 35.

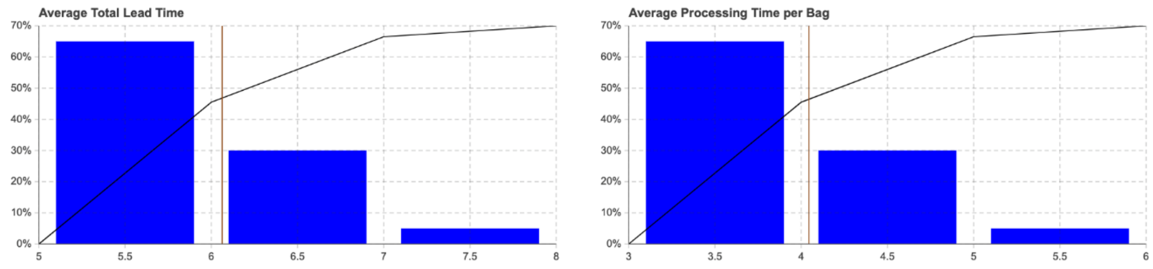


Figure 35: Histograms of simulation output values

The full collection of all values of all outputs from all simulation runs is shown in Table 15 in Appendix A. The average values from the 20 simulation runs for all outputs can be seen in Table 9, which shows realistic values.

Table 9: Output value averages from 20 simulation runs of the base model

Base Model	
Output Name	**Averages**
NrCriticalBags	0,15
NrMissedBags	0
NrFlights	18
NrPassengers	945,3
NrBags	477,35
NrDeniedPassengers	0
AvgTotalLeadTime	6,06
AvgScansPerMin	1,85
AvgWaitTimePassengerAM [min]	0,87
AvgWaitTimePassengerUN [min]	0,27
AvgWaitTimePassengerAL [min]	0,55
AvgWaitTimePassenger [min]	0,50
AvgServiceTimePassengerAM [min]	2,26
AvgServiceTimePassengerUN [min]	1,54
AvgServiceTimePassengerAL [min]	1,92
AvgServiceTimePassenger [min]	1,82
AvgProcessingTimeBagAM [min]	4,25
AvgProcessingTimeBagUN [min]	3,67
AvgProcessingTimeBagAL [min]	4,63
AvgProcessingTimeBag [min]	4,04
AvgQueueLengthAM	0,53
AvgQueueLengthUN	0,10
AvgQueueLengthAL	0,27
AvgNrBagsBeforeScannerAndConveyor	3,68

3.8 Validation

In order to validate the model, the real and simulated time series of luggage and person throughputs are compared. The granularity of both datasets is by the hour of the day. Tables 16 and 17 in Appendix A present the luggage and person throughput from five different days with equal flight schedules. Additionally, the tables contain 95% confidence intervals for the luggage and person throughputs that were computed for each interval. Note that the sample size of 5 requires to use the t-distribution.

As expected, before the model was calibrated, there were similar patterns but still major deviations between interval values, especially in the person time series, shown in Figure 36. The dashed line represents averages from the five days of real data and shows the number of people that passed the security check in each interval. Simulated averages of 20 simulation runs are visualized by the solid line. The mean absolute error between these time series is 26.06.

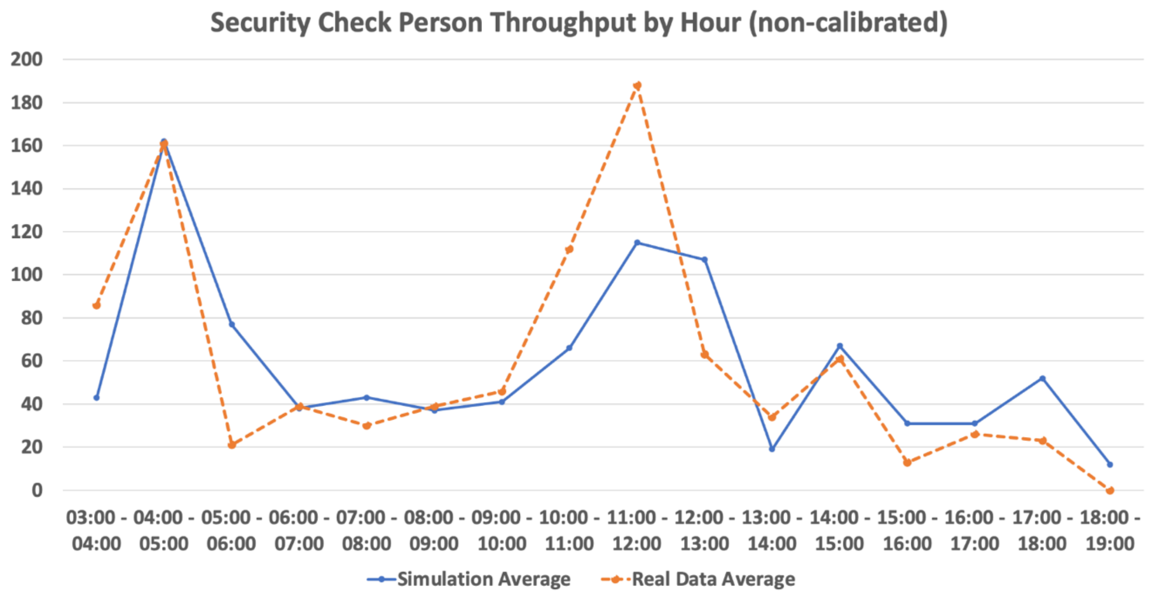


Figure 36: Security Check Person Throughput by Hour (non-calibrated)

With a mean absolute error value of 6.56, the time series of TSA luggage scanner throughputs are more similar to each other. The dashed line in Figure 37 shows the interval averages of real data. Simulated averages are represented by the solid line. Deviations during peak hours, such as towards the end of the day, can be seen.

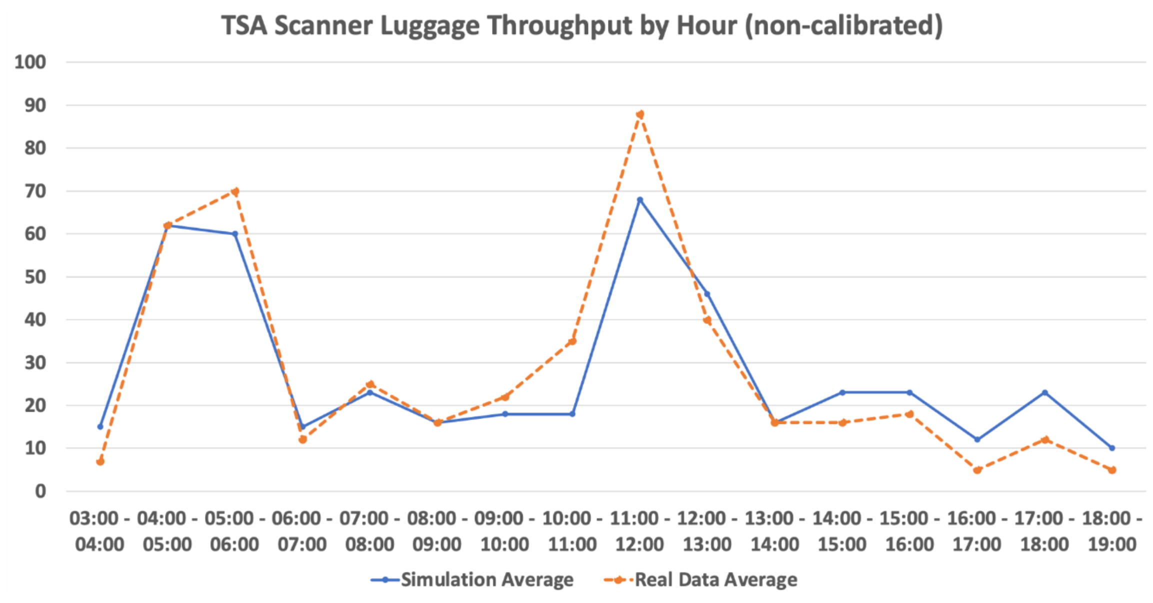


Figure 37: TSA scanner luggage throughput by Hour (non-calibrated)

While passenger and luggage time series are connected, the focus will be on matching the real and the simulated luggage throughput time series. Deviations of passenger throughput are undesirable as well, however, modeling the luggage throughput correctly is more important as the simulation model is supposed to help luggage-related answers.

The simulation model calibration included the following changes: altering passenger load factors of flights increase or decrease the amount of passengers that arrive in certain intervals. Further, two additional passenger arrival distributions that both have slightly bigger and bigger passenger shares arriving later compared to the early distribution were introduced. Besides, two additional distributions that assign the passenger's amount of bags were incorporated. One resulting in a smaller amount of average bags per passenger, and the other resulting in a larger amount. Detailed adjustments by flight can be seen in the model simulation input chapter.

With implemented adjustments, the model was run 20 times again. The measures taken had a minimal impact on the mean absolute error of the person time series, an improvement from 23.06 to 22.81 was achieved. It could be argued that this minimal improvement is due to the model's random-based nature. The TSA scanner throughput time series' similarity improvement was significant. The mean absolute error changed from 6.56 to 3.56. Figure 38 shows the that the solid simulation line is very similar to the dashed line of real data.

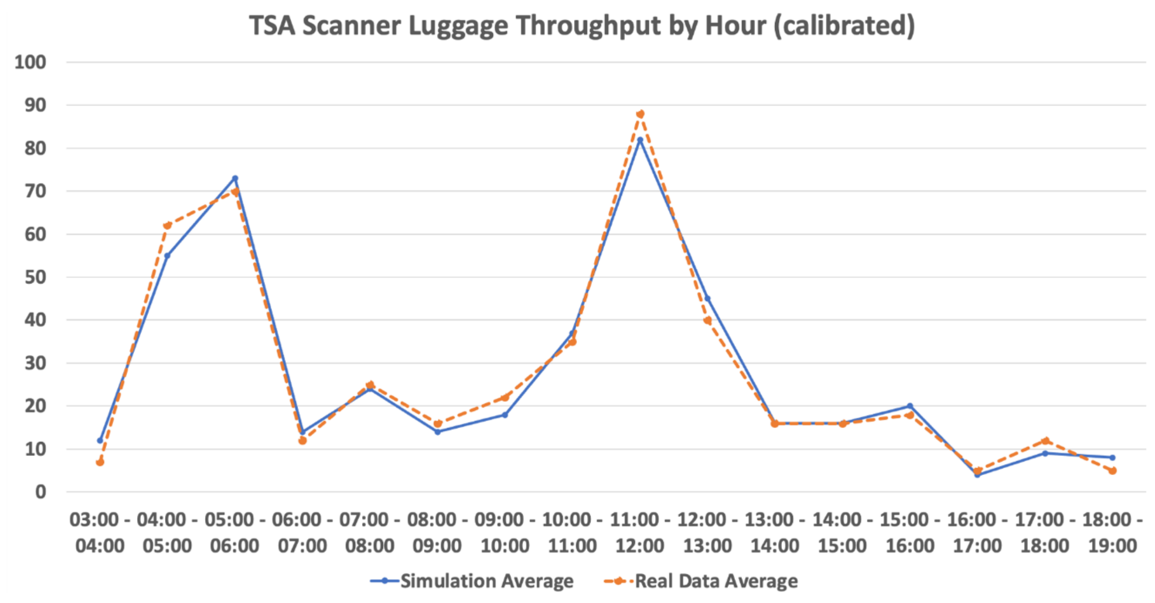


Figure 38: TSA scanner luggage throughput by Hour (calibrated)

As a final step of validation, the 95% confidence intervals of luggage throughput by hour that were derived from real data are compared to the simulated averages. Figure 39 shows the confidence interval limits as lines; the solid line represents the lower limit, while the dashed line represents the upper limit. The unknown true mean of the population lays within these limits at a 5% probability of error. The graph shows that 13 out of 16 simulation averages which are shown as red dots are within respective confidence intervals which is considered as sufficient to accept the model as validated.

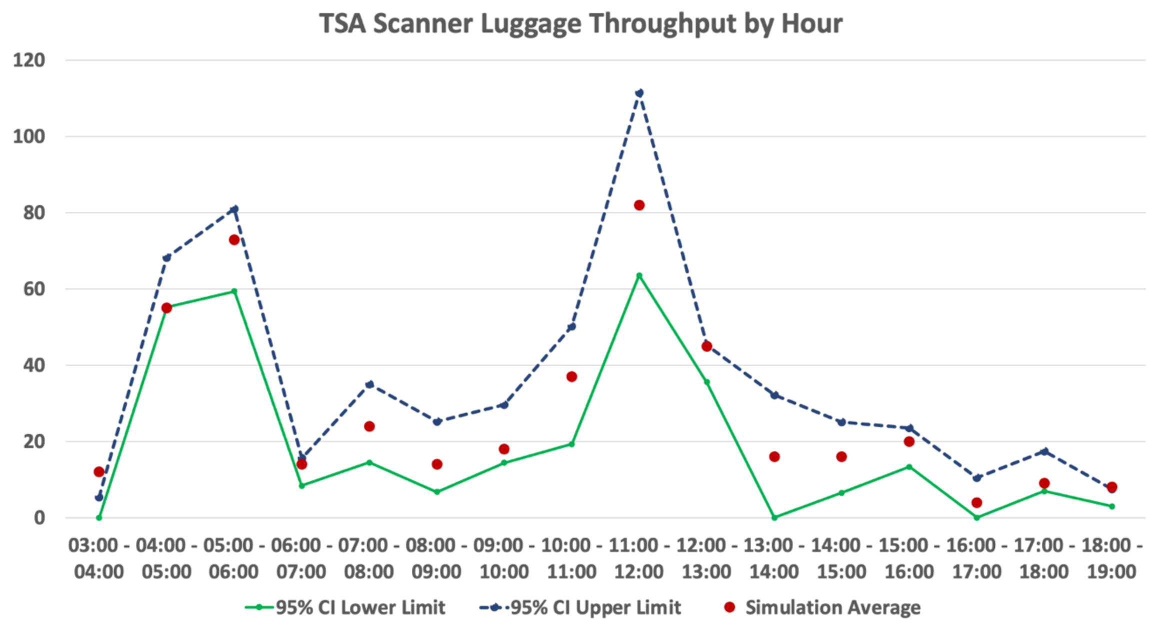


Figure 39: TSA scanner luggage throughput by hour with confidence intervals

4 Expanding the Flight Schedule

When simulating the flight schedule of February 2020 with 18 flights, the luggage handling system is not overburdened as 20 simulation runs showed that this does not result in missed bags. Within this chapter, additional flights will be added to the simulation model. Aircrafts used for extra flights have a passenger capacity of 68. Their passengers follow the early passenger arrival function such as the normal luggage distribution. While observing model behavior and KPI outputs, the number of additional flights is varied in several experiments.

4.1 Optimization Experiment

Anylogic allows setting up experiments, such as optimization experiments. These are characterized by an objective function that is either minimized or maximized. The experiment runs the model multiple times and adjusts up to seven selected simulation parameters in a defined range. This way, many parameter value combinations are tested for their effect on the objective function. Depending in the experiment's setup, it either tries to minimize or maximize the outcome of the objective function.

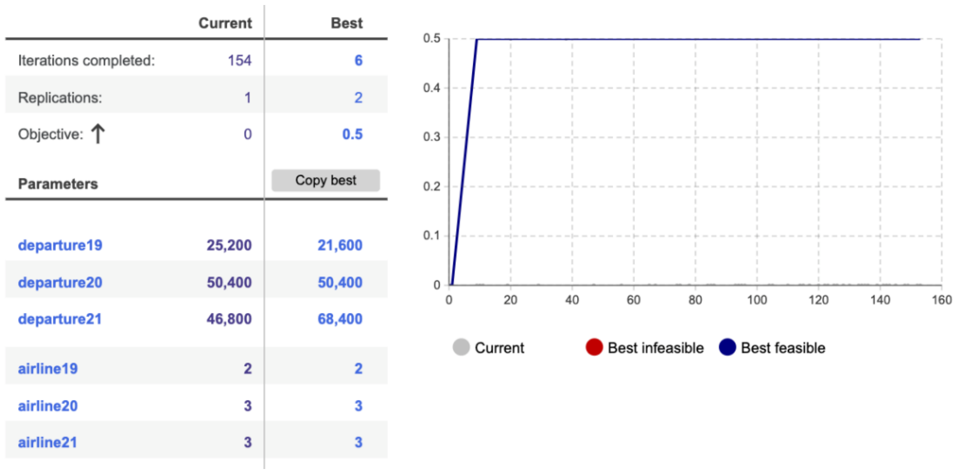
In this experiment, the objective function determines the number of missed bags for each simulation run. Throughout the simulation runs, the objective function is tried to be maximized by parameter adjustments. The simulation parameters that are selected to be adjusted throughout the experiment iterations are departure time and the airline of additional flights. This selection was made because the original flight schedule results in peak times of passenger and luggage volume. By having the experiment adjust departure and airlines, problematic scheduling for a specific amount of extra flights can be identified and transferred into the normal model.

Departure parameters' ranges and step sizes are defined in such a way that the flights can depart on full hours between 5am and 7pm. Airline parameters can take the values 1, 2, or 3, which correspond to American, United, or Alaska Airlines, respectively. This way, three additional flights are randomized by two parameters each. Any flight on top of that has its departure and airline defined by referring to one of the three randomized ones. For instance, the fourth additional flight will depart 15 minutes after the first one and shares the identical airline. The fifth one will apply the same pattern on the parameters of the second flight, et cetera. The experiment is set up to perform 2 replications of all parameter value combinations tested.

Passenger's demand for check-in services is greatly varied in this experiment. This is compensated by changing the number of airline employees to a fixed value of 3 for each airline.

4.2 Experiment Interface

The experiment's graphical user interface is presented in Figure 40. It contains a list of the current parameter values that are tested, such as the parameter values that were found to be the best combination yet. Additionally, the value of the objective function is presented as an average of the two performed replications. Departure values are shown in seconds. During runtime, the graph is showing the development of the highest average amount of missed bags by the number of iterations. For example, the best combination listed below is scheduling a 6am flight with United Airlines, along with a 2pm and 6pm flight with Alaskan Airlines.



	Current	Best
Iterations completed:	154	6
Replications:	1	2
Objective: ↑	0	0.5
Parameters		Copy best
departure19	25,200	21,600
departure20	50,400	50,400
departure21	46,800	68,400
airline19	2	2
airline20	3	3
airline21	3	3

Figure 40: Graphical user interface of optimization experiment

If the objective function is maximized identically with different parameter combinations, the experiment will show the first combination that resulted in the maximized objective function value.

4.3 Testing Additional Flights Scheduled During Various Times of the Day

For testing the system's sensitivity for extra flights, the day was separated into four intervals in which new departures are tested. Table 10 shows the times of the day that represent the early morning peak, the less busy morning, the afternoon peak, and the less busy evening, which can all be recognized in Figure 17, where check-in time windows are shown. The intervals might have gaps in between, which makes sure that check-in time windows from new flights do not overlap with existing time windows from flights from other times of the day.

Table 10: Selected times of the day for testing extra flights

Time of the Day	Departures Between
Early Morning Peak	05:00 - 07:00
Morning	08:00 - 11:00
Afternoon Peak	12:00 - 14:00
Evening	16:00 - 19:00

All times of the day are tested for the number of flights that do and do not result in significant numbers of missed bags. The procedure to iteratively test more additional flights includes running the experiment and implementing the problematic schedule into the regular model. Following that, the model is run 20 times with respective parameter combinations. Finally, the histogram for the number of missed bags from the 20 runs is checked. All histograms can be found in Figures 42 to 49 in Appendix A. The histograms also provide the cumulative density function, such as the mean. Table 11 gives an overview of scenarios that were tested. It shows the scenario number, the time of the day, the number of extra flights, averages of missed bags from the experiment from 20 simulation runs, and the departure and airline combination that the experiment labeled as a good combination. A brief discussion for each scenario is included with the histograms in Appendix A.

Table 11: Overview of tested scenarios

			Maximum Amount of	Average Amount of	Best Parameter Combination Found					
	Time of the	Number of	Missed Bags (Average	Missed Bags from	Departure	Airline	Departure	Airline	Departure	Airline
Scenario	Day	Addiional Flights	of 2 Replications)	20 Simulation Runs	Fligth 19		Fligth 20		Fligth 21	
1	Early Morning	1	9,0	1,7	06:00	United				
2	Early Morning	2	20,5	14,0	06:00	United	06:00	United		
3	Morning	4	2,0	0,5	09:00	United	10:00	United	09:00	United
4	Morning	5	6,6	4,5	11:00	Alaska	11:00	Alaska	11:00	Alaska
5	Afternoon	3	33,00	0,50	12:00	United	13:00	United	13:00	United
6	Afternoon	4	32,50	27,00	13:00	Americ	13:00	United	13:00	United
7	Evening	4	4,50	0,00	18:00	United	17:00	United	18:00	Alaska
8	Evening	5	9,00	5,50	16:00	Americ	16:00	United	16:00	United

Table 12 summarizes the results from each scenario test by looking at the average value of missed bags from 20 simulation runs. If the average is smaller than or equal to 2, the number of added flights from this scenario is entered in the “Not overburdened” column which shows how many extra flights are feasible by time of day. If the average is bigger than 2, the number of flights from the respective scenario is entered in the “Overburdened” column that shows highly infeasible numbers of flights for this time of day.

Table 12: Overview of scenario results of 20 runs by time of the day

		Number of Additional Flights that result in the Baggage System being:	
Time of the Day	**Departures Between**	**Not overburdened**	**Overburdened**
Early Morning Peak	05:00 - 07:00	1	2
Morning	08:00 - 11:00	4	5
Afternoon Peak	12:00 - 14:00	3	4
Evening	16:00 - 19:00	4	5

5 Scenario Analysis

The main purpose of the following scenario analysis is the identification of the luggage system's bottleneck. For this cause, the model is expanded by 6 extra flights. Referring to the number of flights that cause missed bags in Table 12, 2 extra flights in the early morning at 6am and 4 extra flights in the afternoon at 1pm are added to purposefully create missed bags. Early morning flights are operated by American, the afternoon flights by American, United and Alaska Airlines. The model with expanded flight plan is run 20 times and output data is saved. Averages of the outputs are shown in Table 13. For instance, the number of critical and missed bags increased significantly to over 40 and 23, respectively.

Table 13: Output values from the expanded flight schedule

Model Output Value Averages	**Scenario**
Output Name	**A**
NrCriticalBags	40,5
NrMissedBags	23,4
NrFlights	24,0
NrPassengers	1289,8
NrBags	628,1
NrDeniedPassengers	0,0
AvgTotalLeadTime [min]	16,1
AvgScansPerMin	1,8
AvgWaitTimePassengerAM [min]	2,9
AvgWaitTimePassengerUN [min]	0,9
AvgWaitTimePassengerAL [min]	0,8
AvgWaitTimePassenger [min]	1,6
AvgServiceTimePassengerAM [min]	4,2
AvgServiceTimePassengerUN [min]	2,2
AvgServiceTimePassengerAL [min]	2,1
AvgServiceTimePassenger [min]	2,9
AvgProcessingTimeBagAM [min]	8,0
AvgProcessingTimeBagUN [min]	13,6
AvgProcessingTimeBagAL [min]	17,7
AvgProcessingTimeBag [min]	12,2
AvgQueueLengthAM	6,0
AvgQueueLengthUN	1,6
AvgQueueLengthAL	1,2
AvgNrBagsBeforeScannerAndConveyor	18,1

Following that, changes will be applied to the model. For instance, the reduction of time that the TSA scanner needs to scan one bag could improve baggage throughput and decrease the amount of missed bags. This would prove that the scanner is the system's bottleneck. Output data of each model variation will be compared to output data of the non-changed model's output data. Each scenario will be ran 20 times, and average output values will be compared to the values from the non-adjusted model.

5.1 Overview

In order to differentiate these scenarios from previous ones, they will be referred to by letters.

Scenario A includes no adjustments to the model, showing the resulting output values of the expanded flight schedule.

In scenario B, the possible hourly throughput of the TSA scanner is doubled. This is achieved by adjusting the respective model parameter called "ScanTime" which represents the time needed to scan a single bag from 30 to 15 seconds.

Scenario C has a reduced service times for check-in services. For this, the parameters "CheckIn-FullMax", "CheckInFullMed", "CheckInFullMin", "CheckInKioskMax", "CheckInKioskMed" and "CheckInKioskMamin" are reduced by 20 seconds, each.

Scenario D incorporates a second permanent TSA Agent, by increasing the resource capacity while the flexible one is deactivated.

The next scenario, E has a decreased share of oversized bags. The parameter "BigBagsShare" is reduced from 5 to 1%.

Scenario F has a lower share of bags that result in an alert when scanned. The model's parameter "AlertShare" is adjusted from 10 to 5%.

5.2 Results

Table 14 presents averages over 20 simulation runs of selected simulation outputs for all designed scenarios. For each output, the cell representing the most desirable value is highlighted. Checking the line that shows the average number of missed bags per scenario shows that the faster TSA scanner is the only scenario in which the amount of missed bags is significantly reduced to 1.35 bags. Most scenarios have slight deviations from the non-adjusted model value of 23.35. However, reducing the share of oversized bags seems to increase missed bags the most. This proves that the TSA luggage scanner is the system's bottleneck. Table 18 in Appendix A presents the full output results.

Other output values vary as expected depending on the scenario. For instance, the reduced check-in service time in scenario C results in reduced averages of queue wait times, service times and queue lengths.

Table 14: Averages of selected simulation outputs by scenario

Model Output Value Averages	**Scenario**					
Output Name	**A**	**B**	**C**	**D**	**E**	**F**
NrCriticalBags	40,5	6,2	43,4	42,2	53,3	38,5
NrMissedBags	23,4	1,4	27,9	27,8	35,8	21,6
AvgTotalLeadTime [min]	16,1	8,6	16,0	16,0	18,0	15,8
AvgScansPerMin	1,8	3,4	1,8	1,8	1,8	1,8
AvgWaitTimePassenger [min]	1,6	1,7	0,6	1,8	1,7	1,7
AvgServiceTimePassenger [min]	2,9	3,0	1,6	3,1	3,0	3,0
AvgProcessingTimeBag [min]	12,2	4,5	13,7	11,7	14,0	11,8
AvgQueueLengthAM	6,0	6,6	2,4	6,9	6,3	6,2
AvgQueueLengthUN	1,6	1,6	0,8	1,7	1,8	1,7
AvgQueueLengthAL	1,2	1,1	0,5	1,8	1,3	1,3
AvgNrBagsBeforeScannerAndConveyor	18,1	5,8	21,1	17,5	21,0	17,9

6 Recommendations and Discussion

This chapter summarizes findings from the experimentation, answers the problem statement's questions, and discusses the limitations of the simulation model.

First, it has been identified that the TSA luggage scanner is the bottleneck of the system.

Second, the number of additional flights that can be operated without causing major luggage delays and customer dissatisfaction depends on the time of the day, as shown in Table 12: During the early morning, a maximum of 1 additional flight keeps the number of missed bags low. During morning hours, up to 4 additional flights will result in small numbers of missed bags. Up to 3 extra flights can be added within the afternoon hours. During evening hours, up to 4 additional flights can depart without the baggage handling system being overloaded.

Considering output values of the calibrated model without additional flights in Table 9, the increased number of passengers resulting in the average total lead time to jumping from 6.06 minutes to roughly 16 when excluding the faster scanner scenario. When comparing the base model to the scenario with a faster scanner and more flights, the total average lead time increases by approximately 2 minutes. Besides, when excluding the scenario of faster check-in services, the average wait time in the queue jumps from 0.5 to more than 1.6 minutes across all scenarios.

As the simulation model allows to conveniently adjust the flight schedule in the connected Excel database, it can be used as a tactical decision-making tool for capacity analysis of scenarios. Further, the information of how many additional flights can be added during certain time intervals while not causing major luggage delays could be used to build a time-based pricing model. By requiring airlines to pay an increased fee when scheduling new flights during times that are already busy could compensate for additional effort on the airport's side. At the same time, a pricing model like this might have an effect on ensuring that the additional passenger and baggage volume does not exceed the available capacities.

When airport management decides to increase the capacity of its baggage system by making adjustments to the infrastructure, they could consider self-service luggage drop-off machines with integrated scanning functionality. However, current self-service luggage drop-off stations only introduce baggage into the conveyor system and do not include a screening functionality, as shown in Figure 41. Thus, only machines with included screening technology would allow to increase the capacity of SLO airport's system.

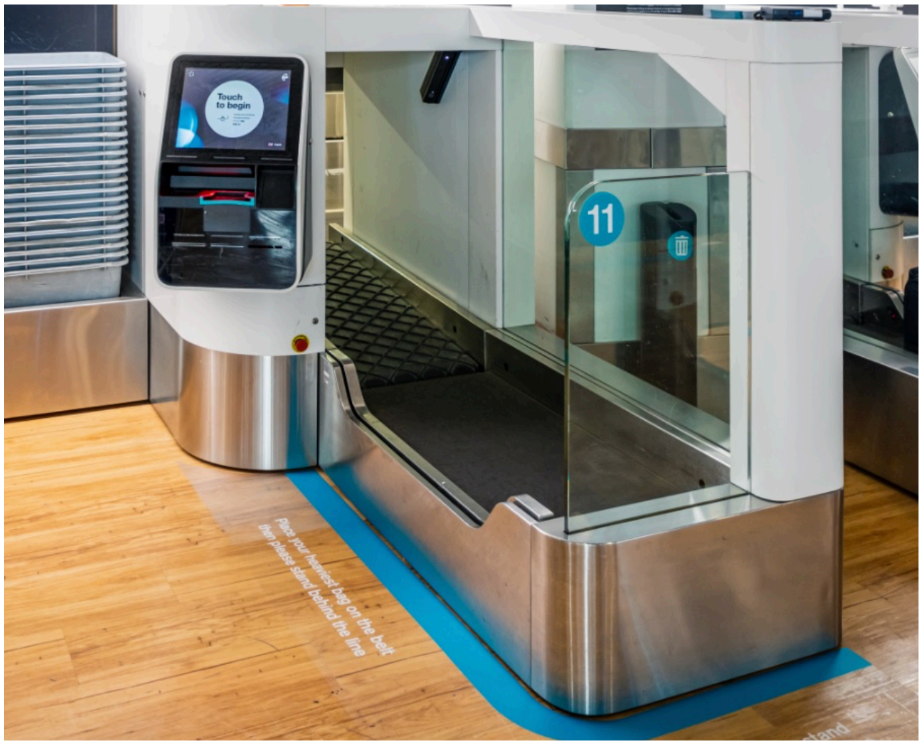

Touch to begin
11
Place your heaviest bag on the belt then please stand behind the line

Figure 41: Self-service baggage drop-off station (Airport Suppliers, 2020)

Even though the simulation model has been designed to mimic the real system as well as possible, it is still limited for various reasons. First, the underlying input parameters such as processing times at check-in counters were selected by best knowledge and airport information. However, they are not based on empirical data. Second, due to non-accessible data on the luggage sorting process and the luggage loading process, the processes are not covered by the simulation model. This diminishes the possibility of having baggage congestion on the conveyor belt caused by the sorting station in the simulation model. A situation like this might occur when the airline workers handling luggage sorting, loading, and unloading are currently busy with loading or unloading luggage. As a result, the simulation model does not account for the effects that this could have on luggage delays.

Further, the simulation model was calibrated to result in similar simulated time series on person and luggage throughputs compared to the time series that were derived from real data. While the model calibration resulted in very similar real and simulated scanner baggage throughput time series, the person time-series still have bigger deviations from each other. This is where the effects of adjusting load factors, passenger arrival patterns, and passenger luggage distributions reached their limits.

Another important difference between the simulation and the real system is that the simulated flight schedule is fixed in terms of departures and used aircrafts. In reality, departures are often delayed, and airlines will change the aircraft in use depending on passenger demand.

7 Conclusion and Future Work

The objectives of this thesis were the identification of feasible additions to the flight schedule as well as the detection of the luggage system's bottleneck. This was successfully achieved by collecting and visualizing the system's process flows as well as their translation into a simulation model. Several experiments were developed that allowed for testing of scenarios whose results enabled answering the above-mentioned questions.

It was found that the capacity of the current luggage conveyor system is able to cover the increased demand created from additional flights, depending on the time of the day, without causing bags to be missed upon departure. Further, the system's bottleneck was found to be the TSA luggage scanner, while other tested model adjustments were found to have minimal impacts on the amount of missed bags.

While the problem statement's questions could be answered by applying simulation, the limitations of the model might still affect the simulation experiment's results. As the model is based on the flight schedule from February 2020, the results of this thesis are connected to the respective flight schedule as well.

However, considering possible future research, the model can be adjusted to run based on any flight schedule. For convenience, this is easy to implement in the Excel database the model is connected to. This way, the simulation model could be used as a tactical decision tool for capacity analysis.

While airlines may be unlikely to schedule new flights based on higher airport pricing as suggested in the recommendations, their requested newly scheduled flights can still be simulated by the model to test for baggage delays. In addition, future changes to the baggage system's entities, such as a check-in process redesign, can be tested before changes are made to the real system by adjusting the model.

References

Wonkyu, K., Yonghwa, P., & Byung Jong, K. (2004). Estimating hourly variations in passenger volume at airports using dwelling time distributions. *Journal of AIr Transport Management*, 395-400.

Postorino, M. N., Mantecchini, L., Malandri, C., & Paganelli, F. (2019). Airport Passenger Arrival Process: Estimation of Earliness Arrival Functions. *Transportation Research Procedia*, 338-345.

Alodhaibia, S., Burdett, R. L., & Yarlagadda, P. K. (2019). Impart of Passenger-Arrival Patterns in Outbound Processes of Airports. *Procedia Manufacturing*, 323-330.

Lin, J. T., Shih, P.-H., Huang, E., & Chiu, C.-C. (2015). Airport baggage handling system simulation modeling using SysML. *2015 International Conference on Industrial Engineering and Operations Management (IEOM)*, 1-10.

Batta, R., Drury, C. G., Appelt, S., & Lin, L. (2007, January). SIMULATION OF PASSENGER CHECK-IN AT A MEDIUM-SIZED US AIRPORT. (WSC, Ed.) *Proceedings of the Winter Simulation Conference*, 1252-1260.

Cavada, J. P., Cortés, C. E., & Rey, P. A. (2017, June). A simulation approach to modelling baggage handling systems at an international airport. *Simulation Modelling Practice and Theory, 75*, 146-164.

Boeing. (2020, October). *boeing.com.* Retrieved November 12, 2020, from Boeing: https://www.boeing.com/resources/boeingdotcom/market/assets/downloads/2020_CMO_PDF_Download.pdf

Van Dijk, N. M., & Joustra, P. E. (2011, December). SIMULATION OF CHECK-IN AT AIRPORTS. (I. C. Socuety, Ed.) *Proceedings of the 2001 Winter Simulation Conference*, 1023-1028.

Egor Yakovlev, T. A. (2019, February). MULTIMETHOD SIMULATION MODELING FOR BUSINESS APPLICATIONS. Chicago: Anylogic. Retrieved from https://www.anylogic.com/resources/white-papers/multimethod-simulation-modeling-for-business-applications/

San Luis Obispo Regional Airport. (2020, April 4). *www.slocounty.ca.gov*. Retrieved October 28, 2020, from https://bit.ly/3e6fw9g

SITA. (2019). *sita.aero.* Retrieved March 2021, from https://www.sita.aero/globalassets/docs/surveys--reports/baggage-it-insights-2019.pdf

San Luis Obispo Regional Airport. (2020, November). *sloairport.com*. Retrieved from https://www.sloairport.com/wp-content/uploads/2020/11/SBP_Route-Map_Nov2020-1024x791.png

Ipsos. (2018, January). *ipsos.com.* Retrieved October 04, 2020, from https://www.ipsos.com/sites/default/files/ct/news/documents/2018-02/a4a-2018-air-travel-survey-topline-02-20-2018.pdf

Airport Suppliers. (2020, January). *airport-suppliers.com*. Retrieved from https://www.airport-suppliers.com/airport_press_release/vilnius-airport-aims-to-cut-check-in-times-with-self-service-baggage-drop-off-stations-as-it-develops-a-new-baggage-handling-system/

Savrasovs, M., Medvedev, A., & Sincova, E. (2009, June). RIGA AIRPORT BAGGAGE HANDLING SYSTEM SIMULATION. 1.

Appendix A

Table 15: Output values from 20 simulation runs of the base model

Base Model	Simulation Run																			
Output Name	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
NrCriticalBags	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
NrMissedBags	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
NrFlights	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18
NrPassengers	936	934	949	954	947	936	945	951	940	951	946	957	939	932	953	935	936	963	951	951
NrBags	490	495	481	486	466	465	476	471	464	490	455	528	460	445	467	465	474	453	506	510
NrDeniedPassengers	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AvgTotalLeadTime	5,53	6,35	5,85	7,36	6,54	5,32	5,49	5,83	5,91	5,43	5,96	6,92	5,69	5,99	6,54	5,67	6,11	5,93	6,87	5,99
AvgScansPerMin	1,86	1,85	1,85	1,85	1,83	1,87	1,84	1,83	1,85	1,87	1,86	1,82	1,86	1,85	1,84	1,84	1,87	1,87	1,85	1,85
AvgWaitTimePassengerAM [min]	0,73	0,81	0,85	1,15	0,72	1,11	1,47	1,26	0,71	0,77	0,71	0,80	0,93	0,83	0,65	0,66	0,76	0,96	0,71	0,88
AvgWaitTimePassengerUN [min]	0,40	0,18	0,38	0,22	0,27	0,44	0,29	0,29	0,18	0,27	0,31	0,14	0,14	0,35	0,29	0,23	0,34	0,24	0,26	0,22
AvgWaitTimePassengerAL [min]	0,32	0,85	0,26	0,35	0,57	0,70	0,70	0,57	0,69	0,21	0,77	0,83	0,52	0,42	0,62	0,28	0,68	0,46	0,44	0,77
AvgWaitTimePassenger [min]	0,49	0,48	0,51	0,52	0,46	0,68	0,72	0,64	0,42	0,42	0,50	0,44	0,45	0,52	0,45	0,37	0,52	0,49	0,43	0,51
AvgServiceTimePassengerAM [min]	2,12	2,27	2,22	2,55	2,09	2,47	2,88	2,60	2,07	2,18	2,08	2,24	2,30	2,22	2,02	2,02	2,20	2,33	2,06	2,34
AvgServiceTimePassengerUN [min]	1,69	1,44	1,67	1,49	1,53	1,72	1,54	1,56	1,47	1,53	1,56	1,44	1,41	1,58	1,56	1,49	1,58	1,47	1,57	1,51
AvgServiceTimePassengerAL [min]	1,73	2,25	1,62	1,72	1,94	2,08	2,05	2,02	2,03	1,62	2,10	2,25	1,84	1,79	1,90	1,66	2,04	1,80	1,89	2,14
AvgServiceTimePassenger [min]	1,83	1,82	1,84	1,85	1,76	2,00	2,03	1,95	1,74	1,75	1,80	1,80	1,76	1,82	1,75	1,68	1,84	1,79	1,77	1,87
AvgProcessingTimeBagAM [min]	4,16	4,24	3,92	4,84	4,78	3,73	3,36	3,71	4,36	3,73	4,30	4,85	3,92	4,67	4,65	3,99	4,80	4,56	4,47	4,03
AvgProcessingTimeBagUN [min]	3,11	3,92	2,96	5,74	4,40	2,84	3,01	3,24	3,48	2,95	3,47	5,36	3,46	3,20	3,94	3,27	3,65	3,37	4,64	3,36
AvgProcessingTimeBagAL [min]	3,33	5,01	5,57	5,25	4,45	3,71	4,12	5,24	4,57	4,30	4,63	3,83	4,02	4,00	6,75	4,71	3,85	4,04	6,03	5,10
AvgProcessingTimeBag [min]	3,52	4,24	3,75	5,33	4,55	3,30	3,33	3,78	3,96	3,48	3,96	4,93	3,71	3,92	4,59	3,77	4,16	3,93	4,82	3,88
AvgQueueLengthAM	0,36	0,54	0,80	0,80	0,27	0,83	0,88	0,92	0,26	0,41	0,43	0,36	0,55	0,54	0,32	0,32	0,40	0,58	0,38	0,60
AvgQueueLengthUN	0,13	0,06	0,20	0,16	0,10	0,12	0,08	0,09	0,04	0,06	0,16	0,03	0,04	0,13	0,11	0,08	0,08	0,11	0,11	0,11
AvgQueueLengthAL	0,10	0,84	0,04	0,16	0,24	0,22	0,23	0,32	0,55	0,07	0,35	0,43	0,12	0,19	0,25	0,05	0,40	0,20	0,16	0,44
AvgNrBagsBeforeScannerAndConveyor	3,21	4,02	3,11	6,02	4,15	2,50	2,35	3,50	3,70	2,64	3,35	5,27	3,16	3,50	3,91	3,32	4,06	3,72	4,70	3,46

Table 16: TSA scanner luggage throughput by hour with confidence intervals

TSA Scanner Luggage Throughput by Hour											
Interval	**Day 1**	**Day 2**	**Day 3**	**Day 4**	**Day 5**	**95% CI Lower Limit**	**95% CI Upper Limit**	**$\bar{x}$**	**σ**	**√n**	**t value**
03:00 - 04:00	7	0	0	0	0	0	5	1,40	3,13	2,24	2,7764
04:00 - 05:00	60	66	60	68	55	55	68	61,80	5,22	2,24	2,7764
05:00 - 06:00	73	72	65	82	59	59	81	70,20	8,70	2,24	2,7764
06:00 - 07:00	11	16	12	8	13	8	16	12,00	2,92	2,24	2,7764
07:00 - 08:00	11	32	27	30	24	15	35	24,80	8,29	2,24	2,7764
08:00 - 09:00	8	12	16	28	16	7	25	16,00	7,48	2,24	2,7764
09:00 - 10:00	18	30	19	16	27	14	30	22,00	6,12	2,24	2,7764
10:00 - 11:00	26	46	46	18	38	19	50	34,80	12,46	2,24	2,7764
11:00 - 12:00	66	79	80	97	116	64	112	87,60	19,32	2,24	2,7764
12:00 - 13:00	34	43	40	41	44	36	45	40,40	3,91	2,24	2,7764
13:00 - 14:00	10	39	9	14	6	0	32	15,60	13,39	2,24	2,7764
14:00 - 15:00	21	25	16	7	10	7	25	15,80	7,46	2,24	2,7764
15:00 - 16:00	14	22	22	14	20	13	23	18,40	4,10	2,24	2,7764
16:00 - 17:00	3	11	3	0	8	0	10	5,00	4,42	2,24	2,7764
17:00 - 18:00	15	12	15	14	5	7	17	12,20	4,21	2,24	2,7764
18:00 - 19:00	5	8	5	5	3	3	7	5,20	1,79	2,24	2,7764

Table 17: Security check person throughput by hour with confidence intervals

Security Check Person Throughput by Hour											
Interval	Day 1	Day 2	Day 3	Day 4	Day 5	95% CI Lower Limit	95% CI Upper Limit	$\bar{x}$	σ	√n	t value
03:00 - 04:00	102	98	69	100	106	77	113	95,00	14,83	2,24	2,7764
04:00 - 05:00	188	187	191	194	134	148	210	178,80	25,19	2,24	2,7764
05:00 - 06:00	34	19	12	21	32	12	35	23,60	9,24	2,24	2,7764
06:00 - 07:00	25	51	39	45	59	28	60	43,80	12,85	2,24	2,7764
07:00 - 08:00	13	41	29	54	29	14	52	33,20	15,30	2,24	2,7764
08:00 - 09:00	41	50	40	37	51	36	52	43,80	6,30	2,24	2,7764
09:00 - 10:00	58	59	69	24	48	30	73	51,60	17,13	2,24	2,7764
10:00 - 11:00	120	154	96	119	132	98	150	124,20	21,15	2,24	2,7764
11:00 - 12:00	264	187	240	174	177	158	259	208,40	40,98	2,24	2,7764
12:00 - 13:00	59	88	72	60	71	55	85	70,00	11,73	2,24	2,7764
13:00 - 14:00	43	42	30	36	39	31	45	38,00	5,24	2,24	2,7764
14:00 - 15:00	71	59	74	68	69	61	75	68,20	5,63	2,24	2,7764
15:00 - 16:00	8	27	10	16	10	5	24	14,20	7,76	2,24	2,7764
16:00 - 17:00	37	38	26	25	21	20	39	29,40	7,64	2,24	2,7764
17:00 - 18:00	41	36	35	14	2	5	46	25,60	16,77	2,24	2,7764
18:00 - 19:00	2	0	0	0	0	0	2	0,40	0,89	2,24	2,7764

The following Figures show histograms that stem from the scenarios shown in chapter 4.1.2. They refer to adding additional flights to the flight schedule and show the number of missed bags from 20 simulation runs, each. Sets of 2 scenarios refer to one time interval of the day.

The histogram of scenario 1 in Figure 42 shows that adding only one additional flight during the early morning peak may result in up to double-digit numbers of missed bags. However the average of 20 simulation runs stays below 2.

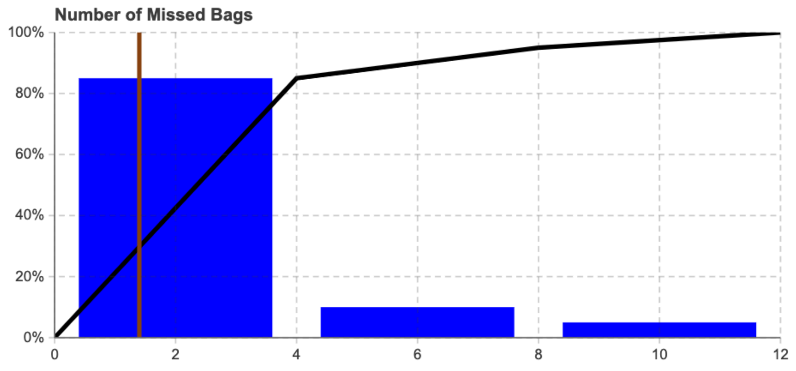


Figure 42: Histogram of missed bags scenario 1

The second scenario incorporates adding two flights during early morning. A histogram resulting from 20 simulation runs in Figure 43 indicates that the conveyor system's throughput demand exceeds the available capacity, resulting in an average of roughly 14 bags being missed. Similar to the first scenario, one range of missed bags occurs in 80% of simulation runs.

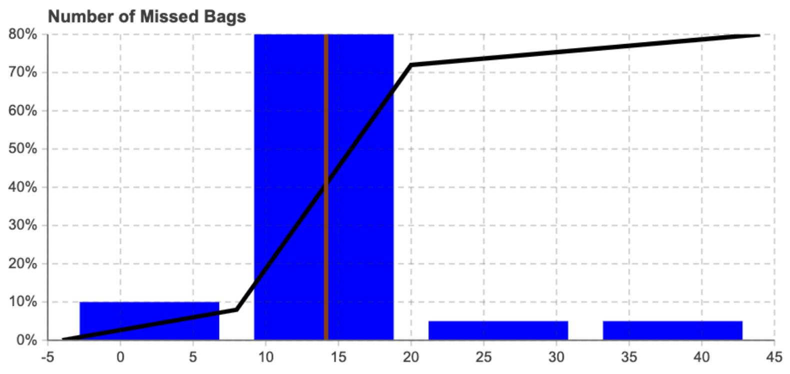


Figure 43: Histogram of missed bags scenario 2

Scenario 3 includes adding 4 extra flights in the morning hours. Figure 44 visualizes how this is not resulting in large numbers of missed bags. With an average of 0.5 bags, this seems to not overburden the system.

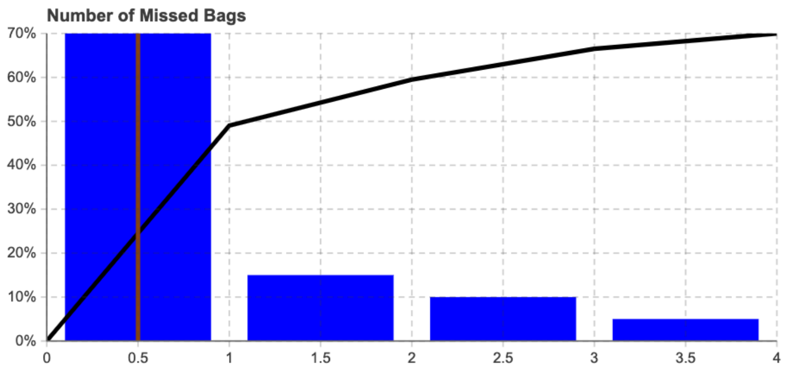


Figure 44: Histogram of missed bags scenario 3

Scenario 4's results are shown in Figure 45. By adding 5 flights during the morning, unacceptable amounts of missed bags will occur as a result.

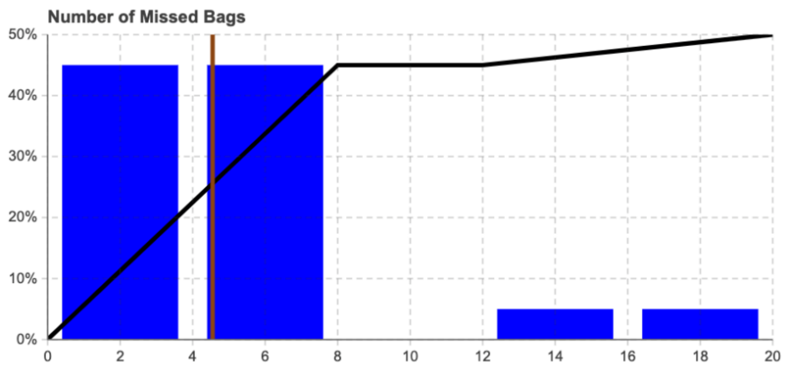


Figure 45: Histogram of missed bags scenario 4

The histogram in Figure 46 shows that the luggage handling system is not overburdened by the luggage throughput demand of scenario 5. Here, 3 flights were added during the afternoon.

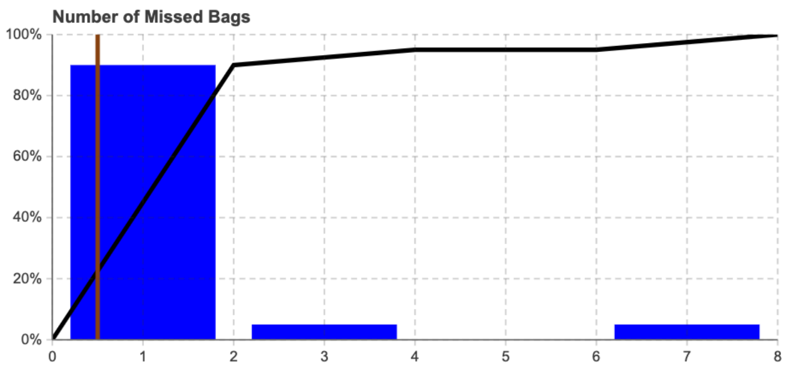


Figure 46: Histogram of missed bags scenario 5

As seen in Figure 47, the system is significantly overburdened by adding 4 extra flights during the afternoon. Scenario 6 results in large number of missed bags in all simulation runs.

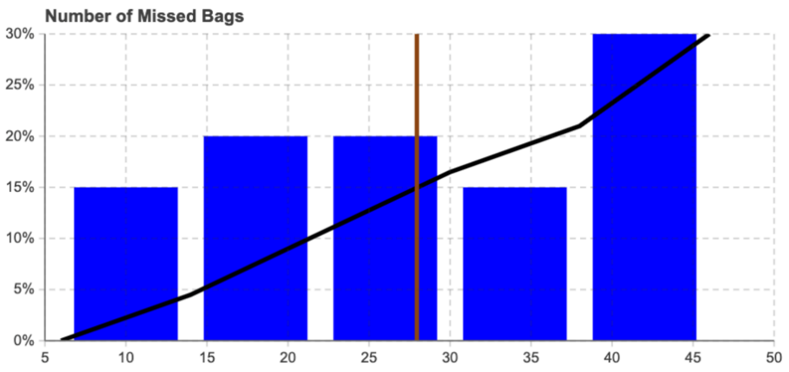


Figure 47: Histogram of missed bags scenario 6

Figure 48 indicates that scenario 7, which incorporates 4 extra flights in the evening, is feasible.

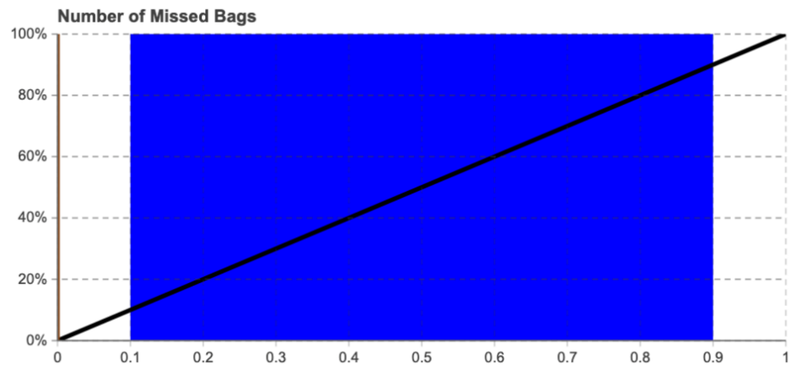


Figure 48: Histogram of missed bags scenario 7

Scenario 8's histogram in Figure 49 shows unacceptable amounts of missed bags. 5 additional flights in the evening seem to overburden the system.

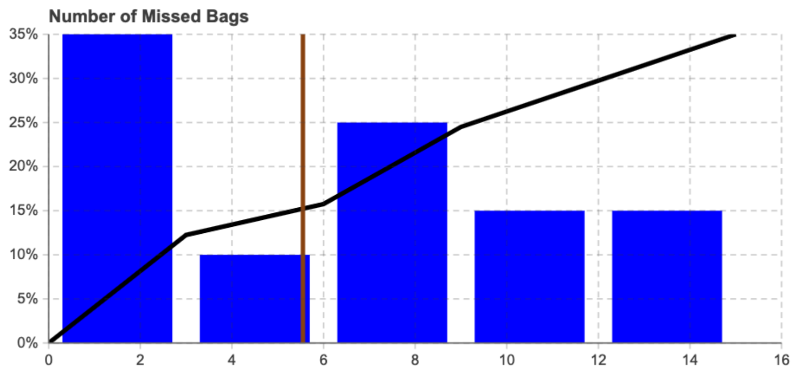


Figure 49: Histogram of missed bags scenario 8

Table 18 shows all output value averages from each scenario.

Table 18: Averages of simulation outputs by scenario

Model Output Value Averages	**Scenario**					
Output Name	**A**	**B**	**C**	**D**	**E**	**F**
NrCriticalBags	40,5	6,2	43,4	42,2	53,3	38,5
NrMissedBags	23,4	1,4	27,9	27,8	35,8	21,6
NrFlights	24,0	24,0	24,0	24,0	24,0	24,0
NrPassengers	1289,8	1290,6	1288,8	1291,4	1289,3	1289,2
NrBags	628,1	632,0	636,2	636,5	642,8	631,4
NrDeniedPassengers	0,0	0,0	0,0	0,0	0,0	0,0
AvgTotalLeadTime	16,1	8,6	16,0	16,0	18,0	15,8
AvgScansPerMin	1,8	3,4	1,8	1,8	1,8	1,8
AvgWaitTimePassengerAM [min]	2,9	3,2	1,1	3,3	3,0	3,0
AvgWaitTimePassengerUN [min]	0,9	0,9	0,4	0,9	1,0	0,9
AvgWaitTimePassengerAL [min]	0,8	0,7	0,3	1,1	0,8	0,9
AvgWaitTimePassenger [min]	1,6	1,7	0,6	1,8	1,7	1,7
AvgServiceTimePassengerAM [min]	4,2	4,5	2,0	4,6	4,3	4,3
AvgServiceTimePassengerUN [min]	2,2	2,2	1,4	2,2	2,3	2,2
AvgServiceTimePassengerAL [min]	2,1	2,0	1,3	2,4	2,2	2,2
AvgServiceTimePassenger [min]	2,9	3,0	1,6	3,1	3,0	3,0
AvgProcessingTimeBagAM [min]	8,0	5,5	8,0	6,2	8,4	7,3
AvgProcessingTimeBagUN [min]	13,6	3,8	15,1	13,9	16,0	13,5
AvgProcessingTimeBagAL [min]	17,7	4,3	22,2	18,3	20,8	16,9
AvgProcessingTimeBag [min]	12,2	4,5	13,7	11,7	14,0	11,8
AvgQueueLengthAM	6,0	6,6	2,4	6,9	6,3	6,2
AvgQueueLengthUN	1,6	1,6	0,8	1,7	1,8	1,7
AvgQueueLengthAL	1,2	1,1	0,5	1,8	1,3	1,3
AvgNrBagsBeforeScannerAndConveyor	18,1	5,8	21,1	17,5	21,0	17,9