TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Operations Management

Models and Methods for Optimizing Baggage Handling at Airports

Dipl.Math.oec. Markus Matthäus Frey

Vollständiger Abdruck der von der Fakultät für Wirtschaftswissenschaften der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor der Wirtschaftswissenschaften (Dr. rer. pol.)

genehmigten Dissertation.

Vorsitzender:		Univ.-Prof. Dr. Martin Grunow
Prüfer der Dissertation:	1.	Univ.-Prof. Dr. Rainer Kolisch
	2.	Prof. Christian Artigues, Ph.D. LAAS-CNRS, Toulouse/France

Die Dissertation wurde am 21.10.2014 bei der Technischen Universität München eingereicht und durch die Fakultät für Wirtschaftswissenschaften am 15.11.2014 angenommen.

Danksagung

Zunächst möchte ich mich bei meinem Doktorvater Rainer Kolisch für die tolle Zeit am Lehrstuhl und die Unterstützung bei der Durchführung meiner Forschung, die zu dieser Arbeit geführt hat, bedanken. Er war mir stets eine große Hilfe und ein guter Ratgeber. Seine herzliche, freundliche Art hat mir vieles vereinfacht und für seinen Rat und Beistand bin ich sehr dankbar. Selbiges gilt für Christian Artigues, der mich in seiner Gruppe in Toulouse warmherzig aufgenommen hat und sowohl zu meiner fachlichen als auch persönlichen Weiterentwicklung beigetragen hat. Mein Dank gilt auch meinen Kollegen Ferdinand Kiermaier, Daniel Gartner und Stephen Starck. Die Zusammenarbeit mit ihnen war und ist mir eine Freude. Ich bin froh darüber, sagen zu können, dass die oben genannten Personen im Laufe der Dissertation zu meinen Freunden geworden sind.

Neben der fachlichen Begleitung und Betreuung ist die persönliche Entwicklung auf dem Weg zum Ziel nicht minder wichtig. Die prägendsten Personen in meinem Leben sind meine Eltern Sonja und Matthäus Frey sowie meine drei Brüder Thomas, Matthias und Christian. Besonders möchte ich meine Mutter hervorheben, die auch bei Rückschlägen stets an mich geglaubt hat. Ebenso bedanken möchte ich mich bei Sylvia und Siegfried Praetorius, die mir immer zur Seite gestanden haben.

Freunde, die mich positiv geprägt und mich in meiner Arbeit fortwährend unterstützt und motiviert haben, sind Marina Schmidmeir, Martin Wyzgala, Christoph Wittmann und Benjamin Roschanski. Ihr habt immer an mich geglaubt und dafür sage ich Euch “ “Danke”.

Augsburg, 20.10.2014 Markus Frey

Contents

List of Tables

List of Figures

Chapter 1

Introduction

> *"First Europe, and then the globe, will be linked by flight, and nations so knit together that they will grow to be next-door neighbors ... What railways have done for nations, airways will do for the world."*
>
> – Claude Grahame-White, 1879 – 1995

1.1 A brief historical review

For centuries mankind dreamed of flying. The human dream of flying can already be found in the Greek legend of "Daedalus and Icarus", where Daedalus and Icarus fabricate wings out of bird feathers and wax. In the 15th century Leonardo da Vinci designed several flying machines. However, his designs had no scientific background, and no prototype was ever created and tested (see Wragg [98]).

It was only in the 18th century that flying was no longer the privilege of gods and angels. In 1783, the French brothers Montgolfier carried out the first manned flight with a balloon (see Gillispie [45]). It would take a further 120 years, before the Wright Brothers constructed the first "heavier than air" aircraft (see Heppenheimer [49]), which was the beginning of a new and successful form of civil transportation which would connect the world as no one would have expected.

At the beginning airplanes were primarily used for military and postal purposes. However, it did not take long until aircraft were also used for civil transportation which led to the opening of the first airports around 1915 in the USA and Europe. These early airports used large grassy areas as runways and aircraft carried between 16 to 20 passengers. Due

Figure 1.1: Passenger and baggage processing in the early years of civil aviation

to technological developments, such as the jet engine in the late 1940s, aircraft became bigger, safer and more comfortable than ever before, which made traveling by airplane more attractive to customers. The resulting increased passenger flow led to the need for bigger airports.

Figure 1.1 shows a picture of civil aviation in the 1950s where passengers are disembarking from an airplane (see Library of Congress [66]). Indeed, up to the 1960s, passengers brought their baggage directly to the airplane on the apron or gate, which made the loading process slow as each bag had to be loaded separately. However, at the beginning of the 1970s, with the era of the big size carriers such as the Boeing 747 having up to 400 seats, the baggage logistics at airports had to change for efficiency and security reasons. Instead of waiting for the passengers at the apron to collect all passengers' baggage, the passenger and baggage flows were separated in the terminal building at the check-in. They dropped off their baggage at check-in counters and the airport transferred the baggage to the airplanes. With the separation, airports could collect passengers' baggage beforehand and load them into airplane cargo holds such that the baggage handling and loading became more predictable, enabling airports to guarantee a given loading time for the airplanes and a weight-balanced aircraft in the air. Modern baggage handling at airports was thus born.

Today, international airports, such as those found in Munich and Frankfurt, use high-

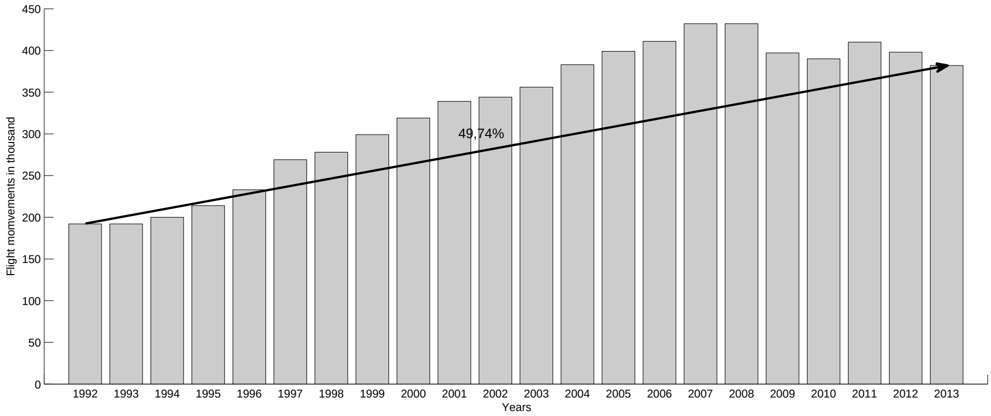


Figure 1.2: Development of air traffic at Munich Airport

tech baggage handling systems (BHS), which are fully automated conveyor belt networks to transfer baggage through terminals. The BHS is also responsible for the baggage safety check by means of advanced security systems.

1.2 Importance of baggage handling

Figure 1.2 and 1.3 show that air traffic as well as passenger transfers have increased during the last 22 years by around 50% at Munich Airport (see Flughafen München GmbH [40]). A similar development can be found at each airport in the world. In 2013 the number of passengers world-wide increased 5.1% to 3.31 billion (see SITA [83]). With the growth of air and passenger traffic, and hence an increase in baggage amount, baggage handling has become more important.

Today for airports, baggage handling is their daily business. Munich Airport, for example, transfers approximately $60,000$ bags per day through the terminals. Baggage handling is one of the major and most challenging tasks at airports and a major indicator of an airport's service quality. Poor baggage handling leads to passenger and airline dissatisfaction. However, the high competition of airports for passengers and airlines makes service quality a crucial factor. Well-organized baggage handling with short transfer times for the baggage from flight to flight (transfer baggage), passenger to flight (outbound baggage) or flight to passenger (inbound baggage) attracts new airlines to the airport as a destination or hub and motivates passengers to use the airport in the future.

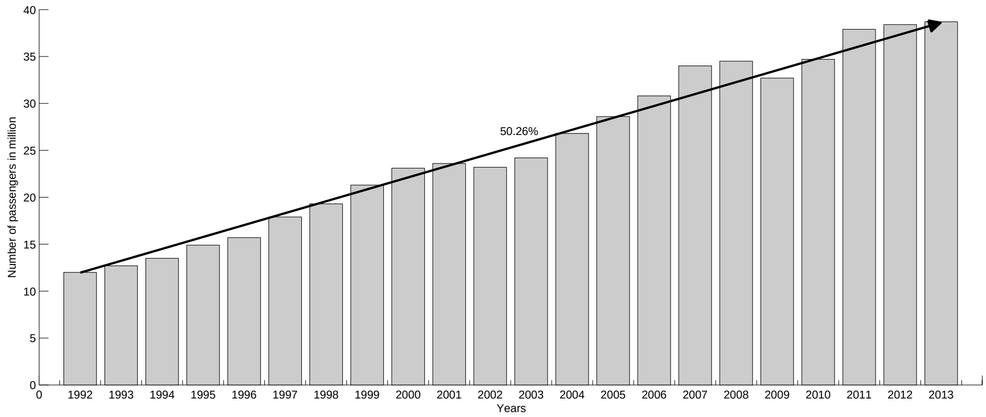


Figure 1.3: Passenger increase at Munich Airport

An efficient usage of the given resources for baggage handling helps to avoid infrastructure expansions which are very costly and constitute political risks. For example, the new satellite terminal for Terminal 2 at Munich Airport will cost around 2.5 billion Euro (see Flughafen München GmbH [39]), while the construction of a third runway was stopped after a referendum in 2012 (see Süddeutsche.de [82]). Moreover, during an expansion, the construction leads to further shortages of resources. Therefore, as the number of passengers is predicted to increase by an average of 5.3% (see SITA [83]) p.a. world-wide to 2016, it is necessary for airports to use the given infrastructure as efficiently as possible. An improved baggage handling also leads to labor cost savings so that the use of costly temporary workers, who are employed to compensate for poor planning, can be avoided.

1.3 Scientific scope

This dissertation studies baggage handling at airports from an operations research (OR) perspective. The planning and control of the baggage flow at the input points, e.g. check-ins, and exit points, e.g. handling facilities, of the BHS are primarily planned by experienced workers, called dispatchers. Their task is to plan the baggage handling such that the given resources are efficiently used. However, due to the huge amount of information, e.g. the amount of baggage to be transferred to a flight or the flight schedule, the task of dispatchers is challenging. Their manual planning often takes a long time and produces rather poor solutions with a poor usage of the available resources. The bad planning often results in costly extensions of the infrastructure or in employing temporary workers carry out the additional work.

This study identifies decision points which can be influenced by means of mathematical programming on a daily operational level. Strategic or tactical decisions, such as the number or size of the (handling) facilities and the number of workers employed to handle the tasks, are not taken into account here. Instead, we plan with the given layout and workforce.

To improve the planning of the baggage handling at airports, we apply state-of-the-art OR-techniques to control the baggage stream at the boundary points of the BHS for the outbound and inbound baggage handling process. The first subprocess describes the transfer of baggage from incoming passengers or flights to departing flights, while the second subprocess plans the transfer of baggage from incoming flights to passengers. By using algorithms for the planning, we show in computational studies, based on real-world data, that an airport obtains a better control of baggage handling with a more robust and efficient planning. Thus, mistakes or bad planning become more predictable. In addition reducing the number of temporary employees, the airport also gains further cost savings through a better usage of the infrastructure which may help avoid costly airports extensions in the future.

Before we analyze the outbound and inbound baggage handling in detail, we provide a detailed process description to handle the baggage flow at airports from the check-in to the airplane's departure and from the arrival to the baggage transfer or baggage claim. Thereby, the baggage handling process is separated into four subprocesses:

1. check-in process
2. outbound baggage handling process
3. transfer baggage handling process
4. inbound baggage handling process.

The primary decisions which have to be taken for each subprocess are presented and the relevant literature discussed. To obtain a deeper understanding of the mathematical structure, a generic mathematical model covering all the subprocesses is presented.

1.4 Overview

The dissertation is divided into three main chapters. Each chapter starts with a description of the content, the purpose and the scientific contribution of the following subsections. The main part presents a detailed problem description and mathematical models. For the study of outbound and inbound baggage handling in Chapter 3 and 4, respectively, we provide a solution methodology and computational results. Each chapter ends with a short summary and further research areas.

In the following we summarize the three chapters as follows: Chapter 2 provides a structured overview of the baggage handling process covering the three main baggage streams outbound, transfer and inbound baggage. An in-depth survey structures past work and classifies relevant solution methods. We present a generic model formulation representing the basic mathematical structure of each main baggage process. The objective of the chapter is to present a unified base for future research in baggage handling, making it possible to consider baggage handling from a holistic view, and to develop integrated solution methods. Our model formulations can be used by researchers and practitioners to obtain first results when examining the baggage handling at airports. The outbound baggage handling process is studied in more detail in Chapter 3. Outgoing flights have to be assigned and scheduled to handling facilities at which the outgoing baggage is loaded into containers. To avoid disruptions of the system the objective is to minimize workload peaks over the entire system. The resource demand of the jobs, which have to be scheduled depend on the arrival process of the baggage. In this paper we present a time-indexed mathematical programming formulation for planning the outbound baggage. An innovative decomposition procedure in combination with a column generation scheme is proposed to solve practical problem instances. The decomposition significantly reduces the symmetry effect in the time-indexed formulation and also speeds up the computational time of the corresponding Dantzig-Wolfe formulation. To further improve our column generation algorithm, we propose state-of-the-art acceleration techniques for the primal problem and subproblem. Computational results based on real data from Munich Airport show that the proposed procedure reduces the maximal workloads by more than 60% in comparison to the current assignment procedure used. In Chapter 4 we consider the planning and scheduling of inbound baggage which leaves the airport through the baggage claim hall. Although, this is a standard process at airports, to the best of our knowledge, there has been no mathematical model proposed in the literature optimizing the inbound baggage handling process. As the inbound baggage handling problem turns out to be NP-hard, we propose a hybrid heuristic combining greedy randomized adaptive search procedure (GRASP) with a guided fast local search (GFLS) and path-relinking. We demonstrate how we implemented the algorithm at a major European Airport where it is in use in order to operate the inbound baggage handling process. In a case study, we compare the results of the mathematical model with the solution of the hybrid heuristic and the solution provided by practice. The proposed algorithm leads to reduced baggage peaks at the baggage carousels and reduced waiting times of passenger for their bags of about 38% and 11% on average, respectively. All computational results are based on real world data. The dissertation concludes with a short summery of the results and outlook for future research in Chapter 5.

Chapter 2

Baggage flows at airports: A survey and a generic model

"True optimization is the revolutionary contribution of modern research to decision processes."

– George Bernhard Dantzig, 1914 – 2005

2.1 Introduction

Increasing passenger volume of about 4% per year (see The Boeing Company Corp. [90]), rising operating costs and high competition force airports to optimize their operations. So far, Operations Research (OR) has focused on the interface between airline and airport like runway planning and gate assignment for aircraft (see e.g. Balakrishnan and Chandran [14], Marín [68] and Dorndorf et al. [32]) and on the interface between passenger and airline or airport, like check–in scheduling or passengers' boarding and de-boarding (see e.g. Bachmat et al. [11], Bachmat et al. [12], Bachmat and Elkin [10], Bachmat et al. [13] and Steffen [84]). However, there have been little applications of OR to baggage handling processes at airports, although at international hub airports, the planning and control of passenger baggage is one of the major challenges. Planing the baggage handling processes impacts key figures of the quality of an airport and consequently an increasing interest in optimized baggage handling can be observed (see SITA [83]). Large European airports like Frankfurt airport, Munich airport and Paris Charles de Gaulles are investing in OR-based technology in order to improve and further automatize their baggage handling processes.

The objective of this chapter is to close the gap between the planning problems for operational baggage handling at airports and OR. For this purpose, we give a detailed description of the baggage handling processes at airports and present relevant literature (a literature review for passenger operations can be found in Tošić [91]). Interdependencies and similarities between the baggage handling processes and other airport processes such as flight positioning or runway planning are discussed. To the best of our knowledge, this is the first work providing a detailed literature overview with a paper classification of baggage processes at airports. The second main contribution of this paper is a generic mathematical model formulation suitable for all discussed baggage handling processes. As we will show, the generic model formulation is NP-hard to solve which also delivers the complexity of all baggage handling processes. For each handling process we present an adapted version of the generic model showing its special structural properties. Thereby, we present a new model formulation for check-in counter planing which can be connected to shift planning models as well as new formulations for the transfer and inbound baggage handling.

In the remainder of this section we provide an overview of the physical parts of an airport, introduce the three main baggage handling processes, the required resources to undertake these processes as well as the actors in charge of planning and controlling them.

Landside and airside An airport can be separated into a landside and an airside part (see Figure 2.1). The landside includes the check-in, parking lots, public transportation stations and access roads. Contrary, the airside of an airport comprises all areas having direct access to aircraft such as runways, taxiways and ramps. The connection between the landside and the airside part is the terminal which contains the baggage handling system (BHS), the central infrastructure for baggage handling. The BHS is an automated baggage transportation system that also provides a storage to temporarily buffer baggage. Bags can enter and leave the BHS from either landside or airside. Figure 2.1 shows the inflows and outflows of baggage at the BHS. On the landside, bags enter the BHS through the check-in (1) and leave the BHS through the passenger's baggage claim (4). On the airside, bags brought along from incoming flights can enter the BHS (3). Bags leaving with an outgoing flight comprise the airside outflow (2).

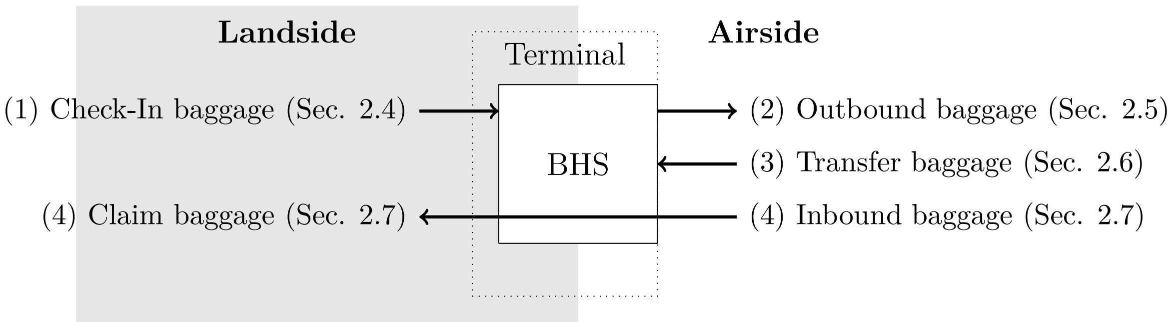


Figure 2.1: Baggage in- and outflow streams at airports

Baggage handling processes Hub airports have four major baggage handling processes which are made up of the four baggage streams presented above: check-in baggage handling, outbound baggage handling, transfer baggage handling and inbound baggage handling.

1. Check-in baggage handling: Check-In baggage is brought in by passengers arriving at an airports' landside. At check-in counters the incoming bags are forwarded to the BHS (1).

2. Outbound baggage handling: Outbound baggage handling encompasses all process steps necessary to forward baggage leaving the BHS to an departing flight (2). The outbound baggage either comes from new arriving passengers through the check-in (1) or from incoming flights (3). Once in the BHS, outbound baggage can either be temporarily stored or it can be forwarded directly to a baggage handling facility where baggage is loaded into container and then transported to the outgoing airplane.

3. Transfer baggage handling: Baggage brought in from an incoming flight that is forwarded to a connecting flight is called transfer baggage. Transfer baggage is typically transported to infeed stations and fed into the BHS (3) from where the further process is controlled by the outbound baggage handling. In urgent cases, transfer baggage can be directly brought from an incoming to an outgoing airplane without traversing the BHS.

4. Inbound baggage handling: Besides transfer baggage, incoming flights also bring along inbound baggage, which leaves the airport through the claiming hall on the airport's landside. Inbound baggage is transported from an incoming flight to an

infeed station and via the BHS send to a baggage claim carousel where it is picked up by the passenger (4).

In Sections 2.4 - 2.7 we will discuss these three baggage handling processes in more detail. Next we briefly describe the resources required for and the actors involved in baggage handling.

Resources In order to operate the baggage handling processes, resources are required. The primary resource is the BHS. Further resources are baggage towing vehicles, baggage containers and human resources (drivers, loaders, ramp agents,...). In Section 2.2, we provide an overview of airport resources necessary for baggage handling and take a specific look at the planning of human resources which are needed for all ground handling tasks.

Actors The baggage handling processes are influenced by different actors. While the BHS's inflows and outflows are planned and steered by the airport, the necessary handling tasks are performed by groundhandler. At deregulated airports, ground handling services have to be assigned not to one but to several companies. The processes of the airport and the ground handler depend on the decisions made by the air traffic control. The air traffic control influences all inflows and outflows of flights by coordinating the arrival and departure times of planes. Finally, the airline is an actor which is responsible for the check-in and therefore for the landside inflow as well as for other relevant processes like container sorting. Table 2.1 provides an overview on how the main actors influence the three baggage handling processes. This chapter focuses on these baggage processes conducted by the airport as given in the upper block of the table.

In Section 2.2 - 2.7 we address the following topics: Section 2.2 provides a detailed description of the airport resources. The generic model suitable for all baggage handling processes is presented in Section 2.3 and its complexity is established. The four main baggage flows - check-in baggage, outbound baggage, transfer baggage, and inbound baggage - are addressed in Section 2.4 -2.7 where each section is structured as follows: First, we provide a detailed description of the baggage handling (sub)processes and discuss the relevant literature in subsection “Literature review”. Afterwards, in subsection “Mathematical model” we formulate the problem as a mixed integer program. Finally, in the paragraph “Future challenges” we point out open research questions and ideas for future

Actors	**Baggage handling processes**		
Task	Outbound	Transfer	Inbound
Airport operators			
Outbound handling	•		
Inbound handling			•
Transfer handling	•	•	•
Flight/Gate positioning	•	•	•
Baggage infrastructure	•	•	•
Government			
Customs		•	•
Airline			
Check-in	•		
Container sorting	•	•	
Ground handler			
Container handling	•	•	•
Staffing	•	•	•

Table 2.1: The influence of airport actors on the baggage handling process

research. Finally, Section 2.8 summarizes our review and provides outlook for further research directions.

2.2 Airport resources

Airport resources can be distinguished in infrastructural resources operated by an airport (e.g. the BHS) and ramp resources operated by ground handlers (e.g. baggage tugs).

2.2.1 Ramp resources

Ramp processes are carried out at the airside of an airport and comprise the loading and transport of containers with passengers' baggage. Required resources are containers, baggage tugs, baggage trailers (dollies or baggage carts), lift trucks and human resources. For the baggage transport in containerized carriers bulk containers or (sometimes) pallets are used. Bulk containers are also called unit load device (ULD) because they are standardized in size to ease the loading into different airplane types. In the following, we will speak of container if we do not need to distinguish between bulk container and palette. If the bag is transported without any container (non-containerized airplane) then we have bulk baggage. Containers are transported on dollies and bulk baggage is transported on baggage carts. Each dolly can transport one container at a time. A baggage tug which is an electric or gasoline powered vehicle can tow up to 6 dollies or baggage carts. Lift trucks are used to load containers or bulk baggage in the cargo hold of airplanes.

2.2.2 Infrastructure resources

The main infrastructure of an airport is given by the BHS which represents the most important and complex of all airport resource. The BHS automatically transports baggage through the airport and sorts baggage according to the departing flight. Figure 2.2 provides an overview of the parts of the BHS: check-in counters, infeed stations, the baggage screening system, the baggage sorting network, a storage system, handling facilities (baggage carousels or chutes) and baggage claim carousels. The BHS and its components have direct influence on passengers' perceived level of service (see Pagani et al. [73]). In general, the design and size of the BHS is a strategical decision of an airport, while the BHS processes such as the baggage routing are operative decisions. A detailed description of the operations of the BHS is given in Hallenborg and Demazeau [47], Johnstone et al. [55], Khosravi et al. [59] and de Neufville and Odoni [30]. Neufville [71] discusses design difficulties in installing and operating the BHS at Denver International Airport. The performance of the BHS is measured by the required time a bag need to be transfered from one point to another point within the system. But also the lost rate of bags is crucial to guarantee a level of service for passengers. In what follows we present the components of the BHS in more detail.

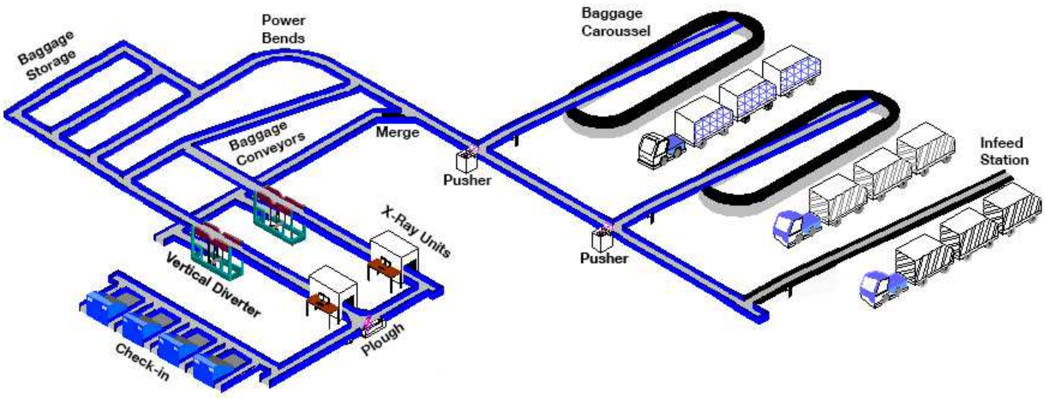


Figure 2.2: A BHS with its components

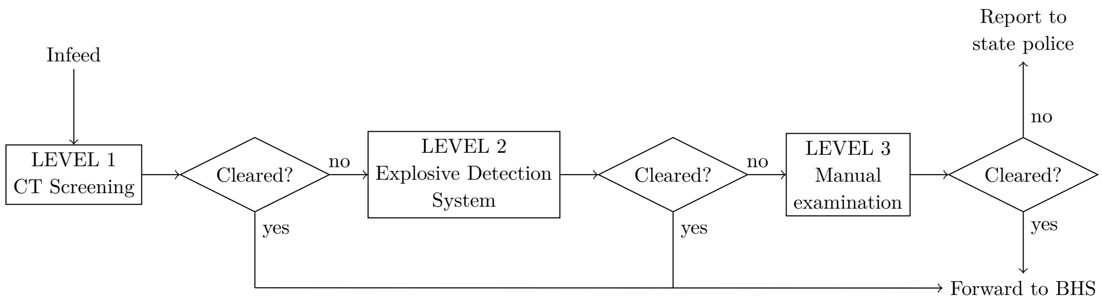


Figure 2.3: A three level screening system used in the US (see de Barros and Tomber [28])

Infeed station Infeed stations are the access points for baggage into the BHS (see Figure 2.2). The infeed stations are either the check-in counters at the landside (see (1) in Figure 2.1), or the transfer and inbound baggage infeed stations at the airside (see (3) in Figure 2.1). At check-in counters the bag is dropped either by an airline agent or in case of self check-in by the passenger on the conveyor belt. At transfer infeed stations the unloading of bags from the container to the conveyor belt is a manual process which is sometimes mechanically supported by lift-aids. A check-in bag is given an identification tag that indicates its itinerary and contains a unique bar code, the Baggage Source Message (BSM). The BSM identifies a bag worldwide as long as the bag does not leave the airside of an airport. The BHS transports baggage using the BSM to its handling facility, the central storage system or to the baggage claim carousel. Le et al. [62] use simulation and binary search to determine optimal infeed rates at infeed stations to minimize the bags' flow time and to maximize the throughput of bags.

Baggage screening A checked in bag is screened before it is forwarded further into the BHS (see Figure 2.2). Depending on the security standard of the preceding airport, transfer baggage may be also screened before it is forwarded through the BHS. Particular due to the risk of terroristic attacks at airports after September 11, the security at airports has received more attention.

Baggage sorting network Within the BHS the baggage transport can be realized by different transportation devices. Conveyor belts as used at Terminal 1 of Munich Airport, tilt trays used at Terminal 5 of London Heathrow Airport, totes or plastic boxes employed at Terminal 2 of Munich Airport, or destination-coded vehicles (DCV), automatically guided vehicles which carry one bag on rails to its destination as used in Toulouse Airport. The latter three transportation systems are preferred because the sorting of baggage within the BHS in order to automatically direct bags to its destination is more flexible and can be easer realized (see Johnstone et al. [55]). Yu and Xu [103] discuss key issues when designing a conveyor belt network and combining them with programmable control technologies.

Storage system Storage systems as part of the BHS have the task to store baggage which can not be immediately directed to a handling facility. The storage consists of several parallel lanes to store the baggage (see Figure 2.2). Each bag in a lane can be individually sent to a handling facility. If the addressed bag is not stored in the front of the lane and thus can not be immediately accessed, the baggage blocking the addressed bag has to be moved. Storage systems either have a centralized or a decentralized architecture. In the first type the storage system is located central in the BHS, while in the latter type several smaller subsystems are used.

Handling facility At the handling facilities groundhandler load the arriving bag into containers. The location of the handling facilities are either centralized or decentralized. Typically, decentralized handling facilities are located very close to the departing flight, that is directly in the pier or gate where the flight is positioned (see Abdelghany et al. [2] and Ascó et al. [7]). In contrast, at centralized handling facilities all facilities are located in one central baggage hall which is located in a more remote position from the departing flight. While new airports like the Terminal 2 of Munich Airport use a centralized infrastructure, older airports such as Frankfurt Airport with its grown infrastructure use both centralized and decentralized handling facilities.

A handling facility is either an oval-shaped (baggage carousel) or a lane based (chute) conveyor belt. While the first type can store more bags at a time on the conveyor belt, the latter type requires less space in the baggage hall. The conveyor belt capacity is given by the number of bags which can be placed on a conveyor belt at a time. Lane based handling facilities are often used in a decentralized environment, which only allows one flight at a time to be handled. Baggage carousels handling facilities allow to handle several flight simultaneously.

In general, the number of flights handled at a handling facility is restricted to the number of available working stations and parking positions for containers. A working station comprises a part of the handling facility's conveyor belt. Containers or dollies are lined up on parking positions parallel to the relevant segments of the conveyor belt. Each working station is equipped with a display and a scanning device. The display provides necessary information about the handled flight such as the number of expected bags, flight's departure time and the destination. The scanning device is used to check whether the bag can be loaded into the container or not. For each bag to be loaded, a scan of the bag's BSM and the barcode of the destination container is performed. If the scans match the bag is loaded into the container, else the bag remains on the conveyor belt of the carousel or is placed next to the handling facility. For security reasons and to reduce the number of mishandled bags, not more than one flight is handled at each working station at a time. At most airports, the loading of baggage into containers is done manually by workers at working stations. But also semi-automatic or full-automatic systems for baggage loading are available. A manually handled working station can be equipped with a loading support system (a lift-aid) which relieves the worker from physically hard work. At semi-automated systems a worker has to control the loading device manually while at fully-automated working stations the loading is done by a robotic system. Automatic systems are used in Amsterdam Schiphol Airport, Netherlands, (automated loading) and Karlstad Airport, Sweden, (semi-automated loading). Most airports do not use semi- or fully-automated systems due to space limitation, high investment costs and low flexibility (see Lenior [65]). The loading rate at a working station is the number of bags loaded into the containers per time interval. The loading rate depends on the used loading system such as manually, semi- or fully automated.

When the handling facility for a flight has already been closed, but there is still baggage for the flight in the sorting network of the BHS it is directed to a "last minute chute". Reasons for the need of a last minute chutes are a late infeed or reading errors of bag's BSM which delay the transport to the handling facility. At last minutes chutes the bag is loaded on a baggage cart (no containers are used) and is directly transported to the corresponding airplane. At last minutes chutes several "rush" or "hot" baggage for different flights can be handled at a time. While Terminal 2 of Munich Airport has dedicated chutes for rush or hot baggage, Frankfurt Airports uses its normal handling facilities.

Baggage claim carousel Baggage claim carousels are the final point of a bag's journey from one airport to another. Baggage claim carousels are grouped in baggage claim halls where arriving passengers pick up their baggage at landside. An infeed station for inbound baggage is either connected via the BHS to more than one baggage claim carousel or the infeed station is directly connected to a dedicated baggage claim carousel. While a direct connection allows a faster processing of the baggage, a connection of an infeed station via the BHS to more than one baggage claim carousel offers greater flexibility and allows to buffer baggage.

Displays for passengers at each baggage claim carousel show the expected and served flights at this carousel. The conveyor belt capacity depends on the size and shape of the baggage claim carousel. The carousel shape differ between airports and terminals (see Ghobrial et al. [44] for different carousel types). Different layouts for baggage claim areas are presented by de Barros and Wirasinghe [29], Gosling [46] and Robusté [79].

2.3 Generic assignment and scheduling problem

The decision problems for the check-in, outbound, inbound and transfer baggage flow presented in this chapter can be mathematically described by a generic assignment and scheduling problem (GASP).

Notation The GASP is defined on discrete planning horizon $\mathcal{T} = \{1, \ldots, T\}$ where each $t \in \mathcal{T}$ defines a unique time interval with all periods t being of equal length. Each flight is represented by a job $i \in \mathcal{F} = \{1, \ldots, F\}$ which has to be planned. When job i is executed, it requires one or several resources $g \in \mathcal{G} = \{1, \ldots, G\}$ with capacity $U_g \geq 0$ per period. Each job i can be processed on the resources in different modes. A mode defines the type and maximum quantity of resources required to process the job. For example, the "check-in of a flight" can be done by using one counter with up to four service lines or two counters with up to two service lines, respectively. Formally, job i can be processed in mode $m \in \mathcal{M}_i$ (with $\mathcal{M} = \bigcup_{i \in \mathcal{F}} \mathcal{M}_i = \{1, \ldots, M\}$) using up to $r_{i,m,g}$ units of resources g during each processing period. Job i has to start its processing within time window $\left[S_i^{\text{es}}, S_i^{\text{ls}}\right]$ and has a given finish time S_i^{e} which generally is set externally by the flight schedule.

Resource allocation Binary decision variable $x_{i,m,t}$ is 1, if processing of job i in mode m starts at the beginning of period t. In order to align a job's resource allocation with the time-varying demand, allocation is done dynamically throughout the job's processing time. We introduce decision variable $z_{i,g,t}$ which stores the number of allocated resources of type g for job i in period t. To guarantee a given level of service, parameter $L_{i,t} \geq 0$ specifies a lower bound for the number of resources that job i requires in period t. For example, based on predicted passenger arrival times for flight i, a minimum number of check-in counters should be opened in period t (see Section 2.4). Finally, for the minimum processing time $p_{i,m,g}$ the full quantity of resource g available in mode m has to be used by job i, i.e. job i allocates exactly the upper bound given by mode m for the usage of resource g. Figure 2.4 illustrates resource allocation for a job i on a resource g.

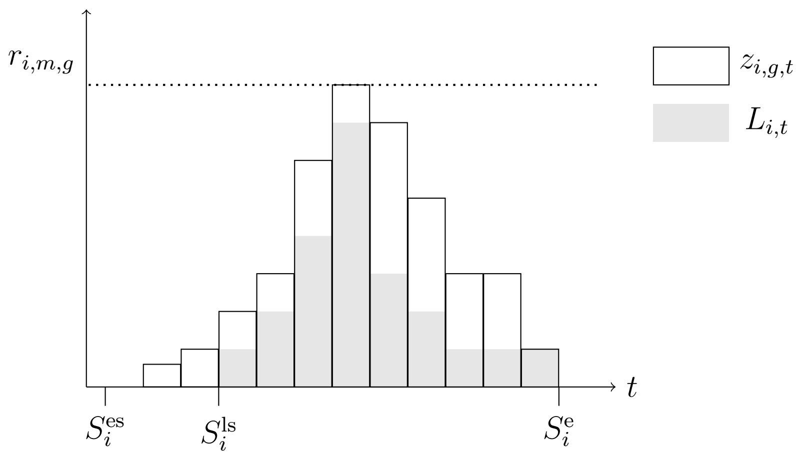


Figure 2.4: Time dependent resource allocation by the GASP.

Model formulation The assignment and scheduling decision determines the flow of passengers or bags. To depict the flow associated with flight i on resource g, we introduce variable $I_{i,g,t}$ which gives the number of passengers or bags of flight i at resource g at the end of period t. Then, the GASP for baggage handling processes can be formulated as follows:

$$\text{Minimize} \sum_{i \in \mathcal{F}} \sum_{g \in \mathcal{G}} \left(\sum_{\tau = S_i^{\mathrm{es}}}^{S_i^{\mathrm{e}}} f^{\mathrm{I,z}}(I_{i,g,t}, z_{i,g,t}) + \sum_{\tau = S_i^{\mathrm{es}}}^{S_i^{\mathrm{ls}}} f^{\mathrm{x}}(x_{i,g,\tau}) \right) \tag{1}$$

subject to

$$\sum_{m \in \mathcal{M}_i} \sum_{\tau = S_i^{\mathrm{es}}}^{S_i^{\mathrm{ls}}} x_{i,m,\tau} = 1 \qquad \forall\, i \in \mathcal{F} \tag{2}$$

$$L_{i,t} \leq \sum_{g \in \mathcal{G}} z_{i,g,t} \qquad \forall\, i \in \mathcal{F}, t \in \left[S_i^{\mathrm{ls}}, S_i^{\mathrm{e}}\right] \tag{3}$$

$$\sum_{i \in \mathcal{F}: t \in \left[S_i^{\mathrm{es}}, S_i^{\mathrm{e}}\right]} z_{i,g,t} \leq U_g \qquad \forall\, g \in \mathcal{G}, t \in \mathcal{T} \tag{4}$$

$$\sum_{m \in \mathcal{M}_i} r_{i,m,g} \cdot \sum_{\tau = S_i^{\mathrm{es}}}^{\min\left\{t, S_i^{\mathrm{ls}}\right\}} x_{i,m,\tau} \geq z_{i,g,t} \qquad \forall\, i \in \mathcal{F}, g \in \mathcal{G}, t \in [S_i^{\mathrm{es}}, S_i^{\mathrm{e}}] \tag{5}$$

$$r_{i,m,g} \cdot \sum_{\tau = \max\left\{t - p_{i,m,g} + 1, S_i^{\mathrm{es}}\right\}}^{\min\left\{t, S_i^{\mathrm{ls}}\right\}} x_{i,m,\tau} = z_{i,g,t} \qquad \forall\, i \in \mathcal{F}, m \in \mathcal{M}_i, g \in \mathcal{G}, t \in [S_i^{\mathrm{es}}, S_i^{\mathrm{e}}] \tag{6}$$

$$I_{i,g,t} = I_{i,g,t-1} + g(z_{i,g,t}) \qquad \forall\, i \in \mathcal{F}, g \in \mathcal{G}, t \in [S_i^{\mathrm{es}}, S_i^{\mathrm{e}}] \tag{7}$$

$$x_{i,m,t} \in \{0, 1\} \qquad \forall\, i \in \mathcal{F}, m \in \mathcal{M}_i, t \in \left[S_i^{\mathrm{es}}, S_i^{\mathrm{ls}}\right] \tag{8}$$

$$I_{i,g,t} \geq 0 \qquad \forall\, i \in \mathcal{F}, g \in \mathcal{G}, t \in [S_i^{\mathrm{es}}, S_i^{\mathrm{e}}] \tag{9}$$

$$z_{i,g,t} \in \mathbb{Z}_+ \qquad \forall\, i \in \mathcal{F}, t \in [S_i^{\mathrm{es}}, S_i^{\mathrm{e}}] \tag{10}$$

Objective function (1) is generic and incorporates the decisions made for the assignment and scheduling of jobs, represented by start variable $x_{i,m,t}$, resource allocation $z_{i,g,t}$, and the number of passengers or bags $I_{i,g,t}$ of job i in mode m at resource g at time t. Constraints (2) ensure that each job is started once within its start time window. Due to constraints (3) the minimal number of required resources are assigned to each job i for any period t of the shortest possible processing duration. Constraints (4) bound the capacities of the resources used for each period the job can be processed. In (5) the number of allocated resources of type g at time t of job i are limited to $r_{i,m,g}$ in mode m, while constraints (6) guarantee that for the minimum duration of $p_{i,g,m}$ the full quantity of recources available in mode m are used by job i. The flow of passengers or bags is covered in equation (7). $I_{i,g,t}$ is defined as classical dynamic inventory variable where $g(z_{i,g,t})$ is the in- or outflow of period t resulting from resource allocation decision $z_{i,g,t}$. Finally, constraints (8) to (10) define the variables.

The presented formulation for resource allocation has several advantages. First, modes can be used to limit the combinations of resources allocated by a job. Thereby, we can allow to allocate resources only within spatial proximity, e.g. when allocating check-in counters or working stations in outbound baggage handling (see Section 2.4 and 2.5). Moreover, combinations can be given when there is a dependency between resources, e.g. an infeed station is connected to a specific baggage claim carousel for handling inbound baggage (see Section 2.7). Excluding the z variables from the model leads to an excessive amount of modes due to the combinatorics while excluding modes from the model, i.e. utilizing only the z-variables, requires further variables and constraints to ensure feasible combinations when allocating resources. Finally, mode variables can be used for branching in branch & bound based solution procedures where they can be seen as cuts in the solution space of the z variables.

Complexity analysis The GASP has $F \cdot (T+1) + G \cdot T \cdot (1 + F \cdot (M+2))$ constraints, $F \cdot T \cdot M$ binary variables, $F \cdot T$ integer variables and $F \cdot G \cdot T$ float variables. Let us consider a simplified GASP (SGASP) in which we only consider constraints (2), (4), (6), and the corresponding variables defined in (8) to (10).

Theorem 2.3.1. *The SGASP is strongly NP-hard*

Proof. We show that an instance of the single machine scheduling problem with release times and deadlines which is strongly NP-hard (see Garey and Johnson [43]), can be polynomially transformed into an instance of SGASP.

INSTANCE: Set $\mathcal{J}^{\mathrm{MS}}$ of jobs, for each job $j \in \mathcal{J}^{\mathrm{MS}}$ a processing time $p_j \in \mathbb{Z}_+$, a release time $r_j \in \mathbb{Z}_+$, and a deadline $e_j \in \mathbb{Z}_+$.

QUESTION: Is there a single machine schedule for $\mathcal{J}^{\mathrm{MS}}$ that satisfies the release time and deadline constraints?

Given an instance of the single machine scheduling problem, we equate each job $j \in \mathcal{J}$ with a job $j' \in \mathcal{J}^{\mathrm{MS}}$. In the GASP, we only have one resource g with $U_g = 1$ corresponding to the single machine and one mode m for all jobs such that $r_{j,m,g} = 1$ for all $j \in \mathcal{J}$. For job j, let $S^{\mathrm{es}}_j = r'_j$ be the period in which the job can start (release time), $S^{\mathrm{ls}}_j = e'_j - p'_j$ the last period in which the job is allowed to start (deadline), and $p_{j,m,g} = p'_j$ the processing time. S^{e}_j can be set to the end of the planning horizon.

Claim: There exists a feasible schedule for the jobs for the SGASP if and only if there is a feasible schedule of jobs for the single machine problem.

The processing of job j has to start between r'_j and $e'_j - p'_j$, claiming single resource g at full capacity for processing time p'_j. After the processing is finished, job j can deallocate resource g by setting the z variable to 0. □

Let us further consider a second simplified GASP (SGASP II) in which we only consider constraints (2) to (4), and the corresponding variables defined in (8) to (10).

Theorem 2.3.2. *The SGASP II is strongly NP-hard*

Proof. We show that the decision problem of the bin packing problem which is strongly NP-hard (see Garey and Johnson [43]), can be polynomially transformed into an instance of SGASP II.

INSTANCE: Set $\mathcal{J}^{\text{BP}}$ of items, each item $j' \in \mathcal{J}^{\text{BP}}$ having size $a'_j \in \mathbb{Z}_+$ and a set of bins $\mathcal{B}$, each bin having size $V \in \mathbb{Z}_+$.

QUESTION: Is there an assignment of items to bins such that the sum of all sizes of items assigned to a bin does not exceed the bins size?

Let us consider a planning horizon with just a single time period t. Given an instance of the bin packing problem, we equate each job $j \in \mathcal{J}$ with an item $j' \in \mathcal{J}^{\text{BP}}$ with $L_{j,t} = a'_j$. For each $b \in \mathcal{B}$ we define a resource g with $U_g = V$ and a mode m with $r_{j,m,g} = V$ such that $|\mathcal{B}| = |\mathcal{G}| = |\mathcal{M}|$.

Claim: There exists a feasible assignment of jobs to resources for the SGASP II if and only if there is a feasible assignment of items to bins without exceeding the bins size for the bin packing problem.

Each job j has to be assigned to exactly one resource consuming at least $z_{i,g,t} \geq a'_j$ units with the resource being limited to V units.

$\square$

Due to Theorem 2.3.1 and 2.3.2, we obtain

Theorem 2.3.3. *The GASP (3) - (10) is strongly NP-hard to solve.*

In the following, we will adopt the GASP to the three baggage handling processes at an airport. The transfer baggage handling problem (see Section 2.6) as well as the inbound baggage handling problem (see Section 2.7) are strongly NP-hard as SGAPS can easily be reduced to both. For the check-in counter planning problem (see Section 2.4) as well as the outbound baggage handling (see Section 2.5), strong NP-hardness has to be shown through a reduction of SGASP II as both problems do not have a minimum processing time $p_{i,g,m}$. We can therefore conclude that all baggage handling models are NP-hard to solve.

2.4 Check-in baggage flow

In the daily operational check-in counter planing the number of opened first, business and economy counters for a flight during the course of a day have to be determined. Thereby, the dynamic arrival of passengers has to be considered. In practice a counter opens between two and three hours before scheduled departure time of the flight. In a dynamic allocation, each counter assigned to a flight can be dynamically opened and closed. Alternatively, in a static allocation the service period is fixed and identical for all counters assigned for a flight. A dynamic allocation increases the flexibility of the check-in counter planning but leads to an increased complexity of the problem. Given the counter assignment for each flight, agents are staffed to the counters (see Bruno and Genovese [20] and Stolletz [86]).

The objective of check-in counter planning is to minimize the operating costs, such as renting costs for counters, while guaranteeing a defined level of service for passengers (see Bruno and Genovese [20], Chun [21, 22, 23], Chun and Mak [24], Van Dijk and Van der Sluis [94], Park and Ahn [74] and Parlar and Sharafali [75]). In literature costs are minimized by the minimization of the number of check-in counters to be opened in a given time interval. A efficient usage of counter leads to reduced renting costs and to reduced labor costs. Common measures for the level of service are the waiting time or the queue length of passengers at the counters (see Bruno and Genovese [20], Chun [21, 22, 23], Chun and Mak [24], Van Dijk and Van der Sluis [94], Park and Ahn [74] and Parlar and Sharafali [75]). A dynamic counter allocation is the preferred strategy against a static counter policy. Another service measurement in check-in counter planning is the walking distance from the counter to the gate (see Yan et al. [99, 100, 101]). Beside operating costs, Yan et al. [99, 100, 101] introduce the inconsistency measure, a penalty for the violation of a defined allocation principles, such as the counter adjacency for a flight. Another decision problem is to balance the workload among the check-in counters to obtain a fair work distribution among the agents. The work shifts of the desk agents should lead to a fair distribution of the workload among all employees (see Stolletz [87]).

For the assignment of flights to counters we distinguish between dedicated (single) check-in counters and common-use check-in counters. For the case of dedicated or single check-in counters each flight has its own counter(s) leased by the airline over a long term, while common-use counters are shared by a group of flights for short term (see Yan et al. [100]). The assignment of flights to common-use counters requires the partitioning of flights to counter blocks and, thus, is a generalization of the assignment of flights to dedicated check-in counters.

2.4.1 Literature review

Although check-in counter planning is a rather new problem (see Bruno and Genovese [20]), the literature comprises a wide range of different approaches from operations management and OR. Table 2.2 provides an overview of the different solution approaches for the check-in counter allocation. In what follows we distinguish between dedicated and common-use check-in counter planning.

Dedicated check-in Van Dijk and Van der Sluis [94] employ simulation to determine the minimal number of required counters for each flight (see Joustra and Dijk [57]) in order to meet a predefined level of service in terms of passenger waiting times at Schipool Airport. Given this lower bound for the number of counters for a flight, the airport check-in counter problem is formulated for a dynamic and static allocation. The objective minimizes the total number of check-in counters over time. Both problems are NP-complete (see Bruno and Genovese [20]) which leads to high computation times when solving real world instances. Therefore, Van Dijk and Van der Sluis [94] propose a LP-heuristic which decomposes the problem into smaller subproblems with natural separation such as domestic and international flights. Bruno and Genovese [20] show alternative MIP formulations for the static and dynamic allocation model of Van Dijk and Van der Sluis [94] which have similarities to the capacitated lot sizing problem (see Bitram and Yanasse [18] and Florian et al. [38]).

Beside mathematical programming, Chun [23] formulates the dynamic check-counter allocation problem as multi-dimensional placement problem and combines simulation with constraint propagation programming to estimate the number of counters for each flight such that a level of service in terms of passenger waiting times is guaranteed at Hong Kong Kai Tak International Airport (see also Chun [21, 22] and Chun and Mak [24]). Considering the arrival patterns of passengers, Park and Ahn [74] derive system of rules for the allocation of flights to counters for the Seoul Gimpo International Airport in Korea.

Parlar and Sharafali [75] present an approximated stochastic dynamic program algorithm (ADP) to determine the optimal number of counters for a dynamic allocation.

Common-use check-in Yan et al. [99, 100] present IPs for static and dynamic allocation

	Dedicated check-in		Common-use check-in	
	Static	Dynamic	Static	Dynamic
Optimization				
Mathematical Programming				
IP	[94]	[94]	[99]	[100, 101]
MIP	[20]	[20]		
Constraint Programming		[21–24]		[24]
Heuristic				
Constructive		[74]		
Improvement		[62]		[62]
Stochastic Programming				
Dynamic Programming		[75]		
Analytical				
Cost model	[54]			
Queuing model				
Deterministic	[51, 52, 72, 76, 89, 91]			
Stochastic	[63, 87]			
Simulation				
Discrete-event	[6, 25, 57, 88]			
Monte-Carlo	[92]			
Other				
Description	[35, 37, 53]			
Literature review	[91]			

Table 2.2: Literature on check-in counter allocation

of flights to common-use counters at the Chiang Kai-Shek International Airport in Taiwan. While Yan et al. [99, 100] decompose the MIP in smaller subproblems which are independently solved in one run, Yan et al. [100] use an iterative improvement procedure. Chun and Mak [24] propose a simulation to dynamically assign flights to common-use check-in counters.

A dynamic reallocation of flights to counters for the daily disruptions management is presented by Yan et al. [101]. The solution methodology is based on decomposition and relaxation of the MIP for which the subproblems are iteratively solved.

Most of the analytical models are applicable to dedicated and common-use check-in counter planning. Hence, we do not distinguish between analytical approaches for dedicated and common-use check-in counters planning. Given an allocation policy for flights, queuing models are used to measure the level of service in terms of passenger waiting times and queue lengths at check-in counters (see Janic [51, 52], Lee [63], Newell [72] and Stolletz [87]). Stochastic queuing models for check in counters assume a Poisson process for passenger arrivals and an uniform or an exponential distributed service times (see Lee [63] and Stolletz [87]). Stochastic queuing models are criticized in the literature as they assume steady state for the arrival rate of passengers. To incorporate the dynamic arrival rates of passengers at check-in counter, Stolletz [87] uses the stationary backlog carryover approach (see Stolletz [85]) which approximates the inhomogeneous arrival process of passengers by dividing the time horizon into small periods in which an constant arrival rate for passengers is assumed. He compares single and common-use counters and concludes that common-use counters lead to a reduction in passenger waiting times. Beside stochastic queuing models, deterministic queuing model based on cumulative flow diagrams to evaluate passengers' waiting time and their queue length at counters are presented by Newell [72], Janic [51, 52], Tanner [89], Park and Ahn [74] Piper [76], and Tošić [91]. A cumulative flow diagram graphically depicts the passenger arrival rate, passenger service rate, passengers waiting time and their queue length.

To avoid the lack of the steady state assumption and to obtain a greater flexibility, simulation is an alternative analysis method for check-in counters. There is one Monte-Carlos simulation of Tošić and Babic [92] and discrete-event simulations (see Appelt et al. [6], Chung and Sodeinde [25], Joustra and Dijk [57] and Takakuwa and Oyama [88]). Using discrete-event simulation Joustra and Dijk [57] compare waiting times of passengers for common-use and dedicated counters. In contrast to Stolletz [87], Joustra and Dijk [57] do not identify any advantage of the common-use check-in procedure in a simulation conducted for Schipool Airport.

A rather unique analytical study is conducted by Johnson [54]. He examines the cost of check-in counters with a multiple-service costing model from micro-economics which allocates fixed operating costs to each served flight at the counters.

2.4.2 Mathematical model

The proposed model formulation for check-in counter planning is based on insight gained at observations made in Munich Airport and build up on the model formulations of Bruno and Genovese [20] (see also Van Dijk and Van der Sluis [94]) and Yan et al. [99]. We assumes a given lower bound $L_{i,t}$ for the required number of check-in counters to satisfy passenger's demand of an outgoing flight (see Van Dijk and Van der Sluis [94]). Each check-in counter offers a given number of service lines, where passengers of flights receive their service. At each line only one flight can be handled at a time, however, at a check-in counter with more than one line, several flights can be assigned at a time. In order to model the common-use check-in counter planning with the GASP, the GASP's sets and parameters read as follows:

Sets:

$\mathcal{F}$ – outgoing flights;
$\mathcal{G}$ – check-in counters;
$\mathcal{M}$ – modes, where one mode m defines the number of service lines used at each counter;

Parameters:

$\left[S_i^{\mathrm{es}}, S_i^{\mathrm{ls}}\right]$ – time window for the start of the check-in of flight i;
S_i^{e} – last period for the check-in of flight i;
$L_{i,t}$ – minimum number of required service lines for flight i at time t;
U_g – maximum number of allocable service lines at counter g;
$\lambda_{i,t}^{+}$ – arriving passengers of flight i at time t;
λ^{-} – number of passengers that can be served at a service line per period;
$r_{i,m,g}$ – number of service lines used by flight i in mode m at counter g.

Variables:

$x_{i,m,t}$ – 1, if the handling of flight i is done in mode m and starts in period t,
0 otherwise;
$z_{i,g,t}$ – number of service lines of counter g that are used by flight i at time t;
$I_{i,t}$ – number of waiting passengers of flight i at time t.

The start time of each flight is within time window $\left[S_i^{\mathrm{es}}, S_i^{\mathrm{ls}}\right]$; the earliest start time S_i^{es} and the latest start time S_i^{ls} is usually three and two hours before flight i's scheduled departure time, respectively. The end of the check-in S_i^{e} depends on the flight type (e.g. transcontinental or domestic flight) and of the airport size and can vary between 45 to 15 minutes before flights scheduled departure time. Each passenger waiting in line is penalized with cost c_i per period while the cost for operating a service line per period is given by c. The the CCPP is given by

$$\text{minimize} \sum_{i\in\mathcal{F}}\sum_{g\in\mathcal{G}}\sum_{t\in\mathcal{T}} (c_i \cdot I_{i,g,t} + c \cdot z_{i,g,t}) \tag{11}$$

subject to

$$\sum_{m\in\mathcal{M}_i}\sum_{\tau=S_i^{\mathrm{es}}}^{S_i^{\mathrm{ls}}} x_{i,m,\tau} = 1 \qquad \forall\, i \in \mathcal{F} \tag{12}$$

$$L_{i,t} \leq \sum_{g\in\mathcal{G}} z_{i,g,t} \qquad \forall\, i \in \mathcal{F}, t \in \left[S_i^{\mathrm{ls}}, S_i^{\mathrm{e}}\right] \tag{13}$$

$$\sum_{i\in\mathcal{F}:t\in\left[S_i^{\mathrm{es}},S_i^{\mathrm{e}}\right]} z_{i,g,t} \leq U_g \qquad \forall\, g \in \mathcal{G}, t \in \mathcal{T} \tag{14}$$

$$\sum_{m\in\mathcal{M}_i} r_{i,m,g} \cdot \sum_{z=S_i^{\mathrm{es}}}^{\min\left\{t,S_i^{\mathrm{ls}}\right\}} x_{i,m,z} \geq z_{i,g,t} \qquad \forall\, i \in \mathcal{F}, g \in \mathcal{G}, t \in [S_i^{\mathrm{es}}, S_i^{\mathrm{e}}] \tag{15}$$

$$I_{i,g,t} = \left(I_{i,g,t-1} + \lambda_{i,t}^{+} - \lambda^{-} \cdot \sum_{g\in\mathcal{G}} z_{i,g,t}\right)^{+} \qquad \forall\, i \in \mathcal{F}, g \in \mathcal{G}, t \in [S_i^{\mathrm{es}}, S_i^{\mathrm{e}}] \tag{16}$$

$$I_{i,g,S_i^{\mathrm{e}}} = 0 \qquad \forall\, i \in \mathcal{F}, g \in \mathcal{G} \tag{17}$$

and (8) to (10)

Objective function (11) minimizes the penalties for the number of passengers for flight i waiting in a queue as well as the costs for operating the service lines. Constraints (13) ensures that the minimal number of required service lines for flight i at time t are allocated. Due to constraints (12) the service of each each flight i is started exactly once in a specific mode m in the time window $\left[S_i^{\mathrm{es}}, S_i^{\mathrm{ls}}\right]$. The maximal number of assigned service lines at each check-in counter g and at each time t is bounded by constraints (14). In constraints (15) the actual number of open lines for flight i is also set. The number of flight i's waiting passengers $I_{i,t}$ at time t is calculated in constraints (16). Due to constraints (17), all passengers of a flight have to be served up to and including the flights last check-in period.

Complexity analysis Due to Theorem 2.3.2 the problem is NP-hard to solve in the strong sense. The number of constraints is $F \cdot (T + 1 + G \cdot (2 \cdot T + 1)) + G \cdot T$.

2.4.3 Future challenges

At most airports the check-in counter allocation and the counter staffing are done manually by experienced planners in a two step approach (see Chun [23] and Yan et al. [99]). In a first step, flights are allocated to counters and in a second step the agents are staffed to open counters. The two step approach where the capacities of the personnel is relaxed in the first step leads to heuristic solutions of the problem. Therefore, an integrated model for the check-in counter allocation problem and for agent staffing should be addressed. Model formulation (11)-(16) can be easily connected with a shift-planing problem as follows: Let $s_{i,g,t}$ be the shift variable for check-in counter staffing which is equal to the number of agents assigned to flight $i \in \mathcal{F}$ at counter g at time t. To connect the check-in counter planning problem with shift-scheduling problem we require the additional constraints

$$z_{i,g,t} - s_{i,g,t} = 0 \qquad \forall\, i \in \mathcal{F}, g \in \mathcal{G}, t \in \mathcal{T} \tag{18}$$

which enforce the number of agents assigned to flight i at counter g at time t to be equal to the number of flight i's open lines for counter g at time t. Constraints (18) are linking constraints between the CCPP and the shift-scheduling problem for agents.

2.5 Outbound baggage flow

By the outbound baggage process baggage is loaded into airplanes' cargo holds. The baggage stream originates from check-in and transfer baggage. Figures 2.5 and 2.6 show the flow of outbound baggage.

For outbound baggage handling the airport operators have to make an assignment and a scheduling decision. In a first step, each departing flight is assigned to (at least) one handling facility and the start of the baggage handling is set. Bags in the storage system can only be depleted, once the flight's baggage handling has been started. At some airports, the airport operator also decide about the storage place, that is the storage lane, for early baggage (see Figure 2.2). Stored baggage in the same storage lane should be depleted simultaneously to reduce the re-organization time of the baggage within the storage and to reduce the mechanical wear of the network devices. The amount of stored bags at a time is restricted by the storage capacity. In a second step, work groups have to be assigned to handling facilities to load the bags into containers.

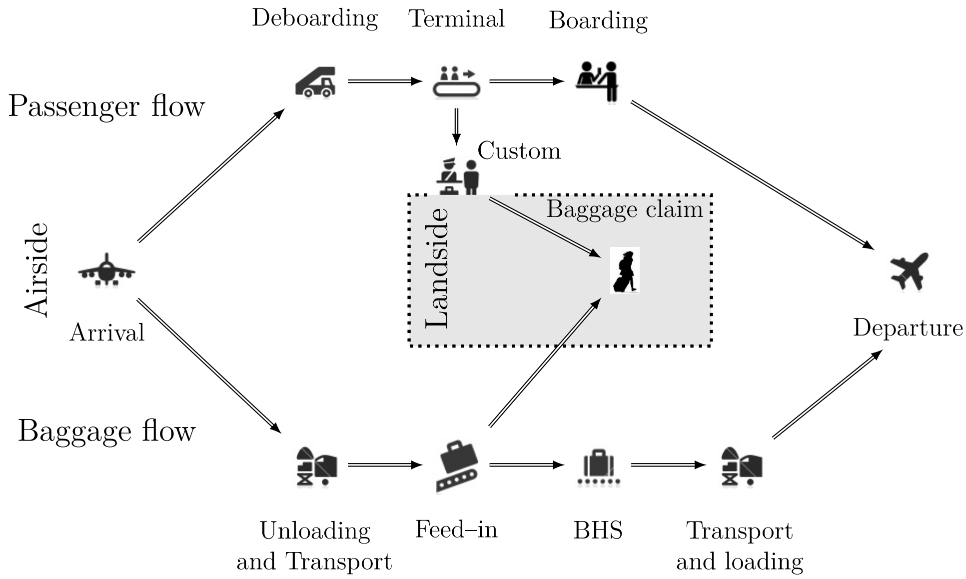


Figure 2.5: Passenger and baggage flow for transfer and inbound processes

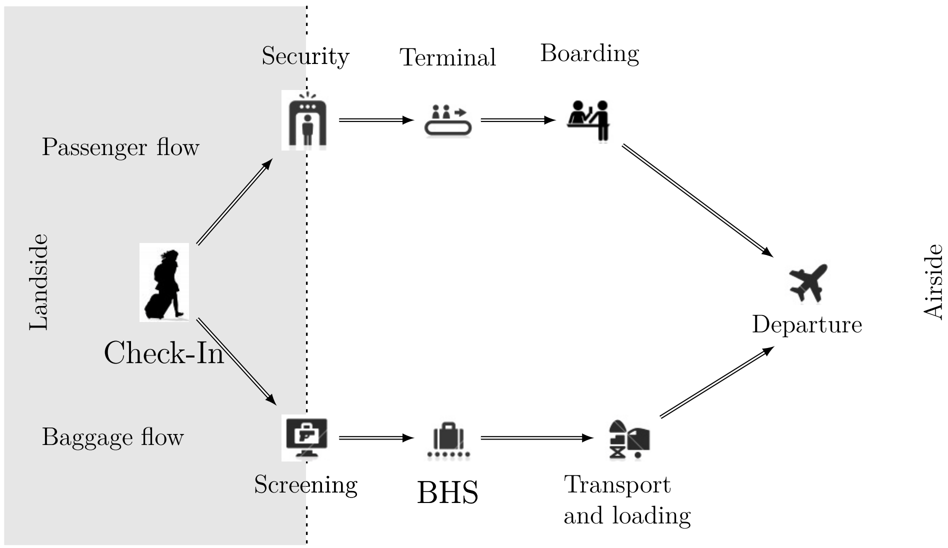


Figure 2.6: Passenger and baggage flow landside to airside

The assignment of flights to handling facilities and the scheduled start time of the baggage handling should avoid workload peaks at the handling facilities. The workload is the number of bags on a conveyor belt at a time. A workload at a handling facility increases when the bag inflow from the sorting network onto the conveyor belt of the handling facility is higher than the loading rate at the working stations. Once the workload reaches conveyor belt's capacity of the handling facility, no more bags within the sorting network of the BHS will be sent to the handling facility. Instead, the bags remain in the sorting network until the workload of the handling facility is decreased. An increased number of bags in the sorting network, in turn, increases the danger of traffic jams and therefore may lead to further delays in sorting baggage through the BHS. In a worst case scenario, too many bags in the BHS may lead to a major break down of the whole system (see Frey and Kolisch [41], Frey et al. [42]). One objective for outbound baggage handling is to obtain a balanced workload across the baggage carousels. Further objectives are a short distance between the handling facility and the parking position of the departing airplane as well as the assignment of handling facility which are preferred by groundhandlers (see Ascó et al. [7]). Each handled flight requires its own work group. According to Abdelghany and Abdelghany [1], Abdelghany et al. [2] and Ascó et al. [7] operating cost (staffing costs) can be reduced if the number of simultaneously handled flights is minimized.

As constraints, the number of simultaneously handled flights is restricted by the total number of available working stations at each handling facility. Moreover, the capacity of the storage system has to be satisfied.

2.5.1 Literature review

Table 2.3 classifies the literature according to solution methods for baggage handling in a decentralized and centralized baggage handling environment. Ascó et al. [7] present a MIP formulation for a decentralized environment with chutes which can handle one flight at a time but one flight can be handled at several adjacent handling facilities. They consider several objectives such as a balanced workload across the chutes, a minimal distance from the handling facility to the flight's parking position and the maximization of the preferences for flights assigned to handling facility. A storage system is not taken into account and the start times for the baggage handling are assumed to be given. Ascó et al. [7] and Ascó et al. [8] present different presorting strategies for departing flights and handling facilities with a greedy allocation heuristic (see also Abdelghany et al. [2] and Abdelghany and Abdelghany [1]).

A MIP formulation for the assignment of flights to centralized baggage carousels and the scheduling of the start time of the baggage handling is presented in Frey and Kolisch [41]. In the proposed model several flights can be handled at a handling facility while processing one flight at more than one handling facilities is not allowed. Frey and Kolisch [41] consider a storage system and define the start time of the baggage handling and the start time of the storage depletion as decision variables. They minimize the workload peak across the baggage carousels. Frey et al. [42] decompose the problem into several subproblems to reduce the complexity. Numerical results of the decomposition heuristic are not reported. Barth and Pisinger [16] present a model formulation for a decentralized and centralized baggage handling environment at Frankfurt Airport. Lenior [65] discusses human interactions with semi- or fully automated systems for baggage sorting.

Optimization	
Mathematical Programming	
IP	[7, 16, 41, 42]
Heuristic	
Constructive	[1, 2, 7, 8]
Other	
Description	[65]

Table 2.3: Literature on baggage handling

2.5.2 Mathematical model

Similar to the work presented in the literature review we assume that we have to assign one flight to exactly one baggage carousel. At each baggage carousel we can assign more than one flight at a time. However, the number of assigned flights is restricted by the number of available working stations where worker can load bags in the bulk-containers. In contrast to Abdelghany et al. [2] and Ascó et al. [7], a storage system for early bags is considered. We assume, that the depletion of the storage starts as soon as the baggage handling begins. The sets and parameters of the GASP for outbound baggage handling read as follows:

Sets:

$\mathcal{F}$ – outgoing flights with outbound baggage;

$\mathcal{G}$ – baggage carousels;

$\mathcal{M}$ – modes where one mode m corresponds to a specific carousel and to a number of working stations of that carousel used for a flight;

Parameters:

$\left[S_i^{\mathrm{es}}, S_i^{\mathrm{ls}}\right]$	–	time window for start of the baggage handling for flight i;
S_i^{e}	–	end of the baggage handling for flight i;
U_g	–	number of working stations available at baggage carousel g;
U^{s}	–	capacity of the central storage system;
$\lambda_{i,t}^{+}$	–	number of newly incoming bags for flight i at time t;
λ^{-}	–	number of bags that can be handled at a working station per period;
$r_{i,m,g}$	–	number of working stations assigned to flight i in mode m at baggage carousel g.

Variables:

$x_{i,m,t}$	–	1, if the handling of flight i is done in mode m and starts in period t, 0 otherwise;
$z_{i,g,t}$	–	number of working stations of carousel g that are used by flight i at time t;
$I_{i,g,t}$	–	workload caused by flight i on baggage carousel g at time t.

Similar to the check-in counter planning, the earliest start time S_i^{es} and the latest start time S_i^{ls} for the baggage handling of each flight $i \in \mathcal{F}$ is three to two hour before the flight's scheduled departure time, respectively, whereas the end of the baggage handling, S_i^{e}, is about 10 minutes before the scheduled departure time. In addition, we need variable $B_{i,t}$ which is equal to the number of bags of flight i that are in the storage at time t. The capacity of the storage system is limited by U^s. When starting the baggage handling bags are sent from the storage to the assigned baggage carousel with a rate of β bags per period. At the end of the baggage handling the containers with the baggage are towed to the departing airplane and the resources at the baggage carousels are set free again. The objective of the presented outbound baggage handling is to obtain a balanced workload across all carousels. The model formulation for outbound baggage handling reads as follows

$$\text{minimize} \sum_{g\in\mathcal{G}} \sum_{t\in\mathcal{T}} \left(\sum_{i\in\mathcal{I}} I_{i,g,t}\right)^2 \tag{19}$$

subject to

$$\sum_{m\in\mathcal{M}_i}\sum_{\tau=S_i^{\text{es}}}^{S_i^{\text{ls}}} x_{i,m,\tau} = 1 \qquad \forall\, i\in\mathcal{F} \tag{20}$$

$$L_{i,t} \leq \sum_{g\in\mathcal{G}} z_{i,g,t} \qquad \forall\, i\in\mathcal{F}, t\in t\in\left[S_i^{\text{ls}}, S_i^{\text{e}}\right] \tag{21}$$

$$\sum_{i\in\mathcal{F}:t\in\left[S_i^{\text{es}},S_i^{\text{e}}\right]} z_{i,g,t} \leq U_g \qquad \forall\, g\in\mathcal{G}, t\in\mathcal{T} \tag{22}$$

$$\sum_{m\in\mathcal{M}_i} r_{i,m,g}\cdot\sum_{\tau=S_i^{\text{es}}}^{\min\left\{t,S_i^{\text{ls}}\right\}} x_{i,m,\tau} \geq z_{i,g,t} \qquad \begin{array}{l}\forall\, i\in\mathcal{F}, g\in\mathcal{G},\\ t\in[S_i^{\text{es}}, S_i^{\text{e}}]\end{array} \tag{23}$$

$$\begin{aligned} I_{i,g,t} = \Big(I_{i,g,t-1} + \Big(\lambda_{i,t}^{+} + \min\{\beta, B_{i,t}\}\Big)\cdot & \qquad \forall\, i\in\mathcal{F}, g\in\mathcal{G},\\ \left.\sum_{\tau=S_i^{\text{es}}}^{\min\left\{t,S_i^{\text{ls}}\right\}}\sum_{m\in\mathcal{M}_i} x_{i,m,\tau} - \lambda^{-}\cdot z_{i,g,t}\right)^{+} & \qquad t\in[S_i^{\text{es}}, S_i^{\text{e}}] \end{aligned} \tag{24}$$

$$\begin{aligned} B_{i,t} = B_{i,t-1} + \lambda_{i,t}^{+}\cdot\left(1-\sum_{m\in\mathcal{M}_i}\sum_{\tau=S_i^{\text{es}}}^{t} x_{i,m,\tau}\right) - & \qquad \forall\, i\in\mathcal{F}, t\leq S_i^{\text{e}}\\ \sum_{m\in\mathcal{M}_i}\sum_{\tau=S_i^{\text{es}}}^{t}\min\{\beta, B_{i,t-1}\}\cdot x_{i,m,\tau} + & \\ \lambda_{i,t}^{+}\cdot\mathbb{1}_{\left[0,S_i^{\text{es}}\right[}(t) & \end{aligned} \tag{25}$$

$$\sum_{i\in\mathcal{F}:t\leq S_i^{\text{e}}} B_{i,t} \leq U^{\text{s}} \qquad \forall\, t\in\mathcal{T} \tag{26}$$

$$B_{i,t} \geq 0 \qquad \forall\, i\in\mathcal{F}, t\in\mathcal{T} \tag{27}$$

and (8) to (10)

As all arriving bags have to be loaded within the handling period, constraints (21) bound the minimal number of working stations in each period from below. In constraints (20) each flight $i \in \mathcal{F}$ is assigned to exactly one mode, corresponding to a number of working stations at a specific carousel. The number of flights which can be assigned to one baggage carousel g is restricted by the number of available working stations U_g in constraints (22). Constraints (23) determine the number of open working stations for flight i at time t. The workload of a carousel is calculated in constraints (24). Constraints (25) calculate flight i's amount of baggage in the central storage system. The total amount of stored bags at time t is restricted in constraints (26). The workload over all baggage carousels is balanced due to quadratic objective function (19). The above model is not-linear. However, it can be linearized by modeling techniques as described in Williams [97].

Complexity analysis Due to Theorem 2.3.2 the problem is NP-hard to solve in the strong sense. The number of constraints is $F \cdot (1 + T \cdot (2 \cdot G + 1)) + T \cdot (G + 1)$.

2.5.3 Future challenges

Beside the capacity constraints of the handling facilities and the storage, the assignment of flights to handling facilities has to satisfy the resource constraints of the groundhandlers. So far, outbound baggage handling is planned separately from the staffing of the groundhandlers which might lead to a shortage of workers at handling facilities. Abdelghany et al. [2] try to avoid staff shortage by minimizing the number of simultaneously executed flights. However, an integrated model for outbound baggage handling and groundhandler staffing would allow to improve the overall solution.

The dynamic planning situation where the system state changes from period to period makes it necessary to update the assignment to handling facilities and the flights' baggage handling schedule during the course of the day. Disruption such as flight position changes, breakdown of carousels or working stations require the re-assignment or re-scheduling of flights's baggage handling. The re-planning has to consider all given resource constraints. So far, the literature has just focused on the generation of daily plan and on the re-planning of work groups to departing flights.

2.6 Transfer baggage flow

Transfer baggage is brought in by incoming flights and forwarded to outgoing flights. Transfer baggage processes thus only arise at hub airports. In 2011, mishandling bags led to a total cost of 2.58 billion US Dollars to the industry (see SITA [83]). According to SITA [83], the main contributor, by far, to the mishandling problem is transfer baggage. 53% of all missing bags are caused by transfer baggage mishandling with numerous hub airports even suffering from considerably higher percentages. Furthermore, the fraction of mishandled transfer bags on all misshandled baggage is growing. Despite the high practical relevance, there are only few papers which address the transfer baggage handling and the corresponding decision problems. Heinz and Pitfield [48] analyze the impact of

British airways' move to Terminal 5 at London Heathrow airport on the transfer baggage performance.

The regular process of forwarding transfer baggage at a typical hub airport is as follows: The unloading process of an incoming airplane begins after its arrival at the final parking position, the on-block position. Usually, baggage is already sorted within the cargo hold of the plane such that urgent transfer baggage with shorter connecting time is unloaded first. The transfer baggage is loaded on a baggage towing vehicle and the vehicle drives to an infeed station where the bags are fed into the BHS.

A further transfer baggage handling process is often called "short connection service". In case that bags can not be regularly transported via the BHS to the outgoing flight's handling facility within the corresponding time window, many airports have a special handling facility to process last minute bags, the so–called "last minute chutes" (see Section 2.2.2). The bags are transported to the "last minute chutes" using a BHS. There, bags from one or more flights leaving in a nearby area are loaded onto a vehicle and transported to the outgoing flights. Besides dedicated "last minute chutes", some airports temporarily use regular handling facilities for short connection services.

The regular process of forwarding transfer baggage and the "short connection service" using "last minute chutes" reach its limit when it comes to very short connecting times. In this case, "direct transfer services" or "ramp direct services" (RDS) have to be undertaken. Dedicated vans are loaded with RDS bags at the incoming airplane and transport the bags directly to the outgoing flight bypassing the BHS. Depending on the incoming flight, it can be necessary to first bring a bag to a custom control station or a baggage screening station.

2.6.1 Mathematical model

For a mathematical model of the regular transfer baggage handling we use the GASP in the following way:

Sets:

$\mathcal{F}$ – incoming flights with transfer baggage;
$\mathcal{G}$ – infeed stations;
$\mathcal{M}$ – modes where one mode m corresponds to a unique infeed station.

Parameters:

$\left[S_i^{\text{es}}, S_i^{\text{ls}}\right]$	–	time window for the infeed of inbound baggage of flight i;
U_g	–	maximal number of flights which can be simultaneously handled by infeed station g;
$r_{m,g}$	–	1, if mode m corresponds to infeed station g, 0 otherwise;
$p_{i,m,g}$	–	processing time of flight i at infeed station g in mode m;
λ_g^+	–	infeed rate at infeed station g.

Variables:

$x_{i,m,t}$	–	1, if the handling of flight i is done in mode m and starts in period t, 0 otherwise;
$z_{i,g,t}$	–	1, if resource g is used by flight i at time t, 0 otherwise;
$I_{i,g,t}$	–	cumulative number of bags of flight i handled at infeed station g up to and including time t.

The earliest start S_i^{es} of the handling process at an infeed station g is predetermined by the arrival time of the flight and the time required for unloading and transferring the baggage to the infeed station. Due to their layout, most infeed stations can only be used to infeed one flight at a time so that $U_g = 1$. The minimum processing time $p_{i,m,g}$ depends on the number of bags of flight i and the infeed rate at station g. The goal is to handle the bags in a way they can reach the baggage carousel of the destinating flight on time. Therefore, parameter $w_{i,g,t}$ defines the number of bags that have to be fed in at infeed station g until time t, assuming that bags are already pre-ordered in a way that urgent bags are handled first. The objective function penalizes the number of bags that are not handled on time. The transfer baggage handling problem is formulated as follows:

$$\text{minimize} \sum_{i\in\mathcal{I}} \sum_{g\in\mathcal{G}} \sum_{\tau=S_i^{\text{es}}}^{S_i^{\text{e}}} \left(w_{i,g,t} - \max\left\{w_{i,g,\tau-1}, I_{i,g,\tau}\right\}\right)^+ \tag{28}$$

subject to

$$\sum_{m\in\mathcal{M}_i} \sum_{\tau=S_i^{\text{es}}}^{S_i^{\text{ls}}} x_{i,m,\tau} = 1 \qquad \forall\, i \in \mathcal{F} \tag{29}$$

$$\sum_{i \in \mathcal{F}: t \in \left[S_i^{\text{es}}, S_i^{\text{e}}\right]} z_{i,g,t} \leq U_g \qquad \forall\ g \in \mathcal{G}, t \in \mathcal{T} \tag{30}$$

$$r_{m,g} \cdot \sum_{\tau=\max\left\{t-p_{i,m,g}+1, S_i^{\text{es}}\right\}}^{\min\left\{t, S_i^{\text{ls}}\right\}} x_{i,m,\tau} = z_{i,m,g} \qquad \forall\ i \in \mathcal{F}, m \in \mathcal{M}_i, g \in \mathcal{G}, \quad t \in [S_i^{\text{es}}, S_i^{\text{e}}] \tag{31}$$

$$I_{i,g,t} = I_{i,g,t-1} + \lambda_g^+ \cdot z_{i,g,t} \qquad \forall\ i \in \mathcal{F}, g \in \mathcal{G}, \quad t \in [S_i^{\text{es}}, S_i^{\text{e}}] \tag{32}$$

and (8) to (10)

Due to constraints (29) a flight is assigned to exactly one infeed station, (30) limits the number of flight that can be handled simultaneously at a station. Constraints (31) assure that the processing of a flight at an infeed station is not interrupted for a minimum duration of $p_{i,m,g}$. The number of bags of flight i that are handled until time t at infeed station g is calculated in (32). The number of handled bags is used in the objective funcion (32) in order to penalize the amount of late baggage.

Complexity analysis Due to Theorem 2.3.1 the problem is NP-hard to solve in the strong sense. The number of constraints is $F \cdot (1 + T \cdot G \cdot (M + 2)) + G \cdot T$.

2.6.2 Future challenges

Besides the work of Clausen and Pisinger [26], there is no paper on the transfer baggage handling decisions. As Clausen and Pisinger [26] focus on short connection and direct transfer service, the literature concerning the regular transfer baggage process via the BHS is virtually void. At the same time, hub airports like Frankfurt airport and Munich airport are investing in OR solutions to improve the transfer baggage handling process. Hence there is need for scientific treatment of this problem. In a further step, it is a challenging task to integrate the planning of outbound and transfer baggage processes.

2.7 Inbound baggage flow

The inbound baggage process merges the separated flow of the passengers and of the baggage at the baggage claim carousels (see Figure 2.5). Passengers disembark from the plane and walk to the assigned baggage claim carousel where they pick up their bags. Depending on the flight origin their way through the terminal to the baggage claim carousels is predefined by the airport operators. For example, passenger from transcontinental flights have to be passed trough customs, while passengers from regional flights access the baggage claim hall directly. But also the gate of the flight (e.g. northern or souther part of the terminal) defines the way to the baggage claim carousels. As soon as the passenger arrives at the baggage claim carousel and identifies his bag on the conveyor belt he picks it up and finally leaves the baggage claim hall when all his bags are claimed.

In the operational inbound baggage handling we are facing a “double sided” assignment problem one at the airside (infeed stations) and the other at the landside (baggage claim carousel). The assignment of the infeed station has influence on the possible assignment of the baggage claim carousel and vice versa (see Section 2.2).

The objective of the operational decision problem can vary between airports and depends on local situations (see Ashford et al. [9]). Since the perceived satisfaction of passengers is an important factor for airlines and airport (see for example Correia and Wirasinghe [27], Martel and Senviratne [69] and de Neufville and Odoni [30]) one objective is to obtain a specific level of service for passengers in terms of waiting times at the baggage claim carousel or the distance from the gate to the carousel (see Barbo [15]). The service quality is also influenced by the utilization of the baggage claim carousel. Hence, another objective is a leveled utilization across all baggage claim carousels leading to reduced passenger congestion at the device frontage (see e.g. Correia and Wirasinghe [27] and Ghobrial et al. [44]).

As inbound and transfer baggage share the same resources such as equipment and personnel it is desired by the groundhandler that the infeed station for inbound baggage is located closely to the infeed station for transfer baggage as transfer baggage is often prioritized over inbound baggage. The joint usage of infeed stations for inbound and transfer baggage is typically not possible. One infeed station can be used to feed in the baggage within one container at a time. Therefore, containers assigned to the same infeed station have to wait in a queue and are processed in a first-come, first-served manner.

At a baggage claim carousel several flights can be served at a time. The number of simultaneously served flights is limited by the display at the baggage claim carousel and the number of maximal number of people who can stand around the baggage claim carousel.

The splitting of a flight to several baggage claim carousel is not common in practice even if carousels are located next to each other. The reasons is the reduced perceived service quality of passengers (see Martel and Senviratne [69] and Robusté [79]). Robusté [79] criticizes the splitting of a flight to several baggage claim carousels as this leads to a reduced service quality for passengers. However, some airports such as Frankfurt Airport use different baggage claim carousels for first, business and economy passengers.

2.7.1 Literature review

Literature for inbound baggage handling encompasses analytical models only(see Table 2.4). Barbo [15], Robusté [79] and Tošić [91] describe the baggage claim process as part of airport passenger terminal operations and elaborate deterministic queuing approaches based on cumulative diagrams for analyzing the claim process (see also Horonjeff [50] and Newell [72]). They show that the arrival process of passengers at baggage claim carousels depends on several factors such as walking distance, type of flight and terminal design. Ghobrial et al. [44] test baggage claim carousel types with different frontage shapes and lengths. Browne et al. [19] and Gosling [46] derive formulas for computing the maximal expected queue length of bags and passengers under the assumption of constant arrival rates for passenger.

Analytical	
Queuing model	
Deterministic	[15, 19, 44, 46, 50, 72, 79, 91]
Survey	[27, 102]
Other	
Description	[30, 69]
Literature review	[91]

Table 2.4: Literature on inbound baggage handling

A survey evaluating passengers' satisfaction is presented by Correia and Wirasinghe [27] and Yen et al. [102]. Yen et al. [102] show that passengers overestimate their waiting times at baggage claim carousel. Correia and Wirasinghe [27] confirm the importance of minimal passenger waiting times in a survey conducted at Calgary International Airport (see also Martel and Senviratne [69]).

2.7.2 Mathematical model

A mathematical model for inbound baggage handling has to consider both sides of the handling process; the airside with its infeed stations and the landside with the baggage claim carousels. To model both sides, we consider each possible combination of infeed station and baggage claim carousel as mode.

Sets:

$\mathcal{F}$	–	incoming flights with inbound baggage;
$\mathcal{G}$	–	baggage claim carousels and infeed stations;
$\mathcal{M}$	–	modes where one mode m corresponds to a combination of infeed station and baggage claim carousel;

Parameters:

$\left[S_i^{\text{es}}, S_i^{\text{ls}}\right]$	–	time window for the start of the infeed of inbound baggage of flight i;
S_i^{e}	–	upper bound for the claiming end of flight i;
U_g	–	maximal number of flights which can be simultaneously handled at baggage claim carousel or infeed station g;
$r_{i,m,g}$	–	1, if baggage claim carousel/infeed station g is used in mode m for flight i, 0 otherwise;
$p_{i,m,g}$	–	infeed duration for flight i in mode m if g corresponds to an infeed station;
	–	claiming duration for flight i in mode m if g corresponds to a baggage claim carousel;
λ^+	–	infeed rate;
$\lambda^-_{i,g,t}$	–	rate at which baggage is picked up by passengers at time t if flight i is assigned to baggage claim carousel g.

Variables:

$x_{i,m,t}$	–	1, if the handling of flight i is done in mode m and starts in period t, 0 otherwise;
$z_{i,g,t}$	–	1, if resource g is used by flight i at time t, 0 otherwise;
$I_{i,g,t}$	–	number of bags of flight i on carousel g not picked up at time t;

The start time of the infeed process depends on the arrival time of flight $i \in \mathcal{F}$. Adding an offset to flight i's actual arrival time for unloading and transferring the container to an infeed station, we obtain the earliest start time S_i^{es} for flight i's infeed. To guarantee that the flight is fed in within a given time period, we define the latest infeed start time S_i^{ls} which should be not more than 20 minutes after S_i^{es}. Passengers' baggage pick up rate $\lambda_{i,g,t}^{-}$ of flight i is time dependent and depends on the passenger's walking distance from flight i's parking position to the assigned baggage claim carousel g. As long as no passenger has arrived at the assigned baggage claim carousel, we have $\lambda_{i,g,t}^{-} = 0$. Given a combination of infeed station and baggage claim carousel and passengers pick up rate, we derive the required claim duration $p_{i,m,g}$ at mode's m corresponding baggage carousel g. S_i^{e} denotes a maximal upper bound for the end time of flight i's claim process, i.e. $S_i^{\text{e}} = \max_{m \in \mathcal{M}, g \in \mathcal{G}} \left\{ S_i^{\text{ls}} + p_{i,m,g} \right\}$.

We have $U_g = 1$, if g represents an infeed station. Whereas U_g is equal the number of flights that can be handled simultaneously, if g corresponds to a baggage claim carousel.

As objective we balance the workload over all baggage claim carousels. The infeed process of each flight must not bet artificially delayed. Thus, a delay of an infeed is penalized by a value ϵ_t, with $\epsilon_t < \epsilon_{t+1}$ for all $t \in \mathcal{T}$. Thus, the basic model formulation for the inbound baggage handling is stated as

$$\text{minimize} \sum_{g \in \mathcal{G}} \sum_{t \in \mathcal{T}} \left(\sum_{i \in \mathcal{F}} I_{i,g,t} \right)^2 + \sum_{i \in \mathcal{F}} \sum_{S_i^{\text{es}} \leq \tau \leq S_i^{\text{ls}}} \epsilon_\tau \cdot x_{i,m,\tau} \tag{33}$$

subject to

$$\sum_{m \in \mathcal{M}_i} \sum_{\tau = S_i^{\text{es}}}^{S_i^{\text{ls}}} x_{i,m,\tau} = 1 \qquad \forall\, i \in \mathcal{F} \tag{34}$$

$$\sum_{i \in \mathcal{F} : t \in \left[S_i^{\text{es}}, S_i^{\text{e}} \right]} z_{i,g,t} \leq U_g \qquad \forall\, g \in \mathcal{G}, t \in \mathcal{T} \tag{35}$$

$$r_{i,m,g} \cdot \sum_{\tau=\max\left\{t-p_{i,m,g}+1, S_i^{\text{es}}\right\}}^{\min\left\{t, S_i^{\text{ls}}\right\}} x_{i,m,\tau} = z_{i,g,t} \qquad \forall\, i \in \mathcal{F}, g \in \mathcal{G}, m \in \mathcal{M}_i, t \in [S_i^{\text{es}}, S_i^{\text{e}}] \tag{36}$$

$$I_{i,g,t} = \left(I_{i,g,t-1} + \lambda^{+} \cdot z_{i,g,t} - \lambda^{-}_{i,g,t}\right)^{+} \qquad \forall\, i \in \mathcal{F}, g \in \mathcal{G}, t \in [S_i^{\text{es}}, S_i^{\text{e}}] \tag{37}$$

and (8) to (10)

Objective function (33) balances the workload across all baggage claim carousels and penalizes an artificial delay of a flight's infeed. The above model is not-linear. However, it can be linearized by modeling techniques as described in Williams [97]. Constraints (34) assign each flight $i \in \mathcal{F}$ to one mode m where the start time for the infeed has to be within flight i's time window. The number of flights, which can be simultaneously served at one baggage claim carousel or infeed station at a time is restricted by constraints (35). Constraints (36) ensures that flight i allocates infeed station g or baggage claim carousel g as long as flight i's bags are fed in or claimed, respectively. The amount of baggage of flight i on baggage claim carousel g at time t is calculated in constraints (37).

Complexity analysis Due to Theorem 2.3.1 the problem is NP-hard to solve in the strong sense. The number of constraints is $F \cdot (1 + G \cdot T \cdot (M + 2)) + G \cdot T$.

2.7.3 Future challenges

As stated above there are a lot of analytical models such as queuing and simulation models as well as surveys. However, no work has been undertaken on the optimization of inbound baggage handling such as the assignment of flights to infeed stations and baggage claim carousel. The presented model formulation (33) - (37) is the first of this kind. The objective could be further extended by considering also the level of service quality for passengers as similar to check-in counter planning where the waiting time of passengers is penalized (see Section 2.4). Airports such as Terminal 2 of Munich airport and Frankfurt airport have realized the importance of inbound baggage handling and work on effective and efficient planning approaches.

2.8 Conclusion

With increasing flight traffic, airports seek to use their given resources as efficient as possible in order to safe money and time. Quite some work has already been done on optimizing baggage handling at airports. In particular during the last decades the topic baggage became more and more the focus of the OR community which underlies its importance. In this chapter, we structured the baggage handling process and classified the research. We discussed constraints and objectives of the three subprocesses that are common for many airports. The introduced GASP mathematically demonstrates similarities and differences of the subprocesses. The model formulations clearly define the assignment and scheduling decisions to take for each subprocess and shows the corresponding effects on the baggage flow and, hence, the objective function. In so doing, we derived a more general model formulation (11)- (10) for the CCPP which combines two check-in counter planning strategies found in literature. Hence, the chapter with the mathematical classification serves as a basis for future research, bridging the gap between new practical problems at airports and OR approaches.

Future challenges will be not only to improve the single processes of the baggage handling through improved models and solution methods, but also to integrate all aspects of the handling process and give the different stakeholders a common process view. Ideally, this should lead to improved customer satisfaction, more robust baggage handling, while postponing investments in expensive infrastructure.

Chapter 3

Outbound baggage handling

"Continuous improvement is better than delayed perfection."

– Mark Twain, 1835 – 1910

3.1 Introduction

The number of flight passengers worldwide is rapidly growing at about 5.3% per year (see SITA [83]), which is challenging for the existing infrastructure at international airports. The increasing number of travelers corresponds to a growing volume of baggage to be handled.

In what follows we describe the outbound baggage handling process which starts with receiving outbound baggage from check-in passengers or transfer flights. Within the terminal, outbound baggage is transported to its destinations, a handling facility or a central storage system by the baggage handling system (BHS), which is a conveyor belt network. While check-in baggage enters the BHS through check-in counters, transfer baggage is brought with baggage tugs from the transfer flight to infeed-stations, located at the apron, where the baggage is transfered into the BHS. If baggage for an outgoing flight arrives at the check-in or a transfer infeed-station during the flight's baggage handling period, which begins one to three hours before the scheduled departure time and ends 10 to 15 minutes before a flight's departure time, the arriving baggage is directed to the assigned handling facility, where ground handlers sort and load the incoming bags into bulk-containers. In the case bags arrive before baggage handling has started, the bags are directed to a central storage system, where they are stored until baggage handling begins. For example, the capacity of storage system at a major European Airport is limited to about 3,000 bags, which is rather small in comparison to the total amount of approximately 30,000 bags which arrive per day. At the end of the baggage handling process, the containers are transported to the airplane to be loaded into the cargo hold.

The handling facility, also denoted carousel, is an oval-shaped conveyor belt on which several flights can be processed at one time (see Figure 3.1). A slide system connects the BHS conveyor belt with the carousel's conveyor belt. Bags from the BHS which are transfered to the carousel via the slide system are picked up from the carousel's conveyor belt by workers and loaded into containers. If the arrival rate of bags to the carousel exceeds the loading rate of the workers at the carousel, the number of bags on the carousel is increasing. The number of bags on the carousel defines the carousel's workload. If the workload of a carousel reaches the carousel's capacity, i.e. the maximum number of bags which can be on the carousel, the bags in the BHS' conveyor system are no longer forwarded to the carousel's conveyor belt and remain in the BHS until the workload on the carousel falls below its capacity. From a technical perspective, additional bags in the BHS increase the danger of bottlenecks in the network, which may result in delays and disruptions in the outbound baggage handling process. From the employees' perspective, high workload peaks may lead to an unfair distribution of work among the workers. To avoid both problems, groundhandlers employ additional temporary workers, which however leads to increased labor costs.

For airlines and their groundhandlers it is important to know the assigned carousels and departing flight's baggage handling period some hours before the handling of the flights starts. The planning of outbound baggage handling comprises the assignment of flights to carousels, the scheduling of the start time of flights' handling process and the start time for the storage depletion. Most airports generate a daily plan by using simple allocation heuristics. Because the general problem is NP-hard to solve (see Section 3.3) and over 350 flights at 22 carousels have to be handled per day in average at an international Airport, the generated plans result in poor solutions with high workloads on the carousels leading to the problems described above. As the number of available carousels is rather small in comparison to the number of outgoing flights and an expansion of the handling facilities is very costly, the airport seeks to find advanced approaches which reduce the workload peaks subject to the given infrastructure. In this chapter, we present a model

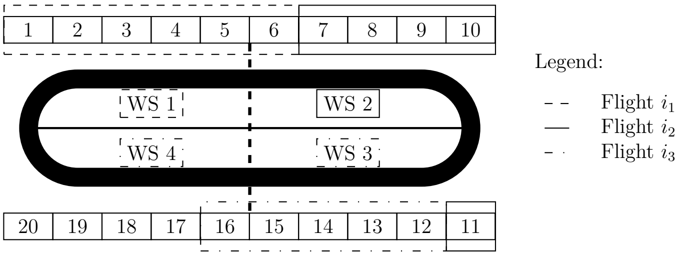


Figure 3.1: Feasible assignment of three flights at a carousel with 20 parking positions and four working stations

and a solution procedure to assign outgoing flights to carousels and schedule the baggage handling, taking into account the BHS' capacities and the arrival profiles of the baggage streams. The objective is to obtain a workload below a given target value across all carousels during the main peak periods of the day.

Literature on planning the outbound baggage handling processes is very scarce. Abdelghany et al. [2] and [8] (see also Ascó et al. [7]) propose greedy assignment algorithms to assign baggage handling facilities to departing flights assuming a given service periods for flights' baggage handling. However, they neither considers capacities of the airport's infrastructure for baggage handling nor the dynamics of the baggage arrival process in their planning. Furthermore, instead of assuming given handling periods for flights' baggage handling, we determine the schedule for baggage handling, which makes our approach more flexible. A preliminary mixed-integer program for a similar problem statement to ours is presented by [42] and [41] but no efficient method is proposed to solve the problem.

Our main contributions in this chapter are as follows:

(1) A time-indexed formulation for the outbound baggage handling problem (OBHP) considering the dynamics of the incoming baggage stream and the capacities of airport's infrastructure is presented;

(2) Structural properties and the problem complexity are established;

(3) An efficient solution methodology based on column generation is presented which improves heuristic solutions by 64.97%;

(4) Acceleration techniques for column generation are discussed.

The remainder of this chapter is organized as follows: In § 3.2 we provide a formal problem description of the OBHP. The time-indexed mathematical model is presented in § 3.3. We discuss structural properties in Section 3.3.2 and exhibit dominance criteria to strengthen the solution space in § 3.3.3. The solution procedure is proposed in § 3.4, which decomposes the model into a master problem (MP) and three scheduling subproblems. Techniques to improve the column quality are discussed in § 3.5, and § 3.6 presents computational experiments. Conclusions are drawn in § 3.7.

3.2 Problem description

The set of all outgoing flights is denoted by $\mathcal{F} = \{1, \ldots, F\}$. Baggage handling for all flights takes place within the discrete planning horizon $\mathcal{T} = \{t_0, \ldots, t_T\}$ with $t_0 = 0 < t_1 < \ldots < t_T$; all time points are evenly spaced in time, and each time t_k represents the begin of period $[t_k, t_{k+1}[$ for $k = 0, \ldots, T-1$. The amount of baggage arriving during time period t for flight $i \in \mathcal{F}$ at the BHS is given by $A_i = (A_{i,t})_{t=0,\ldots,T-1}$. The capacity of the central storage system is given by K^{s}.

Next, we give a detailed description of the assignment process of flights to carousels and the scheduling of flight baggage handling; the required notation for the assignment and scheduling is summarized in Table 3.1.

Flight assignment to carousels A two-sided carousel $c \in \mathcal{C} = \{1, \ldots, C\}$ is comprised of working stations and offers space for a limited number of containers on its parking positions lined-up along longside of a carousel's conveyor belt where one parking position can accommodate one container at a time (see Figure 3.1). At a working station, workers load bags circling on the conveyor belt into bulk-containers. To ensure that each bag is placed into the right container, each bag and bulk-container is identified with a unique bar-code. With a hand-held scanning device of a working station, a worker scans the bar-code of a bag and a bulk-container. If the two codes match, the bag is loaded into the containers. For security reasons, a working station is equipped with exactly one scanning device which can only be used for one flight at a time. On both sides, a carousel has the same number of working stations as well as the same number of parking positions. We distinguish between $R = \{1, \ldots, R\}$ carousel types where the set of carousels of type $r \in \mathcal{R}$ is denoted by $\mathcal{C}_r = \{1, \ldots, C_r\}$ with $\mathcal{C}_{r_1} \cap \mathcal{C}_{r_2} = \emptyset$ for each pair $r_1, r_2 \in \mathcal{R}$ with

$r_1 \neq r_2$; index r_c indicates carousel c's resource type r. Carousel types differ in the number of offered parking positions K_r^{pp}, working stations K_r^{ws} and conveyor belt capacity K_r^{cb} (see Table 3.1). Each working station belongs to a segment of the carousel's conveyor belt and has a given number of K_r^{ppws} adjacent parking positions lined-up along its segment. In Figure 3.1, for example, the carousel consists of four working stations where each working station has five adjacent parking positions. The number of bags which can be on a carousel's conveyor belt in a period is bounded by carousel's conveyor belt capacity K_r^{cb}. The workload on a carousel relative to the carousels' capacity yields the carousel's utilization.

Carousel type r	
K_r^{cb}	Conveyor belt capacity
K_r^{pp}	Number of parking positions for bulk-containers
K_r^{ws}	Number of working stations
K_r^{ppws}	Number of parking positions in a working station segment
Flight i	
A_i	Baggage arrival vector
P_i	Number of required bulk-containers
$\left[W_{i,r}^{\min}, W_{i,r}^{\max}\right]$	Minimal and maximal number of required working stations at carousels type r
$\left[W_i^{\min}, W_i^{\max}\right]$	Minimal and maximal number of required working stations across all carousels
$\left[S_i^{\text{es}}, S_i^{\text{ls}}\right]$	Time window to start flight's baggage handling
S_i^{e}	End time of flight i's baggage handling

Table 3.1: Carousel type and flight parameters

The number of containers P_i for flight i which have to be placed on carousel's parking positions is provided by the airline. As an airline considers sorting criteria for baggage such as first and economy baggage and also distinguishes between transfer baggage and inbound baggage, i.e. baggage leaving the destination airport by customs, the airport is forced to place all required containers simultaneously at the carousel at a time. Moreover, a flight has to be assigned to exactly one carousel, but a carousel can handle more than one flight at a time. To reduce bags' lift distances from the conveyor belt to the containers for workers, a flight can only be handled at a working station if at least one of the flight's containers is placed on one of the parking positions in its segment. However, a flight's container-row can overlap with parking positions of another working stations segment without using its working station. To ease the towing of flight containers with a baggage tug to the aircraft, the containers are sequentially ordered. If a flight's containers are assigned to both sides of the carousel, then the container rows are lined up opposite of each other and start at the rightmost or leftmost parking position on each side (see flight i_2 in Figure 3.1).

Given the assignment regularities above, the minimal and maximal number of working stations for a flight i at a carousel type r results from the number of required containers P_i and is given by $W_{i,r}^{\min} = \max\left\{\left\lfloor \frac{P_i}{K_r^{ppws}} \right\rfloor, 1\right\}$ and $W_{i,r}^{\max} = \left\lceil \frac{P_i}{K_r^{ppws}} \right\rceil$, respectively.

Flights baggage handling schedule Let $\left[S_i^{\mathrm{es}}, S_i^{\mathrm{ls}}\right]$ denote the time window within flight i can start its baggage handling. S^{es} and S_i^{ls} are the earliest and latest start time for baggage handling and are set according to flight i's expected number of bags. Baggage handling at a carousel ends at S_i^{e} which is set 10 minutes before the flights' scheduled departure. The storage depletion for a flight can only start, if the baggage handling of the flight has already been started. As soon as the storage depletion begins, we assume a constant baggage transfer rate from the central storage to the assigned carousel.

In the following, we denote a baggage handling schedule by start time tuples $\tau = \langle s_\tau^{\mathrm{h}}, s_\tau^{\mathrm{d}} \rangle$, where decision s_τ^{h} and s_τ^{d} stand for the start time of the baggage handling at a carousel and start time of storage depletion, respectively. For flight i, set $\mathcal{S}_i$ denotes the set of all feasible start time tuples τ. As the number of loaded bags per period depends on the number of assigned working stations we denote by $\mathcal{S}_i(w) \subseteq \mathcal{S}_i$ the feasible start time tuples if w working stations are assigned. Start time tuples $\tau_1, \tau_2 \in \mathcal{S}_i$ are ordered with relation $\preceq$ where $\tau_1 \preceq \tau_2$ iff $s_{\tau_1}^{\mathrm{h}} < s_{\tau_2}^{\mathrm{h}}$ or iff $s_{\tau_1}^{\mathrm{h}} = s_{\tau_2}^{\mathrm{h}}$ and $s_{\tau_1}^{\mathrm{d}} \leq s_{\tau_2}^{\mathrm{d}}$.

To avoid a high utilization, airport's objective is to obtain an utilization near or below a target utilization $u^{\mathrm{ta}} < 1$ across all carousels during working day's main peak periods subject to the assignment and scheduling regulations described above. Figure 3.2 shows the baggage arrival rate during a day at a major European Airport. We can identify three peaks, one in the morning, one in the afternoon and one in the evening. To avoid peaks in any of the time intervals, we divide the planning horizon $\mathcal{T}$ into $M \geq 1$ disjoint time intervals $\mathcal{T}_1, \ldots, \mathcal{T}_M$. For example in Figure 3.2 we have $M = 3$.

In the next section, we will present the mathematical model formulation. We will give a model analysis and exhibit dominance criteria to strengthen the solution space.

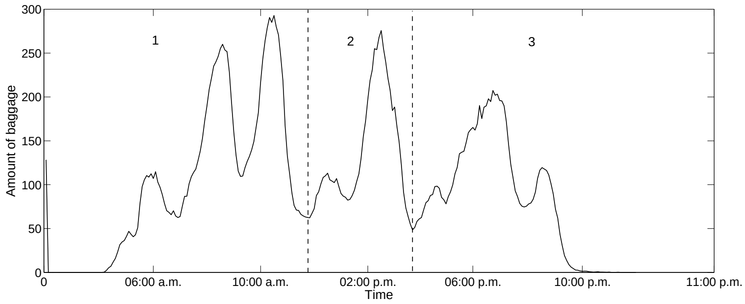


Figure 3.2: Average arrival curve of baggage during the course of a day

3.3 Mathematical model

In § 3.3.1 we present a time-indexed formulation for the OBHP and provide an analysis of it in § 3.3.2. As the model includes a high number of variables, we present in § 3.3.3 dominance criteria to strengthen the solution space.

3.3.1 Time indexed formulation

Threshold values $0 < u_1 < \ldots < u_K$ measure the deviation from target utilization u^{ta} of each carousel during any time interval $1 \leq m \leq M$. A deviation greater than u_k incurs a penalty of $0 < p_1 < \ldots < p_k < \ldots < p_K$ for $k = 1, \ldots, K$.

The number of feasible start time tuple for each flight i in set $\mathcal{S}_i$ is bounded by $\max_{i \in \mathcal{F}} \{|\mathcal{S}_i|\} \leq \max_{i \in \mathcal{F}} \left\{ \sum_{z=0}^{S_i^{\text{ls}} - S_i^{\text{es}}} S_i^{\text{e}} - (S_i^{\text{es}} + z) \right\}$. Given flight i's arrival process A_i, a feasible start time tuple $\tau \in \mathcal{S}_i$ and a given number of working stations w, we can derive the following two parameters for flight i:

- $\Gamma_{t,\tau}^{i,w}$ – flight i's workload on a carousel during period t;
- $\Phi_{t,\tau}^{i}$ – amount of flight i's stored baggage in the central storage system during period t.

The key decision variable for the presented time-indexed formulation is the four indexed binary variable $x_{i,c,w,\tau}$, which is equal to 1 if flight i is processed at carousel $c \in \mathcal{C}$ using w working stations and employs start time tuple τ. Binary auxiliary variable $y_{c,k,,m}$ is equal to 1, if the utilization of carousel c exceeds target utilization u^{ta} by u_k in time interval m.

The OBHP for planning the outbound baggage handling can now be stated as follows

$$\text{minimize} \quad \sum_{c \in \mathcal{C}} \sum_{k=1}^{K} \sum_{m=1}^{M} p_k \cdot y_{c,k,m} \tag{38}$$

subject to

$$\sum_{c \in \mathcal{C}} \sum_{w=W_{i,r}^{\min}}^{W_{i,r}^{\max}} \sum_{\tau \in \mathcal{S}_i(w)} x_{i,c,w,\tau} = 1 \qquad \forall\ i \in \mathcal{F} \tag{39}$$

$$\sum_{i \in \mathcal{F}: S_i^{\text{es}} \le t < S_i^{\text{e}}} \sum_{w=W_{i,r}^{\min}}^{W_{i,r}^{\max}} \sum_{\tau \in \mathcal{S}_i(w): s_\tau^{\text{h}} \le t} w \cdot x_{i,c,w,\tau} \le K_r^{\text{ws}} \qquad \forall\ r \in \mathcal{R}, c \in \mathcal{C}_r, t \in \mathcal{T} \tag{40}$$

$$\sum_{i \in \mathcal{F}: S_i^{\text{es}} \le t < S_i^{\text{e}}} \sum_{w=W_{i,r}^{\min}}^{W_{i,r}^{\max}} \sum_{\tau \in \mathcal{S}_i(w): s_\tau^{\text{h}} \le t} P_i \cdot x_{i,c,w,\tau} \le K_r^{\text{pp}} \qquad \forall\ r \in \mathcal{R}, c \in \mathcal{C}_r, t \in \mathcal{T} \tag{41}$$

$$\sum_{i \in \mathcal{F}: S_i^{\text{es}} \le t < S_i^{\text{e}}} \sum_{c \in \mathcal{C}} \sum_{w=W_{i,r}^{\min}}^{W_{i,r}^{\max}} \sum_{\tau \in \mathcal{S}_i(w)} \Phi_{t,\tau}^{i} \cdot x_{i,c,w,\tau} \le K^{\text{s}} \qquad \forall\ t \in \mathcal{T} \tag{42}$$

$$\frac{1}{K_r^{\text{cb}}} \cdot \sum_{i \in \mathcal{F}: S_i^{\text{es}} \le t < S_i^{\text{e}}} \sum_{w=W_{i,r}^{\min}}^{W_{i,r}^{\max}} \sum_{\tau \in \mathcal{S}_i(w)} \Gamma_{t,\tau}^{i,w} \cdot x_{i,c,w,\tau} - \sum_{k=1}^{K} u_k \cdot y_{c,k,m} \le u^{\text{ta}} \qquad \forall\ r \in \mathcal{R}, c \in \mathcal{C}_r, 1 \le m \le M, t \in \mathcal{T}_m \tag{43}$$

$$\sum_{k=1}^{K} y_{c,k,m} \le 1 \qquad \forall\ c \in \mathcal{C}, 1 \le m \le M \tag{44}$$

$$x_{i,\tau,c,w} \in \{0,1\} \qquad \forall\ i \in \mathcal{F}, c \in \mathcal{C}, \tau \in \mathcal{S}_i, W_{i,r_c}^{\min} \le w \le W_{i,r_c}^{\max} \tag{45}$$

$$y_{c,k,m} \in \{0,1\} \qquad \forall\ c \in \mathcal{C}, 1 \le k \le K, 1 \le m \le M \tag{46}$$

Objective function (38) minimizes the sum of all penalty values for the violation of threshold value in the M time intervals. Constraints (39) ensure that for each flight $i \in \mathcal{F}$ a feasible start time tuple of set $\mathcal{S}_i$ is selected and each flight i is assigned to one carousel and to $W_{i,r}^{\min} \leq w \leq W_{i,r}^{\max}$ working stations. Capacity constraints (40) and (41) impose the resource restrictions given by the available number of working stations and parking positions at a carousel, respectively. The capacity of the central storage system is bounded by constraints (42). The threshold value for the capacity violation in each time interval $1 \leq m \leq M$ is selected in constraints (44). Constraints (43) set the violation of the threshold value during time interval m.

A solution of time-indexed formulation (38)-(46) does not consider the assignment rules described in Section 3.2. However, given a feasible solution for problem formulation (38)-(46) it is always possible to map it into a solution respecting the assignment rules in a post-processing step.

Theorem 3.3.1. *Time-indexed formulation (38)-(46) solves the OBHP.*

Proof. The proof of Theorem 3.3.1 is constructive and contains a polynomial time assignment algorithm to obtain a feasible solution for the OBHP. Each carousel $c \in \mathcal{C}$ can be represented as a circle where the parking positions and working stations are clock-wise ordered from 1 to K_i^{pp} and 1 to K_c^{ws}, respectively (see Figure 3.3 for the circle representation of the carousel of Figure 3.1 where 2 flights are assigned). The working stations split the circle into K_c^{ws} circular sectors. To obtain a feasible solution for the OBHP in terms of the regulations discussed in Section 3.2, flights' container rows are placed on the border of the circle where the container row of one flight covers a connected part of circle's border such that the assigned working stations are reached and the container row does not overlap with any other container row of a flight handled simultaneously.

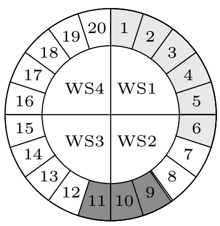


Figure 3.3: Circle representation of the carousel in Figure 3.1; gray shaded parking positions are used by flights

To proof the theorem it is sufficient to consider flights assigned to the same carousel. We have to show that, given a solution by TIF, it is possible to place all containers of assigned flights to the carousel such that all assigned working stations for the flights are reached and there is no overlapping with other flight handled simultaneously. For the assignment, we order the flights by increasing baggage handling ending times. A container row of a flight is assigned clockwise beginning with the parking position immediately following the container row of the previously assigned flight which ensures that all required working stations are reached. If no flight is assigned so far, we start with parking position 1. Figure 3.3, for example, shows the feasible assignment of 2 flights. Assume, flights i_1 to i_k are handled simultaneously where flight i_1 to i_k are ordered according to their baggage handling ending times. Let us assume that flight i_1's to i_{k-1}'s container row are assigned with the above assignment rule and assume that w.l.o.g. flight i_1 to i_k require K_c^{ws} working station. If flight i_k's container row overlaps with flight i_1's container row by $p \geq 1$ parking positions then we right-shift the container row of flight i_1 for p parking positions in clockwise order until the overlapping conflict is solved. If it is not possible to right-shift flight i_1's container row as it is blocked by flight i_2's container row, we sequentially right-shift the container rows of the succeeding flights i_2 to i_{k-2} until we obtain the additional p parking positions for flight i_k; a right-shift of flight i_{k-1}'s container rows is not required as it would directly lead to a reduction of available parking positions for flight i_k. In Figure 3.3, for example, flight i_1's container row can be right-shifted by at most 2 parking positions. If it is not possible to right-shift flight i_1 to i_{k-2}'s container rows for p parking positions, then there are are either not enough parking positions for flight i_k available or a right-shift of flight i_1's container row is not possible as it would contradict the requirement that container row reaches the assigned working stations. In the first case, we violate the assumption that it is possible to handle the flights i_1 to i_k simultaneously due to a violation of the parking position capacity. In the latter case, flight i_k would require at least $p \geq K_c^{\text{ppws}}$ additional parking positions, and hence at least one additional working station, which violates the working station capacity and therefore also contradicts the assumption that the flights can be handled simultaneously. □

3.3.2 Model analysis

The OBHP contains at most $(3 \cdot F \cdot \max_{i \in \mathcal{F}} \{|\mathcal{S}_i|\} + K \cdot M) \cdot C$ variables and $T \cdot (C \cdot (2 + M) + 1) + C \cdot M + F$ constraints. Consider a real-world scenario with 350 flights, a flights' baggage handling time window of 120 minutes and 22 carousels for a planning horizon of 236 periods, where we plan from 03:00 a.m. to 10:40 p.m. in five minutes intervals. If the number of threshold values and time intervals is equal to three, OBHP leads to at most $554,598$ variables and $34,246$ constraints.

As part of the OBHP, the question arises whether a feasible solution exists, i.e. whether F flights (items) can be assigned and scheduled to C carousels (bins). Hence, the decision problem can be reduced from the BIN-PACKING PROBLEM, which is known to be NP-hard (see [43]) yielding OBHP's complexity.

Theorem 3.3.2. *OBHP is NP-complete.*

However, the NP-hardness is not only established because of the assignment problem but also the scheduling problem is NP-hard to solve. Assume a feasible assignment of flights to carousels is given and the scheduling decision to start flights' baggage handling still has to be done such that a minimized utilization is obtained. Then, the problem as a generalization of the 3-PARTITIONING PROBLEM is NP-complete. Hence, we can state

Theorem 3.3.3. *Given a feasible assignment of flights to carousels, the scheduling problem to obtain a minimized workload is NP-complete.*

Proof. It is clear that the scheduling problem to minimize the utilization is in NP. We prove the NP-hardness by reduction from the 3-PARTITIONING PROBLEM, which is known to be NP-complete in the strong sense (see [43]). Given a finite set $\mathcal{E}$ of $3 \cdot m$ elements, a bound $B \in \mathbb{Z}_+$ and a size $s(e) \in \mathbb{Z}_+$ for each $e \in \mathcal{E}$ with $\sum\limits_{e \in \mathcal{E}} s(e) = m \cdot B$. The question to be answered is whether there is a partition of $\mathcal{E}$ into m disjoint sets $\mathcal{E}_1, \ldots, \mathcal{E}_m$ such that $\sum\limits_{e \in \mathcal{E}_k} = B$ for $1 \leq k \leq m$.

For the transformation, we consider 1 carousel with a conveyor belt capacity of 1. The capacities for central storage, carousel's parking positions and working stations are relaxed. Let $\mathcal{F} = \mathcal{E}$ be the set of departing flights. The baggage handling of each flight starts within the time window $[1, T-1]$ with $T - 1 = m$ and ends at T. Moreover, if we assume that the start of flights' storage depletion is equal to the start of the baggage handling, we obtain $\mathcal{S}_i = \{\langle 1, 1\rangle, \langle 2, 2\rangle \ldots, \langle T-1, T-1\rangle\}$ as set for possible start time tuples for each $i \in \mathcal{F}$. Moreover, for each flight i we set $\Gamma^i_{t,\tau} = s(e)$ if $t = s^{\mathrm{h}}_{\tau}$, and 0 otherwise. Let binary variable $x_{i,\tau}$ be 1, if start time tuple τ is selected for flight, and 0 otherwise. Then, the scheduling problem

$$\sum_{\tau \in \mathcal{S}_i} \Gamma^i_{t,\tau} \cdot x_{i,\tau} \leq B \qquad \forall\, t \in \mathcal{T} \tag{47}$$

$$\sum_{\tau \in \mathcal{S}_i} x_{i,\tau} = 1 \qquad \forall\, i \in \mathcal{F} \tag{48}$$

$$x_{i,\tau} \in \{0, 1\} \qquad \forall\, i \in \mathcal{F}, \tau \in \mathcal{S}_i \tag{49}$$

solves the 3-PARTINIONING PROBLEM. However, scheduling problem (47)-(48) represents a relaxed version of the scheduling problem to minimize the utilization in which the decision whether there is a maximal utilization of at most B in each time interval has to be answered. $\square$

3.3.3 Preprocessing

We strengthen the solution space by tightening the bound for the storage capacity in constraints (42) and by reducing the number of decision variables $x_{i,\tau,c,w}$. Each flight $i \in \mathcal{F}$ consumes storage capacity at time t if flight i has its earliest start time for baggage handling after time $t < S_i^{\text{es}}$ and if $A_{i,t} > 0$. Thus, bound K^{s} for the storage capacity of the central storage in constraints (42) is strengthened by $K_t^{\text{s}} = K^{\text{s}} - \sum_{i \in \mathcal{F}: t < S_i^{\text{es}}} \Phi_{t,\tau}^i$ for all $t > 0$ and one $\tau \in \mathcal{S}_i$. For $t = 0$ we set $K_0^{\text{s}} = K^{\text{s}} - \sum_{i \in \mathcal{F}} A_i^{\text{early}}$ where A_i^{early} stands for the amount of early baggage which has arrived before time 0.

The number of variables $x_{i,\tau,c,w}$ heavily depends on the cardinality of sets $\mathcal{S}_i$. To reduce the number of feasible start time tuples in $\mathcal{S}_i$, first, we strengthen the time windows for the baggage handling, before we exhibit dominance criteria for two different start time tuples of the same flight i.

Earliest start time Assume for flight $i \in \mathcal{F}$ that no bag has arrived up to the earliest baggage handling starting time S_i^{es}. Thus, if baggage handling is delayed by one period to $S_i^{\text{es}} + 1$, there will be no consumption of the central storage capacity by flight i. At the same time, the time span in which flight i is assigned to the carousel is decreased. As the objective function does not increase when starting flight i at $S_i^{\text{es}} + 1$ instead of S_i^{es}, the earliest start time can be set to $S_i^{\text{es}} = S_i^{\text{es}} + 1$. The update is repeated as long as no bags have arrived to the current earliest start time for baggage handling.

Latest start time The latest start time of a flight is left-shifted one period earlier, if the capacity of the storage system is violated in case of flight i is scheduled with the latest possible start time tuple $\tau_i^{\text{ls}} = \langle S_i^{\text{ls}}, S_i^{\text{ls}} \rangle$. Let $\mathcal{F}_i^{\text{parF}}$ be the the subset of flights which can be handled in parallel with flight i. If inequality $\left(\sum_{j \in \mathcal{F}_i^{\text{parF}}} \Phi_{t,\tau_j^{\min}}^j \right) + \Phi_{t,\tau_i^{\text{ls}}}^i \leq K_t^{\text{t}}$ with $\tau_j^{\min} = \{\tau \in \mathcal{S}_j \mid \tau \preceq \tau' \text{ for all } \tau' \in \mathcal{S}_j\}$ is violated for at least one $t \in [S_i^{\text{es}}, S_i^{\text{e}}]$, the latest start time is updated to $S_i^{\text{ls}} = S_i^{\text{ls}} - 1$. The update is repeated as long as the capacity of the central storage system is violated when start time tuple τ_i^{ls} is selected.

Arbitrary start time tuple We can state dominance criteria for two different start time tuple of the same flight based on the following observation: A right-shift of a flight's baggage handling starting time decreases the handling period and may increase the number of bags in the central storage system, while a left-shift has the opposite effect.

The first proposition refers to start time tuples with the same baggage handling start time but different start times for the storage depletion.

Proposition 3.3.4. *Let* $\tau_1, \tau_2 \in \mathcal{S}_i$ *be two feasible start time tuples for flight* $i \in \mathcal{F}$ *with* $s^h_{\tau_1} = s^h_{\tau_2}$ *and* $s^d_{\tau_1} < s^d_{\tau_2}$*. If there is a* $W^{min}_i \leq \bar{w} \leq W^{max}_i$ *such that* $\Gamma^{i,\bar{w}}_{t,\tau_1} \leq \Gamma^{i,\bar{w}}_{t,\tau_2}$ *for all* $t \in \mathcal{T}$*, then* τ_2 *can be removed from set* $S_i(w)$ *for* $\bar{w} \leq w \leq W^{max}_i$ *without excluding the optimal solution.*

Proposition 3.3.4 is not valid for two start time tuples $\tau_1, \tau_2 \in \mathcal{S}_i$ with $s^{\mathrm{h}}_{\tau_1} < s^{\mathrm{h}}_{\tau_2}$, as a postponed start for baggage handling may violate the central storage's capacity. In order to state a dominance criterion for arbitrary tuples we extend Proposition 3.3.4 by the storage perspective .

Proposition 3.3.5. *Let* $\tau_1, \tau_2 \in \mathcal{S}_i$ *be two feasible start time tuples for flight* $i \in \mathcal{F}$ *with* $\tau_1 \prec \tau_2$*. Assume there is a* $W^{min}_i \leq \bar{w} \leq W^{max}_i$ *such that* $\Gamma^{i,\bar{w}}_{t,\tau_1} \geq \Gamma^{i,\bar{w}}_{t,\tau_2}$ *for all* $t \in \mathcal{T}$*, and inequality* $\left(\sum_{j \in \mathcal{F}^{parF}_i} \Phi^j_{t,\tau^{max}_j}\right) + \Phi^i_{t,\tau_2} \leq K^s_t$*, with* $\tau^{max}_j = \{\tau \in \mathcal{S}_j \mid \tau \succeq \tau' \text{ for all } \tau' \in \mathcal{S}_j(\bar{w})\}$*, is satisfied for any* $t \in \left[S^{es}_i, s^d_{\tau_2} + p^d_i(s^h_{\tau_2}) - 1\right]$ *where* $p^d_i(s^h_\tau) \geq 0$ *is the duration to send all stored bags to the assigned carousel if the baggage handling start at* s^h_τ*. Then tuple* τ_1 *can be removed from sets* $\mathcal{S}_i(w)$ *for all* $\bar{w} \leq w \leq W^{max}_i$ *without excluding the optimal solution.*

In our computational study in § 3.6, Proposition 3.3.4 and 3.3.5 lead to a column reduction of 35.16% and 11.25%, respectively (see Table 3.6 and 3.4).

3.4 Solution methodology

Solving OBHP with a traditional branch and bound algorithm as it is implemented in off the shelf MIP-Solvers such as CPLEX results in a bad convergence behavior with high computational times. Reasons are OBHP's complexity, the large number of variables and the large number of equal assignment patterns of flights to carousels due to the symmetry effect, i.e. during the branch-and-bound procedure the same solution is investigated more than one time. Indeed, we could not solve any real-world instance using CPLEX.

To overcome these problems, we provide a Dantzig-Wolfe formulation of the OBHP leading to a reduction of symmetry. The received master problem (MP) in Section 3.4.1

is solved by means of column generation. During column generation we consider three types of subproblems which are presented in Section 3.4.2: MP's pricing problems (PPs) generating new columns (see Section 3.4.2.1), two scheduling problems improving the solution found by MP (see Section 3.4.2.2), and updating an upper bound u^{fix} for carousels' utilization (see Section 3.4.2.3). Section 3.5 discusses advanced implementation details to accelerate column generation and to improve the quality of generated columns.

Algorithm 1 provides an overview of the branch-and-cut-and-price with the three subproblems and the reference to the subsection describing these problems. The column generation (see Algorithm 1 step 2 to 19) terminates as soon as each PP does not provide a column with negative reduced cost and the LP bound of MP is reached (see Algorithm 1 step 20). If no feasible solution is found at the end of column generation, the solution space is partitioned (see Algorithm 1 step 21) with the multi-pattern branching of [33] who extend the two-pattern branching rule of [81]. As search strategy we apply depth-first branching. The generated cuts for each node of the branch-and-price tree are collected in cut set $\mathcal{CS}$. In each loop of the algorithm, we examine one cut $o \in \mathcal{CS}$ which is added to RMP's PPs to forbid some columns (see Section 3.4.2.1). RMP with added cut $o \in \mathcal{CS}$ can be fathomed if

- RMP's LP-bound is greater or equal to u^{fix};
- RMP is not feasible;
- RMP returns a feasible (integer) solution.

Otherwise, the node corresponding to cut o is branched and new cutting planes are added to set $\mathcal{CS}$. The mathematical model formulations and their usage in Algorithm 1 will be discussed in the following subsections.

3.4.1 Master problem

Let $\mathcal{D}_r(u^{\text{fix}})$ be the set of duties for carousels of type $r \in \mathcal{R}$. A duty $d \in \mathcal{D}_r(u^{\text{fix}})$ represents a feasible assignment of flights to a carousel of type r and schedule for flights' baggage handling period in which u^{fix} is an upper bound for duties d's maximal allowed utilization in each time interval $1 \leq m \leq M$. Moreover, a duty d contains the information

- $\Theta^{\text{a}}_d = \left(\Theta^{\text{a}}_{d,i}\right)_{i \in \mathcal{F}}$ – equal to one if flight i is contained in duty d;

Algorithm 1 Branch-and-Price-and-Cut

$u^{\text{fix}} \leftarrow \infty$; $U_m \leftarrow u_K$ for all $1 \leq m \leq M$; $\mathcal{CP} \leftarrow \{\emptyset\}$
while $\mathcal{CP} \neq \emptyset$ **do**
  RPP ← `addNextCuts`(RPP) ▷ RPP, see Sec. 3.4.2.1
  while RMP's lower bound is not reached **do**
    `solve`(RMP) ▷ RMP, see Section 3.4.1
    $\mathcal{D}'(u^{\text{fix}}) \leftarrow$ `solve`(RPP)
    if RMP is feasible **then**
      $\left(u^{\text{fix}}, \mathcal{T}_1(u^{\text{fix}}), \ldots, \mathcal{T}_M(u^{\text{fix}})\right) \leftarrow$ `solve`(FSP) ▷ FSP, see Sec. 3.4.2.2
      for $1 \leq m \leq M$ **do**
        for $r \in \mathcal{R}$ and $t \in \mathcal{T}_m(u^{\text{fix}})$ **do**
          `solve`(DSP(r,t)) ▷ DSP(r,t), see Sec. 3.4.2.3
        end for
        `Update`$(U_m(u^{\text{fix}}))$
      end for
      if DSP is not feasible **then**
        STOP
      end if
    end if
  end while
  if RMP can not be fathomed **then**
    $\mathcal{CP} \leftarrow$ `branch-and-cut`(RMP)
  end if
end while

- $\Theta^{\text{s}}_d = \left(\Theta^{\text{s}}_{d,t}\right)_{t \in \mathcal{T}}$ – number of bags in the central storage during time period t;
- H_d – penalty value for the threshold violations summed up over all time intervals.

Binary variable z_d is 1, if duty $d \in \mathcal{D}(u^{\text{fix}}) = \bigcup_{r \in \mathcal{R}} \mathcal{D}_r(u^{\text{fix}})$ is selected, and 0 otherwise. The MP, which is equivalent to time-indexed formulation (38)– (46) if $u^{\text{fix}} = \infty$, can now be stated as

$$\text{minimize} \quad \sum_{r \in \mathcal{R}} \sum_{d \in \mathcal{D}_r(u^{\text{fix}})} H_d \cdot z_d \tag{50}$$

subject to

$$\sum_{r \in \mathcal{R}} \sum_{d \in \mathcal{D}_r(u^{\text{fix}})} \Theta^{\text{a}}_{d,i} \cdot z_d = 1 \qquad \forall\, i \in \mathcal{F} \tag{51}$$

$$\sum_{r \in \mathcal{R}} \sum_{d \in \mathcal{D}_r(u^{\text{fix}})} \Theta^{\text{s}}_{d,t} \cdot z_d \leq K^{\text{s}}_t \qquad \forall\, t \in \mathcal{T} \tag{52}$$

$$\sum_{d \in \mathcal{D}_r(u^{\text{fix}})} z_d \leq |\mathcal{C}_r| \qquad \forall\, r \in \mathcal{R} \tag{53}$$

$$z_d \in \{0, 1\} \qquad \forall\, d \in \mathcal{D} \tag{54}$$

Constraints (51) assign each flight i to one duty. The storage capacity is bounded in constraints (52). Constraints (53) restrict the number of available carousels of each type r. Objective function (50) minimizes the sum of penalties for the utilization across all carousels.

The number of feasible duties for each carousel type is exponential in size. To increase MP's computational tractability, MP is restricted to a subset of duties $\mathcal{D}'_r(u^{\text{fix}}) \subseteq \mathcal{D}_r(u^{\text{fix}})$ for each carousel type $r \in \mathcal{R}$, leading to the restricted MP (RMP). As $\mathcal{D}'(u^{\text{fix}}) = \bigcup_{r\in\mathcal{R}} \mathcal{D}'_r(u^{\text{fix}})$ may not contain duties resulting in an optimal or even feasible solution, we generate new columns by means of column generation (see [31] or [67] among others). Hence, the linear relaxation of RMP (L-RMP) yields the dual variables of constraints (51) to (53) as query to find new duties having negative reduced cost in one of the R pricing problems (PP) (see Section 3.4.2.1). If one negative reduced cost column for type r is found, we add this column RMP (see Algorithm 1 step 3).

In Section 3.5.2 we show a modified version of RMP which adds Chebyshev cutting planes in the solution space of RMP. These cuts help to avoid dual degeneracy and, therefore, improve the quality of the dual variables required to query for appropriate column in the PP.

3.4.2 Subproblems

While running column generation we consider three types of subproblems. The first type of subproblem is the PP to generate new columns (see Section 3.4.2.1). To improve the quality of generated duties, we only consider a subset of threshold values for the utilization violation when PP is solved. The second and third subproblems are scheduling problems solved after RMP has found a feasible solution. In a first step, RMP's feasible solution is improved by re-scheduling flights' baggage handling. In a second step, we search for the next subset of threshold value for PP which lead to an improved feasible solution.

3.4.2.1 Pricing problem

We restrict the dual space of dual variable corresponding to constraints (51) to the negative side of the dual polyhedron by replacing the "=" constraint by the"≥" constraint. Thus, each flight is assigned to at least one duty. As we minimize the utilization, flights assigned to more than one duty can be removed by which the objective function does not increase. Restricting constraints (51)'s dual variables to the positive side of the dual polyhedron is a valid dual inequality for the OBHP (see [5] for a general discussion of dual inequalities).

Let $\lambda = (\lambda_i)_{i \in \mathcal{F}} \in \mathbb{R}^F_+$, $\mu = (\mu_t)_{t \in \mathcal{T}} \in \mathbb{R}^T_-$ and $\nu = (\nu_r)_{r \in \mathcal{R}} \in \mathbb{R}^R_-$ be the dual variables of constraints (51) to (54), respectively. In the PP, a duty d is represented by vector $x_{d,r} = (x_{d,r,i,w,\tau})_{i \in \mathcal{F}, \tau \in \mathcal{S}_i(w), W^{\min}_{i,r} \leq w \leq W^{\max}_{i,r}}$, where $x_{d,r,i,w,\tau}$ is 1, if flight i belongs to duty d with w working stations assigned to flight i at a carousel of type r and start time tuple τ is selected, and 0 otherwise. Auxiliary binary vector $y_{d,r} = (y_{d,r,k,m})_{1 \leq k \leq K, 1 \leq m \leq M}$ is 1 at entry $y_{d,r,k,m}$, if duty d of type r causes a violation of u_k in time interval m, and 0 otherwise. Then, a duty with negative reduced cost is a negative solution of one of the R PP

$$rc_r(x_{d,r}, y_{d,r}) = \min \Bigg\{ \sum_{k=1}^{K} \sum_{m=1}^{M} p_k \cdot y_{d,r,k,m} - \left(\sum_{i \in \mathcal{F}} \sum_{w=W^{\min}_{i,r_c}}^{W^{\max}_{i,r_c}} \sum_{\tau \in \mathcal{S}_i(w)} x_{d,r,i,w,\tau} \cdot \rho_{r,i,w,\tau} + \nu_r \right) \mid d \in \mathcal{D}_r(u^{\text{fix}}) \Bigg\} \tag{55}$$

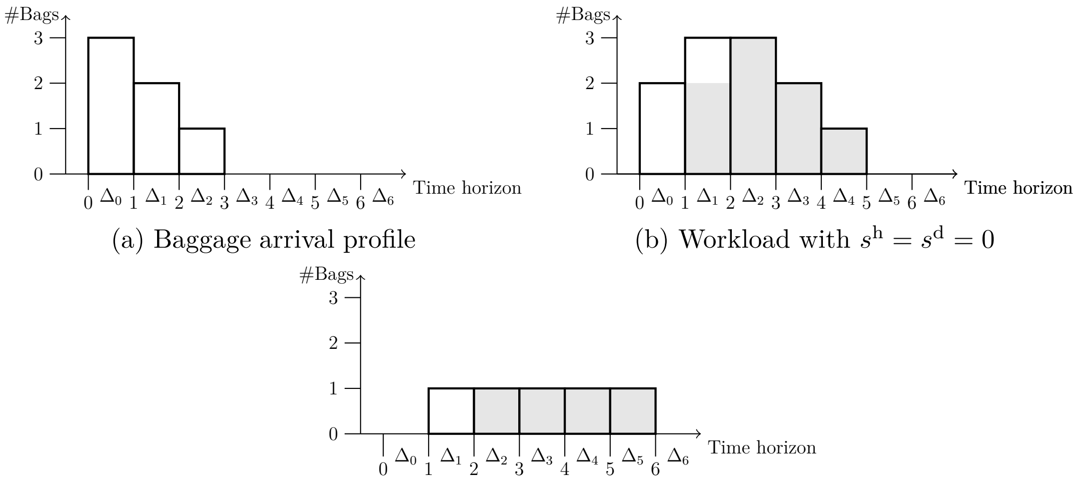
with $\rho_{r,i,w,\tau} = \left(\lambda_i + \sum_{t \in \mathcal{T}} \left(\mu_t \cdot \Phi^i_{t,\tau}\right)\right)$.

To obtain a low penalty value for the threshold violations (see first term of (55)), the PP will only assign few flights to a duty. This makes it harder to provide a positive answer for the NP-hard decision problem whether there is a feasible solution for the OBHP (see Section 3.3.2). To "simplify" the answer, we only consider a subset $\mathcal{U}_m = \{u_{k-n}, \ldots, u_k\}$ of threshold values with $n \geq 0$ in each time interval $1 \leq m \leq M$, leading to the restricted PP (RPP); all threshold values $u_k \in \mathcal{U}_m$ satisfy $u^{\text{ta}} + u_k \leq u^{\text{fix}}$. As the RPP is a generalization of the MULTI-DIMENSIONAL KNAPSACK PROBLEM which is known to be NP-complete (see [43]), we present a LP-heuristic to solve RPP in Section 3.5.4.

3.4.2.2 Flight scheduling problem

Whenever RMP finds a feasible solution, we solve the flight scheduling problem (FSP) in step 8 of Algorithm 1. Given RMP's assignment of flights to carousels, the FSP re-schedules flights' baggage handling periods and re-assigns the working stations to flights.

To illustrate the influence of the re-scheduling let us consider the baggage arrival profile of a flight in Figure 3.4 (a). We assume that the flight is assigned to one working station where we have a loading rate of 1 bag per period. The capacity of the central storage system is 3 bags per period and 1 bag is sent to the carousel per period when the storage depletion starts. If in RMP's solution flight's baggage handling and storage depletion start at time $s^{\mathrm{h}} = s^{\mathrm{d}} = 0$ we obtain the workload depicted in Figure 3.4 (b). If the start of baggage handling and the storage depletion are right-shifted from time 0 to $s^{\mathrm{h}} = 1$ and to $s^{\mathrm{d}} = 2$, respectively, the maximal workload is reduced to 1 bag as illustrated in Figure 3.4 (c).



(a) Baggage arrival profile

(b) Workload with $s^{\mathrm{h}} = s^{\mathrm{d}} = 0$

(c) Workload with $s^{\mathrm{h}} = 1$ and $s^{\mathrm{d}} = 3$

Figure 3.4: Influence of FSP on a given solution; gray shaded are bags which could not be loaded in the previous periods

The FSP has the same number of constraints as the OBHP, and due to Theorem 3.3.3, the FSP is NP-complete. Since, however, flights' assignment is given the number of variables is reduced to $(3 \cdot F \cdot \max_{i \in \mathcal{F}} \{|\mathcal{S}_i|\} + K' \cdot M)$ with $K' \leq K$. Moreover, as only a subset of threshold values of RPP are considered in the objective function and FSP need only to be solved after a new feasible solution is found, we use CPLEX to solve FSP. The solution of FSP yields a new upper bound u^{fix}.

3.4.2.3 Descent scheduling problem

We update the set of threshold values $\mathcal{U}_m$ in steps 9 to 17 of Algorithm 1. First, all threshold values in $\mathcal{U}_m$ leading to a violation of upper bound u^{fix} in each time interval m are removed, i.e. all $u_{k'} \in \mathcal{U}_m$ with $u^{\text{ta}} + u_{k'} > u^{\text{fix}}$. Second, we use the descent scheduling problem (DSP) to compute a minimal descent direction v from upper bound u^{fix} which yields the next utilization values which can be reached by RMP. The smallest possible descent direction $v = \frac{1}{K_R^{\text{ca}}}$ may not be appropriate, due to the irregular arrival of flights' baggage. For example, consider three flights where each flight causes a workload of 4 bags in the same period, independently of the chosen start time tuple. We assume that there are at least 2 carousels with a capacity of 12 bags each. If all three flights are assigned to the same carousel an utilization of 100% occurs. If $u^{\text{fix}} = 1$ is the upper bound for the utilization, the utilization can only be reduced by assigning one flight to another carousel. Therefore, the next possible upper bound would be $\frac{2}{3}$, which results in a descent direction of $\frac{1}{3}$ instead of $\frac{1}{12}$.

Let $\mathcal{T}_m(u^{\text{fix}})$ be the points in time interval m in which utilization u^{fix} is reached; for example we obtain $\mathcal{T}_m(3) = \{1, 2\}$ for the workload of Figure 3.4 (b). In the DSP, decision variable $x_{i,w,\tau}$ is 1, if flight i is planned with w working stations and start time tuple τ. For each period in time interval $1 \leq m \leq M$, $t \in \mathcal{T}_m(u^{\text{fix}})$ and carousel type $r \in \mathcal{R}$, we solve the DSP (DSP(r,t))

$$\text{minimize} \quad z_{\text{DSP}}(r,t) = v \tag{56}$$

$$\text{subject to (39)} - \text{(42)} \qquad \text{restricted to } r, t$$

$$\sum_{i \in \mathcal{F}: S_i^{\text{es}} \leq t < S_i^{\text{e}}} \sum_{w=W_{i,r}^{\min}}^{W_{i,r_c}^{\max}} \sum_{\tau \in \mathcal{S}_i(w)} \Gamma_{t,s}^{i,w} \cdot x_{i,w,\tau} + v = \left\lfloor K_r^{\text{b}} \cdot u^{\text{fix}} \right\rfloor \tag{57}$$

$$1 \leq v \leq \left\lfloor K_r^{\text{b}} \cdot u^{\text{fix}} \right\rfloor \tag{58}$$

$$x_{i,w,\tau} \in \{0,1\} \qquad \forall\, i \in \mathcal{F} : S_i^{\text{es}} \leq t < S_i^{\text{e}},\ W_{i,r}^{\min} \leq w \leq W_{i,r}^{\max},\ \tau \in \mathcal{S}_i(w). \tag{59}$$

Constraints (39) to (42) ensure the resource capacities of the carousels and central storage system for the considered carousel type r and time period t. Constraints (57) determine the maximal descent direction value v. Constraints (58) and (59) set the domains of the variables.

For each time interval m we distinguish two cases when updating threshold value sets $\mathcal{U}_m$ in step 13. Either, there is at least one carousel type r such that DSP(r,t) is feasible for all $t \in \mathcal{T}_m$, or DSP is not feasible for m and any carousel type $r \in \mathcal{R}$. In the first case, let u_k be the largest threshold value with $u^{\text{ta}} + u_k \leq u^{\text{fix}}$. For the next iteration, we include threshold value u_k up to threshold value u_{k-n} with $u_{k-n} > \max_{r\in\mathcal{R}, t\in\mathcal{T}_m(u_{\text{fix}})} \left\{ \lfloor \frac{z_{\text{DSP}}(r,t)}{K_r^{\text{cb}}} \rfloor \right\}$ in $\mathcal{U}_m$. In the second case, we set $\mathcal{U}_m = \{\emptyset\}$, i.e. in time interval m RMP is transfered into a feasibility problem such that the carousels' utilization satisfies upper bound u^{fix}. If there exists no feasible solution for DSP in all time intervals, the objective function can not be further improved and the optimal solution of OBHP is found (see Algorithm 1 steps 15 and 16). As in DSP only a subset of flights are considered we can solve DSP efficiently with CPLEX.

3.5 Implementation details

In this section we present details of our column generation implementation. In Section 3.5.1, a sequential allocation heuristic is presented to produce initial start columns. The heuristic is based on the procedure currently used in practice. To increase the chance of finding a feasible solution for RMP, the variety of duties from two consecutive column generation iterations is increased in § 3.5.2 by means of the Chebyshev cutting-plane method of [64]. To search for a feasible solution in RMP, we apply a primal set-covering heuristic presented in Section 3.5.3. The increased assignment variety, however, also increases the combinatorial complexity of each RPP as more flights can be assigned to a duty. In Section 3.5.4 a greedy based MIP-heuristic for the RPP is presented speeding up each iteration of column generation.

3.5.1 Initial columns

To obtain initial columns for RMP we run a greedy heuristic which uses the expected amount of baggage as measure for the carousel assignment; see also [8] for a discussion of greedy heuristics for baggage handling without considering the baggage arrival process.

The greedy heuristic sequentially calls the following methods (see Appendix A.1 for the pseudo-code):

`order`($\mathcal{F}$): All flights are increasingly ordered according to the latest start time of baggage handling, where the scheduled departure time serves as tie breaker. Values $\bar{w}$ and $\bar{\tau}$ denote the number of assigned working stations and the selected start time tuple, respectively, for flights assigned to a carousel. When flight $i \in \mathcal{F}$ is next in the greedy order, the start time for the baggage handling as well as the storage depletion is set in the middle of the flight's baggage handling time window and the minimal number of working stations is assigned.

`assign`(i): Let $i \in \mathcal{F}$ be the next flight in the order, and $\mathcal{C}^{\text{pos}}$ be the subset of feasible carousel in terms of parking positions and working stations. The flights, already assigned to a carousel $c \in \mathcal{C}$ in previous iterations are denoted by $\mathcal{F}_c^{\text{as}}$. Then, flight i is assigned to the carousel c^* leading to the best leveling for the amount of baggage during flight i's baggage handling, i.e. we select carousel

$$c^* = \arg\min_{c \in \mathcal{C}^{\text{pos}}} \left\{ \sum_{t \in \mathcal{T}} \left(\sum_{j \in \mathcal{F}_c^{\text{as}} : s^h_{\bar{\tau}_j} \leq t < S_i^{\text{e}}} \frac{A_{j,t} + \mathbb{1}_{\left\{ s^{\text{h}}_{\bar{\tau}_i} \leq t < S_i^{\text{e}} \right\}} \cdot A_{i,t}}{K^{\text{cb}}_{r_c}} \right)^2 \right\}. \tag{60}$$

The right term is the sum of utilizations at carousel c over planning period $\mathcal{T}$. As high utilization values have to be avoided, the utilization values are penalized by the quadratic function at any time t (see e.g. Rieck et al. [78]).

`leftShift`(i): If it is not possible to assign a flight i due to a resource conflicts at a carousel, i.e. not enough parking positions or working stations are available and therefore $\mathcal{C}^{\text{pos}} = \{\emptyset\}$, the starting time of the baggage handling and the storage depletion is post-poned by one period and (60) is evaluated again. If the resource conflict at the carousels can not be resolved for a flight or the capacity of the central storage system is violated by the flight, we assign the flights to a dummy carousel with infinite capacity. The dummy duty is inserted into the RMP to guarantee that a solution exists; the use of this duty is penalized in the objective function.

`postProcessing`: We apply a post-processing step in which not assigned working stations are sequentially assigned to those flights causing the highest workload on the carousels.

3.5.2 Chebyshev cutting-plane method

Reduced costs (55) allow a flight $i \in \mathcal{F}$ only to be assigned to a duty if inequality $\lambda_i \geq \sum_{t\in\mathcal{T}} \mu_t \cdot \Phi^i_{t,\tau}$ holds. However, because of dual degeneracy this inequality is often valid only for a small subset of flights which leads to duties with a small flight density. To enforce this inequality for a larger subset of flights we "artificially" increase dual values λ_i by [64]'s cutting-plane method which sets the dual variables to the gravity point of L-RMP's dual polyhedron, the so called Chebyshev center.

Let $\|x\|_2$ be the 2-norm of a vector x, b be the vector of the right hand side of constraints (51) – (53) and be Z^{LB} a lower bound for L-RMP's objective function value. To get the gravity point in the dual polyhedron, consider the Chebyshev centered dual problem

$$\text{maximize} \quad z \tag{61}$$

subject to

$$\sum_{i\in\mathcal{F}} \Theta^{\text{a}}_{d,i} \cdot \lambda_i - \sum_{t\in\mathcal{T}} \Theta^{\text{s}}_{d,t} \cdot \mu_t - \nu_r + \|(\Theta^{\text{a}}_d, \Theta^{\text{s}}_d, 1)\|_2 \cdot z \leq H_d \qquad \forall\, d \in \mathcal{D}'_r(u^{\text{fix}}),\ r \in \mathcal{R} \tag{62}$$

$$-\lambda_i + \alpha^{\text{a}}_i \cdot z \leq 0 \qquad \forall\, i \in \mathcal{F} \tag{63}$$

$$-\mu_t + \alpha^{\text{s}}_t \cdot z \leq 0 \qquad \forall\, t \in \mathcal{T} \tag{64}$$

$$-\sum_{i\in\mathcal{F}} \lambda_i + \sum_{t\in\mathcal{T}} K^{\text{s}}_t \cdot \mu_t + \sum_{r\in\mathcal{R}} C_r \cdot \nu_r + \alpha^{\text{obj}} \cdot \|b\|_2 \cdot z \leq -Z^{\text{LB}} \tag{65}$$

$$\lambda, \mu, \nu, z \geq 0. \tag{66}$$

Constraints (62) measure the distance of a point inside the polyhedron from the hyperplanes. To obtain an increased assignment variety of flights, we extend [64]'s method by constraints (63) and (64), which set the distance for a dual point from the non-negativity hyperplanes of λ and μ to at least $\alpha^{\text{a}} \cdot z$ and $\alpha^{\text{s}} \cdot z$, respectively. As both constraints do not influence objective function (61), proximity parameters α^{a} and α^{s} can be set ≥ 0 without cutting off the optimal solution. Constraint (65) moves the objective function hyperplane from the best known objective function value Z_{LB} to the interior of the polyhedron. [64] use the proximity value $\alpha^{\text{obj}} > 0$ (see Theorem 3 of [64]) to move the point either closer to Z^{LB} by lowering its value or to move the point closer to the current polyhedral describing hyperplanes by increasing α^{obj}.

The primal problem of the Chebyshev centered dual problem is denoted as the Chebyshev centered restricted primal master problem (CRMP) (see Appendix A.2). With the CRMP the column generation principle remains almost the same and the LP bound gives a feasible lower bound for MP (see [64]). When applying column generation, duties with negative reduced cost are added to the linear relaxed CRMP (L-CRMP). If no such duty exists and $z > 0$, lower bound Z^{LB} is updated according to $Z^{\mathrm{LB}} = \sum_{i \in \mathcal{F}} \lambda_i + \sum_{t \in \mathcal{T}} K_t^{\mathrm{s}} \cdot \mu_t + \sum_{r \in \mathcal{R}} C_r \cdot \nu_r$. Otherwise, when $z = 0$, value Z^{LB} yields the lower bound and column generation is stopped.

To increase the assignment variety, we choose α_i^{a} randomly within interval $(0, 1]$ for each flight $i \in \mathcal{F}$ after each column generation iteration, while α_t^{s} is set to 0 for all $t \in \mathcal{T}$, which motivates the dual variables to satisfy inequality $\lambda_i \geq \sum_{t \in \mathcal{T}} \mu_t \cdot \Phi_{t,\tau}^i$. Proximity parameter α^{obj} is increased by an increment after each column generation iteration, as proposed by [64].

3.5.3 Primal set-covering heuristic

To accelerate the search for a feasible solution in RMP, we make use of the framework for a primal heuristic proposed by [56]. In our primal heuristic, a set-covering heuristic (see Algorithm 4 in Appendix A.3) fixes some of the z_d-variables in the L-CRMP to 1 and, hence, some of the duties in order to set up a primal search tree. The nodes of a branch represent these fixed duties. To explore the neighborhood of a partial solution, backtracking as a diversification mechanism is applied in which a tabu list avoids choosing the same duties selected in previous branches. The tabu list at a node contains all children of the node as well as the union of the tabu lists of the ancestor nodes; the tabu list is empty at the root node. A node that is not the first child node of its parent node is explored only if the size of its tabu list is smaller or equal to $maxDiscrepency = 6$ and its tree depth is smaller or equal to $maxDeepth = \lfloor \frac{C}{2} \rfloor$.

Given fixed variables z_d, column generation is applied to seek for a feasible solution by generating appropriate columns for the partial solution. The residual L-CRMP is re-optimized as long as the L-CRMP becomes either feasible or the RPP does not return a column with negative reduced costs. In the latter case, the next branch is examined. If there is no branch left, we proceed with column generation for the next $nextIterations = 2 \cdot C$ iterations until a new search tree is built. Duties used for the previous search tree are forbidden to set up the new search tree. Because feasibility of RMP can be achieved after the addition of a new duty, the set-covering heuristic in Algorithm 4 is applied after each column generation iteration.

3.5.4 Pricing problem heuristic

In the RPP we have to add containers and working stations (items) of flights to a carousel with limited resources (knapsack) over the planning horizon. Hence, the RPP is at least as hard to solve as a MULTI-DIMENSIONAL KNAPSACK PROBLEM which is known to be NP-hard (see [43]). In particular, L-CRMP produces a great number of ρ-values ≥ 0 increasing RPP's computational intractability. As there is no need to solve RPP exactly until the last iteration of column generation, an approximated solution is used which finds good solutions in an acceptable time. We implement a heuristic based on linear programming and adapted the adaptive fixing procedure of Bertsimas and Demir [17] for the multidimensional knapsack problem. The heuristic iteratively rounds variables of the linear programming relaxation of the PP. In our pre-experiments, the procedure yields much better solutions for RPP than any other greedy based algorithms for the multi-dimensional knapsack problem (see Moser et al. [70], Khan et al. [58] and Akbar et al. [3]). When the pricing problem heuristic does not return a duty with negative reduced cost, we solve RPP by means of CPLEX to proof either the optimality of L-CRMP or to generate a new duty.

3.6 Computational study

In this section we report empirical results of the proposed model and solution methodology. For computations we employ an Intel Dual Core 2.8 GHz workstation with 2 GB of RAM memory running on a Windows 7 platform. The mathematical model and algorithms are implemented in JAVA language. LP and MIP problems are solved using CPLEX callable library version 12.4.

In Section 3.6.1 we provide a description of the underlying data. An experimental study in Section 3.6.2 compares the performance of Algorithm 1 with the time-indexed formulation and the standard column generation procedure for RMP. The performance of Algorithm 1 for real-world scenarios is shown in Section 3.6.3.

3.6.1 Data base

All test instances are derived from 2010 data of Terminal 2 of Munich Airport. The planning horizon is set from 03:00 am to 10:40 pm, i.e. $T = 236$ for periods of 5 minutes length. As shown in Figure 3.2 the planning horizon is divided into 3 time intervals, in which the target value for the utilization is set to $u^{\text{ta}} = 0.5$. The threshold values for the deviations from the target value are $\mathcal{U}_m = \{u_1, \ldots, u_{13}\} = \{0.1, 0.2, \ldots, 0.9, 1, 2, 10, 100\}$ in each time interval $1 \leq m \leq 3$. To minimize the utilization in each of the 3 time intervals, we penalize a deviation of u_k with $p_k = k^k$ for $k \in \{1, \ldots, 13\}$. All presented results are rounded after the second decimal place. Next, we provide a description of the baggage arrival process, of airport's infrastructure and the considered outgoing flights.

Arrival process OBHP's solution quality significantly depends on baggage arrival process A_i for each flight $i \in \mathcal{F}$ which yields the workload $\Theta^{i,w}_{\tau,t}$. Planning with the average arrival process underestimates flights' true workload leading to high utilization during the course of the operating day. When we consider the data of Terminal 2 of Munich Airport, the amount of baggage arriving before flight's scheduled departure time shows daily patterns. To obtain a proper estimation for flight i's baggage arrival process A_i for a planning day, we accumulate the historical amount of arrived baggage arriving in 5 minutes intervals before flight i's scheduled departure time. To avoid seasonal dependences (e.g. vacations, holidays, festivals, etc.) the data collection is restricted to maximal 3 month before the planning day. Having the accumulated amount of baggage in each time interval, we derive the 50% (5), 70% (7), 80% (8) and 90% (9) quantile for the estimation of A_i. The quantiles represent different robustness degrees of the obtained solution. The higher the quantile the lower the chance that the amount of arrived baggage per period and, hence, the workload is underestimated. On a planning day, we calculate the average underestimation of the historical arrival process A^{h}_i from the estimated arrival process A_i with $a(i) = \frac{1}{T} \cdot \sum_{t \in \mathcal{T}} \left(A^{\text{h}}_{i,t} - A_{i,t} \right)^+$. Table 3.2 reports the average root mean squared error (RMSE) and the average underestimation ($\bar{a}$) of the historical arrival process of $1,000$ flights. In the table the flights are ordered according to the expected amount of baggage $\mathbb{E}(A)$. The results show, the higher the quatile is, the lower is the underestimation of the amount of baggage.

	50%			70%			80%			90%		
$\mathbb{E}(A)$	< 50	$[50, 100]$	> 100	< 50	$[50, 100]$	> 100	< 50	$[50, 100]$	> 100	< 50	$[50, 100]$	> 100
RMSE	0.58	1.03	1.72	0.60	1.03	1.72	0.68	1.11	1.86	0.91	1.42	2.37
$\bar{a}$	0.10	0.19	0.34	0.07	0.15	0.27	0.06	0.12	0.21	0.04	0.08	0.14

Table 3.2: Average RMSE and the average $a(i)$-values of 1,000 flights

Infrastructure The layout for the handling facilities is based on the one of Terminal 2 of Munich Airport. There, we have 22 carousels and 3 carousel types with different conveyor belt capacities, number of parking positions and working stations (see Table 3.3 (a)). The capacity of the central storage is bounded by 3,500 bags, while the storage depletion rate is 19 bags per 5 minute. Each working station has a working rate of 8 bags per 5 minute.

	Capacity		
Type r	K_r^{pp}	K_r^{ws}	K_r^{ca}
1	8	4	20
2	12	4	25
3	24	6	40

(a) Carousels

	Handling period (min)	
$\mathbb{E}(A)$	Min	Max
< 50	30	60
$[50, 100]$	60	120
> 100	120	180

(b) Handling periods

Table 3.3: Carousels (a) and flights' handling periods in minutes (min) (b)

Outgoing flights For each flight the time window for the start of the baggage handling is derived from the minimal and maximal duration for the baggage handling periods which depends on flight's expected amount of baggage (see Table 3.3 (b)). The scheduled time of departure of the flight as well as the number of containers required for each flight is taken from the historical data.

3.6.2 Experimental study

We denote the test instances with I-$n(q)$ where n stands for the instance number and $q \in \{5, 7, 8, 9\}$ represents the used quantile to estimate flights' baggage arrival process. For the generation of test instances I-1(q) to I-8(q), we systematically vary the number of flights and the number of handling facilities. The flights are randomly selected from a pool of 413 different flights for each instance. For instances I-1(q) to I-5(q) we consider

10 carousels of type 2 for instances I-6(q) to I-8(q) we additionally consider 7 carousels of type 1. For each instance, Table 3.4 shows in row "$C(r)$" the number of carousels and the corresponding type (r) as defined in Table 3.3 (a). As indicator for the hardness of an instance, we consider the included BIN-PACKING PROBLEM in the OBHP in which we have to find a feasible assignment and schedules of flights to carousels (see Section 3.3.2). Row "LB" of Table 3.4 shows the LP-bound for the minimal number of required carousels for each instance. The last row "Pro. 1/2" gives the average percentage reduction of the start time tuple sets due to Proposition 3.3.4 and 3.3.5.

Instance	I-1(q)	I-2(q)	I-3(q)	I-4(q)	I-5(q)	I-6(q)	I-7(q)	I-8(q)
F	25	50	100	150	200	250	300	413
$C(r)$	10(2)	10(2)	10(2)	7(1)/10(2)	7(1)/10(2)	7(1)/10(2)	7(1)/10(2)	7(1)/10(2)
LB	1	1.89	3.83	4.67	6.33	8.33	12.72	14.17
Pro. 1/2	40.28/14.63	39.88/12.03	42.49/12	42.64/12.24	42.33/11.77	41.16/12.23	42.37/11.97	44.17/12.40

Table 3.4: LP-bound for the number of required carousels and average reduction of the start time tuple sets due to Propositions 3.3.4 and 3.3.5 in %

Table 3.5 shows the results of the instances I-1(q) to I-4(q) where we compare the column generation of Algorithm 1 (Alg1-CG) with the standard column generation implementation for RMP (SCG) in which the PP and not the RPP is considered meaning that all threshold values are considered in PP (see Section 3.4.2.1) and the time-indexed formulation (TIF) (38)–(46). In both column generation implementations we make no use of the Chebyshev-center cutting plane method and as soon as the LP-bound is reached the procedures terminate. During the column generation, the set-covering heuristic of Section 3.5.3 searches for feasible solutions after a new column is added. Columns "Cols" and "Time (s)" report the total number of generated columns and the computing time (in seconds) until the LP-bound is reached. From the best solution found we report in column "u^*" the highest utilization across all time intervals. If the LP-bound is not reached within 1 hour, the computation time until the best utilization obtained in column "u^*" is reported in column "Time (s)". If the LP-bound for Algorithm 1 or RMP's standard column generation implementation is not reached we indicate the test instances with "†" and "‡", respectively. The LP-bound of the column generation and the time-indexed formulation is reported in columns "LP-CG" and "LP", respectively. The number of updates for RPP's threshold value sets is shown in column "Upd". The results reveal that the column generation of Algorithm 1 requires not only less columns than the column generation for RMP to reach the LP-solution but also generates better columns in terms of finding

	Alg1-CG				SCG				TIF		
Inst	u^*	Time (s)	Cols	Upd	u^*	Time (s)	Cols	LP-CG	u^*	Time (s)	LP
I-1(5)	0.18	1.53	28	2	0.36	148.37	1878	$4.8 \cdot 10^{-3}$	0.18	12.42	$4.8 \cdot 10^{-3}$
I-1(7)	0.18	3.51	50	2	0.26	113.89	1814	$4.2 \cdot 10^{-3}$	0.18	12.80	$4.2 \cdot 10^{-3}$
I-1(8)	0.24	4.07	28	2	0.3	106.62	1886	$8.2 \cdot 10^{-3}$	0.24	4.98	$8.2 \cdot 10^{-3}$
I-1(9)‡	0.1	3.34	50	3	0.74	8.86	258	$6.2 \cdot 10^{-3}$	5	0.1	$2.2 \cdot 10^{-3}$
I-2(5)	0.22	3.94	142	3	0.5	507.58	3387	$8.8 \cdot 10^{-3}$	0.22	66.26	$6.8 \cdot 10^{-3}$
I-2(7)	0.22	4.42	105	2	0.72	405.37	3314	$9.3 \cdot 10^{-3}$	0.22	55.64	$6.2 \cdot 10^{-3}$
I-2(8)‡	0.26	5.18	106	2	1.18	26.86	101	$7 \cdot 10^{-3}$	0.26	310.24	$7 \cdot 10^{-3}$
I-2(9)	0.26	6.56	102	2	1.9	320.19	2306	0.03	0.26	280.69	0.01
I-3(5)	0.22	35.18	236	3	1.24	2982.78	5919	$9.8 \cdot 10^{-3}$	0.22	560.02	$6.8 \cdot 10^{-3}$
I-3(7)‡	0.2	15.66	201	2	0.34	292.92	330	$8.2 \cdot 10^{-3}$	0.2	724.19	$5 \cdot 10^{-3}$
I-3(8)‡	0.26	19.57	371	4	1.66	139.34	266	0.01	0.26	644.36	$9.6 \cdot 10^{-3}$
I-3(9)‡	0.2	32.68	529	6	2.14	168.74	258	0.01	0.2	655.37	$7.6 \cdot 10^{-3}$
I-4(5)‡	0.22	48.60	402	3	0.62	352.30	432	$8.5 \cdot 10^{-3}$	0.22	1434.42	$5.4 \cdot 10^{-3}$
I-4(7)‡	0.24	30.98	656	5	1.02	161.87	332	0.03	0.24	1463.66	0.01
I-4(8)‡	0.24	44.88	476	5	0.78	570.27	683	$8.2 \cdot 10^{-3}$	0.24	1523.49	$8.2 \cdot 10^{-3}$
I-4(9)	0.28	57.87	878	7	2.32	2794.70	5274	0.06	0.28	1553.45	0.03

Table 3.5: Comparison between the column generation of Algorithm 1 (Alg1-CG), a standard column generation implementation for RMP (SCG) and the time-indexed formulation (TIF)

a feasible solution with the set-covering heuristic. All found solutions for Algorithm 1 are optimal as the target value 0.5 is reached. In the standard column generation for RMP the quality of the generated duties is worse, in particular at the beginning of the column generation, due to the sparse density of flights in the duties. So, the standard column generation implementation for RMP reaches the LP-bound only in half of the instances and, hence, terminates. The LP-bound for column generation turned out to be equal with the LP-bound for the time-indexed formulation. The gap between the LP-bound and the optimal solution varied between 90% and 97%.

Instance	I-10(q)	I-11(q)	I-12(q)	I-13(q)	I-14(q)	I-15(q)	I-16(q)
F	436	389	413	386	440	331	353
LB	19.25	18.5	19	18	18.5	17.5	18
Pro. 1/2	28.2/9.12	37.1/12.18	36.28/12.22	34.48/12.41	38.06/12.18	33.31/10.9	24.64/0.48

Table 3.6: Initial lower bound and the average column reduction of the arrival matrices due to Proposition 3.3.4 and 3.3.5 in per cent for the real-world test instances I-10(q) to I-16(q)

In the second study, shown in Table 3.7, we run Algorithm 1 with (Alg1-CCP) and without (Alg1-NCCP) the Chebyshev cutting-plane method. If the LP-bound is used we branch according to the rule described in Section 3.4. Instances not solved to optimality with Alg and Alg1-CCP are indicated with "†" and "‡", respectively. In column "Nodes" the number of visited nodes is reported. Column "∅Cols" are the average number of generated columns in each node. We observe that using Chebyshev cutting-plane reduces

	Alg1-NCCP				Alg1-CCP			
Inst	u^*	Time (s)	∅Cols	Nodes	u^*	Time (s)	∅Cols	Nodes
I-5(5)	0.22	96.11	824	0	0.22	156.62	392	0
I-5(7)	0.24	98.19	1105	0	0.24	83.9	331	0
I-5(8)	0.26	82.93	906	0	0.26	295.33	604	0
I-5(9)	0.28	112.18	843	0	0.28	186.72	686	0
I-6(5)	0.22	135.77	1214	0	0.22	141.06	471	0
I-6(7)	0.24	145.87	942	0	0.24	169.65	442	0
I-6(8)	0.26	173.85	1267	0	0.26	307.52	836	0
I-6(9)	0.58	172.46	878	0	0.58	169.44	406	0
I-7(5)	0.22	224.99	710	0	0.22	1642.21	972	0
I-7(7)	0.24	262.96	629	0	0.22	479.23	677	0
I-7(8)	0.26	278.44	560	0	0.26	607.55	697	0
I-7(9)†‡	0.58	297.56	499	104	0.58	417.54	210.2	91
I-8(5)	0.22	449.27	919	0	0.22	360.68	733	0
I-8(7)	0.24	349.64	668	0	0.24	227.73	1184	0
I-8(8)	0.24	1245.85	1014	0	0.24	446.48	905	0
I-8(9))†‡	0.96	436.71	613	103	0.7	280	376	90
I-9(5)	0.22	732.78	877	100	0.22	586.08	978	20
I-9(7)	0.24	3179.72	1049	83	0.24	614.80	933	64
I-9(8)†	0.56	3571.46	988	95	0.26	2772.77	1747	87
I-9(9)†‡	0.58	1753.24	734	100	0.58	1417.95	1053	131

Table 3.7: Comparison between Algorithm 1 without (Alg1-NCCP) and with (Alg1-CCP) acceleration for instances I-5(q) to I-9(q) with $q \in \{5, 7, 8, 9\}$

the number of examined nodes due to the increased assignment variety during the first run of column generation which increases also the chance of finding a feasible solution with the set-covering heuristic 4. The improved column quality can be also seen by the number of generated columns in algorithm Alg1-CCP, which is less than in Alg1-NCCP. However, in some instances the time per column generation in Alg1-CCP is higher than in Alg1-NCCP which is due to the higher number of dual variables which have to be considered in the RPP.

3.6.3 Real-world study

For the real-world instances we randomly selected 7 arbitrary planning days from our historical data. Table 3.6 show information for the real-world instances. In each instance we use 4 carousels of type 4, 17 carousels of type 2, and 1 carousel of type 3 (see Table 3.3 (a)). The test instances are solved with Algorithm 1 where all the implementation details of Section 3.5 are incorporated.

Table 3.8 shows the results of the real-world instances. The greedy heuristic of Section 3.5.1 denoted by “Heu” serves as benchmark for the solutions. If the heuristic does not find a feasible solutions the value is labeled with “+”. The results of the subparts of Algorithm 1 are presented in columns 3 to 8. The average improvement of the utilization in % when calling FSP is presented in column “∅ Impr”. Column “∅ Time (s)” shows RMP’s average computing time between two calls of FSP and DSP; for FSP column “∅ Time (s)” gives the average computing time over all calls. Due to reason of practicality Algorithm 1 is aborted after a running time of 8 hours. For all test instances labeled with “†”, optimality could not be proved. In the last two columns we report the results of a simulation for the outbound baggage handling in which we implemented the solution obtained by the heuristic and Algorithm 1. The simulation uses the historical arrival process of baggage for the flights of the planning day represented by the test instance. Column “Heu” and “Alg 1” show the maximal utilization obtained in the simulation during the course of the planning day.

Table 3.8 shows that 5 of 21 of the real-world instances could be optimally solved by Algorithm 1. The estimated maximal utilization obtained by Algorithm 1 always outperforms the estimated maximal utilization of the heuristic. Considering the evaluation of the solution, the improvement becomes even more significant. We can also see that the solution quality of Algorithm 1 becomes more robust the higher the used quantile for the estimation of the baggage arrival process is. For the heuristic we could not identify such a strong correspondence between used quantile and solution quality. For most cases of our test-instances, the 80%-quantile leads to the best solution with an maximal utilization near 1.

			RMP			FSP		Simulation	
Inst	Heu	Upd	u^*	∅Time (s)	∅ Cols	∅Impr	∅Time (s)	Heu	Alg 1
I-10(5)	1.36	1	0.8	15.51	75	68.25	6.87	3.23	2.21
I-10(7)	2.72	7	0.96	54.75	291.71	15.77	3.26	3.45	1.14
I-10(8)†	3.08	12	1.24	747.77	693.66	8.81	4.60	2.93	0.90
I-10(9)	6.96	12	0.90	359.84	509.33	11.00	4	2.21	0.87
I-11(5)†	2.8	7	1.05	626.77	523.29	9.75	3.57	4.97	2.44
I-11(7)†	3.08	7	1.05	2775.96	443.14	19.00	2.95	4.47	1.23
I-11(8)†	3.09	7	1.12	4435.82	1731.2	22.57	3.12	3.87	0.89
I-11(9)†	8.24	10	2.18	1498.51	834.6	9.65	4.31	3.98	0.99
I-12(5)†	2.22	4	1.1	336.30	980.5	26.51	2.06	4.13	2.42
I-12(7)†	3.12	7	1.32	1382.55	1145	14.82	2.47	4.60	1.22
I-12(8)†	4.16	7	1.32	1177.96	1360.71	14.78	2.98	4.78	0.53
I-12(9)†	7.52	12	2.25	1847.16	1043.92	9.34	4.84	4.32	0.84
I-13(5)†	4.1$^+$	3	1.15	253.01	926.67	28	2.89	3.23	2.12
I-13(7)†	4.3	4	1.2	78.52	230.75	23.13	2.18	3.12	1.01
I-13(8)†	4.95$^+$	3	1.33	182.06	357.67	41.16	3.11	3.02	1.28
I-13(9)†	7.35	7	1.85	4803.03	1309.57	14	4.18	2.89	1.07
I-14(5)	1.72	5	0.68	96.68	411.2	21.14	1.86	2.12	1.34
I-14(7)	3.08$^+$	4	0.88	308.33	871.75	24.13	2.14	2.28	1.13
I-14(8)†	2.72	5	0.96	923.53	1334	22.92	3.67	2.55	0.64
I-14(9)†	8.08	8	0.96	312.79	426.38	19.04	5.81	2.10	0.87
I-15(5)†	3.16$^+$	3	1.15	2937.81	1409.66	32.57	2.12	2.12	1.87
I-15(7)†	3.44	4	1.2	5435.51	1232.75	25.36	2.33	1.98	1.15
I-15(8)†	3.96	6	1.3	4882.72	1696.71	14.82	2.91	1.80	0.78
I-15(9)†	8.32	6	2.45	406.98	393.5	13.56	4.53	1.76	0.99
I-16(5)†	3$^+$	4	1.25	632.31	1081.67	20.42	1.92	3.02	1.67
I-16(7)†	3.44	6	1.92	754.45	1065.5	14.69	2.32	3.14	1.10
I-16(8)†	3.96	7	1.92	4338.34	2220.71	15.31	3.56	2.98	0.83
I-16(9)†	8.10$^+$	4	3.48	6903.30	2450.8	17.16	3.55	2.78	1.14

Table 3.8: Computational results for real-world instances

3.7 Conclusion

We presented a time indexed mathematical model for assigning flights to handling facilities and scheduling their baggage handling period. The problem is of high practical relevance. We analyzed the problem complexity and showed dominance criterion to strengthen the solution space. To receive a balanced utilization during the course of the day, we presented a “guided” column generation framework. In our computational study we showed that the presented column generation framework leads to a significant reduction of computer times in comparison to the time indexed formulation or a standard column generation framework.

Chapter 4

Inbound baggage handling

"Es ist nicht genug zu wissen – man muss auch anwenden. Es ist nicht genug zu wollen – man muss auch tun."

– Johann Wolfgang von Goethe, 1749 – 1831

4.1 Introduction

The number of airline passengers is rapidly increasing at about 5.9% per year which forces airports to use their existing infrastructure more efficiently. At the same time, the high competition between airports demands higher quality services to satisfy passengers. One crucial task for airports impacting both these issues is the baggage claim. On the one hand, expanding the baggage claim area is cost-intensive and limited by airport's infrastructure. On the other hand, as the former chairman of Eastern Airlines, Frank Borman said: "baggage claim is the last chance to disappoint the customer" (see Robusté 79). Passengers service quality could be increased, for example, by shortening walking distances from flights parking position to the baggage carousel, or by reducing waiting times of passengers at the baggage carousels (see Martel and Senviratne 69 and Correia and Wirasinghe 27).

Baggage brought-in by an incoming flight and claimed by passengers is called inbound baggage. The inbound baggage handling process starts as soon as an incoming flight reaches its parking position. First, inbound baggage containers are unloaded from the aircraft. Then, a baggage tug tows the containers to one of several infeed station. More than one trip might be necessary, depending on the number of containers. At the infeed station, the baggage is transferred into the baggage handling system (BHS) which is an automated baggage transportation system that delivers bags to baggage carousels. The baggage is routed to one of the available baggage carousels where it is retrieved by the passengers. Depending on the layout of the BHS, one or more baggage carousels can be reached from an infeed station.

For the planning of the inbound baggage handling process, two assignment decisions have to be made by the airport, one for the airside and one for the landside. At the airside, the infeed process has to be planned where each incoming flight carrying inbound baggage is assigned to an infeed station. At the landside, the claiming process has to be planned where the baggage carousel, to which the bags are transferred, has to be selected. Both assignment decisions on the airside and landside are interrelated which can be illustrated by the following example: If the infeed process of a container is delayed, the arrival of the corresponding bags at the baggage carousel will also be delayed, which in turn might increase passenger waiting times. On the other hand, if too many flights are served at a baggage carousel simultaneously, the baggage carousel will be at full capacity. The result is that no further bags can be sent to the baggage carousel, which might lead to a congestion at an infeed station. As we will see in this chapter, operating the inbound baggage handling is a complex task.

Usually, experienced planners, called dispatchers, decide on the infeed stations and baggage carousels assigned to flights. Due to the combinatorial complexity of the planning problem, the resulting assignments often leads to unbalanced usage of the baggage carousels or a poor level of service quality for passengers. As the inbound baggage handling is a dynamic and volatile process with its dependency on the arrival time of flights, it is necessary to update the plan over the course of a day as soon as new flight information is received. However, the dispatcher updates his planning at rather irregular time intervals throughout the day of operation. Hence, it happens quiet frequently that the plan of the dispatcher is based on outdated information which further lowers the quality of the assignment.

The literature on inbound baggage handling is generally of analytical nature. [15] published the first analytical work in which he proposes a queuing model and shows that passengers arrive in predictable patterns in the baggage claim hall. [19] presents a predictive model for computing the maximal queue length of passengers and baggage at claim areas. Based on their results, [44] derive a model to measure the length of passengers' waiting times when waiting for baggage at different types of claim facilities. In all proposed queuing models, the effect of a given assignment of flights to baggage carousels for different carousel layouts is analyzed. In contrast, in this chapter we develop a mathematical model to assign incoming flights to infeed stations and baggage carousels and to set the infeed order of flights' baggage tugs arriving at the same time at the infeed station in order to balance the load across baggage carousels and to guarantee a high level of service for passengers.

The assumptions of [69] that the level of service quality of an airport can be improved by reducing passengers' waiting time or passenger density at baggage carousels are confirmed by a survey of [27]. It should be noted that [102] observe that the perceived waiting time is considerably higher than the real waiting time at baggage carousels. Further work concerning inbound baggage handling is published by [29] as well as [46] who study the design of the baggage claim hall.

In this is chapter we present a mathematical model formulation and a heuristic for the inbound baggage handling process at airports. The model is generic and thus can be applied to many airport infrastructures. The objective is to balance the usage of the baggage carousels and to maximize the level of service for passengers. The latter objective is kept general as passengers level of service depends on the given airport infrastructure.

The remainder of this chapter is structured as follows: Section 4.2 provides a detailed problem description of the inbound baggage handling process in which the required notation and assumptions for the mathematical model are presented. The mixed-integer model is formulated in Section 4.3. Further, its problem complexity is established and structural properties are discussed in Section 4.3. As solution we present an efficient hybrid greedy randomized adaptive search procedure (GRASP) with a guided fast local search (GFLS) and path-relinking, which we will call hybrid HGGLS (hybrid GRASP-GFLS) in the following. To the best of our knowledge, this is the first hybridization of GRASP and GFLS. Our computational tests show, that HGGLS leads to high quality solutions for short running time. The proposed solution procedure has been implemented at a major European Airport, where we also conducted a survey to measure which factors influence passengers' level of service. In Section 4.4 and Section 4.5 the HGGLS heuristic and issues regarding the practical implementation are discussed. Moreover, we discuss features which we incorporate into the practical implementation. For the evaluation of our model formulation and the hybrid heuristic, we use real-world data and compare the results of the approach presented in a real-world simulation study with the airport's planning procedure. The solution obtained by our heuristic, embedded in a rolling planning framework, leads to a significant improvement of the carousel utilization with an increase service quality by reducing waiting times by about 3 minutes on average. The computational study is presented in Section 4.6. Section 4.7 concludes this chapter by summarizing the results.

4.2 Problem description

As stated above, the inbound baggage handling process is made up of two subprocesses: the baggage infeed process at the airport's airside and the claiming process at the airport's landside. In the following, we will describe both processes.

Infeed process For each flight, one baggage tug is available to tow the flight's containers to exactly one infeed station, where workers load the bags from the container onto the infeed station's conveyor belt. A baggage tug can tow up to 3 containers at a time. If there are more than 3 containers, the baggage tug has to make several trips from the airplane to the infeed station and back. As the driver of the baggage tug also unloads bags at the infeed station, the completion time of the unloading process determines the start time of the subsequent trip necessary to handle the same flight.

At an infeed station, only one container can be handled at a time. Multiple containers of the same flight are handled sequentially. If there is more than one baggage container of different flights in front of the infeed station, the 'first come, first serve' (FCFS) discipline is applied in order to sequence the containers. Only when two trips arrive in the same time period (having a length of one minute) a sequence of the containers of the two flights has to be determined. We distinguish between direct and remote infeed stations. A direct infeed station is located close to a baggage carousel and it is connected to one unique carousel, while a remote infeed station is connected to one or more baggage carousels (see Figure 4.1). As direct infeed stations are very close to a terminal's baggage carousels they are often separated from the baggage carousel by a wall only. In contrast, remote infeed stations are often located in a building on the apron, further away from the terminal building.

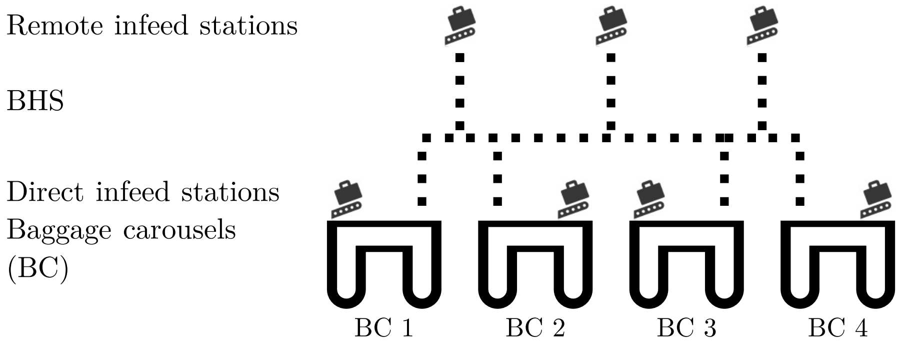


Figure 4.1: Infrastructure for the inbound process

If a bag is fed-in at a remote infeed station, it is transported through the BHS's conveyor belt network to the assigned baggage carousel. At a direct infeed station, bags are directly transported to the corresponding baggage carousel through a conveyor belt link. Such a direct access to a baggage carousel leads to shorter transportation times for bags than for a remote access. For example, at a major European Airport, a bag requires less than 5 seconds to reach the carousel's conveyor belt at a direct infeed station, while a bag fed-in at a remote infeed station requires 6 minutes on average to reach a baggage carousel.

The capacity of a carousel is defined as the number of bags which can be placed on the carousel's conveyor belt at a time. The number of bags on a baggage carousel relative to the carousel's capacity yields the utilization of a baggage carousel. Once the utilization of a baggage carousel reaches 100%, no additional bags can be placed. A fully loaded baggage carousel leads to congestion at the associated direct infeed station, and the unloading process of baggage from containers to the infeed station has to be halted until the baggage carousel's utlization falls below 100%. If a remote infeed station is used and the baggage carousel is utilized to full capacity, the BHS serves as temporary storage, which ensures that the unloading process at the infeed stations continues without interruption. However, using the BHS as temporal storage is not desired as it might affect other baggage flows within the BHS. To avoid congestion at infeed stations or the usage of the BHS as temporary storage, the primary goal for the inbound baggage handling from an operational perspective is to obtain a balanced utilization at the baggage carousels.

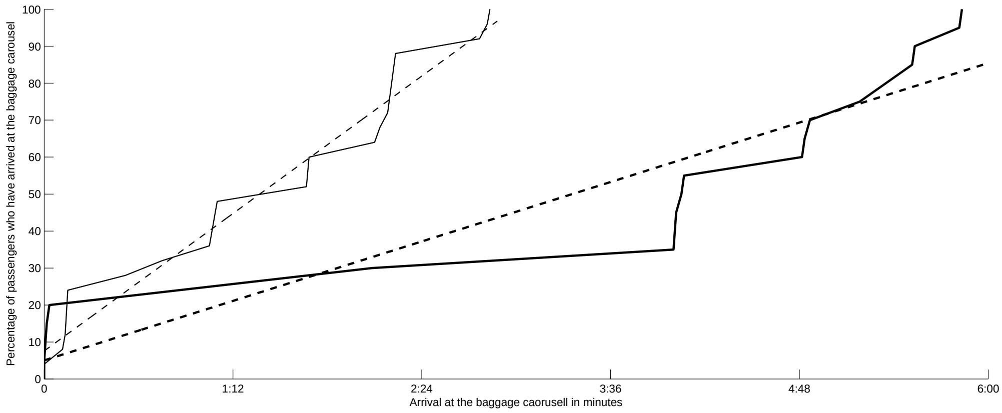


Figure 4.2: Passenger arrival at a baggage carousel for a domestic (thin) and transcontinental flight (bold)

Claiming process While a flight's infeed process is taking place, the passengers disembark from the aircraft. Depending on the flight's parking position, they enter the terminal either through an air bridge or by using a shuttle bus. The architecture of the airport as well as the origin of a flight influence the passenger arrival at the baggage carousel. For example, passengers from transcontinental flights need to pass through immigration and customs which is not the case for domestic flights. Figure 4.2 shows a typical cumulative arrival diagram of passengers at a baggage carousel for a domestic and a transcontinental flight as it can be observed at a major European Airport. The dashed-lines represent a linear approximation of the arrival function. The goodness of fit in terms of the adjusted-R^2 is on average 0.88 for transcontinental flights and 0.97 for domestic flights. When a passenger has arrived at the baggage carousel he immediately picks up his bags once they are on the conveyor belt. After retrieving all of his bags, the passenger leaves the baggage claim hall. The number of simultaneously handled flights at a baggage carousel is restricted by the number of flights which can be presented on the baggage carousel's display. A flight's service time on a baggage carousel is displayed as soon as the flight reaches its parking position ("goes on-block") and remains until all bags have been retrieved. To avoid confusing passengers, it is common practice at airports to assign a flight to only one baggage carousel (see [79]).

4.3 Mathematical model

In this section we present a model for the inbound baggage handling problem (IBHP). We start by providing notations and assumptions in Section 4.3.1 before we present the model formulation of the IBHP in Section 4.3.2. Section 4.3.3 discusses problem characteristics and establishes the IBHP's complexity.

4.3.1 Notation and assumptions

In the following we distinguish between sets and parameters which are immediately given by the problem statement ("Given sets and parameters") and parameters which are derived by preprocessing from these given parameters ("Preprocessed parameters").

Given sets and parameters The sets and parameters used throughout the chapter are summarized in Table 4.1 and 4.2. All time points t in the discrete planning horizon $\mathcal{T} = \{t_0, \ldots, t_T\}$ are equally spread across time and each time $t_k \in \mathcal{T}$ is connected to a unique time interval $[t_k, t_{k+1}[$ for $k = 0, \ldots, T-1$. In the following we will speak of time point t_k when we refer to the beginning of time interval $[t_k, t_{k+1}[$, and we speak of time period t_k when we refer to the time interval.

Sets	Description
$\mathcal{T}$	Discrete planning horizon
$\mathcal{F}$	Flights with inbound baggage
$\mathcal{L}$	Set of trips
$\mathcal{L}_i$	Set of trips for flight i
$\mathcal{L}^{\mathrm{d}}$	Set of trips including dummy trips
$\mathcal{E}$	Infeed stations
$\mathcal{E}_c$	Infeed stations connected to baggage carousel c
$\mathcal{C}$	Baggage carousels
$\mathcal{C}_e$	Carousels accessible from infeed station e

Table 4.1: Sets for the IBHP

The inbound baggage of a flight is transported to one infeed station $e \in \mathcal{E} = \{1, \ldots, E\}$, which has access to one or several baggage carousels $c \in \mathcal{C} = \{1, \ldots, C\}$ in the baggage claim area. The set of baggage carousels reachable from infeed station e is denoted by $\mathcal{C}_e \subseteq \mathcal{C}$, while all infeed stations in $\mathcal{E}_c \subseteq \mathcal{E}$ reach baggage carousel c. Tuple $(e, c) \in \mathcal{E} \times \mathcal{C}_e$ indicates a potential assignment of an infeed station e with one of its corresponding carousels c. Carousel c's conveyor belt capacity and the number of flights which can be shown on the display are given by K_c^{cb} and K_c^{di}, respectively.

The set of arriving flights carrying inbound baggage is denoted by $\mathcal{F} = \{1, \ldots, F\}$. For flight $i \in \mathcal{F}$, P_i passengers carry at least one inbound bag; passengers or flights with no inbound baggage are not the focus of inbound baggage handling and therefore are not taken into account. With the maximal number of bags N_i carried by one passenger, the percentage of passengers of flight i carrying $1 \leq n \leq N_i$ bags is given by $p_i(n)$.

Each flight i requires a predefined number of trips L_i to transport all containers to the assigned infeed station. The set of trips for flight i is denoted by $\mathcal{L}_i = \{1_i, \ldots, L_i\}$, where 1_i and L_i are the first and last trip, respectively; the set of trips for all flights is denoted by $\mathcal{L}$. For the number of baggage transported in trip l, we assume that it always carries the maximal number of the remaining baggage not transported in previous trips such that the trip capacity is not violated. Index i_l denotes the flight corresponding to trip l.

The infeed process of flight i has to start as soon as the assigned infeed station becomes available. An "artificial" delay of the infeed process for each trip is not permitted and does not occur in practice. After flight i's on-block time S_i^{ob} a sequence of sub-processes determines the earliest possible infeed start time $S_{1_i,e}^{\text{es}} \in \mathcal{T}$ of flight i' first trip 1_i at infeed station e. More precisely, the earliest infeed time is the sum of the offset to unload the containers from flight i's cargo hold, duration Δ^{pl} to place the containers on the baggage tug and duration $\Delta_{i,e}^{\text{trip}}$ to transport containers from flight i's parking position to infeed station e. To avoid a late infeed, we also set a latest possible infeed $S_{1_i,e}^{\text{ls}} \in \mathcal{T}$ which is 30 minutes after the earliest infeed $S_{1_i,e}^{\text{es}}$. The time windows $\left[S_{l_i,e}^{\text{es}}, S_{l_i,e}^{\text{ls}}\right]$ for the earliest and latest infeed time of flight i's succeeding trips $l_i \in \mathcal{L}_i \setminus \{1_i\}$ result from infeed start $s \in \left[S_{l_i-1,e}^{\text{es}}, S_{l_i-1,e}^{\text{ls}}\right]$ and infeed duration $\Delta_{l_i-1}^{\text{inf}}$ of the previous trip $l_i - 1$ plus duration $2 \cdot \Delta_{i,e}^{\text{trip}} + \Delta^{\text{pl}}$ to transfer the next containers to the infeed station. As soon as the infeed process of trip l starts it takes $\Delta_{e,c}^{\text{dur}}$ periods to transfer a bag from infeed station e to carousel c.

At infeed station e, relation $(\mathcal{L}, \preceq_e)$ defines the sequence order or infeed order for two trips $l, h \in \mathcal{L}$ with $l \preceq_e h$ if trip l can arrive at infeed station e before trip h. If $l \preceq_e h$ and $h \preceq_e l$, i.e. trip l can arrive at infeed station e before trip l and vice versa, we write $l \cong_e h$. Hence, for the first trips 1_i and 1_j of flights $i, j \in \mathcal{F}$ we have $1_i \preceq_e 1_j$ iff $S_{1_i,e}^{\text{es}} \leq S_{1_j,e}^{\text{es}}$, as flight i's first trip arrives at infeed station e before flight j's first trip, and we have $1_i \cong_e 1_j$ iff both trips arrive in the same period. For all succeeding trips $l_i \in \mathcal{L}_i \backslash \{1_i\}$ of flight i and for an arbitrary trip $h_j \in \mathcal{L}_j$ of a flight j with $i \neq j$ we have $l_i \cong_e h_j$ iff the time windows of trips l_i and h_j overlap, otherwise either $l_i \preceq_e h$ or $h \preceq_e l_i$ holds.

Parameters	**Description**
$A_{i,c}$	Arrival time of the first passenger of flight i at baggage carousel c
B_l	Number of inbound bags of trip l
B_i^{tot}	Total number of inbound bags for flight i
$\left[S_{l,e}^{\text{es}}, S_{l,e}^{\text{ls}}\right]$	Time window to start handling of trip l at infeed station e
S_i^{ob}	On block time of flight i
N_i	Maximal number of bags carried by one passenger of flight i
P_i	Number of passengers of flight i
$p_i(n)$	Percentage of passengers of flight i carrying n inbound bags
$\Delta_{e,c}^{\text{dur}}$	Duration to transfer a bag from infeed station e to baggage carousel c
K_c^{cb}	Conveyor belt capacity of baggage carousel c
K_c^{di}	Display capacity of baggage carousel c
$\Delta_{i,e}^{\text{trip}}$	Duration to drive from flight i's parking position to infeed station e
Δ^{pl}	Duration to place containers on a baggage tug
μ^{w}	Baggage loading rate of workers at an infeed station
μ_i^{p}	Arrival rate of passengers of flight i

Table 4.2: Parameters for the IBHP

Depending of flight i's parking position and by assuming constant walking speed of passengers the first passenger arrives at carousel c at $A_{i,c}$. Based on the results presented in Figure 4.2, we approximate the arrival process of passengers at the baggage carousel by arrival rate μ_i^{p} once the first passenger has arrived at the baggage carousel at $A_{i,c}$ (see Ghobrial et al. 44).

Preprocessed parameters Given the sets and parameters introduced above, we derive the parameters presented in Table 4.3. In the following, let flight i be assigned to tuple (e, c) and let the infeed process for flight i's trip $l \in \mathcal{L}_i$ start at time $s \in \left[S_{l,e}^{\text{es}}, S_{l,e}^{\text{ls}}\right]$.

- Given the number of bags B_l per trip l together with the infeed rate μ^{w} yields trip l's infeed duration

$$\Delta_l^{\text{inf}} = \frac{B_l}{\mu^{\text{w}}.}$$

Preprocessed parameters	**Description**
Δ_l^{inf}	Infeed duration of trip l
$\Delta_{l,h,e}^{\text{lag}}$	Minimum time lag between trip l and h at infeed station e
$\Delta_{i,e,c,s}^{\text{claim}}$	Baggage claim duration for flight i, if tuple (e, c) is selected and the infeed process starts at time s
$\Psi_{i,c}^{\text{walk}}$	Walking distance penalty for passengers of flight i, if baggage carousel c is selected
$\Psi_{l,e,c,s}^{\text{wait}}$	Waiting time penalty for passengers of trip l, if tuple (e, c) is selected and the infeed process starts at time s
$\Phi_{l,e,c,s,t}$	Number of bags of trip l on baggage carousel c at time t, if infeed station e is selected and the infeed process starts at time s

Table 4.3: Parameters derived by preprocessing

- The minimum time lag between two succeeding trips $l \preceq_e h$ of different flights, i.e. $i_l \neq i_h$, and of the same flight, i.e. $i_l = i_h$, at infeed station e are given by

$$\Delta_{l,h,e}^{\text{lag}} = \Delta_l^{\text{inf}} \text{ and (ii) } \Delta_{l,h,e}^{\text{lag}} = \Delta_l^{\text{inf}} + (h-l) \cdot 2 \cdot \Delta_{i_l,e}^{\text{trip}} + (h-l) \cdot \Delta^{\text{pl}},$$

  respectively.

- Function $b_{l,e,c,s}(t)$ and $a_{i,c}(t)$ in $[0, 1]$ give for flight i's trip l the percentage of bags and passengers, respectively, arriving at baggage carousel c in period $t \in \mathcal{T}$. Then, the number of bags of trip l on baggage carousel c in period t is (see also [44] and Robusté and Daganzo [80])

$$\Phi_{l,e,c,s,t} = B_l \cdot \left(b_{l,e,c,s}(t) - a_{i,c}(t) \cdot b_{l,e,c,s}(t)\right).$$

- The total duration for claiming all bags of flight i is

$$\Delta_{i,e,c,s}^{\text{claim}} = \min\left\{t \geq A_{i,c} \mid a_{L_i,c}(t) = b_{L_i,e,c,s}(t) = 1\right\} - S_i^{\text{ob}}.$$

- Passengers' service level can be increased by a short waiting time for baggage at the baggage carousel and a short walking distance from flight's parking position to the baggage carousel. The realized waiting time Δ^{wait} and walking distance d^{walk} is measured by utility function $g(\Delta^{\text{wait}})$ and $g(d^{\text{walk}})$, respectively. The higher the waiting time and the longer the walking distance for a passenger the higher the value of $g(\Delta^{\text{wait}})$ and $g(d^{\text{walk}})$, respectively. In Section 4.5.1 we show how to derive the utility function in practice for a major European Airport. Given the utility function, we derive the penalty coefficients $\Psi_{i,c}^{\text{walk}}$ and $\Psi_{l,e,c,s}^{\text{wait}}$ for the walking distance of flight i's passengers and the waiting time for passengers having bags in trip $l \in \mathcal{L}_i$, respectively.

– Walking distance: The penalty for the walking distance of passengers going to baggage carousel c is given by

$$\Psi_{i,c}^{\text{walk}} = \frac{P_i}{\sum\limits_{i \in \mathcal{F}} P_i} \cdot g(d^{\text{walk}}).$$

– Waiting times: To calculate the penalty term for passengers of flight i waiting t^{wait} minutes from their arrival at the baggage carousel until they pick-up their last bag, for all $t_1, t_2 \geq A_{i,c}$, t_1 being the arrival period of the passenger and t_2 being the arrival period of the passengers' last bag at the baggage carousel, with $t^{\text{wait}} = t_2 - t_1$, passengers' utility value $g(t_2 - t_1)$ is weighted with probability $p_{l,e,c,s}^{\text{wait}}(t_2)$ that all baggage of a passenger has arrived exactly till period t_2, i.e. $g\left(t_2 - t_1\right) \cdot p_{l,e,c,s}^{\text{wait}}\left(t_2\right)$. To derive the penalty across all passengers having baggage in trip l of flight i, we assume in the following that trip l is fed in at infeed station e and the infeed starts at time s. The period when the first bag of trip l arrives on the assigned baggage carousel c is denoted by $t_{l,e,c,s}^{\text{first}}$. Hence, the passengers arrive at baggage carousel c during interval $\left[A_{i,c}, A_{i,c} + \left\lfloor \frac{P_i}{\mu_i^{\text{p}}} \right\rfloor\right]$, while the bags of flight i's trip l arrive on baggage carousel c during interval $\left[t_{l,e,c,s}^{\text{first}}, t_{l,e,c,s}^{\text{first}} + \left\lfloor \frac{B_l}{\mu^{\text{w}}} \right\rfloor\right]$. The fraction of newly arriving passengers of flight i at carousel c in period t is $\bar{a}_{i,c}(t) = a_{i,c}(t) - a_{i,c}(t-1)$. Then the sum of penalties over all passengers waiting during trip l's infeed period yielding trip l's waiting time penalty is given by

$$\Psi_{l,e,c,s}^{\text{wait}} = \frac{1}{|\mathcal{F}| \cdot \max\left\{g(\Delta^{\text{wait}})\right\}} \cdot \sum_{t_1 = A_{i,c}}^{A_{i,c} + \left\lfloor \frac{P_i}{\mu_i^{\text{p}}} \right\rfloor} \bar{a}_{i,c}(t_1) \cdot \left(\sum_{t_2 = \max\left\{t_{l,e,c,s}^{\text{first}}, t_1\right\}}^{t_{l,e,c,s}^{\text{first}} + \left\lfloor \frac{B_l}{\mu^{\text{w}}} \right\rfloor} g\left(t_2 - t_1\right) \cdot p_{l,e,c,s}^{\text{wait}}(t_2) \right)$$

To derive probability $p^{\text{wait}}_{l,e,c,s}(t)$ let $B_{i,e,c}(t) = \sum_{h \in \mathcal{L}_i} B_h \cdot b_{h,e,c,s}(t)$ be the amount of flight i's baggage which has been sent from infeed station e to carousel c up to period t. As trips of the same flight are assigned to the same infeed station and 'first come, first serve' is applied, we can assume that all bags of previous trips of flight i have already arrived at the carousel. For a passenger carrying $1 \leq n \leq N_i$ bags, the likelihood that n bags have arrived until time t is calculated by applying the hypergeometric distribution

$$p^{\text{wait}}_{l,e,c,s}(t) = \sum_{n=1}^{N_i} p_i(n) \cdot \left(\mathbb{1}_{\{B_{i,e,c}(t) \geq n\}} \cdot \frac{\binom{B_i^{\text{tot}} - n}{B_{i,e,c}(t) - n}}{\binom{B_i^{\text{tot}}}{B_{i,e,c}(t)}} - \mathbb{1}_{\{B_{i,e,c}(t-1) \geq n\}} \cdot \frac{\binom{B_i^{\text{tot}} - n}{B_{i,e,c}(t-1) - n}}{\binom{B_i^{\text{tot}}}{B_{i,e,c}(t-1)}} \right) \tag{67}$$

where $\mathbb{1}_{\{B_{i,e,c}(t) \geq n\}}$ is the indicator function which is equal to 1, if $B_{i,e,c}(t) \geq n$, and 0 otherwise. Since we need the likelihood of the passenger's last bag to arrive exactly in period t, we have to subtract the likelihood that all of the passenger's bags have arrived before period t.

4.3.2 Basic model

The model formulation requires for each trip $l \in \mathcal{L}$ one predecessor and one successor trip at each infeed station. To have also a predecessor and successor trip for the first and last trip assigned to an infeed station e, we introduce two dummy trips l_e^0 and l_e^{L+1}; the set of all trips including the dummy trips is given by set $\mathcal{L}^{\text{d}}$. Dummy trips do not send any baggage to the carousel and the start time of dummy trip l_e^0 is set at the beginning of the planning horizon t_0 while the start time of the dummy trip l_e^{L+1} is set to the end of the planning horizon t_T. Moreover, the relation $l_e^0 \preceq_e l$ and $l \preceq_e l_e^{L+1}$ are valid for all $l \in \mathcal{L}$ and $e \in \mathcal{E}$.

Using the decision variables listed in Table 4.4 the model formulation for the inbound baggage handling problem (IBHP) can be written as follows

$$\text{Minimize } f(\mathbf{x}) = \lambda \cdot \left(\sum_{c \in \mathcal{C}} \sum_{t \in \mathcal{T}} u^2_{c,t} \right) + \tag{68}$$

Variable	Description
$x_{l,e,c,s}$	1, if trip l starts the infeed at s and assignment tuple (e, c) is selected, and 0 otherwise
$y_{l,h,e}$	1, if trip l is handled before trip h at infeed station e, and 0 otherwise
$e_{i,c,t}$	1, if flight i's claiming period ends on baggage carousel c at time t, and 0 otherwise
$u_{c,t}$	utilization of baggage carousel c at time t

Table 4.4: Decision variables for the IBHP

$$\begin{aligned} &(1-\lambda)\cdot\left(\sum_{i\in\mathcal{F}}\sum_{e\in\mathcal{E}}\sum_{c\in\mathcal{C}}\sum_{s=S^{\text{es}}_{1_i,e}}^{S^{\text{ls}}_{1_i,e}}\Psi^{\text{walk}}_{i,c}\cdot x_{l_i,e,c,s}\right)\\ \text{Minimize } f(\mathbf{x}) = \lambda\cdot&\left(\sum_{c\in\mathcal{C}}\sum_{t\in\mathcal{T}}u^2_{c,t}\right)+ \\ &(1-\lambda)\cdot\left(\sum_{l\in\mathcal{L}}\sum_{e\in\mathcal{E}}\sum_{c\in\mathcal{C}}\sum_{s=S^{\text{es}}_{l,e}}^{S^{\text{ls}}_{l,e}}\Psi^{\text{wait}}_{l,e,c,s}\cdot x_{l,e,c,s}\right)\end{aligned} \tag{68'}$$

subject to

$$\sum_{e\in\mathcal{E}}\sum_{c\in\mathcal{C}_e}\sum_{s=S^{\text{es}}_{1_i,e}}^{S^{\text{ls}}_{1_i,e}} x_{1_i,e,c,s} = 1 \qquad \forall\, i\in\mathcal{F} \tag{69}$$

$$\sum_{c\in\mathcal{C}_e}\sum_{s=S^{\text{es}}_{l,e}}^{S^{\text{ls}}_{l,e}}\left(x_{l,e,c,s}-x_{1_i,e,c,s}\right) = 0 \qquad \begin{aligned}&\forall\, i\in\mathcal{F},\\ &l\in\mathcal{L}_i\setminus\{1_i\},\\ &e\in\mathcal{E}\end{aligned} \tag{70}$$

$$\sum_{e\in\mathcal{E}_c}\sum_{s=S^{\text{es}}_{l,e}}^{S^{\text{ls}}_{l,e}}\left(x_{l,e,c,s}-x_{1_i,e,c,s}\right) = 0 \qquad \begin{aligned}&\forall\, i\in\mathcal{F},\\ &l\in\mathcal{L}_i\setminus\{1_i\},\\ &c\in\mathcal{C}\end{aligned} \tag{71}$$

$$\begin{aligned}\sum_{c\in\mathcal{C}_e}\sum_{s=S^{\text{es}}_{l,e}}^{S^{\text{ls}}_{l,e}}\left(s+\Delta^{\text{lag}}_{l,h,e}\right)\cdot x_{l,e,c,s}- &\qquad \forall\, e\in\mathcal{E}, l,h\in\mathcal{L}^{\text{d}}:\\ M\cdot(1-y_{l,h,e}) \le \sum_{c\in\mathcal{C}_e}\sum_{s=S^{\text{es}}_{h,e}}^{S^{\text{ls}}_{h,e}} s\cdot x_{h,e,c,s} &\qquad l\neq h\wedge l\preceq_e h\end{aligned} \tag{72}$$

$$\sum_{c\in\mathcal{C}_e}\sum_{s=S^{\mathrm{es}}_{l,e}}^{S^{\mathrm{ls}}_{l,e}}\left(s+\Delta^{\mathrm{lag}}_{l,h,e}\right)\cdot x_{l,e,c,s}+M\cdot(1-y_{l,h,e}) \qquad \forall\, e\in\mathcal{E}, l,h\in\mathcal{L}^{\mathrm{d}}: \tag{73}$$

$$>\sum_{c\in\mathcal{C}_e}\sum_{s=S^{\mathrm{es}}_{h,e}+1}^{S^{\mathrm{ls}}_{h,e}}(s-1)\cdot x_{h,e,c,s} \qquad l\neq h\wedge l\preceq_e h$$

$$M\cdot\left(y_{1_j,h,e}-1\right)+S^{\mathrm{es}}_{1_j,e} \qquad \forall\, i,j\in\mathcal{F}: i\neq j, \tag{74}$$

$$\leq\sum_{c\in\mathcal{C}_e}\sum_{s=S^{\mathrm{es}}_{e,h-1}}^{S^{\mathrm{ls}}_{e,h-1}}\left(s+\Delta^{\mathrm{lag}}_{h-1,h,e}\right)\cdot x_{h-1,e,c,s} \qquad h\in\mathcal{L}_i\setminus\{1_i\}:$$

$$\leq M\cdot\left(y_{h,1_j,e}-1\right)+S^{\mathrm{es}}_{1_j,e} \qquad 1_j\cong_e h, e\in\mathcal{E}$$

$$M\cdot(y_{l,h,e}-1)+ \qquad \forall\, i,j\in\mathcal{F}: i\neq j, \tag{75}$$

$$\sum_{c\in\mathcal{C}_e}\sum_{s=S^{\mathrm{es}}_{e,l-1}}^{S^{\mathrm{ls}}_{e,l-1}}\left(s+\Delta^{\mathrm{lag}}_{l-1,l,e}\right)\cdot x_{l-1,e,c,s} \qquad l\in\mathcal{L}_i\setminus\{1_i\}, h\in\mathcal{L}_j\setminus\{1_j\}:$$

$$\leq\sum_{c\in\mathcal{C}_e}\sum_{s=S^{\mathrm{es}}_{e,h-1}}^{S^{\mathrm{ls}}_{e,h-1}}\left(s+\Delta^{\mathrm{lag}}_{h-1,h,e}\right)\cdot x_{h-1,e,c,s} \qquad l\cong_e h, e\in\mathcal{E}$$

$$y_{l,h,e}+y_{h,l,e}\leq 1 \qquad \begin{array}{l}\forall\, e\in\mathcal{E}, l,h\in\mathcal{L}:\\ l\neq h\wedge l\cong_e h\end{array} \tag{76}$$

$$\sum_{c\in\mathcal{C}_e}\sum_{s=S^{\mathrm{es}}_{l,e}}^{S^{\mathrm{ls}}_{l,e}}x_{l,e,c,s}=\sum_{h\in\mathcal{L}\cup\left\{l_0^{\mathrm{start}}\right\}:h\neq l}y_{h,l,e} \qquad \forall\, l\in\mathcal{L}, e\in\mathcal{E} \tag{77}$$

$$\sum_{c\in\mathcal{C}_e}\sum_{s=S^{\mathrm{es}}_{l,e}}^{S^{\mathrm{ls}}_{l,e}}x_{l,e,c,s}=\sum_{h\in\mathcal{L}\cup\left\{l_0^{\mathrm{end}}\right\}:h\neq l}y_{l,h,e} \qquad \forall\, l\in\mathcal{L}, e\in\mathcal{E} \tag{78}$$

$$\sum_{l\in\mathcal{L}}y_{l_0^0,l,e}\leq 1 \qquad \forall\, e\in\mathcal{E} \tag{79}$$

$$\sum_{l\in\mathcal{L}}y_{l,l_0^{\mathrm{L}+1},e}\leq 1 \qquad \forall\, e\in\mathcal{E} \tag{80}$$

$$\sum_{e\in\mathcal{E}}\sum_{s=S^{\mathrm{es}}_{L_i,e}}^{S^{\mathrm{ls}}_{L_i,e}}\left(s+\Delta^{\mathrm{claim}}_{i,e,c,s}\right)\cdot x_{L_i,e,c,s}= \qquad \forall\, i\in\mathcal{F}, c\in\mathcal{C} \tag{81}$$

$$\sum_{t\geq S^{\mathrm{ob}}_i}t\cdot e_{i,c,t}$$

$$\sum_{c\in\mathcal{C}}\sum_{t=S_i^{\mathrm{ob}}}^{T} e_{i,c,t} = 1 \qquad \forall\, i\in\mathcal{F}, c\in\mathcal{C} \tag{82}$$

$$\sum_{i\in\mathcal{F}:S_i^{\mathrm{ob}}\le t}\left(1-\sum_{z=S_i^{\mathrm{ob}}}^{t} e_{i,c,z}\right) \le K_c^{\mathrm{di}} \qquad \forall\, c\in\mathcal{C}, t\in\mathcal{T} \tag{83}$$

$$\frac{1}{K_c^{\mathrm{cb}}}\sum_{l\in\mathcal{L}}\sum_{e\in\mathcal{E}_c}\sum_{s=S_{l,e}^{\mathrm{es}}}^{S_{l,e}^{\mathrm{ls}}} \Phi_{l,e,c,s,t}\cdot x_{l,e,c,s} \le u_{c,t} \qquad \forall\, c\in\mathcal{C}, t\in\mathcal{T} \tag{84}$$

$$x_{l,e,c,s}\in\{0,1\} \qquad \forall\, l\in\mathcal{L}, e\in\mathcal{E}, c\in\mathcal{C}_e,\ S_{i,e}^{\mathrm{es}}\le s\le S_{i,e}^{\mathrm{ls}} \tag{85}$$

$$y_{l,h,e}\in\{0,1\} \qquad \forall\, e\in\mathcal{E}, l,h\in\mathcal{L}:\ l\neq h\wedge l\preceq_e h \tag{86}$$

$$e_{i,c,t}\in\{0,1\} \qquad \forall\, i\in\mathcal{F}, c\in\mathcal{C}, t\in\mathcal{T} \tag{87}$$

$$u_{c,t}\ge 0 \qquad \forall\, c\in\mathcal{C}, t\in\mathcal{T}. \tag{88}$$

Objective function (68) and (68') minimize the carousels' utilization and additionally passengers' walking distances and waiting times, respectively. The domain of the variables are defined in (85) to (88). IBHP's set of constraints can be separated into constraints (69) to (80) modeling the infeed process and constraints (81) to (84) modeling the passenger process.

Infeed process In constraints (69) the infeed start time of each trip is selected and assigned to one infeed and one baggage carousel. Due to (70) and (71) each trip $l\in\mathcal{L}_i\setminus\{1_i\}$ of flight $i\in\mathcal{F}$ is assigned to the same infeed station and to the same baggage carousel as the first trip 1_i. The precedence relation between two consecutive trips are set in constraints (72) to (80). Constraints (72) ensure that two trips are consecutively served at an infeed station, while an artificially delay of the infeed process of a trip is forbidden by constraints (73). For example, consider two trip l and h consecutively assigned at infeed station e, i.e. $y_{l,h,e}=1$, where trip l starts its infeed at time $t=2$ and ends at $t=3$. Let us first assume that trip h can start the infeed process at $t=2$, i.e. trip h's time window to start the infeed overlaps with the time window of trip l. Then the left term of constraints (73) is equal to 3 enforcing that the right term has to be 2 which sets the start time of trip h to 3, i.e. trip h is immediately fed in after trip l. If we assume that trip h's earliest infeed time is 4, i.e. the time windows for the start times do not overlap, it follows that the right hand side of constraints (73) has to be 0 and therefore $x_{h,e,c,4} = 1$ and, hence, trip h is fed in with no artificial delay after trip l. To guarantee that also the first trip is fed as soon as the trip arrives, we set the infeed duration of dummy trip l_e^0 equal to a small constant > 0, while the infeed duration for the dummy trip $l^{\mathrm{N}+1}$ is set to 0. Due to constraints (74) and (75), trips arriving first at infeed station e are served first. To guarantee that either trip l is served before trip h, or vice versa, we require constraints (76). Constraints (77) and (78) set for each trip $l \in \mathcal{L}$ exactly one predecessor and one successor trip at the assigned infeed station e. Constraints (79) and (80) set at most one trip l as successor or predecessor for dummy trips l_e^0 and l_e^{L+1}, respectively.

Passenger process The number of handled flights at a baggage carousel c at a time is limited by constraints (81) to (83). Constraints (84) measure baggage carousel c's utilization at any time t.

Linearization In order to linearize the first quadratic part of the objective function, the utilization is approximated with a step-wise function. We introduce threshold values $u_1 < \ldots < u_k < \ldots u_K$ and associate to each threshold value greater than u_k a penalty $p_1^{\mathrm{u}} < \ldots < p_k^{\mathrm{u}} < \ldots < p_K^{\mathrm{u}}$. The model formulation for the linearized IBHP (L-IBHP) can be found in Appendix A.4.

4.3.3 Complexity analysis

The L-IBHP has $L \cdot E \cdot (C \cdot S^* + L - 1) + C \cdot T \cdot (F + K)$ binary variables, $C \cdot T$ integer variables and $F \cdot (1 + (L - F) \cdot (E + C) + (F - 1) \cdot (L - F) \cdot (E + L - F)) + E \cdot (3 \cdot L^2 - L + 2) + C \cdot (F + 2 \cdot T)$ constraints where $S^* = \max\limits_{l \in \mathcal{L}, e \in \mathcal{E}} \left\{ S_{l,e}^{\mathrm{ls}} - S_{l,e}^{\mathrm{es}} + 1 \right\}$ is the maximal length of the time windows. For example, in a real world scenario (see Section 4.6) we plan flights arriving during the next 180 minutes in advance where we take a period length of 1 minute. In this time range, we have to plan 50 flights on average in which each flight needs on average 2 trips where the length of the time window to feed in the baggage is equal to 10 minutes. As infrastructure we have 7 carousels and 12 infeed stations. Using 6 threshold values for the utilization in the linearized L-IBHP's objective function leads to at most 68,484 binary variables, 1,260 variables and 4,191,064 constraints.

Beside the large number of variables the problem can be separated into the two NP-complete decision problems

DP I – Is there a feasible assignment of flights to infeed stations?

DP II – Is there a feasible assignment of flights to carousels such that a given workload peak u^* is not exceeded?

Theorem 4.3.1. *DP I is NP-complete.*

Proof. Obviously, DP I is in NP. That is, given any assignment and schedule, we can test feasibility in polynomial time. To show that IBHP is NP-hard, we proof that the feasibility problem for DP I is already NP-hard. Therefore, we reduce the feasibility problem of DP I in polynomial time to the K-colorability problem which is known to be NP-hard (see Garey and Johnson 43). In the K-colorability problem it is asked, whether the vertices $v \in \mathcal{V}$ of a given graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ can be colored with at most K colors such that no adjacent vertices v and v' with $(v, v') \in \mathcal{A}$ have the same color. □

Suppose an arbitrary K-colorability problem with an underlying graph $\mathcal{G}$ is given. Assume the infeed duration for each trip is equal to one and each trip has to be handled immediately, i.e. $S^{\text{es}}_{l,e} = S^{\text{ls}}_{l,e}$. Moreover, let $S^{\text{es}}_{l,e} = S^{\text{es}}_{l,e'}$ for all $e, e' \in \mathcal{E}$, meaning that the start time of the infeed process of trip l is identical for all infeed stations. Now we define the following transformation:

- For each vertex $v \in \mathcal{V}$ we define a flight $i \in \mathcal{F}$ in DP I so that the number of flights equals the number of vertices.
- For each color K we define an infeed station such that an infeed station always corresponds to exactly one color.
- For each edge $(v, v') \in \mathcal{A}$ we define a time period $t \in \mathcal{T}$ and two trips l and l', one for each flight that corresponds to either vertex v or v'. Further, let $S^{\text{es}}_{l,e} = S^{\text{es}}_{l',e} = t$ for all $e \in \mathcal{E}$, i.e. both trips have to be handled in period t.

According to the transformation, for each pair of vertices that are connected in $\mathcal{A}$, i.e. which cannot be assigned the same color, there is a pair of trips of the flights corresponding to the vertices which cannot be assigned to the same infeed station. As all trips of a flight have to be assigned to the same infeed station, a flight (vertex) is assigned to exactly one infeed station (color) and all trips that have to be handled in the same period (each pair of adjacent vertices) cannot be assigned to the same infeed station (color). Therefore, solving DP I answers the decision problem of the K-colorability problem.

The K-colorability problem is already NP-hard for $K = 3$, meaning that only three infeed stations lead to the NP-hardness of the IBHP. The problem becomes even more complex when considering the carousels with the capacity restrictions on the number of flights which can be served at a time.

Theorem 4.3.2. *DP II is NP-complete.*

Proof. We proof the assumption by reduction of DP II to the Bin-Packing Problem which is known to be NP hard in the strong sense (see [43]). The decision problem of the Bin-Packing Problem can be stated as follows: Given a set of items $\mathcal{I}$, with each item $i \in \mathcal{I}$ having a size $i_s \in \mathbb{Z}_+$, can these items be packed in K bins each of size $S \in \mathbb{Z}_+$.

We consider trips $\mathcal{L} = \mathcal{I}$ trips that are handled at $|\mathcal{L}|$ remote infeed stations having access to $C = K$ carousels. W.l.o.g. we consider planning period t in which all s_l bags of each trip $l \in \mathcal{L}$ arrive at the assigned carousel, assuming the infeed rate and the transportation time is long enough. Moreover, we assume that no passenger arrives at one of the carousels in period t. With the given setting, the question arises whether we can find an assignment of trips to carousels such that the carousel capacity S is not exceeded. □

Since the problem is NP-hard to solve in terms of the inbound problem at the airside and the utilization problem at the landside (see Theorem 4.3.1 and 4.3.2) and because of the large number of variables and constraints, MIP formulation (68) – (102) does not guarantee to obtain a solution within an acceptable time for a practical implementation (see Section 4.6). The time aspects becomes crucial as the volatility of the arrival time of flights and the number of arriving bags makes it necessary to adjust the solution over the whole planning day within short time periods, e.g. at a major European airport the algorithm presented in Section 4.4 runs after each touch down of a flight.

4.4 Solution methodology

In order to be able to solve real-world instances we propose a heuristic integrating a guided fast local search (GFLS) within a greedy randomized adaptive search procedure (GRASP) framework, which we will call hybrid GRASP-GFLS (hybrid HGGLS) in the following. In the GRASP framework (see Alvarez-Valdes et al. 4 or Feo and Bard 36), a randomized greedy algorithm generates starting solutions in a construction phase. A subsequent local search phase is used to improve the starting solutions. Past reasearch has shown that further improvement of GRASP procedures can be made by adding a third, memory-based search phase based on path-relinking (see Laguna and Marti 61 or Resendel and Ribeiro 77). The heuristic we present is composed of the following three phases:

1. Following the GRASP framework, the first phase is a memory-less construction phase using a greedy randomized heuristic. A greedy heuristic iteratively extends a (partial) solution by adding a new element (e.g. an assignment) in a greedy manner according to the best value of a defined greedy function. To overcome the deterministic nature of a greedy heuristic, in each iteration, the current solution is extended by an element that is randomly chosen out of a restricted candidate list $\mathcal{RC}$. Based on a greedy function, the $\mathcal{RC}$ is composed of all elements that lead to a new solution within α% of the best solution found when adding an element.

2. In the improvement phase, a GFLS is applied to improve the solution $\mathbf{x}_0$ found in the construction phase. GFLS is a combination of guided local search (GLS) and fast local search (FLS) (see Faroe et al. 34 or Voudouris and Tsang 96). GLS is a local search scheme that augments the objective function of a local search procedure by incorporating penality terms. Every time the local search procedure settles in a local minimum, GLS modifies the objective function using the penalty terms in order to overcome the local minimum. For this, so-called solution features are used to determine the penalties. A feature $\phi \in \Phi$ defines a property of the problem having direct or indirect impact on solution $\mathbf{x}$. A feature can be any property of a solution that is non-trivial in the sense that not all solutions have this property. The augemented cost function of an objective function $f(\mathbf{x})$ and a solution $\mathbf{x}$ is defined as

$$h(\mathbf{x}) = f(\mathbf{x}) + \lambda \cdot \sum_{\phi \in \Phi} p_\phi \cdot I_\phi(\mathbf{x}) \tag{89}$$

where p_ϕ is the penaltiy associated with a feature ϕ and indicator function $I_\phi(\mathbf{x})$ is 1 if $\mathbf{x}$ has feature ϕ and 0 otherwise. The feature penalties are weighted with parameter λ. Initially, the penalties of all features are set to 0. Each time a local optimum according to $h(\mathbf{x})$ is found, penalities of features having the highest utility

$$u_\phi(\mathbf{x}) = I_\phi(\mathbf{x}) \cdot \frac{cost_\phi}{1 + p_\phi}$$

are penalized by incrementing their penaltly by one. $cost_\phi$ denotes the feature cost which can either be constant or variable.

Furthermore, FLS is used to reduce the size of the neighborhoods in order to improve runtime.

3. As the constructive and the improvement phase are memoryless procedures, i.e. they generate solutions independently of previously generated solutions, path-relinking is applied to further improve the found solution. The basic idea of path-relinking is to iteratively change an initial solution towards a guiding solution in order to explore the solution space between promising solutions.

For an overview of a single iteration of the heuristic procedure see Algorithm 2; Table 4.5 provides a description of the used variables and functions.

In the following, a solution is represented by a set of tuples $\{(e, c)_i \in \mathcal{E} \times \mathcal{C}_e | i \in \mathcal{F}\}$ where each flight is assigned to a tuple (e, c) and for each infeed station $e \in \mathcal{E}$ we define an infeed order $\prec_e$ which leads to a unique solution in case more than one trip is arriving at the same time at the same infeed station. Note that each representation leads to a unique solution as the infeed begin of a trip cannot be artificially delayed in the infeed process as stated in IBHP.

Function	**Description**
h^{cstr}	cost function in constructive phase
h	augmented cost function in improvement phase
Solution	**Description**
$\mathbf{x}^{best}$	best solution found based on function h^{cstr}
$\mathbf{x}$	current iteration best solution found based on function h^{cstr}

Table 4.5: Notation for the pseudo code

Algorithm 2 HGGLS heuristic

repeat
  $\mathbf{x} \leftarrow ConstructionHeuristic$ // Determine start solution with GRASP (see Sec. 4.4.1)
  $\mathbf{x} \leftarrow GFLS(N^1(\mathbf{x}))$ // GFLS on neighborhood $N^1(\mathbf{x})$ (see Sec. 4.4.2 and App. A.7, Algorithm 5)
  $\mathbf{x} \leftarrow GFLS(N^2(\mathbf{x}))$ // GFLS on neighborhood $N^2(\mathbf{x})$ (see Sec. 4.4.2 and App. A.7, Algorithm 5)
  if $h^{\text{cstr}}(\mathbf{x}) > h^{\text{cstr}}(\mathbf{x}^{best})$ **then**
    $Path-Relinking(\mathbf{x}, \mathbf{x}^{best})$ // Path-relinking between solution $\mathbf{x}$ and solution $\mathbf{x}^{best}$ (see Section 4.4.2)
  end if
until STOPPING CRITERION MET

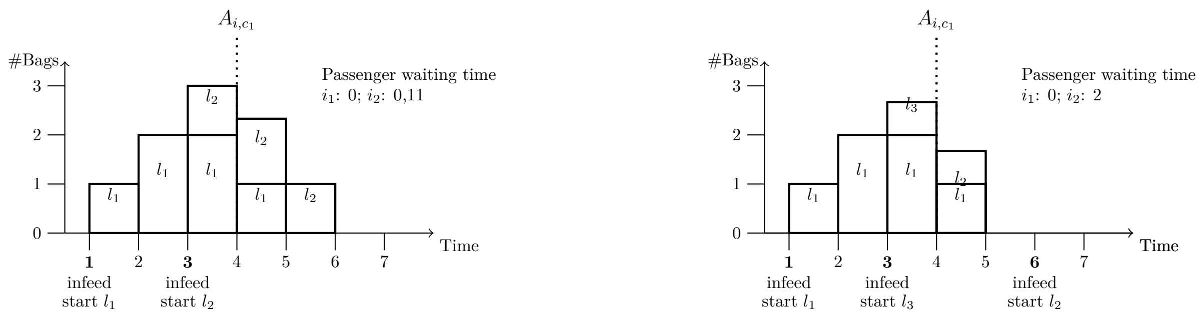
4.4.1 Constructive phase

In the constructive phase, the set of flights $\mathcal{F}$ is ordered according to increasing scheduled time of arrival. Let $\mathbf{x}^{i-1}$ be the partial solution with the first $i-1$ flights of $\mathcal{F}$ already assigned, and let $i \in \mathcal{F}$ be the next flight to be assigned. For this, all tuples (e, c) are evaluated. If flight i is assigned to tuple (e, c), flight i's trips $\mathcal{L}_i$ have to respect infeed order $\preceq_e$ at infeed station e. For each trip l of flight i and each trip h already assigned to station e with $l \cong_e h$, both infeed orders $l \preceq_e h$ and $h \preceq_e l$ have to be considered. For each tuple (e, c) and infeed order $\prec_e$ we evaluate the cost value $h(\mathbf{x}^i)$ where $\mathbf{x}^i$ is the partial solution we obtain after adding tuple $(e, c)_i$ to the set of tuples of partial solution $\mathbf{x}^{i-1}$, respecting infeed order $\prec_e$ (see 4.4.2 for a description of cost function $h(\mathbf{x})$). Assigning flight i to tuple (e, c) requires the re-calculation of the utilization and the service quality at all carousels $c \in \mathcal{C}_e$ reached by infeed station e. All elements (i.e. combinations of tuple (e, c) and infeed order $\prec_e$) leading to partial solutions not worse than $\alpha\%$ in comparision to the best partial solution found are put into $\mathcal{RC}$. Finally, flight i is randomly assigned to a tuple (e, c) and infeed order $\prec_e$ from $\mathcal{RC}$, and the partial solution is updated. The procedure continuous until all flights are assigned and scheduled in solution $\mathbf{x}^{\text{best}}$.

To demonstrate the effect of a new assignment on carousels' utilization and the level of service quality, let us consider the following example: Assume we have three trips l_1, l_2 and l_3 of three flights i_1,i_2 and i_3 (see Table 4.6). There is one remote infeed station e with an infeed rate of 1 bag per period which has access to carousels c_1 and c_2. Further, we assume that transportation time from the infeed station to the carousel is neglected. The arrival time of the first passenger is the same at both carousels and the arriving rate of passengers is equal to 1 passenger per period. Each bag belongs to exactly one passenger. The objective is to minimize passenger waiting times. Finally, at infeed station e we have order $\preceq_e$ with $l_1 \preceq_e l_3 \preceq_e l_2$. Assigning the first two trips to tuple (e, c_1) results in the bag arrival profile shown in Figure 4.3 (a). Based on $p^{\text{wait}}_{l,e,c,s}(t)$, passengers's average waiting times are 0 and 0.11 periods for trip l_1 and l_2, respectively. Next, we assign trip l_3 to assignment tuple (e, c_2). According to the order $\preceq_e$, we have to schedule trip l_3 before trip l_2. Figure 4.3 (b) shows the resulting workload of carousel c_1. As can be seen, the delayed infeed of trip l_2 influences the workload and waiting times at carousel c_1, even though trip l_3 is not assigned to carousel c_1. While the workload has decreased, average passenger waiting time for trip l_2 has increased to 2 periods.

Flight	Trip	B_l	$S^{\text{es}}_{l,e}$	$A_{i,c}$	Assignment tuple
i_1	l_1	2	1	4	(e, c_1)
i_2	l_2	3	2	4	(e, c_1)
i_3	l_3	3	1	3	(e, c_2)

Table 4.6: Example with 3 flights and their trips



(a) Infeed order $l_1 \prec_e l_2$

(b) Infeed order $l_1 \prec_e l_3 \prec_e l_2$

Figure 4.3: Workload and passenger waiting times at carousel c_1 before and after the assignment of trip l_3 to remote infeed station e; the dotted line indicate the arrival time of passengers at the carousel

4.4.2 Local search

Neighborhood Given an initial solution $\mathbf{x}$ the local search is based on two neighborhoods $\mathcal{N}^1(\mathbf{x})$ and $\mathcal{N}^2(\mathbf{x})$. Neighborhood $\mathcal{N}^1(\mathbf{x})$ includes all solutions which can be obtained by changing the solution concerning a single flight. Therefore, two neighborhood moves can be performed for each flight i:

1) Change the tuple of flight i; e.g. change $(e, c)_i$ to $(e', c')_i$ with $e \neq e' \vee c \neq c'$. Note that all possible infeed orders $\prec_{e'}$ have to be considered in case $e \neq e'$.

2) Change the infeed order of two simultaneously arriving trips l and h with $l \in \mathcal{L}_i \vee h \in \mathcal{L}_i$ at an infeed station e; e.g. if l is served before h, replace $l \prec_e h$ by $h \prec_e l$;

It is obvious that we are capable of reaching each solution in the space of feasible solutions through the defined neighborhood $\mathcal{N}^1(\mathbf{x})$.

Besides assigning a single flight to a new tuple (e, c) or changing the infeed order of a flight's trip, we also apply a 2-opt neighborhood search in neighborhood $\mathcal{N}^2(\mathbf{x})$. Neighborhood $\mathcal{N}^2(\mathbf{x})$ swaps the carousel assignment of two flights i and j assigned to different carousels. For both flights, all infeed stations connected to the new carousel are evaluated, i.e. the tuples of flight i and j are changed according to:

$(e_i, c_i)_i \rightarrow (e'_i, c_j)_i, e'_i \in \mathcal{E}_{c_j}$ and $(e_j, c_j)_j \rightarrow (e'_j, c_i)_i, e'_j \in \mathcal{E}_{c_i}$.

Evaluating the new assignment includes all possible combinations of infeed orders $\prec_{e'_i}$ and $\prec_{e'_j}$ if $e'_i \neq e_i$ or $e'_j \neq e_j$, respectively.

Note that if a neighborhood move changes the infeed station of a flight, it can have an impact on the start time of the infeed of all subsequent flights assigned to the formerly and newly assigned infeed station of the flight and consequently on all carousels that can be reached by these infeed stations. Therefore, in the worst case, the utilization and the level of service quality at all carousels has to be re-calculated (see the example in Section 4.4.1). The same holds for a change of the infeed order at an infeed station.

Augmented cost function To evaluate a solution $\mathbf{x}$ we extend objective function $f(\mathbf{x})$, given in (68), by two set of features. The first set of features corresponds to the carousels' capacity violations in terms of assigned flights at a time whereas the second set corresponds to the tardiness of the infeed for a flight's trips, i.e. to the number of periods the infeed begin violates the latest infeed time.

For each carousel $c \in \mathcal{C}$, let indicator function $I_c^{\text{cap}}(\mathbf{x})$ be 1 if solution $\mathbf{x}$ violates the capacity restrictions (81) - (83) (i.e. the maximum number of flights) on carousel c, and

0 otherwise. For each flight $i \in \mathcal{F}$ and trip $l \in \mathcal{L}_i$, let indicator function $I^{\text{ls}}_{i,l}(\mathbf{x})$ be 1 if the infeed of flight i's trip l starts later than its latest infeed time $\mathcal{S}^{\text{ls}}_{i,l}$. The augmented cost function

$$h(\mathbf{x}) = f(\mathbf{x}) + \sum_{c \in \mathcal{C}} p^{\text{cap}}_c \cdot I^{\text{cap}}_c(\mathbf{x}) + \sum_{i \in \mathcal{F}} \sum_{l \in \mathcal{L}_i} p^{\text{ls}}_{i,l} \cdot I^{\text{ls}}_{i,l}(\mathbf{x}) \tag{90}$$

evaluates each solution $\mathbf{x}$ where p^{cap}_c and $p^{\text{ls}}_{i,l}$ are the penalties for the two features. If GFLS gets stuck in a local minimum, penalties of features having the highest utlity

$$\begin{aligned} u^{\text{cap}}_c(\mathbf{x}) &= I^{\text{cap}}_c(\mathbf{x}) \cdot \frac{(\#\text{capacity violations on carousel } c)}{1 + p^{\text{cap}}_c} \qquad \text{and} \\ u^{\text{ls}}_{i,l}(\mathbf{x}) &= I^{\text{ls}}_i(\mathbf{x}) \cdot \frac{(\text{start time of flight } i) - S^{\text{ls}}_i}{1 + p^{\text{ls}}_{i,l}} \end{aligned} \tag{91}$$

are incremented by one.

Augmented cost function $h(\mathbf{x})$ is used as evaluation function in the constructive phase and in the improvement phase. Since we want to start the local search with a feasible solution, the values for the carousel capacity and the tardiness penalties in the constructive phase are set to a big M value (e.g. $p^{\text{cap}}_c = T \cdot p^{\text{u}}_K$ and $p^{\text{ls}}_{i,l} = p^{\text{w}}_K$) and we denote function $h(\mathbf{x})$ by $h^{const}(\mathbf{x})$. At the beginning of the local search we set the penalty values to zero and increment the penalties by one according to the highest utiliy (91) each time the local search settles in a local minimum.

If local search in either neighborhood $\mathcal{N}^1(\mathbf{x})$ or $\mathcal{N}^2(\mathbf{x})$ does not find an improvement, we update the augmented objective function $h(\mathbf{x})$ according to (91).

Fast local search To speed up the local search we make use of fast local search (FLS) techniques (see Faroe et al. 34, Tsang and Voudouris 93 and Voudouris 95) which subdivides the neighborhood into small sub-neighborhoods that can be either active or inactive. The idea is to examine the sub-neighborhoods in a given order, searching only the active ones. After exploring a sub-neighborhood, the search continues with the next active sub-neighborhood. A sub-neighborhood is set inactive when no solution leading to an improvement can be found. We define the sub-neighborhoods of $\mathcal{N}^1(\mathbf{x})$ and $\mathcal{N}^2(\mathbf{x})$ as follows:

- $\mathcal{N}^1_i(\mathbf{x})$: For each flight i, we define $\mathcal{N}^1_i(\mathbf{x})$ as the set of solutions that can be obtained by performing a neighborhood move in $\mathcal{N}^1(\mathbf{x})$ for flight i

- $\mathcal{N}_i^2(\mathbf{x})$: For each flight i, we define $\mathcal{N}_i^2(\mathbf{x})$ as the set of solutions that can be obtained by performing a neighborhood move in $\mathcal{N}^2(\mathbf{x})$ where one of the swapped flights is i and the neighborhood of the other flight is active.

We order both sets of sub-neighborhoods according to the flight's expected time of arrival. If an improvement is found, all affected sub-neighborhoods are reactivated. A flight i and its corresponding sub-neighborhood can either be affected when the infeed time of at least one of its trips changes or when the utilization at the flights' carousel during the processing of i changes.

We can further speed up local search by reducing the neighborhood size. For $\mathcal{N}_i^1(\mathbf{x})$ and $\mathcal{N}_i^2(\mathbf{x})$ we consider only infeed stations within a maximal distance **MaxDist** from flight i's current infeed station. We can further reduce the size of neighborhood $\mathcal{N}_i^2(\mathbf{x})$ only swapping the carousel assignment of flight i with those flights which arrive on block within the time window $\left[S_i^{\text{ob}} - \textbf{MaxTime}, S_i^{\text{ob}} + \textbf{MaxTime}\right]$. Algorithm 5 in Appendix A.7 shows the pseudo code for the local search.

Path-relinking Path-relinking is used to explore a trajectory from the currently generated solution to the best solution found so far and vice versa in order to find new solutions (forward and backward path-relinking). The solutions can differ in two respects:

- A flight is assigned to a different infeed station or to a different carousel
- Two trips competing for the same infeed station in the same period have a different infeed order

We transform a start solution into the target solution by sequentially swapping the assignment or infeed order of the initial solution into the assignment or infeed order of the target solution in a best first manner. In each step, we can either change the infeed order of two trips at an infeed station or the assignment tuple of a flight. In case of moving a flight to a new infeed station, we adapt the infeed order of the target solution. For example, assume two flights i and j that are assigned to tuple (e_i, c_i) and (e_j, c_j) in the start solution and to assignment tuple (e'_i, c'_i) and(e'_j, c'_j) in the target solution, respectively. If $e_i \neq e'_i \vee c_i \neq c'_i$ and $e_j \neq e'_j \vee c_j \neq c'_j$ we swap the assignment of either flight i or flight j in a best first manner based on objective $h^{const}(\mathbf{x})$.

4.5 Practical implementation

This section describes and discuss some of the features of the algorithm presented in Section 4.4 which are required when we implemented the algorithm at a major European Airport. We show in Section 4.5.1 how we set the penalties and threshold values for the service levels in the objective function. When using a weighted sum objective in multi-criteria optimization one crucial point is the right choice of weights. In order to give the dispatcher several solutions for different weight combinations in short time, we adapt path-relinking in an innovative way and introduce multi-objective path-relinking (see Section 4.5.2). Finally, Section 4.5.3 discuss the requirements when implementing the heuristic in a rolling planning framework to reduce the instability of the obtained solutions.

4.5.1 Measuring the level of service

The daily planning of the inbound baggage handling affects passengers' level of service on an operational level (see Correia and Wirasinghe 27 and Martel and Senviratne 69). The relevant aspects for passengers' level of service depend on the given airport infrastructure and varies between airports. To investigate the operational decisions influencing passengers' service quality we conducted a survey at a major European Airport. According to [27], we asked the passengers picking up their baggage to assess the

a) walking distance from flight's parking position to the baggage carousel;

b) waiting time at the baggage carousel;

c) space available when picking up their baggage.

The details and the results of the survey are provided in Appendix A.5. The results showed that passengers' waiting times are most relevant for the level of service. To obtain utility function $g(d^{\text{wait}})$ for passengers' waiting times, we interpolate the points between passengers' assessment for the waiting time on a scale from 1 (=very satisfied) to 6 (=not satisfied) and passengers' realized waiting time measured in seconds (see Figure 4.4). The best interpolation is a polynomial function of degree 2. Hence, with increasing waiting time, passengers' tolerance of waiting decreases quadratically.

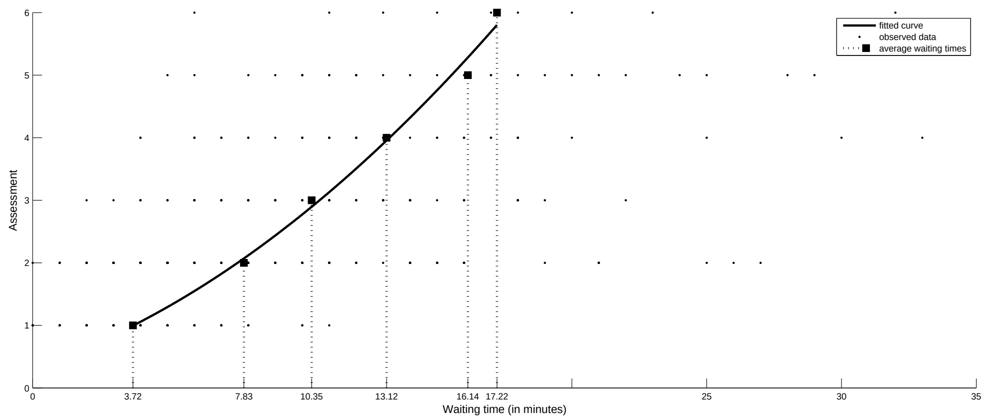


Figure 4.4: Interpolation between passengers' assessment and the realized waiting time; the size of the dots represents the number of observations in the same minute

4.5.2 Multi-Objective path-relinking

The solution of the multi-objective objective function depends on weight λ (see objective function (68)). The value of λ can be based on different factors like experience, airports operational goals or customer survey evaluations (see Section 4.5.1). Besides using a best practice value for λ, we provide the dispatcher with an interactive method so that the dispatcher can consider different choices of λ in short time. In order to quickly produce new results, we do not want to resolve the problem from scratch when using a different weighting of the objectives because this is too time consuming when done for several λ values. Therefore, we propose an innovative approach which makes use of the path-relinking idea. The procedure consists of the following three steps:

- In a first step, a solution $\mathbf{x}^{\text{init}}$, based on weight λ^{init}, is generated and presented to the dispatcher.
- In a second step, we solve the problem for values $\lambda^{\text{dec}} < \lambda^{\text{init}}$ and $\lambda^{\text{inc}} > \lambda^{\text{init}}$ leading to solutions $\mathbf{x}^{\text{dec}}$ and $\mathbf{x}^{\text{inc}}$. The dispatcher can decide whether he selects the initial solution $\mathbf{x}^{\text{init}}$ based on λ^{init} or further explores the solution space by either incrementing the weight towards λ^{inc} or by decrementing the weight towards λ^{dec}.
- In case of a decrease of the weight from λ^{init} towards λ^{dec} or an increase towards λ^{inc} a third step is initiated. In the first case, a path-relinking procedure from starting

solution $\mathbf{x}^{\text{init}}$ to guiding solution $\mathbf{x}^{\text{dec}}$ is performed, while in the latter case a path-relinking procedure from starting solution $\mathbf{x}^{\text{init}}$ to guiding solution $\mathbf{x}^{\text{inc}}$ is performed. Following the path relinking procedure presented in Section 4.4.2, moves between the solutions are made in a best first manner, but now the best move is evaluted based on the new objective weight set by the dispatcher.

The procedure iteratively presents new solutions $\mathbf{x}^{\text{new}}$ to the dispatcher without having to resolve the problem from scratch each time (see Figure 4.5). In the following, we call the procedure multi-objective path-relinking (MOPR)

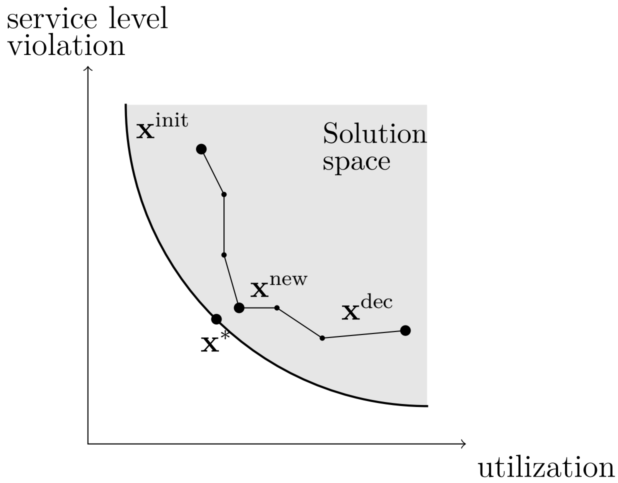


Figure 4.5: Solution space if the objective weight is decreased from λ^{init} towards λ_{dec} with $\mathbf{x}^{\text{new}}$ as best solution found when exploring the path between solutions $\mathbf{x}^{\text{init}}$ and $\mathbf{x}^{\text{dec}}$; $\mathbf{x}^{*}$ shows the optimal solution for the new weight of λ

4.5.3 Rolling planning

The quality of a solution depends, among other things, on a proper estimation of the arrival time of flights. For example, a change in flights' arrival time could deteriorate the current solution if the time change leads to an overlapping of flights' handling times at infeed stations or on a baggage carousel. To decrease the dependencies of the solution on flights' arrival times and to obtain robust solutions over the planning day the algorithm updates the current solution after each new touch-down of a flight. In the re-planning, flights currently handled at an infeed station or baggage carousel are fixed according to the previously undertaken assignment. For flights arriving in the future, the actual expected arrival times are used when the algorithm starts to updates the solution. As the actual arrival time is updated by the incoming flights during their journey flights', actual expected arrival times becomes more accurate the closer the airplane comes to the airport, which reduces the volatility of the solutions.

Since the assignment of the carousel and infeed station has to be determined some time before the flight is on-block, the computing time to obtain a solution is restricted to at most Δ^{time} minutes (e.g. 3 minutes). The computing time should be chosen such that the dispatcher may have some time left to elaborate other weight combinations with the multi-objective path-relinking method (see Section 4.5.2).

To improve the quality of solutions obtained within the given computation time of Δ^{time}, we reduce the size of the solution space and, hence, the number of possible neighborhoods to be examined, by considering only flights arriving during the next $\Delta^{\text{next}} > 0$ minutes (e.g. 180 minutes) in expectation; flights further in the future have less influence on flights currently handled. Moreover, as the plan is updated during the course of the day and as the announcement of the baggage carousel is only important for incoming passengers already on the ground, a plan for all flights is not required for inbound baggage handling.

4.6 Computational study

All experiments are conducted on a Windows 7 platform with a 2.8 GHz CPU and 4 GB RAM. To solve the MIP we use CPLEX 12.6. For the evaluation of the real-world instances in a real-world scenario we implemented a simulation in the programming language JAVA.

For our computational results we use real-world data of a major European Airport from October 2012. The instances are based on an infrastructure with 7 baggage carousels where we have 5 similar baggage carousels with a conveyor belt capacity for 75 bags and 2 baggage carousels with a capacity for 90 bags. Each baggage carousel can serve up to 6 flights simultaneously. For the infeed process, we have 7 direct infeed stations connected with exactly one baggage carousel and 6 remote infeed stations which are connected to each of the 7 baggage carousels (see Figure 4.1). The penalties $\Psi^{\text{wait}}_{l,e,c,s}$ in the objective function for passengers' waiting times are derived from the survey presented in Section

	Threshold values				
	u_1	u_2	u_3	u_4	u_5
Utilization $\leq$	0.1	0.4	0.8	1	2
Penalties	0.1	1.6	6.4	10	100

Table 4.7: Penalties and threshold values for the utilization of L-IBHP

4.5.1.

In the remainder of this section we first evaluate the performance of the heuristic in Section 4.6.1, before we conduct a real-world study in Section 4.6.1.2 to show the performance of the algorithm when embedded in a rolling planning framework as implemented at a major European Airport.

4.6.1 Theoretical study

To conduct the theoretical study, we create 6 different sets of test instances numbered 'instance set 1' to 'instance set 6'. Each instance set consists of 10 instances based on 10 different days randomly chosen from the real-world data set. For each of the 10 days, a planning interval of $\Delta^{\text{next}} = 180$ minutes is considered with periods of 1 minute length. The number of considered flights in the planning interval for 'instance set 1' to 'instance set 5' increases in segments of 10 flights from 20 to 60 flights. As objective weights we use $\lambda = 0, 0.2, 0.5, 0.8, 1.0$. To find a feasible solution after each touch down of a flight, the runtime of the heuristic is limited to $\Delta^{\text{time}} = 3$ minutes. The presented results for each combination of instance set and λ-value are the average values obtained for the 10 instances contained in each set.

In Section 4.6.1.1, we compare L-IBHP with the HGGLS heuristic proposed in Section 4.4. The performance of each subroutine of the heuristic and the solution quality of the multi-objective path-relinking (see Section 4.5.2) is studied in Section 4.6.1.2.

4.6.1.1 IBHP vs. heuristic

For the comparison of L-IBHP and HGGL we consider 'instance set 1' to 'instance set 4'. The results obtained are presented in Table 4.8 and 4.9, respectively. Column "λ" shows the λ-value for the objective function for each instance set "Inst". In Table 4.8 column "Time (s)" gives the required time in seconds (s) to obtain the objective value in column

“OV”. If no solution is found in 3 minutes, we report the objective value and time of the first feasible solution found within 3 minutes. An instance set where the time limit has been exceeded for at least one of its contained instances is indicated by ‘†’. The objective value of the optimal solution for L-IBHP is given in column “OS” which is obtained by solving L-IBHP without time limit. Column “GAP” shows the gap in % between the objective value of the solution found in column “OV” and the objective value of the optimal solution. In Table 4.9 columns “GAP” and “GAP$^{\text{OS}}$” report the gap in % between the objective value of the heuristic solution and the solution of L-IBHP as reported in column “OV”, and the objective function value of the optimal solution of L-IBHP reported in column “OS”, respectively. The number of periods in which the utilization is within the intervall $]0.0, 0.1]$, $]0.1, 0.4]$, $]0.4, 0.8]$ and $]0.8, 1.0]$ is shown in “utilization” columns 0.1, 0.4, 0.8, 1.0, respectively. If a carousel’s utilization is 0 in a period, it is not taken into account. The percentage of passengers waiting less than 3, 8, 14, 17 and 180 minutes are given in the respective columns “waiting time”.

							utilization $\leq$				waiting time $\leq$ (min)				
λ	Inst	F	OS	OV	Time (s)	GAP	0.1	0.4	0.8	1.0	3	8	14	17	180
0	1	20	2.01	2.01	66.35	0	23	41	21	1	42.5	30.5			16
	2	30	1.97	1.97	180	0	38	40	23		43.5	31.5			15
	3†	40	1.73	1.97	90	39.52	40	63	29	2	5.46	13.08	5.94	5.28	20.31
0.2	1	20	6.78	6.78	104.73	1.52	19	13			16.18	45.63	8.27	4.1	23.88
	2†	30	6.74	12.87	339.7	37.99	28	11	5		16.92	43.55	6.27	3.19	29.64
	3†	40	6.65	9.76	193.44	57.71	27	17	1		13.17	41.25	8.7	6.95	29.79
0.5	1	20	12.08	12.08	38.11	0	15	12			13.79	35.85	17.03		33.14
	2†	30	12.31	25.49	219.39	33.97	22	11	5		14.22	45.78	8.26		31.32
	3†	40	11.95	31.6	190	76.14	21	18	5		7.84	40.01	6.41	8.91	33.07
0.8	1	20	17.31	17.31	35.65	0	14	12			14.18	35.85	17.03		33.14
	2†	30	17.4	31.73	201.44	31.25	20	11			10.95	40.75	10.95		36.35
	3†	40	17.32	46.91	421.32	82	26	19			10.18	33.92	14.04	3.09	36.66
1	1	20	20.8	20.8	40.28	0	15	12				3.14	12.71	15.11	69.37
	2†	30	20.78	22.8	363.01	6.6	20	13				8.04	4.72	5.29	82.11
	3†	40	20.81	43.2	401.2	64.16	24	14	3			2.61	4	5.42	88.33

Table 4.8: Results for L-IBHP

While the average solution time optimizing for different λ values for ‘instance set 1’ using CPLEX is at most 104 seconds, the average time to find a feasible solution for instance sets with more than 10 flights exceeds the time limit of 180 seconds for all $\lambda > 0$. Moreover, the quality of the first solution found after 180 seconds is 0.2% to 61.31% worse than the solution found by the heuristic. Comparing the solution quality of the heurstic with the optimal solutions of L-IBHP, we see that the solution of the heurstic is at most 5.03% worse than the optimal solution. For instance sets 5’ and 6’ with 50 and 60 flights, L-IBHP could not find an optimal or even feasible solution in some of the sets instances within 45 minutes.

						utilization ≤				waiting time ≤ (min)				
λ	Inst	F	OV	GAP^{OS}	GAP	0.1	0.4	0.8	1.0	3	8	14	17	200
0	1	20	2.01	0	0	23	41	21	1	42.5	30.5			16
	2	30	1.97	0	0	34	43	25		43.5	31.5			15
	3	40	1.76	1.73	−10.66	40	63	27	1	45	26.5			12.65
0.2	1	20	6.96	2.65	2.65	17	13			9.95	48.5	13.5	3.65	24.5
	2	30	6.91	2.52	−45.93	17	13			13.2	46	13.2	2.85	26.5
	3	40	6.83	2.71	−30.02	17	13			14	45	16.5	2.5	20.15
0.5	1	20	12.6	4.3	4.3	11	13			4.95	41	22.5	6.5	24.5
	2	30	12.85	4.39	−0.2	12	13			10.75	39.5	20	5.5	23.5
	3	40	12.5	4.6	−60.44	12	13			14	35.5	21.5	4.5	23
0.8	1	20	18.18	5.03	5.03	11	13			6.45	43	19	7	25
	2	30	18.22	4.7	−42.58	12	13			11.7	39	15.25	5.5	27.5
	3	40	18.15	4.8	−61.31	11	13			12	36.5	22	4.5	23
1	1	20	21.7	4.32	4.32	9	13			1.43	14.35	24	13.05	47.05
	2	30	21.7	4.42	−4.82	9	13			1.17	24.5	14	9	51
	3	40	21.75	4.51	−49.65	10	13			2.01	19	48.20	11.3	45

Table 4.9: Results for the HGGLS

The results reveal that minimizing carousels' utilization and passengers' waiting times are conflicting objectives. Considering for example 'instance set 3', minimizing carousels' utilization only, i.e. $\lambda = 1$, a utilization of 0.4 is never exceeded, while up to 45% of the passengers wait longer than 17 minutes and only 2.01% of the passengers wait less than 3 minutes. In contrast, when minimizing passengers' waiting times only, i.e. $\lambda = 0$, carousels' utlization is increased such that a utilization 0.4 is exceeded 28 times, while only at most 12.65% of passengers wait longer than 17 minutes and 45% of passengers wait less than 3 minutes for their bags.

4.6.1.2 Performance of the heuristic

To test the performance of the subroutines GFLS and path-relinking, we use 'instance set 4' to 'instance set 6' with 40, 50 and 60 flights. In Table 4.10, we report the average percental improvements of the constructive solution obtained by the GRASP when applying GFLS and path-relinking subsequently (see rows 4 and 5 in Table 4.10), and the average improvement when applying only path-relinking (see row 6 in Table 4.10).

The initial solution of the construction heuristic is improved by up to 20.61% using both GFLS and path-relinking subsequently, while applying only path-relinking yields an improvement of up to 12.49%.

The last set of rows of Table 4.10 shows the gap in % between the solution obtained for the heuristic, when optimizing the objective with weight λ and the solution obtained when using multi-objective path-relinking (MOPR) between the HGGLS solutions for weight λ^{init} and λ^{inc} with an objective weight of λ. The gap between the heuristic solution and the solution obtained by the multi-objective path-relinking is at most 4.8%. The effectiveness of the MOPR is emphasized when considering the computing times of less than 1 second for all instances. Multi-objective path-relinking is a highly valuable method for the dispatcher to approximate high quality (heuristic) solutions of different weights in short runtime with only small decreases of the solution quality compared to the best heuristic solution found.

4.6.2 Real-world scenarios

In this section we show the performance of the heuristic in real-world scenarios in which the heuristic is embedded in a rolling planning framework as described in Section 4.5.3. We provide the results obtained over 1 historical week, where one day has $1,440$ periods, i.e. the day is partitioned in 1 minute periods. In turn, the rolling planning framework is implemented into a simulation model as described in Appendix A.6. The algorithmic setting is equal to the setting as used in Section 4.6.1 with $\Delta^{\text{next}} = 180$ and $\Delta^{\text{time}} = 3$. In Table 4.11 the heuristic (see rows for "Heuristic") is compared with the solution used at the cooperating Airport (see row "Airport). The heuristic is run for λ-values $0, 0.2, 0.5, 0.8$ and 1. Columns 3 to 6 show in how many time periods, relative to the overall claiming period, the utilization is within the intervall $]0.0, 0.1]$, $]0.1, 0.4]$, $]0.4, 0.8]$ and $]0.8, 1.0]$, respectively. An utilization value of 0 is not taken into account. The percentage of passengers waiting not longer than $3, 8, 14, 17$ and 120 minutes are given in columns 7 to 11. The average maximal utilization as well as the average waiting time of passengers in minutes obtained during the simulation are presented in the columns "Avg max utilization" and "Avg waiting time". In the last column "Var waiting time" we give the variance for the average expected waiting time for passengers according to distribution (67) to passengers' realized waiting time in the simulation. To get valid results, we simulate each λ-value 100 times.

During the simulation we had to optimize 44.5 flights in average in the time intervals. The utilization obtained with the heuristic outperforms the utilization obtained with the manually solution procedure by at least 3.33% ($\lambda = 0$) and by at most 46.67% ($\lambda = 1$). Moreover, with the procedure from the airport, we obtained 3 to 17 times an utilization of 100%, which was never the case when using the heuristic. However, the higher the λ-value the higher the average waiting times for passengers. The heuristic is even worse in terms of average waiting time in comparisson to the aiprort procedure for $\lambda \geq 0.5$. Also the variance of the waiting time increases with increasing λ-values which can be explained by the increased maximal waiting times for passengers when λ is set high. Applying the χ^2-test with the realized waiting times in the simulation and expected waiting times computed with distribution (67) we obtain a significant level of 6%. The coefficient of variance is between 0.3 and 0.4.

4.7 Conclusion

In this chapter we have discussed theoretically and practically the inbound baggage handling at international airports. A MIP formulation minimizing the carousel's utilization and the violation of passengers' level of service quality. The problem complexity was analyzed by considering the assignment problem for the flights' trips to infeed stations and carousels and by considering the objective of obtaining a minimized utilization. Both problems turned out to be NP-hard to solve. To tackle the problem we proposed an hybrid GRASP/GFLS heuristic with path-relinking. Aspects for practical application of the algorithm in a rolling planning framework were discussed. There we present the results of a survey measuring the level of service quality and show a technique for the dispatcher how to get solutions with different objective weights within short time when using path-relinking. In a real-world simulation we showed that our algorithm outperforms the solution quality used at an international European Airport. Based on the results in our simulation study, we set the λ-value for the cooperating Airport to 0.2 and 0. With these weights we never reached an utilization higher than 0.8 and obtained average waiting times of 5.9 minutes. Given some incremented steps between the λ-values 0 and 0.2, we implemented the multi-objective path-relinking (see Section 4.5.2) to calculate, e.g., 4 solution possibilities which can be selected by the dispatcher. To guarantee flexibility, the λ-values as well as the incremented steps between two λ-values for the MOPR can be defined by the dispatcher in the future. Moreover, currently, we conduct surveys in order to evaluate the improvement in terms of passanger satisfaction, especially the impact of the trade-off between passengers' waiting time and passengers' walking distance.

λ	0			0.2			0.5			0.8			1		
Inst	4	5	6	4	5	6	4	5	6	4	5	6	4	5	6
F	40	50	60	40	50	60	40	50	60	40	50	60	40	50	60
GFLS	8.02	9.65	8.45	7.55	8.05	11.6	9.95	9.75	7.65	8.95	6.68	9.01	11.35	13.5	13.6
PR	6.4	5.65	8.7	3.63	7.01	6.85	9.05	7.38	8.9	8.75	6.7	7.85	6.25	5.17	7.9
PR	11.4	11.55	7.23	7.89	9.74	12.20	11.03	5.93	14.55	11.71	7.85	9.85	12.8	10.15	12.49
GAP MOPR															
$\lambda^{\text{init}} = 0.0$, $\lambda^{\text{inc}} = 0.5$				4.7	0.8	1.0									
$\lambda^{\text{init}} = 0.2$, $\lambda^{\text{inc}} = 0.8$							1.1	0.0	1.0						
$\lambda^{\text{init}} = 0.5$, $\lambda^{\text{inc}} = 1.0$										1.9	0.0	4.8			

Table 4.10: Performance of the GFLS and (multi-objective) path-relinking

Solution procedure	λ	$\leq$ utilization 0.1	0.4	0.8	1	$\leq$ waiting time 3	8	14	17	120	Avg max utilization	Avg waiting time	Var waiting time
Heuristic	0	29.6	11.8	1	0.1	32.3	44.5	19.3	2.4	1.5	0.58	5.37	1.16
	0.2	26	7	1		21.82	38.4	22.17	1	1.61	0.37	5.9	2.61
	0.5	26	7.91	0.1		15.45	37.4	26.2	3	18.32	0.35	8.9	3.2
	0.8	24.9	6.5			14	32.2	29	4.2	19.8	0.33	11.6	4.65
	1	15.2	5			6	15.7	23.3	10	45	0.32	16.8	6.8
Airport		31.2	12	3.2	1	24	39.4	31.2	3.7	2.2	0.6	6.7	

Table 4.11: Airport's solution embedded in the simulation

Chapter 5

Conclusion

"Chaque problème que j'ai résolu est devenu une règle qui a servi plus tard pour résoudre d'autres problèmes."

– René Descartes, 1596 – 1650

5.1 Summary and conclusions

This dissertation has studied optimization potentials for baggage handling processes at airports. Chapter 1 provided a general overview which described the baggage flow at airports and separated the baggage handling process into the four subprocesses: check-in baggage, outbound baggage, transfer baggage and inbound baggage flows. It turned out that although a major task at airports, the latter three subprocesses have been studied less in the operations research field. To provide mathematical insight and to show the optimization potentials, we derived a mathematical model formulation, the GASP, covering all four subprocesses. The GASP also led to the most general model formulation for the well known CCPP. It turned out that the GASP is a generalization of the multi-mode resource constrained project scheduling problem (see Kolisch and Drexl [60]) with flexible resource profiles. The complexity of all subprocesses were proven to be NP-complete.

Chapters 3 and 4 presented mathematical models to optimize the outbound baggage handling and inbound baggage handling. For both problems, we faced different problem settings. While for outbound baggage handling a plan for all outgoing flights during the planning day was required, for inbound baggage handling a rolling planning was established to update the plan in short time intervals for a subset of flights arriving in the near future.

As both problems are NP-hard to solve and each model formulation contains a large number of variables and constraints, the model formulations were intractable to solve the specific problem settings for real-world instances.

For the outbound baggage handling problem, we developed an exact solution procedure. We re-formulated the problem by means of Dantzig-Wolfe, which was solved by Branch-and-Price incorporating acceleration techniques. In the computational study we showed that the solution procedure outperforms the TIF as well as a “standard” Branch-and-Price implementation. Our model formulation considering the baggage arrival stream also generated a greater robustness in terms of workload peaks on the carousels than model formulations which do not consider the baggage arrival streams (see Ascó et al. [8]).

As in the inbound baggage handling, we were interested in operative planning to automatize the planning of inbound baggage handling during the course of a day. Therefore, we needed a solution procedure which generated feasible solutions in short time intervals. The HGFLS, which is the first hybridization of GRASP with GFLS, produced solutions of high quality and even outperforms the MIP solutions, when restricting the running time to 3 minutes. As the inbound baggage handling tackles two contrasting objectives, we developed the multi-objective path-relinking, which guarantees a fast examining of solutions for different objective weights. The algorithm for the inbound baggage handling was implemented at a major European Airport, and the results obtained in our computational study were verified in practice.

5.2 Future research

Similar to inbound baggage handling, we develop an algorithm for the operative planning of outbound baggage handling, which is able to update solutions in short time intervals during the course of a day. The algorithm will be embedded in a rolling planning framework and tested in a extensive simulation study. Moreover, we are working on the combination of outbound baggage handling with the planning of the groundhandlers. Based on the plan for outbound baggage handling, the groundhandler assign their personnel to the carousels and working stations. However, due to a lack of worker capacities the groundhandlers are not able to meet the required demand at the carousels, which leads to an under-staffing and an increased workload on the carousels. Through a combined planning of outbound baggage handling and the planning of the workers, an improved baggage handling may be obtained.

For inbound baggage handling the results of our extensive survey revealed the correlation between passengers' satisfaction and passengers' waiting times and walking distance for the planning used at an airport (see Section 4.5.1). We are currently undertaking a second survey at the same airport to show the improvement of passengers' satisfaction following the implementation of our algorithm. We hope to obtain further insight into the benefits of using OR in baggage handling processes.

Appendix A

Appendix: Outbound baggage handling

A.1 Outbound baggage handling: Initial start columns

Algorithm 3 Greedy heuristic

1: $\mathcal{F}^{\preceq} \leftarrow$ `order`$(\mathcal{F})$
2: **while** there is a $i \in \mathcal{F}^{\preceq}$ **do**
3:     set $\mathcal{F}^{\preceq} = \mathcal{F}^{\preceq} \setminus \{i\}$
4:     **if** `assign`(i) **then**
5:         $c^* \leftarrow$ `assign`(i)
6:     **else**
7:         **if** `rightShift`(i) **then**
8:             go to step 4
9:         **else**
10:             add flight i to dummy carousel
11:         **end if**
12:     **end if**
13: **end while**
14: `postProcessing`

A.2 Chebyshev centered restricted primal linear master problem

The Chebyshev centered restricted primal linear master problem is given by

$$\text{minimize} \sum_{d \in \mathcal{D}'(u^{\text{fix}})} H_d \cdot z_d - Z^{\text{LB}} \cdot q \tag{92}$$

subject to

$$\sum_{d \in \mathcal{D}'(u_{\text{fix}})} \Theta^{\text{a}}_{d,i} \cdot z_d - w^{\text{a}}_i - 1 \cdot q \leq 0 \qquad \forall\, i \in \mathcal{F} \tag{93}$$

$$\sum_{d \in \mathcal{D}'(u_{\text{fix}})} \Theta^{\text{s}}_{d,t} \cdot z_d - w^{\text{s}}_t - K^{\text{s}}_t \cdot q \leq 0 \qquad \forall\, t \in \mathcal{T} \tag{94}$$

$$\sum_{d \in \mathcal{D}'_r(u_{\text{fix}})} z_d - |\mathcal{C}_r| \cdot q \leq 0 \qquad r \in \mathcal{R} \tag{95}$$

$$\sum_{d \in \mathcal{D}'(u_{\text{fix}})} \|(\Theta^{\text{a}}_d, \Theta^{\text{s}}_d, 1)\|_2 \cdot z_d + \sum_{i \in \mathcal{F}} w^{\text{a}}_i \cdot \alpha^{\text{a}}_i + \sum_{t \in \mathcal{T}} w^{\text{s}}_t \cdot \alpha^{\text{s}}_t + \|b\|_2 \cdot q \geq 1 \tag{96}$$

$$z_d \geq 0 \qquad \forall\, d \in \mathcal{D}'(u^{\text{fix}}) \tag{97}$$

$$w^{\text{a}}_i \geq 0 \qquad \forall\, i \in \mathcal{F} \tag{98}$$

$$w^{\text{s}} \geq 0 \qquad \forall\, t \in \mathcal{T} \tag{99}$$

$$q \geq 0 \tag{100}$$

A.3 Set-covering heuristic

To build the search tree for the primal heuristic, we use the greedy-based set-covering heuristic shown in Algorithm 4. The heuristic consists of the following three methods:

`selection`($\mathcal{D}^{\text{pos}}$): $\mathcal{D}^{\text{pos}}$ denotes the set of duties which can be selected. The procedure first chooses those duties corresponding to an integer solution, i.e. $z_d = 1$. Then, these duties are selected containing at least one priority flight, i.e. a flight which could not be assigned during the last run of the set-covering heuristic. If there is also no duty with priority flights available, we select duties containing the maximal number of flights not assigned so far in the partial solution. The set of selected duties is given by $\mathcal{D}^{\text{sel}}$.

`storageCapacityCheck`($\mathcal{D}^{\text{sel}}$): After a duty d is added to partial solution $\mathcal{D}^{\text{sel}}$, the heuristic checks violations of the central storage capacity. Let $\mathcal{F}_t^{\text{parT}}$ be the subset of flights handled during period $t \in \mathcal{T}$. If the storage capacity during some period t is violated, flight

$$i_d^* = \arg \max_{i \in \mathcal{F}_t^{\text{parT}}} \left\{ \frac{\Phi^i_{t,\bar{\tau}_i} - \Phi^i_{t,\tau_i}}{P_i \cdot \left(s^{\text{h}}_{\bar{\tau}_i} - s^{\text{h}}_{\tau_i} \right)} \mid \{\tau_i, w_i, P_i\} \text{ is feasible for duty } d \right\}$$

is left-shifted to start time tuple $\tau_i \preceq \bar{\tau}_i$; the number of working stations assigned w_i to flight i is equal to the maximal possible number. We continue left-shifting flights as long as the storage conflict is not solved or there is no flight which can be left-shifted. In the latter case, we delete the flight from a duty with the greatest number of stored bags relative to the consumed parking positions up to period t, i.e. flight

$$i^* = \arg \max_{i \in \mathcal{F}_t^{\text{parT}}} \left\{ \frac{\Phi^i_{t,\bar{\tau}_i}}{P_i \cdot (S_i^E - s^{\text{h}}_{\bar{\tau}_i})} \mid d \in \mathcal{D}^{\text{sel}}(u^{\text{fix}}) \wedge t < S_i^{\text{E}} \right\}$$

is deleted from its duty d and is marked as not assigned again.

`add`($\mathcal{D}^{\text{sel}}$,$\mathcal{F}^{\text{nA}}$): Flights $\mathcal{F}^{\text{nA}}$, not added at the end of the heuristic, are assigned to one of the selected duties $\mathcal{D}^{\text{sel}}$. The flights are added in decreasing order of the number of required parking positions. Flights not added are collected in a priority set.

Algorithm 4 Outbound baggage handling: Set-covering heuristic

```
set D^pos = D'(u_fix) and D^sel = ∅; F^nA = F
while D^pos ≠ ∅ do
    d ← selection(D^pos_{u_fix})
    set D^pos_{u_fix} = D^pos_{u_fix} \ {d}, C_{r_d} = C_{r_d} − 1, F^nA = F\F_d
    if C_{r_d} = 0 then
        delete all duties d of type r from duty set D^pos
    end if
    F^nA ← storageCapacityCheck(D^sel)
end while
add(D^sel,F^nA)
```

All columns generated new during one run of the set-covering heuristic are added to the RMP.

Appendix: Inbound baggage handling

A.4 Linearization of IBHP

$$\text{Minimize } f^{\text{lin}}(\mathbf{x}) = \lambda \cdot \sum_{1 \le k \le K} \sum_{c \in \mathcal{C}} \sum_{t \in \mathcal{T}} p_k^{\text{u}} \cdot z_{k,c,t}^{\text{u}} + (1-\lambda) \cdot \cdots \tag{1''}$$

$$\text{subject to } (69) - (83), (85) - (88)$$

$$\frac{1}{K_c^{\text{cb}}} \sum_{l \in \mathcal{L}} \sum_{e \in \mathcal{E}_c} \sum_{S_{l,e}^{\text{es}} \le s \le S_{l,e}^{\text{ls}}} \Phi_{l,e,c,s,t} \cdot x_{l,e,c,s} - \sum_{1 \le k \le K} u_k \cdot z_{k,c,t}^{\text{u}} \le 0 \qquad \forall\ c \in \mathcal{C}, t \in \mathcal{T} \tag{84'}$$

$$\sum_{1 \le k \le K} z_{k,c,t}^{\text{u}} \le 1 \qquad \forall\ c \in \mathcal{C}, t \in \mathcal{T} \tag{101}$$

$$z_{k,c,t}^{\text{u}} \in \{0,1\} \qquad \forall\ 1 \le k \le K, c \in \mathcal{C}, t \in \mathcal{T} \tag{102}$$

A.5 Measuring the service level

We interviewed 386 passengers during the baggage pick up process over 7 days in a week from 6 a.m. to 10 p.m.. We asked the passengers to assess a) to c) on a scale from 1 to 6, where 1 means very satisfied and 6 very unsatisfied. We also asked the passengers to assess the overall satisfaction. Due to the spacious claiming hall at our cooperation partner, it turned out that only a) and b) influence passengers satisfaction. The first two rows of Table A.1 show the adjusted-R^2 and the root mean squared error between the given assessment of objectives a), b) and passengers' true walking distances and waiting times, respectively. The results reveal a stronger correlation between waiting time and

Assessment	**True**	**adj-R²**	**RMSE**
Walking distance	Walking distance	0.05	235.94
Waiting time	Waiting time	0.35	4.98
Overall satisfaction	Walking distance	0.08	1.07
	Waiting time	0.56	0.74

Table A.1: Results of the survey conducted at cooperation partner

assessment, than between walking distance and assessment. Moreover, the correlation between passengers' overall satisfaction and passengers' true waiting time is also also higher than the correlation between passengers' overall satisfaction and passengers walking distance (see the second part of Table A.1).

Short walking distances from flights' parking positions to baggage carousels and short waiting times for the passengers' baggage at the baggage carousels are conflicting objectives. For example, a long walking distance from flight's parking position to the baggage carousel may result in a short waiting time for passengers, as it gives plenty of time for the infeed process. Whereas, a short walking distance has the contrary effect, i.e. the waiting time for passengers at the baggage carousels become longer. Asking directly the passenger for shorter waiting times or walking distances, 63% of all interviewed passengers prefere shorter waiting times instead of shorter walking distances.

A.6 Simulation environment

The discrete event simulation model is implemented in JAVA with the package SSJ. The smallest entities are bags and passengers, where we assign a subset of bags to exactly one passenger, i.e. only when the specific passenger arrives at the assigned baggage carousel the corresponding bags are picked up. The time accuracy of the simulation model is in seconds. The input data are based on the historical data provided by a major European Airport. While the first 10 parameters of Table 4.2 have exactly the value as in the historical days, the last 6 parameters are approximated. In the following we describe the primary event modules (EVENT) of the simulation. If we state intervals for the duration or the schedule of an event, we uniformly draw the realized value in the simulation out of these intervals.

Infeed process During the simulation, the historical expected on-block times are used for flights which have not arrived at the airport. These expected on-block times are updated during the simulation run with the historical updates for the expected on-block time(ARRIVALUDPATE). Once a flight is on-block (ONBLOCK), event TRIPSTART is initiated, which simulates the arrival time of the baggage tug at the assigned infeed station. For the travel time we use the distance matrix of a European Airport and assume a travel speed between 25 to 30 km/h. Once, the infeed station is available, the infeed of the trip starts (INFEEDBEGIN) and the baggage of the containers is placed on the conveyor belt (INFEEDBAG). Afterward, the bags are transported to the assigned carousel (BAGTRANSITION). If the assigned carousel has capacity left, the bag is placed on the conveyor belt (BAGARRIVAL), else the bag remains either in the BHS (remote infeed stations) or on the infeed station's conveyor belt (direct infeed station) until the conveyor belt has space for additional bags. When the bag can be sent to the assigned carousel or a remote infeed station is used, event INFEEDBAG is called again, if bags are left in the container, else the next trip waiting at the infeed station starts its infeed by calling event INFEEDBEGIN. A randomized delay between two calls of event INFEEDBAG varying between 5 and 7 seconds simulates the loading rate.

Claiming process Once a flight is on-block it take between 3 and 8 minutes until a passenger can disembark (PASSENGERDISEMBARK). Passenger's walking speed from varies between 4 and 7 km/h. For transcontinental flights we additional add an offset to simulate costumes which takes between 1 and 2 minutes. Together with the distance matrix from a flight's parking position to the carousels, event PASSENGERARRIVAL is scheduled. As soon as the passenger arrives at the baggage carousel he can either pick up his bags, if the bags have arrived at the carousel, or the passenger has to wait. If the passenger is at the carousel and the bag arrives, he removes the bag after an offset between 10 and 40 seconds (REMOVEBAG). As soon as all bags are picked up by the passenger he leaves the baggage claim hall (PASSENGERLEAVE).

A.7 Pseudo code for the GFLS

Function	**Description**
h^{cstr}	cost function in constructive phase
h	augmented cost function in improvement phase
Solution	**Description**
$\mathbf{x}^{best}$	best solution found based on function h^{cstr}
$\mathbf{x}$	current iteration best solution found based on function h^{cstr}
$\mathbf{x}^{\text{ls}}$	current local search solution
$\mathbf{x}^{\text{next}}$	next solution in a neighborhood of solution $\mathbf{x}^{\text{ls}}$
$\mathbf{x}^{\text{lsb}}$	current local search iteration best solution

Table A.2: Notation for Algorithm 5

Algorithm 5 GFLS for Neighborhood $\mathcal{N}^k(\mathbf{x})$ for $k = 1, 2$

```
x^ls ← x
x^lsb ← x
while ActiveNeighborhood do
    N^k(x^l) ← NextActiveNeighborhood // Choose next active neighborhood (see Sec. 4.4.2)
    while there is a x^next ∈ N^k(x^ls) and x^next is not explored do
        if h^cstr(x^next) < h^cstr(x) then
            x ← x^next
            if h^cstr(x) < h^cstr(x^best) then
                x^best ← x
            end if
        end if
    end while
    if h(x^ls) < h(x^lsb) then
        x^lsb ← x^ls
        ReactivateNeighborhoods // Reactivate neighborhood (see Sec. 4.4.2)
    else
        DeactivateNeighborhood // Deactivate current neighborhood (see Sec. 4.4.2)
        UpdateFeaturePenalties // Update features' penalties (see Sec. 4.4.2)
    end if
end while
```

Appendix B

Abbreviations

Alg1-CG	Algorithm 1 with a standard column generation implementation
Alg1-CCP	Algorithm 1 with the Chebychev cutting-plane method
Alg1-NCCP	Algorithm 1 without the Chebychev cutting-plane method
App.	appendix
BC	baggage carousel
BHS	baggage handling system
BSM	baggage source massage
CG	column generation
CCPP	check-in counter planning problem
CRMP	Chebyshev centered restricted primal master problem
DP	decision problem
DSP	descent scheduling problem
FCFS	first come, first serve
FLS	fast local search
FSP	flight scheduling problem
GASP	generic assignment and scheduling problem
GFLS	guided fast local search
GRASP	greedy randomized adaptive search procedure

HGGLS	hybridization of greedy randomized adaptive search procedure with guided fast local search
IBHP	inbound baggage handling problem
LP	linear programming
L-CRMP	linearized Chebyshev centered restricted primal master problem
L-IBHP	linearized inbound baggage handling problem
L-RMP	linearized restricted pricing problem
MIP	mixed integer program
MORP	multi-objective path-relinking
MP	master problem
NP	non-deterministic polynomial
OBHP	outbound baggage handling problem
OR	operations research
PP	pricing problem
PR	path-relinking
RDS	ramp direct service
RMP	restricted pricing problem
RMSE	root mean squared error
RSP	restricted subproblem
SCG	standard column generation
Sec.	section
SGASP	simplified generic assignment and scheduling problem
TIF	time indexed formulation
ULD	unit load device

Appendix C

Mathematical symbols

General

Sets

$\mathbb{R}$	real numbers
$\mathbb{R}^n_+$	non-negative real space of dimension n
$\mathbb{R}^n_-$	non-positive real space of dimension n
$\mathbb{Z}$	integer numbers
$\mathbb{Z}_+$	non-negative integer numbers

Functions

$\lvert\mathcal{A}\rvert$	power of a set $\mathcal{A}$
$\mathbb{1}_{\mathcal{A}}(x)$	1, if $x \in \mathcal{A}$, and 0 otherwise
$\mathbb{E}(A)$	expectation value of random variable A
$(x)^+$	maximum between x and 0

Operators

$\wedge$	logical “and”
$\vee$	logical “or”

Baggage flows at airports: A survey and generic a model

Sets

$\mathcal{B}$ .... bins
$\mathcal{F}$ .... jobs/flights
$\mathcal{G}$ .... resources
$\mathcal{J}^{\mathrm{BP}}$ .... items to be packed into a bin
$\mathcal{J}^{\mathrm{MS}}$ .... jobs to be scheduled on machines
$\mathcal{M}$ .... modes
$\mathcal{M}_i$ .... modes for job/flight i
$\mathcal{T}$ .... planning horizon

Parameters

F .... number of jobs/flights
G .... number of resources
$L_{i,t}$ .... number of required resources by job i at time t
M .... number of modes
$p_{i,m,g}$ .... minimum processing time of job i at resource g in mode m
$r_{i,m,g}$ .... resource usage of job i for resource g in mode m
S_i^{es} .... earliest start time for job i
S_i^{ls} .... latest start time for job i
S_i^{e} .... end time for job i
.... T end of planing horizon
U_g .... capacity of resource g
U^s .... capacity of the central storage system
λ^+ .... general infeed rate

λ^- .... number of passengers or bags that can be handled per time period
λ^+g .... infeed rate at infeed station g
$\lambda^+_{i,t}$ .... number of arriving passengers or bags for flight i at time t
$\lambda^-_{i,g,t}$ .... depletion rate of passengers at time t, if flight i is assigned to baggage carousel g

Variables

$B_{i,t}$ .... number of stored bags for flight i in period t
$I_{i,g,t}$ .... number of passengers or bags of job/flight i at resource g at the end of period t
$x_{i,m,t}$ .... 1, if processing of job i in mode m starts at the beginning of period t, and 0 otherwise
$z_{i,g,t}$ .... number of allocated resources of type g for job/flight i in period t

Functions

$f^{\mathrm{I,z}}(I_{i,g,t}, z_{i,g,t})$ .... cost function for variable $I_{i,g,t}$ and $z_{i,g,t}$
$f^{\mathrm{x}}(x_{i,g,t})$ .... cost function for variable $x_{i,g,t}$
$g(z_{i,g,t})$ .... in- and outflow of period t resulting from resource allocation $z_{i,g,t}$

Outbound baggage handling

Sets

$\mathcal{C}$ .... carousels
$\mathcal{C}_r$ .... carousels of type r

$\mathcal{C}^{\text{pos}}$	possible carousels where a flight can be assigned to
$\mathcal{CS}$	cut set
$\mathcal{D}_r(u^{\text{fix}})$	duties of carousel type r respecting upper bound u^{fix}
$\mathcal{D}'(u^{\text{fix}})$	restricted duty subset respecting upper bound u^{fix}
$\mathcal{D}'_r(u^{\text{fix}})$	restricted duty subset of carousel type r respecting upper bound u^{fix}
$\mathcal{D}(u^{\text{fix}})$	duties respecting upper bound u^{fix}
$\mathcal{D}^{\text{pos}}$	duties which can be selected in the next iteration in the set-covering heuristic
$\mathcal{D}^{\text{sel}}$	selected duties in the set-covering heuristic
$\mathcal{E}$	elements
$\mathcal{F}$	outgoing flights
$\mathcal{F}_c^{\text{as}}$	flights assigned to carousel c
$\mathcal{F}^{\text{nAs}}$	not assigned flights in the the set-covering heuristic
$\mathcal{F}_t^{\text{parT}}$	flights handled in parallel during period t
$\mathcal{F}_i^{\text{parF}}$	flights handled in parallel with flight i
$\mathcal{F}^{\preceq}$	ordered flights according to flights' ending time for the baggage handling
$\mathcal{R}$	carousel types
$\mathcal{S}_i$	feasible start time tuples for flight i
$\mathcal{S}_i(w)$	feasible start time tuples for flight i, if w working stations are assigned
$\mathcal{T}$	planning horizon
$\mathcal{T}_m$	planning horizon of time segment m
$\mathcal{T}_m(u^{\text{fix}})$	time points in time interval m in which utilization u^{fix} is reached
$\mathcal{U}_m$	threshold values in time segment m

Parameters

$A_i = (A_{i,t})_{t\in\mathcal{T}}$	estimated baggage arrival vector for flight i
A_i^{early}	amount of baggage for flight i which has arrived before time 0
$A_i^{\text{h}} = \left(A_{i,t}^{\text{h}}\right)_{t\in\mathcal{T}}$	historical baggage arrival vector for flight i
B	upper bound
C	number of carousels
C_r	number of carousels of type r
F	number of flights
H_d	penalty value of duty d for threshold violations summed up over all time intervals
K	number of threshold values for the utilization of a carousel
K_r^{cb}	conveyor belt capacity of carousel type r
K_r^{pp}	number of parking positions at carousel type r
K_r^{ppws}	number of parking positions in a working stations segment at carousel type r
K^{s}	capacity of the central storage system
K_t^{s}	capacity of the central storage system at time t
K_r^{ws}	number of working stations at carousel type r
M	number of time intervals
P_i	required bulk-containers for flight i at a carousel
p_k	penalty for violating utilization threshold value k
R	number of carousel types
S_i^{e}	end time of flight i's baggage handling
S_i^{es}	earliest start time for the baggage handling of flight i
S_i^{ls}	latest start time for the baggage handling of flight i
T	end of planning horizon
u_k	threshold values for carousels' utilization
u^{fix}	upper bound for the carousel utilization

u^{ta} .... target utilization

$W_i^{\max}$ .... maximal number of possible working stations for flight i across all carousel types

$W_{i,r}^{\max}$ .... maximal number of possible working stations for flight i at carousel type r

$W_i^{\min}$ .... minimal number of required working stations for flight i across all carousel types

$W_{i,r}^{\min}$ .... minimal number of required working stations for flight i at carousel type r

Z^{LB} .... lower bound for the restricted master problem

$\Gamma_{t,\tau}^{i,w}$ .... workload of flight i at time t, if start time tuple τ is selected and w working stations are assigned

$\Phi_{t,\tau}^{i}$ .... amount of stored baggage for flight i at time t, if start time tuple τ is selected

$\Theta_d^{\text{a}} = \left(\Theta_{d,i}^{\text{a}}\right)_{i\in\mathcal{F}}$ .... flight vector of duty d, which equal to 1 at element i, if flight i is assigned to duty d, and 0 otherwise

$\Theta_d^{\text{s}} = \left(\Theta_{d,t}^{\text{a}}\right)_{t\in\mathcal{T}}$ .... number of bags in the central storage of flights assigned to duty d during time period t

$\rho_{r,i,w,\tau}$ .... sum of dual values for the flight assignments and storage capacity

Variables

s_τ^{h} .... start time for the baggage handling of start time tuple τ

s_τ^{d} .... start time for the storage depletion of start time tuple τ

$x_{i,c,w,\tau}$ .... 1, if flight i is assigned to carousel c with w working stations assigned and start time tuple τ is selected, and 0 otherwise

$x_{d,r} = \left(x_{d,r,i,w,\tau}\right)_{i\in\mathcal{F},\tau\in\mathcal{S}_i(w),W_{i,r}^{\min}\le w\le W_{i,r}^{\max}}$ binary vector with $x_{d,r,i,w} = 1$, if flight i belongs to duty d with w working stations assigned to flight i at a carousel of type r and start time tuple τ is selected, and 0 otherwise

$y_{c,k,m}$ .... 1, if the utilization of carousel c exceeds target utilization u^{ta} by u_k in time interval m, and 0 otherwise

$y_{d,r} = (y_{d,r,k,m})_{1\leq k\leq K, 1\leq m\leq M}$ .... binary vector with $y_{d,r,k,m} = 1$, if duty d of type r causes a violation of u_k in time interval m, and 0 otherwise

z_d .... 1, if duty d is selected, and 0 otherwise

$\lambda = (\lambda_i)_{i\in\mathcal{F}}$ .... dual vector for flight assignment constraints

$\mu = (\mu_t)_{t\in\mathcal{T}}$ .... dual vector for the storage constraints

$\nu = (\nu_r)_{r\in\mathcal{R}}$ .... dual vector for the number of carousel types

Functions

$a(i)$ .... underestimation of the baggage arrival amount for flight i

$\bar{a}$ .... average underestimation of the baggage arrival

$rc_r(x_{d,r}, y_{d,r})$ .... reduced costs for carousel type r

$s(e)$ .... size of element e

$z_{\text{DSP}}(r,t)$ .... minimal objective value for the descent scheduling problem for carousel type r at time t

$\preceq$ .... relation for start time tuples

Inbound baggage handling

Sets

$\mathcal{A}$ .... arcs

$\mathcal{C}$ .... baggage carousels

$\mathcal{C}_e$ .... baggage carousels connected to infeed station e

$\mathcal{E}$ .... infeed stations
$\mathcal{E}_c$ .... infeed stations connected to baggage carousel c
$\mathcal{F}$ .... flights with inbound baggage
$\mathcal{I}$ .... items
$\mathcal{L}$ .... trips
$\mathcal{L}_i$ .... trips for flight i
$\mathcal{L}^d$ .... trips including dummy trips
$\mathcal{N}^1(\mathbf{x})$ .... neighborhood 1 of solution $\mathbf{x}$
$\mathcal{N}^2(\mathbf{x})$ .... neighborhood 2 of solution $\mathbf{x}$
$\mathcal{RC}$ .... restricted candidate list
$\mathcal{T}$ .... planning horizon
$\mathcal{V}$ .... vertices
Φ .... features

Parameters

$A_{i,c}$ .... arrival time of the first passenger of flight i at baggage carousel c
B_l .... number of inbound bags of trip l
B_i^{tot} .... total number of inbound bags of flight i
C .... number of baggage carousels
d^{walk} .... realized walking distance
E .... number of infeed station
F .... number of flights with inbound baggage
K_c^{cb} .... conveyor belt capacity of baggage carousel c
K_c^{di} .... display capacity of baggage carousel c
L .... number of trips
L_i .... number of trips for flight i
l_e^0 .... start dummy trip for infeed station e
l_e^{L+1} .... end dummy trip for infeed station e
N_i .... maximal number of bags carried by one passenger of flight i

P_i .... number of passengers of flight i
$p_i(n)$ .... percentage of passengers of flight i carrying n inbound bags
p_k^{u} .... penalty for violating threshold value u_k
p_c^{cap} .... penalty for the capacity violation at baggage carousel c
$p_{i,l}^{\mathrm{ls}}$ .... penalty for the violation of the latest infeed restriction for trip l
S_i^{ob} .... on block time of flight i
$S_{l,e}^{\mathrm{es}}$ .... earliest start time for the infeed of trip l at infeed station e
$S_{l,e}^{\mathrm{ls}}$ .... latest start time for the infeed of trip l at infeed station e
$t_{l,e,c,s}^{\mathrm{first}}$ .... period when the first bag of trip l arrives on baggage carousel c, if infeed station e is selected and the infeed process starts at time s
u_k .... threshold value for the capacities at baggage carousels
u_c^{cap} .... utility for the capacity violation of carousel c
$u_{i,l}^{\mathrm{ls}}$ .... utility for the violation of the latest infeed restriction for trip l
$\mathbf{x}$ .... solution representation
$\mathbf{x}^{\mathrm{best}}$ .... best solution found in the constructive phase
$\mathbf{x}^{\mathrm{dec}}$ .... solution for a decrease weight in the multi-objective path-relinking
$\mathbf{x}^{\mathrm{inc}}$ .... solution for a increase weight in the multi-objective path-relinking
$\mathbf{x}^{\mathrm{init}}$ .... initial solution for the multi-objective path-relinking
$\mathbf{x}^{\mathrm{ls}}$ .... current local search solution
$\mathbf{x}^{\mathrm{lsb}}$ .... current local search iteration best solution
$\mathbf{x}^{\mathrm{next}}$ .... next solution in the neighborhood of solution $\mathbf{x}^{\mathrm{ls}}$
$\mathbf{x}^{\mathrm{new}}$ .... new solution in the multi-objective path-relinking

$\mathbf{x}^i$	partial solution with i flights assigned in the constructive phase
$\mathbf{x}^*$	optimal solution
$\Delta^{\text{claim}}_{i,e,c,s}$	baggage claim duration for flight i, if infeed station e and baggage carousel c are selected and the infeed process starts at time s
$\Delta^{\text{dur}}_{e,c,}$	duration to transfer a bag from infeed station e to baggage carousel c
Δ^{inf}_{l}	infeed duration of trip l
$\Delta^{\text{lag}}_{l,h,e}$	minimum time lag between trip l and h at infeed station e
Δ^{next}	duration planned in advanced in a rolling planning framework
Δ^{pl}	duration to place containers on a baggage tug
Δ^{time}	computing time of an optimization algorithm in a rolling planning framework
$\Delta^{\text{trip}}_{i,e}$	duration to drive from flight i's parking position to infeed station e
Δ^{wait}	realized waiting time
λ	weight for the utilization
λ^{dec}	decreased weight for the utilization for the multi-objective path-relinking
λ^{inc}	increased weight for the utilization for the multi-objective path-relinking
λ^{init}	initial weight for the utilization for the multi-objective path-relinking
μ^{w}	baggage loading rate of workers at an infeed station
μ^{p}_{i}	arrival rate of passengers of flight i
$\Psi^{\text{wait}}_{l,e,c,s}$	waiting time penalty for passengers of trip l, if infeed station e and baggage carousel c are selected and the infeed process starts at time s
$\Psi^{\text{walk}}_{i,c}$	walking distance penalty for passengers of flight

	i, if baggage carousel c is selected
$\Phi_{l,e,c,s,t}$	number of bags of trip l on baggage carousel c at time t, if the infeed station e is selected and the infeed process starts at time s

Variables

$e_{i,c,t}$	1, if flight i's claiming period ends on baggage carousel c at time t, and 0 otherwise
$u_{c,t}$	utilization of baggage carousel c
$x_{l,e,c,s}$	1, if trip l starts the infeed at s and infeed station e and baggage carousel c is selected, and 0 otherwise
$y_{l,h,e}$	1, if trip l is handled before trip h at infeed station e, and 0 otherwise

Functions

$a_{i,c}(t)$	percentage of passengers arriving at baggage carousel c in period t
$\bar{a}_{i,c}(t)$	percentage of newly arriving passengers in period t
$B_{i,e,c}(t)$	number of bags for flight i which have been sent from infeed station e to carousel c up to period t
$b_{l,e,s,t}(t)$	percentage of bags arriving at baggage carousel c for trip l in period t
$f(\mathbf{x})$	objective function in the inbound baggage handling problem
$g(\bullet)$	utility function for passengers' realized waiting time or walking distance
$h(\mathbf{x})$	augmented cost function
$h^{\text{cstr}}(\mathbf{x})$	cost function in the constructive phase

$I_c^{\text{cap}}(\mathbf{x})$	1, if the number of flights which can be assigned to baggage carousel c is exceeded
$I_c^{\text{ls}}(\mathbf{x})$	1,if the infeed of flight i's trip l starts later than the latest possible infeed start time
$p_{l,e,c,s}^{\text{wait}}(t)$	probability that a passenger picks up all his bags at baggage carousel c in period t, if the baggage handling starts at infeed station e at time s
$\preceq_e$	infeed order of trips at infeed station e
$\cong_e$	arrival time of 2 trips at infeed station e is in the same period

Bibliography

[1] A. Abdelghany and K. Abdelghany. *Modeling Applications in the Airline Industry.* Ashgate Publishing, Ltd., 2009.

[2] A. Abdelghany, K. Abdelghany and R. Narasimhan. Scheduling baggage-handling facilities in congested airports. *Journal of Air Transport Management*, 12(2):76–81, 2006.

[3] M. Akbar, M. Rahman, M. Kaykobad, E. Manning and G. Shoja. Solving the multidimensional multiple-choice knapsack problem by constructing convex hulls. *Computers & Operations Research*, 33(5):1259 – 1273, 2006.

[4] R. Alvarez-Valdes, E. Cresp, J. Tamarit and F. Villa. GRASP and path relinking for project scheduling under partially renewable resources. *European Journal of Operational Research*, 189:1153 – 1170, 2008.

[5] H. Amor, J. Desrosiers and J. Valério de Carvalho. Dual-optimal inequalities for stabilized column generation. *Operations Research*, 54(3):454 – 463, 2006.

[6] S. Appelt, R. Bata, L. L. and C. Drury. Simulation of passenger check-in at a medium-sized us airport. In *Proceedings of the 2007 Winter Simulation Conference*, pages 1252–1260. IEEE, 2007.

[7] A. Ascó, J. A. D. Atkin and E. Burke. The airport baggage sorting station allocation problem. In *Proceedings of the 5th Multidisciplinary International Scheduling Conference*, pages 419–444, 2011.

[8] A. Ascó, J. Atkin and E. Burke. An analysis of constructive algorithms for the airport baggage sorting station assignment problem. *Journal of Scheduling*, pages 1–19, 2013.

[9] N. Ashford, M. Stanton and C. Moore. *Airport Operations*. McGraw-Hill, 2nd Ed., 1997.

[10] E. Bachmat and M. Elkin. Bounds on the performance of back-to-front airplane boarding policies. *Operations Research Letters*, 36(5):597–601, 2008.

[11] E. Bachmat, D. Berend, L. Sapir and S. Skiena. Airplane boarding, disk scheduing and spacetime geometry. In *Proceedings of the Algorithmic Applications in Management*, pages 192–202, Berlin, 2005.

[12] E. Bachmat, D. Berend, L. Sapir and S. Skiena. Optimal boarding policies for thin passengers. *Advances in Applied Probability*, 39(4):1098–1114, 2007.

[13] E. Bachmat, D. Berend, L. Sapir, S. Skiena and N. Stolyarov. Analysis of airplane boarding times. *Operations Research*, 57(2):499–513, 2009.

[14] H. Balakrishnan and B. Chandran. Algorithms for scheduling runway operations under constrained position shifting. *Operations Research*, 58(6):1650–1665, 2010.

[15] W. Barbo. The use of queueing models in design of beggage claim areas at airports. Technical report, Institute of Transportation Studies, University of California, Berkeley, 1967.

[16] T. Barth and D. Pisinger. Scheduling of outbound luggage handling at airports. In D. Klatte, H.-J. Lüthi and K. Schmedders, editors, *Operations Research Proceedings 2011*, pages 251–256. Springer, 2012.

[17] D. Bertsimas and R. Demir. An approximation dynamic programming approach to multidimensional knapsack problems. *Management Science*, 48(4):550 – 565, 2002.

[18] G. Bitram and H. Yanasse. Computational complexity of the capacitated lot size problem. *Management Science*, 28(10):1174 – 1186, 1982.

[19] J. Browne, J. Kelley and P. Le Bourgeois. Maximum inventories in baggage claim: A double ended queuing system. *Transportation Science*, 4(1):64–78, 1970.

[20] G. Bruno and A. Genovese. A mathematical model for the optimization of the airport check-in service problem. *Electronic Notes in Discrete Mathematics*, 36: 703–710, 2010.

[21] H. Chun. Solving check-in counter constraints with ILOG solver. In *Proceedings of the 1st ILOG Solver and ILOG Schedule Int. Users Meeting*, Abbaye des Vaux de Cernay, France, 1995.

[22] H. Chun. Representing constraints in airport resource alloaction. In *Proceedings of the 8th Int. Symp. Artificial Intellegence*, Monterrey, Mexico, 1995.

[23] H. Chun. Scheduling as a multi-dimensional placement problem. *Engeneering Application of Artificial Intellegence*, 9(3):261–273, 1996.

[24] H. Chun and R. Mak. Intelligent resource simulation for an airport check-in counter allocation system. *IEEE Transactions on Systems, MAN, and Cybernetics, Part C: Applications and Reviews*, 29(3):325 – 335, 1999.

[25] C. Chung and T. Sodeinde. Simulataneous service approach for reducing air passenger queue time. *Journal of Transportation Engineering*, 126(1):85 – 88, 2000.

[26] T. Clausen and D. Pisinger. Dynamic routing of short transfer baggage. Technical report, DTU Management Engineering, 2010.

[27] A. R. Correia and S. C. Wirasinghe. Level of service analysis for airport baggage claim with a case study of the calgary international airport. *Journal of Advanced Transportation*, 44(2):103 – 112, 2010.

[28] A. de Barros and D. Tomber. Quantitative analysis of passenger and baggage security screening at airports. *Journal of Advanced Transportation*, 41(2):171–193, 2007.

[29] A. de Barros and S. Wirasinghe. Sizing the baggage claim area for the new large aircraft. *ASCE Journal of Transportation Engineering*, 130(3):274 – 279, 2004.

[30] R. de Neufville and A. Odoni. *Airport Systems: Planning Design and Management*. McGraw-Hill, 2002.

[31] G. Desaulniers, J. Desrosiers and M. Solomon, editors. *Column Generation*. Springer, 1 Ed., 2005.

[32] U. Dorndorf, A. Drexl, Y. Nikulin and E. Pesch. Flight gate scheduling: State-of-the-art and recent developments. *Omega*, 35(3):326–334, 2007.

[33] S. Elhedhli, L. Li, M. Gzara and J. Naoum-Sawaya. A branch-and-price algorithm for the bin-packing problem with conflicts. *Journal on Computing*, 23(3):404–415, 2011.

[34] O. Faroe, D. Pisinger and M. Zachariasen. Guided local search for the three-dimensional bin-packing problem. *Journal on Computing*, 15(3):267–283, 2003.

[35] R. Felkel. Planning check-in facilities for the future at Frankfurt Airport. *Journal of Airport Management*, 2(2):142–152, 2008.

[36] T. Feo and J. Bard. Flight scheduling and maintenance base planning. *Management Science*, 35(12):1415 – 1432, 1989.

[37] D. Field. Remote control. *Airline Business*, 24(1):45–47, 2008.

[38] M. Florian, J. Lenstra and A. Rinnooy Kan. Deterministic production planning algorithm and complexity. *Management Science*, 26(7):669 – 679, 1980.

[39] Flughafen München GmbH. Ausbau Terminal 2. `http://www.munich-airport.de/de/company/konzern/ausbau/satellit/index.jsp`.

[40] Flughafen München GmbH. 1992 und 2012 im Vergleich. `http://www.munich-airport.de/de/micro/20jahre/statistik/index.jsp`, 2014.

[41] M. Frey and R. Kolisch. Scheduling of outbound baggage at airports. In *Proceedings of the 12th International Conference on Project Management and Scheduling*, pages 187–190, Tours, France, 2010.

[42] M. Frey, C. Artigues, R. Kolisch and P. Lopez. Scheduling and planning the outbound baggage process at international airports. In *Proceedings of the 2010 IEEE International Conference on Industrial Engineering and Engineering Management*, pages 2129–2133, 2010.

[43] M. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York, 1979.

[44] A. Ghobrial, C. F. Daganzo and T. Kazimi. Baggage claim area congestion at airports: An empirical model of mechanized claim device performance. *Transportation Science*, 16(2):246–260, May 1982.

[45] C. Gillispie. The montgolfier brothers and the invention of aviation, 1783-1784; with a word on the importance of ballooning for the science of heat and the art of building railroads. In J. Langins, editor, *Technology and Culture*, volume 26, pages 85–87. The Johns Hopkins University Press, 1985.

[46] G. Gosling. *An economic framework for the planning of airport passenger terminals*. PhD thesis, Departement of Civil Engineering and Institute of Transportation Studies, University of California, Barkeley, 1979.

[47] K. Hallenborg and Y. Demazeau. DECIDE: Applying multi-agent design and decision logic to a baggage handling system. *Engineering Environment-Mediated Multi-Agent Systems*, 5049:148–165, 2008.

[48] S. Heinz and D. Pitfield. British Airways' move to terminal 5 at London Heathrow Airport: A statistical analysis of transfer baggage performance. *Journal of Air Transport Management*, 17(2):101–105, 2011.

[49] T. Heppenheimer. *First Flight: The Wright Brothers and the Invention of the Airplane*. Wiley, 2003.

[50] R. Horonjeff. Analyses of passenger and baggage flows in airport terminal buildings. *Journal of Aircraft*, 6(5):446–451, 1969.

[51] M. Janic. Assessment and management of quality of service at an aiport passenger terminal. *Transportation Planning and Technology*, 26(3):239 – 263, 2003.

[52] M. Janic. *Airport Analysis, Planing and Design: Demand, Capacity and Congestion*. Nova Science Publishers Inc, 2009.

[53] J. Jarell. Self-service kiosks: Museum pieces or here to stay? *Jounral of Airport Management*, 2(1):23–29, 2007.

[54] A. Johnson. Multi-service costing. *Journal of Operational Research Society*, 29(6): 551 – 558, 1978.

[55] M. Johnstone, D. Creighton and S. Nahavandi. Status-based routing in baggage handling systems: Searching versus learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 40(2):189–200, 2010.

[56] C. Joncour, S. Michel, R. Sadykov, D. Sverdlov and F. Vanderbeck. Column generation based primal heuristics. *Electronic Notes in Discrete Mathematics*, 36(0): 695–702, 2010.

[57] P. Joustra and N. Dijk. Simulation of check-in at airports. In *Proceedings of the 2001 Winter Simulation Conference*, pages 1023–1028, 2001.

[58] S. Khan, K. Li, E. Manning and M. Akbar. Solving the knapsack problem for adaptive multimedia system. *Studia Informatica Universalis*, 2:161 – 182, 2002.

[59] A. Khosravi, S. Nahavandi and D. Creighton. Interpreting and modeling baggage handling systems as a system of systems. In *Proceedings of the IEEE International Conference on Industrial Technology*, Churchill, Victoria, Australia, 2009.

[60] R. Kolisch and A. Drexl. Local search for nonpreemptive multi-mode resource-constrained project scheduling. *IIE Transactions*, 29(11):987–999, 1997.

[61] M. Laguna and R. Marti. Grasp and path relinking for 2-layer straight line crossing minimizationgrasp and path relinking for 2-layer straight line crossing minimization. *Journal on Computing*, 11(1):44–52, 1999.

[62] V. Le, D. Creighton and S. Nahavandi. Simulation-based input loading condition optimisation of airport baggage handling systems. In *Proceedings of the 2007 IEEE Intelligent Transportation Systems Conference, Seattle, WA, USA*, pages 574–579, 2007.

[63] A. Lee. *Applied Queueing Theory*. Macmillan, 1966.

[64] C. Lee and S. Park. Chebyshev center based column generation. *Discrete Applied Mathematics*, 159(18):2251 – 2265, 2011.

[65] O. Lenior. Airport baggage handling – Where do human factors fit in the challenges that airports put on a baggage system? *Work: A Journal of Prevention, Assessment and Rehabilitation*, 41:5899–5904, 2012.

[66] Library of Congress. Passengers alighting from Lot plane. `http://loc.gov/pictures/resource/matpc.22394/`.

[67] M. E. Lübbecke and J. Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007 – 1023, 2005.

[68] A. Marín. Airport management: Taxi planning. *Annals of Operational Research*, 143(1):191–202, 2006.

[69] N. Martel and P. Senviratne. Analysis of factors influencing quality of service in passenger terminal buildings. *Transportation Research Records*, 1273(1273):1–10, 1990.

[70] M. Moser, D. Jokanović and N. Shiratori. An algorithm for the multidimensional multiple–choice knapsack problem. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 80(3):582–589, 1997.

[71] R. Neufville. The baggage system at Denver: Prospects and lessons. *Journal of Air Transportation Management*, 1(4):229–236, 1994.

[72] G. Newell. *Application of Queueing Theory*, volume 2. Chapman & Hall, 2 Ed., 1982.

[73] J. Pagani, A. Abd El Halim, Y. Hassan and S. Easa. User-perceived level-of-service evaluation model for airport baggage-handling systems. *Transportation Research Record: Journal of the Transportation Research Board*, 1788(1):33–42, 2002.

[74] Y. Park and S. Ahn. Optimal assignment for check-in counters based on passenger arrival behaviour at an airport. *Transportation Planning and Technology*, 26(5): 397–416, 2003.

[75] M. Parlar and M. Sharafali. Dynamic allocation of airline check-in counters – A queuing optimization approach. *Management Science*, 54(8):1410 – 1424, 2008.

[76] H. Piper. Designing principles for decentralized terminals. *Airport Forum*, 4:39–51, 1974.

[77] M. Resendel and C. Ribeiro. Grasp with path-relinking: Recent advances and applications. In T. Ibaraki, K. Nonobe and M. Yagiura, editors, *Operations Research/Computer Science Interfaces Series*, volume 32, pages 29–63–. Springer US, 2005.

[78] J. Rieck, J. Zimmermann and T. Gather. Mixed-integer linear programming for resource leveleing problems. *European Journal of Operational Research*, 221(1):27–37, 2012.

[79] F. Robusté. *Analysis of baggage handling operations at airports*. PhD thesis, Institute of Transportation Studdies, University of California, Berkeley, 1988.

[80] F. Robusté and C. Daganzo. Analysis of baggage sorting schemes for containerized aircraft. *Transportation Research Part A. Policy and Practice*, 26(1):75–92, 1992.

[81] D. Ryan and B. Foster. An integer programming approach scheduling. In A. Wren, editor, *Computer and Scheduling of Public Transport Urban Pasenger Vehicle and Crew Scheduling*, pages 269 – 280. North-Holland, Amsterdam, 1981.

[82] Süddeutsche.de. Münchner stoppen dritte Startbahn – Ude gratuliert Gegnern. `http://www.sueddeutsche.de/muenchen/buergerentscheid-zum-flughafenausbau-muenchner-stoppen-\\dritte-startbahn-ude-gratuliert-gegnern-1.1385352`, June 2012.

[83] SITA. Baggage report 2014. `http://www.sita.aero/surveys-reports/industry-surveys-reports/baggage-report-2014`, 2014.

[84] J. Steffen. Optimal boarding method for airline passengers. *Journal of Air Transport Management*, 14(3):146–150, 2008.

[85] R. Stolletz. Approximation of the non-stationary m(t)/m(t)/c(t)-queue using stationary queuing models: The stationary blacklog-carryover approach. *European Journal of Operational Research*, 190(2):478–493, 2008.

[86] R. Stolletz. Operational workforce planning for check-in counters at airports. *Transportation Research Part E*, 46(3):414–425, 2010.

[87] R. Stolletz. Analysis of passenger queues at airport terminals. *Research in Transportation Business and Management*, 1(1):144–149, 2011.

[88] S. Takakuwa and T. Oyama. Simulation analysis of international-departure passenger flows in an airport terminal. In *Proceedings of the 2003 Winter Simulation Conference*, pages 1627–1634, 2003.

[89] J. Tanner. A queuing model for departure baggage handling at airports. Graduate report, The Institute of Transportation and Traffic Engineering, University of California, Berkley, 1966.

[90] The Boeing Company Corp. Current market outlook. `http://www.boeing.com/commercial/cmo/pdf/Boeing_Current\_Market_Outlook_2012.pdf`, 2012.

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

[92] V. Tošić and O. Babic. Quantitative evaluation of passenger terminal orientation. *Journal of Advanced Transportation*, 18(3):279–295, 1984.

[93] E. Tsang and C. Voudouris. Fast local search and guided local search and their application to british teleocom's workforce scheduling problem. *Operations Research Letters*, 20:119–127, 1997.

[94] N. Van Dijk and E. Van der Sluis. Check-in computation and optimization by simulation and IP in combination. *European Journal of Operational Research*, 171 (3):1152–1168, 2006.

[95] C. Voudouris. *Guided local search for combinatorial optimisation problems*. PhD thesis, Departement of Computer Science, University of Essex, Colchester UK, 1997.

[96] C. Voudouris and E. Tsang. Guided local search and its application to the traveling sales problem. *European Journal of Operational Research*, 113:469 – 499, 1999.

[97] H. Williams. *Model Building in Mathematical Programming*. John Wiley & Sons, 5 Ed., 2013.

[98] D. Wragg. *Flight before flying*. Osprey, 1974.

[99] S. Yan, C.-H. Tang and M. Chen. A model and a solution algorithm for airport common use check-in counter assignment. *Transportation Research Part A. Policy and Practice*, 38(5):101 – 125, 2004.

[100] S. Yan, C.-H. Tang and C.-H. Chen. Minimizing inconsistencies in airport common-use check-in counter assignments with a variable number of counters. *Journal of Air Transport Management*, 11(2):107–116, 2005.

[101] S. Yan, C.-H. Tang and C.-H. Chen. Reassignment of commen-use check-in counters following airport incidents. *Journal of Operational Research Society*, 59(8):1100 – 1108, 2008.

[102] J.-R. Yen, C.-H. Teng and P. Chen. Measuring the level of services at airport passenger terminals: Comparison of perceived and observed time. *Transportation Research Record: Journal of the Transportation Research Board*, 1744(1):17–33, Jan. 2001.

[103] H. Yu and C. Xu. Desing and implementation of the key technologies for baggage handling control system. In *Proceedings of the International Conference on Measuring Technology and Mechatronics Automation*, 2010.