TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Operations Management

Models and Methods for Optimizing Baggage and Ground Handling at Airports

Dipl.-Inf. Univ. Ferdinand Kiermaier

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

Doktors der Wirtschaftswissenschaften (Dr. rer. pol.)

genehmigten Dissertation.

Vorsitzender:		Univ.-Prof. Dr. Martin Grunow
Prüfer der Dissertation:	1.	Univ.-Prof. Dr. Rainer Kolisch
	2.	Prof. Dr. Jonathan F. Bard University of Texas at Austin / USA

Die Dissertation wurde am 31.03.2015 bei der Technischen Universität München eingereicht und durch die Fakultät für Wirtschaftswissenschaften am 15.04.2015 angenommen.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction to Airport Baggage Handling

When a flight arrives at an airport, baggage and cargo must be transferred to the terminal, passengers and the crew have to be transported, and the aircraft must be prepared for its next trip. These functions, amongst others, fall under the heading of ground handling processes (see Ashford1997). Operating and planning ground handling processes involves airports, airlines, and ground handling companies. As illustrated in figure 1.1, airports provide the terminal infrastructure separating the airport's airside, which is accessible to the public, from the airport's landside, which is a security restricted area with access to aircraft. While ground handling providers are responsible for the operative implementation of passenger and baggage transport between the aircraft and the terminal infrastructure, it is the airport's responsibility to effectively plan the different flows.

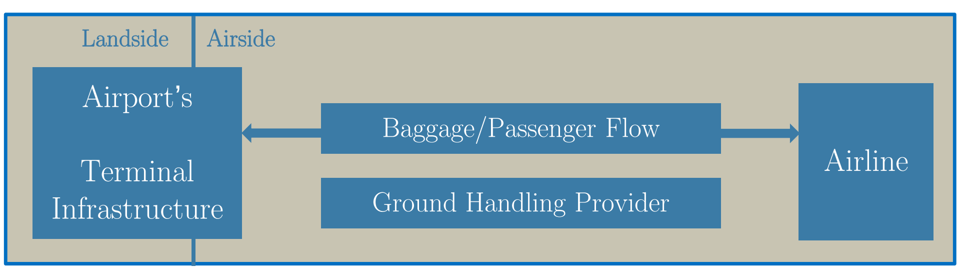


Figure 1.1: Ground handling actors

With the steady increase in worldwide passenger numbers and ongoing globalization (see IATA [70] and Airbus [4]), airports are increasingly exposed to an open and competitive market where service quality is a crucial competitive factor to attract airlines and their passengers. To simultaneously keep pace with the growth, airports are more and more optimizing the use of their existing infrastructure before investing in cost intensive expansion programs. A major factor for passenger satisfaction and an airport's economic performance is the quality of baggage handling. To analyze the quality, a key figure is the number of so-called 'mishandled bags' where each mishandled bag corresponds to a report of a delayed, damaged or pilfered bag (see SITA [115]). While the number of mishandled bags worldwide peaked in 2007 with 46.9 million mishandled bags out of 2.48 billion passengers, the airport industry's efforts led to a decrease of 21.8 million mishandled bags out of 3.13 billion passengers in 2013. While considerable progress has been made, the costs for mishandled bags are still 2.09 billion US dollars per year. This dissertation, in part, focuses on the improvement of baggage flows from an airport's planning perspective.

1.2 Introduction to Airport Ground handling

Ground handling services are provided by airport managing bodies, airlines, and independent ground handlers. In 1996, airports dominated the European market with 68 percent of ground handling being provided by airports, 25 percent by airlines and only 7 percent by independent ground handlers, see figure 1.2. In order to open the ground handling market, to reduce the fees airlines had to pay, and to improve the quality of the ground handling processes, the European Council has passed Directive 96/67 in 1996. The directive defined 11 categories of ground handling activities, amongst which four may be subject to a limited restriction: baggage handling, cargo handling, apron services, and fueling services. For each of those services, an EU Member States may limit the number of service providers to no more than two for all airports on their territory with an annual passenger volume of more than 2 million or a freight volume of more than 50,000 tons. At least one of the service provides has to be independent in terms of a direct or indirect influence exerted by the airport or an airline that delivered more than 25 percent of the airport's passenger or freight volume, respectively. Beginning January 1st 2001, the directive entered into force leading to a significant shift in market shares as illustrated in figure 1.2. The competitive pressure, due to the liberalization of the market, forces ground handling providers to efficiently employ their resources. Taking into account that labor

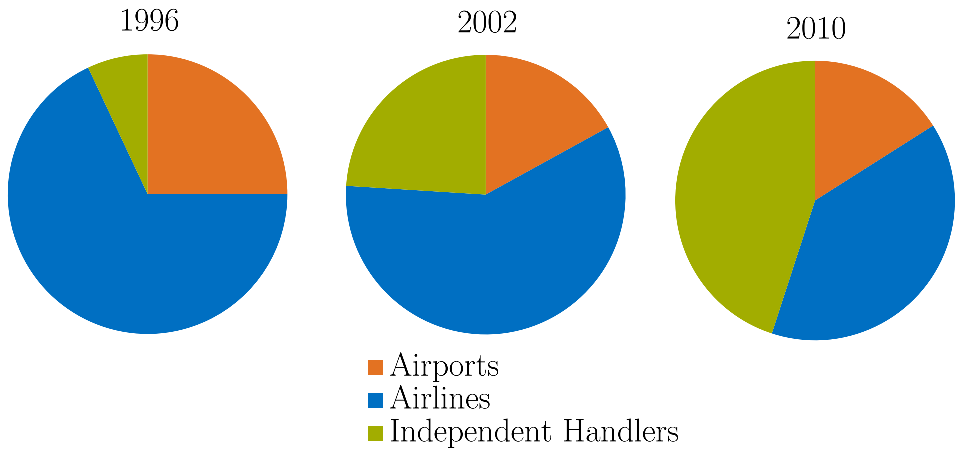


Figure 1.2: Market shares in ground handling

costs constitute about 80 percent of a ground handling provider's total costs (Müller et al. [93]), the second key aspect of this dissertation is on an efficient planning and scheduling of personnel with a focus on airport ground handling staff. Ground handling providers often employ hundrets of workers, making the personnel scheduling a challenging task. This second focus of this dissertation is on personnel scheduling on a tactical and operational planning level which is applied to airport ground handling.

1.3 Scientific scope

When investigating baggage handling at airports from an Operations Reseach (OR) point of view, four major baggage streams can be identified: outbound baggage handling, inbound baggage handling, tranfer baggage handling, and check-in baggage handling. Despite the growing economic importance of an optimization of baggage handling, many airports still need to adapt to the new conditions. On the operational level, experienced manual planners are often responsible for planning one or more of the baggage flows. While, in practice, system visualisation often supports the planners, decision support tools based on OR methods have yet to be established. The first part of this disseration focuses on the development of methods to support airport's planning. Therefore, the different baggage handling processes are presented in detail. For each of the processes, OR relevant literature is reviewed and an innovative mixed integer programming (MIP)-based mathematical model is presented. Further, the complexity of each of the planning problems is studied and opportunities for future research are highlighted. For planning problems associated with inbound baggage handling we present a MIP-based model for their optimization at a major European airport. A heuristic solution method is presented which is capable of solving real world instances. In simulation studies, results of both methods are compared against the airport's current planning indicating a significant improvement in key figures of both processes. In the context of this work, optimization algorithms for the transfer and inbound baggage handling to support the manual planner were implemented at a major European airport.

In the second part, the focus of this dissertation is on personnel scheduling problems. We study personnel scheduling on the tactical and operational level. Airport ground handlers often employ cyclic rosters to schedule their workforce (see Herbers [68]). In cyclic rostering, the workforce is divided into groups of equal size and each group is assigned the same schedule but lagged in time. Those plans are typically established for a tactical planning period with a rather stable flight plan. At a major European ground handling company, different cyclic rosters are used throughout the summer and winter flight season in an innovative manner as they extend classical cyclic rostering by allowing for regulated weekly adaptions of the workers schedules. We review the OR related literature on cyclic rostering and propose a stochastic multi-stage program together with two relaxations to tackle the beforementioneds problem for real world instance sizes. In a computational study we compare our solution with the solution provided by the ground handling company, show the benefits of the new cyclic rostering approach in contrast to classical cyclic rostering, and investigate the value of the stochastic solution for cyclic rostering. On the operational planning level, we study the impact of shift and break flexibility in the context of weekly personnel schedules. Allowing for flexiblity is a key issue for airport ground handlers due to the high fluctuations in worker demand throughout the day. A mixed-integer programming (MIP) model is presented that includes shift and days-off scheduling along with break assignments for a multi-skilled workforce. As ground handling often requires to plan hundreds of workers, we suggest a decomposition procedure to solve large-scale instances in reasonable time. To achieve tractability, a two-stage decomposition procedure separates the tour scheduling problem from the break assignment problem. To better understand the literature and the models previously developed, a 3-field break classification scheme is introduced. A complexity analysis of the resulting break regulations is undertaken and computations are presented for a wide variety of scenarios using data provided by a European airport ground handler company where over 500 instances were investigated.

1.4 Overview

This dissertation is structured as follows. Chapter 2 provides an overview of baggage handling processes, resources needed for baggage handling, as well as the different baggage streams. Each of the baggage streams entails a planning problem to the airport for which an optimization model is presented. A detailed investigation of the inbound handling is given in chapter 3. An optimization method to tackle the problem is presented together with insights into the implementation of the method at a major European airport justifying the research's relevance in practice. In chapters 4 and 5, we switch the perspective and optimize personnel scheduling from a ground handling provider's point of view on the tactical and operational planning level. On the tactical level, chapter 4 introduces an innovative scheme for cyclic rostering which turns out to be especially suitable for ground handling, where cyclic rostering is often required due to union regulations. A stochastic mathematical model formulation is presented and compared with traditional personnel scheduling approaches. On the operational level, we present different mathematical models and solution procedures to tackle the personnel scheduling problem with a special focus on break assignment in chapter 5. Break regulations play an important role in ground handling as workforce demand, based on the flight schedule, evidences large fluctuations. We show the benefits as well as the disadvantages associated with flexible break regulations when scheduling ground handlers. Chapter 6 summarizes this dissertation and gives an outlook on future research.

Chapter 2

Baggage flows at airports: A survey and a generic model

2.1 Introduction

Increasing passenger volume of about 4% per year (see The Boeing Company Corp. [122]), 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 [19], Marín [89] and Dorndorf et al. [52]) 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. [15], Bachmat et al. [16], Bachmat and Elkin [14], Bachmat et al. [17] and Steffen [116]). 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. Planning 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 [115]). 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 [126]). 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 paper 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 problem is NP-hard 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 four 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 access roads, car-parking lots, public transportation stations and check-in. In constrast, the airside 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 airside, incoming flights feed baggage into the BHS which in the case of inbound baggage is forwarded to the baggage claim (4) at the landside, or in the case of transfer baggage (3) is forwarded to handling facilities where the baggage is loaded into containers for connecting flights. Together with Check-in baggage brought in by newly arriving passengers at the landside, transfer baggage is denoted as outbound baggage (4) as soon as it enters the BHS.

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

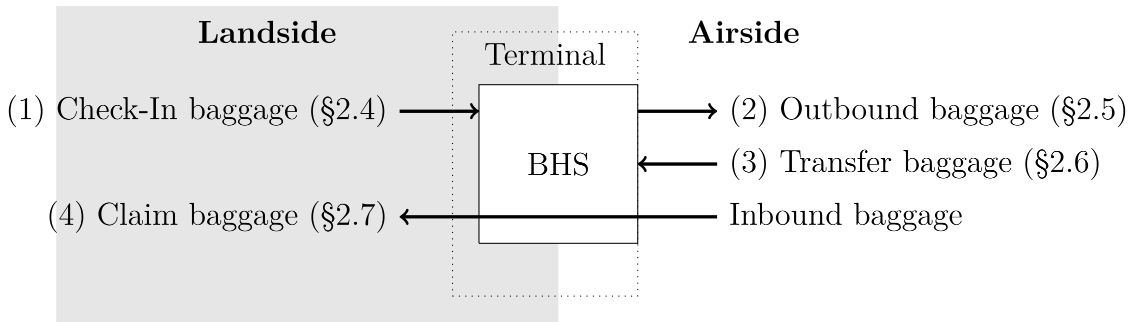


Figure 2.1: Baggage in- and outflow streams at airports

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). Outbound baggage originates either from the check-in (1) or, in case of a hub airport, from passengers of incoming flights transferring to connection 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 the four 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.

The remainder of this chapter is structured as follows: 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 research. Finally, Section 2.8 summarizes our review and provides outlook for further research directions.

Table 2.1: Airport actors on the baggage handling process and their responsibilities

Actors / Task	**Baggage handling processes** 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	•	•	•

2.2 Airport resources

Airport resources can be partitioned into 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 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 labor. For the baggage transport in containerized carriers bulk containers or, for smaller sized carriers, open containers are used. Bulk containers are also called unit load devices (ULD) because they are standardized in size to ease the loading into different airplane types. In the following, we will speak of containers if we do not need to distinguish between bulk container and palette. If a 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 resources. 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. [100]). 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 operational decisions. A detailed description of the operations of the BHS is given in Hallenborg and Demazeau [66], Johnstone et al. [75], Khosravi et al. [77] and de Neufville and Odoni [51]. Neufville [97] discusses design difficulties in installing and operating the BHS at Denver International Airport. The performance of the BHS is measured by the time required to transfer a bag from one point to another point within the system. But also the percentage of bags not delivered to the outgoing airplane or baggage claim in time is crucial to guarantee a level of service for passengers. In what follows we present the components of the BHS in more detail.

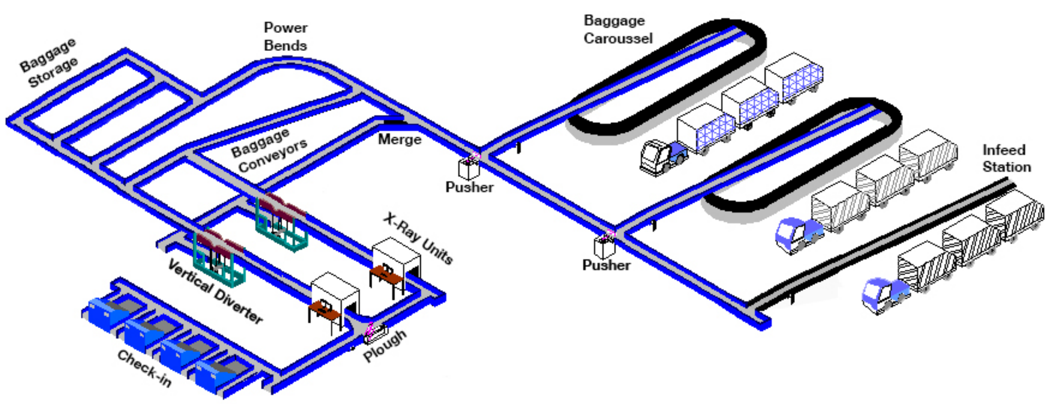


Figure 2.2: A BHS with its components

Infeed station Infeed stations are the access points for baggage into the BHS. 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. [83] 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. [75]). Yu and Xu [138] 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. [8]). 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 evolved 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 segment 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 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 [85]). 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 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. [64] for different carousel types). Different layouts for baggage claim areas are presented by de Barros and Wirasinghe [50], Gosling [65] and Robusté [109].

2.3 Generic assignment and scheduling problem

The decision problems for the check-in, outbound, inbound and transfer baggage processes discussed 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 resources as well as the maximum quantity of the resources which are employed to process the job. For example, the "check-in of a flight" can be done by using one counter each with up to four service lines or two counters with up to two service lines, respectively. Formally, job i can be processed in mode $m \in \mathcal{M}_i$ 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 $\left[S_i^{\text{es}}, S_i^{\text{ls}}\right]$ with S_i^{es} and S_i^{ls} as the earliest and latest possible start time, respectively. The finish time S_i^e of job i is defined 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 resources of type g allocated to 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 $r_{i,m,g}$ given by mode m for the usage of resource g. Figure 2.3 illustrates resource allocation for a job i on a resource g.

Passenger or bag flow 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

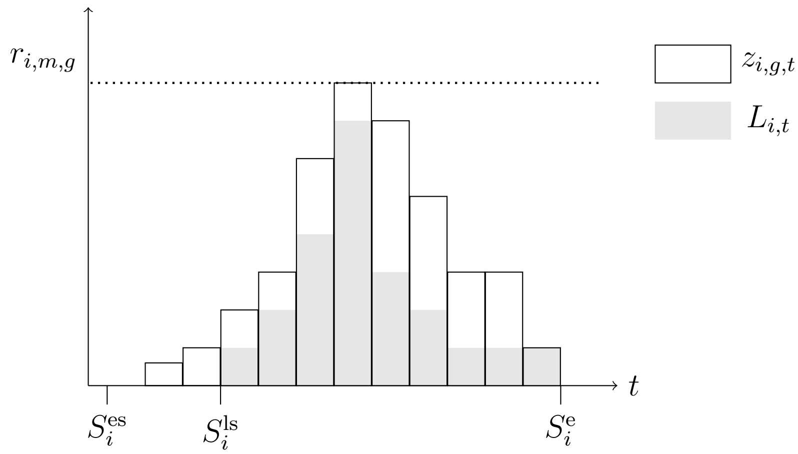


Figure 2.3: Time dependent resource allocation by the GASP.

variable $I_{i,g,t}$ which gives the number of passengers or bags for flight i at resource g at the end of period t.

Model formulation The GASP for baggage handling processes can be formulated as follows:

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

subject to

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

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

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

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

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

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

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

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

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

Objective function (1) is generic and respects 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 minimum processing time $p_{i,m,g}$. 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 to 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 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 $F \cdot G \cdot T$ continuous 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 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 [63]), 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 $r_j \in \mathbb{Z}_0^+$, 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 $S_j^{\mathrm{es}} = r'_j$ be the period in which the job can start (release time), $S_j^{\mathrm{ls}} = e'_j - p'_j$ the last period in which the job is allowed to start (deadline), and $p_{j,m,g} = p'_j$ the processing time. S_j^{e} 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. *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 [63]), can be polynomially transformed into an instance of SGASP II.

INSTANCE: Set $\mathcal{J}^{\mathrm{BP}}$ of items, each item $j' \in \mathcal{J}^{\mathrm{BP}}$ 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 $j' \in \mathcal{J}^{\mathrm{BP}}$ 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 1 and 2, we obtain

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

In the following, we will adopt the GASP to the four 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

In the daily operational check-in counter planning 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 the 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 [36] and Stolletz [118]).

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 [36], Chun [42, 43, 44], Chun and Mak [45], Van Dijk and Van der Sluis [130], Park and Ahn [101] and Parlar and Sharafali [102]). In the literature, costs are minimized by the minimization of the number of check-in counters to be opened in a given time interval. An efficient usage of the counter leads to reduced renting and 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 [36], Chun [42, 43, 44], Chun and Mak [45], Van Dijk and Van der Sluis [130], Park and Ahn [101] and Parlar and Sharafali [102]). 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. [134, 135, 136]). Besides operating costs, Yan et al. [134, 135, 136] 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 [119]).

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. [135]). The assignment of flights to common-use counters requires the assignment 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 [36]), 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 [130] employ simulation to determine the minimal number of required counters for each flight (see Joustra and Dijk [76]) 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 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 [36]) which leads to high computation times when solving real world instances. Therefore, Van Dijk and Van der Sluis [130] propose a LP-heuristic which decomposes the problem into smaller subproblems with natural separation such as domestic and international flights. Bruno and Genovese [36] show alternative MIP formulations for the static and dynamic allocation model of Van Dijk and Van der Sluis [130] which have similarities to the capacitated lot sizing problem (see Bitram and Yanasse [32] and Florian et al. [60]).

Beside mathematical programming, Chun [44] 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 [42, 43] and Chun and Mak [45]). Considering the arrival patterns of passengers, Park and Ahn [101] derive system of rules for the allocation of flights to counters for the Seoul Gimpo International Airport in Korea.

Parlar and Sharafali [102] 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. [134, 135] present IPs for static and dynamic allocation of flights to common-use counters at the Chiang Kai-Shek International Airport in Taiwan. While Yan et al. [134, 135] decompose the MIP in smaller subproblems which are independently solved in one run, Yan et al. [135] use an iterative improvement procedure. Chun and Mak [45] 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. [136]. 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

Table 2.2: Literature on check-in counter allocation

	Dedicated check-in		Common-use check-in	
	Static	Dynamic	Static	Dynamic
Optimization				
Mathematical Programming				
IP	[130]	[130]	[134]	[135, 136]
MIP	[36]	[36]		
Constraint Programming		[42–45]		[45]
Heuristic				
Constructive		[101]		
Improvement		[83]		[83]
Stochastic Programming				
Dynamic Programming		[102]		
Analytical				
Cost model	[74]			
Queuing model				
Deterministic	[71, 72, 98, 103, 121, 126]			
Stochastic	[84, 119]			
Simulation				
Discrete-event	[7, 46, 76, 120]			
Monte-Carlo	[127]			
Other				
Description	[57, 59, 73]			
Literature review	[126]			

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 [71, 72], Lee [84], Newell [98] and Stolletz [119]). 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 [84] and Stolletz [119]). 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 [119] uses the stationary backlog carryover approach (see Stolletz [117]) which approximates the inhomogeneous arrival process of passengers by dividing the time horizon into small periods in which a 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 [98], Janic [71, 72], Tanner [121], Park and Ahn [101] Piper [103], and Tošić [126]. 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 are Monte-Carlo simulation (see Tošić and Babic [127]) and discrete-event simulations (see Appelt et al. [7], Chung and Sodeinde [46], Joustra and Dijk [76] and Takakuwa and Oyama [120]). Using discrete-event simulation Joustra and Dijk [76] compare waiting times of passengers for common-use and dedicated counters. In contrast to Stolletz [119], Joustra and Dijk [76] 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 [74]. 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 [36] (see also Van Dijk and Van der Sluis [130]) and Yan et al. [134]. 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 [130]). 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 processed 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:

$\left[S_i^{\text{es}}, S_i^{\text{ls}}\right]$	–	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 in period 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 in period t;
$I_{i,t}$	–	number of waiting passengers of flight i at the end of period t.

The start time of each flight is within time window $\left[S_i^{\text{es}}, S_i^{\text{ls}}\right]$; 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. Then, the check-in counter planing problem (CCPP) is given by

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

subject to

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

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

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

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

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

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

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. Due to constraints (12) the service of each each flight i is started exactly once in a specific mode m in the time window $\left[S_i^{\text{es}}, S_i^{\text{ls}}\right]$. Constraints (13) ensure that the minimal number of required service lines for flight i at time t are allocated. 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 number of open lines for flight i is 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 the flights last check-in period.

Complexity analysis Due to Theorem 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 [44] and Yan et al. [134]). In a first step, flights are allocated to counters and in a second step the agents are assigned to open counters. The two step approach where the capacities of the personnel are relaxed in the first step leads to heuristic solutions of the problem. Therefore, an integrated model for the check-in counter allocation and 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

During the outbound baggage process, baggage is loaded into cargo holds of the aircraft. The baggage stream originates from check-in and transfer baggage. Figures 2.4 and 2.5 show the flow of outbound baggage.

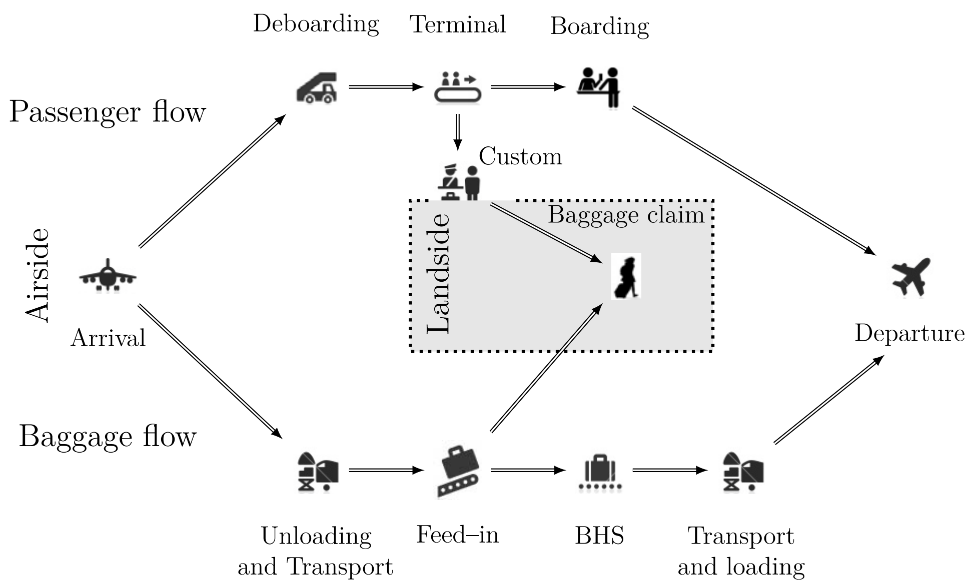


Figure 2.4: Passenger and baggage flow for transfer and inbound processes

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

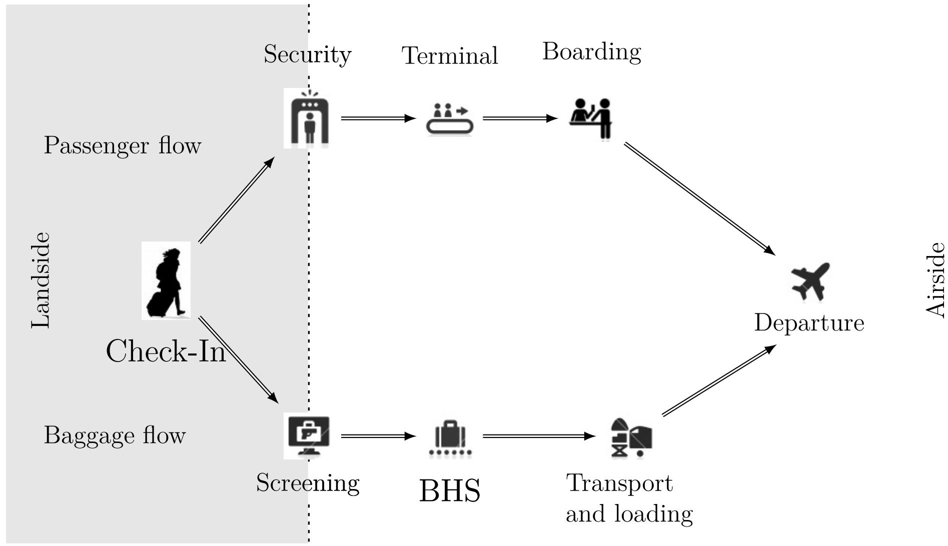


Figure 2.5: Passenger and baggage flow landside to airside

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.

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 [61], Frey et al. [62]). 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. [8]). Each handled flight requires its own number of working stations at which workers load the baggage into containers. Thus, according to Abdelghany and Abdelghany [1], Abdelghany et al. [2] and Ascó et al. [8, 9] 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. [8] 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. [8] and Ascó et al. [9] 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 [61]. 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 [61] 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. [62] decompose the problem into several subproblems to reduce the complexity. Numerical results of the decomposition heuristic are not reported. Barth and Pisinger [26] present a model formulation for a decentralized and centralized baggage handling environment at Frankfurt Airport. Lenior [85] discusses human interactions with semi- or fully automated systems for baggage sorting.

Table 2.3: Literature on baggage handling

Optimization	
Mathematical Programming	
IP	[8, 26, 61, 62]
Heuristic	
Constructive	[1, 2, 8, 9]
Other	
Description	[85]

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. [8], a storage system for early bags is considered and the start time for the baggage handling has to be determined. 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:

$\left[S_i^{\text{es}}, S_i^{\text{ls}}\right]$ – 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 in period 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 at the beginning of period t,
0 otherwise;

$z_{i,g,t}$ – number of working stations of carousel g that are used by flight i in period t;

$I_{i,g,t}$ – workload caused by flight i on baggage carousel g at the end of period 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

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

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

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

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

$$I_{i,g,t} = (I_{i,g,t-1} + \left(\lambda^{+}_{i,t} + \min\{\beta, B_{i,t}\}\right) \cdot \sum_{\tau=S^{\mathrm{es}}_i}^{\min\left\{t,S^{\mathrm{ls}}_i\right\}} \sum_{m \in \mathcal{M}_i} x_{i,m,\tau} - \lambda^{-} \cdot z_{i,g,t})^{+} \qquad \forall\, i \in \mathcal{F}, g \in \mathcal{G}, t \in [S^{\mathrm{es}}_i, S^{\mathrm{e}}_i] \tag{24}$$

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

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

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

and (8) to (10)

The workload over all baggage carousels is balanced due to the quadratic objective function (19). 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). The amount of bags on carousel g of flight i is the sum of flight i's arriving bags from the storage, $\min\{\beta, B_{i,t}\}$, and the number of bags $I_{i,g,t-1}$ which could not be loaded in previous period $t-1$ minus flight i's loading rate $\lambda^{-} \cdot z_{i,g,t}$. Constraints (25) calculate flight i's amount of baggage in the central storage system. Before baggage handling starts, i.e. $\left(1 - \sum_{m \in \mathcal{M}_i} \sum_{\tau=S^{\mathrm{es}}_i}^{t} x_{i,m,\tau}) = 1\right)$, flight i's amount of baggage $B_{i,t}$ in the storage system at the end of period t is the number of baggage stored in the previous period $t-1$ plus the newly arriving amount of baggage $\lambda^{+}_{i,t}$. The total amount of stored bags at time t is restricted in constraints (26). The above model is non-linear. However, it can be linearized straight forward (see Williams [133]).

Complexity analysis Due to Theorem 2 the problem is NP-hard 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 shortages 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.

In practice, unforeseen disruptions such as flight position changes, breakdown of carousels or working stations require the re-assignment or re-scheduling of flights' baggage handling. Hence, it is necessary to update the assignment to handling facilities and the flights' baggage handling schedule during the course of the day. So far, the literature has just focused on the generation of daily plan and on the re-planning of work groups for departing flights.

2.6 Transfer baggage

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 [115]). According to SITA [115], the main contributor, by far, to the mishandling problem is transfer baggage. Transfer baggage mishandling causes 53% of all missing bags 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 [67] 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" through the 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" reaches 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:

$\left[S_i^{\mathrm{es}}, S_i^{\mathrm{ls}}\right]$ – 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 infeed station g is used by flight i in period t, 0 otherwise;
$I_{i,g,t}$ – cumulative number of bags of flight i handled at infeed station g up to the end of period 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 such that the baggage reaches the handling facility 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:

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

subject to

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

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

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

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

and (8) to (10)

The number of handled bags is used in the objective function (28) in order to penalize the amount of late baggage. 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).

Complexity analysis Due to Theorem 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 [47], there is no paper on the transfer baggage handling decisions. As Clausen and Pisinger [47] 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

The inbound baggage process merges the separated flow of the passengers and of the baggage at the baggage claim carousels (see Figure 2.4). 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 pass 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. [11]). Since the perceived satisfaction of passengers is an important factor for airlines and airport (see for example Correia and Wirasinghe [48], Martel and Senviratne [90] and de Neufville and Odoni [51]) 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 [21]). 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 [48] and Ghobrial et al. [64]).

As inbound and transfer baggage is brought to the infeed stations by the same baggage tug it is desired by the groundhandler that the assigned infeed stations for transfer baggage and inbound baggage are located closely to each other. Since transfer baggage has to be transferred to outgoing flights, it is more time critical than inbound baggage and is usually prioritized over inbound baggage. The joint usage of infeed stations for inbound and transfer baggage is generally not possible. One infeed station can be used to feed in the baggage of one container at a time. Therefore, containers assigned to the same infeed station have to wait in a queue and are processed first-come, first-served.

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 [90] and Robusté [109]). Robusté [109] 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 [21], Robusté [109] and Tošić [126] 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 [69] and Newell [98]). 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. [64] test baggage claim carousel types with different frontage shapes and lengths. Browne et al. [33] and Gosling [65] derive formulas for computing the maximal expected queue length of bags and passengers under the assumption of constant arrival rates for passenger.

Table 2.4: Literature on inbound baggage handling

Analytical	
Queuing model	
Deterministic	[21, 33, 64, 65, 69, 98, 109, 126]
Survey	[48, 137]
Other	
Description	[51, 90]
Literature review	[126]

A survey evaluating passengers' satisfaction is presented by Correia and Wirasinghe [48] and Yen et al. [137]. Yen et al. [137] show that passengers overestimate their waiting times at baggage claim carousel. Correia and Wirasinghe [48] confirm the importance of minimal passenger waiting times in a survey conducted at Calgary International Airport (see also Martel and Senviratne [90]).

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:

$\left[S_i^{\mathrm{es}}, S_i^{\mathrm{ls}}\right]$	–	time window for the start of the infeed of inbound baggage of flight i;
S_i^{e}	–	upper bound for the end of flight i's baggage claim;
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 in period 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 baggage carousel/infeed station g is used by flight i in period t, 0 otherwise;

$I_{i,g,t}$ – number of bags of flight i on carousel g not picked up at the end of period 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 $t \in \mathcal{T}$. The basic model formulation for the inbound baggage handling is stated as

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

subject to

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

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

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

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

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. 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). The above model is non-linear. However, it can be linearized straight forward (see Williams [133]).

Complexity analysis Due to Theorem 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. Model (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 efficiently as possible. Quite some work has already been done on optimizing baggage handling at airports. In particular, during the last decades the issue of baggage became more and more the focus of OR which underlies its importance. In this chapter, we provide a structured description of the baggage handling process and classify the research on this topic. We discuss constraints and objectives of the four subprocesses that are common for many airports. A generic model for baggage handling processes at airports is introduced which employs a new and innovative resource assignment and scheduling formulation. Based on the generic model we derive the presented baggage subprocesses and show their similarities and differences. In so doing, we derived in Section 2.4.2 a more general model formulation for the CCPP which combines two check-in counter planning strategies found in literature. Our paper with the mathematical classification should serve 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 optimization 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

Inbound baggage handling

3.1 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 an 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 [109]). Passengers service quality can 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 [90] and Correia and Wirasinghe [48]).

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 available infeed stations. 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 forwards 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 of a flight, 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 two examples: 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. Barbo [21] 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. Browne et al. [33] present a predictive model for computing the maximal queue length of passengers and baggage at claim areas. Based on their results, Ghobrial et al. [64] 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 Martel and Senviratne [90] 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 Correia and Wirasinghe [48]. It should be noted that Yen et al. [137] 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 de Barros and Wirasinghe [50] as well as Gosling [65] who study the design of the baggage claim hall.

To the best of our knowledge, this paper is the first to provide 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 minimize waiting times for passengers.

The remainder of this chapter is structured as follows: Section §3.2 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 §3.3. Further, its problem complexity is established and structural properties are discussed in section §3.3. 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 §3.4 and §3.5 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 and an increased service quality by reducing waiting times by about 3 minutes on average. The computational study is presented in §3.6. §3.7 concludes this chapter by summarizing the results.

3.2 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 more than one trip 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 influences 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 (see Figure 3.1). 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. Direct infeed stations are very close to a terminal's baggage carousels. Often they are 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.

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

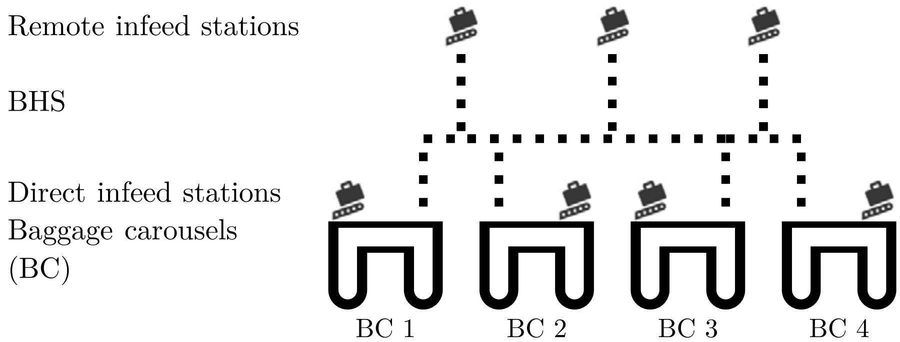


Figure 3.1: Infrastructure for the inbound process

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 anymore. 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.

Claiming process While a flight's baggage 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

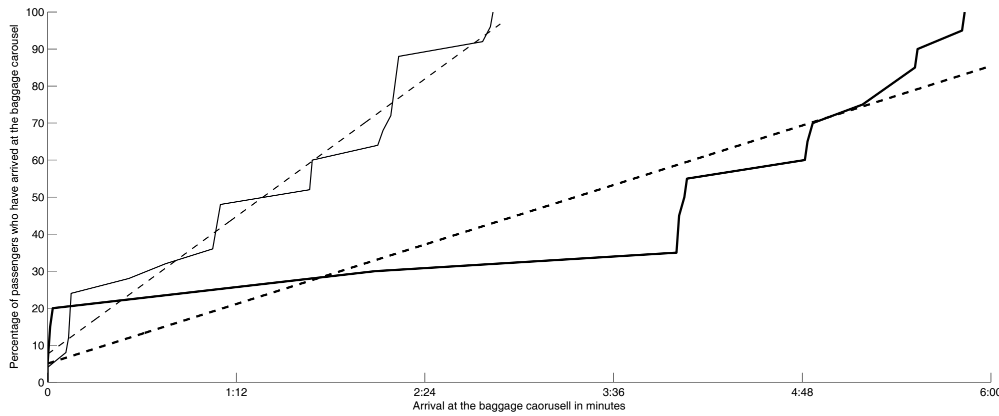


Figure 3.2: Passenger arrival at a baggage carousel for a domestic (thin) and transcontinental flight (bold)

carousel. For example, passengers from transcontinental flights need to pass through immigration and customs which is not the case for domestic flights. Figure 3.2 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 shown 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 Robusté [109]).

3.3 Mathematical model

In this section we present a model for the inbound baggage handling problem (IBHP). We start by providing notations and assumptions in §3.3.1 before we present the model formulation of the IBHP in §3.3.2. §3.3.3 discusses problem characteristics and establishes the IBHP's complexity.

3.3.1 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 3.1 and 3.2. 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.

Table 3.1: Sets for the IBHP

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

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 $l_i \in \mathcal{L}_i \backslash \{1_i\}$ 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. Depending of flight i's parking position and by assuming constant walking speed of passengers the

Table 3.2: Parameters for the IBHP

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
S_i^{ob}	On block time of 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
K_c^{cb}	Conveyor belt capacity of baggage carousel c
K_c^{di}	Display capacity of baggage carousel c
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
Δ^{pl}	Duration to place containers on a baggage tug
$\Delta_{i,e}^{\text{trip}}$	Duration to drive from flight i's parking position to infeed station e
μ^{w}	Baggage loading rate of workers at an infeed station
μ_i^{p}	Arrival rate of passengers of flight i

first passenger arrives at carousel c at $A_{i,c}$. Based on the results presented in Figure 3.2, 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. [64]).

Preprocessed parameters Given the sets and parameters introduced above, we derive the parameters presented in Table 3.3. 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

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

- 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

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

Table 3.3: Parameters for the IBHP derived by preprocessing

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 the infeed station e is selected and the infeed process starts at time s

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 Ghobrial et al. [64] and Robusté and Daganzo [110])

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

- 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 (see Appendix A.2). The realized waiting time Δ^{wait} is measured by utility function $g(\Delta^{\text{wait}})$. The higher the waiting time the higher the value of $g(\Delta^{\text{wait}})$. In §3.5.1 we show how to derive the utility function in practice for a major European airport. Given the utility function, we derive the penalty coefficients $\Psi_{l,e,c,s}^{\text{wait}}$ for the waiting time for passengers having bags in trip $l \in \mathcal{L}_i$. 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 $p^{\text{wait}}_{l,e,c,s}(t_2)$ that all baggage of a passenger has arrived exactly till period t_2, i.e. $g\left(t_2 - t_1\right) \cdot p^{\text{wait}}_{l,e,c,s}\left(t_2\right)$. 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 $t^{\text{first}}_{l,e,c,s}$. Hence, the passengers arrive at baggage carousel c during interval $\left[A_{i,c}, A_{i,c} + \left\lfloor \frac{P_i}{\mu^{\text{p}}_i} \right\rfloor\right]$, while the bags of flight i's trip l arrive on baggage carousel c during interval $\left[t^{\text{first}}_{l,e,c,s}, t^{\text{first}}_{l,e,c,s} + \left\lfloor \frac{B_l}{\mu^{\text{w}}} \right\rfloor\right]$. 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

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

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

$$p^{\text{wait}}_{l,e,c,s}(t) = \sum_{n=1}^{N_i} p_i(n) \cdot \left(\mathbb{1}_{\{B_{i,e,c}(t) \geq n\}} \cdot \frac{\binom{B^{\text{tot}}_i - n}{B_{i,e,c}(t) - n}}{\binom{B^{\text{tot}}_i}{B_{i,e,c}(t)}} - \mathbb{1}_{\{B_{i,e,c}(t-1) \geq n\}} \cdot \frac{\binom{B^{\text{tot}}_i - n}{B_{i,e,c}(t-1) - n}}{\binom{B^{\text{tot}}_i}{B_{i,e,c}(t-1)}} \right) \tag{38}$$

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.

3.3.2 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 $\mathcal{L}^{\mathrm{d}}$. 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 $e \in \mathcal{E}$.

Table 3.4: Decision variables for the IBHP

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

Using the decision variables listed in Table 3.4 the model formulation for the inbound baggage handling problem (IBHP) can be written as follows

$$\text{Minimize } f(\mathbf{x}) = \lambda \cdot \left(\sum_{c \in \mathcal{C}} \sum_{t \in \mathcal{T}} u_{c,t}^2 \right) + \qquad (39)$$

$$(1-\lambda) \cdot \left(\sum_{i \in \mathcal{F}} \sum_{e \in \mathcal{E}} \sum_{c \in \mathcal{C}} \sum_{s=S^{\mathrm{es}}_{1_i,e}}^{S^{\mathrm{ls}}_{1_i,e}} \Psi_{i,c}^{\mathrm{walk}} \cdot x_{l_i,e,c,s} \right)$$

$$\text{Minimize } f(\mathbf{x}) = \lambda \cdot \left(\sum_{c \in \mathcal{C}} \sum_{t \in \mathcal{T}} u_{c,t}^2 \right) + \qquad (39')$$

$$(1-\lambda) \cdot \left(\sum_{l \in \mathcal{L}} \sum_{e \in \mathcal{E}} \sum_{c \in \mathcal{C}} \sum_{s=S^{\mathrm{es}}_{l,e}}^{S^{\mathrm{ls}}_{l,e}} \Psi_{l,e,c,s}^{\mathrm{wait}} \cdot x_{l,e,c,s} \right)$$

subject to

$$\sum_{e\in\mathcal{E}}\sum_{c\in\mathcal{C}_e}\sum_{s=S^{\text{es}}_{1_i,e}}^{S^{\text{ls}}_{1_i,e}} x_{1_i,e,c,s} = 1 \qquad \forall\ i\in\mathcal{F} \tag{40}$$

$$\sum_{c\in\mathcal{C}_e}\sum_{s=S^{\text{es}}_{l,e}}^{S^{\text{ls}}_{l,e}} \left(x_{l,e,c,s} - x_{1_i,e,c,s}\right) = 0 \qquad \begin{array}{l}\forall\ i\in\mathcal{F},\\ l\in\mathcal{L}_i\backslash\left\{1_i\right\},\\ e\in\mathcal{E}\end{array} \tag{41}$$

$$\sum_{e\in\mathcal{E}_c}\sum_{s=S^{\text{es}}_{l,e}}^{S^{\text{ls}}_{l,e}} \left(x_{l,e,c,s} - x_{1_i,e,c,s}\right) = 0 \qquad \begin{array}{l}\forall\ i\in\mathcal{F},\\ l\in\mathcal{L}_i\backslash\left\{1_i\right\},\\ c\in\mathcal{C}\end{array} \tag{42}$$

$$\begin{aligned}&\sum_{c\in\mathcal{C}_e}\sum_{s=S^{\text{es}}_{l,e}}^{S^{\text{ls}}_{l,e}} \left(s+\Delta^{\text{lag}}_{l,h,e}\right)\cdot x_{l,e,c,s} - && \forall\ e\in\mathcal{E}, l,h\in\mathcal{L}^{\text{d}}:\\ &\quad M\cdot\left(1-y_{l,h,e}\right) \leq \sum_{c\in\mathcal{C}_e}\sum_{s=S^{\text{es}}_{h,e}}^{S^{\text{ls}}_{h,e}} s\cdot x_{h,e,c,s} && l\neq h \wedge l \preceq_e h\end{aligned} \tag{43}$$

$$\begin{aligned}&\sum_{c\in\mathcal{C}_e}\sum_{s=S^{\text{es}}_{l,e}}^{S^{\text{ls}}_{l,e}} \left(s+\Delta^{\text{lag}}_{l,h,e}\right)\cdot x_{l,e,c,s} + M\cdot\left(1-y_{l,h,e}\right) && \forall\ e\in\mathcal{E}, l,h\in\mathcal{L}^{\text{d}}:\\ &\quad > \sum_{c\in\mathcal{C}_e}\sum_{s=S^{\text{es}}_{h,e}+1}^{S^{\text{ls}}_{h,e}} (s-1)\cdot x_{h,e,c,s} && l\neq h \wedge l \preceq_e h\end{aligned} \tag{44}$$

$$\begin{aligned}&M\cdot\left(y_{1_j,h,e}-1\right) + S^{\text{es}}_{1_j,e} && \forall\ i,j\in\mathcal{F}: i\neq j,\\ &\quad \leq \sum_{c\in\mathcal{C}_e}\sum_{s=S^{\text{es}}_{e,h-1}}^{S^{\text{ls}}_{e,h-1}} \left(s+\Delta^{\text{lag}}_{h-1,h,e}\right)\cdot x_{h-1,e,c,s} && h\in\mathcal{L}_i\backslash\left\{1_i\right\}:\\ &\qquad \leq M\cdot\left(y_{h,1_j,e}-1\right) + S^{\text{es}}_{1_j,e} && 1_j \cong_e h, e\in\mathcal{E}\end{aligned} \tag{45}$$

$$\begin{aligned}&M\cdot\left(y_{l,h,e}-1\right) + && \forall\ i,j\in\mathcal{F}: i\neq j,\\ &\quad \sum_{c\in\mathcal{C}_e}\sum_{s=S^{\text{es}}_{e,l-1}}^{S^{\text{ls}}_{e,l-1}} \left(s+\Delta^{\text{lag}}_{l-1,l,e}\right)\cdot x_{l-1,e,c,s} && l\in\mathcal{L}_i\backslash\left\{1_i\right\}, h\in\mathcal{L}_j\backslash\left\{1_j\right\}:\\ &\qquad \leq \sum_{c\in\mathcal{C}_e}\sum_{s=S^{\text{es}}_{e,h-1}}^{S^{\text{ls}}_{e,h-1}} \left(s+\Delta^{\text{lag}}_{h-1,h,e}\right)\cdot x_{h-1,e,c,s} && l \cong_e h, e\in\mathcal{E}\end{aligned} \tag{46}$$

$$y_{l,h,e} + y_{h,l,e} \leq 1 \qquad \forall\, e \in \mathcal{E}, l, h \in \mathcal{L} : l \neq h \wedge l \cong_e h \tag{47}$$

$$\sum_{c \in \mathcal{C}_e} \sum_{s=S^{\text{es}}_{l,e}}^{S^{\text{ls}}_{l,e}} x_{l,e,c,s} = \sum_{h \in \mathcal{L} \cup \left\{l_0^{\text{start}}\right\} : h \neq l} y_{h,l,e} \qquad \forall\, l \in \mathcal{L}, e \in \mathcal{E} \tag{48}$$

$$\sum_{c \in \mathcal{C}_e} \sum_{s=S^{\text{es}}_{l,e}}^{S^{\text{ls}}_{l,e}} x_{l,e,c,s} = \sum_{h \in \mathcal{L} \cup \left\{l_0^{\text{end}}\right\} : h \neq l} y_{l,h,e} \qquad \forall\, l \in \mathcal{L}, e \in \mathcal{E} \tag{49}$$

$$\sum_{l \in \mathcal{L}} y_{l_0^0,l,e} \leq 1 \qquad \forall\, e \in \mathcal{E} \tag{50}$$

$$\sum_{l \in \mathcal{L}} y_{l,l_0^{\text{L}+1},e} \leq 1 \qquad \forall\, e \in \mathcal{E} \tag{51}$$

$$\sum_{e \in \mathcal{E}} \sum_{s=S^{\text{es}}_{L_i,e}}^{S^{\text{ls}}_{L_i,e}} \left(s + \Delta^{\text{claim}}_{i,e,c,s}\right) \cdot x_{L_i,e,c,s} = \sum_{t \geq S^{\text{ob}}_i} t \cdot e_{i,c,t} \qquad \forall\, i \in \mathcal{F}, c \in \mathcal{C} \tag{52}$$

$$\sum_{c \in \mathcal{C}} \sum_{t=S^{\text{ob}}_i}^{T} e_{i,c,t} = 1 \qquad \forall\, i \in \mathcal{F}, c \in \mathcal{C} \tag{53}$$

$$\sum_{i \in \mathcal{F} : S^{\text{ob}}_i \leq t} \left(1 - \sum_{z=S^{\text{ob}}_i}^{t} e_{i,c,z}\right) \leq K^{\text{di}}_c \qquad \forall\, c \in \mathcal{C}, t \in \mathcal{T} \tag{54}$$

$$\frac{1}{K^{\text{cb}}_c} \sum_{l \in \mathcal{L}} \sum_{e \in \mathcal{E}_c} \sum_{s=S^{\text{es}}_{l,e}}^{S^{\text{ls}}_{l,e}} \Phi_{l,e,c,s,t} \cdot x_{l,e,c,s} \leq u_{c,t} \qquad \forall\, c \in \mathcal{C}, t \in \mathcal{T} \tag{55}$$

$$x_{l,e,c,s} \in \{0,1\} \qquad \forall\, l \in \mathcal{L}, e \in \mathcal{E}, c \in \mathcal{C}_e, S^{\text{es}}_{i,e} \leq s \leq S^{\text{ls}}_{i,e} \tag{56}$$

$$y_{l,h,e} \in \{0,1\} \qquad \forall\, e \in \mathcal{E}, l, h \in \mathcal{L} : l \neq h \wedge l \preceq_e h \tag{57}$$

$$e_{i,c,t} \in \{0,1\} \qquad \forall\, i \in \mathcal{F}, c \in \mathcal{C}, t \in \mathcal{T} \tag{58}$$

$$u_{c,t} \geq 0 \qquad \forall\, c \in \mathcal{C}, t \in \mathcal{T}. \tag{59}$$

Objective function (39) minimizes the carousels' utilization and additionally passengers' waiting times. The domain of the variables are defined in (56) to (59). IBHP's set of constraints can be separated into constraints (40) to (51) modeling the infeed process and constraints (52) to (55) modeling the passenger process.

Infeed process In constraints (40) the infeed start time of each trip is selected and assigned to one infeed and one baggage carousel. Due to (41) and (42) 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 (43) to (51). Constraints (43) 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 (44). 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 (44) 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 (44) 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 (45) and (46), 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 (47). Constraints (48) and (49) set for each trip $l \in \mathcal{L}$ exactly one predecessor and one successor trip at the assigned infeed station e. Constraints (50) and (51) 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 (52) to (54). Constraints (55) 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.1.

3.3.3 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 $S^* = \max\limits_{l \in \mathcal{L}, e \in \mathcal{E}} \left\{ S^{\mathrm{ls}}_{l,e} - S^{\mathrm{es}}_{l,e} + 1 \right\}$ is the maximal length of the time windows. For example, in a real world scenario (see §3.6) 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 following 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. *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 [63]). 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. $S^{\mathrm{es}}_{l,e} = S^{\mathrm{ls}}_{l,e}$. Moreover, let $S^{\mathrm{es}}_{l,e} = S^{\mathrm{es}}_{l,e'}$ 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. □

Already the 3-colorability problem is NP-hard, meaning that only 3 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 5. *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 Garey and Johnson [63]). 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 feasibility problem of the inbound problem at the airside and the utilization problem at the landside are NP hard (see Theorem 4 and 5) and because of the large number of variables and constraints, MIP formulation (39) – (147) does not guarantee to obtain a solution within an acceptable time for a practical implementation (see §3.6).

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 §3.4 runs after each touch down of a flight.

3.4 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. [6] or Feo and Bard [58]), a randomized greedy algorithm generates starting solutions in a construction phase. A subsequent local search phase is used to improve the starting solutions. Research 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 [80] or Resendel and Ribeiro [107]). 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. [55] or Voudouris and Tsang [132]). GLS is a local search scheme that augments the objective function of a local search procedure by incorporating penalty 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 augmented cost function of an objective function $f(\mathbf{x})$ and a solution $\mathbf{x}$ is defined as

$$h(\mathbf{x}) = f(\mathbf{x}) + \lambda \cdot \sum_{\phi \in \Phi} p_\phi \cdot I_\phi(\mathbf{x}) \tag{60}$$

where p_ϕ is the penalty associated with a feature ϕ and indicator function $I_\phi(\mathbf{x})$ is 1 if $\mathbf{x}$ has feature ϕ and 0 otherwise. The sum of 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, penalties of features having the highest GLS utility

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

are penalized by incrementing their penalty 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 1; Table 3.5 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. 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.

Algorithm 1 HGGLS heuristic

repeat
  $\mathbf{x} \leftarrow ConstructionHeuristic$ // Determine start solution with GRASP (see §3.4.1)
  $\mathbf{x} \leftarrow GFLS(N^1(\mathbf{x}))$ // GFLS on neighborhood $N^1(\mathbf{x})$ (see §3.4.2 and Appendix A.4, Algorithm 3)
  $\mathbf{x} \leftarrow GFLS(N^2(\mathbf{x}))$ // GFLS on neighborhood $N^2(\mathbf{x})$ (see §3.4.2 and Appendix A.4, Algorithm 3)
  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 §3.4.2)
  end if
until STOPPING CRITERION MET

3.4.1 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 3.4.2 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

Table 3.5: Notation for the pseudo code

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}

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 3.6). 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 3.3 (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 3.3 (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.

Table 3.6: Example with 3 flights and their trips

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)

3.4.2 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:

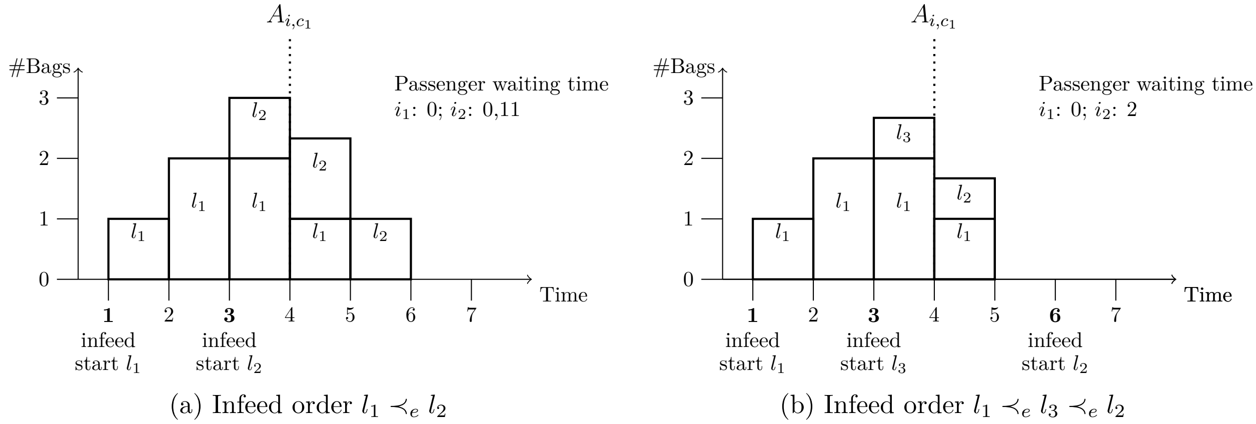


Figure 3.3: Workload and passenger waiting times at carousel c_1 before and after the assignment of trip l_3 to remote infeed station e; the dotted line indicate the arrival time of passengers at the carousel

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 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 §3.4.1). 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 (39), 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.

In §3.3.3, we have shown that the feasibility of both, the airside and landside decision problem of IBHP, are NP hard. To relax the hardness of both problems, we temporarily allow infeasibility by defining features for capacity violations at the landside and start time violations at the airside. 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 (52) - (54) (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 $I_{i,l}^{\text{ls}}(\mathbf{x})$ be 1 if the infeed of flight i's trip l starts later than its latest infeed time $\mathcal{S}_{i,l}^{\text{ls}}$. The augmented cost function

$$h(\mathbf{x}) = f(\mathbf{x}) + \sum_{c \in \mathcal{C}} p_c^{\text{cap}} \cdot I_c^{\text{cap}}(\mathbf{x}) + \sum_{i \in \mathcal{F}} \sum_{l \in \mathcal{L}_i} p_{i,l}^{\text{ls}} \cdot I_{i,l}^{\text{ls}}(\mathbf{x}) \tag{61}$$

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

$$\begin{aligned} u_c^{\text{cap}}(\mathbf{x}) &= I_c^{\text{cap}}(\mathbf{x}) \cdot \frac{(\#\text{capacity violations on carousel } c)}{1 + p_c^{\text{cap}}} \qquad \text{and} \\ u_{i,l}^{\text{ls}}(\mathbf{x}) &= I_{i,l}^{\text{ls}}(\mathbf{x}) \cdot \frac{(\text{start time of flight } i) - S_i^{\text{ls}}}{1 + p_{i,l}^{\text{ls}}} \end{aligned} \tag{62}$$

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 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 1 according to the highest utility (62) 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 (62).

Fast local search To speed up the local search we make use of fast local search (FLS) techniques (see Faroe et al. [55], Tsang and Voudouris [128] and Voudouris [131]) 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:

- $\mathcal{N}_i^1(\mathbf{x})$: For each flight i, we define $\mathcal{N}_i^1(\mathbf{x})$ 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. Moreover, we can reduce the size of neighborhood $\mathcal{N}_i^2(\mathbf{x})$ by 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 3 in Appendix A.4 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 $h^{const}(\mathbf{x})$.

3.5 Practical implementation

This section describes and discuss some of the features of the algorithm presented in §3.4 which are required when we implemented the algorithm at a major European Airport. We show in §3.5.1 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 §3.5.2). Finally, §3.5.3 discuss the requirements when implementing the heuristic in a rolling planning framework to reduce the instability of the obtained solutions.

3.5.1 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 [48] and Martel and Senviratne [90]). 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 Correia and Wirasinghe [48], 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.2. The results showed that passengers' waiting times are most relevant for the level of service. To obtain

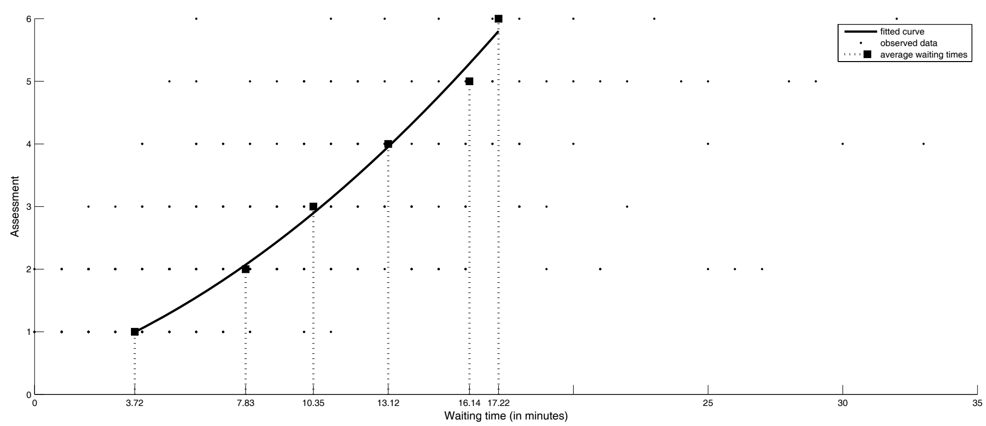


Figure 3.4: Interpolation between passengers' assessment and the realized waiting time; the size of the dots represents the number of observations in the same minute

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 3.4). The best interpolation is a polynomial function of degree 2. Hence, with increasing waiting time, passengers' tolerance of waiting decreases quadratically.

3.5.2 Multi-Objective path-relinking

The solution of the multi-objective objective function depends on weight λ (see objective function (39)). The value of λ can be based on different factors like experience, airports operational goals or customer survey evaluations (see §3.5.1). 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 <u>inc</u>rementing the weight towards λ^{inc} or by <u>dec</u>rementing 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 §3.4.2, moves between the solutions are made in a best first manner, but now the best move is evaluated 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 3.5). In the following, we call the procedure <u>m</u>ulti-<u>o</u>bjective <u>p</u>ath-<u>r</u>elinking (MOPR)

3.5.3 Rolling horizon 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,

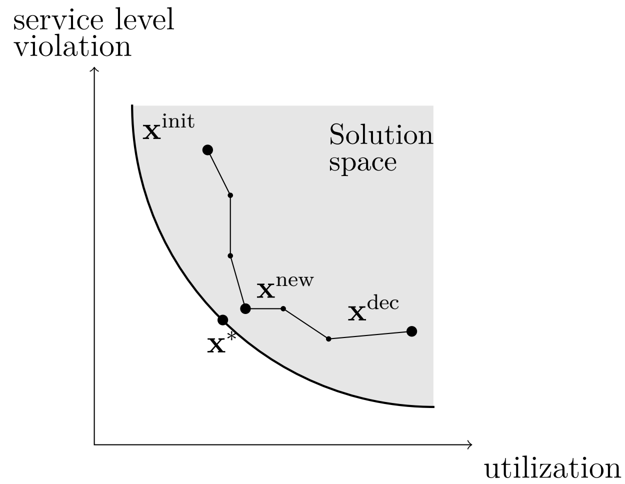


Figure 3.5: Solution space if the objective weight is decreased from λ^{init} towards λ_{dec} with $\mathbf{x}^{\text{new}}$ as best solution found when exploring the path between solutions $\mathbf{x}^{\text{init}}$ and $\mathbf{x}^{\text{dec}}$; $\mathbf{x}^{*}$ shows the optimal solution for the new weight of λ

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 as the airplane approaches 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 §3.5.2).

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.

3.6 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 3.1). The penalties $\Psi^{\text{wait}}_{l,e,c,s}$ in the objective function for passengers' waiting times are derived from the survey presented in §3.5.1.

In the remainder of this section we first evaluate the performance of the heuristic in §3.6.1, before we conduct a real-world study in §3.6.1.2 to show the performance of the algorithm when embedded in a rolling planning framework as implemented at a major European Airport.

3.6.1 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 §3.6.1.1, we compare L-IBHP with the HGGLS heuristic proposed in §3.4. The

Table 3.7: Penalties and threshold values for the utilization of L-IBHP

	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

performance of each subroutine of the heuristic and the solution quality of the multi-objective path-relinking (see §3.5.2) is studied in §3.6.1.2.

3.6.1.1 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 3.8 and 3.9, respectively. Column "λ" shows the λ-value for the objective function for each instance set "Inst". In Table 3.8 column "Time (s)" gives the required time in seconds (s) to obtain the objective value in column "f". 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 "f^*" 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 "f" and the objective value of the optimal solution. In Table 3.9 columns "GAP" and "GAP*" report the gap in % between the objective value of the heuristic solution and the solution of L-IBHP within the computational time limit as reported in column "f", and the objective function value of the optimal solution of L-IBHP reported in column "f^*", 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".

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 20 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

Table 3.8: Results for L-IBHP

							utilization ≤				waiting time ≤ (min)				
λ	Inst	F	f^*	f	Time (s)	GAP*	0.1	0.4	0.8	1.0	3	8	14	17	180
	1	20	2.01	2.01	66.35	0	23	41	21	1	42.5	30.5			16
0	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
	1	20	6.78	6.78	104.73	1.52	19	13			16.18	45.63	8.27	4.1	23.88
0.2	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
	1	20	12.08	12.08	38.11	0	15	12			13.79	35.85	17.03		33.14
0.5	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
	1	20	17.31	17.31	35.65	0	14	12			14.18	35.85	17.03		33.14
0.8	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	20	20.8	20.8	40.28	0	15	12				3.14	12.71	15.11	69.37
1	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

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.

Table 3.9: Results for the HGGLS

						utilization ≤				waiting time ≤ (min)				
λ	Inst	F	f	GAP*	GAP	0.1	0.4	0.8	1.0	3	8	14	17	200
	1	20	2.01	0	0	23	41	21	1	42.5	30.5			16
0	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
	1	20	6.96	2.65	2.65	17	13			9.95	48.5	13.5	3.65	24.5
0.2	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
	1	20	12.6	4.3	4.3	11	13			4.95	41	22.5	6.5	24.5
0.5	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
	1	20	18.18	5.03	5.03	11	13			6.45	43	19	7	25
0.8	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	20	21.7	4.32	4.32	9	13			1.43	14.35	24	13.05	47.05
1	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

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.

3.6.1.2 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 3.10, 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 3.10), and the average improvement when applying only path-relinking (see row 6 in Table 3.10). 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 3.10 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.

3.6.2 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 §3.5.3. 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.3. The algorithmic setting is equal to the setting as used in §3.6.1 with $\Delta^{\text{next}} = 180$ and $\Delta^{\text{time}} = 3$. In Table 3.11 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 interval $]0.0, 0.1]$, $]0.1, 0.4]$, $]0.4, 0.8]$ 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 (38) 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 (38) we obtain a significant level of 6%. The coefficient of variance is between 0.3 and 0.4.

3.7 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 §3.5.2) 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 passenger satisfaction, especially the impact of the trade-off between passengers' waiting time and passengers' walking distance.

Table 3.10: Performance of the GFLS and (multi-objective) path-relinking

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 3.11: Airport's solution embedded in the simulation

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
	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
Heuristic	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	

Chapter 4

Flexible cyclic rostering in the service industry

4.1 Introduction

Daily and weekly fluctuations in demand coupled with the use of a part-time flexible workforce are major factors that set apart the service sector from manufacturing where standard 8-hour shifts are the rule. The challenge for service managers is to continually balance customer satisfaction with personnel costs and worker preferences for shifts. Call centers, urban transportation systems, airports, and hospitals are examples where this balance drives staffing decisions at all levels of planning. In such environments, cyclic scheduling is often the best way to cover demand while assuring an equal allocation of work over the planning horizon. In the cyclic model, the workforce is typically divided into groups of equal size and each group is assigned the same schedule but lagged in time (see e.g., Rocha et al. 111). When a group reaches the last scheduling period, it starts from the beginning. In contrast to tour scheduling, where employees may receive a weekly or monthly non-repeating schedule on short notice, cyclic rosters offer a high degree of fairness and regularity. Everyone works the same schedule (although displaced in time) and so can see their on and off days well in advance.

Cyclic rosters, however, admit several disadvantages, the most notable being lack of management flexibility when reacting to weekly demand fluctuations or unusually high undercoverage when, say, more than a few employees in the same group are absent on the same day. In an effort to address these and related shortcomings, we present a novel approach to cyclic rostering derived from practice. In particular, we introduce the concept of a *flexible cyclic roster* which offers many of the benefits of tour scheduling while maintaining a high degree of repeatability from period to period.

In cyclic rostering, the days in the planning horizon are grouped into periods of equal size, for example, weeks or months, and each day is considered to be either on or off. For the on-days, each worker is assigned a shift, which is defined by its starting time, duration, and perhaps break characteristics. In some cases, the option may exist for a subset of "shifts" to be *off-days* rather than working days (see Kiermaier et al. 78). The number of periods in the planning horizon is generally equal to the number of groups, which, in practice can be five or more (Rocha et al. 111, Musliu 94). In contrast, the planning horizon for tour scheduling is usually much shorter; e.g., Purnomo and Bard [104] used two-week rotations to construct nurse rosters, while Brunner and Bard [34] developed weekly tours for U.S. Postal workers.

In our version of flexible cyclic rostering, we integrate tactical and short-term planning within a two-phase model. At the tactical or longer-term level, either a day off or a shift type is chosen for each day in the cyclic roster. In the second phase, just prior to the upcoming week, one shift out of the set of eligible shifts is selected for each day on in the schedule. Employees thus know their working days from one week to the next, although they may not know their actual shifts. However, they all share the same base schedule with the same days-off pattern and workload. From the scheduler's point of view, more freedom exists to match shifts with expected demand and to accommodate fluctuations in worker availability.

Formalizing the generation of flexible cyclic rosters leads to a stochastic multi-stage problem when different demand scenarios are taken into account. To tackle the curse of dimensionality, we introduce two approximations. The first is the conversion of the multi-stage problem into a two-stage problem, and the second relates to how shifts are chosen in the second stage of the latter. Nevertheless, the approximations are sufficiently general to permit the resolution of standard cyclic rostering problems and are shown to provide high-quality solutions over a wide range of scenarios. To the best of our knowledge, this is the first attempt to optimize cyclic rostering problems with probabilistic demand. Our methodology is evaluated using data obtained from a ground handling company that provides services to a large European airport as well as data that was randomly generated from the real set.

The main contributions of the chapter are as follows.

- The introduction of flexible cyclic rostering, which combines the features of tour scheduling and traditional cyclic rostering
- The presentation of a multi-stage formulation together with a two-stage approximation model covering both flexible and traditional cyclic rostering
- A comparative analysis that shows the advantages of flexible cyclic rostering over traditional cyclic rostering and the value of stochastic information for either approach based on data associated with airport ground handlers

The chapter is organized as follows. In § 5.2 we review the literature on cyclic rostering and stochastic programming for personnel scheduling. The problem of flexible cyclic rostering is described § 5.3 and the multi-stage formulation along with our two-stage approximations are given in § 4.4. The demand scenarios used to evaluate the models and scheduling procedures are described in § 4.6, while the computational comparisons are reported in § 5.6. We close with some insights and observations on the proposed methodology in § 5.7.

4.2 Literature review

For a general overview of staff scheduling problems, we refer the reader to Ernst et al. [54] and Van den Bergh et al. [129]. An early review of workforce scheduling with cyclic demand is given by Baker [18]. He examined the case where demand requirements repeated every 24 hours so it was only necessary to solve a one-day shift scheduling problem and repeat the assignments over the week. Bartholdi III et al. [28] tackled the (k, m)-cyclic staffing problem where each person works a shift of k consecutive periods and is then idle for $m - k$ periods, continuing in a cyclic fashion. Solutions were found by reformulating the problem as a bounded series of network flow problems that naturally produced integer solutions. Alternatively, they solved the linear relaxation of the original integer program and applied a rounding algorithm to attain integrality. In a follow-on study Bartholdi III [27] showed that a more general version of the problem that allowed for breaks within a shift is NP hard. He then presented a new linear programming heuristic with a novel round-off scheme that produced high quality lower bounds.

Rosenbloom and Goertzen [113] proposed a three-stage algorithm for constructing cyclic schedules for hospital nurses. In the first stage, a set of possible schedules is generated. In the second stage, an integer program is solved with the objective of minimizing shift costs while ensuring that the minimum daily coverage requirements is met. The last stage converts the second stage solution into actual work patterns for the nurses. The approach is limited in that only workstretches and days off are considered, and then only for a single shift per day.

Balakrishnan and Wong [20] present a network flow-based formulation for a cyclic rostering problem but did not permit different shift types within a workstretch. Their solution approach was based on Lagrangian duality and the enumeration of up to k-shortest paths in the network. A similar approach was followed by Millar and Kiragu [91] who tackled cyclic and non-cyclic scheduling problems for nurses by solving a shortest path problem. Because only two non-overlapping shifts were considered, they were able to efficiently enumerate all feasible workstretches. Taking an interactive approach, Musliu et al. [95] developed a four-step algorithm to generate a cyclic rosters, which was implemented in a commercial scheduling package. Their algorithm first chooses a set of lengths for consecutive work blocks and then chooses a sequence of blocks that satisfies certain weekend rules. Next, possible shift sequences for the chosen work blocks are determined, and finally the shift sequences are assigned to work blocks.

Felici and Gentile [56] introduced an integer programming model to maximize staff satisfaction that gives preference to solutions in which desirable pairs of shifts are assigned consecutively. To speed convergence, valid inequalities were introduced that tightened the LP relaxation. The approach was implemented by Alitalia to generate rosters for their ground staff. Laporte et al. [81] present an integer linear programming-based algorithm to generate cyclic rosters for police and fire departments. Three non-overlapping shifts were included in the model, and under the given constraints, shift changes were only allowed after a day off.

An integer programming model for scheduling teams was proposed by Rocha et al. [111] who used it with only minor modifications for two case studies. The first concerned a glass manufacturing facility and the second a “continuous” care unit in a hospital.

In both cases, all teams worked the same schedule but with an offset parameter that specifies the number of days between cycles. As in the above mentioned papers, their model was built around shift-based demand meaning that the number of shifts that have to be assigned to rosters is given. Shift-based demand implies that some preprocessor or planning authority determines the composition and number of shifts needed. This makes the rostering problem easier. Others have built models around period-based demand where the demand for workers is given per time period but the composition and number of shifts is not predetermined. Herbers [68] claimed to be the first to present a period-based approach. He designed a branch-and-price algorithm to solve the cyclic rostering problem and tested it using real-world airport planning scenarios. Similarly, Purnomo and Bard [104] used branch and price to generate cyclic nurse rosters with the objective of minimizing a weighted combination of undercoverage and individual preference violations related to days off patterns and back-to-back shifts changes. In light of the way rotations were defined, they were able to take advantage of two levels of aggregation to substantially reduce the size of the problem.

Lusby et al. [88] addressed a ground crew rostering problem for a major European airline that required fixed work patterns. In particular, all employees worked a 6&3 pattern in which six days on were followed by three days off. A cutting stock-based integer programming formulation was proposed and solved heuristically using a combination of decomposition, column generation, and variable fixing. Testing was done for a six-month planning horizon that was divided into smaller, overlapping blocks. A solution was then found for each block in sequence, where the solution for the overlapping portion of a predecessor block served as input to the current block.

Returning to demand uncertainty, Bard et al. [25] developed a two-stage integer, stochastic optimization model for a staff planing and scheduling problem at U.S. Postal Service mail processing centers. In the first stage, the size and composition of the permanent workforce is determined; in the second stage, demand is observed and weekly tours with overtime are generated for the staff. Instances with a 52-week planning horizon and three demand scenarios were solved. Further examples of stochastic programming as applied to workforce scheduling can be found in Zhu and Sherali [139], Robbins and Harrison [108], and Campbell [39]. For more information on stochastic programming, see Louveaux and Schultz [86], Birge and Louveaux [31], and Sen [114].

4.3 Problem description

Staff planning and scheduling problems can be defined hierarchically by their time scale and consequently divided into one of three major categories. Strategic planning is at the highest level and is concerned with the size and structure of the workforce. In this chapter, we are concerned with the tactical and operational levels where periodic schedules are constructed and then adjusted prior to implementation to take into account changes in demand and worker availability. For some organizations such as airlines and hospitals, a period is a month or four weeks; for others such as mail processing centers and big box retail chains, it is a week. In our application, it is common to plan in weekly increments.

When constructing cyclic rosters, the workforce is typically split into groups of equal size and a weekly schedule is derived for each. The groups systematically rotate through the set of schedules, so over the planning horizon (one cycle) they all have the same workload, and days on and off but in different weeks. Table 4.1 illustrates a cyclic roster for three groups with specific shifts and days off assigned for a three-week planning horizon. In week 1, group 1 works exactly the same schedule that group 2 works in week 2, and group 3 in week 3. In week 4, the cycle restarts and all groups are scheduled as in week 1. In this type of arrangement, the length of one cycle of a roster is equal to the number of groups. Note that the cycle length and therefore the group size can either be determined a priori as in our case, or it can be a decision variable whose value is determined as part of the rostering process. Although models can be developed for the latter case, they would be unnecessarily complex and intractable for instances of realistic size. A much more efficient approach would be to treat the group size as a parameter and iterative through a range of values.

In defining flexible cyclic rosters, we make use of the term ‘shift type,’ which is a predetermined set of shifts with general characteristics. Examples might be *early shifts*, *late shifts* and *12-hour shifts*, where, say, an early shift type includes all feasible shifts starting between 6 am and 9 am with a 7- to 9-hour duration. In contrast to a cyclic roster, a flexible cyclic roster specifies shift types rather than shifts. As such, each day in the planning horizon is either designated as a day off or is assigned a specific shift type. An example of a flexible cyclic roster is given in Table 4.2, where ‘E’ and ‘L’ stand for early shift types and late shift types, respectively. Note that the pattern of days on and off in Tables 4.1 and 4.2 is the same.

Table 4.1: Example of a cyclic roster for three weeks and groups

Group	**Calendar week**	**Mo**	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**
1	1	7a-3p	7.30a-3p	7.30a-3p	1p-9p	2p-9p		
	2			7a-2p	7.30a-3p	8a-3p	3p-9p	3p-9p
	3	1p-9p	1p-10p				7a-3p	7a-3p
2	1	1p-9p	1p-10p				7a-3p	7a-3p
	2	7a-3p	7.30a-3p	7.30a-3p	1p-9p	2p-9p		
	3			7a-2p	7.30a-3p	8a-3p	3p-9p	3p-9p
3	1			7a-2p	7.30a-3p	8a-3p	3p-9p	3p-9p
	2	1p-9p	1p-10p				7a-3p	7a-3p
	3	7a-3p	7.30a-3p	7.30a-3p	1p-9p	2p-9p		

Table 4.2: Example of a flexible cyclic roster for three weeks and groups

Group	**Calendar week**	**Mo**	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**
1	1	E	E	E	L	L		
	2			E	E	E	L	L
	3	L	L				E	E
2	1	L	L				E	E
	2	E	E	E	L	L		
	3			E	E	E	L	L
3	1			E	E	E	L	L
	2	L	L				E	E
	3	E	E	E	L	L		

The roster in Table 4.2 is really a set of weekly templates and not a set of schedules. Prior to the week of operation, the scheduler must choose an appropriate shift for each working day for each group. A possible realization of such a roster for groups 1 to 3 in calendar week 2 is given in Table 4.3.

Table 4.3: Example of a final schedule in calendar week 2

Group	**Days of the week**						
	Mo	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**
1			7a-3.30p	7.30a-3p	7.30p-3.30p	3p-10p	3p-10p
3	6a-2p	6a-2p	7a-3p	2p-9p	2.30p-9p		
2	2p-10p	2p-10p				7a-3.30p	7.30a-4p

Hierarchically, we have the following two-step process.

- Tactical level: Generation of flexible cyclic rosters and assignment of workers to groups
- Operational level: Weekly tour generation based on the cyclic rosters

In the first step, a flexible cyclic roster is constructed and is then finalized in the second step where a weekly tour is specified for each group. Of course, legal restrictions, organizational policies, and perhaps union agreements must be respected if the schedules are to be feasible. For our application, the following regulations must be observed:

W1. The number of workdays per week is within a specified range.

W2. The overall number of working hours per week is within a specified range.

W3. For any given day, all workers in the same group work the same shift

C1. The maximum number of consecutive working days is limited.

C2. No work on at least A out of B consecutive weekends.

C3. The assigned shift types and the start time of a shift must be non-decreasing from one day to the next when the days are consecutive, i.e., the working days are subject to *forward rotation*.

While constraints W1 to W3 can be considered independently for each week in the planning horizon, constraints C1 to C3 apply to consecutive days and weeks. The forward rotation for shift types, C3, means that when a particular shift is assigned on day d an earlier shift cannot be assigned on day $d+1$. The next section provides a stochastic model for the generation of traditional and flexible cyclic rosters.

4.4 Model development

In Section 4.4.1, the notation used in developing the basic model is presented. This is followed in Section 4.4.2 with a multi-stage stochastic mixed-integer programming formulation for the flexible cyclic rostering problem.

4.4.1 Notation

The required sets and parameters are summarized in Tables 5.2 and 4.5, respectively. As previously mentioned, the planning horizon is divided into weeks, or more generally, time periods of equal length, and the number of weeks in a cycle is equal to the number of groups, which is denoted by G. For a 10-week problem, for example, $|G|= 10$. The workforce is partitioned amongst the groups and has to cover the demand over a planning horizon of W weeks. In our model, a day is divided into forty-eight 30-minute periods, and the number of workers required in each period is considered to be an input parameter, at least in the deterministic version of the problem. This is opposed to shift-based demand where the number and type of shifts are given and worker demand is specified for each (e.g., see Bard et al. 25, Rocha et al. 111).

Table 4.4: Sets and indices

Sets	Definition
$\mathcal{P} = \{1, ..., P\}$	set of shift types (e.g., early/day/late)
$\mathcal{S} = \{1, ..., S\}$	set of shifts
$\mathcal{S}(p) \subseteq \mathcal{S}$	set of shifts associated with shift type p
$\mathcal{T} = \{1, ..., T\}$	set of time periods per day
$\mathcal{D} = \{1, ..., D\}$	set of days per week
$\mathcal{W} = \{1, ..., W\}$	set of weeks in the planning horizon
$\mathcal{G} = \{1, ..., G\}$	set of groups

Table 4.5: Model parameters

Parameter	Definition
$a_{s,t}$	number of workers per group if shift s covers period t, 0 otherwise
$d^{\min}$	minimum number of working days per week allowed
$d^{\max}$	maximum number of working days per week allowed
f_s	earliest period covered by shift s
f_p	earliest period covered by any shift associated with shift type p
l_s	number of periods covered by shift s
$\widetilde{\mathbf{D}}$	random vector for demand
$\widetilde{D}_{d,t}$	random variable for demand on day d in period t
$n^{\max}$	maximum number of consecutive working days allowed
$h^{\max}$	maximum number of working periods per week allowed
$h^{\min}$	minimum number of working periods per week allowed
M	arbitrarily large positive number

Demand patterns for service workers, such as airport ground handlers and retail clerks, can vary widely over the day, week and season. For tactical planning, the robustness of generated plans is a critical factor in controlling labor costs. To account for demand fluctuations and uncertainty from one period to the next, we present below a stochastic programming model that can be used to construct cyclic rosters for a fixed workforce.

It is assumed that the demand in each week $w \in \mathcal{W}$ is independent of the demand in the other weeks, and that each week has the same demand distribution. The random vector for the demand is denoted by $\widetilde{\mathbf{D}} = \left(\widetilde{D}_{d,t}\right)_{d\in\mathcal{D},t\in\mathcal{T}}$ and $\widetilde{\mathbf{D}}(\omega) = \left(\widetilde{D}_{d,t}(\omega)\right)_{d\in\mathcal{D},t\in\mathcal{T}}$ gives a realization for the demand of a sample point or scenario $\omega \in \Omega$, where Ω represents the finite support or sample space. The corresponding probability mass function is give by $P\left(\widetilde{\mathbf{D}} = \widetilde{\mathbf{D}}(\omega)\right) = p(\omega)$.

When selecting realizations for the demand, several different approaches may be followed. In practice as well as in literature, a single scenario representing a 'typical week' is often favored. In such cases, historical averages or even forecasts are taken as input. When a robust solution is called for, a worst case scenario may be chosen. In our analysis, we include up to 10 realizations reflecting high and low demand curves, as well as small and large fluctuations over the week. Each is discussed in Section 4.6.

4.4.2 Multi-stage modeling

Given that the demand curve is uncertain from week to week, the problem can be modeled as a stochastic program (see Louveaux and Schultz 86). In our study, demand is the only random parameter. In the first stage of the proposed model, a flexible cyclic roster is generated that satisfies regulations W1 to W3 as well as C2 and C3 prior to knowing the demand in the upcoming weeks $w \in W$. Once the roster is fixed, we observe demand $\widetilde{\mathbf{D}}(\omega)$ for $\omega \in \Omega$ and make use of shift flexibility to minimize undercoverage by setting the shift start times over the week for each worker. Due to the forward rotation requirement C2, the schedule for Sunday in week w affects the shift start time decisions of Monday in week $w+1$ as well as all upcoming days within this week. Hence, the decision taken in week w is dependent on the decision in week $w-1$ for $2 \leq w \leq W$ leading to a multi-stage stochastic program.

In defining the variables, it is instructive to separate them into two categories. Those in the first category are used to assign a shift type $p \in \mathcal{P}$ to a group $g \in \mathcal{G}$ on each day $d \in \mathcal{D}$, or to designate d as a day off. In light of these assignments, the variables in the second category are used to (i) select a specific shift $s \in \mathcal{S}(p)$ on each day $d \in \mathcal{D}$ that is not a day off, (ii) deal with weekend constraints, and (iii) account for undercoverage in each period $t \in \mathcal{T}$. Specifically, we have

$x_{p,g,d} = 1$, if shift type p is assigned to group g on day d, 0 otherwise

$y_{g,s,w,d}(\omega) = 1$, if shift s is assigned to group g on day d in week w for random event ω, 0 otherwise

$v_g = 1$, if at least one weekend shift type is assigned for group g, 0 otherwise

$u_{w,d,t}(\omega) =$ undercoverage in period t of day d of week w for random event ω

Multi-stage stochastic cyclic rostering problem (M-SCRP)

$$\text{Minimize} \quad \mathbb{E}_{\widetilde{\mathbf{D}}}\left[h_1(x, \widetilde{\mathbf{D}})\right] = z_{\mathrm{M}} \tag{63}$$

subject to

$$\sum_{p \in P} x_{p,g,d} \leq 1 \quad \forall\, w \in \mathcal{W},\ d \in \mathcal{D} \tag{64}$$

$$d^{\min} \leq \sum_{p\in\mathcal{P}} \sum_{d\in\mathcal{D}} x_{p,g,d} \leq d^{\max} \quad \forall\ g \in \mathcal{G} \tag{65}$$

$$\sum_{\delta=d}^{d+n^{\max}} \sum_{p\in\mathcal{P}} x_{p,g,\delta} \leq n^{\max} \quad \forall\ d \leq D - n^{\max},\ g \in \mathcal{G} \tag{66}$$

$$\sum_{\delta=d}^{D} \sum_{p\in\mathcal{P}} x_{p,g,\delta} + \sum_{\delta=1}^{n^{\max}-(D-d)} \sum_{p\in\mathcal{P}} x_{p,g+1,\delta} \leq n^{\max} \quad \forall\ D - n^{\max} < d \leq D,\ g \in \mathcal{G} \setminus \{10\} \tag{67}$$

$$\sum_{\bar{d}=d}^{D} \sum_{p\in\mathcal{P}} x_{p,G,d} + \sum_{\delta=1}^{n^{\max}-(D-d)} \sum_{p\in\mathcal{P}} x_{p,1,\delta} \leq n^{\max} \quad \forall\ D - n^{\max} < d \leq D \tag{68}$$

$$\sum_{p\in\mathcal{P}} (x_{p,g,D-1} + x_{p,g,D}) \leq 2 \cdot v_g \quad \forall\ g \in \mathcal{G} \tag{69}$$

$$\sum_{\psi\in\{g,..,g+B-1\}} v_\psi \leq B - A \quad \forall\ g \in \{1,..,G-B+1\} \tag{70}$$

$$\sum_{\psi\in\{g,..,G\}} v_\psi + \sum_{\psi\in\{1,..,B-(G-g)-1\}} v_\psi \leq B - A \quad \forall\ g \in \{G-B+2,..,G\} \tag{71}$$

$$\sum_{p\in\mathcal{P}} f_p \cdot x_{p,g,d} \leq \sum_{p\in\mathcal{P}} f_p \cdot x_{p,g,d+1} + \left(1 - \sum_{p\in\mathcal{P}} x_{p,g,d+1}\right) \cdot M \quad \forall\ g \in \mathcal{G},\ d \in \mathcal{D} \setminus \{D\} \tag{72}$$

$$\sum_{p\in\mathcal{P}} f_p \cdot x_{p,g,D} \leq \sum_{p\in\mathcal{P}} f_p \cdot x_{p,g+1,1} + \left(1 - \sum_{p\in\mathcal{P}} x_{p,g+1,1}\right) \cdot M \quad \forall\ g \in \mathcal{G} \setminus \{G\} \tag{73}$$

$$\sum_{p\in\mathcal{P}} f_p \cdot x_{p,G,D} \leq \sum_{p\in\mathcal{P}} f_p \cdot x_{p,1,1} + \left(1 - \sum_{p\in\mathcal{P}} x_{p,1,1}\right) \cdot M \tag{74}$$

$$x_{p,g,d}, v_g \in \{0,1\} \quad \forall\ p \in \mathcal{P},\ g \in \mathcal{G},\ d \in \mathcal{D} \tag{75}$$

where for $2 \leq w \leq W$, we have

$$h_w\left(x, y_{w-1}, \widetilde{\mathbf{D}}(\omega)\right) = \text{minimize} \sum_{d\in\mathcal{D}} \sum_{t\in\mathcal{T}} u_{w,d,t}(\omega) + \mathbb{E}_{\widetilde{\mathbf{D}}}\left[h_{w+1}\left(x, y_w, \widetilde{\mathbf{D}}\right)\right] \tag{76}$$

$$\text{subject to} \sum_{s\in\mathcal{S}(p)} y_{g,s,w,d}(\omega) = x_{p,g,d} \quad \forall\ p \in \mathcal{P},\ g \in \mathcal{G},\ d \in \mathcal{D} \tag{77}$$

$$h^{\min} \leq \sum_{d\in\mathcal{D}} \sum_{s\in\mathcal{S}} \sum_{t\in\mathcal{T}} a_{s,t} \cdot y_{g,s,w,d}(\omega) \leq h^{\max} \quad \forall\ g \in \mathcal{G} \tag{78}$$

$$\sum_{g\in\mathcal{G}} \sum_{s\in\mathcal{S}} a_{s,t} \cdot y_{g,s,w,d}(\omega) + u_{w,d,t}(\omega) \geq \widetilde{D}_{d,t}(\omega) \quad \forall\ d \in \mathcal{D},\ t \in \mathcal{T} \tag{79}$$

$$\sum_{s\in\mathcal{S}} f_s \cdot y_{g,s,w,d}(\omega) \leq \sum_{s\in\mathcal{S}} f_s \cdot y_{g,s,w,d+1}(\omega) + \left(1 - \sum_{s\in\mathcal{S}} y_{g,s,w,d+1}(\omega)\right) \cdot M \quad \forall\, d \in \mathcal{D} \setminus \{D\} \tag{80}$$

$$\sum_{s\in\mathcal{S}} f_s \cdot y_{g,s,w-1,D} \leq \sum_{s\in\mathcal{S}} f_s \cdot y_{g,s,w,1}(\omega) + \left(1 - \sum_{s\in\mathcal{S}} y_{g,s,w,1}(\omega)\right) \cdot M \quad \forall\, g \in \mathcal{G} \tag{81}$$

$$y_{g,s,w,d}(\omega), v_w \in \{0,1\} \quad \forall\, g \in \mathcal{G},\ s \in \mathcal{S},\ w \in \mathcal{W},\ d \in \mathcal{D} \tag{82}$$

$$u_{w,d,t}(\omega) \in \mathbb{Z}_+ \quad \forall\, w \in \mathcal{W},\ d \in \mathcal{D},\ t \in \mathcal{T} \tag{83}$$

and for $w = 1$ we have

$$h_w\left(x, \widetilde{\mathbf{D}}(\omega)\right) = \text{minimize} \quad \sum_{d\in\mathcal{D}} \sum_{t\in\mathcal{T}} u_{w,d,t}(\omega) + \mathbb{E}_{\widetilde{\mathbf{D}}}\left[h_2\left(x, y_1, \widetilde{\mathbf{D}}\right)\right] \tag{84}$$

subject to (77) – (80), (82), (83)

with $h_{W+1}(x, y_W, \widetilde{\mathbf{D}}(\omega)) \equiv 0$ for all $\omega \in \Omega$.

The objective functions (63), (76) and (84) minimize the expected uncovered demand across the W planning weeks. In what follows, we describe the constraints for the (deterministic) first stage and (stochastic) subsequent stages.

First stage In light of constraints (64), at most one shift type is selected for each day in the W-week schedule. Constraints (65) bound the minimum and maximum number of working days per week as required by W1. Constraints (66) – (68) bound the maximum number of consecutive working days to at most $n^{\max}$ as required by C1. Constraints (66) enforce the maximum consecutive working days within a week, while (67) account for the transition from one week to the next except for the transition from the last week to the first which is modeled in constraints (68). The A out of B weekends requirement as specified by C2 is modeled by (69) – (71). Constraints (69) determine whether a shift is assigned on a weekend within a week. Constraints (70) and (71) enforce the weekend requirement for B consecutive weeks within the same cycle and account for the transition from one cycle to the next, respectively. Constraints (72) – (74) assign shift types in accordance with the forward rotation requirement C3 , e.g., an early shift cannot immediately follow a late shift. Constraints (72) establish forward rotation within each week, while (73) account for transitions between consecutive weeks, and (74) take care of the transition from the last week in the cycle to the first week. The variables are defined in (75).

Stages $1 \leq w \leq W$ Constraints (77) serve as a link between shift and the shift types selected in the first stage for each day. These constraints connect the first stage decisions with the decisions taken in the stochastic stages 1 to W. Thus, $x_{p,g,d}$ can bee seen as a parameter. The minimum and maximum number of working hours per week are bounded by constraints (78), as required by W2. The undercoverage for each random event ω with respect to the selected shift start times is accounted for by the variables $u_{w,d,t}(\omega)$ in probabilistic constraints (79). Note that the group size is reflected in the parameter $a_{s,t}$, which is equal to the number of workers per group if shift s covers period t and 0 otherwise. Thus, (79) implies that all members of a group work the same schedule.

Constraints (80) – (81) are similar to (72) – (74) and enforce non-decreasing shift starting times from one day to the next and hence guarantee forward rotation throughout the roster. More specifically, (80) guarantees forward rotation within each week, while (81) accounts for transitions between consecutive weeks in which $y_{g,s,w-1,d}$ represents a parameter for the decision taken in the previous stage, that is, week $w-1$. The variables are defined in (82) and (83).

It can easily be shown that each of the W models (76) – (83) is NP-hard (see Lau 82). In theory, M-SCRP can be solved by dynamic programming when the set of scenarios Ω is finite. In such a case, we would have to solve $|\Omega|^W$ problems to evaluate all scenario realizations across W weeks for one given solution of the first stage model (63) – (75). If, for example, only 2 demand scenarios were considered for a week, there would be 2^{10} possible realizations across a 10-week planning horizon. To make the problem tractable for real-world applications, two approximations are introduced.

4.5 Model approximations

In the first approximation, given in Section 4.5.1, we relax the forward rotation constraints (81) and assume the same demand realization $\widetilde{\mathbf{D}}(\omega)$ in all weeks of the planning horizon. This reduces the number of scenarios from $|\Omega|^W$ to $|\Omega|$. In the second approximation, given in Section 4.5.2, we limit the options available for selecting shifts in the second stage of the reduced model. The aim is to further decrease model complexity and hence ease the computational burden.

4.5.1 Two-stage model approximation

In the M-SCRP, the decisions between the W stages are dependent due to forward rotation constraints (81). To make them independent, we relax constraints (81) by assuming that the same random event ω occurs in all weeks $w \in \mathcal{W}$; that is, the demand realization in each week in the planning horizon is identical. As a result, the W multistage stochastic problems are now independent and identical and can be replaced with a single problem with $|\Omega|$ scenarios. This leads to a two-stage stochastic integer program, which we now define using the following modified notation.

$y_{g,s,d}(\omega) = 1$, if shift s is assigned to group g on day d for random event ω, 0 otherwise

$u_{d,t}(\omega) =$ undercoverage in period t of day d for random event ω

Two-stage SCRP (2-SCRP)

$$\text{Minimize} \quad W \cdot \mathbb{E}_{\widetilde{\mathbf{D}}}\left[h(x, \widetilde{\mathbf{D}})\right] = z_2 \tag{85}$$

$$\text{subject to} \quad (64) - \ (75)$$

where

$$h(x, \widetilde{\mathbf{D}}(\omega)) = \text{minimize} \quad \sum_{d \in \mathcal{D}} \sum_{t \in \mathcal{T}} u_{d,t}(\omega) \tag{86}$$

$$\text{subject to} \quad (82), (83)$$

$$\sum_{s \in \mathcal{S}(p)} y_{g,s,d}(\omega) = x_{p,g,d} \quad \forall\ p \in \mathcal{P},\ g \in \mathcal{G}, d \in \mathcal{D} \tag{87}$$

$$\sum_{s \in \mathcal{S}} \sum_{g \in \mathcal{G}} a_{s,t} \cdot y_{g,s,d}(\omega) + u_{d,t}(\omega) \geq \widetilde{D}_{d,t}(\omega) \quad \forall\ d \in \mathcal{D},\ t \in \mathcal{T} \tag{88}$$

$$h^{\min} < \sum_{d \in \mathcal{D}} \sum_{s \in \mathcal{S}} \sum_{t \in \mathcal{T}} a_{s,t} \cdot y_{g,s,d}(\omega) \leq h^{\max} \quad \forall\ g \in \mathcal{G} \tag{89}$$

$$\sum_{s \in \mathcal{S}} f_s \cdot y_{g,s,d}(\omega) \leq \sum_{s \in \mathcal{S}} f_s \cdot y_{g,s,d+1}(\omega) + \left(1 - \sum_{s \in \mathcal{S}} y_{g,s,d+1}(\omega)\right) \cdot M \quad \forall\ g \in \mathcal{G},\ d \in \mathcal{D} \setminus \{D\} \tag{90}$$

$$\sum_{s \in \mathcal{S}} f_s \cdot y_{g,s,D}(\omega) \leq \sum_{s \in \mathcal{S}} f_s \cdot y_{g,s,1}(\omega) + \left(1 - \sum_{s \in \mathcal{S}} y_{g,s,1}(\omega)\right) \cdot M \quad \forall\, g \in \mathcal{G} \tag{91}$$

When only one scenario is considered, i.e., $|\Omega| = 1$, the solution to 2-SCRP provides an upper bound for the M-SCRP (see Theorem 6 in the next subsection). However, because 2-SCRP does not consider the transition between two weeks with different demand realizations, its solution is less robust than that of M-SCRP. The reduction of the scenario space from $|\Omega|^W$ to $|\Omega|$, though, makes the model more tractable for real-world instances. When more than one scenario is considered, i.e., $|\Omega| > 1$, we cannot make a statement whether the objective function value derived from the two-stage model approximation is lower, higher or equal to the M-SCPR's objective function value. Hence, 2-SCRP does not constitute a lower or upper bound for M-SCRP.

Consider a small example where $|\mathcal{T}| = 2$ with only one shift type that allows for two different shifts, i.e., $\mathcal{S} = \{1, 2\}, \mathcal{P} = \{1\}, \mathcal{S}(1) = \{1, 2\}$. Shift 1 covers the first period of the day while shift 2 covers the second period, implying that $a_{1,1} = 1, a_{1,2} = 0, a_{2,1} = 0$, and $a_{2,2} = 1$. Without loss of generality, we consider only two days ($\mathcal{D} = \{1, 2\}$) and assume that the group size is equal to 1 ($|\mathcal{G}| = 1$), there are two weeks in the planning horizon ($|\mathcal{W}| = 2$), and there are two scenarios ($\Omega = \{\omega_1, \omega_2\}$) with both random events having the same likelihood in each $w \in \mathcal{W}$. Finally, we assume no restrictions on the number of working days per week. Let z^*_{M} and z^*_2 be the optimal solutions to M-SCRP and 2-SCRP, respectively. To show that z^*_{M} can be larger than z^*_2, we assume that for ω_1 and ω_2 we get demand realizations $\widetilde{\mathbf{D}}(\omega_1)$ with $\widetilde{D}_{d,1}(\omega_1) = 0, \widetilde{D}_{d,2}(\omega_1) = 1$ for all $d \in \mathcal{D}$ and $\widetilde{\mathbf{D}}(\omega_2)$ with $\widetilde{D}_{d,1}(\omega_2) = 1, \widetilde{D}_{d,2}(\omega_2) = 0$ for all $d \in \mathcal{D}$, respectively. Table 4.6 enumerates the different combinations of random events for both weeks and shows the resulting demand realizations as well as resulting undercoverage for the optimal solution for M-SCRP and 2-SCRP. For an optimal solution of the problem, demand that cannot be covered due to forward rotation constraints is circled. For this instance we get $z^*_{\mathrm{M}} = (0{+}1{+}0{+}0)/4 = 0.25$ and $z^*_2 = 2 \cdot (0 + 0)/2 = 0$.

In contrast, we show that it is possible for $z^*_{\mathrm{M}} < z^*_2$ by assuming that for ω_1 and ω_2 we get demand realizations $\widetilde{\mathbf{D}}(\omega_1)$ with $\widetilde{D}_{0,1}(\omega_1) = 0, \widetilde{D}_{0,2}(\omega_1) = 1, \widetilde{D}_{1,1}(\omega_1) = 1, \widetilde{D}_{1,2}(\omega_1) = 0$ and $\widetilde{\mathbf{D}}(\omega_2)$ with $\widetilde{D}_{0,1}(\omega_2) = 1, \widetilde{D}_{0,2}(\omega_2) = 0, \widetilde{D}_{1,1}(\omega_2) = 0, \widetilde{D}_{1,2}(\omega_2) = 1$, respectively.

Table 4.6: Example 1 – objective function values for M-SCRP and 2-SCRP

(a) M-SCRP

Realizations		Demand								under-
Week 1	Week 2	Week 1				Week 2				coverage
		Day 1		Day 2		Day 1		Day 2		
		1	2	1	2	1	2	1	2	
ω_1	ω_1	0	1	0	1	0	1	0	1	0
ω_1	ω_2	0	1	0	1	1	0	1	0	1
ω_2	ω_1	1	0	1	0	0	1	0	1	0
ω_2	ω_2	1	0	1	0	1	0	1	0	0

(b) 2-SCRP

Realizations		Demand				under-
Week 1	Week 2	Day 1		Day 2		coverage
		1	2	1	2	
ω_1	ω_1	0	1	0	1	0
ω_2	ω_2	1	0	1	0	0

Table 4.7 depicts the different combinations of random events for both weeks and shows the resulting demand realizations as well as resulting undercoverage for the optimal solution for M-SCRP and 2-SCRP. For this instance we get $z^*_{\mathrm{M}} = (2 + 1 + 1 + 0)/4 = 1$ and $z^*_2 = 2 \cdot (1 + 1)/2 = 2$.

Symmetry constraints When the number of scenarios is not too large, the two-stage model can be solved as a single integer program by enumerating (85) – (91) for all scenarios. For any optimal solution, there are at least $|\mathcal{W}| - 1$ symmetric alternative solutions with the same objective function value. This can be seen by rotating the weekly schedules forward without changing their order. To reduce the symmetry, we introduce a set of symmetry-breaking constraints.

$$\sum_{\omega\in\Omega}\sum_{d\in\mathcal{D}}\sum_{s\in\mathcal{S}} l_s \cdot y_{1,s,d}(\omega) \geq \sum_{\omega\in\Omega}\sum_{d\in\mathcal{D}}\sum_{s\in\mathcal{S}} l_s \cdot y_{g,s,d}(\omega) \quad \forall \quad g \in \mathcal{G} \setminus \{1\} \tag{92}$$

Constraints (92) require that the expected number of total working hours in the first week of the flexible cyclic roster, i.e., for $g = 1$, are greater than or equal to the expected

Table 4.7: Example 2 – objective function values for M-SCRP and 2-SCRP

(a) M-SCRP

Realizations		Demand								under-
Week 1	Week 2	Week 1				Week 2				coverage
		Day 1		Day 2		Day 1		Day 2		
		1	2	1	2	1	2	1	2	
ω_1	ω_1	0	1	1	0	0	1	1	0	2
ω_1	ω_2	0	1	1	0	1	0	0	1	1
ω_2	ω_1	1	0	0	1	0	1	1	0	1
ω_2	ω_2	1	0	0	1	1	0	0	1	0

(b) 2-SCRP

Realizations		Demand				under-
Week 1	Week 2	Day 1		Day 2		coverage
		1	2	1	2	
ω_1	ω_1	0	1	1	0	1
ω_2	ω_2	1	0	0	1	1

number of total working hours in any of the other $G-1$ weeks of the schedule.

4.5.2 Shift flexibility relaxation

We now further relax 2-SCRP by removing the possibility of adjusting the shift parameters to better match different demand realizations. Instead we choose those shifts that offer the best coverage over all demand realizations and then make adjustments in a post-processing step. In other words, we don't consider the full range of possibilities that a shift type offers. For each day of the planning horizon, the model assigns a specific shift instead of a shift type as is the case when solving the classical cyclic shift scheduling problem. Instead of selecting an early or late shift type for each day, for example, a single shift in the set $\mathcal{S}$ is selected that meets requirements W1 to W3 and C1 to C2. The goal is to cover the demand $\widetilde{\mathbf{D}}(\omega)$ for all $\omega \in \Omega$ as best as possible without adjusting shift starting times to match the different scenarios. In the new formulation, we use the decision variable $x_{s,g,d}$,where $x_{s,g,d} = 1$, if shift s is assigned to day d for group g, 0 otherwise.

Relaxed SCRP (R-SCRP)

$$\text{Minimize} \quad W \cdot \mathbb{E}_{\widetilde{\mathbf{D}}} \left[\sum_{\omega \in \Omega} \sum_{d \in \mathcal{D}} \sum_{t \in \mathcal{T}} u_{d,t}(\omega) \right] = z_{\mathrm{R}} \tag{93}$$

subject to

$$\sum_{s \in \mathcal{S}} x_{s,g,d} \leq 1 \quad \forall\, g \in \mathcal{G},\ d \in \mathcal{D} \tag{94}$$

$$d^{\min} \leq \sum_{s \in \mathcal{S}} \sum_{d \in \mathcal{D}} x_{s,g,d} \leq d^{\max} \quad \forall\, g \in \mathcal{G} \tag{95}$$

$$\sum_{\delta=d}^{d+n^{\max}} \sum_{s \in \mathcal{S}} x_{s,g,\delta} \leq n^{\max} \quad \forall\, g \in \mathcal{G},\ d \leq D - n^{\max} \tag{96}$$

$$\sum_{\delta=d}^{D} \sum_{s \in \mathcal{S}} x_{s,g,\delta} + \sum_{\delta=1}^{n^{\max}-(D-d)} \sum_{s \in \mathcal{S}} x_{s,g+1,\delta} \leq n^{\max} \quad \forall\, g \in \mathcal{G} \setminus \{G\},\ D - n^{\max} < d \leq D \tag{97}$$

$$\sum_{\delta=d}^{D} \sum_{s \in \mathcal{S}} x_{s,G,\delta} + \sum_{\delta=1}^{n^{\max}-(D-d)} \sum_{s \in \mathcal{S}} x_{s,1,\delta} \leq n^{\max} \quad \forall\, D - n^{\max} < d \leq D \tag{98}$$

$$\sum_{s \in \mathcal{S}} (x_{s,g,D-1} + x_{s,g,D}) \leq 2 \cdot v_g \quad \forall\, g \in \mathcal{G} \tag{99}$$

$$\sum_{\psi \in \{g,..,g+B-1\}} v_\psi \leq B - A \quad \forall\, g \in \{1,..,G-B+1\} \tag{100}$$

$$\sum_{\psi \in \{g,..,G\}} v_\psi + \sum_{\psi \in \{1,..,B-(G-g)-1\}} v_\psi \leq B - A \quad \forall\, g \in \{G-B+2,..,G\} \tag{101}$$

$$\sum_{s \in \mathcal{S}} f_s \cdot x_{s,g,d} \leq \sum_{s \in \mathcal{S}} f_p \cdot x_{s,g,d+1} + \left(1 - \sum_{s \in \mathcal{S}} x_{s,g,d+1}\right) \cdot M \quad \forall\, g \in \mathcal{G},\ d \in \mathcal{D} \setminus \{D\} \tag{102}$$

$$\sum_{s \in \mathcal{S}} f_s \cdot x_{s,g,D} \leq \sum_{s \in \mathcal{S}} f_s \cdot x_{s,g+1,1} + \left(1 - \sum_{s \in \mathcal{S}} x_{s,g+1,1}\right) \cdot M \quad \forall\, g \in \mathcal{G} \setminus \{G\} \tag{103}$$

$$\sum_{s \in \mathcal{S}} f_s \cdot x_{s,G,D} \leq \sum_{s \in \mathcal{S}} f_p \cdot x_{s,1,1} + \left(1 - \sum_{p \in \mathcal{P}} x_{s,1,1}\right) \cdot M \tag{104}$$

$$\sum_{s \in \mathcal{S}} \sum_{g \in \mathcal{G}} a_{s,t} \cdot x_{s,g,d} + u_{d,t}(\omega) \geq \widetilde{D}_{d,t}(\omega) \quad \forall\, \omega \in \Omega,\ d \in \mathcal{D},\ t \in \mathcal{T} \tag{105}$$

$$h^{\min} < \sum_{d \in \mathcal{D}} \sum_{s \in \mathcal{S}} \sum_{t \in \mathcal{T}} a_{s,t} \cdot x_{s,g,d} \leq h^{\max} \quad \forall\, g \in \mathcal{G} \tag{106}$$

$$\sum_{d \in \mathcal{D}} \sum_{s \in \mathcal{S}} l_s \cdot x_{s,1,d} \geq \sum_{d \in \mathcal{D}} \sum_{s \in \mathcal{S}} l_s \cdot x_{s,g,d} \quad \forall\, g \in \mathcal{G} \setminus \{1\} \tag{107}$$

$$x_{s,g,d}, v_g \in \{0,1\} \quad \forall\ s \in \mathcal{S},\ g \in \mathcal{G},\ d \in \mathcal{D} \tag{108}$$

$$u_{w,d,t} \in \mathbb{Z}_+ \quad \forall\ w \in \mathcal{W},\ d \in \mathcal{D},\ t \in \mathcal{T} \tag{109}$$

The objective function (93) minimizes expected uncovered demand. In contrast to constraints (64) – (74), constraints (94)– (104) fix the shifts instead of shift types such that requirement W1 to W2 and C1 to C3 are satisfied. The demand coverage and, hence, requirement W3 as well as the range for working hours is modeled in constraints (105) and (106). The last constraints (107) are similar to the symmetry constraints (92).

Once a solution to R-SCRP is obtained, a post-processor is called to associate each shift to a shift type. This operation can be performed in constant time by choosing a shift type which is feasible in terms of the forward rotation requirement. While the assignment is trivial when shift types don't overlap, when they do, we use a straightforward approach that assigns shift type p to shift $s \in \mathcal{S}(p)$ such that $f_p \leq f_{p'}$ for all $p' \in \mathcal{P}$, i.e., we assign the eligible shift type with the earliest possible shift starting time. Alternatively, one could choose the shift type that allows for the largest set of shifts, i.e., shift type p such that $s \in \mathcal{S}(p)$ and $|\mathcal{S}(p)| \geq \mathcal{S}(p)$ for all $p' \in \mathcal{P}$.

R-SCRP does not consider shift types and thus contains fewer variables and constraints than does M-SCRP and 2-SCRP. However, R-SCRP provides an upper bound on the optimal solution to 2-SCRP.

Theorem 6. *Let z^*_M, z^*_2 and z^*_R be the optimal solution values for M-SCRP, 2-SCRP and R-SCRP, respectively. Then,*

(i) $z^*_R \geq z^*_2$, *if* $|\Omega| \geq 2$;

(ii) $z^*_M \leq z^*_R = z^*_2$, *if* $|\Omega| = 1$.

Proof. To prove that $z^*_{\mathrm{R}} \geq z^*_2$ for $|\Omega| \geq 2$ in part (i), we first point out that each feasible solution (x, u) to R-SCRP is also feasible to 2-SCRP. This can be seen by setting $y_{g,s,d}(\omega) = x_{s,g,d}\ \forall g \in \mathcal{G}, s \in \mathcal{S}, d \in \mathcal{D}, \omega \in \Omega$ in 2-SCRP. Next, we show that there are instances for which $z^*_{\mathrm{R}} > z^*_2$ for $|\Omega| \geq 2$. Consider the three shift types early, day and late with corresponding shifts given in Table 4.8 for one group with one worker. Further consider the two discrete realizations for the demand given in Table 4.9, where each realization has a probability of 0.5. The solution of R-SCRP is any one of the shifts defined for shift type 'early' with an expected undercoverage of 4. Thus, after post-processing we would obtain shift type 'early.' If, however, the same problem instance was solved with 2-SCRP, the objective function value would decrease to 3 since we would now select shift type 'day' rather than an early shift. For scenario ω_1, the shift starting in period 5 and ending in period 8 would lead to an undercoverage of 2 while for scenario ω_2, the shift starting in period 6 and ending in period 9 would lead to an undercoverage of 1 for a total of 3.

Table 4.8: Shift types and their corresponding shifts

Shift type	Shifts (Starting to ending period)
Early shift	$\{3-6, 4-7\}$
Day shift	$\{5-8, 6-9\}$
Late shift	$\{9-12, 10-13\}$

Table 4.9: Two random demand scenarios

	Time periods										
Realizations	3	4	5	6	7	8	9	10	11	12	13
$\widetilde{\mathbf{D}}(\omega_1)$	1	1	1		1						
$\widetilde{\mathbf{D}}(\omega_2)$						1	1		1		

For the deterministic case considered in part (ii) with only one scenario, the y-variables in M-SCRP and 2-SCRP are equivalent to the x-variables in R-SCRP. Also, for 2-SCRP and R-SCRP the forward rotation condition holds for the wrap-around transition from week $w = W$ to $w = 1$, but not for M-SCRP. There are no constraints that require the starting time of the first shift in week $w = 1$ to be no later than the starting time of the last shift of week $w = W$ if they are back to back. □

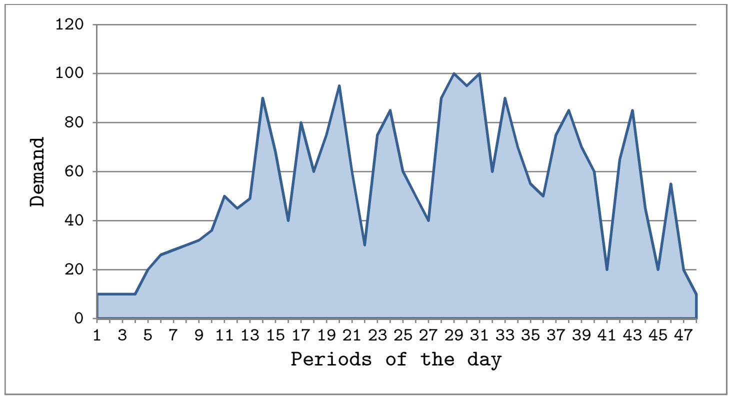
When applying model M-SCRP to traditional cyclic rostering, the problem becomes equivalent to setting $|\mathcal{P}| = |\mathcal{S}|$ and $|\mathcal{S}(p)| = 1$ for all $p \in \mathcal{P}$ such that that each shift type corresponds to a single shift $s \in \mathcal{S}$. Consequently, the first stage decisions for variables x already determine the decisions for variables y in the consecutive stages. As a result, the same set of shifts is selected in each of the stages with the forward rotation between the last day of the week and the first day of the next week already being accounted for in the first stage. Therefore, we can conclude

Theorem 7. *When used for traditional cyclic rostering, M-SCRP and R-SCRP yield the same optimal values regardless of the number of scenarios, that is,* $z_R^* = z_M^*$ *for* $|\Omega| \geq 1$.

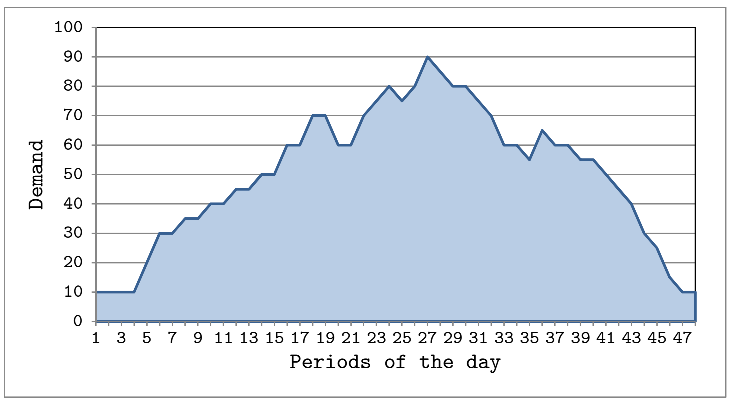
4.6 Distribution of demand

The input data for the computational study was provided by a European airport ground handling company. Each incoming and outgoing flight generates a set of jobs $\mathcal{J}$ that need to be completed within a predetermined amount of time. Each job is defined by its starting time, duration, and personnel requirements, and when taken together, form the basis of the demand curve. For scheduling purposes, we divide the planning horizon into 30-minute periods.

One of the main characteristics of the demand curve for ground handling services is it's sinusoidal structure with multiple peaks and periods of relatively little activity. Over the day, the ratio between peak and non-peak demand can be as large as three-to-one. We consider two different set of jobs; the first derived from airport ground handling activity and the second randomly generated from the first to reflect smoother transitions between periods and hence more stability. The typical one-day structure of both curves is shown in Figure 4.1.



(a) Variable demand curve



(b) Stable demand curve

Figure 4.1: Demand curves used in study

In our study, we consider different sample spaces Ω, some of which are continuous and hence infinite. To generate scenarios, we follow a two-step approach. We start with a representative set of jobs that define the variable and stable demand curves. We then define 7 different perturbations that incorporate uncertainty in ways that are commonly experienced throughout the service industry. Each perturbation together with the set of jobs leads to a different set of scenarios $\Omega_i, i \in \{1, ..., 7\}$.

To construct instances that are both representative and solvable, we generate a finite number of realizations from Ω_i; that is, we sample a finite number of times. The use of sampling serves two purposes. First, we want to create a sample set of realizations $\hat{\Omega}_i^- \subseteq \Omega_i$ with $|\hat{\Omega}_i^-| \leq 10$ to be used as input to 2-SCRP and R-SCRP given in § 4.5.

Second, we want to generate a much larger set of realizations, call it $\hat{\Omega}_i$, to evaluate the rosters produced by 2-SCRP and R-SCRP. The evaluation of flexible cyclic rosters can be done with 2-SCRP by fixing the $x_{p,g,d}$ variables at the values of the corresponding rosters.

Computationally, we are limited to solving instances of 2-SCRP and R-SCRP with up to 10 scenarios; however, a much larger number of scenarios can be evaluated for a particular roster since each evaluation is done independently of the others. In particular, the roster generation step considers all $\omega \in \hat{\Omega}_i^-$ simultaneously while the evaluation step considers each $\omega \in \hat{\Omega}_i$ separately. In the experimental study, each $\hat{\Omega}_i$ holds a large number of scenarios which represent a complete enumeration of Ω_i or are designed to provide a robust approximation of Ω_i.

For each set Ω_i, we assume that its elements follow a uniform distribution, implying that each scenario is equally likely. Figures 4.2 – 4.5 depict examples of demand curve realizations derived from the four different perturbations discussed below.

Pertubation 1 - Delay of all jobs Ω_1 and Ω_2 correspond to the set of scenarios associated with a uniform delay of all jobs. For this case, the entire demand curve is shifted by an amount determined by a perturbation denoted by del. The two sets differ from one another with respect to the possible bandwidth of the delays. Specifically, $del \in \{-60, .., 0, .., 60\}$ and $del \in \{-30, .., 0, .., 30\}$ for Ω_1 and Ω_2, respectively, and there is one scenario for each delay.

High case (Ω_1):

$\hat{\Omega}_1^-$ 5 scenarios: $del \in \{-60, -30, 0, 30, 60\}$

$\hat{\Omega}_1$ all 121 scenarios in Ω_1

Low case (Ω_2):

$\hat{\Omega}_2^-$ 3 scenarios: $del \in \{-30, 0, 30\}$

$\hat{\Omega}_2$ all 61 scenarios in Ω_2

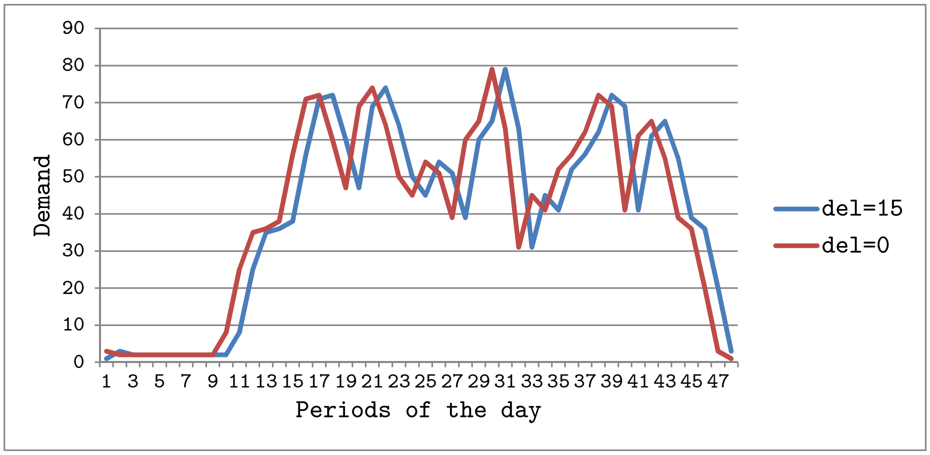


Figure 4.2: Example of scenario set $\hat{\Omega}_1$ and $\hat{\Omega}_2$

Pertubation 2 - Intensified peak periods Scenarios Ω_3 and Ω_4 represent the case were demand increases during peak periods. This requires the introduction of an additional set of jobs $\mathcal{J}^+$ to account for the increase. The number of jobs that are added per scenario is given by $j \in \{0, ..., |\mathcal{J}^+|\}$. There is a scenario for each combination of j jobs that are added from $\mathcal{J}^+$ to the set of jobs, $\mathcal{J}$.

High case (Ω_3)**:**

$\mathcal{J}^+$ 30 jobs starting in periods 11 – 14, 28 – 30, and 42 – 44, with a duration of 30 or 60 minutes, and a personnel requirement of 5 or 10 workers

$\hat{\Omega}_3^-$ 3 scenarios: $n \in \{0, 15, |\mathcal{J}^+|\}$

$\hat{\Omega}_3$ 150 scenarios sampled from Ω_3

Low case (Ω_4)**:**

$\mathcal{J}^+$ 20 jobs starting in periods 11 –14 and 28 – 30, with a duration of 30 – 60 minutes and a personnel requirement of 2 or 5 workers

$\hat{\Omega}_4^-$ 3 scenarios: $n \in \{0, 10, |\mathcal{J}^+|\}$

$\hat{\Omega}_4$ 150 scenarios sampled from Ω_4

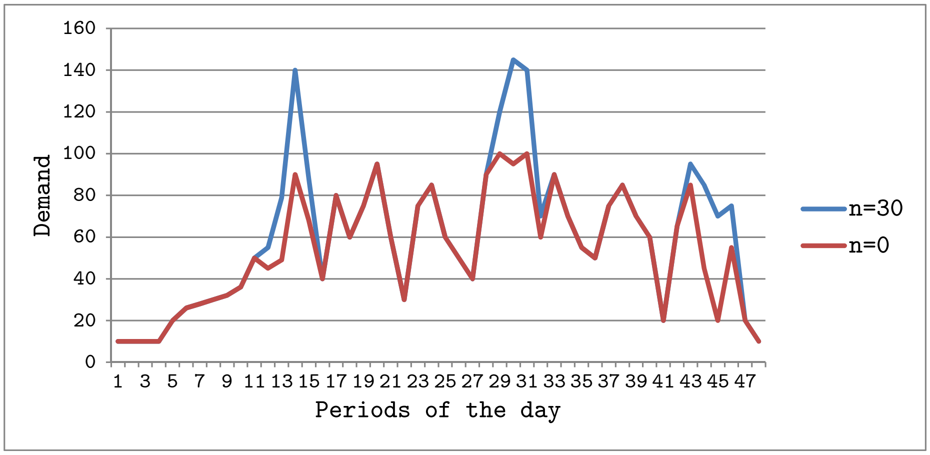


Figure 4.3: Example of scenario set $\hat{\Omega}_3$

Pertubation 3 - Adding and removing jobs Ω_5 and Ω_6 denote a set of scenarios in which jobs are added and removed uniformly over the planning horizon. The number of jobs removed from $\mathcal{J}$ is given by $\lfloor p^- \cdot |\mathcal{J}| \rfloor$, while the number of jobs added is given by $\lfloor p^+ \cdot |\mathcal{J}| \rfloor$ where $p^+ \in \mathcal{P}^+$ and $p^- \in \mathcal{P}^-$ are fractions. Each job consists of up to 4 sub-jobs with the duration of $dur \in \{5, ..., 30\}$ minutes and a personnel requirement of $req \in \{1, 2\}$ workers. Ω_5 and Ω_6 differ in the number of jobs added and removed.

High case (Ω_5):

$\mathcal{P}^+$ $\quad$ $\mathcal{P}^+ = \{0.0, 0.10, 0.20\}$

$\mathcal{P}^-$ $\quad$ $\mathcal{P}^- = \{0.0, 0.10, 0.20\}$

$\hat{\Omega}_5^-$ $\quad$ 10 scenarios: each scenario sampled from Ω_5 with $(p^+, p^-) \in \{(0.0, 0.0),$ $(0.05, 0.0), (0.10, 0.0), (0.20, 0.0), (0.0, 0.05), (0.0, 0.10), (0.0, 0.20),$ $(0.5, 0.5), (0.10, 0.10), (0.20, 0.20)\}$

$\hat{\Omega}_5$ $\quad$ 150 scenarios sampled from Ω_5

Low case (Ω_6):

$\hat{\Omega}_6^-$ $\quad$ 10 scenarios: each scenario sampled from each Ω_6 with $(p^+, p^-) \in \{(0.0, 0.0),$ $(0.02, 0.0), (0.04, 0.0), (0.06, 0.0), (0.0, 0.02), (0.0, 0.04), (0.0, 0.06),$ $(0.02, 0.02), (0.04, 0.04), (0.06, 0.06)\}$

$\hat{\Omega}_6$ 150 scenarios sampled from Ω_6

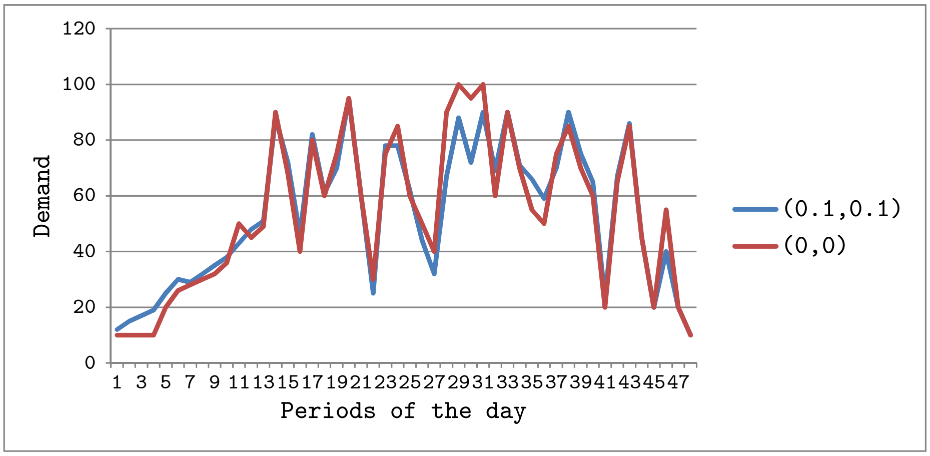


Figure 4.4: Example of scenario set $\hat{\Omega}_5$

Pertubation 4 - Delay of a subset of jobs In contrast to Ω_1, in Ω_7, only a subset of all jobs gets delayed. The number of delayed jobs is given by $\lfloor p^+ \cdot |J| \rfloor$ with $p^+ \in \{0.00, 0.05, 0.20, 0.40\}$ and the length of the delay is $del \in \{1, 2, 4\}$ periods.

$\hat{\Omega}_7^-$ 4 scenarios: each scenario sampled from Ω_7 with $(p, b) \in \{(0.00, 0), (0.05, 4), (0.20, 4), (0.4, 4)\}$

$\hat{\Omega}_7$ 450 scenarios sampled from Ω_7

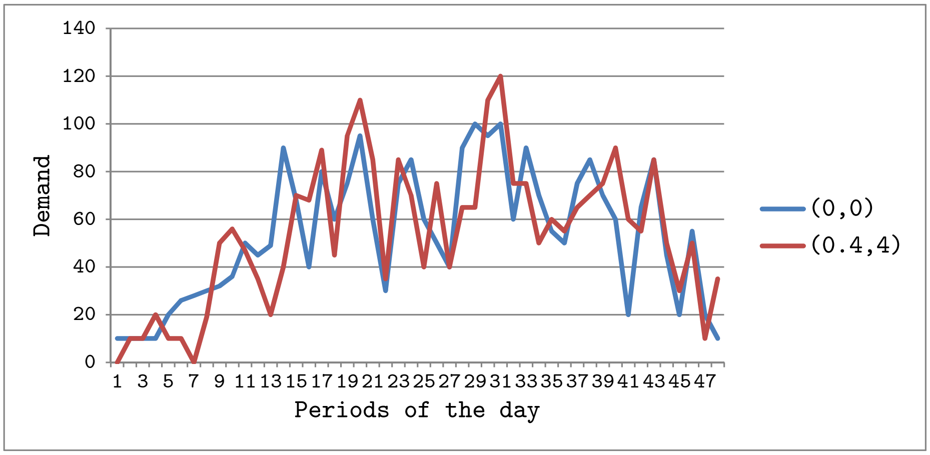


Figure 4.5: Example of scenario set $\hat{\Omega}_7$

4.7 Computational Study

The computations were performed on a Windows 7 platform with 4 GB RAM and a 2.8 GHz processor. All models were implemented in JAVA and solved with CPLEX 12.6. The scheduling procedures and models were evaluated using the demand curves depicted in Figure 4.1 but after they were modified by the sampling procedures described in Section 4.6. The variable demand curves derived from Figure 4.1(a) are identified as 'VD' while the stable demand curves derived from Figure 4.1(b) are identified as 'SD' in what follows. In the testing, we considered two shift regulations that differed by the number of shifts and shift types available. The parameter values associated with these regulations are given in Table 4.10.

Table 4.10: Shift type regulations 'R2' and 'R4'

Regulation	Set of starting periods for shift types	shift durations	$d^{\min}$	$d^{\min}$	$h^{\min}$	$h^{\max}$	$n^{\max}$
R2	$\{[6,13],[20,28]\}$	$16-21$	4	6	76	84	5
R4	$\{[6,10],[9,13],[20,25],[24,28]\}$	$16-21$					

To uniquely identify a test instance we use the notation $I(\Omega_i, pro, reg)$, with scenario set $\Omega_i \in \{\Omega_1, ..., \Omega_7\}$, demand profile $pro \in \{\text{VD}, \text{SD}\}$, and shift regulation $reg \in \{\text{R2}, \text{R4}\}$. For all instances, we assume a cycle length of 10 weeks, which implies 10 work groups. A total of 20 full-time workers are associated with each group.

4.7.1 Metrics for the value of stochastic information

As mentioned, the full model, denoted by M-SCRP, proved too difficult to solve for any of our instances. Instead, we focused on solving the two stochastic approximation models presented in Section 4.5. To assess the value of the solutions they provided, we used two of the concepts introduced by Birge and Louveaux [31] to measure the quality of stochastic programming solutions with respect to simpler and less computationally expensive ways of handling uncertainty, e.g., with models considering only the expected demand.

The first concept is the *expected value of perfect information* (EVPI) which is based on knowing at the outset all future demand realization $\widetilde{D}(\omega)$. Based on each realization ω, we would take the decision in the first stage for the x-variables that solved $\min_{x\in\mathcal{X}} h\left(x, \widetilde{D}(\omega)\right)$. The weighted sum of optimal solutions for all possible realizations yields the *wait-and-see solution* given by

$$\mathrm{WS} = W \cdot \mathbb{E}_{\widetilde{\mathbf{D}}}\left[\min_{x\in\mathcal{X}} h(x, \widetilde{\mathbf{D}})\right]$$

where $\mathcal{X}$ represents the feasible space for the first stage. The wait-and-see solution, representing the best solution that can be obtained, is compared with the solution to what is called the *resource problem.*

$$\mathrm{RP} = \min_{x\in\mathcal{X}} W \cdot \mathbb{E}_{\widetilde{\mathbf{D}}}\left\{\left[h(x, \widetilde{\mathbf{D}})\right]\right\}$$

In our case, RP is either 2-SCRP or R-SCRP, and yields the so called *here-and-now solution.* Nevertheless, computational difficulties prevented us from solving RP based on the scenario set $\hat{\Omega}_i$ so we resorted to $\hat{\Omega}_i^-$. The difference between WS and RP gives the expected value of perfect information, i.e.,

$$EVPI = RP - WS$$

Stochastic programs are usually harder to solve then their deterministic counterparts due to the additional variables and constraints so RP is usually much more difficult than WS. The second concept compares the RP solution to the solution derived from a much simpler problem based on the expected demand, $\mathbb{E}\left[\widetilde{\mathbf{D}}\right]$. Given this value, we solve the *expected value problem*

$$EV = \min_{x \in \mathcal{X}} h\left(x, \mathbb{E}\left[\widetilde{\mathbf{D}}\right]\right).$$

to get the optimal solution of the first stage variables; call it $\bar{x}$. Note that in light of Theorem 6, EV and optimal values for $\bar{x}$ are identical for all three models M-SCRP, 2-SCRP and R-SCRP. Now, we hold $\bar{x}$ constant and solve

$$EEV = \mathbb{E}_{\widetilde{\mathbf{D}}}\left[W \cdot h(\bar{x}, \widetilde{\mathbf{D}})\right]$$

The difference between this result and RP yields the *value of the stochastic solution* (VSS), i.e.,

$$VSS = EEV - RP$$

According to Birge and Louveaux [31], RP $\leq$ EEV holds for linear functions. However, since we only solve RP for a subset of possible realizations (see Section 4.6) the inequality does not necessarily hold in our case.

4.7.2 Overview of the results

For the computational comparison, flexible cyclic rosters and traditional cyclic rosters were generated using model formulations 2-SCRP and R-SCRP. Once the flexible cyclic rosters and traditional cyclic rosters are found, they are evaluated using the multiple demand realizations of the respective perturbations defined by $\hat{\Omega}_1$ to $\hat{\Omega}_7$ in the previous section, i.e. the rostering approaches were evaluated for each scenario $\omega \in \hat{\Omega}_i$ of instance $I(\Omega_i, pro, reg)$. Note that for each set of scenarios, we make the assumption that each scenario is equally likely. To evaluate a roster with respect to a set of scenarios, $\hat{\Omega}_i$, the metrics for each element ω is calculated separately. That way, we evaluate $|\hat{\Omega}_i|$ different scenarios but do not consider transitions between different weekly scenarios. Otherwise $|\hat{\Omega}_i|^W$ combinations would have to be evaluated.

For traditional cyclic rostering, the evaluation shows how good the cyclic roster covers the demand of the weekly demand curve realization of ω. R-SCRP is used to evaluate the cyclic roster by fixing all $x_{s,w,d}$ variables and calculating the resultant objective function value, z_{R}. For flexible cyclic rostering, we make use of the flexibility that the scheduling procedure offers to evalute how the demand of the weekly demand curve realization of ω is covered. Therefore model 2-SCRP is used for evaluation fixing the $x_{p,g,d}$ variables according to the flexible cyclic roster, but allowing an adaption of the $y_{g,s,d}(\omega)$ variables according to the weekly demand realization of scenario ω.

Table 4.11 gives an overview of the average results of the evaluation. In each column of Table 4.11, the average objective value, i.e. the average undercoverage, over all scenarios ω of instance $I(\Omega_i, pro, reg)$ is reported when the roster is generated with the recourse problem (RP, using all scenarios $\omega \in \hat{\Omega}_i^-$) or the expected value problem (EEV, using average demand scenario of $\hat{\Omega}_i$), scheduling is based on traditional cyclic rostering (trad) or flexible cyclic rostering (flex), and model formulation 2-SCRP (z_2) or R-SCRP (z_{R}) is used.

Note that Theorem 7 states that R-SCRP yields the same result as the multi-stage and two-stage model for traditional cyclic rostering. Therefore, for traditional cyclic rostering, we only present results for rosters generated with R-SCRP. When using R-SCRP to generate flexible cyclic rosters, each shift s associated with the $x_{s,w,d}$ variables in the solution needs to be replaced by a shift type p. This is done in a post-processing step in which a shift type p is selected such that $s \in \mathcal{S}(p)$ and $f_p \leq f_{p'} \ \forall \ p' \ \in \mathcal{P}$.

In the following we focus on the comparison of the different scheduling approaches and model approximations.

4.7.3 Flexible cyclic rostering vs. cyclic rostering

In this section, we compare the new idea of flexible cyclic rostering with traditional cyclic rostering. We focus on flexible cyclic rosters generated with R-SCRP together with a post-processor such that all traditional and flexible cyclic rosters that we compare in this part are generated based on the solution of R-SCRP. But as can be seen in Section 4.7.5, there is only little difference between the flexible cyclic rosters generated with 2-SCRP and R-SCRP.

Table 4.12 gives the results obtained by comparing cyclic rostering with flexible cyclic rostering for each of the 28 test instances, which were derived from all combinations of perturbations, variable and stable demand curves, and the two shift regulations in Table 4.10. Column 'GAP(EEV)' reports the average percentage gap between the undercoverage

Table 4.11: Average objective values of the evaluation

Instance	z_{R} (trad, EEV)	z_{R} (flex, EEV)	z_{R} (trad, RP)	z_{R} (flex, RP)	z_2 (flex, RP)
$\{\Omega_1, \mathrm{VD}, \mathrm{R4}\}$	5014	4110	4848	4101	4098
$\{\Omega_1, \mathrm{VD}, \mathrm{R2}\}$	5014	3829	4848	3820	3819
$\{\Omega_1, \mathrm{SD}, \mathrm{R4}\}$	1880	1644	1840	1489	1483
$\{\Omega_1, \mathrm{SD}, \mathrm{R2}\}$	1880	1413	1840	1386	1384
$\{\Omega_2, \mathrm{VD}, \mathrm{R4}\}$	4556	3883	4186	3773	3766
$\{\Omega_2, \mathrm{VD}, \mathrm{R2}\}$	4556	3719	4186	3698	3695
$\{\Omega_2, \mathrm{SD}, \mathrm{R4}\}$	1548	1413	1462	1327	1327
$\{\Omega_2, \mathrm{SD}, \mathrm{R2}\}$	1548	1247	1462	1242	1242
Ω_1 and Ω_2 average	17.53	15.89	4.76	2.71	0.11
$\{\Omega_3, \mathrm{VD}, \mathrm{R4}\}$	6978	6682	6904	6597	6597
$\{\Omega_3, \mathrm{VD}, \mathrm{R2}\}$	6978	6373	6904	6302	6302
$\{\Omega_3, \mathrm{SD}, \mathrm{R4}\}$	3896	3341	3699	3335	3335
$\{\Omega_3, \mathrm{SD}, \mathrm{R2}\}$	3896	3153	3699	3139	3139
$\{\Omega_4, \mathrm{VD}, \mathrm{R4}\}$	5356	5133	5344	5083	5083
$\{\Omega_4, \mathrm{VD}, \mathrm{R2}\}$	5356	4891	5344	4891	4891
$\{\Omega_4, \mathrm{SD}, \mathrm{R4}\}$	2346	2058	2248	2037	2037
$\{\Omega_4, \mathrm{SD}, \mathrm{R2}\}$	2346	1947	2248	1938	1938
Ω_3 and Ω_4 average	11.03	9.18	2.62	0.68	0.00
$\{\Omega_5, \mathrm{VD}, \mathrm{R4}\}$	4760	4446	4741	4440	-
$\{\Omega_5, \mathrm{VD}, \mathrm{R2}\}$	4760	4142	4741	4138	-
$\{\Omega_5, \mathrm{SD}, \mathrm{R4}\}$	2202	2008	2198	2009	-
$\{\Omega_5, \mathrm{SD}, \mathrm{R2}\}$	2202	1908	2198	1905	-
$\{\Omega_6, \mathrm{VD}, \mathrm{R4}\}$	4624	4382	4615	4385	-
$\{\Omega_6, \mathrm{VD}, \mathrm{R2}\}$	2624	4169	4615	4166	-
$\{\Omega_6, \mathrm{SD}, \mathrm{R4}\}$					-
$\{\Omega_6, \mathrm{SD}, \mathrm{R2}\}$					-
Ω_5 and Ω_5 average	8.68	8.57	0.073	0.030	-
$\{\Omega_7, \mathrm{VD}, \mathrm{R4}\}$	5326	4898	5320	4969	4968
$\{\Omega_7, \mathrm{VD}, \mathrm{R2}\}$	5326	4597	5320	4596	4596
$\{\Omega_7, \mathrm{SD}, \mathrm{R4}\}$	3082	2624	3102	2726	2726
$\{\Omega_7, \mathrm{SD}, \mathrm{R2}\}$	3082	2513	3102	2532	2532 8
Ω_7 average	13.75	12.38	-0.29	-1.52	0.0078

of flexible cyclic rostering and cyclic rostering over all scenarios ω of instance $I(\Omega_i, pro, reg)$ when the rosters are generated with the expected value of the random variables, i.e., the result from the EV solution. For example, for all instances i with pertubation Ω_1, the expected value is calculated based on all scenarios in $\hat{\Omega}_1$ while each $\omega \in \hat{\Omega}_i$ is considered independently for evaluation purposes.

Column 'GAP(RP)' gives the average percentage gap between the undercoverage of flexible cyclic rostering and cyclic rostering over all scenarios of instance $I(\Omega_i, pro, reg)$ when generating the rosters based on the scenarios included in $\hat{\Omega}_i^-$. This effectively compares the solution quality associated with the recourse problem. Again, each $\omega \in \hat{\Omega}_i$ is considered independently for roster evaluation purposes.

The results indicate that for all test instances, flexible cyclic rostering significantly outperforms cyclic rostering in both the expected value problem and the recourse problem. The lowest gap between both procedure is found for instance $\{\Omega_6, \mathrm{SD}, \mathrm{R4}\}$ with 3.90% while the average gap over all instances is more than 12%. The difference can also be seen in terms of absolute numbers. For examples, with test instances $I(\Omega_1, VD, R2)$, flexible cyclic rostering reduces average undercoverage by $5014 - 3829 = 1185$ periods which is equal to a reduction of 592.5 hours of undercoverage per week. Assuming a maximum number of 42 working hours per week at least an additional 14 workers are needed to cover the difference. The gap between the two procedures decreases when the number of eligible shifts remains the same but the number of shift types increases. This can be seen when comparing the results for each instance $I(\Omega_i, pro, R2)$ with the results for instance $I(\Omega_i, pro, R4)$. In all cases, for EEV as well as for RP, the gap decreases. As flexibility is lost, the results converge. In the extreme case each shift corresponds to one shift type and both procedures are identical. The design of shift types, their number and composition is always a trade-off between the week-to-week consistency in each worker's schedule and the employer's flexibility to react to uncertainty.

The gap between the two scheduling procedures also decreases when increasing the stochastic information for the roster generation as can be seen in the next section.

4.7.4 Value of the stochastic information

The diversity of the scenarios increases from Ω_1 to Ω_7 in terms of the fluctuations of the corresponding demand curve realizations (see Figures 4.2 through 4.5). When fluctuations in the demand profiles increase, the benefit of stochastic optimization decreases since we can only solve the models with 10 demand realizations. This can be seen in columns 'GAP(trad,VSS)' and 'GAP(flex,VSS)' of Table 4.13 which show the gap in % between the solution of the expected value problem and the recourse problem for cyclic and flexible cyclic rostering, respectively. Note that the column entries are equal to VSS/RP (= (EEV − RP)/RP). We can further see that for flexible cyclic rostering the gap between the stochastic and expected value solutions becomes larger when shifts are split among an increased number of shift types. The average gap between the stochastic solution and the expected value solution is 0.33% for all instances $I(\Omega_i, pro, R2)$ with regulations $R2$ in contrast to 1.19% for all instances $I(\Omega_i, pro, R4)$ regulations $R4$. This is not surprising

Table 4.12: Comparison of traditional cyclic and flexible cyclic rostering (all data entries are in terms of %)

Instance	GAP(EEV)	GAP(RP)
$\{\Omega_1, \mathrm{VD}, \mathrm{R4}\}$	18.02	16.83
$\{\Omega_1, \mathrm{VD}, \mathrm{R2}\}$	23.63	21.07
$\{\Omega_1, \mathrm{SD}, \mathrm{R4}\}$	12.54	19.04
$\{\Omega_1, \mathrm{SD}, \mathrm{R2}\}$	24.84	24.96
$\{\Omega_2, \mathrm{VD}, \mathrm{R4}\}$	14.76	9.65
$\{\Omega_2, \mathrm{VD}, \mathrm{R2}\}$	18.37	11.50
$\{\Omega_2, \mathrm{SD}, \mathrm{R4}\}$	8.71	9.08
$\{\Omega_2, \mathrm{SD}, \mathrm{R2}\}$	19.39	14.95
Ω_1 and Ω_2 average	17.53	15.89
$\{\Omega_3, \mathrm{VD}, \mathrm{R4}\}$	4.24	4.41
$\{\Omega_3, \mathrm{VD}, \mathrm{R2}\}$	8.67	9.00
$\{\Omega_3, \mathrm{SD}, \mathrm{R4}\}$	14.22	9.80
$\{\Omega_3, \mathrm{SD}, \mathrm{R2}\}$	19.05	14.39
$\{\Omega_4, \mathrm{VD}, \mathrm{R4}\}$	4.15	4.87
$\{\Omega_4, \mathrm{VD}, \mathrm{R2}\}$	8.67	7.50
$\{\Omega_4, \mathrm{SD}, \mathrm{R4}\}$	12.26	9.40
$\{\Omega_4, \mathrm{SD}, \mathrm{R2}\}$	16.99	14.45
Ω_3 and Ω_4 average	11.03	9.18
$\{\Omega_5, \mathrm{VD}, \mathrm{R4}\}$	6.58	7.19
$\{\Omega_5, \mathrm{VD}, \mathrm{R2}\}$	12.97	12.64
$\{\Omega_5, \mathrm{SD}, \mathrm{R4}\}$	8.79	8.64
$\{\Omega_5, \mathrm{SD}, \mathrm{R2}\}$	13.32	13.32
$\{\Omega_6, \mathrm{VD}, \mathrm{R4}\}$	5.23	4.62
$\{\Omega_6, \mathrm{VD}, \mathrm{R2}\}$	9.83	9.74
$\{\Omega_6, \mathrm{SD}, \mathrm{R4}\}$	3.90	4.00
$\{\Omega_6, \mathrm{SD}, \mathrm{R2}\}$	8.79	8.40
Ω_5 and Ω_6 average	8.68	8.57
$\{\Omega_7, \mathrm{VD}, \mathrm{R4}\}$	8.03	5.83
$\{\Omega_7, \mathrm{VD}, \mathrm{R2}\}$	13.67	13.61
$\{\Omega_7, \mathrm{SD}, \mathrm{R4}\}$	14.84	11.67
$\{\Omega_7, \mathrm{SD}, \mathrm{R2}\}$	18.46	18.40
Ω_7 average	13.75	12.38

since the procedure is less able to adjust to demand fluctuations with fewer shift types and therefore has to rely more on stochastic information given for the roster generating. We can conclude that the benefits increase as the options for adjusting the schedule decrease once the shift types are specified.

The gap between traditional and flexible cyclic rostering decreases when solving the recourse problem to generate rosters instead of the expected value problem as shown in Table 4.12. When generating rosters based on the expected values, flexible cyclic rostering can still adjust the schedule to better cover the demand of the evaluation scenarios. Cyclic rosters, though, are restricted to their initial shift schedule and so cannot be adjusted to match the perturbations in demand that accompany multiple realizations. This is also reflected in Table 4.13 where the avergage benefit of using the stochastic information is higher for all instances with pertubations Ω_1 and Ω_2, Ω_3 and Ω_4, Ω_5 and Ω_6, and Ω_7, respectively, for traditional cyclic rostering than for flexible cyclic rostering.

Column 'GAP(EVPI)' reports the gap in % between the solution of the recourse problem and the wait-and-see solution for flexible cyclic rostering. The column's values are equal to EVPI/WS (= (RP – WS)/WS). For all test instances associated with scenario sets $\Omega_1 - \Omega_4$ and Ω_7, the value of RP is based on the solution provided by 2-SCRP while we had to use the solution from R-SCRP for all test instances associated with scenario sets Ω_5 and Ω_6 since we could not generate the rosters with 2-SCRP within our imposed time limit of 18,000 seconds. EVPI is calculated based on the scenarios that define $\hat{\Omega}_i$ in line with the assumption made for 2-SCRP that we have the same scenario in every week of the planning horizon. Therefore, the wait-and-see solution is similar to the solution one would get by solving a traditional tour scheduling problem. For 78.6% of the test instances, GAP(EVPI) is less than 5%. Only for the 6 test instances with scenario set Ω_1 and with scenario set Ω_7 along with stable demand profiles, are the gaps between 5.68% and 13.42%.

4.7.5 2-SCRP vs. R-SCRP

Recall that according to Theorem 7, we can use R-SCRP to generate traditional cyclic rosters for the expected value problem as well as for obtaining the stochastic here-and-now solution with multiple demand realizations without any loss in solution quality. For flexible cyclic rostering, Theorem 6 states that R-SCRP provides an upper bound on 2-SCRP. Therefore, we focus on evaluating the quality of solutions found by R-SCRP as an approximation for 2-SCRP next.

Column 'GAP(2-SCRP)' of Table 4.14 lists the average percentage gap between the undercoverage of the flexible cyclic rosters generated with 2-SCRP and R-SCRP over all evaluation scenarios given in §4.6. The results indicate that R-SCRP provides a tight

Table 4.13: Value of the stochastic information for traditional cyclic and flexible cyclic rostering (all data entries are in terms of %)

Instance	GAP(trad, VSS)	GAP(flex, VSS)	GAP(EVPI)
$\{\Omega_1, \mathrm{VD}, \mathrm{R4}\}$	3.31	0.21	13.42
$\{\Omega_1, \mathrm{VD}, \mathrm{R2}\}$	3.31	0.23	13.07
$\{\Omega_1, \mathrm{SD}, \mathrm{R4}\}$	2.08	9.42	9.71
$\{\Omega_1, \mathrm{SD}, \mathrm{R2}\}$	2.08	1.90	9.36
$\{\Omega_2, \mathrm{VD}, \mathrm{R4}\}$	8.10	2.84	1.51
$\{\Omega_2, \mathrm{VD}, \mathrm{R2}\}$	8.10	0.55	1.40
$\{\Omega_2, \mathrm{SD}, \mathrm{R4}\}$	5.53	6.07	1.54
$\{\Omega_2, \mathrm{SD}, \mathrm{R2}\}$	5.53	0.43	1.40
Ω_1 and Ω_2 average	4.76	2.71	6.43
$\{\Omega_3, \mathrm{VD}, \mathrm{R4}\}$	1.06	1.27	0.63
$\{\Omega_3, \mathrm{VD}, \mathrm{R2}\}$	1.06	1.11	0.63
$\{\Omega_3, \mathrm{SD}, \mathrm{R4}\}$	5.04	0.19	0.35
$\{\Omega_3, \mathrm{SD}, \mathrm{R2}\}$	5.04	0.45	0.35
$\{\Omega_4, \mathrm{VD}, \mathrm{R4}\}$	0.22	0.97	0.35
$\{\Omega_4, \mathrm{VD}, \mathrm{R2}\}$	0.22	0.012	0.35
$\{\Omega_4, \mathrm{SD}, \mathrm{R4}\}$	4.14	1.01	1.72
$\{\Omega_4, \mathrm{SD}, \mathrm{R2}\}$	4.14	0.46	1.72
Ω_3 and Ω_4 average	2.62	0.68	0.76
$\{\Omega_5, \mathrm{VD}, \mathrm{R4}\}$	0.38	0.15	2.47
$\{\Omega_5, \mathrm{VD}, \mathrm{R2}\}$	0.38	0.11	2.47
$\{\Omega_5, \mathrm{SD}, \mathrm{R4}\}$	0.16	-0.054	3.10
$\{\Omega_5, \mathrm{SD}, \mathrm{R2}\}$	0.16	0.17	3.10
$\{\Omega_6, \mathrm{VD}, \mathrm{R4}\}$	0.18	-0.07	1.00
$\{\Omega_6, \mathrm{VD}, \mathrm{R2}\}$	0.18	0.078	1.00
$\{\Omega_6, \mathrm{SD}, \mathrm{R4}\}$	-0.43	-0.0028	0.93
$\{\Omega_6, \mathrm{SD}, \mathrm{R2}\}$	-0.43	-0.14	0.93
Ω_5 and Ω_6 average	0.073	0.030	1.88
$\{\Omega_7, \mathrm{VD}, \mathrm{R4}\}$	0.10	-1.46	0.0039
$\{\Omega_7, \mathrm{VD}, \mathrm{R2}\}$	0.10	0.032	0.0035
$\{\Omega_7, \mathrm{SD}, \mathrm{R4}\}$	-0.67	-3.9	5.68
$\{\Omega_7, \mathrm{SD}, \mathrm{R2}\}$	-0.67	-0.77	5.68
Ω_7 average	-0.29	-1.52	2.86

upper bound on 2-SCRP since all gaps are less or equal than 0.36% with most being zero.

This is especially valuable in light of the runtimes for 2-SCRP, which are considerably higher than those of R-SCRP. The size and computation times for R-SCRP and 2-SCRP are presented in Table 4.15. The number of binary variables for R-SCRP is given in column '#bin' and is the same for all test instances whether EEV or RP is being solved since the shift decisions are the same for each scenario. Further, the number of integer variables, the number of constraints, and the runtime in seconds for R-SCRP are given in columns 'intvar()','constr()', and 'time()', respectively, with 'EEV' and 'RP' indicating whether R-SCRP was used to solve EEV or RP, respectively. Note that the number of integer variables and constraints linearly increases with the number of scenarios (compare e.g. $\{\Omega_1, \mathrm{VD}, \mathrm{R4}\}$, 5 scenarios with $\{\Omega_2, \mathrm{VD}, \mathrm{R4}\}$, 3 scenarios). The number of binary variables, the number of integer variables, the number of constraints, and the runtime in seconds for 2-SCRP are shown in columns 'bin(2-RP)', 'intvar(2-RP)', 'constr(2-RP)', and 'time(2-RP)', respectively. The number of binary variables as well as the number of constraints increases for 2-SCRP with an increase in the number of scenarios. This follows because different shift assignment decisions can be made for different scenarios in the same evaluation set while the number of integer variables $u(\omega)$, which tracks the undercoverage in each period, remains constant for R-SCRP. While we could compute cyclic rosters with R-SCRP in less than 200 seconds for all instances, it took between 4,493 and 9,334 seconds with 2-SCRP for those instances with scenario sets $\hat{\Omega}_1^- - \hat{\Omega}_4^-$ and $\hat{\Omega}_7^-$. Depending on the operational conditions, this might still be acceptable for tactical planning. For all test instances associated with scenario sets Ω_5 and Ω_6, which correspond to the perturbations of adding and removing jobs uniformly over the planning horizon, we were not able to solve 2-SCRP within our imposed time limit of 18,000 seconds. Note that the number of integer variables does not increase from R-SCRP to 2-SCRP, whereas the number of constraints and the number of binary variables linearly increases with $|\hat{\Omega}_i^-|$.

4.8 Conclusion

Cyclic rostering is frequently employed by service organizations facing repetitive demand patterns over the day and week but has traditionally been limited to selecting fixed shift assignments for each cycle. Flexible cyclic rostering offers a more robust approach in which shift types are selected for each cycle and then adjusted within some bounds to accommodate daily demand variations. To model this extension, we have introduced a multi-stage stochastic program and a two-stage relaxation that can be used to generate flexible cyclic rosters. These models generalize the traditional formulations and, along with our case study, represent one of the first attempts to to deal with stochastic issues in a real setting.

Table 4.14: Comparison of R-SCRP and 2-SCRP for flexible cyclic rostering and the expected value of the perfect information (all data entries are in terms of %)

Instance	GAP(2-SCRP)
$\{\Omega_1, \mathrm{VD}, \mathrm{R4}\}$	0.083
$\{\Omega_1, \mathrm{VD}, \mathrm{R2}\}$	0.015
$\{\Omega_1, \mathrm{SD}, \mathrm{R4}\}$	0.36
$\{\Omega_1, \mathrm{SD}, \mathrm{R2}\}$	0.11
$\{\Omega_2, \mathrm{VD}, \mathrm{R4}\}$	0.17
$\{\Omega_2, \mathrm{VD}, \mathrm{R2}\}$	0.09
$\{\Omega_2, \mathrm{SD}, \mathrm{R4}\}$	0.010
$\{\Omega_2, \mathrm{SD}, \mathrm{R2}\}$	0.0106
Ω_1 and Ω_2 average	0.11
$\{\Omega_3, \mathrm{VD}, \mathrm{R4}\}$	0.00
$\{\Omega_3, \mathrm{VD}, \mathrm{R2}\}$	0.00
$\{\Omega_3, \mathrm{SD}, \mathrm{R4}\}$	0.00
$\{\Omega_3, \mathrm{SD}, \mathrm{R2}\}$	0.00
$\{\Omega_4, \mathrm{VD}, \mathrm{R4}\}$	0.00
$\{\Omega_4, \mathrm{VD}, \mathrm{R2}\}$	0.00
$\{\Omega_4, \mathrm{SD}, \mathrm{R4}\}$	0.00
$\{\Omega_4, \mathrm{SD}, \mathrm{R2}\}$	0.00
Ω_3 and Ω_4 average	0.00
$\{\Omega_5, \mathrm{VD}, \mathrm{R4}\}$	-
$\{\Omega_5, \mathrm{VD}, \mathrm{R2}\}$	-
$\{\Omega_5, \mathrm{SD}, \mathrm{R4}\}$	-
$\{\Omega_5, \mathrm{SD}, \mathrm{R2}\}$	-
$\{\Omega_6, \mathrm{VD}, \mathrm{R4}\}$	-
$\{\Omega_6, \mathrm{VD}, \mathrm{R2}\}$	-
$\{\Omega_6, \mathrm{SD}, \mathrm{R4}\}$	-
$\{\Omega_6, \mathrm{SD}, \mathrm{R2}\}$	-
Ω_5 and Ω_6 average	-
$\{\Omega_7, \mathrm{VD}, \mathrm{R4}\}$	0.031
$\{\Omega_7, \mathrm{VD}, \mathrm{R2}\}$	0.00
$\{\Omega_7, \mathrm{SD}, \mathrm{R4}\}$	0.00
$\{\Omega_7, \mathrm{SD}, \mathrm{R2}\}$	0.00
Ω_7 average	0.0078

For our 28 different test instance sets, the computations showed that the two-stage relaxation could be used to solve all problems with up to five scenarios in less than 3 hours, while the remaining ten scenario cases could not be solved within 3 hours. Because runtimes can be critical in some environments, we presented a further relaxation of the two-stage model excluding the possibility to adjust shifts in the different scenarios which was able to solve all instances in less than 200 seconds with less than a 0.4% gap with respect to the solutions provided by the two-stage model. Comparing flexible cyclic rostering with traditional cyclic rostering, we found that the flexiblity added by the new scheduling procedure reduced undercoverage by more then 10%. The value of the stochastic solution was investigated for both rostering approaches. The results indicated that the value of the stochastic information decreases when fluctuations in the demand profile increase as we cannot solve the problem with more than a dozen of scenarios. Further, as the options decrease for adjusting to demand fluctuation, the value of the stochastic information increases. The implication is that exploiting stochastic information is especially valuable for traditional cyclic rostering which does not allow for reactive decisions. The stochastic information was notably high for demand pertubations where all jobs are delayed with an average improvement of the undercoverage of 4.75% for cyclic rostering and 2.71% for flexible cyclic rostering while we had even slightly negative impact for the highly fluctuating demand pertubation where only a subset of jobs gets delayed. For 78.6% of the test instances, the expected value of the perfect information was less than 5% which is especially interesting since it is also a tight approximation to the gap that is obtained when comparing flexible cyclic rostering to tour scheduling where each weekly roster is generated from scratch.

Table 4.15: Model size and runtime comparisons for R-SCRP and 2-SCRP

Instance	#bin	#bin(2-RP)	#int(EEV)	#int(RP)	#int(2-RP)	#constr(EEV)	#constr(RP)	#constr(2-RP)	time(EEV)	time(RP)	time(2-RP)
$\{\Omega_1, \mathrm{VD}, \mathrm{R4}\}$	7,150	35,990	336	1,680	1,680	614	1,958	3,023	19.64	27.38	15,495
$\{\Omega_1, \mathrm{VD}, \mathrm{R2}\}$		35,850							20.69	25.22	9,334
$\{\Omega_1, \mathrm{SD}, \mathrm{R4}\}$		35,990							26.70	29.30	17,627
$\{\Omega_1, \mathrm{SD}, \mathrm{R2}\}$		35,850							28.37	31.66	10,241
$\{\Omega_2, \mathrm{VD}, \mathrm{R4}\}$	7,150	35,990	336	1,008	1,008	614	1,286	1,913	18.83	21.02	8,588
$\{\Omega_2, \mathrm{VD}, \mathrm{R2}\}$		35,850							18.72	21.29	9,492
$\{\Omega_2, \mathrm{SD}, \mathrm{R4}\}$		35,990							31.63	33.46	13,912
$\{\Omega_2, \mathrm{SD}, \mathrm{R2}\}$		35,850							30.55	33.72	10,644
Ω_1 and Ω_2 average	7,150	35,920	336	1,344	1,344	614	1,622	2,468	24.52	27.88	11,916
$\{\Omega_3, \mathrm{VD}, \mathrm{R4}\}$	7,150	14,570	336	672	672	614	950	1,358	19.90	20.26	6,351
$\{\Omega_3, \mathrm{VD}, \mathrm{R2}\}$		14,430							19.88	18.14	4,716
$\{\Omega_3, \mathrm{SD}, \mathrm{R4}\}$		14,570							21.54	34.31	8,440
$\{\Omega_3, \mathrm{SD}, \mathrm{R2}\}$		14,430							23.95	41.31	7,981
$\{\Omega_4, \mathrm{VD}, \mathrm{R4}\}$	7,150	14,570	336	672	672	614	950	1,358	28.45	25.10	6,950
$\{\Omega_4, \mathrm{VD}, \mathrm{R2}\}$		14,430							22.50	20.40	6,186
$\{\Omega_4, \mathrm{SD}, \mathrm{R4}\}$		14,570							29.71	65.87	8,323
$\{\Omega_4, \mathrm{SD}, \mathrm{R2}\}$		14,430							30.92	59.74	7,084
Ω_3 and Ω_4 average	7,150	14,500	336	672	672	614	950	1,358	24.60	35.63	7,003
$\{\Omega_5, \mathrm{VD}, \mathrm{R4}\}$	7,150	71,690	336	3,360	3,360	614	3,638	5,798	25.12	61.65	-
$\{\Omega_5, \mathrm{VD}, \mathrm{R2}\}$		71,550							23.43	56.40	-
$\{\Omega_5, \mathrm{SD}, \mathrm{R4}\}$		71,690							24.80	59.87	-
$\{\Omega_5, \mathrm{SD}, \mathrm{R2}\}$		71,550							51.66	103.01	-
$\{\Omega_6, \mathrm{VD}, \mathrm{R4}\}$	7,150	71,690	336	3,360	3,360	614	3,638	5,798	85.47	170.10	-
$\{\Omega_6, \mathrm{VD}, \mathrm{R2}\}$		71,550							73.08	113.64	-
$\{\Omega_6, \mathrm{SD}, \mathrm{R4}\}$		71,690							87.65	134.75	-
$\{\Omega_6, \mathrm{SD}, \mathrm{R2}\}$		71,550							90.62	129.31	-
Ω_5 and Ω_6 average	7,150	71,620	336	3,360	3,360	614	3,638	5,798	57.74	103.60	-
$\{\Omega_7, \mathrm{VD}, \mathrm{R4}\}$	7,150	28,850	336	1,344	1,344	614	1,622	2,468	18.24	18.01	5,020
$\{\Omega_7, \mathrm{VD}, \mathrm{R2}\}$		28,710							18.74	18.40	4,493
$\{\Omega_7, \mathrm{SD}, \mathrm{R4}\}$		28,850							19.28	19.04	4,857
$\{\Omega_7, \mathrm{SD}, \mathrm{R2}\}$		28,710							18.22	19.90	4,901
Ω_7 average	7,150	28,780	336	1,344	1,344	614	1,622	2,468	18,62	18,84	4,817

Chapter 5

The flexible break assignment problem for large tour scheduling problems with an application to airport ground handlers

5.1 Introduction

In most personnel scheduling problems workforce demand is a function of the number of required tasks per period, and, depending on the application, can have extremely high variance. To optimally cover demand, a tour scheduling problem needs to be solved to determine the days off (days-off scheduling) and shift assignments (shift scheduling) for each worker on each day of the planning horizon. Once the days off are fixed, there are two immediate ways to adapt a worker's tour when demand is highly variable: adjusting the start time and length of a shift, and assigning breaks to shifts. Although much has been written about the first option, there is a noticeable absence in the literature when it comes to analyzing different break regulations, especially in the tour scheduling environment. Moreover, there is little practical evidence that planners have exploited the benefits offered by such regulations when it comes to better matching supply with demand.

One of the objectives of this chapter is to show how break assignments can be used to improve the scheduling of airport ground handlers. When a flight arrives at an airport, baggage and cargo must be transferred to the terminal, handicapped passengers must be transported to either their next flight or the baggage area, and the aircraft must be prepared for its next trip. These functions fall under the heading of *ground handling* and, depending on the number of arriving passengers and the distance flown, may require a dozen or more service workers to perform (Ashford et al. [11]). Personnel costs are the largest expense faced by the service provider and must be managed carefully in today's competitive environment. Ground handling as well as many other personnel scheduling problems require the explicit assignment of shifts and days off to individual employees rather than to a generic workforce. This means that information on individual skills, availability, and overtime balances must be taken into account (e.g., see Love and Hoey [87]). In ground handling, the difference between peak and non-peak demand can be as large as three to one over the day. Meeting the demand without an excess of under- or over-coverage is an extremely challenging task. To the best of our knowledge, this chapter is the first to analyze the full range of break options in tour scheduling.

The approach we take involves decomposing the full problem into a tour scheduling problem and a break assignment problem. By separating the two, we create a framework in which different break models can be computationally evaluated without controlling for the basic tour scheduling parameters. The majority of the work in the area of breaks has focused on model development and the design of computational techniques for reducing problem size and runtimes in the context of shift scheduling. The primary example centers on the the implicit modeling of breaks in shift scheduling (see Aykin [13], Bechtold and Jacobs [29], Rekik et al. [106]). Our decomposition approach enables us to fully investigate the various implicit formulations in the wider context of tour scheduling and a hierarchical workforce.

As part of the developments, we introduce a classification scheme for the break assignment problem (BAP) and present a complexity analysis of all major variations. Based on the different properties of the accompanying models, the flexibility and operational value of each is evaluated using data from a large European ground handling company. For applications where breaks are assigned dynamically over the day, e.g., for ground handling and mail processing operations, the decomposition approach provides an upper bound on costs. In such environments, the break assignment model can be used in a rolling horizon manner to support reactive planning.

In order to generate tours, we rely on templates that define a set of permissible shifts for each day of the planning horizon. The templates also include the option for a day off and therefore allow us to consider the availability of each employee each day of the week. To show the impact of shift starting time and shift duration flexibility and its interaction with break flexibility, we investigate two different scenarios. In the case of high shift flexibility, when early and late shifts are considered, we get more than $1,400$ possible shifts for the different combinations of starting times and working hours alone. This is in contrast to less than 30 possible shifts for the low flexibility case. As expected, we show that model size is strongly correlated with our ability to solve realistic instances in a timely manner.

The main contributions of this chapter are as follows.

- A classification scheme and a complexity analysis of the break assignment problem for all practical variations
- The presentation of an efficient implicit formulation for assigning breaks to shifts in a manner that accounts for hierarchical skill levels
- A heuristic decomposition procedure for scheduling a hierarchical workforce that can accommodate all permissible break placements in a shift (the heuristic supports reactive planning of breaks in response to unforeseen changes in hourly demand)
- An analysis showing the impact of shift and break flexibility on solution quality and runtime

The chapter is organized as follows. In § 5.2 we review the literature on ground handling operations and break assignment models. This is followed in § 5.3 with a detailed description of the full problem, including an example and definitions related to templates and tours. An integrated mixed-integer programming (MIP) formulation of the tour scheduling problem based on weekly templates is presented in Appendix A.5. In § 5.4, we introduce a classification scheme for the different BAPs and analyze the complexity of each. The decomposition approach is highlighted in § 5.5 along with an efficient implicit model for break assignments for a hierarchical workforce. This is followed by a computational study in § 5.6 where the various break assignment models and the decomposition procedure are empirically evaluated. We close with some insights and observations on the proposed methodology.

5.2 Literature review

Personnel scheduling is the process of constructing work timetables for the staff of an organization that best meet the demand for its goods and services (Ernst et al. [54]). An overview of the subject can be found in Tien and Kamiyama [124], Ernst et al. [54], and Van den Bergh et al. [129]. In our review, we focus on tour scheduling, break assignments, implicit break formulations, and airport ground staff scheduling.

The first integer programming formulation for personnel scheduling problems was given by Dantzig [49] who introduced a set covering model in which each decision variable corresponded to a feasible tour. For all but the simplest of applications, however, problem size grows exponentially and becomes unmanageable when more than a handful of shift types and break options are considered. As a consequence, a large number of alternative formulations have been developed to reduce model size. Foremost are implicit formulations that either solve the problem by aggregating decisions or by modeling the rules for building shifts, breaks, or tours as constraints. Stolletz [118] introduced a reduced set covering-based formulation for the tour scheduling problem for check-in counter personnel in which all feasible shifts are enumerated explicitly as columns. A decision variable for each worker, each day of the planning horizon, and each shift is employed while the rules for building tours are modeled as constraints.

The increase of problem complexity can be seen in Brunner and Stolletz [35] where meal breaks must be assigned to shifts within a specified time window. While the set-covering formulation proposed by Stolletz [118] could be solved to optimality with off-the-shelf software (CPLEX) for small instances, as the shift definition expanded and breaks were considered it become impossible to find optimal solutions. To tackle the larger instances, branch and price was applied. However, computational results still showed gaps of up to 7% with 65 workers and a 1,000 node limit on the size of the search tree. In those instances, a 15 day scheduling problem with 34 periods per day and an upper bound of 435 shifts per day (when break information is not included in the shift) is considered. In contrast, we consider a 7-day scheduling problem with 96 periods per day, up to 638 shifts per day and significantly higher complexity in break regulations. In § 5.4, we give an overview of the various break regulations that have appeared in the in literature and classify relevant work.

Implicit break formulations for shift scheduling are presented by Bechtold and Jacobs

[29], Aykin [13] and Rekik et al. [106]. Bechtold and Jacobs [29] reduced the size of the set-covering-type model by excluding the break information from the shift matrix. That is, breaks are treated in the aggregate and not explicitly assigned to shifts by the MIP. A set of constraints is introduced to guarantee that feasible breaks are available for the shifts chosen. In a post-processing step, breaks are assigned to shifts. A different approach is proposed by Aykin [13] where a variable is introduced for each combination of shift and eligible break starting time. Again, beak information is removed from the shift matrix. In both formulations, predetermined break time windows and a fixed break duration are assumed.

The modeling of sub-breaks for shift scheduling was introduced by Rekik et al. [106] who used the term *fractionable breaks*. A fractionable break has an overall duration that can be split into one or more sub-breaks with the restriction that the number of sub-breaks is within a predefined range. In their work, sub-break placement is constrained by work stretch durations between consecutive breaks. They present two formulations for shift scheduling based on extensions of the models of Bechtold and Jacobs [29] and Aykin [13]. In our experience, however, these formulations cannot be directly adapted to the much more complex tour scheduling problem.

While many organizations take advantage of flexible break patterns within a shift, to the best of our knowledge, there has been no research on their use in the extended problem of tour scheduling. In part, this chapter is aimed at filling this void. By separating the tour scheduling problem from the BAP it is possible to use implicit formulations for dealing with a range of skill requirements. Building on the work of Rekik et al. [106] and others, we present two implicit formulations for the BAP in which hierarchical skills are taken into account. The one based on Bechtold and Jacobs [29], however, needs a post-processor to assign breaks to workers and so cannot be applied in all environments (e.g., bus driver scheduling; see Rekik et al. [106]). To circumvent this limitation we introduce and evaluate an alternative implicit formulation that models the break rules as constraints.

Contributions to tour scheduling for ground handlers and other airport personnel can be found in Dowling et al. [53], Chu [41], Ásgeirsson [10], and Lusby et al. [88]. Dowling et al. [53] proposed a system for the monthly rostering of 500 workers at a large airport. They used simulated annealing to minimize idle time under highly fluctuating demand. A novel component of their approach was an external rule engine to check for feasibility after each neighborhood search. Chu [41] optimized daily schedules for baggage handlers at the Hong Kong International Airport using a goal programming-based algorithm. Taking a behavior-based approach, Ásgeirsson [10] designed a heuristic that emulates the logic of personnel managers when creating schedules for an airport ground service company with 53 employees. He begins with a partial schedule derived from employee requests and then builds a full schedule that satisfies the prevailing labor laws and company policies. Solution quality was judged by how closely the days-on and days-off assignments matched employee requests. Finally, Lusby et al. [88] tackled a ground crew rostering problem with a column generation-based heuristic for a European airline. Their goal was to create long-term rosters based on 6 days-on, 3 days-off work patterns.

5.3 Problem description

Our tour scheduling model is built around individual *templates* that specify the feasible set of shifts and days off that may be assigned to a worker each day of the planning horizon. We define a *shift* as a given number of continuous periods with a fixed start time and duration. A *shift type* is a set of shifts whose start times and durations fall within predefined ranges (e.g., early, mid-day or late). The set of all feasible shifts is denoted by $\mathcal{S}$.

To illustrate the concept of a template, consider a one-week planning horizon and three workers, each having a different template as shown in Table 5.1. The notation E/L means that a worker can be assigned either an early or a late shift; L/X means that a worker can be assigned either a late shift or a day off, depending on demand. In the latter case, some workers may be on while others may be off. If a template offers all shift types on all days, i.e., E/L/X in the case of early and late shifts, the problem is equivalent to a tour scheduling problem without templates.

At the European service company that provided our ground handler application, the templates for each worker for the upcoming week are derived from a cyclic template defined for a flight season (e.g., summer or winter), as well as each worker's individual vacation plan, and are assumed to be given. The actual tasks for which the worker is responsible are assigned on the day of operation by a dispatcher. For the current tours, we want to assure that sufficient personnel are available to cover the demand. We now formalize the

Table 5.1: Example of three different weekly templates for three workers with the following shift types: E=early, L=late, X=day off

Worker	**Days of the week**						
	Mo	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**
1		E/X	E/X	E	E	E/L	L
2	L	L/X				E/X	E
3	E	E	E/L	L	L	L/X	

problem of creating a weekly tour for each member of the workforce based on his or her template.

An overview of sets and indices used in the formulation is given in Table 2. The workforce is denoted by $\mathcal{W}$ such that each worker $w \in \mathcal{W}$ is assigned a template $p \in \mathcal{P}$, where p defines the eligible shift types (e.g., early, off, late, combination) for each day $d \in \mathcal{D}$ in the week. The set $\mathcal{M}(d,p) \subseteq \mathcal{M}$ identifies the shift types that are allowed in template p on day d. A day in template p can either be a mandatory working day, an optional working day or a day off.

The set of feasible shifts for day d is given by $\mathcal{S}(d) \subseteq \mathcal{S}$. On mandatory working days, an employee must be uniquely assigned to a shift $s \in \mathcal{S}$, whereas on optional working days he may or may not be assigned to a shift, depending on the demand. Each day is divided into a set of periods $\mathcal{T} = \{1,\ldots,T\}$ of equal length (typically, 15 minutes). A shift $s \in \mathcal{S}$ in weekly template p has to satisfy the following rules.

S1. Starting and ending time windows: Each shift type m must start within the time band $\mathcal{T}^{\text{shS}}(m) \subset \mathcal{T}$ and must end within the time band $\mathcal{T}^{\text{shE}}$.

S2. Overall working time: The overall shift length including breaks must fall within the interval $\left[\Delta^{\text{minS}}, \Delta^{\text{maxS}}\right]$.

Multiple breaks are permitted per shift but the actual number must fall between some lower and upper bound. This allows for maximum flexibility when making the break assignments. However, the length of each sub-break must also fall between some given bounds as must the time between them and their total duration. Adopting the nomen-

Table 5.2: Sets and indices used in the formulation

Sets	Definition
$\mathcal{W}$	Workers, $w \in \mathcal{W}$
$\mathcal{W}(q)$	Workers with skill q
$\mathcal{S}$	Shifts, $s \in \mathcal{S}$
$\mathcal{S}(d)$	Shifts on day d
$\mathcal{S}(q,t)$	Shifts covering demand for skill q in period t
$\mathcal{B}$	Break patterns
$\mathcal{B}(s)$	Break patterns for shift s
$\mathcal{Q}$	Skill levels, $q \in \mathcal{Q}$
$\mathcal{P}$	Weekly templates, $p \in \mathcal{P}$
$\mathcal{M}$	Shift types, $m \in \mathcal{M}$
$\mathcal{M}(p)$	Eligable shift types in template p
$\mathcal{M}(d,p)$	Eligable shift types in template p on day d
$\mathcal{T}$	Number of planning periods per day, $t \in \mathcal{T}$
$\mathcal{T}^{\text{shS}}(m)$	Shift starting times for the shift type m
$\mathcal{T}^{\text{shE}}$	Shift ending times
$\mathcal{T}^{\text{brS}}(r)$	Break starting times for the r-th sub-break
$\mathcal{T}^{\text{brE}}$	Break ending times of all sub-breaks
$\mathcal{T}^{\text{brE}}(r)$	Break ending times of the r-th sub-break
$\mathcal{D}$	Planning days, $d \in \mathcal{D}$
$\mathcal{D}(p)$	Planning days in template p

clature introduced by Rekik et al. [106] for fractionable breaks, we impose the following rules:

B1. Number of breaks: There are $B^{\min} \leq r \leq B^{\max}$ sub-breaks in a shift.

B2. Workstretch between first break and shift start: The first break must start within $\left[\Delta^{\text{minBF}}, \Delta^{\text{maxBF}}\right]$ periods after the shift starts.

B3. Single break duration: Break $r \in [B^{\min}, B^{\max}]$ has a minimum and maximum duration of Δ_r^{minBD} and Δ_r^{maxBD} periods.

B4. Overall break duration: The minimum and maximum overall break duration must be within the interval $\left[\Delta^{\text{minBD}}, \Delta^{\text{maxBD}}\right]$.

B5. Workstretch duration: The workstretch duration after the r-th break has to be within
$\left[\Delta_r^{\text{minBW}}, \Delta_r^{\text{maxBW}}\right]$ periods.

B6. Workstretch between last break and shift end: The last break must end within $\left[\Delta^{\text{minBL}}, \Delta^{\text{maxBL}}\right]$ periods before the shift ends.

Given the parameters in B1 to B6, the range of sub-break r's possible starting and ending times can be restricted to the subsets $\mathcal{T}^{\text{brS}}(r)$ and $\mathcal{T}^{\text{brE}}(r)$, respectively. In the following, we define a break pattern $b \in \mathcal{B}$ as a feasible set of sub-breaks with starting times and durations such that (B1) to (B6) are satisfied. The subset of feasible break patterns within a shift s that satisfies B5 and B6 is denoted by $\mathcal{B}(s)$. As we shall see in § 5.4 these break regulations are the the most general and flexible investigated to date. High flexibility however, is accompanied by high complexity.

An additional attribute of the workforce is the level of skill that each individual brings to the job, whether he is a baggage handler, driver, ramp agent or other designation. Skill levels are ordered hierarchically, meaning that a worker at the higher end of the spectrum can fill in for one at the lower end when needed. The procedure in which workers are assigned tasks below their skill level should they have idle time in their schedule is sometimes called downgrading (Bard [22]). Let $Q = \{q_1, \ldots, q_n\}$ be the ordered set of skills, where the relation $q_k \preceq q_l$ means that that a task requiring skill level $q_k \in \mathcal{Q}$ can be performed by any worker having a skill level of at least q_l for $l \geq k$. The vector $K = (K_{q,t,d})_{q \in \mathcal{Q}, t \in \mathcal{T}, d \in \mathcal{D}}$ is used to represent the demand of a task, where each element $K_{q,t,d}$ gives the required demand for skill level q in time period t on day d. Each shift covering demand for skill level q at time t is contained in the set $\mathcal{S}(q,t)$.

At the beginning of the planning horizon, each worker w has an accumulated bank of either overtime ($O_w \geq 0$) or undertime ($O_w < 0$) that must be kept within a target bandwidth $[O^-, O^+]$. Given a weekly template p_w for worker w, when generating a tour, the following rules must be taken into account.

T1. Tour bandwidth: The length of a tour must be within the tour bandwidth $[O^-, O^+]$.

T2. Forward rotation: The start time of shift type m must be nondecreasing from one day to the next during the week.

T3. Bandwidth: For each shift type, there is a maximum allowed bandwidth of Δ^{bw} for shift starting times.

T4. Starting times: The number of maximum allowed shift starting times per week for shift type m is limited to K^{shift}.

The main objective of the problem is to minimize labor costs while covering as much demand as possible by fixing shift start and end times, days off, and breaks. Each worker incurs a cost of c^{work} per period which is assumed to be less or equal than the unit cost of undercoverage c^{uc}, i.e., $c^{\text{work}} \leq c^{\text{uc}}$. In determining labor costs, the time that a worker is not on duty, such as during a break, is not factored into the calculations. Table 5.3 lists the cost parameters.

Table 5.3: Cost parameters

Parameter	Definition
c^{work}	unit cost for a worker when on duty
c^{uc}	unit cost of undercoverage

It is well known that tour scheduling problems are NP-hard (see Lau [82]), which is one explanation for the lengthy runtime we experienced when trying to solve our first MIP model (see Appendix §A.5) of the ground handler staffing problem (GHSP) with CPLEX. In practice, breaks are often set at the beginning of the working day and not beforehand to better accommodate unforeseen changes in the demand profile. Therefore, to allow for this level of flexibility, as well as to reduce the difficulty of the problem, we have chosen to separate the BAP from the tour scheduling problem (see §5.5). In the next section, we categorize the full range of BAPs and undertake a complexity analysis of each.

5.4 Break Assignment Problem

The GHSP contains very general and flexible break regulations. Since the need for break assignments is common in many shift and tour scheduling problems, we classify and analyze the complexity of various BAPs found in the literature. A computational study of different versions of the problem is presented in §5.6.

Classification Scheme

We use three sets of parameters to classify the BAP:

- *Single(**S**), multiple(**M**) or fractionable(**F**) breaks.* Break assignment formulations can be classified as single, multiple or fractionable break models. In single break BAPs only one break per shift may be assigned whereas a fixed number $m \in \mathbb{Z}_+$ of breaks are required per shift for multiple BAPs. In contrast, fractionable break models have a lower and upper bound for the number of breaks that can be assigned to a shift.

- *Fixed(**X**) or variable(**V**) break length.* For BAPs with single or multiple breaks, each sub-break can either have a fixed length or a lower and upper bound on its length. Considering fractionable breaks, the length of each is inherently variable but there is an aggregate break length given which is then split into sub-breaks for the shift.

- *Time windows(**T**) or workstretch(**W**) durations.* Time windows define the periods of a shift in which a break can start. In contrast, the workstretch duration defines a lower and upper bound on the number of consecutive periods of work in a shift. Workstretch durations implicitly define time windows, with the difference being that the size and position of each are interdependent and therefore not static.

In what follows, we classify the various versions of the BAP using the 3-tuple $[S, M, F|X, V|T, W]$. Figure 5.1 gives a hierarchical overview of the complexity of the more interesting models based on the findings given below. Those models associated with the parameter combinations not shown are for the single break case where the workstretch restrictions lead to simple time windows and are therefore effectively the same problem class.

Figure 5.2 depicts four realizations of break types using the 3-field notation and their possible positions in a shift. In panels (a)-(c), the gray rectangles define the time windows in which a break can take place due to work regulations. In panel (d), there are no predefined break windows as workstretch durations limit the solution space. However, there is a minimum workstretch of 2 periods before the first break and between the first and second break, and a minimum workstretch duration of 1 period after the last break. In all examples, the overall break duration is equal to 4 periods. The overall break length in (b) is split equally between the first and second break, while in (c), the lengths of the first and second breaks are variable.

$$
\begin{tikzcd}
\{S|X|T\} \arrow[r] \arrow[d] & \{M|X|T\} \arrow[r] \arrow[d] & \{M|X|W\} \arrow[d] \\
\{S|V|T\} \arrow[r] & \{M|V|T\} \arrow[r] \arrow[d] & \{M|V|W\} \arrow[ddl] \\
 & \{F|V|T\} \arrow[d] & \\
 & \{F|V|W\} &
\end{tikzcd}
$$

Figure 5.1: Hierarchical overview of the complexity of various BAPs

The variety of possible break assignments provides increased flexibility but at a cost. This will be confirmed in a theoretical sense by studying the computational complexity of the different break regulations. Table 5.4 provides an overview of break regulations in the staff scheduling literature. As one can see, there has been little research on flexible break regulations despite their positive impact on solution quality (see §5.6).

Table 5.4: Break regulations in the staff scheduling literature

Break regulation	References
$\{S\|X\|T\}$	Alfares [5], Bard [23], Bard and Wan [24], Bard et al. [25] Bechtold and Jacobs [29], Brusco and Jacobs [37, 38] Rong [112], Topaloglu and Ozkarahan [125]
$\{M\|X\|T\}$	Avramidis et al. [12], Rekik et al. [105], Thompson and Pullman [123] Aykin [13], Ni and Abeledo [99]
$\{M\|V\|T\}$	Mirrazavi and Beringer [92]
$\{M\|X\|W\}$	Kuo et al. [79]
$\{F\|V\|T\}$	Beer et al. [30]
$\{F\|V\|W\}$	Rekik et al. [106]

Complexity Analysis

In the analysis, we assume that a set of shifts $\mathcal{S}' \subset \mathcal{S}$ is given such that each $s \in \mathcal{S}'$ is associated with a unique worker $w \in \mathcal{W}$. The objective of the BAP is to assign breaks to each shift s without violating the break constraints, i.e., B1 to B6, so that the costs associated with shift assignments and the uncovered demand are minimized. As mentioned in § 5.3, each worker incurs a cost of c^{work} per period which is assumed to be less than or equal to the unit cost of undercoverage c^{uc}, that is, $c^{\text{work}} \leq c^{\text{uc}}$; off duty periods during a shift do not incur any cost. In the analysis, we work with a set partitioning formulation for the BAP in which all feasible break patterns $\mathcal{B}$ are enumerated. The set of feasible break patterns for shift s is denoted by $\mathcal{B}(s)$. The following parameters and variables are used in the developments. **Parameters**

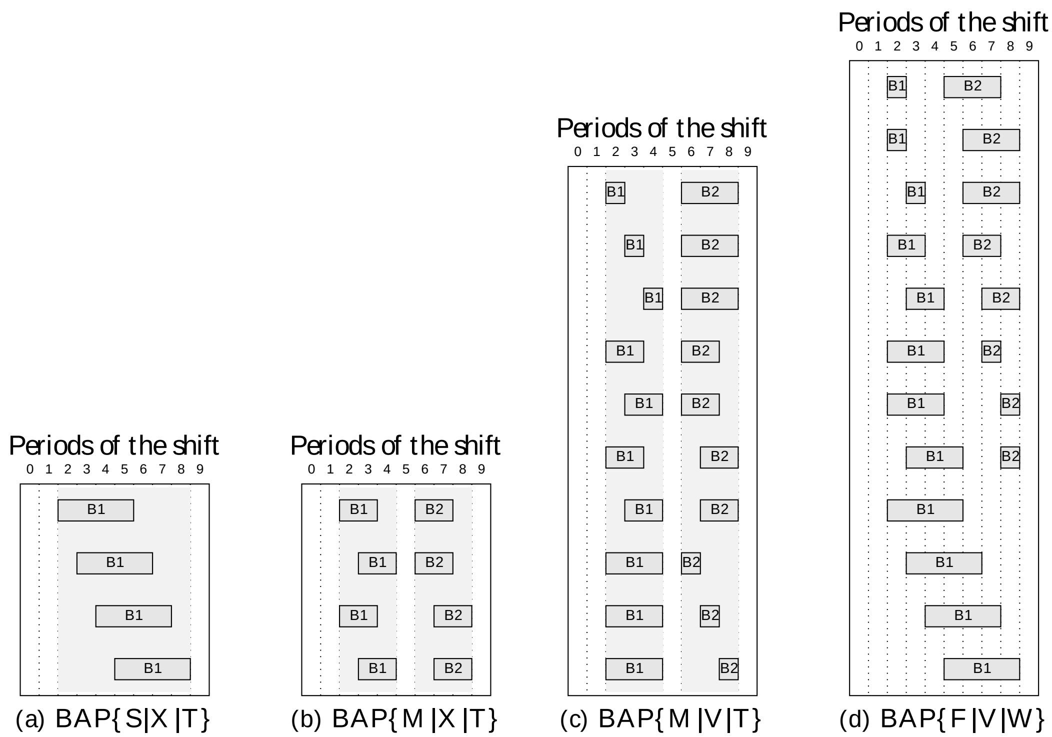


Figure 5.2: Break types and their possible assignments in a shift

$c^{\text{work}}_{s,b}$ = cost for a worker assigned shift s with break pattern b

$a_{s,b,q,t}$ = 1, if break pattern b in shift s covers demand for skill q in period t, 0 otherwise

$D_{q,t}$ = amount of over-coverage of demand for skill q in period t (can be negative)

Variables

$z_{s,b} = 1$, if break pattern b is assigned to shift s, 0 otherwise

$y^-_{q,t}$ = amount of uncovered demand for skill q in period t after break assignments

$y^+_{q,t}$ = amount of excess demand for skill q in period t after break assignments

Set partitioning formulation for the BAP

$$\text{Minimize} \sum_{s \in \mathcal{S}} \sum_{b \in \mathcal{B}(s)} c^{\text{work}}_{s,b} \cdot z_{s,b} + \sum_{q \in \mathcal{Q}} \sum_{t \in \mathcal{T}} c^{\text{uc}} \cdot y^-_{q,t} \tag{110}$$

subject to

$$\sum_{s \in \mathcal{S}'} \sum_{b \in \mathcal{B}(s)} a_{s,b,q,t} \cdot z_{s,b} + y^+_{q,t} - y^-_{q,t} = \max\{D_{q,t}, 0\} \quad \forall\, q \in \mathcal{Q}, t \in \mathcal{T} \tag{111}$$

$$\sum_{b \in \mathcal{B}(s)} z_{s,b} = 1 \quad \forall\, s \in \mathcal{S}' \tag{112}$$

$$z_{s,b} \in \{0,1\} \quad \forall\, s \in \mathcal{S}', b \in \mathcal{B}(s) \tag{113}$$

$$y^+_{q,t}, y^-_{q,t} \geq 0 \quad \forall\, q \in \mathcal{Q}, t \in \mathcal{T} \tag{114}$$

The objective function (110) minimizes the sum of labor costs and the cost of uncovered demand over all periods. When a subset of periods has an initial shortage, i.e., $D_{q,t} < 0$, it might be desirable to assign a larger positive weight than c^{uc} to the corresponding variables to discourage additional undercoverage. Constraints (111) account for demand reduction due to the break assignments to the shifts. The first term on the left-hand side reduces the demand in period t for each shift that includes a break in period t. The next two terms are complementary variables. The first represents the amount of remaining coverage in period t after the breaks have been assigned and will be nonnegative if a shortage exists. The second variable represents the undercoverage in period t after the breaks have been assigned and is minimized in (110). The right-hand side of (111) is taken to be nonnegative to avoid penalizing existing undercoverage.

The convexity constraints (112) enforce the requirement that each shift is assigned exactly one break pattern, and constraints (113) and (114) define the variables. In an optimal solution at most y^+_t or y^-_t will be positive but never both because we are trying to minimize the latter. Also, it can be seen that these variables will always be integral in an optimal solution so there is no need to impose such a restriction.

Proposition 1. *When the break length of BAP$\{S/X/T\}$ is one period the A-matrix associated with model (110)–(114) is totally unimodular (TU) implying that the associated problem is polynomially solvable.*

Proof. First, let us assume that we have only one skill level for the workforce and the more general case in which breaks can be more than one period. Consider shift s and its corresponding set of feasible break patterns $\mathcal{B}(s)$. By examining constraint (111) we see that each column possesses the consecutive 1's property. Let $A^1 \equiv (a^1_{i,j})$ be the matrix associated with the first shift in (111). Such a matrix is TU (Nemhauser and Wolsey [96]). As an example, consider a shift of 10 periods that requires a break of either 3 or 4 periods starting in either periods 3, 4 or 5, as shown in Figure 5.3.

$$
(a^1_{i,j}) = \begin{pmatrix} 0&0&0&0&0&0\\ 0&0&0&0&0&0\\ 1&0&0&1&0&0\\ 1&1&0&1&1&0\\ 1&1&1&1&1&1\\ 0&1&1&1&1&1\\ 0&0&1&0&1&1\\ 0&0&0&0&0&1\\ 0&0&0&0&0&0\\ 0&0&0&0&0&0 \end{pmatrix} \cdots \begin{pmatrix} 1&-1&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&1&-1&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&-1&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&\ldots&&&&&&&\\ 0&0&0&0&0&0&&&&&&&&\\ 0&0&0&0&0&0&0&1&-1&0&0&0&0&0\\ 0&0&0&0&0&0&&&&\ldots&&&&\\ 0&0&0&0&0&0&&&&&&&&\\ 0&0&0&0&0&0&0&0&0&0&1&-1&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&1&-1 \end{pmatrix}
$$

$$
\begin{pmatrix} 1&1&1&1&1&1 \end{pmatrix} \cdots \begin{pmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&0&0 \end{pmatrix}
$$

Figure 5.3: Constraint matrix for BAP$\{S|X|T\}$ with one skill level.

The row of 1's below A^1 in this figure corresponds to constraint (112); the second matrix, call it Y, on the right corresponds to the y_t^+ or y_t^- variables. Although constraints (112) destroy the consecutive 1's property in general, when the break length for each shift is one period the portion of the A-matrix in (111) associated with each shift has a single 1 in each column. This entry appears in a row in which a break is permissible (this is not the case shown in Figure 5.3). The remaining row entries in the column are 0. Thus each column in the full matrix has at most two elements that are either 1 or -1, and, in light of the special structure of constraints (112), we see that it satisfies the sufficiency condition for TU. That is, there exits a partition of the set M of rows into two sets M_1 and M_2 such each column j containing two nonzero elements satisfies $\sum_{i \in M_1} a_{i,j} - \sum_{i \in M_2} a_{i,j} = 0$. Note that the Y-matrix can be written as $Y = [I, -I]$, where I is the $|T| \times |T|$ identity matrix so each column contains only a single nonzero element. $\square$

Corollary 1. *When the length of each sub-break of BAP$\{M|X|T\}$ is one period the A-matrix associated with model (110)–(114) is totally unimodular (TU) implying that the associated problem is polynomially solvable.*

Proof. Again, let A^1 be the matrix associated with the first shift in (111) and let A^2 be the matrix associated with constraints (112), also for the first shift. As an example, Figure 5.4 depicts a portion of the constraint matrix for a shift with 7 periods that requires two breaks of one period each. For feasibility, the first must start in either period 2 or 3 and the second in either period 5 or 6. As indicated in the figure, when the length of each sub-break is equal to one period, each column of A^1 contains a single entry of 1 in the row where the sub-break exists with the remaining entries being 0. In a solution, one of the first two columns and one of the second two columns must be selected.

Now, when adding time windows to regulate sub-break starting times, there is no interdependency between the possible starting times of different sub-breaks, i.e., the starting time of the i-th sub-break is independent of the starting time of the j-th sub-break for all $i, j \in [1, B^{\max}]$. Each column of A^2 also contains a single entry of 1 or -1 with the other entries being 0 due to the fact that each column of A^1 and A^2 corresponds to a single sub-break. Therefore, as in the proof of Proposition 1, the sufficiency condition for TU still holds in the case where the sub-break length is one period for each shift.

$$(a^1_{i,j}) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \cdots \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & & & & \\ 0 & 0 & 0 & 0 & & & & & \\ 0 & 0 & 0 & 0 & & & & & \\ 0 & 0 & 0 & 0 & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix}$$

$$(a^2_{i,j}) = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \cdots \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Figure 5.4: Constraint matrix for BAP$\{M|X|T\}$ with two break windows and one skill level.

□

Corollary 2. *For a multi-skilled, hierarchical workforce, when the sub-break length of BAP$\{M|X|T\}$ is one period the A-matrix associated with model (110)–(114) is totally unimodular (TU), implying that the associated problem is polynomially solvable.*

Proof. Following the proof of Corollary 1, let A^1 be the matrix associated with the first shift in (111) and let A^2 be the matrix associated with constraints (112), also for the first shift. As an example, Figure 5.5 shows a portion of the constraint matrix for a shift with 5 periods that requires one break. The break must start in either period 2, 3 or 4 of the shift and lasts for one period. We assume that there are two skill levels q_1 and q_2 and that the worker associated with the first shift is scheduled to work a job requiring skill q_1 in the second period and q_2 in the third and fourth periods, respectively. In periods 1 and 5, he is either working a job requiring q_1 or q_2 but, due to the time window, can not have a break. The submatrix above the horizontal line illustrates constraints (111) for q_1 and the lower submatrix the corresponding constraints for q_2. When the length of the sub-break is equal to one period, each column of A^1 contains a single entry of 1 in the row where the sub-break exists (columns 1 and 2) with the remaining entries being 0. Column 3 shows that the consecutive 1's property is lost when a sub-break incorporates more than a single skill.

Similar to the single skill case, when employing time windows to regulate sub-break starting times, there is no interdependency between the possible starting times of different sub-breaks. Therefore, each column of A^2 also contains a single entry of 1 or -1 with the other entries being 0 due to the fact that each column of A^1 and A^2 corresponds to a single sub-break. Now again, similar to the proof of Proposition 1, the sufficiency condition for TU still holds in the case where the sub-break length is one period for each shift.

$$
(a^1_{i,j}) = \left(\begin{array}{ccc}
0 & 0 & 0 \\
1 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\hline
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
0 & 0 & 0
\end{array}\right) \cdots \left(\begin{array}{ccccccccc}
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \cdots & & & & \\
0 & 0 & 0 & 0 & & & & & \\
0 & 0 & 0 & 0 & & & & & \\
\hline
0 & 0 & 0 & 0 & & & & & \\
0 & 0 & 0 & 0 & & & & & \\
0 & 0 & 0 & 0 & & & & & \\
0 & 0 & 0 & 0 & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1
\end{array}\right)
$$

$$
(a^2_{i,j}) = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \cdots \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}
$$

Figure 5.5: Constraint matrix for BAP$\{M|X|T\}$ with two skill levels.

$\square$

The implication of Proposition 1 is of particular interest when scheduling relief breaks, which are short breaks of 15 minutes or less and typically one period in length. Thompson and Pullman [123] reviewed 75 staff scheduling papers and found that 15% of them contained relief breaks. If considered separately, they can be assigned in polynomial time once the regular breaks are fixed.

When a break consists of more than one period, the sufficiency condition is lost so it is not possible to conclude that the full matrix for the set partitioning-like model (110)– (114) is TU. We now show that solving the problem in which only a single break of arbitrary length is to be assigned is not polynomially solvable.

Theorem 8. *The problem of assigning a single break to each shift to minimize the number of uncovered periods, that is, BAP$\{S/X/T\}$, is strongly NP-hard.*

Proof. We will show that an instance of the single machine scheduling problem with release times and deadlines, which Garey and Johnson [63] state is strongly NP-hard, can be polynomially transformed into an instance of BAP$\{S|X|T\}$.

INSTANCE: Set $\mathcal{J}$ of jobs, for each job $j \in \mathcal{J}$, 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 $\mathcal{J}$ that satisfies the release time constraints and meets all the deadlines?

Given an instance of the single machine scheduling problem, we equate a shift with a job, so $\mathcal{S}$ is equivalent to $\mathcal{J}$. For shift $s \in \mathcal{S}$, let r_s be the period in which the break can start (release time), e_s the last period in which the break is allowed (deadline), and p_s the number of periods of the break (processing time). Also, let the surplus of coverage in each period be 1.

Claim: There exists a feasible assignment of breaks to shifts such that there are no manpower shortages in any period if and only if there is a feasible schedule of jobs for the single machine problem.

The condition of no manpower shortage in any period means that no two breaks overlap. This is equivalent to a schedule on a single machine in which each job is processed between its release time and deadline without conflicting with any other job. Given that the above transformation is linear in the number of shifts, and that any candidate solution to the BAP can be checked for feasibility in polynomial time, the statement of the theorem follows. □

The BAP$\{M|X|T\}$ is equivalent to the break model that occurs as a subproblem in Aykin [13]. By the principle of restriction stated by Garey and Johnson [63], the BAP$\{M|X|T\}$ is also NP-hard in the strong sense since it includes the BAP$\{S|X|T\}$ as a special case; that is, when $m = 1$. Similarly, because the BAP$\{S|X|T\}$ and the BAP$\{M|X|T\}$ are special cases of the BAP$\{S|V|T\}$ and the BAP$\{M|V|T\}$, respectively, the latter two problems are also strongly NP-hard. Now, noting that the BAP$\{F|V|T\}$ is a generalization of the the BAP$\{M|V|T\}$, with BAP$\{M|V|T\}$ being the special case of equalizing the lower and upper bound for the number of sub-breaks of the BAP$\{F|V|T\}$, the relationships in Figure 5.1 follow. However, the BAP$\{M|X|T\}$ and the BAP$\{S|V|T\}$ cannot be hierarchically related to each other.

Proposition 2. *The BAP$\{F/V/W\}$ is strongly NP-hard and includes the BAP$\{F/V/T\}$ as* *a special case.*

Proof. For the BAP$\{F|V|T\}$ with n distinct time windows, assume that the overall break length ranges between Δ^{minBD} and Δ^{maxBD} periods, and that the minimum and maximum number of sub-breaks are between $B^{\min}$ and $B^{\max}$. With each time window $[t_r^s, t_r^e], r \in \{1, ..., n\}$, we associate a minimum break length of $d_r^{\min}$ and a maximum break length of $d_r^{\max}$. Accordingly, for each $i \in \{1, ..., B^{\max}\}$ the i-th sub-break can start in the (i)-th,...,$(n - B^{\min} + i)$-th time window and the length of the i-th sub-break is between $lb_i^{\text{bl}} = min_{r\in\{i,\ldots,n-B^{\min}+i\}}(d_r^{\min})$ and $ub_i^{\text{bl}} = max_{r\in\{i,\ldots,n-B^{\min}+i\}}(d_r^{\max})$ periods.

For the BAP$\{F|V|W\}$, we set the range for the overall break length as well as the minimum and maximum number of sub-breaks equal to those of the BAP$\{F|V|T\}$. Further, for each sub-break $i \in \{1, ..., B^{\max}\}$, we set the minimum and the maximum break length equal to lb_i^{bl} and ub_i^{bl}, respectively. To proof our statement, it is sufficient to set the minimum and maximum workstretch after each sub-break equal to 0 and the shift length, respectively. Based on the assumptions, BAP$\{F|V|W\}$ includes all the breaks of the BAP$\{F|V|T\}$ but not vice versa.

Now, place a cost of $M > 0$ on each sub-break in the BAP$\{F|V|W\}$ that is not feasible to BAP$\{F|V|T\}$. The procedure can be done in polynomial time as for each shift, the overall number of breaks in the BAP$\{F|V|W\}$ is bounded by $\prod_{i\in\{1,\ldots,B^{\max}\}}(ub_i^{\text{bl}} - lb_i^{\text{bl}}) \cdot \Delta^{\text{maxS}}$. Given both models, we claim that the BAP{F|V|T} has a solution with no uncovered periods if and only if the BAP$\{F|V|W\}$ has a solution with no uncovered periods and a cost of 0. First suppose that the BAP$\{F|V|T\}$ has a solution with no uncovered periods. Then the same break patterns can be used for the BAP$\{F|V|W\}$ to obtain the same solution. This proves the "if" part of the statement. Now suppose that the optimal solution to the BAP$\{F|V|W\}$ has a cost greater than 0. By implication, this solution contains sub-breaks that have a cost M and hence is not feasible to the BAP$\{F|V|T\}$. Therefore, BAP$\{F|V|T\}$ does not have a solution with uncovered periods. This proves the "only if" part of the statement.

Now observe that the BAP$\{F|V|T\}$ does not include the BAP$\{F|V|W\}$ since the break windows of two or more consecutive sub-breaks can overlap. Consider a BAP$\{F|V|W\}$ with two sub-breaks. Let Δ^{minBF} and Δ^{maxBF} be the minimum and maximum workstretch duration before the first sub-break, respectively. Further, let Δ^{durB} be the minimum duration of the first sub-break and let Δ^{minBW} be the minimum workstretch duration between the first and second sub-breaks. When $\Delta^{\text{minBF}} + \Delta^{\text{durB}} + \Delta^{\text{minBW}} < \Delta^{\text{maxFB}}$, the break windows of the first and second sub-breaks overlap. In general, in the presence of workstretch durations, the break window for a particular sub-break can only be derived after the realization of the other sub-breaks.

□

In a similar manner, the relations between BAP$\{M|X|T\}$ and BAP$\{M|X|W\}$, and between BAP$\{M|V|T\}$ and BAP$\{M|V|W\}$ can be shown. Each of these problems is strongly NP-hard.

5.5 Decomposition Procedure

In the GHSP under investigation we have to tackle two NP-hard problems: the tour scheduling problem (TShP) and the (fractionable) BAP$\{F|V|W\}$. Solving the integrated model given in Appendix A.5, however, was seen to be intractable with CPLEX (see §5.6) so we adopted a sequential procedure. First, the TShP is solved to get feasible tours for each worker. Because this problem by itself is still intractable when two or more skill levels are considered simultaneously, we use downgrading. This approach starts with the highest skill level and works its way down to the lowest, covering as much demand as possible as the computations progress (§5.5.1). In light of the TShP solution, breaks are assigned to each shift by solving the BAP$\{F|V|W\}$ (§5.5.2).

5.5.1 Tour scheduling problem

Model formulation In the TShP, the start and end times of each shift for each worker, as well as their days off, are determined for the planning horizon. When applying the downgrading procedure, the TShP is solved for all workers $w \in \mathcal{W}(q_k)$ of a given single skill level $q_k \in \mathcal{Q}$ one level at a time. Let $\mathcal{Q}(q^+, q_k)$ be the set of skills that are higher or equal to skill q^+ and lower or equal to skill q_k. In each iteration, the set of workers $\mathcal{W}(q_k)$ is scheduled such that they cover the demand for skill levels in $\mathcal{Q}(q^+, q_k)$ with the proviso that they first cover higher level demand. In presenting the model, let TShP(q^+, q_k) be the tour scheduling problem being solved for workers with skill level q_k with demand restricted to skill levels in $\mathcal{Q}(q^+, q_k)$.

Variables

$y_{m,w,d,t} = 1$, if shift of type m for worker w starts in period t on day d, 0 otherwise

$y^{\text{end}}_{w,d,t} = 1$, if shift for worker w ends in period t on day d, 0 otherwise

$z_{w,q,d,t} = 1$, if worker w is active during period t on day d at skill level q, 0 otherwise

$v_{m,w}$ = worker w's earliest starting time for shift type m

$w_{w,t} = 1$, if a shift for worker w starts in period t, 0 otherwise

$y^{+}_{q,d,t}$ = shortfall in demand for skill level q in period t on day d

TShP(q_k, q^+)

$$\text{Minimize} \quad \sum_{q \in \mathcal{Q}(q^+,k)} \sum_{d \in \mathcal{D}} \sum_{t \in \mathcal{T}} \left(\sum_{w \in \mathcal{W}(q_k)} \left(c^{\text{work}} + p_q \right) \cdot z_{w,q,d,t} + c^{\text{uc}} \cdot y^{+}_{q,d,t} \right) \tag{115}$$

subject to

Objective constraints

$$\sum_{w \in \mathcal{W}(q_k)} z_{w,q,d,t} + y^{+}_{q,d,t} \geq K_{q,d,t} \qquad \forall\, q \in \mathcal{Q}(q^+,k) \in \mathcal{D}, t \in \mathcal{T} \tag{116}$$

$$\sum_{m \in \mathcal{M}(d,p_w)} \sum_{l \in \mathcal{T}^{\text{shS}}(m): l \leq t} y_{m,w,d,l} - \sum_{l \in \mathcal{T}^{\text{shE}}: l \geq t} y^{\text{end}}_{w,d,l} = z_{w,q,d,t} \qquad \forall\, w \in \mathcal{W}(q_k), d \in \mathcal{D}(p_w), t \in \mathcal{T} \tag{117}$$

$$\sum_{q \in \mathcal{Q}(q^+,k)} z_{w,q,d,t} \leq 1 \qquad \forall\, w \in \mathcal{W}(q_k), d \in \mathcal{D}(p_w), t \in \mathcal{T} \tag{118}$$

Shift constraints

$$\mathbb{1}_{\mathcal{D}^{\text{fix}}(p_w)}(d) \leq \sum_{m \in \mathcal{M}(d,p_w)} \sum_{t \in \mathcal{T}^{\text{shS}}(m)} y_{m,w,d,t} \leq 1 \qquad \forall\, w \in \mathcal{W}(q_k), d \in \mathcal{D}(p_w) \tag{119}$$

$$0 \leq \sum_{t \in \mathcal{T}^{\text{shE}}} t \cdot y^{\text{end}}_{w,d,t} - \sum_{m \in \mathcal{M}(d,p_w)} \sum_{t \in \mathcal{T}^{\text{shS}}(m)} \left(t + \Delta^{\text{minS}} \right) \cdot y_{m,w,d,t} \leq \Delta^{\text{maxS}} \qquad \forall\, w \in \mathcal{W}(q_k), d \in \mathcal{D}(p_w) \tag{120}$$

Tour constraints

$$O^{-} - O_w \leq \sum_{d \in \mathcal{D}(p_w)} \left(\sum_{t \in \mathcal{T}^{\text{shE}}} t \cdot y^{\text{end}}_{w,d,t} - \right. \qquad \forall\, w \in \mathcal{W}(q_k) \tag{121}$$

$$\left.\sum_{m\in\mathcal{M}(d,p_w)}\sum_{t\in\mathcal{T}^{\text{shS}}(m)} t\cdot y_{m,w,d,t}\right) \le O^{+} - O_w$$

$$\sum_{t\in\mathcal{T}^{\text{shS}}(m)} t\cdot\left(y_{m,w,d,t} - y_{m,w,d',t}\right)\cdot y_{m,w,d',t} \le 0 \qquad \forall\, m\in\mathcal{M}(d,p_w),\ w\in\mathcal{W}(q_k),\ d,d'\in\mathcal{D}(p_w): d<d' \tag{122}$$

$$t\cdot y_{m,w,d,t} \ge v_{m,w}\cdot y_{m,w,d,t} \qquad \forall\, m\in\mathcal{M}(d,p_w),\ w\in\mathcal{W}(q_k), d\in\mathcal{D}(p_w),\ t\in\mathcal{T}^{\text{shS}}(m) \tag{123}$$

$$\sum_{t\in\mathcal{T}^{\text{shS}}(m)} t\cdot y_{m,w,d,t} - v_{m,w} \le \Delta^{\text{bw}} \qquad \forall\, m\in\mathcal{M}(d,p_w),\ w\in\mathcal{W}(q_k), d\in\mathcal{D}(p_w) \tag{124}$$

$$w_{w,t}\cdot M \ge \sum_{d\in\mathcal{D}(p_w)} y_{m,w,d,t} \qquad \forall\, m\in\mathcal{M}(p_w),\ w\in\mathcal{W}(q_k), t\in\mathcal{T}^{\text{shS}}(m) \tag{125}$$

$$\sum_{t\in\bigcup_{\substack{d\in\mathcal{D}(p_w),\\ m\in\mathcal{M}(d,p_w)}}\mathcal{T}^{\text{shS}}(m)} w_{w,t} \le K^{\text{shifts}} \qquad \forall\, w\in\mathcal{W}(q^{\text{work}}) \tag{126}$$

$$y_{m,w,d,t}, w_{w,t}, y^{\text{end}}_{w,d,t'}, z_{w,q,d,t'} \in \{0,1\} \qquad \forall\, w\in\mathcal{W}(q_k),\ q\in\mathcal{Q}(q^{+},k), d\in\mathcal{D}(p_w),\ m\in\mathcal{M}(d,p_w),\ t\in\mathcal{T}^{\text{shS}}(m), t'\in\mathcal{T}^{\text{shE}} \tag{127}$$

$$v_{m,w}\in\mathbb{N} \qquad \forall\, m\in\mathcal{M}, w\in\mathcal{W}(q_k) \tag{128}$$

$$y^{\text{uc}}_{q,d,t} \ge 0 \qquad \forall\, q\in\mathcal{Q}(q^{+},k), d\in\mathcal{D},\ t\in\mathcal{T} \tag{129}$$

$$z_{w,q,d,t}\in\{0,1\} \qquad \forall\, w\in\mathcal{W}(q_k),\ q\in\mathcal{Q}(q^{+},k),\ d\in\mathcal{D}, t\in\mathcal{T} \tag{130}$$

Objective function The objective function (115) minimizes the sum of labor costs and the cost of not meeting the demand. When workers with skill level q_k are being considered, demand coverage for skill levels $q_k > \ldots > q_1$ is penalized with $p_{q_k} = 0 < \ldots < p_{q_1}$ such that workers with skill level q_k are prevented from covering demand that requires lower skills before covering demand requiring skill q_k.

Objective function constraints Constraints (116) ensure that either the demand for the skill levels in $\mathcal{Q}(q^+, k)$ is covered or a shortage is identified. Variable $z_{w,q,d,t}$, which indicates whether or not worker w is on duty in period t on d at skill level q, i.e., $z_{w,q,d,t} = 1$ or 0, is set in constraints (117). Constraints (118) guarantee that worker w is assigned at most one skill level in each period.

Shift constraints (S1) Constraints (119) set the start time of the shifts in worker w's weekly template, $p_w \in \mathcal{P}$. Here, the indicator function $\mathbb{1}_{\mathcal{D}^{\text{fix}}(p_w)}(d)$ is equal to 1 if day d is an obligatory day on for worker w, and 0 otherwise. (S2) The minimum and maximum shift length inclusive of breaks is bounded in the two-sided constraints in (120).

Tour constraints (T1) Bounds are placed on the maximum allowed undertime O^- and overtime O^+ for each worker w in constraints (121). (T2) The forward rotation of shift starting times over the week is enforced by constraints (122). (T3) The earliest starting time for shift type m is determined by constraints (123) while the maximum permitted deviation between start times of each shift type during the week is restricted by constraints (124). (T4) The maximum number of shift starts in a week is bounded by constraints (125) and (126). The remaining constraints define the variables.

Sequential assignment procedure The sequential procedure for assigning shifts to workers to cover demand for all skill levels in each time period of the planning horizon is outlined in Algorithm 2. Assume that Algorithm 2 is currently at skill level q_2 in step 3, i.e., $q_k = q^+ = q_2$. If it is possible to cover the entire demand $K_{q_2,t,d}$ for skill level q_2 in all periods t on any day $d \in \mathcal{D}$ (see Step 8), Steps 9 through 11 are executed; that is, skill level q^+ is decremented by 1 such that $q^+ = q_1$, and demand $K_{q_1,d,t}$ for skill level q_1 on day d for all periods t is added to constraints (149). In objective function (115), covering demand for skill level $q < q_k$ is penalized by p_q to force the workers being scheduled to cover tasks requiring their skill level first. When TShP(q_k, q^+) is solved for the first time, we set $q_k = q^+ = q_n$ in Step 3.

The sub-procedure in Steps 5 – 13 continues until undercoverage appears for some demand $K_{q,d,t}$ with $q \in \mathcal{Q}(q^+, k)$ on any day d in some period t. Before decrementing q_k and starting the next iteration in which TShP(q_k, q^+) is solved for workers with the next lower skill level, we reduce the demand $K_{q,d,t}$ for $q \in \mathcal{Q}(q^+, k)$ in Step 14 in periods that

Algorithm 2 Sequential Downgrading MIP-heuristic

`Input:` Set of skills $\mathcal{Q}$ and demand vector K
`Output:` Set of shifts $\mathcal{S}^*(d)$ for each day $d \in \mathcal{D}$
for $q_k = q^+ = q_n$ to q_1 **do**
    next $\leftarrow$ TRUE
    while *next* **do**
        next $\leftarrow$ FALSE
        solve TShP(q_k, q^+)
        if $K_{q,d,t}$ is covered for any $d \in \mathcal{D}$ in all $t \in \mathcal{T}$ for all $q \in \mathcal{Q}(q^+, k)$ with $q \leq q_k$ **then**
            put $q^+ \leftarrow q^+ - 1$
            add demand $K_{q^+,d,t}$ on day d for all periods t to constraints (116)
            next $\leftarrow$ TRUE
        end if
    end while
    compute residual demand $K_{q,d,t}$ for all $q \in \mathcal{Q}(q^+, k)$, $d \in \mathcal{D}$ and $t \in \mathcal{T}$
end for

are covered by workers in $\mathcal{W}(q_k)$. For example, consider two time periods t_1 and t_2 on day d and two skill levels q_1 and q_2 with the following demand.

$$K_{q_1,d,t_1} = 2, K_{q_1,d,t_2} = 1$$
$$K_{q_2,d,t_1} = 1, K_{q_2,d,t_2} = 1$$

For skill level q_2, assume that two workers are scheduled in t_1 and t_2, i.e., the demand for skill level q_2 is completely covered. Hence, the next lower skill level is added, i.e., $q^+ = q_1$. When running TShP(q_k, q^+) again, the entire demand for skill level q_2 is covered, while one additional unit of demand is covered by one worker in period t_2 for skill level q_1. Finally, updating the residual demand for skill level q_1 gives $K_{q,d,t_1} = 1$ and $K_{q,d,t_1} = 0$.

5.5.2 Break assignment problem

Algorithm 2 terminates with a set of shifts $\mathcal{S}^*(d)$ for each day d, where each shift $s \in \mathcal{S}^*(d)$ uniquely corresponds to one worker $w \in \mathcal{W}$. We now assign break patterns to these shifts by solving the BAP. Because there are no restrictions on break patterns from one day to the next, the computations can be done independently for each day d.

5.5.2.1 Implicit formulations

The set partitioning formulation (110)–(114) explicitly assigns break patterns to shifts $\mathcal{S}(d)$. If we consider, for example, 400 shifts in which up to 3 sub-breaks are permitted, and each sub-break can start within a time window of 12 periods and have a duration between 2 and 3, we obtain $|\mathcal{B}| = 5,529,600$. This value defines the number of columns (and hence variables) in the set partitioning formulation. The resultant problem is much too large to solve in practice. Consequently, we have adopted an implicit modeling approach to the BAP and investigate three different formulations.

The primary aim of implicit modeling is to reduce the number of decision variables by avoiding the need to enumerate each possible combination of shift and sub-break sequence. This is achieved by modeling the rules for building breaks as constraints. The implicit formulation associated with the GHSP is given in Appendix A.7. The other two formulations are based on the shift scheduling work of Rekik et al. [106] who extended the multiple break models of Aykin [13] and Bechtold and Jacobs [29] to include fractionable breaks. However, they found that due to the large number of variables still required by Aykin's formulation, computational difficulties arose when there is a high degree of break flexibility. With this in mind, we present an implicit formulation based on Bechtold and Jacobs [29] that can accommodate significant break flexibility as well as a hierarchical workforce. Insights on the computation times of the different formulations are provided in § 5.6.

5.5.2.2 Notation and components of implicit BAP formulations

To help define our model, we introduce the terms *break profile* and *shift profile*. Each break profile $\beta \in \mathcal{BP}$ corresponds to a number of sub-breaks denoted by B_β and their respective lengths. For example, using minutes as the unit of time, $\beta_1 = 15/30/15$ and $\beta_2 = 30/15/15$ are two break profiles for a fractionable break with an overall length of 60 minutes. A sub-break $k \in \mathcal{K}$ is defined for each break profile, each position in the break profile, each possible starting time and each skill level. Notice that we need a unique sub-break for each skill level as we have to consider the skill of the worker that is taking the break. In Rekik et al. [106], a *shift profile* (or a *shift* in their terminology) $j \in \mathcal{J}$ is defined as a unique combination of start time, length, and admissible break profile. Because we are considering a hierarchical workforce, we have extended the definition of a shift profile to include a specific skill level. Some additional notation follows.

Sets

$\mathcal{J}_{(\beta,q)}$ = set of shift profiles associated with break profile β and skill q

$\mathcal{B}_{(\beta,r,q)}$ = set of sub-breaks associated with break profile β, position r and skill q

Parameters

$\rho_{t,k}$ = 1, if break k covers period t, 0 otherwise

$h_{s,q}$ = number of workers assigned to shift s having skill q

$l_{\hat{q},q,t}$ = number of shifts for workers with skill $\hat{q}$ covering demand for jobs requring skill q in period t

d_k = length of break k

Variables

S_j = number of workers assigned to shift profile $j \in J$

E_k = number of workers that are given sub-break k

$P_{\hat{q},q,t}$ = number of workers having skill $\hat{q}$ who are given a break in period t from a job requiring skill q

The implicit aspect of the BAP formulation requires constraints that:

Match shifts with break profiles: From the TShP, we obtain the set of shifts as well as the skill of the corresponding worker for each day of the planning horizon. When solving the BAP, we consider workers with the same shift and skill jointly. Therefore, instead of explicitly assigning each worker's shift a break profile, we split the shifts into groups that are then associated with eligible break profiles. *Example:* Suppose that 5 workers with skill q_2 are assigned a shift from 4 am to 11 am that is associated with two permissible break profiles β_1 =15/15/15 and β_2 =15/30. The workers are considered as a group, e.g., with 2 workers being assigned to β_1 and 3 workers to β_2.

Match sub-breaks with shift profiles: The number of sub-breaks that are eligible in the r-th position of break profile β is selected by means of forward and backward constraints. For each break profile, each position within it, and each skill level, these constraints ensure the feasibility of a transportation problem (to be defined presently) between the break profiles' corresponding shift profiles (supply nodes) and the break profiles' eligible sub-breaks (demand nodes). Thus, it is not necessary to explicitly assign sub-breaks to each shift profile.

Meet workstretch durations between sub-breaks: To ensure that the workstretch duration restriction between consecutive sub-breaks is met, a set of forward and backward constraints is established. For each break profile, each position in the break profile but the last, and each skill level, these constraints ensure the feasibility of a transportation problem that balances the number of sub-breaks with the number of their corresponding successor sub-breaks.

Assign breaks to shifts: When sub-break k covers period t, the demand coverage in period t has to be reduced by the number of workers that are given sub-break k. For each skill $\hat{q}$, we must assure that the number of sub-breaks associated with skill $\hat{q}$ and covering period t does not exceed the number of workers with skill $\hat{q}$ that are scheduled to work in period t in the TShP solution. Furthermore, in the context of downgrading, it is necessary to consider whether workers with skill $\hat{q}$ are covering demand requiring lower skill. Accordingly, for each period t and skill q, the number of sub-breaks of type k covering period t are assigned to demand q for $q \leq \hat{q}$ in an aggregated way such that the solution of TShP is respected. By using aggregation we do not have to explicitly reduce the demand for a specific skill for each sub-break and each period the sub-break covers. *Example:* In period 5, 4 workers with skill q_2 are assigned to a job requiring skill q_2 and 2 workers are assigned to a job requiring skill q_1. BAP has to ensure that not more than 4 (2) workers with skill q_2 are given a break in period 5 from a job requiring skill q_2 (q_1). If 3 workers with skill q_2 have a break in period 5, their breaks can be assigned aggregately to jobs by either assigning 0, 1, or 2 breaks to a job requiring skill q_1 while assigning 3, 2, or 1 breaks, respectively, to a job requiring skill q_2.

Before presenting our model, we outline the idea of using forward and backward constraints to ensure the feasibility of certain balanced transportation problems that were first introduced in Rekik et al. [105] and Çecik and Günlük [40]. We also give a high level description of the transportation problems included in our models. In all cases, the sum of the supply equals the sum of the demand.

Given a bipartite graph with a set of arcs A connecting the two sets of nodes N_1 and N_2, with each node $i \in N_1$ having supply O_i and each node $j \in N_2$ having demand D_i, we wish to determine the flow $Y_{i,j}$ from node i to node j for each $(i,j) \in A$. The constraints for the corresponding transportation problem $T(N_1, N_2)$ can be stated as follows.

$$\sum_{j:(i,j)\in\mathcal{A}} Y_{i,j} = O_i \qquad \forall\ i \in N_1 \tag{131}$$

$$\sum_{i:(i,j)\in\mathcal{A}} Y_{i,j} = D_j \qquad \forall\ i \in N_2 \tag{132}$$

Under the assumptions that a total order relation $\prec$ can be defined on set N_2, each supply node i is connected to a set of consecutive demand nodes P_i, and there exists no extraordinary overlap, i.e., there exists no two nodes i_1 and i_2 such that $P_{i_1} \subset P_{i_2}$, then the feasibility of transportation problem (131) – (132) is guaranteed by a set of forward, backward, and flow balance constraints. Using the following sets

$$N_2^s = \bigcup_{i=N_1} \{min(P_i)\}$$

$$N_2^e = \bigcup_{i=N_1} \{max(P_i)\}$$

$$N_2^B(j) = \{j' \in N_2 | j \preceq j'\} \quad \forall\ j \in N_2^s$$

$$N_2^F(j) = \{j' \in N_2 | j' \preceq j\} \quad \forall\ j \in N_2^e$$

$$N_1^B(j) = \left\{i \in N_1 | P_i \subseteq N_2^B(j)\right\} \quad \forall\ j \in N_2^s$$

$$N_1^F(j) = \left\{i \in N_1 | P_i \subseteq N_2^F(j)\right\} \quad \forall\ j \in N_2^e$$

and letting $n_2^s = min(N_2)$ and $n_2^e = max(N_2)$, the necessary constraints are

$$\sum_{j' \in N_2^F(j)} D_{j'} \geq \sum_{i \in N_1^F(j)} O_i \quad \forall\ j \in N_2^e \setminus \{n_2^e\}$$

$$\sum_{j' \in N_2^B(j)} D_{j'} \geq \sum_{i \in N_1^B(j)} O_i \quad \forall\ j \in N_2^s \setminus \{n_2^s\}$$

$$\sum_{i \in N_1} O_i - \sum_{j \in N_2} D_j = 0$$

To match sub-breaks with shift profiles, a transportation problem for each break profile, each position within the break profile, and each skill level has to be considered, i.e., a transportation problem $T(\mathcal{J}_{(\beta,q)}, \mathcal{B}_{(\beta,r,q)})$ for each $\beta \in \mathcal{BP}$, $r \in \{1, ..., B_\beta\}$ and $q \in \mathcal{Q}$. The supply associated with node j is S_j, whereas the demand for node k is E_k. An arc (j, k) exists between a shift $j \in \mathcal{J}_{(\beta,q)}$ and a sub-break $k \in \mathcal{B}_{(\beta,r,q)}$ if break k is eligible in the r-th break position of shift j. The sub-breaks' starting times define a total order on set $\mathcal{B}_{(\beta,r,q)}$. Finally, we assume no extraordinary overlap between breaks of different shift profiles in $J_{\beta,q}$. This is not a limitation for the tour scheduling problem under consideration or for most practical applications since there exist extensions that allow for extraordinary overlap by modifying the forward and backward constraints, e.g., see Addou and Soumis [3]. Note that to adapt the forward and backward constraints presented in Rekik et al. [106] to respect hierarchical skills, besides associating each $j \in \mathcal{J}$ and $k \in \mathcal{K}$ with a specific skill level, a separate transportation problem has to introduced for each skill level.

To obtain a feasible solution with respect to workstretch durations, the feasibility of a second set of transportation problems, $T(\mathcal{B}_{(\beta,r,q)}, \mathcal{B}_{(\beta,r+1,q)})$, must be assured. Here, the nodes correspond to breaks and are defined for all $\beta \in \mathcal{BP}$, $r \in \{1, ..., B_\beta - 1\}$ and $q \in \mathcal{Q}$. The supply and demand of each node k is E_k. Nodes $k_1 \in \mathcal{B}_{(\beta,r,q)}$ and $k_2 \in \mathcal{B}_{(\beta,r+1,q)}$ are connected if the workstretch duration between their corresponding breaks is within δ^{minBW} and δ^{maxBW}. Further, based on the break starting times, a totally ordered relationship $\prec$ on $\mathcal{B}_{(\beta,r,q)}$ is defined which per se excludes extraordinary overlap.

5.5.2.3 Implicit BAP formulation based on the Bechtold and Jacobs model

Our implicit formulation of the BAP is based on the model developed by Bechtold and Jacobs. It embodies the four components mentioned above along with a set of demand constraints. The objective here and for the two other formulations in Appendices A.6 and A.7 is to minimize the cost of uncovered periods minus the savings that result when workers are on their breaks.

Implicit BAP I (IBAP1)

$$\text{Minimize} \sum_{q\in\mathcal{Q}}\sum_{t\in\mathcal{T}} c^{\mathrm{uc}} \cdot y^{-}_{q,t} - \sum_{j\in\mathcal{J}} \sum_{r\in\{1,\ldots,B_j\}} \sum_{k\in\mathcal{K}_{j(r)}} c^{\mathrm{work}} \cdot d_k \cdot E_k \tag{133}$$

subject to

$$\sum_{\hat{q}\geq q} P_{\hat{q},q,t} + y^{+}_{q,t} - y^{-}_{q,t} = D_{q,d,t} \qquad \forall\ q \in \mathcal{Q}, t \in \mathcal{T} \tag{134}$$

$$\sum_{j\in\mathcal{J}_{s,q}} S_j = h_{s,q} \qquad \forall\ s \in \mathcal{S}, q \in \mathcal{Q} \tag{135}$$

$$\sum_{q\leq\hat{q}} P_{\hat{q},q,t} = \sum_{k\in\mathcal{K}_{\hat{q}}} \rho_{t,k} \cdot E_k \qquad \forall\ \hat{q} \in \mathcal{Q}, t \in \mathcal{T} \tag{136}$$

$$P_{\hat{q},q,t} \leq l_{\hat{q},q,t} \qquad \forall\ \hat{q}, q \in \mathcal{Q} : q \leq \hat{q}, t \in \mathcal{T} \tag{137}$$

$$\sum_{k'\in\mathcal{B}^{F}_{(\beta,r,q)}(k)} E_{k'} - \sum_{j\in\mathcal{J}^{F}_{(\beta,q)}(k)} S_j \geq 0 \qquad \forall\ \beta \in \mathcal{BP}, r \in \{1, \ldots, B_\beta\}, \; q \in \mathcal{Q}, k \in \mathcal{B}^{\mathrm{e}}_{(\beta,r,q)} \setminus \left\{k^{\mathrm{e}}_{(\beta,r,q)}\right\} \tag{138}$$

$$\sum_{k'\in\mathcal{B}^{B}_{(\beta,r,q)}(k)} E_{k'} - \sum_{j\in\mathcal{J}^{B}_{(\beta,q)}(k)} S_j \geq 0 \qquad \forall\ \beta \in \mathcal{BP}, r \in \{1, \ldots, B_\beta\}, \; q \in \mathcal{Q}, k \in \mathcal{B}^{\mathrm{s}}_{(\beta,r,q)} \setminus \left\{k^{\mathrm{s}}_{(\beta,r,q)}\right\} \tag{139}$$

$$\sum_{k\in\mathcal{B}_{(\beta,r,q)}} E_k - \sum_{j\in\mathcal{J}_{(\beta,q)}} S_j = 0 \qquad \forall\ \beta \in \mathcal{BP}, r \in \{1, \ldots, B_\beta\}, \; q \in \mathcal{Q} \tag{140}$$

$$\sum_{k'\in\mathcal{B}^{F}_{(\beta,r+1,q)}(k)} E_{k'} - \sum_{k'\in\mathcal{B}^{F}_{(\beta,r,q)}(k)} E_{k'} \geq 0 \quad \forall\ \beta \in \mathcal{BP}, r \in \{1, \ldots, B_\beta - 1\}, \; q \in \mathcal{Q}, k \in \mathcal{B}^{\mathrm{e}}_{(\beta,p+1,q)} \setminus \left\{k^{\mathrm{e}}_{(\beta,p+1,q}\right\} \tag{141}$$

$$\sum_{k'\in\mathcal{B}^{B}_{(\beta,r+1,q)}(k)} E_{k'} - \sum_{k'\in\mathcal{B}^{B}_{(\beta,r,q)}(k)} E_{k'} \geq 0 \quad \forall\ \beta \in \mathcal{BP}, r \in \{1, \ldots, B_\beta - 1\}, \; q \in \mathcal{Q}, k \in \mathcal{B}^{\mathrm{s}}_{(\beta,r+1,q)} \setminus \left\{k^{\mathrm{s}}_{(\beta,r+1,q}\right\} \tag{142}$$

$$\sum_{k\in\mathcal{B}_{(\beta,r+1,q)}} E_k - \sum_{k\in\mathcal{B}_{(\beta,r,q)}} E_k = 0 \qquad \forall\ \beta \in \mathcal{BP}, r \in \{1, \ldots, B_\beta - 1\}, \; q \in \mathcal{Q} \tag{143}$$

$$P_{\hat{q},q,t} \in \mathbb{Z}_+ \qquad \forall\ \hat{q}, q \in \mathcal{Q} : q \leq \hat{q}, t \in \mathcal{T} \tag{144}$$

$$S_j, E_k \in \mathbb{Z}_+ \qquad \forall\ \ j \in \mathcal{J}, k \in \mathcal{K} \tag{145}$$

Constraints (134) determine the amount of demand for skill level q on day d in period t that cannot be covered. While Rekik et al. [106] considered a shift scheduling problem, for the BAP that we are solving, the set of shifts are derived from the TShP and taken as input. To match shifts with permissible sub-breaks, each shift needs to be associated with a break profile. In constraints (135), the linking of shifts with break profiles is done implicitly, i.e., instead of explicitly assigning a break profile to each shift, identical shifts assigned to workers with the same skill are aggregated based on admissible break profiles. Constraints (136) ensure that the number of sub-breaks given in period t for workers with skill $\hat{q}$ are only distributed among jobs that require at most skill $\hat{q}$. Note that this is only possible when each sub-break k is associated with a specific skill. In constraints (137), the assignment of sub-breaks is bounded by $l_{\hat{q},q,t}$ such that breaks can only be given when the underlying jobs are being performed by workers having the specified skill. Again, workers with skill $\hat{q}$ who are performing a job that requires skill q in period t are grouped together. This arrangement guarantees that a feasible solution to the problem of explicitly assigning sub-breaks to workers can be found in a post-processing step.

The feasibility of the transportation problems for workstretch duration restrictions between successive sub-breaks is ensured by the forward, backward and flow balance constraints given in (138), (139) and (140), respectively. Similarly, constraints (141), (142) and (143) are respectively the forward, backward and flow balance constraints that ensure that enough sub-breaks matching the shift profiles are assigned. Variables are defined in (144) and (145).

5.6 Computational Study

The computations were performed on a Windows 7 platform with 4 GB RAM and a 2.8 GHz processor. All models were implemented in JAVA and solved with CPLEX 12.6. The data sets were provided by a major European ground handling service company. Each working day was divided into ninety-six, 15-minute periods.

Our study is based on two templates: 'Fix' and 'Flex,' differing in the length of the start time window, the duration of a shift (see Table 5.5), and in the flexibility of the possible shift types per day (see Table 5.6). The resulting number of shifts for the Fix template is at least the number of starting times, which is 8, while for Flex templates there are at least 319 (= 29×11 = number of start times multiplied by the number of shift lengths). Note that each hour is divided in four, 15-minute periods. If on one day, two shift types are possible, e.g., see Table 5.6 in week 1 on Wednesday, then the number of feasible shifts doubles.

Table 5.5: Parameter values for the fixed and flexible work templates

	Start time window		Duration
Type	Early shift	Late shift	(periods)
Fix	[4am, 6am]	[12am, 2pm]	41
Flex	[3am, 10am]	[11am, 6pm]	[36, 46]

We consider from 10 to 300 workers and five skill levels, where it is assumed that the demand for each level is 20% of the total. The maximum amount of overtime and undertime per worker is given by $O^+ = O^- = 100$; the individual amount of overtime or undertime per worker was randomly assigned from the interval [0,100]. Since undercoverage in a period indicates missing workers, we assume that the cost per period of undercoverage is equal to the labor cost per period which is 1.

Table 5.6: Work templates – Fix and Flex

(a) Fix

	Day						
Week	Mo	Tu	We	Th	Fr	Sa	Su
1				E	E	L	L
2	L	L					
3	E	E	E/L	L	L		
4				E	E	E	E
5	L	L				E	E
6	E	E	L	L			
7			E	E	L	L	L
8	L				E	E	E
9	L	E	E				
10		E	E	L	L	L	L

(b) Flex

	Day						
Week	Mo	Tu	We	Th	Fr	Sa	Su
1	E	E	E/L	L	L	L/X	
2				E	E	E/L	L
3	L	L/X					E
4	E	E/L	L	L	L/X		
5			E	E	E/L	L	L
6	L/X					E	E
7	E/L	L	L	L/X			
8		E	E	E/L	L	L	L/X
9					E	E	E/L
10	L	L	L/X				

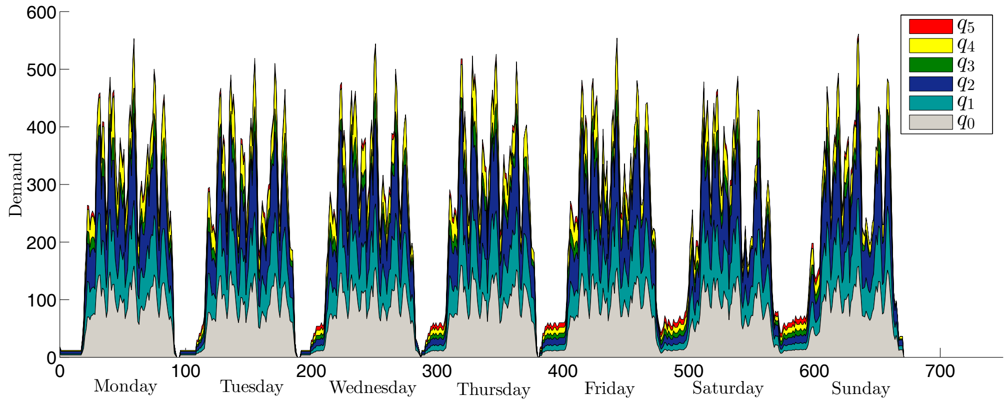
Table 5.7 presents the settings for the break regulations. The total break length (see column "Dur tot") in each break regulation is set to 6 periods. The first sub-break starts between periods 8 and 24 periods after the shift start (column "FB"). The last break (column "LB") for multiple or fractional breaks has to start between 3 and 16 periods

before the shift ends. For all regulations with multiple breaks, 3 sub-breaks have to be assigned. To guarantee a total break length of 6, in multiple break scenario $\{M|X|T\}$ with fixed sub-break length, each sub-break is 2 periods, while in multiple break scenario $\{M|V|T\}$ with variable sub-break length, each sub-break takes between 1 and 4 periods (column "Dur SB"). After the shift starts, the time windows of the second and third breaks are $[26, 29]$ and $[30, 33]$, respectively (column "TW"). In the two scenarios with fractional breaks, $\{F|V|T\}$ and $\{F|V|W\}$, 1 to 3 sub-breaks can be assigned to each shift, which are either based on time windows or workstretch durations. The time windows for the second and third sub-breaks associated with $\{F|V|T\}$ are equal to those in the multiple break regulation case. For fractional break regulation $\{F|V|T\}$ the workstretch duration between the successively sub-breaks after the first sub-break is from 3 to 24 periods (column "WS").

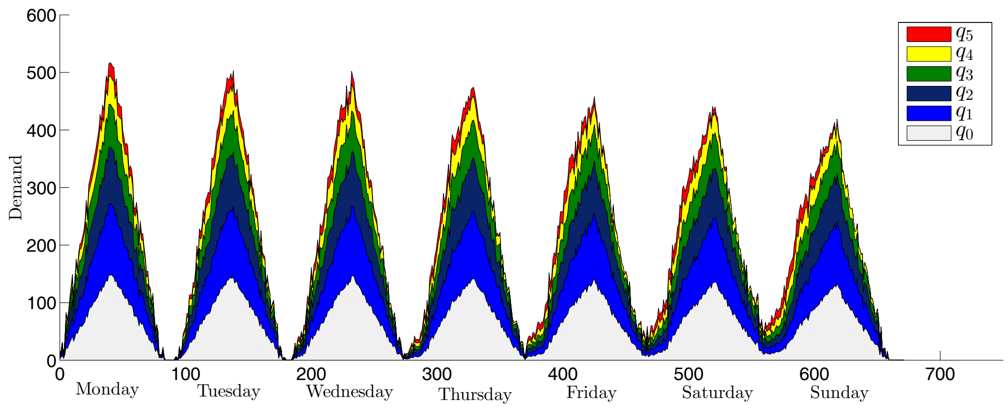
Table 5.7: Settings for the break regulations

	Setting						
Break regulation	#Breaks	Dur tot	Dur SB	WS	TW	FB	LB
$\{S\|X\|T\}$	1	6				$[8, 24]$	
$\{M\|X\|T\}$	3	6	2		$[26, 29]$, $[30, 33]$	$[8, 24]$	
$\{M\|V\|T\}$	3	6	$[1, 4]$		$[26, 29]$, $[30, 33]$	$[8, 24]$	$[3, 16]$
$\{F\|V\|T\}$	$[1, 3]$	6	$[1, 6]$		$[26, 29]$, $[30, 33]$	$[8, 24]$	$[3, 16]$
$\{F\|V\|W\}$	$[1, 3]$	6	$[1, 6]$	$[3, 24]$		$[8, 24]$	$[3, 16]$

With respect to demand, both low- and high-level scenarios were investigated along with two different demand profiles: "varying demand" (VD) and "stable demand" (SD). Figure 5.6 depicts the profiles VD and SD for Monday to Sunday where each day is divided into 98 periods. The VD high level demand curve represents the true demand during the course of a week for each qualification of the sponsoring company. The VD low-level demand curves were derived from this curve by randomly lowering the demand. Appendix A.8 yields the three stable demand curves for the same days and periods.



(a) Varying high level demand



(b) Stable high level demand

Figure 5.6: Profiles for high-level demand

To uniquely identify a test instance we use the 4-tuple representation $I\,\{wor, pro, dem, temp\}$ specifying the workforce size $wor \in \{10, \ldots, 50, 100, 150, 200, 250, 300, 350, 400\}$, the demand profile $pro \in \{\text{VD, SD}\}$, the level $dem \in \{\text{low}, \text{high}\}$, and the work template in Table 5.6 $temp \in \{\text{Fix}, \text{Flex}\}$.

The analysis is presented in two separate subsections: In §5.6.1, we compare the solution quality of our MIP-heuristic with solutions obtained with the compact MIP (CMIP) formulation in Appendix A.5, which integrates the tour scheduling problem and the BAP. In §5.6.2, we evaluate the "cost" of break flexibility by first generating tours by solving TShP (115) – (129). In light of those results, we study the different combinations of BAP formulations and BAP regulations to determine runtime and solution quality.

5.6.1 Performance MIP-Heuristic

To study the performance of the MIP-heuristic in comparison to the CMIP we restrict the workforce size to be between 10 and 65 employees for the low demand scenario. The break regulations investigated are $\{S|X|T\}$, which represents the most common break regulations considered in literature (see Table 5.4), and BAP$\{F|V|T\}$, which offers the greatest flexibility (see Figure 5.1). In striving for a fair comparison, we solve Implicit BAP formulation III in Appendix A.7 in the second step of the MIP-heuristic due to its structural similarity to the break assignment constraints in the CMIP.

The results for BAP$\{S|V|T\}$ and BAP$\{F|V|W\}$ are summarized in Tables 5.8 and 5.9, respectively. Column "Instance" identifies the scenario with respect to number of workers, work template, and demand characteristics. Columns "obj" and "undercov" report the objective value and undercoverage costs, respectively. The computation times are given in column "runtime" in seconds. Column "GAP*" gives the % gap between the optimal solution of the CMIP and the MIP-heuristic. In Table 5.9, we have added the column "GAP" to report the % gap between the solution found by the MIP-heuristic and the best LP bound obtained when no solution was found for the CMIP within the 5-hour (18,000 sec) limit placed on each run.

The computations reveal that the MIP-heuristic performs well with respect to the CMIP. The largest gaps of 7.52% and 9.33% for the variable and stable demand curves, respectively, are reached when no feasible or optimal solution for CMIP is found (see Table 5.9, instances $\{50, \text{low, SD,, fix}\}$) and $\{40, \text{low, VD, fix}\}$)). If we restrict the results to those instances for which the optimal solution of CMIP is found, the gaps of the MIP-heuristic are at most 4.48% (see Table 5.9, instance $\{50, \text{low, SD, fix}\}$). The gaps between CMIP and the MIP-heurstic for the instances with the fix work templates are on average greater than those for the flex work template. In the first step of the MIP-heuristic, the shifts obtained under flexible work templates provide more demand coverage than the shifts obtained with fix work templates and, therefore, lower the benefits of break flexibility.

A closer look at Tables 5.8 and 5.9 reveals the impact of break flexibility. Consider, for example, the instances with 50 workers. In the singe break scenario $\{S|V|T\}$, if we compare instances for fixed and flexible work templates based on the stable demand curve, e.g., instances $\{50, \text{low, SD, fix}\}$ and $\{50, \text{low, SD, flex}\}$, flexible work templates im-

Table 5.8: Results for the CMIP and the MIP-heuristic for break regulation $\{S|V|T\}$

	CMIP			MIP-heuristic			
Instance	obj	undercov	runtime (s)	obj	undercov	runtime (s)	GAP*
$\{10, \text{low}, \text{ SD}, \text{fix}\}$	$7,570$	$6,520$	15.77	$7,570$	$6,520$	3.68	0.00
$\{20, \text{low}, \text{ SD}, \text{fix}\}$	$7,680$	$5,055$	30.92	$7,680$	$5,055$	4.45	0.00
$\{30, \text{low}, \text{ SD}, \text{fix}\}$	$8,035$	$4,010$	78.81	$8,035$	$4,010$	8	0.00
$\{40, \text{low}, \text{ SD}, \text{fix}\}$	$8,695$	$3,095$	272	$8,750$	$3,150$	9.42	0.63
$\{50, \text{low}, \text{ SD}, \text{fix}\}$	$9,805$	$2,630$	268.53	$9,975$	$2,800$	9.51	1.7
$\{65, \text{low}, \text{ SD}, \text{fix}\}$	$11,150$	$2,050$	55.3	$11,450$	$2,360$	11.8	2.71
$\{10, \text{low}, \text{ SD}, \text{flex}\}$	$7,350$	$6,033$	33.9	$7,350$	$5,690$	8.82	0.00
$\{20, \text{low}, \text{ SD}, \text{flex}\}$	$7,350$	$5,447$	94.78	$7,350$	$4,595$	6.83	0.00
$\{30, \text{low}, \text{ SD}, \text{flex}\}$	$7,350$	$4,652$	$1,147.93$	$7,350$	$3,620$	8.53	0.00
$\{40, \text{low}, \text{ SD}, \text{flex}\}$	$7,350$	$3,857$	$2,868.06$	$7,350$	$3,220$	13.45	0.00
$\{50, \text{low}, \text{ SD}, \text{flex}\}$	$7,350$	$3,328$	$10,074.84$	$7,350$	$3,010$	12.72	0.00
$\{65, \text{low}, \text{ SD}, \text{flex}\}$	$7,380$	$2,077$	$14,074$	$7,420$	$2,495$	57.436	0.54
$\{10, \text{low}, \text{ VD}, \text{fix}\}$	$9,086$	$8,036$	2.6	$9,086$	$8,036$	1.34	0.00
$\{20, \text{low}, \text{ VD}, \text{fix}\}$	$9,121$	$6,496$	8.48	$9,141$	$6,517$	2.14	0.22
$\{30, \text{low}, \text{ VD}, \text{fix}\}$	$9,283$	$5,258$	99.49	$9,307$	$5,245$	5.92	0.26
$\{40, \text{low}, \text{ VD}, \text{fix}\}$	$9,663$	$3,957$	748.99	$9,851$	$5,599$	10.64	1.91
$\{50, \text{low}, \text{ VD}, \text{fix}\}$	$10,188$	$3,013$	830.01	$10,486$	$3,311$	12.58	2.84
$\{65, \text{low}, \text{ VD}, \text{fix}\}$	$11,234$	$2,134$	$499,07$	$11,671$	$2,572$	$12,76$	3.74
$\{10, \text{low}, \text{ VD}, \text{flex}\}$	$9,084$	$7,857$	20.55	$9,089$	$7,133$	4.93	0.01
$\{20, \text{low}, \text{ VD}, \text{flex}\}$	$9,084$	$7,070$	436.42	$9,105$	$5,759$	11.72	0.23
$\{30, \text{low}, \text{ VD}, \text{flex}\}$	$9,084$	$6,158$	$2,243.9$	$9,113$	$4,740$	21.34	0.32
$\{40, \text{low}, \text{ VD}, \text{flex}\}$	$9,084$	$5,028$	$5,326.08$	$9,147$	$3,850$	36.08	0.69
$\{50, \text{low}, \text{ VD}, \text{flex}\}$	$9,084$	$4,546$	$8,343.92$	$9,185$	$3,503$	87.52	1.10
$\{65, \text{low}, \text{ VD}, \text{flex}\}$	$9,368$	$3,727$	$18,000$	$9,518$	$3,225$	614.95	1.58

prove the solution by $2,455$ $(= 9,805 - 7,350)$ or 25%. In contrast, comparing instance $\{50, \text{low}, \text{ SD}, \text{flex}\}$ for break regulations $\{S|V|T\}$ and $\{F|V|W\}$, we see that latter improves the solution by 318 $(= 9,805 - 9,487)$, which represents 13% $(= 100\% \times 318/2,455)$ of the improvement in the previous comparison. The improvement between the two break regulations becomes more significant when we focus on instances with varying demand. The difference between instances $\{50, \text{low}, \text{ VD}, \text{fix}\}$ and $\{50, \text{low}, \text{ VD}, \text{flex}\}$ with break regulation $\{S|V|T\}$ is $1,104$ $(= 10,188 - 9084)$ or 10.8%. The comparisons between the two break regulations $\{S|V|T\}$ and $\{F|V|W\}$ for these instances leads to a decrease in the objective of 404 $(= 10,188 - 9,784)$ which is equivalent to 37% of the improvement that is obtained when flexible rather than fixed work templates are employed.

Our two-step approach is seen to provide high quality solutions in less than 100 sec-

Table 5.9: Results for the CMIP and the MIP-heuristic for break regulation $\{F|V|W\}$

	CMIP			MIP-heuristic				
Instance	obj	undercov	runtime (s)	obj	undercov	runtime (s)	GAP*	GAP
{10, low, SD, fix}	7,510	6,460	32.32	7,510	6,460	3.56	0.00	
{20, low, SD, fix}	7,590	4,965	189.32	7,590	4,965	4.10	0.00	
{30, low, SD, fix}	7,890	3,865	1,076.60	7,900	3,875	13.92	0.13	
{40, low, SD, fix}	8,412	2,813	2,454.28	8,550	2,740	19.55	1.61	
{50, low, SD, fix}	9,487	2,312	2,744.08	9,640	2,465	22.21	1.59	
{65, low, SD, fix}				11,164	2,064	29.47		7.52
{10, low, SD, flex}	7,350	5,690	203.92	7,350	5,690	9.05	0.00	
{20, low, SD, flex}				7,350	4,595	11.16		0.32
{30, low, SD, flex}				7,350	3,620	13.62		0.78
{40, low, SD, flex}				7,350	3,220	15.20		0.96
{50, low, SD, flex}				7,350	3,010	17.71		1.12
{65, low, SD, flex}				7,420	2,495	59.18		0.94
{10, low, VD, fix}	9,084	8,034	31.97	9,086	8,036	3.54	0.02	
{20, low, VD, fix}	9,108	6,483	164.98	9,133	6,500	4	0.27	
{30, low, VD, fix}	9,247	5,222	4,345.21	9,247	5,222	13.32	0.43	
{40, low, VD, fix}	9,545	3,839	7,938, 34	9,545	3,862	14.29	1.37	
{50, low, VD, fix}	9,784	2,564	10,800.3	10,243	3,068	18.21	4.48	
{65, low, VD, fix}				11,443	2,343	42.01		9.33
{10, low, VD, flex}	9,084	7,580	843.80	9,084	7,128	8.72	0.00	
{20, low, VD, flex}				9,087	5,741	12.20		0.12
{30, low, VD, flex}				9,093	4,720	28.30		0.42
{40, low, VD, flex}				9,128	3,831	43.16		1.00
{50, low, VD, flex}				9,160	3,478	98.51		1.35
{65, low, VD, flex}				10,367	4,286	628.68		1.64

onds. Timely computations are critical when there is a frequent need to replan shifts and breaks due to workforce and demand fluctuations. Moreover, when the demand is largely uncertain from day to day, the only option is to reset the breaks since shift start and end times cannot be adjusted during the course of the day or even for the upcoming day without the likelihood of violating labor laws or incurring overtime costs. In the next section, break flexibility is explored in more detail.

5.6.2 Benefits of break flexibility

In the second phase of our experiments, we examined the relationships among break flexibility, undercoverage, and runtime for the various scenarios in Table 5.7. To the best of our knowledge, this is the first study in which real data are used to determine the effect of break regulations on large tour scheduling problems. To begin, Algorithm 2 is solved to generate shifts and then the BAP is solved using the three different implicit formulations. The study demonstrates the degree to which the various break regulations can serve to reduce undercoverage when the shifts are given. For the variable and stable demand curves with both low and high demand, we consider the workforce sizes given in Table 5.10.

Table 5.10: Number of workers considered for the two demand levels

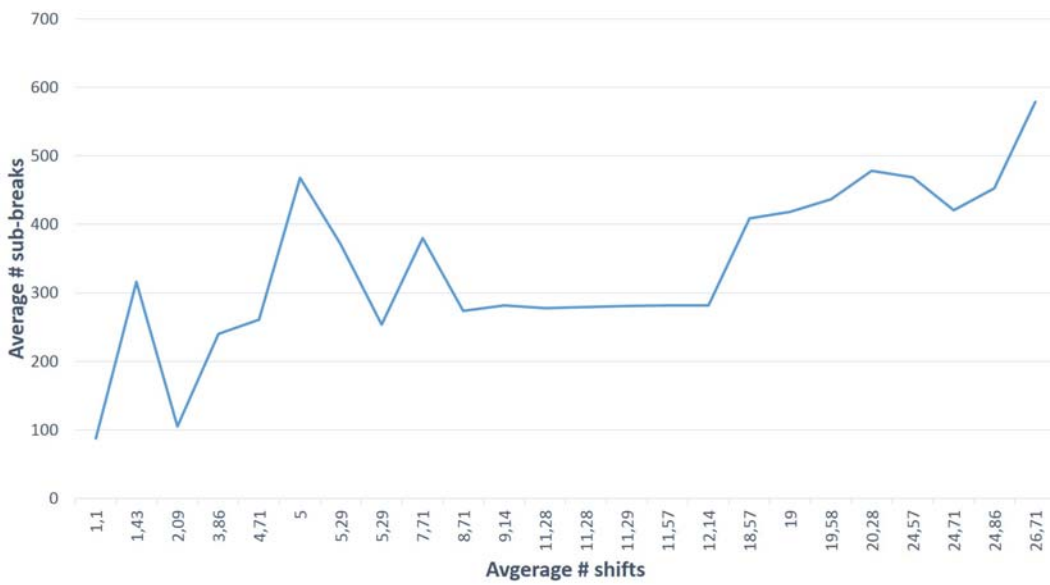
Demand curve	#Workers
low	$[10, 50, 100]$
high	$[200, 250, 300]$

Table 5.11 provides the statistics obtained from Algorithm 2 in the first step of the MIP-heuristic. Column “time (s)” reports the runtime in seconds. The average number of workers and the resulting number of shifts per day are shown in column “#worker” and “#shifts,” respectively. Based on these shifts, the columns under the major heading “#sub-breaks” report the number of sub-breaks for each break regulation in Table 5.7. It can be seen that for fixed work templates fewer shifts are generated, and hence there are fewer sub-breaks for each break regulation compared to the instances that have flexible work templates. For example, for any number of workers associated with instances $\{\cdot,$ high, VD, flex$\}$, there are significantly more sub-breaks generated for any considered break regulation than for instances $\{\cdot,$ high, VD, fix$\}$ with the same number workers.

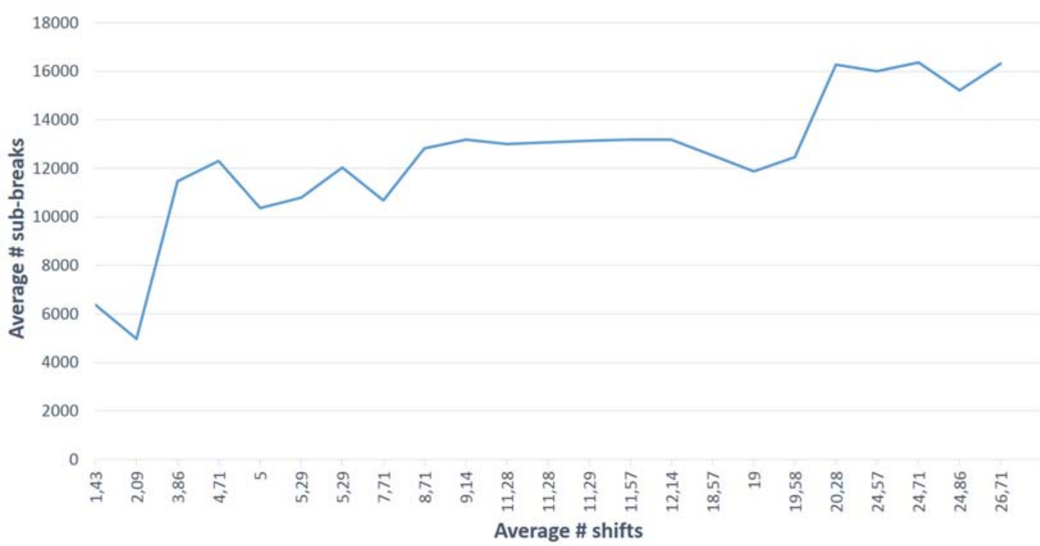
Table 5.11: Results of the ThSP

Instance	time (s)	#workers	#shifts	#sub-breaks				
				$\{S\|V\|T\}$	$\{M\|X\|T\}$	$\{M\|V\|T\}$	$\{F\|V\|T\}$	$\{F\|V\|W\}$
{10, low, SD, fix}	0.79	4.28	1.1	87.43	82.29	97.71	1, 429.71	4, 278.86
{50, low, SD, fix}	4.25	29.29	3.86	240	308.57	3, 685.71	5, 093.14	11, 460
{100, low, SD, fix}	8.56	58.57	4.71	260.57	336.86	4, 122.86	5, 696.57	12, 303.43
{10, low, SD, flex}	3.4	7.14	1.43	316.29	128.57	1, 285.71	2, 108.57	6, 378
{50, low, SD, flex}	11.45	25	5	468	425.14	4, 251.43	6, 227.14	10, 354.29
{100, low,SD, flex}	1, 552.43	50	7.71	379.71	281.14	2, 811.43	4, 238.57	10, 675.34
{10, low, VD, fix}	1.96	4.28	2.09	104.57	122.57	1, 465.71	2, 027.14	4, 981.71
{50, low, VD, fix}	4.53	29.29	12.14	282	418.29	4, 662.86	6, 428.57	13, 182
{100, low,VD, fix}	9.22	58.57	11.29	281.14	421.71	4, 714.29	6, 498.86	13, 146.86
{10, low, VD, flex}	8.21	7.14	5.29	371.14	250.29	2, 502.86	4, 228.29	10, 798.29
{50, low, VD, flex}	66.15	28.14	26.71	578.57	795.42	7, 954.29	11, 544	16, 317.43
{100, low, VD, flex}	2, 567.69	51.71	20.28	478.29	798	7, 980	11.490	16, 263.43
{200, high, SD, fix}	51.82	117, 14	5.29	253.71	339.43	3, 394.29	5, 374.29	12, 022.29
{250, high, SD, fix}	60.75	146.43	8.71	273.43	397.71	3, 977.14	6, 147.43	12, 830.57
{300, high, SD, fix}	40.6	175.71	9.14	282	424.29	4, 242.86	6, 534	13, 182
{200, high, SD, flex}	2, 102.32	112	19	418	404.57	4, 045.71	6, 167.14	11, 882.57
{250, high, SD, flex}	2, 503.74	125	18.57	408.57	487.71	4, 877.14	7, 311.43	12, 538.29
{300, high, SD, flex}	7, 628.69	150	19.85	436.86	524.57	5, 254.29	7, 518	12, 467
{200, high, VD, fix}	33.15	117.14	11.57	282	400.29	4, 002.86	6, 182.57	13, 182
{250, high, VD, fix}	35.71	164.43	11.28	277.71	423.43	4, 234.29	6, 498.86	13, 006.29
{300, high, VD, fix}	40.50	175.71	11.28	279.43	405.43	4, 054.29	6, 252.86	13, 076, 57
{200, high, VD, flex}	3, 212.99	107.42	24.57	468.86	799.71	7, 997.14	12, 022.29	15, 997.71
{250, high, VD, flex}	4, 590.07	142.86	24.86	452.57	789.43	7, 894.29	11, 478.86	15, 223.71
{300, high, VD, flex}	4, 625.97	152.57	24.71	420.86	833.14	8, 331.43	12, 081.43	16, 368

The results in Table 5.11 reveal that as the workforce increases so does the runtime of Algorithm 2 when flexible work templates are used. Also, the type of demand curve is seen to affect runtime; instances with higher fluctuating demand take longer to converge. The number of shift types depends on the workforce size and on the work template type. For flexible work templates more individual tours are generated to cover the demand. Of course, it is no surprise that the greater the flexibility available to generate shifts, the greater the number shifts and, hence, sub-breaks. Similar to the hypothetical example in Figure 5.2, the number of sub-breaks increases geometrically with the flexibility of the break regulations. We also observed that as soon as a distinct number of shifts is used as a basis for the generation of sub-breaks, increasing the number of shifts beyond that threshold leads to a decrease in the number of newly generated sub-breaks. For example, for break regulations $\{S|V|T\}$ and $\{F|V|W\}$ the number of newly generated sub-breaks only increases slightly when 5 or more shifts are used (see Figure 5.7). A similar effect can be observed for the other three break regulations in Table 5.11. Hence, for any break regulation there exist a bounded set of shifts leading to a bounded set of all possible sub-breaks.



(a) $\{S|V|T\}$



(b) $\{F|V|W\}$

Figure 5.7: Number of sub-breaks generated for $\{S|V|T\}$ and $\{F|V|W\}$ as the number of shifts increases

Tables 5.12 and 5.13 compare the three BAP formulations: Implicit BAP I given in Section 5.5.2.3, Implicit BAP II given in Appendix A.6, and Implicit BAP III given in

Appendix A.7, for break regulation $\{S|V|T\}$ and $\{F|V|W\}$. Column "comp (s)" reports the running time of CPLEX in seconds for each BAP formulation summed over the 7 planning days. For Implicit BAP I and II, it is necessary to derive the sets and parameters accompanying the models, given the set of shifts for each day, e.g., for Implicit BAP formulation I, we need sub-break set $\mathcal{B}$ and parameters $h_{s,q}$. The time in seconds for all the pre-processing summed over the 7 planning days is reported in column "pre (s)." The average number of variables and constraints for each BAP formulation per day is given in columns "#var" and "#constr," respectively. If the fields in the table are empty, the corresponding BAP formulation could not find the optimal solution within the allowed time limit of 30 minutes (1,800 seconds).

All problem instances with break regulation $\{S|V|T\}$ could be solved for all of three BAP formulations (see Table 5.12). Comparing the total of runtime plus pre-processing time, Implicit BAP I and II slightly outperformed BAP III. The reason may be found in the increased number of variables in BAP III. However, when considering the results for the most flexible break regulation $\{F|V|T\}$ in Table 5.13, Implicit BAP II performed worse than its other two competitors and is only solvable for small instances with less than 100 workers and fixed work templates. For a larger workforce, or for flexible work templates, the large number of sub-breaks ($> 10,000$ sub-breaks, see Table 5.11), leads to more than $500,000$ variables in Implicit BAP formulation II, which makes the problem intractable. However, not all the instances associated with Implicit BAP III could be solved either due to the huge number of variables and constraints.

In addition to the above comparisons of the computational tractability of the different formulations, Table 5.15 highlights the objective function deviation in % of the four BAP regulations $\{S|X|T\}$, $\{S|V|T\}$, $\{M|V|T\}$ and $\{F|V|T\}$ from the most flexible break regulation $\{F|V|W\}$. While the gaps between break regulations $\{S|V|T\}$ and $\{F|V|W\}$ are rather large with deviations up to 17.61% (see instance $\{50,$ low, VD, fix$\}$), break regulations with fractional breaks and time windows show gaps of at most 3.56% (see instance $\{100,$ high,VD, fix$\}$).

The improvements from models with less flexibility to models with more flexibility are highlighted in Table 5.14. For example, when break regulation $\{M|V|T\}$ is used, undercoverage decreases by 4.11%, on average, compared to the results obtained with break regulation $\{S|V|T\}$ (see row "Fix and Flex" in Table 5.14 which considers all templates

Table 5.12: Comparison of implicit BAP formulations for break regulation $\{S|V|T\}$

Instance	IBAP I				IBAP II				IBAP III		
	comp (s)	pre (s)	#var	#constr	comp (s)	pre (s)	#var	#constr	time (s)	#var	#constr
{10, low, SD, fix}	1.35	0.02	2, 803.71	2, 727.43	1.08	0.03	2, 730.86	2, 721.43	2.89	790.29	2, 823.43
{50, low, SD, fix}	1.74	0.07	3, 411.86	3, 204.86	1.63	0.4	3, 237.43	3, 195	5.26	12, 573	2, 040.29
{100, low, SD, fix}	1.97	0.09	3, 433.29	3, 211.71	1.26	0.9	3, 252.86	3, 201	25.26	23, 994.86	3, 973.14
{10, low, SD, flex}	0.67	0.05	3, 485.71	3, 185.43	0.57	0.04	3, 200.86	3, 178.86	0.58	2, 699.14	898.86
{50, low, SD, flex}	1.76	0.18	3, 641	3, 214	1.46	0.03	3, 283	3, 203	1.56	5, 982	1, 511
{100, low,SD, flex}	13.34	0.58	3, 555.43	3, 234.29	10.73	1	3, 345.43	3, 222	3.26	10, 418.57	2, 314
{10, low, VD, fix}	1.47	0.02	2, 821.86	2, 735.43	1.4	0.4	2, 748.86	2, 728.43	0.14	2, 013.43	756
{50, low, VD, fix}	2.18	0.17	3, 462.14	3, 271.14	1.86	0.4	3, 386.57	3253	2, 07	7, 653.43	1, 806
{100, low,VD, fix}	2.31	0.18	3, 460.43	3, 264.29	2.9	0.32	3, 371.14	3, 247	13.26	14, 154.86	3, 036
{10, low, VD, flex}	0.98	0.14	3, 544.43	3, 126.29	1.73	0.02	3, 276.43	3, 201	0.51	2, 826	941.14
{50, low, VD, flex}	1.83	0.63	3, 779.29	3, 385.43	0.34	0.04	3, 332.57	3, 357	2.51	6, 879.86	1, 725.43
{100, low, VD, flex}	1.77	2.19	3, 658.14	3, 267.14	1.07	0.23	3, 440.71	3, 251	4.71	10, 366.29	2, 243.43
{200, high, SD, fix}	1.72	0.23	3, 427	3, 216.29	2.75	0.05	3, 263.14	3, 205	18.97	27, 157.71	5, 496
{250, high, SD, fix}	3.15	0.29	3, 450.14	3, 243.71	2.73	0.06	3, 324.86	3229	19.5	33, 659.14	6, 726
{300, high, SD, fix}	2.55	0.2	3459.14	3, 247.14	2	0.04	3, 332.57	3232	20.96	40, 160.57	7, 956
{200, high, SD, flex}	2.02	0.44	3, 516.29	3, 318.57	1.8	0.08	3, 605	3, 301	10.16	25, 488	5, 571.43
{250, high, SD, flex}	1.55	0.59	3, 569	3, 329.71	0.23	0.27	3, 595.14	3, 298	11.92	26, 324.14	5591.71
{300, high, SD, flex}	1.2	0.16	3, 561, 86	3, 238.57	1.57	1.19	3, 624.71	3, 307	16.96	28, 550.86	5, 658.86
{200, high, VD, fix}	2.96	0.32	3, 461.57	3, 266.57	3.22	0.06	3, 376.29	3, 249	17.34	27, 157.71	5, 496
{250, high, VD, fix}	2.88	0.35	3, 457	3, 264.29	1.38	0.04	3, 371.14	3, 247	22.17	33, 659.14	6, 726
{300, high, VD, fix}	2.2	0.26	3, 458.71	3, 264.29	2.76	0.04	3, 371.14	3, 247	27.76	40, 160.57	7, 956
{200, high, VD, flex}	2.65	0.94	3, 613.86	3, 281.21	1.73	0.17	3, 486.71	3, 265	13.3	25, 488	5, 571.43
{250, high, VD, flex}	2.06	0.24	3, 573.43	3, 234	4.15	1.41	3, 345.43	3, 222	20.03	27, 343.34	6, 432.32
{300, high, VD, flex}	5.02	0.36	3, 596.71	3, 234.86	2.43	0.43	3, 348.71	3, 223	29.03	29.038.32	8, 231.21

Table 5.13: Comparison of implicit BAP formulations for break regulation $\{F|V|W\}$

	IBAP I				IBAP II				IBAP III		
Instance	comp (s)	pre (s)	#var	#constr	comp (s)	pre (s)	#var	#constr	time (s)	#var	#constr
{10, low, SD, fix}	3.9	1.49	7,008	3,173.14	56.49	41.42	87,178.29	4,160.29	1.17	790.29	2,823.43
{50, low, SD, fix}	14.18	63.43	14,689.71	4,096.29					17.95	12,573.43	2,040.29
{100, low, SD, fix}	20.28	146.87	15,546.86	4,214.57	512.45		516,442.3	11,948.29	25.26	23,994.86	3,973.14
{10, low, SD, flex}	5.52	2.84	9,568.86	3,761.14					1.02	3,847.71	920.29
{50, low, SD, flex}	11	8.18	13,602.29	4,202					3.2	9,222	1,786
{100, low,SD, flex}	30.36	8.37	14,521.71	4,590.43					50.23	37,453.23	5,433.33
{10, low, VD, fix}	5.63	10.16	7,726.86	3,311.14	3.19	11.09	52,421.86	3,690.43	0.56	2,823.43	790.29
{50, low, VD, fix}	31.39	749.23	16,544.29	5,239.71					52.13	12,573.43	2,040.29
{100, low,VD, fix}	43.1	791.62	16,495.43	5.121.43	125.23	2,435.46	516,442.3	11,948.29	34.5	23,994.86	3,504.57
{10, low, VD, flex}	21.78	275.38	14,050.86	4,293.43					1.42	4,143.71	962.57
{50, low, VD, flex}	81.2	1,401	19,912.86	7,032.43					7.6	1,809.56	38,201.5
{100, low, VD, flex}	70.445	43.25	19,756	6,330.86					18.29	15,996	2,399.43
{200, high, SD, fix}	18.93	120.32	15,274.86	4,293.43					60.01	46,837.71	7,370.29
{250, high, SD, fix}	27.23	363.83	16,138	4,766.57							
{300, high, SD, fix}	26.24	393.88	16,496.29	4,825.71					2,109.57	69,680.57	10,767.43
{200, high, SD, flex}	43.5	1,182.73	15,973.43	6,329.14					157	46,837.71	7,370.29
{250, high, SD, flex}	46.10	643.39	16,074.14	6,074.86							
{300, high, SD, flex}	90.01	295.13	15,973.43	6,329.14							
{200, high, VD, fix}	33.18	621.77	16,535.14	5,160.86							
{250, high, VD, fix}	33.62	585.01	16,354.86	5,121.43							
{300, high, VD, fix}	32.4	598.46	16,425.14	5,121.43							
{200, high, VD, flex}	91.87	1,580.92	19,558.14	6,628.14							
{250, high, VD, flex}	82.06	1,109.23	18,782.57	6,541.29							
{300, high, VD, flex}	76.21	815.15	19,728.29	6,689							

in Tables 5.5 and 5.6). If we restrict the benefits of increased break flexibility to instances with fixed work templates only, then we obtain average improvements of 3.19%, 0.34%, 0.22% and 0.24%, respectively (see Table 5.14 in row "Fix"). Hence, the greatest average benefits that result from increasing break flexibility are realized for the scenarios associated with fixed work templates and variable demand.

Table 5.14: Average benefits from break flexibility

	Benefits (%)			
Work template	$\{S\|V\|T\} \mapsto \{M\|V\|T\}$	$\{M\|X\|T\} \mapsto \{M\|V\|T\}$	$\{M\|V\|T\} \mapsto \{F\|V\|T\}$	$\{F\|V\|T\} \mapsto \{F\|V\|W\}$
Fix and Flex	4.11	0.48	0.38	0.42
Flex	0.92	0.13	0.15	0.17
Fix	3.19	0.34	0.22	0.24

One possible explanation for this result stems from the fact that flexible work templates do a better job at covering demand than fixed work templates in the first step of the MIP-heuristic. For the high demand scenarios, then, there is likely to be more uncovered demand when fixed work templates are used so there is more opportunity to take advantage of break flexibility in the second step.

Table 5.15: Comparison of break flexibility with respect to $\{F|V|W\}$

	Break regulation			
Instance	$\{S\|V\|T\}$	$\{M\|X\|T\}$	$\{M\|V\|T\}$	$\{F\|V\|T\}$
{10, low, SD, fix}	0.15	0.07	0.07	0.00
{50, low, SD, fix}	5.44	1.23	0.00	0.00
{100, low, SD, fix}	0.78	0.78	0.70	0.25
{10, low, SD, flex}	0.00	0.00	0.00	0.00
{50, low, SD, flex}	6.02	3.34	0.98	0.97
{100, low, SD, flex}	1.71	0.41	0.20	0.17
{10, low, VD, fix}	17.61	3.02	1.62	0.00
{50, low, VD, fix}	2.85	1.16	0.45	0.14
{100, low,VD, fix}	4.96	3.56	3.04	3.04
{10, low,VD, flex}	1.09	0.12	0.02	0.00
{50, low,VD, flex}	0.06	0.00	0.00	0.00
{100, low,VD, flex}	2.88	1.22	1.2	0.43
{200, high,SD, fix}	13.04	0.99	0.15	0.04
{250, high,SD, fix}	16.68	1.25	0.69	0.31
{300, high,SD, fix}	8.96	0.95	0.19	0.10
{200, high,SD, flex}	11.43	1.32	1.13	0.01
{250, high,SD, flex}	3.37	1.78	1.65	1.23
{300, high,SD, flex}	0.56	0.02	0.02	0.02
{200, high,VD, fix}	9.62	0.37	0.03	0.00
{250, high,VD, fix}	11.39	3.65	2.07	0.67
{300, high,VD, fix}	4.59	2.56	2.27	1.35
{200, high,VD, flex}	0.25	0.20	0.20	0.01
{250, high,VD, flex}	2.11	1.43	1.43	1.02
{300, high,VD, flex}	3.83	1.23	1.02	0.42

5.7 Discussion and Conclusions

The modeling and analysis in this chapter addresses several unexplored aspects of tour scheduling. The first was the idea of generating tours based on individual work templates, thereby allowing planners to take into account the availability, skill level, and overtime budget of each employee. For a wide range of service organizations, reducing labor costs by exploiting shift and break flexibility can provide a critical competitive advantage. For greater insight into the different break regulations discussed in literature, we introduced a three-field classification scheme for the various models. An accompanying complexity analysis showed that all but the simplest of problems are strongly NP-hard. Our computations confirmed that as break flexibility increases, the corresponding break assignment problems become exponentially more difficult to solve.

Allowing variability in break length, fractionable breaks, and the use of workstretch durations in contrast to predefined break time windows was seen to reduce demand undercoverage and labor cost. The best improvement, though, was obtained by changing from single to multiple breaks per shift. A higher degree of break flexibility turned out to be especially valuable when shift regulations are rather rigid. Also, when the demand fluctuates greatly over the day, break flexibility can compensate for inflexible shift regulations and hence represents an alternative to loosening them up.

Because a compact formulation of the tour scheduling problem could not be solved for large-scale instances, a MIP-decomposition heuristic was proposed that separates the break assignments from tour scheduling. The approach led to promising results with respect to runtimes and solution quality for all instances with up to 300 workers. In contrast, the compact formulation could only solve those instances with up to 50 workers and little shift flexibility.

To solve the break assignment problem, three implicit formulations were investigated. BAP III in Appendix A.7 has the advantage of not requiring any pre- or post-processing and could solve all instances with up to 100 workers in less than a minute. For all larger instances with 200 to 300 workers, only the formulation based on Bechtold and Jacobs [29] led to solution times of less than 30 minutes.

Currently, no procedure exists for solving the integrated problem to optimality in reasonable time. Its practical relevance offers ample justification for more algorithmic development. In the future, we plan to explore exact decomposition approaches to tackle the ThSP and the flexible BAPs in an integrated way. The study of the BAP alone is also of interest, especially for intraday schedule adjustments. During the course of the day, shifts cannot in general be adjusted when a the demand profile changes but it may be possible to reassign the breaks. In this regard, we are now designing a rolling horizon framework for calibrating the value of flexible breaks in short-term disruption scenarios.

Chapter 6

Conclusion

6.1 Summary and conclusions

In this dissertation, we studied the optimization of baggage handling and ground handling at airports. The first part of this dissertation was focused on baggage handling at hub-airports. Chapter 2 gave a structured overview of the baggage handling process covering the four main baggage streams check-in, outbound, transfer and inbound baggage. The complexity of all processes were proven to be NP-complete. An in-depth survey structured past work and classified relevant solution methods. With the GASP model formulation, we presented a generic and novel model formulation representing the basic mathematical structure of each main baggage process. The aim was 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 presented model formulations can be used by researchers and practitioner to obtain first results when examining the baggage handling at airports. For the check-in baggage stream, our model formulation can be seen as a generalization of optimization models found in literature. For the three other baggage handling streams, we presented novel model formulations.

In chapter 3, we considered the planning and scheduling of inbound baggage which is picked up by passengers at 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 turned out to be NP-hard, we proposed a hybrid heuristic combining GRASP with a GFLS and path-relinking. We integrated a bi-criteria objective function, minimizing passengers' waiting time as well as leveling the utilization of baggage carousels. To efficiently generate new solutions with respect to different weightings of the objectives, we developed multi-objective path-relinking. We demonstrated 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 compared the results of the mathematical model with the solutions of the hybrid heuristic and the solutions provided by the airport. The proposed algorithm reduces baggage peaks at the baggage carousels by 38% and waiting times of passengers by 11%. All computational results are based on an extensive simluation incorporating real world data.

In the second part of this dissertation, we shifted our focus towards ground handling and the scheduling of ground staff. In chapter 4, we put the focus on cyclic rostering problems introducing flexible cyclic rostering, a novel approach to schedule personnel in a cyclic fashion derived from practice and intended to overcome the inflexibility of traditional cyclic rostering. With M-SCRP, a multi-stage stochastic programming formulation of the problem was presented. As it turned out that M-SCRP includes an exponential number of NP-hard problems, we presented 2-SCRP, converting the multi-stage problem into a two-stage model approximation. To further speed up computation, with R-SCRP, a second approximation was presented that relates to how shifts are chosen in the second stage of 2-SCRP. All formulations presented can also be applied to traditional cyclic rostering, for which, to the best of our knowledge, no stochastic approaches exist in literature. Moreover, we could show that R-SCRP is equivalent to M-SCRP for traditional cyclic rostering. In a computational study, we compared traditional cyclic rostering with flexible cyclic rostering testing the different model approximations. Therefore, data from airport ground handling was used and altered with different variations of demand fluctuation. It turned out that flexible cyclic rostering could overcome the inflexiblity of cyclic rostering by a large extend. Considering the undercoverage resulting from the generated staff schedules, an improvement of up to 24.96% could be achieved. On the other hand, the study also showed that incorporating stochastic information could significantly increase the quality of cyclic rosters with improvements of up to 8.1%. Furthermore, results indicated that R-SCRP is a promising approximation for 2-SCRP with all gaps between both model formulations being 0.36% or less while average computing times were reduced from more than 8000 seconds for 2-SCRP to less than 50 seconds for R-SCRP.

Chapter 5 examined the complexity of assigning multiple breaks to shifts in the context of large-scale tour scheduling. A MIP model was presented that includes shift and days-off scheduling along with break assignments for a multi-skilled workforce. To achieve tractability, a two-stage decomposition procedure was proposed that separates the tour scheduling problem (TShP) from the break assignment problem (BAP). The former MIP was first solved to determine the shifts and days off for the workforce that minimize labor and shortages costs over the planning horizon. The results were then used as input to a second MIP that optimally places the breaks to minimize the costs of working hours and uncovered periods. Three implicit BAP formulations were investigated. To better understand the literature and the models previously developed, a 3-field break classification scheme was introduced. The first field characterizes the number of breaks permitted per shift, the second specifies whether the length of the breaks is fixed or variable, and the third limits their position in a shift. A complexity analysis of the resulting 12 BAPs along with a few special cases was then undertaken. Most were shown to be strongly NP-hard. Computations were presented for a wide variety of scenarios for both the TShP and the BAP using data provided by a European airport ground handler company. In all, over 500 instances were investigated using high and low demand fluctuation curves and the various break and shift flexibility options. The results indicated that increasing flexibility in break regulations can make a significant difference in coverage, but the degree depends on the underlying structure of the demand curve as well as on the types of shifts permitted. Instances with the most flexible shift and break regulations reduced undercoverage by up to 16.68% compared to the most common scenarios in which shifts are limited to a single lunch break.

6.2 Future research

Our investigation of literature on optimization of baggage handling processes revealed that despite its high relevance in airport operations practice there is still only few literature on the optimization of baggage handling. Further, the planning of baggage handling processes can be further improved by taking the personnel situation into account as personnel costs are one of the biggest cost drivers. As an example, we work on an integration of check-in counter planning and personnel scheduling.

For inbound baggage handling the results of our survey showed a correlation between passengers' satisfaction and passengers' waiting times and walking distance for the planning used at an airport. We are currently undertaking a follow-up survey at the same airport to show the improvement of passengers' satisfaction following the implementation of our algorithm. We hope to obtain further practical insight into the benefits of using OR in baggage handling processes.

In chapter 4, we saw that the value of stochastic information increases when there are less options to adapt to demand fluctuations at a later stage. Therefore, an integration of stochastic information in tour scheduling problems seems promising. To further speed up 2-stage model formulations like 2-SCRP for flexible cyclic rostering, decomposition approaches can be used as an alternative to the approximation method we proposed. Thereby, particular attention must be paid to the integrality of the decision variables.

Chapter 5 showed the potential improvements that can be made by exploiting flexible break regulations. In a rolling horizon framework, we want to further investige how different break regulations can be exploited when breaks can be planned dynamically over the course of a day. Finally, to the best of our knowledge, there exists no solution approach to solve the tour scheduling problem with complex break regulations like fractionable breaks. We try to tackle the problem for an anonymous workforce with a branch-and-price algorithm.

Appendix A

Appendix

A.1 Linearization of IBHP

$$\text{Minimize } f^{\text{lin}}(\mathbf{x}) = \lambda \cdot \sum_{1 \leq k \leq K} \sum_{c \in \mathcal{C}} \sum_{t \in \mathcal{T}} p_k^{\text{u}} \cdot z_{k,c,t}^{\text{u}} + (1 - \lambda) \cdot \cdots \tag{1”}$$

$$\text{subject to } (40) - (54), (56) - (59)$$

$$\frac{1}{K_c^{\text{cb}}} \sum_{l \in \mathcal{L}} \sum_{e \in \mathcal{E}_c} \sum_{S_{l,e}^{\text{es}} \leq s \leq S_{l,e}^{\text{ls}}} \Phi_{l,e,c,s,t} \cdot x_{l,e,c,s} - \sum_{1 \leq k \leq K} u_k \cdot z_{k,c,t}^{\text{u}} \leq 0 \qquad \forall\, c \in \mathcal{C}, t \in \mathcal{T} \tag{55’}$$

$$\sum_{1 \leq k \leq K} z_{k,c,t}^{\text{u}} \leq 1 \qquad \forall\, c \in \mathcal{C}, t \in \mathcal{T} \tag{146}$$

$$z_{k,c,t}^{\text{u}} \in \{0, 1\} \qquad \forall\, 1 \leq k \leq K, c \in \mathcal{C}, t \in \mathcal{T} \tag{147}$$

A.2 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-$\mathbf{R^2}$**	**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).

A.3 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 3.2 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). Afterwards, 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.4 Pseudo code

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 pseudo code

Algorithm 3 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§3.4.2)
    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 3.4.2)
    else
        DeactivateNeighborhood // Deactivate current neighborhood (see §3.4.2)
        UpdateFeaturePenalties // Update features' penalties (see §3.4.2)
    end if
end while
```

A.5 Compact mixed integer program

The ground handler staffing problem (GHSP) is to minimize the total cost of the workforce plus the cost for undercoverage for a given planning horizon. The compact MIP (CMIP) below determines shifts with fractionable breaks for each day in the week and creates a duty roster for each worker such that as much demand as possible is covered in each period for the different skill levels.

Variables

$b^{\text{start}}_{r,w,d,t} = 1$, if sub-break r for worker w starts in period t on day d, 0 otherwise

$b^{\text{end}}_{r,w,d,t} = 1$, if sub-break r for worker w ends in period t on day d, 0 otherwise

$b^{\text{last}}_{w,d,t} = 1$, if the last sub-break for worker w ends in period t on day d, 0 otherwise

CMIP

$$\text{minimize} \sum_{q\in\mathcal{Q}}\sum_{d\in\mathcal{D}}\sum_{t\in\mathcal{T}}\left(\sum_{w\in\mathcal{W}} c^{\text{work}}\cdot z_{w,q,d,t} + c^{\text{uc}}\cdot y^{\text{uc}}_{q,d,t}\right) \tag{148}$$

subject to (119) − (129)

$$\sum_{q'\in\mathcal{Q}:q'\geq q}\sum_{w\in\mathcal{W}(q')} z_{w,q,d,t} + y^{+}_{q,d,t} - y^{-}_{q,d,t} = K_{q,d,t} \quad \forall\, d\in\mathcal{D}, t\in\mathcal{T}, q\in\mathcal{Q} \tag{149}$$

$$\begin{aligned}\sum_{m\in\mathcal{M}(d,p_w)}\sum_{l\in\mathcal{T}^{\text{shS}}:l\leq t}&(y_{m,w,d,l}-\\ &\left(\sum_{r=1}^{B^{\max}}\left(\sum_{l\in\mathcal{T}^{\text{brS}}(r):l\geq t} b^{\text{start}}_{r,w,d,l} - \sum_{l\in\mathcal{T}^{\text{brE}}(r):l\geq t} b^{\text{end}}_{r,w,d,l}\right)+\right.\\ &\left.\sum_{l\in\mathcal{T}^{\text{shE}}:l\geq t} y^{\text{end}}_{w,d,l}\right) = z_{w,q,d,t} \quad \forall\, w\in\mathcal{W}, d\in\mathcal{D}(p_w), t\in\mathcal{T}\end{aligned} \tag{150}$$

$$\sum_{q\in\mathcal{Q}} z_{w,q,d,t} \leq 1 \quad \forall\, w\in\mathcal{W}, d\in\mathcal{D}(p_w), t\in\mathcal{T} \tag{151}$$

Break regulations

$$\sum_{r=1}^{B^{\max}}\sum_{t\in\mathcal{T}^{\text{brS}}(r)} b^{\text{start}}_{r,w,d,t} \geq \sum_{t\in\mathcal{T}^{\text{brS}}(1)} b_{1,w,d,t}\cdot B^{\min} \quad \forall\, w\in\mathcal{W}, d\in\mathcal{D}(p_w) \tag{152}$$

$$\begin{aligned}\Delta^{\text{maxBF}} \geq &\left(\sum_{t\in\mathcal{T}^{\text{brS}}(1)} t\cdot b^{\text{start}}_{1,w,d,t}-\right.\\ &\left.\sum_{m\in\mathcal{M}(d,p_w)}\sum_{t\in\mathcal{T}^{\text{shS}}(m)}\left(t-\Delta^{\text{minBF}}\right)\cdot y_{m,w,d,t}\right) \geq 0 \quad \forall\, w\in\mathcal{W}, d\in\mathcal{D}(p_w)\end{aligned} \tag{153}$$

$$\begin{aligned}\Delta^{\text{maxBD}}_r \geq &\sum_{t\in\mathcal{T}^{\text{brE}}(r)} t\cdot b^{\text{end}}_{r,w,d,t} - \sum_{t\in\mathcal{T}^{\text{brS}}(r)} t\cdot b^{\text{start}}_{r,w,d,t} \geq\\ &\Delta^{\text{minBD}}_r\cdot\sum_{t\in\mathcal{T}^{\text{brS}}(r)} b^{\text{start}}_{r,w,d,t} \quad \forall\, w\in\mathcal{W}, d\in\mathcal{D}(p_w), 1\leq r\leq B^{\max}\end{aligned} \tag{154}$$

$$\Delta^{\text{maxBD}} \geq \sum_{r=1}^{B^{\max}_p}\left(\sum_{t\in\mathcal{T}^{\text{brE}}(r)} t\cdot b^{\text{end}}_{r,w,d,t} - \sum_{\mathcal{T}^{\text{brS}}(r)} t\cdot b^{\text{start}}_{r,w,d,t}\right) \geq$$

$$\Delta^{\text{minBD}} \cdot \sum_{m \in \mathcal{M}(d,p_w)} \sum_{t \in \mathcal{T}^{\text{shS}}(m)} y_{m,w,d,t} \quad \forall\, w \in \mathcal{W}, d \in \mathcal{D}(p_w) \tag{155}$$

$$\Delta_r^{\text{maxBW}} \geq \sum_{t \in \mathcal{T}_m^{\text{brS}}(r)} t \cdot b_{r+1,w,d,t}^{\text{start}} - \sum_{t \in \mathcal{T}^{\text{brE}}(r)} t \cdot b_{r,w,d,t}^{\text{end}} \geq \Delta_r^{\text{minBW}} \cdot \sum_{t \in \mathcal{T}^{\text{brS}}(r)} b_{r+1,w,d,t}^{\text{start}} \qquad \begin{array}{l}\forall\, w \in \mathcal{W}, d \in \mathcal{D}(p_w),\\ 1 \leq r < B^{\max}\end{array} \tag{156}$$

$$\sum_{t \in \mathcal{T}^{\text{shE}}} y_{w,d,t}^{\text{end}} - \Delta^{\text{minBL}} \geq \left(\sum_{t \in \mathcal{T}^{\text{brE}}} t \cdot b_{w,d,t}^{\text{last}} \right) \geq \sum_{t \in \mathcal{T}^{\text{shE}}} y_{w,d,t}^{\text{end}} - \Delta^{\text{maxBL}} \quad \forall\, w \in \mathcal{W}, d \in \mathcal{D} \tag{157}$$

$$\sum_{t \in \mathcal{T}^{\text{brE}}} t \cdot b_{w,d,t}^{\text{last}} \geq \left(\sum_{t \in \mathcal{T}^{\text{brE}}(r)} t \cdot b_{r,w,d,t}^{\text{end}} \right) \quad \forall\, w \in \mathcal{W}, d \in \mathcal{D}, B^{\min} \leq r \leq B^{\max} \tag{158}$$

$$\sum_{t \in \mathcal{T}^{\text{brE}} \setminus \{0\}} (t-1) \cdot b_{w,d,t}^{\text{last}} \leq \sum_{t \in \mathcal{T}^{\text{brE}}(r)} t \cdot b_{r,w,d,t}^{\text{end}} + \sum_{t \in \mathcal{T}^{\text{brS}}(r)} b_{r+1,w,d,t}^{\text{start}} \cdot T \quad \forall\, w \in \mathcal{W}, d \in \mathcal{D}, B^{\min} \leq r < B^{\max} \tag{159}$$

$$\sum_{t \in \mathcal{T}^{\text{brE}} \setminus \{0\}} (t-1) \cdot b_{w,d,t}^{\text{last}} \leq \sum_{t \in \mathcal{T}^{\text{brE}}(B^{\max})} t \cdot b_{B^{\max},s,t}^{\text{end}} + \left(1 - \sum_{t \in \mathcal{T}^{\text{brS}}(B^{\max})} b_{B^{\max},w,d,t}^{\text{start}} \right) \cdot T \quad \forall\, w \in \mathcal{W}, d \in \mathcal{D} \tag{160}$$

$$\sum_{t \in \mathcal{T}^{\text{brE}} \setminus \{0\}} b_{w,d,t}^{\text{last}} \leq 1 \quad \forall\, w \in \mathcal{W}, d \in \mathcal{D} \tag{161}$$

$$\sum_{t \in \mathcal{T}^{\text{brS}}(r-1)} b_{r-1,w,d,t}^{\text{start}} \geq \sum_{t \in \mathcal{T}^{\text{brS}}(r)} b_{r,w,d,t}^{\text{start}} \quad \forall\, w \in \mathcal{W}, d \in \mathcal{D}(p_w), 1 < r \leq B^{\max} \tag{162}$$

$$\sum_{t \in \mathcal{T}^{\text{brS}}(r)} b_{r,w,d,t}^{\text{start}} \leq 1 \wedge \sum_{t \in \mathcal{T}^{\text{brE}}(r)} b_{r,w,d,t}^{\text{end}} \leq 1 \quad \forall\, w \in \mathcal{W}, d \in \mathcal{D}(p_w), 1 \leq r \leq B^{\max} \tag{163}$$

$$\sum_{r=1}^{B^{\max}} \sum_{t \in \mathcal{T}^{\text{brS}}(r)} b_{r,w,d,t}^{\text{start}} = \sum_{r=1}^{B^{\max}} \sum_{t \in \mathcal{T}^{\text{brE}}(r)} b_{r,w,d,t}^{\text{end}} \quad \forall\, w \in \mathcal{W}, d \in \mathcal{D}(p_w) \tag{164}$$

$$z_{w,q,d,t} \in \{0,1\} \quad \forall\, w \in \mathcal{W}, d \in \mathcal{D}(p_w), t \in \mathcal{T} \tag{165}$$

$$b_{r,w,d,t}^{\text{start}}, b_{r,w,d,t'}^{\text{end}}, b_{w,d,t'}^{\text{last}} \in \{0,1\} \quad \forall\, w \in \mathcal{W}, d \in \mathcal{D}(p_w), t \in \mathcal{T}^{\text{brS}}, t' \in \mathcal{T}^{\text{brE}} 1 \leq r \leq B^{\max} \tag{166}$$

The objective function (148) along with the constraints (149)–(151) are similar to (115) and (116)–(118), respectively, but now all skill levels are included. Constraints (152)–(164) describe the break regulations. With respect to B1, for each worker w at least $B^{\min}$ sub-breaks per shift are set in constraints (152), while the minimum and maximum number of periods from the start of the shift to the first sub-break, as indicated in B2, are set in constraints (153)). The length of each sub-break, as restricted by B3, and the overall break length, as restricted by B4, are bounded by constraints (154) and (155), respectively. The minimal and maximal bandwidth between two sub-breaks, B5, is set in constraints (156). The distance of the last sub-break to the end of the shift, B6, is modeled in constraints (157). The decision variables $b^{\text{last}}_{w,t,d}$, which control the latest ending time over all sub-breaks assigned to worker w on day d are determined by (158)– (161).

The last set of constraints (162)–(164) define general modeling requirements. Logically, the r-th sub-break can only be started if its immediate predecessor sub-break $r-1$ has been started [constraints (162)]. Also, the unique starting period of each sub-break is defined in constraints (163) and, due to constraints (164), a sub-break can only end if it has been started.

If we consider breaks with time windows in the GHSP, i.e., BAP$\{\cdot|\cdot|T\}$, the above model has to be slightly modified. Let Δ^{minTW} and Δ^{maxTW} be the minimal and maximal number of periods of the r-th sub-break, respectively, for $r > 1$ from the shift start. Then we replace the two-sided constraints (156) with

$$\begin{aligned}\Delta^{\text{maxTW}}_r &\geq \sum_{t\in\mathcal{T}^{\text{brS}}(r)} t\cdot b^{\text{start}}_{r,w,d,t} - \sum_{m\in\mathcal{M}(d,p_w)}\ \sum_{t\in\mathcal{T}^{\text{shS}}(m)} t\cdot y_{m,w,d,t}\Delta^{\text{minTW}}_r \\ &\geq \sum_{m\in\mathcal{M}(d,p_w)} \Delta^{\text{minTW}}_r \cdot y_{m,w,d,t} \quad \forall\ w\in\mathcal{W}, d\in\mathcal{D}(p_w), 1<r\leq B^{\max} \qquad (156')\end{aligned}$$

which is analogous to constraints (153) for the first sub-breaks.

A.6 Implicit BAP formulation based on Aykin's approach

For shift scheduling, Rekik et al. [106] present a model with fractionable breaks based on the implicit formulation for multiple breaks first introduced by Aykin [13]. In the following, we show how their approach can be adapted for the tour scheduling problem with a hierarchical workforce as was discussed in § 5.5.2. The additional sets required for the model are listed in Table A.3.

Sets	**Description**
$\mathcal{J}_{s,q}$	set of shift profiles associated with shift s and skill q
$\mathcal{K}_{j(r)}$	set of sub-breaks admissible as r'th sub-break in shift profile j
$\mathcal{Q}$	set of skills

Table A.3: Sets used in implicit break assignment model IBAP2

We make use of the definitions of shift profiles and break profiles in § 5.5.2 and similarly, define a sub-break $k \in \mathcal{K}$ for each break profile, each position in the break profile, each possible starting time, and each skill level. Also, we need a unique sub-break for each skill level since we have to consider the skill of the worker that is taking the break. Instead of assigning sub-breaks explicitly to shifts, a variable $X_{j,k}$ is defined that stores how often sub-break $k \in \mathcal{K}$ is assigned to shift profile $j \in J$.

To obtain a feasible solution with regard to workstretch durations, the feasibility of transportation problems between successive breaks has to be ensured. The following set is used to constructed the problems.

$$\mathcal{K}_{j(r)} = \text{set of sub-breaks admissible as } r\text{-th sub-break in shift profile } j$$

Based on the break starting times, a totally ordered relationship $\prec$ on $\mathcal{K}_{j(r)}$ is defined. Then, a transportation problem $T(\mathcal{K}_{j(r)}, \mathcal{K}_{j(r+1)})$ from the nodes in $\mathcal{K}_{j(r)}$ to the nodes in $\mathcal{K}_{j(r+1)}$ can be established for each $j \in \mathcal{J}$ and each $r \in 1, .., B_j - 1$. Nodes $k_1 \in \mathcal{K}_{j(r)}$ and $k_2 \in \mathcal{K}_{j(r+1)}$ are connected if the workstretch duration between their corresponding breaks is within the interval $[\delta^{\text{minBW}}, \delta^{\text{maxBW}}]$. The feasibility of the transportation problem ensures that there are enough successor breaks for all sub-breaks k that are assigned to shift profile j and is guaranteed by a set of forward and backward constraints based on the principles described in **§ 5.5.2**. Also, because there are no pairs of nodes $k_1, k_2 \in \mathcal{K}_{j(r)}$ for which $\mathcal{K}^{\text{suc}}_{k_1} \subset \mathcal{K}^{\text{suc}}_{k_2}$, there is no extraordinary overlap.

In the formulation, we make use of the following additional parameters and variables:

Parameters

$$B_j = \text{number of sub-breaks in shift } j$$

Variables

$X_{j,k}$ = number of workers assigned to shift j that are given sub-break k

Implicit BAP formulation II (IBAP2)

$$\text{Minimize} \sum_{q \in \mathcal{Q}} \sum_{t \in \mathcal{T}} c^{\text{uc}} \cdot y^{-}_{q,t} - \sum_{j \in \mathcal{J}} \sum_{r \in \{1,...,B_j\}} \sum_{k \in \mathcal{K}_{j(r)}} c^{\text{work}} \cdot d_k \cdot X_{j,k} \tag{167}$$

subject to

$$\sum_{\hat{q} \geq q} P_{\hat{q},q,t} + y^{+}_{q,t} - y^{-}_{q,t} = D_{q,d,t} \qquad \forall\, q \in \mathcal{Q}, t \in \mathcal{T} \tag{168}$$

$$\sum_{j \in \mathcal{J}_{s,q}} S_j = h_{s,q} \qquad \forall\, s \in \mathcal{S}, q \in \mathcal{Q} \tag{169}$$

$$\sum_{k \in \mathcal{K}_{j(r)}} X_{j,k} - S_j = 0 \qquad \forall\, j \in \mathcal{J}, r \in \{1, ..., B_j\} \tag{170}$$

$$P_{\hat{q},q,t} \leq l_{\hat{q},q,t} \qquad \forall\, \hat{q}, q \in \mathcal{Q} : q \leq \hat{q}, t \in \mathcal{T} \tag{171}$$

$$\sum_{q \leq \hat{q}} P_{\hat{q},q,t} = \sum_{j \in \mathcal{J}_{\hat{q}}} \sum_{r \in \{1,...,B_j\}} \sum_{k \in \mathcal{K}_{j(r)}} X_{j,k} \cdot \rho_{t,k} \qquad \forall\, \hat{q} \in \mathcal{Q}, t \in \mathcal{T} \tag{172}$$

$$\sum_{k' \in \mathcal{K}^F_{j(r+1)}(k)} X_{j,k'} - \sum_{k' \in \mathcal{K}^F_{j(r)}(k)} X_{j,k'} \geq 0 \qquad \begin{array}{l} \forall\, j \in \mathcal{J}, \\ \quad r \in \{1, ..., B_j - 1\}, \\ \quad k \in \mathcal{K}^{\text{e}}_{j(r+1)} \setminus \left\{k^{\text{e}}_{j(r+1)}\right\} \end{array} \tag{173}$$

$$\sum_{k' \in \mathcal{K}^B_{j(r+1)}(k)} X_{j,k'} - \sum_{k' \in \mathcal{K}^B_{j(r)}(k)} X_{j,k'} \geq 0 \qquad \begin{array}{l} \forall\, j \in \mathcal{J}, \\ \quad r \in \{1, ..., B_j - 1\}, \\ \quad k \in \mathcal{K}^{\text{s}}_{j(r+1)} \setminus \left\{k^{\text{s}}_{j(r+1)}\right\} \end{array} \tag{174}$$

$$S_j, w_{w,t}, P_{\hat{q},q,t} \in \mathbb{Z}_+ \qquad \forall\, j \in \mathcal{J}, \hat{q}, q \in \mathcal{Q} : q \leq \hat{q}, t \in \mathcal{T} \tag{175}$$

$$X_{j,k} \in \mathbb{Z}_+ \qquad \forall\, j \in \mathcal{J}, k \in \bigcup_{r \in \{1,...,B_j\}} \mathcal{K}_{j(r)} \tag{176}$$

Constraints (168) determine the amount of demand for skill level q on day d in period t that cannot be covered, while constraints (169) ensure that all shifts of type s for workers with skill q are connected to a shift profile j. Due to constraints (170), the number of sub-breaks for shift j in each break position $r \in \{1, ..., B_j\}$ corresponds to the number of shifts of type j. Further, there cannot be more workers with skill $\hat{q}$ that take a break from a job requiring skill q in period t then there are workers in the input tour plan with skill $\hat{q}$ carrying out a job in period t that requires skill q. This is enforced by (171). In addition, all breaks taken in period t by workers with skill level $\hat{q}$ must be distributed among jobs requiring at most skill $\hat{q}$. This is enforced by (172). Constraints (173) and (174) respectively are the forward and backward constraints used to ensure that the total number of sub-breaks is correct. Equality constraints for the total sub-break supply and demand are not needed due to constraints (170). Variables are defined in (175) and (176).

A.7 Implicit BAP formulation III

The BAP formulation contained in CMIP is stated below using the following additional notation.

Parameters

S_s^{start} = start time of shift s

S_s^{end} = end time of shift s

C_s^{work} = cost for shift s (without breaks)

Variables

$b_{r,s,t}^{\text{start}} = 1$, if the r-th sub-break for shift s starts at time t, 0 otherwise

$b_{r,s,t}^{\text{end}} = 1$, if the r-th sub-break for shift s ends at time t, 0 otherwise

$b_{s,t}^{\text{last}} = 1$, if the last sub-break for shift s ends at time t, 0 otherwise

$a_{s,t} = 1$, if there is a break in shift s at time t, 0 otherwise

$z_{s,q,t} = 1$, if period t is a working period for shift s that covers skill level q, 0 otherwise

Implicit BAP formulation III (IBAP3)

$$\text{Minimize} \sum_{t \in \mathcal{T}} \left(\sum_{s \in \mathcal{S}} c^{\text{work}} \cdot a_{s,t} + \sum_{q \in \mathcal{Q}} c^{\text{uc}} \cdot y_{q,t}^{-} \right) \tag{177}$$

subject to

$$\sum_{s \in \mathcal{S}(q,t)} a_{s,t} - y_{q,t}^{-} + y_{q,t}^{+} = D_{q,d,t} \qquad \forall\, q \in \mathcal{Q}, t \in \mathcal{T} \tag{178}$$

$$\sum_{r=1}^{B^{\max}} \sum_{t \in \mathcal{T}^{\text{brS}}(r)} b^{\text{start}}_{r,s,t} \geq B^{\min} \qquad \forall\ s \in \mathcal{S}'(d) \tag{179}$$

$$\Delta^{\text{minBF}} \leq \left(\sum_{t \in \mathcal{T}^{\text{brS}}(r)} t \cdot b^{\text{start}}_{1,s,t} - S^{\text{start}}_{s} \right) \leq \Delta^{\text{maxBF}} \qquad \forall\ s \in \mathcal{S}'(d) \tag{180}$$

$$\begin{aligned} \Delta^{\text{minBD}}_{r} \cdot \sum_{t \in \mathcal{T}^{\text{brS}}(r)} b^{\text{start}}_{r,s,t} \leq \sum_{t \in \mathcal{T}^{\text{brE}}(r)} t \cdot b^{\text{end}}_{r,s,t} & \qquad \forall\ s \in \mathcal{S}'(d), \\ - \sum_{t \in \mathcal{T}^{\text{brS}}(r)} t \cdot b^{\text{start}}_{r,s,t} \leq \Delta^{\text{maxBD}}_{r} & \qquad 1 \leq r \leq B^{\max} \end{aligned} \tag{181}$$

$$\Delta^{\text{minBD}} \leq \sum_{r=1}^{B^{\max}} \left(\sum_{t \in \mathcal{T}^{\text{brE}}(r)} t \cdot b^{\text{end}}_{r,s,t} - \sum_{t \in \mathcal{T}^{\text{brS}}(r)} t \cdot b^{\text{start}}_{r,s,t} \right) \leq \Delta^{\text{maxBD}} \qquad \forall\ s \in \mathcal{S}'(d) \tag{182}$$

$$\begin{aligned} \Delta^{\text{minBW}}_{r} \cdot \sum_{t \in \mathcal{T}^{\text{brS}}(r+1)} b^{\text{start}}_{r+1,s,t} \leq \sum_{t \in \mathcal{T}^{\text{brS}}(r+1)} t \cdot b^{\text{start}}_{r+1,s,t} - & \qquad \forall\ s \in \mathcal{S}'(d) \\ \sum_{t \in \mathcal{T}^{\text{brE}}(r)} t \cdot b^{\text{end}}_{r,s,t} \leq \Delta^{\text{maxBW}}_{r} & \qquad 1 \leq r \leq B^{\max} - 1 \end{aligned} \tag{183}$$

$$S^{\text{end}}_{s,d} - \Delta^{\text{minBL}} \geq \left(\sum_{t \in \mathcal{T}^{\text{brE}}} t \cdot b^{\text{last}}_{s,t} \right) \geq S^{\text{end}}_{s} - \Delta^{\text{maxBL}} \qquad \forall\ s \in \mathcal{S}'(d) \tag{184}$$

$$\sum_{t \in \mathcal{T}^{\text{brE}}} t \cdot b^{\text{last}}_{s,t} \geq \left(\sum_{t \in \mathcal{T}^{\text{brE}}(r)} t \cdot b^{\text{end}}_{r,s,t} \right) \qquad \forall\ s \in \mathcal{S}'(d),\ B^{\min} \leq r \leq B^{\max} \tag{185}$$

$$\begin{aligned} \sum_{t \in \mathcal{T}^{\text{brE}}} (t-1) \cdot b^{\text{last}}_{s,t} \leq \sum_{t \in \mathcal{T}^{\text{brE}}(r)} t \cdot b^{\text{end}}_{r,s,t} + & \qquad \forall\ s \in \mathcal{S}'(d) \\ \sum_{t \in \mathcal{T}^{\text{brS}}(r)} b^{\text{start}}_{r+1,s,t} \cdot S^{\text{end}}_{s,d} & \qquad B^{\min} \leq r < B^{\max} \end{aligned} \tag{186}$$

$$\begin{aligned} \sum_{t \in \mathcal{T}^{\text{brE}}} (t-1) \cdot b^{\text{last}}_{s,t} \leq \sum_{t \in \mathcal{T}^{\text{brE}}(B^{\max})} t \cdot b^{\text{end}}_{B^{\max},s,t} + & \qquad \forall\ s \in \mathcal{S}'(d) \\ \left(1 - \sum_{t \in \mathcal{T}^{\text{brS}}(B^{\max})} b^{\text{start}}_{B^{\max},s,t} \right) \cdot S^{\text{end}}_{s,d} & \end{aligned} \tag{187}$$

$$\sum_{t \in \mathcal{T}^{\text{brE}}} b^{\text{last}}_{s,t} = 1 \qquad \forall\ s \in \mathcal{S}'(d) \tag{188}$$

$$\sum_{r=1}^{B^{\max}} \sum_{t \in \mathcal{T}^{\text{brS}}(r)} b^{\text{start}}_{r,s,t} = \sum_{r=1}^{B^{\max}} \sum_{t \in \mathcal{T}^{\text{brE}}(r)} b^{\text{end}}_{r,s,t} \qquad \forall\ s \in \mathcal{S}'(d) \tag{189}$$

$$\sum_{r=1}^{B^{\max}} \left(\sum_{l \in \mathcal{T}^{\text{brS}}(r): l \geq t} b^{\text{start}}_{r,s,l} - \sum_{l \in \mathcal{T}^{\text{brE}}(r): l \geq t} b^{\text{end}}_{r,s,l} \right) = a_{s,t} \qquad \forall\ s \in \mathcal{S}'(d),\ S^{\text{start}}_s \leq t \leq S^{\text{end}}_s \tag{190}$$

$$\sum_{t \in \mathcal{T}^{\text{brS}}(r)} b^{\text{start}}_{r,s,t} \leq 1 \wedge \sum_{t \in \mathcal{T}^{\text{brE}}(r)} b^{\text{end}}_{r,s,t} \leq 1 \qquad \forall\ 1 \leq r \leq B^{\max}\ s \in \mathcal{S}'(d) \tag{191}$$

$$\sum_{t \in \mathcal{T}^{\text{brS}}(r+1)} b^{\text{start}}_{r+1,s,t} \leq \sum_{t \in \mathcal{T}^{\text{brS}}(r)} b^{\text{start}}_{r,s,t} \qquad \forall\ 1 \leq r < B^{\max}\ s \in \mathcal{S}'(d) \tag{192}$$

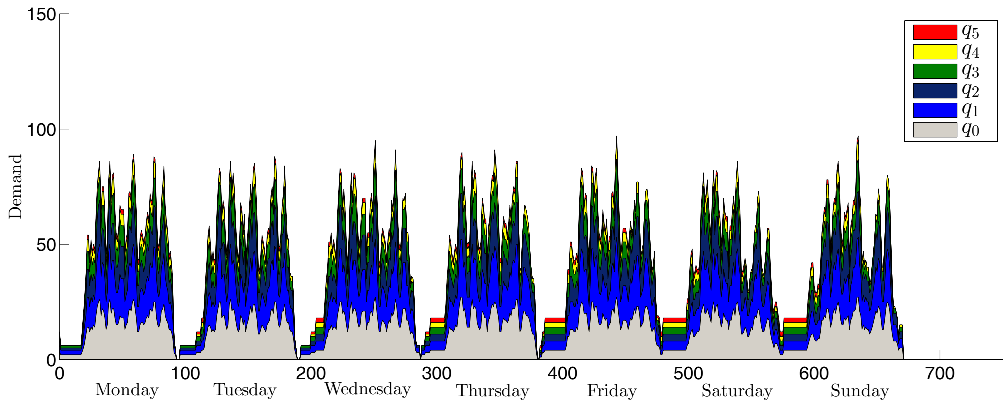
$$b^{\text{start}}_{r,s,t}, b^{\text{end}}_{r,s,t'}, b^{\text{last}}_{s,t''} \in \{0,1\} \qquad \forall\ s \in \mathcal{S}'(d),\ B^{\min} \leq r \leq B^{\max},\ t \in \mathcal{T}^{\text{brS}}(r),\ t' \in \mathcal{T}^{\text{brE}}(r),\ t'' \in \mathcal{T}^{\text{brE}} \tag{193}$$

$$a_{s,t} \in \{0,1\} \qquad \forall\ s \in \mathcal{S}'(d), q \in \mathcal{Q},\ S^{\text{start}}_s \leq t \leq S^{\text{end}}_s \tag{194}$$

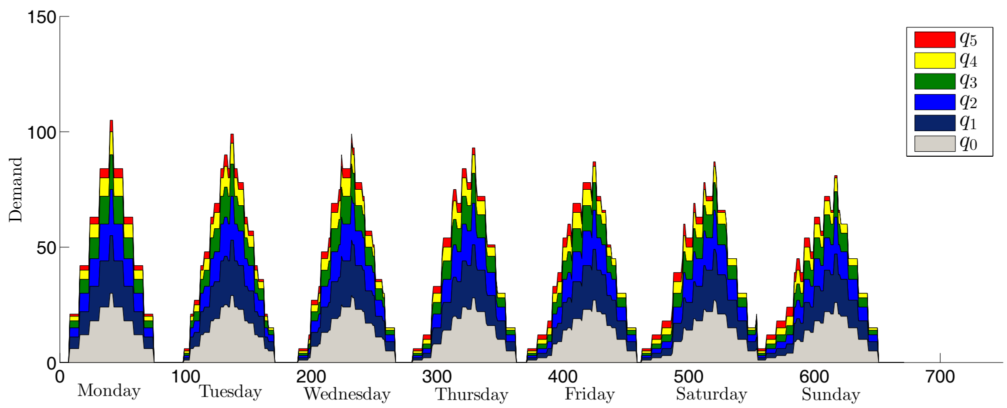
$$y^+_{q,t}, y^-_{q,t} \geq 0 \qquad \forall\ q \in \mathcal{Q},\ S^{\text{start}}_s \leq t \leq S^{\text{end}}_s \tag{195}$$

Constraints (178)–(195) are equivalent to (152)–(164), but now workers' shifts $\mathcal{S}^*(d)$ for each day $d \in \mathcal{D}$ are provided by Algorithm 2.

A.8 Low demand curves



(a) Variable demand



(b) Stable demand

Figure A.1: Variable and stable low demand curves

Appendix B

Abbreviations

App.	appendix
BAP	break assignment problem
BC	baggage carousel
BHS	baggage handling system
BSM	baggage source massage
CCPP	check-in counter planning problem
CMIP	compact mixed integer program
DP	decision problem
EEV	expected result of expected value problem
EV	expected value problem
EVPI	expected value of perfect information
FCFS	first come, first serve
FLS	fast local search
FSP	flight scheduling problem
GASP	generic assignment and scheduling problem
GFLS	guided fast local search
GHSP	ground handler staffing problem
GRASP	greedy randomized adaptive search procedure
HGGLS	hybridization of greedy randomized adaptive search procedure with guided fast local search
IBAP	implicit break assignment
IBHP	inbound baggage handling problem

LP	linear programming
L-IBHP	linearized inbound baggage handling problem
MIP	mixed integer program
MORP	multi-objective path-relinking
M-SCRP	multi-stage stochastic cyclic rostering problem
NP	non-deterministic polynomial
OBHP	outbound baggage handling problem
OR	operations research
PP	pricing problem
PR	path-relinking
RDS	ramp direct service
RMSE	root mean squared error
RR	recourse problem
R-SCRP	relaxed stochastic cyclic rostering problem
Sec.	section
SGASP	simplified generic assignment and scheduling problem
TIF	time indexed formulation
TShP	tour scheduling problem
TU	totally unimodular
ULD	unit load device
VSS	value of the stochastic solution
WS	wait-and-see solution
2-SCRP	two-stage stochastic cyclic rostering problem

Appendix C

Mathematical symbols

General

Sets

$\mathbb{R}$	real numbers
$\mathbb{Z}$	integer numbers
$\mathbb{Z}_+$	non-negative integer numbers

Functions

$\|\mathcal{A}\|$	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
$\mathcal{J}^{\mathbf{BP}}$	items to be packed into a bin
$\mathcal{J}^{\mathbf{MS}}$	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
$S_i^{\mathbf{es}}$	earliest start time for job i
$S_i^{\mathbf{ls}}$	latest start time for job i
$S_i^{\mathbf{e}}$	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
$\lambda^+ g$	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

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

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
$B_i^{\mathbf{tot}}$	total number of inbound bags of flight i
C	number of baggage carousels
$d^{\mathbf{walk}}$	realized walking distance
E	number of infeed station
F	number of flights with inbound baggage
$K_c^{\mathbf{cb}}$	conveyor belt capacity of baggage carousel c
$K_c^{\mathbf{di}}$	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
$p_k^{\mathbf{u}}$	penalty for violating threshold value u_k
$p_c^{\mathbf{cap}}$	penalty for the capacity violation at baggage carousel c
$p_{i,l}^{\mathbf{ls}}$	penalty for the violation of the latest infeed restriction for trip l
$S_i^{\mathbf{ob}}$	on block time of flight i
$S_{l,e}^{\mathbf{es}}$	earliest start time for the infeed of trip l at infeed station e
$S_{l,e}^{\mathbf{ls}}$	latest start time for the infeed of trip l at infeed station e
$t_{l,e,c,s}^{\mathbf{first}}$	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
$u_c^{\mathbf{cap}}$	utility for the capacity violation of carousel c
$u_{i,l}^{\mathbf{ls}}$	utility for the violation of the latest infeed restriction for trip l
$\mathbf{x}$	solution representation
$\mathbf{x}^{\mathbf{best}}$	best solution found in the constructive phase
$\mathbf{x}^{\mathbf{dec}}$	solution for a decrease weight in the multi-objective path-relinking
$\mathbf{x}^{\mathbf{inc}}$	solution for a increase weight in the multi-objective path-relinking
$\mathbf{x}^{\mathbf{init}}$	initial solution for the multi-objective path-relinking
$\mathbf{x}^{\mathbf{ls}}$	current local search solution
$\mathbf{x}^{\mathbf{lsb}}$	current local search iteration best solution
$\mathbf{x}^{\mathbf{next}}$	next solution in the neighborhood of solution $\mathbf{x}^{\mathrm{ls}}$
$\mathbf{x}^{\mathbf{new}}$	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
$\Delta^{\mathbf{claim}}_{i,e,c,s}$	baggage claim duration for flight i, if infeed station e and baggage carousel c are selected and the infeed process starts at time s
$\Delta^{\mathbf{dur}}_{e,c,}$	duration to transfer a bag from infeed station e to baggage carousel c
$\Delta^{\mathbf{inf}}_{l}$	infeed duration of trip l
$\Delta^{\mathbf{lag}}_{l,h,e}$	minimum time lag between trip l and h at infeed station e
$\Delta^{\mathbf{next}}$	duration planned in advanced in a rolling planning framework
$\Delta^{\mathbf{pl}}$	duration to place containers on a baggage tug
$\Delta^{\mathbf{time}}$	computing time of an optimization algorithm in a rolling planning framework
$\Delta^{\mathbf{trip}}_{i,e}$	duration to drive from flight i's parking position to infeed station e
$\Delta^{\mathbf{wait}}$	realized waiting time
λ	weight for the utilization
$\lambda^{\mathbf{dec}}$	decreased weight for the utilization for the multi-objective path-relinking
$\lambda^{\mathbf{inc}}$	increased weight for the utilization for the multi-objective path-relinking
$\lambda^{\mathbf{init}}$	initial weight for the utilization for the multi-objective path-relinking
$\mu^{\mathbf{w}}$	baggage loading rate of workers at an infeed station
$\mu^{\mathbf{p}}_{i}$	arrival rate of passengers of flight i
$\Psi^{\mathbf{wait}}_{l,e,c,s}$	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
$\Psi^{\mathbf{walk}}_{i,c}$	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
$h^{\mathbf{cstr}}(\mathbf{x})$	cost function in the constructive phase

$I_c^{\mathbf{cap}}(\mathbf{x})$	1, if the number of flights which can be assigned to baggage carousel c is exceeded
$I_c^{\mathbf{ls}}(\mathbf{x})$	1,if the infeed of flight i's trip l starts later than the latest possible infeed start time
$p_{l,e,c,s}^{\mathbf{wait}}(t)$	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

Flexible Cyclic Rostering in the Service Industry

Sets

$\mathcal{D}$	set of days per week
$\mathcal{G}$	set of groups
$\mathcal{P}$	set of shift types
$\mathcal{S}$	set of shift
$\mathcal{S}(p)$	set of shifts associated with shift type p
$\mathcal{T}$	set of time periods per day
$\mathcal{W}$	set of weeks in the planning horizon
Ω	sample space for demand scenarios

Parameters

$a_{s,t}$	number of workers per group if shift s covers period t, 0 otherwise
$d^{\mathbf{min}}$	minimum number of working days per week allowed
$d^{\mathbf{max}}$	maximum number of working days per week allowed

f_s	earliest period covered by shift s
f_p	earliest period covered by any shift associated with shift type p
l_s	number of periods covered by shift s
$h^{\mathbf{max}}$	maximum number of working hours per week allowed
$h^{\mathbf{min}}$	minimum number of working hours per week allowed
$n^{\mathbf{max}}$	maximum number of consecutive working days allowed
$\mathbf{D}_{d,t}$	random variable for the demand on day d in period t
D	number of days
G	number of groups
P	number of shift types
S	number of shifts
T	number of time periods per day
W	number of days in the planning horizon

Variables

$u_{d,t}(\omega)$	undercoverage in period t of day d for random event ω
$u_{w,d,t}(\omega)$	undercoverage in period t of day d of week w for random event ω
v_g	1, if at least one weekend shift type is assigned for group g, 0 otherwise
$x_{p,g,d}$	1, if shift type p is assigned to group g on day d, 0 otherwise
$x_{s,g,d}$	1, if shift s is assigned to day d for group g, 0 otherwise
$y_{g,s,d}(\omega)$	1, if shift s is assigned to group g on day d for

	random event ω, 0 otherwise
$y_{g,s,w,d}(\omega)$	1, if shift s is assigned to group g on day d in week w for random event ω, 0 otherwise

Functions

$\mathbb{E}$	expected value
$\widetilde{\mathbf{D}}(\omega)$	demand realization vector for random event ω
$\widetilde{D}_{d,t}(\omega)$	demand on day d in period t for random event ω
$P(\widetilde{\mathbf{D}})$	propability mass function of $\widetilde{\mathbf{D}}$

The flexible break assignment problem for large tour scheduling problems with an application to airport ground handlers

Sets

$\mathcal{B}$	break patterns
$\mathcal{B}_{(\beta,r,q)}$	sub-breaks associated with break profile β, position r and skill q
$\mathcal{B}(s)$	break patterns for shift s
$\mathcal{BP}$	break profiles
$\mathcal{D}$	planning days
$\mathcal{D}(p)$	planning days in template p
$\mathcal{J}$	shift profiles, defined as a unique combination of starting time, length, and admissible break profile
$\mathcal{J}_{(\beta,q)}$	shift profiles associated with break profile β and skill q
$\mathcal{J}_k$	shifts that can be assigned sub-break k
$\mathcal{J}_q$	shifts associated with skill q

$\mathcal{J}_{s,q}$	shifts associated with shift pattern s and skill q
$\mathcal{K}$	sub-breaks, one for each break profile, each position in the break profile, each possible starting time and each skill level
$\mathcal{K}_{j(r)}$	sub-breaks admissible as r'th sub-break in shift j
$\mathcal{M}$	shift types
$\mathcal{M}(p)$	eligable shift types in template p
$\mathcal{M}(d,p)$	eligable shift types in template p on day d
$\mathcal{P}$	weekly templates
$\mathcal{Q}$	skills
$\mathcal{Q}(q^+, q_k)$	skills that are larger or equal to skill q^+ and lower or equal to skill q_k
$\mathcal{S}$	shifts
$\mathcal{S}(d)$	shifts on day d
$\mathcal{S}(q,t)$	shifts covering demand for skill q in period t
$\mathcal{T}$	planning periods per day
$\mathcal{T}^{\mathbf{brE}}$	break ending times for all sub-break
$\mathcal{T}^{\mathbf{brE}}(r)$	break ending times of the r-th sub-break
$\mathcal{T}^{\mathbf{brS}}(r)$	break starting times for the r-th sub-break
$\mathcal{T}^{\mathbf{shE}}$	shift ending times
$\mathcal{T}^{\mathbf{shS}}(m)$	shift starting times for the shift type m
$\mathcal{W}$	workers
$\mathcal{W}(q)$	workers with skill q

Parameters

$a_{s,b,q,t}$	1, if break pattern b in shift s covers demand for skill q in period t, 0 otherwise
B_β	number of sub-breaks of break profile β
B_j	number of sub-breaks in shift j
$B^{\mathbf{min}}$	minimum number of sub-breaks

$B^{\mathbf{max}}$	maximum number of sub-breaks
$C_s^{\mathbf{work}}$	cost for shift s (without breaks)
$c^{\mathbf{uc}}$	unit cost for undercoverage
$c^{\mathbf{work}}$	unit cost for a worker when on duty
$c_{s,b}^{\mathbf{work}}$	cost for a worker assigned shift s with break pattern b
d_k	length of break k
$D_{q,t}$	amount of over-coverage of demand for skill q in period t (can be negative)
$h_{s,q}$	number of workers assigned to shift pattern s having skill q
$K^{\mathbf{shift}}$	the number of maximum allowed shift starting times per week for shift type m
$K_{q,d,t}$	demand for skill level q on day d in time period t
$l_{\hat{q},q,t}$	number of shifts for workers with skill $\hat{q}$ covering demand for jobs requring skill q in period t
O^-	lower bound for overtime bank
O^+	upper bound for overtime bank
$S_s^{\mathbf{end}}$	end time of shift s
$S_s^{\mathbf{start}}$	start time of shift s
$\Delta^{\mathbf{maxBD}}$	maximum overall break duration
$\Delta^{\mathbf{maxBF}}$	latest period in which the first sub-break can start
$\Delta^{\mathbf{maxBL}}$	laximum workstretch after the last sub-break
$\Delta_r^{\mathbf{maxBW}}$	maximum workstretch duration after sub-break r
$\Delta^{\mathbf{minBF}}$	earliest period in which the first sub-break can start
$\Delta_r^{\mathbf{minBD}}$	minimum duration of sub-break r
$\Delta_r^{\mathbf{minBD}}$	maximum duration of sub-break r
$\Delta^{\mathbf{minBD}}$	minimum overall break duration
$\Delta^{\mathbf{minBL}}$	minimum workstretch after the last sub-break
$\Delta_r^{\mathbf{minBW}}$	minimum workstretch duration after sub-break r

$\Delta^{\mathbf{bw}}$	maximum allowed bandwidth for shift starting times of the same shift type within a tour
$\rho_{t,k}$	1, if break k covers period t, 0 otherwise

Variables

$a_{s,t}$	1, if there is a break in shift s at time t, 0 otherwise
$b^{\mathbf{end}}_{r,s,t}$	1, if the r-th sub-break for shift s ends at time t, 0 otherwise
$b^{\mathbf{end}}_{r,w,d,t}$	1 , if break r of worker w ends in period t on day d, and 0 otherwise
$b^{\mathbf{last}}_{s,t}$	1, if the last sub-break for shift s ends at time t, 0 otherwise
$b^{\mathbf{last}}_{w,d,t}$	1, if the last break of worker w ends in period t on day d, and 0 otherwise
$b^{\mathbf{start}}_{r,s,t}$	1, if the r-th sub-break for shift s starts at time t, 0 otherwise
$b^{\mathbf{start}}_{r,w,d,t}$	1, if break r of worker w starts in period t on day d, and 0 otherwise
E_k	number of workers that are given sub-break k
O_w	overtime bank of worker w
$P_{\hat{q},q,t}$	number of workers having skill $\hat{q}$ who are given a break in period t from a job requiring skill q
S_j	number of workers assigned to shift profile $j \in J$
$w_{w,t}$	1, if a shift of worker w starts in period t, 0 otherwise
$X_{j,k}$	number of workers assigned to shift j that are given sub-break k
$y_{m,w,d,t}$	1, if shift of type m for worker w starts in period t on day d, 0 otherwise
$y^{+}_{q,d,t}$	shortfall in demand for skill level q in period t

	on day d
$y^-_{q,t}$	amount of uncovered demand for skill q in period t after break assignments
$y^+_{q,t}$	amount of excess demand for skill q in period t after break assignments
$y^{\mathbf{end}}_{w,d,t}$	1, if shift for worker w ends in period t on day d, 0 otherwise
$z_{s,b}$	1, if break pattern b is assigned to shift s, 0 otherwise
$z_{s,q,t}$	1, if period t is a working period for shift s that covers skill level q, 0 otherwise
$z_{w,q,d,t}$	1, if worker w is active during period t on day d at skill level q, 0 otherwise

Functions

$\prec$	total order relation

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] I. Addou and F. Soumis. Bechtold-jacobs generalized model for shift scheduling with extraordinary overlap. *Annals of Operations Research*, 155(1):177–205, 2007.

[4] Airbus. Global market forecast. `ttp://www.airbus.com/company/market/forecast/`, 2014.

[5] H. K. Alfares. Operator staffing and scheduling for an IT-help call centre. *European Journal of Industrial Engineering*, 1(4):414–430, 2007.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] E. I. Ásgeirsson. Bridging the gap between self schedules and feasible schedules in staff scheduling. *Annals of Operations Research*, pages 1–19, 2012.

[11] N. Ashford, H. Stanton and C. Moore. *Airport operations*. McGraw Hill (New York), 1997.

[12] A. N. Avramidis, W. Chan, M. Gendreau, P. E'Cuyer and O. Pisacane. Optimizing daily agent scheduling in a multiskill call center. *European Journal of Operational Research*, 200(3):822–832, 2010.

[13] T. Aykin. Optimal shift scheduling with multiple break windows. *Management Science*, 42(4):591–602, 1996.

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

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

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

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

[18] K. R. Baker. Workforce allocation in cyclical scheduling problems: A survey. *Operational Research Quarterly*, pages 155–167, 1976.

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

[20] N. Balakrishnan and R. T. Wong. A network model for the rotating workforce scheduling problem. *Networks*, 20(1):25–42, 1990.

[21] 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.

[22] J. F. Bard. Staff scheduling in high volume service facilities with downgrading. *IIE Transaction on Scheduling and Logistics*, 36:985–997, 2004.

[23] J. F. Bard. Selecting the appropriate input data set when configuring a permanent workforce. *Computers & Industrial Engineering*, 47(4):371–389, 2004.

[24] J. F. Bard and L. Wan. Workforce design with movement restrictions between workstation groups. *Manufacturing & Service Operations Management*, 10(1):24–42, 2008.

[25] J. F. Bard, D. P. Morton and Y. M. Wang. Workforce planning at usps mail processing and distribution centers using stochastic optimization. *Annals of Operations Research*, 155(1):51–78, 2007.

[26] 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.

[27] J. J. Bartholdi III. A guaranteed-accuracy round-off algorithm for cyclic scheduling and set covering. *Operations Research*, 29(3):501–510, 1981.

[28] J. J. Bartholdi III, J. B. Orlin and H. D. Ratliff. Cyclic scheduling via integer programs with circular ones. *Operations Research*, 28(5):1074–1085, 1980.

[29] S. Bechtold and L. Jacobs. Implicit modeling of flexible break assignments in optimal shift scheduling. *Management Science*, 36:1339–1351, 1990.

[30] A. Beer, J. Gartner, N. Musliu, W. Schafhauser and W. Slany. An ai-based break-scheduling system for supervisory personnel. *Intelligent Systems, IEEE*, 25(2):60–73, 2010.

[31] J. R. Birge and F. Louveaux. *Introduction to stochastic programming.* Springer Science & Business Media, 2011.

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

[33] 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.

[34] J. O. Brunner and J. F. Bard. Flexible weekly tour scheduling for postal service workers using a branch and price. *Journal of Scheduling*, 16(1):129–149, 2013.

[35] J. O. Brunner and R. Stolletz. Stabilized branch and price with dynamic parameter updating for discontinuous tour scheduling. *Computers* & *Operations Research*, 44: 137–145, 2014.

[36] 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.

[37] M. J. Brusco and L. W. Jacobs. Personnel tour scheduling when starting-time restrictions are present. *Management Science*, 44(4):534–547, 1998.

[38] M. J. Brusco and L. W. Jacobs. Optimal models for meal-break and start-time flexibility in continuous tour scheduling. *Management Science*, 46(12):1630–1641, 2000.

[39] G. M. Campbell. A two-stage stochastic program for scheduling and allocating cross-trained workers. *Journal of the Operational Research Society*, 62(6):1038–1047, 2011.

[40] T. Çecik and O. Günlük. Reformulating linear programs with transportation constraints - with applications to workforce scheduling. *Naval Research Logistics*, 51 (2):275–296, 2004.

[41] S. Chu. Generating, scheduling and rostering of shift crew-duties: Applications at the hong kong international airport. *European Journal of Operational Research*, 177: 1764–1778, 2007.

[42] 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.

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

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

[45] 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.

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

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

[48] 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.

[49] G. B. Dantzig. Letter to the editor-a comment on Edie's "traffic delays at toll booths". *Journal of the Operations Research Society of America*, 2(3):339–341, 1954.

[50] 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.

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

[52] 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.

[53] D. Dowling, M. Krishnamoorthy, H. Mackenzie and D. Sier. Staff rostering at a large international airport. *Annals of Operations Research*, 72:125–147, 1997.

[54] A. T. Ernst, H. Jiang, M. Krishnamoorthy and D. Sier. Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, 153:3–27, 2004.

[55] 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.

[56] G. Felici and C. Gentile. A polyhedral approach for the staff rostering problem. *Management Science*, 50:381–393, 2004.

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

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

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

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

[61] 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.

[62] 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.

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

[64] 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.

[65] 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.

[66] 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.

[67] 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.

[68] J. Herbers. *Models and Algorithms for Ground Staff Scheduling on Airports*. PhD thesis, RWTH Aachen, 2005.

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

[70] IATA. Iata airline industry forecast 2013-2017. `http://www.iata.org/pressroom/pr/pages/2013-12-10-01.aspx`, 2013.

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

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

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

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

[75] 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.

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

[77] 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.

[78] F. Kiermaier, M. Frey and J. F. Bard. The flexible break assignment problem for large tour scheduling problems with an application to airport ground handlers. *Working paper*, 2015.

[79] Y.-H. Kuo, J. Leung and C. Yano. Scheduling of multi-skilled staff across multiple locations. *Production and Operations Management*, 23(4):626–644, 2014.

[80] 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.

[81] G. Laporte, Y. Nobert and J. Biron. Rotating schedules. *European Journal of the*, 4(1):24–30, 1980.

[82] H. C. Lau. On the complexity of manpower shift scheduling. *Computers & Operations Research*, 23(1):93–102, 1996.

[83] 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.

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

[85] 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.

[86] F. V. Louveaux and R. Schultz. Stochastic integer programming. *Handbooks in operations research and management science*, 10:213–266, 2003.

[87] R. R. Love and J. M. Hoey. Management science improves fast-food operations. *Interfaces*, 20(2):21–29, 1990.

[88] R. Lusby, A. Dohn, T. M. Range and J. Larsen. A column generation-based heuristic for rostering with work patterns. *Journal of the Operational Research Society*, 63 (2):261–277, 2011.

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

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

[91] H. H. Millar and M. Kiragu. Cyclic and non-cyclic scheduling of 12 h shift nurses by network programming. *European Journal of the Operational Research*, 104(3): 582–592, 1998.

[92] S. Mirrazavi and H. Beringer. A web-based workforce management system for Sainsburys Supermarkets Ltd. *Annals of Operations Research*, 155(1):437–457, 2007.

[93] J. Müller, G. Orak, E. Petkov and S. Schulz. Restructuring of the european ground handling market after the eu market liberalization. Technical report, School of Economics, Gap Team Berlin and HUB.

[94] N. Musliu. Heuristic methods for automatic rotating workforce scheduling. *International Journal of Computational Intelligence Research*, 2(4):309–326, 2006.

[95] N. Musliu, J. Gärtner and W. Slany. Efficient generation of rotating workforce schedules. *Discrete Applied Mathematics*, 118(1):85–98, 2002.

[96] G. Nemhauser and A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, New York, 1988.

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

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

[99] H. Ni and H. Abeledo. A branch-and-price approach for large-scale employee tour scheduling problems. *Annals of Operations Research*, 155(1):167–176, 2007.

[100] 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.

[101] 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.

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

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

[104] H. Purnomo and J. F. Bard. Cyclic preference scheduling for nurses using branch and price. *Naval Research Logistics*, 54(2):200–220, 2007.

[105] M. Rekik, J.-F. Cordeau and F. Soumis. Using Benders' decomposition to implicitly model tour scheduling. *Annals of Operations Research*, 128:111–133, 2004.

[106] M. Rekik, J.-F. Cordeau and F. Soumis. Implicit shift scheduling with multiple breaks and work stretch duration restrictions. *Journal of Scheduling*, 13:49–75, 2010.

[107] 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.

[108] T. R. Robbins and T. P. Harrison. A stochastic programming model for scheduling call centers with global service level agreements. *European Journal of Operational Research*, 207(3):1608–1619, 2010.

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

[110] 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.

[111] M. Rocha, J. F. Oliveira and M. A. Carravilla. Cyclic staff scheduling: Optimization models for some real-life problems. *Journal of Scheduling*, 16(2):231–242, 2013.

[112] A. Rong. Monthly tour scheduling models with mixed skills considering weekend off requirements. *Computers & Industrial Engineering*, 59(2):334–343, 2010.

[113] E. S. Rosenbloom and N. F. Goertzen. Cyclic nurse scheduling. *European Journal of Operational Research*, 31(1):19–23, 1987.

[114] S. Sen. Algorithms for stochastic mixed-integer programming models. *Handbooks in operations research and management science*, 12:515–558, 2005.

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

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

[117] 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.

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

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

[120] 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.

[121] 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.

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

[123] G. M. Thompson and M. E. Pullman. Scheduling workforce relief breaks in advance versus in real-time. *European Journal of Operational Research*, 181(1):139–155, 2007.

[124] J. M. Tien and A. Kamiyama. On manpower scheduling algorithms. *SIAM Review*, 24:275–287, 1982.

[125] S. Topaloglu and I. Ozkarahan. An implicit goal programming model for the tour scheduling problem considering the employee work preferences. *Annals of Operations Research*, 128(1-4):135–158, 2004.

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

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

[128] 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.

[129] J. Van den Bergh, J. Beliën, P. De Bruecker, E. Demeulemeester and L. De Boeck. Personnel scheduling: A literature review. *European Journal of Operational Research*, 226(3):367–385, 2013.

[130] 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.

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

[132] 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.

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

[134] 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.

[135] 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.

[136] 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.

[137] 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.

[138] 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.

[139] X. Zhu and H. D. Sherali. Two-stage workforce planning under demand fluctuations and uncertainty. *Journal of the Operational Research Society*, 60(1):94–103, 2009.