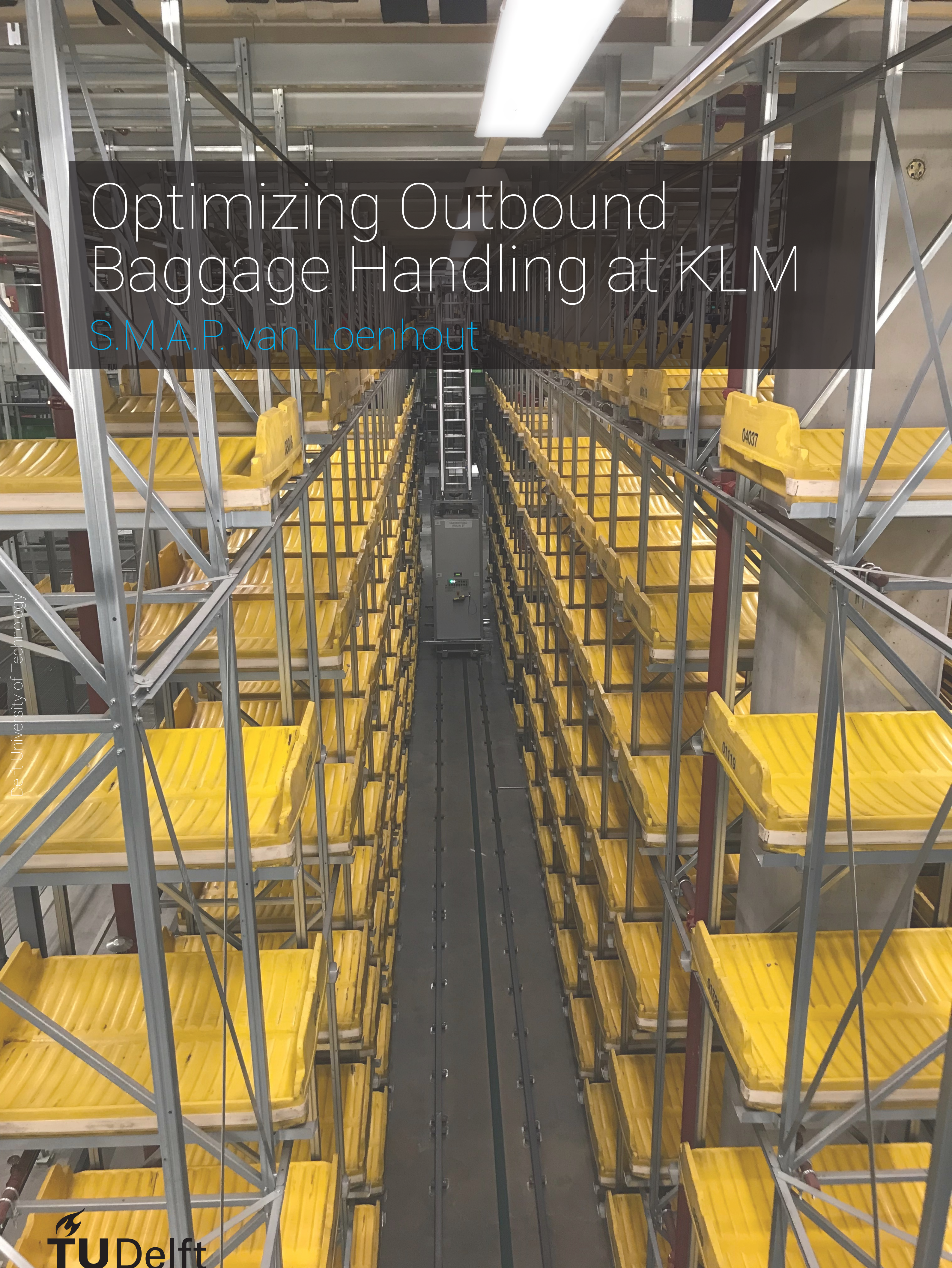
Optimizing Outbound Baggage Handling at KLM

S.M.A.P. van Loenhout



Delft University of Technology

Optimizing Outbound Baggage Handling at KLM

by

S.M.A.P. van Loenhout

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday January 30, 2019 at 2:00 PM.

Project duration:	May 1, 2018 – January 30, 2019	
Thesis committee:	Prof. dr. ir. K. I. Aardal,	TU Delft
	Dr. ir. J.T. van Essen,	TU Delft, supervisor
	Dr. C. Kraaikamp,	TU Delft
	Ir. G. De Wit,	ORTEC, supervisor

This thesis is confidential and cannot be made public until January 30, 2021.

TU Delft

ORTEC
OPTIMIZE YOUR WORLD

Abstract

In order to handle the increase in baggage units, KLM needs to optimize the outbound baggage handling process. This study focuses on accurately modeling the outbound baggage handling process while minimizing the number of employees working on this process. The results are based upon data on the outbound baggage handling process at KLM in 2018. In order to create a feasible lateral planning, the satisfaction of the demand and the rules on positioning certain flights have been implemented as hard constraints. ORTEC already created a basic model for assigning flights to baggage halls. However, this model did not lead to the expected improvements and did not model reality very well, and therefore, some extra extensions are needed. The first extension focuses on planning the outbound baggage handling process on a more detailed level, by assigning flights to specific laterals and baggage sections. Several methods are considered to construct this more detailed planning: a mixed integer programming formulation, a hierarchical solution method and column generation. For the mixed integer programming formulation, several speed-ups and valid inequalities are suggested, as first tests showed that the LP-relaxation provides a weak lower bound on the optimal solution. The MIP and hierarchical solution method resulted to outperform the other methods for small data instances created. Because the flight schedule does not result in feasible solutions for the first extension, delaying lateral opening times is allowed under certain circumstances, in the second extension of this research. This results in more flexibility and feasible solutions can be obtained for the flight schedules. The hierarchical solution method resulted in the best feasible objective value within a certain time limit. Driving tasks, which need to be fulfilled in order to bring the baggage to the corresponding departing aircraft, are added in the last extension of the research, such that the number of drivers can be minimized together with the employees working in the baggage hall. The lateral plannings constructed in this research meet all the constraints which are based on the daily operation, such that the reality is accurately modelled.

Key words: Outbound baggage handling, mixed integer programming, valid inequalities, column generation, symmetry breaking constraints

Preface

This thesis has been submitted as the final requirement to obtain the degree Master of Science in Applied Mathematics at Delft University of Technology. The research was conducted in collaboration with ORTEC BV and KLM Royal Dutch Airlines in the period from May 2018 until January 2019.

Throughout this project I have been provided with a great deal of freedom, that has allowed me to grow both scientifically as well as personally. This work would never have been possible without the valuable advice and support of many people that I would like to thank in this preface.

First of all, I would like to thank Theresia van Essen for her guidance during my graduation project. You have trusted me and provided me with a great deal of freedom and helped me to overcome every obstacle, both personally and intellectually. Thank you for all the pleasant meetings and patience.

Secondly, I want to thank Guido de Wit for offering me an interesting, practical, social and challenging project. From the first meeting at KLM and the voice message on my first working day, until this day, you were an amazing supervisor.

Many thanks to the personnel of KLM, especially to Robbert van Amerongen, for all the meetings and information needed for my thesis. Jankees Holst, thank you for the tour in the baggage system and introducing me to the people within KLM that could help me with my thesis.

Furthermore, I want to thank Karen Aardal and Cor Kraaikamp for being a member of my graduation committee. Karen, I owe my internship at MIT to you and I am really grateful for this unique opportunity. Cor, thank you for all the meetings, funny chats, personal backups and achievements we gained together with the OCW throughout the past years of my studies.

Moreover, I would like to express my gratitude to some people who supported me the last 7.5 years and made my time as a student in Delft an amazing and unforgettable time. My friends for all the fun times, the needed relaxation and for accepting everyone just the way they are. My board members, for the most memorable and amazing year, all the constructive feedback and for always standing by my side when I needed it the most. Furthermore, I would like to thank Rik, Joost, Sharon and all the other EWI personnel who made my year as a Commissioner of Education and trusted me with all the responsibilities. Thanks for all the chats, advice and for never underestimating me.

Last, but definitely not least, I would like to thank three of the most important people in my life. Dear Daan, I am grateful that we can share many professional experiences, which makes it easy to discuss and share memories. Writing our thesis together made it more fun and I would like to thank you for your endless patience, positiveness and never doubting me. Most importantly, I would like to thank my mother and grandmother for their unconditional love. You raised me to the person I am today, taught me to be myself and to stand up for myself. You are always there for me, gave me many wonderfull memories and pulled me through though situations. Hopefully we will have many more healthy years together.

S.M.A.P. van Loenhout
Delft, January 2019

Contents

I

Introduction

1

Research Motivation and Outline

Air transportation has become a popular means of transport with a world-wide growth of around 3.7 billion passengers since 1970 [4]. A growth is not only obtained at the global level: according to the summary of the baggage logistics for KLM provided by Aarts [1], an increase of 32% in the total number of passengers at Schiphol Airport for both KLM and Delta was obtained in 2017 compared to 2011. Next to passengers, we can also find an increase in the amount of baggage units, even though the average number of bags per passenger decreased (the collifactor). Aarts [1] showed that the collifactor decreased from 1.03 in 2011 down to 0.92 in 2017. However, the total number of bags handled by KLM increased (from 13.2 million in 2011 up to 15 million in 2017) due to the stronger increase in passengers. In addition, [1] also states that an increase in the collifactor is expected and has been obtained in 2018.

Unfortunately, the number of personnel working at baggage handling is not expanding in the same way, which causes an increase in workload among the employees. This increase in workload puts pressure on the employees which makes them unsatisfied about the working conditions and this also causes some friction between employees and employer. This has for example, led to employees from several baggage handling companies to go on strike lately, at Schiphol Airport in the spring of 2018 [23]. They demanded a lower workload and more healthy workforce schedules.

Although the baggage handlers of KLM did not go on a strike last year, they have expressed concerns about the workload becoming higher. Hence, KLM faces a situation in which more baggage units have to be handled, without increasing the workload of the employees. For example, this could be done by hiring more people or by optimizing the current baggage handling process while using the current shift capacity of the employees, i.e. the number of baggage that can be handled by an employee per time slot. Because the latter is more cost efficient, this research was started in order to optimize the baggage handling process while minimizing the workload among employees. This research focuses on the outgoing baggage, which needs to be transported to an aircraft. The baggage that needs to go to the reclaim area is left outside the scope of this research.

This first chapter introduces all the aspects that provide the context and motivation of this thesis project. First, some background information on the involved companies is provided in Section 1.1. Second, the baggage process is described in more detail in Section 1.2, and third, the planning process is described in Section 1.3. Next, the research motivation and research question is outlined in Section 1.4. Finally, the outline of this thesis is described in Section 1.5.

1.1. Background

This thesis was written under supervision of ORTEC BV in Zoetermeer, The Netherlands. It is part of the master Applied Mathematics at the Delft University of Technology, in relation of the track Computational Science and Engineering with the specialization Optimization. KLM gave rise to the problem discussed in this thesis, as they are one of the key accounts of ORTEC.

1.1.1. ORTEC

ORTEC is one of the largest providers of advanced optimization software and analytics solutions. ORTEC was founded in 1981 by five Econometrics students. ORTEC has over 2000 customers worldwide, more than 900 employees and several offices in Europe, America and the Pacific Region. ORTEC turns complex challenges into easy-to-use solutions for clients among different industries, namely: Transportation, Retail, Oil, Gas & Chemicals, Food & Beverages and Consumer Goods.

ORTEC has three divisions and this project is part of the ORTEC Consulting division. This division offers tailor made and off-the-shelf analytics and optimization models and tools.

1.1.2. KLM

KLM Royal Dutch Airlines is the oldest still operating airline in the world and was founded in 1919. KLM is the Dutch abbreviation for Royal Airline. KLM operates flights worldwide with over 200 aircraft and employing 32,000 people from the Amsterdam basis. In 2017, KLM expanded its network and reached a total of 165 destinations worldwide. The main basis for KLM is Amsterdam Airport Schiphol and this thesis is based on the outbound baggage process at this airport.

KLM is a member of the SkyTeam, which is a major global network of 20 different airlines. At Amsterdam Schiphol Airport, KLM handles the baggage of most of its partners, together with the baggage of Transavia flights. Transavia is a low-cost airline which is wholly owned subsidiary by KLM.

1.2. Baggage Process

An overview of the outbound baggage handling process is illustrated in Figure E.1. A baggage handling system (BHS) is a conveyor belt network that runs through the entire airport and automatically transports the baggage from one location to the other. Outbound baggage is received from either check-in passengers or transfer passengers. Transfer baggage is unloaded from an arriving aircraft and transported with baggage tugs to infeed-stations, where the baggage is transferred into the BHS. Check-in baggage enters the BHS via the check-in counters and needs to be screened first. The baggage system automatically transports the baggage to laterals in the baggage halls, from which they are unloaded on baggage tugs. A lateral is the conveyor belt on which baggage is dropped when it exits the BHS. The baggage tugs are driven to the departing aircraft and the baggage is loaded into the aircraft. In case the baggage enters the BHS before the lateral is open, the bags are directed to a central storage system, where they are stored until baggage handling begins.

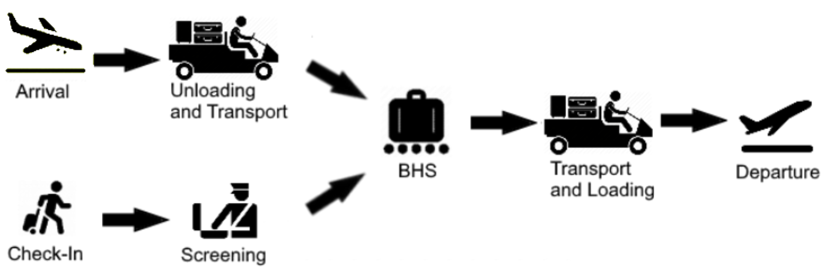


Figure 1.1: Graphical representation of the outbound baggage loading process

KLM operates their outbound baggage process in three halls, namely the D, E and South hall. The laterals to which baggage from the flights that are being handled by KLM can be assigned are located in baggage hall D, E or South. These baggage halls are exclusively used by KLM. A map of the entire BHS of Schiphol is illustrated in Figure 1.2 and for each baggage hall, a more detailed map can be found in Appendix C. Each hall has several baggage sections which contains a set of laterals. Employees are assigned to baggage sections and each baggage section has a section lead.

Different types of laterals are used in the halls, which is mostly caused by just expanding the halls over the years in which new types of laterals were invented. Also, some laterals are more easily used for bigger aircraft such as a street, which are two laterals above each other. A picture of such a street is given in Figure C.4 in

Appendix C. Robots are used in some halls, which automatically load the tugs. A robot is illustrated in Figure C.5 in Appendix C. Furthermore, carousels are used which can handle three flights at the same time. However, these three flights may require at most two laterals. Figure C.6 in Appendix C illustrates carousels.

Figure 1.2: Map of the baggage handling system at Schiphol Airport

1.3. Planning Process

This section provides more information about the process of creating a planning for the flight assignments to laterals. First, the required input is described with all the needed extra information. Secondly, the current planning process is described.

1.3.1. Required Input

In order to create a baggage lateral schedule for a specific day, the flight schedule for that day is required. The flight schedule provides the flight number, the scheduled time of departure, the scheduled departure gate and the estimated number of bags. An overview of the flight characteristics can be obtained, using the flight number. An example of a data instance in the flight characteristics overview is provided in Table 1.1. There are four different flight categories, namely ICA (Intercontinental Aviation), Europe, Transavia and Commuter. WIBO (Wide Body), NABO (Narrow Body) and Commuter are the three aircraft types which are used by KLM.

Flight	Category	Airplane Type	Open DE	Close DE	Open South	Close South	Required
KL0895	ICA	WIBO	-210	0	-120	0	2

Table 1.1: Example of one data instance of the flight characteristics overview

‘Open DE’ and ‘Open South’ denotes the amount of minutes that the lateral needs to be open *before* (hence the negative sign) the scheduled time of departure for hall D or E and South, respectively. ‘Close DE’ and ‘Close South’ denotes the closing time before the scheduled time of departure for a flight for the D or E hall and South hall, respectively. The value belonging to ’Required’ denotes how many laterals are required for the flight.

For each combination of flight category, aircraft type, baggage hall and scheduled time of departure, a drop off profile is given. A drop off profile indicates what percentage of all bags of a flight are expected to ‘drop’ during certain time slots before the flight. This profile depends on the baggage hall and the time of the day. As mentioned before, each baggage hall contains multiple baggage sections to which employees are assigned and each baggage section contains a set of laterals. For all the laterals, it is denoted to which baggage section and which hall it belongs as well as its lateral capacity.

Furthermore, some settings are needed in order to know how many employees are needed to fulfill a certain schedule. Employees are planned to work in groups and one of the settings is the number of employees that work in a group. Another setting is how often a working shift may start, for example every quarter or every hour.

1.3.2. Current Planning Process

Roughly six months in advance, the planning for the upcoming period is made by hand by two Baggage Flow Controllers (BFCs). KLM currently operates with a Summer and a Winter period: the Summer period starts at the last Sunday of March and Winter starts the last Sunday of October. Hence, there are seven months of Summer and five months of Winter in the planning of KLM.

The manual planning is solely focused on feasibility. The BFCs are very experienced and use their field knowledge in order to create a planning that matches reality. When the planning is feasible, it is send to the Tactical Planning department. This department will manually enter the planning in a program called PlanControl. PlanControl calculates the expected amount of personnel and other interesting KPI’s such as total costs and baggage in the BHS per time slot for example. Decisions are mostly based upon these calculations, such as the number of people that needs to be hired. The amount of employees that are needed to fulfill the schedule are assumed to work full-time. Employees are scheduled to work 8.5 hours such that the lunch break is included in the shift. These lunch break and additional coffee breaks are not scheduled beforehand, but depend on the actual work force. The section lead will determine when it is time to have a lunch or coffee break. When the planning is created, a different department will turn the working shifts into a work roster for the employees. A full-time shift is then split into multiple part time shifts in case there are people who work part time.

Although feasible, the lateral planning created by the BFCs is not robust and does not take into account unexpected events, such as flight delays, different number of baggage units etc. These unexpected delays are handled on short notice by the BFCs which are located in a control center near the BHS. They have access to all the information and cameras of the system, the flight schedule and all the other required information. Using this information, they are able to act on the real time situation and to make the necessary changes in the planning.

1.4. Research Motivation

The previously discussed process is not very efficient, as the planning is constructed manually and the BFCs mainly focus on creating a feasible schedule and do not try to optimize the schedule. The planning can be optimized by solving a mathematical model, which determines to which laterals the baggage of a certain flight must be transported. As mentioned before, the BFCs use their knowledge to create a schedule which fits reality very well. Therefore, it is important that the model is able to accurately model reality.

ORTEC already created a model which decides to which hall the baggage of a certain flight is assigned. However, because of certain assumptions and local optimization, this model did not lead to the expected improvements. It also did not model reality very well.

Three main extensions are examined in this research in order to model reality and to find a good and feasible planning. The first one is to extend the model to lateral level, which means that the flights will be assigned to a specific lateral instead of to just a baggage hall. This gives a more detailed and precise schedule than the current one on baggage hall level. Secondly, the lateral opening times must be delayed for some of the flights in order to obtain a feasible schedule. Namely, with all the flights and predefined opening and closing times, it is often not possible to find a feasible schedule. Lastly, the model should be extended such that the part of driving the baggage to the aircraft is included and the number of drivers can be minimized as well. This will result in an overall minimization of the number of employees working on the outbound baggage process.

Based on the previous mentioned motivation, the primary goal of this thesis is to answer the following research question:

"How can reality be accurately modelled to minimize the number of employees working on the outbound baggage handling process?"

1.5. Thesis Outline

Part I of this thesis, the introduction, will continue with a review of the related scientific literature in Chapter 2. In Chapter 3, the current ORTEC model is explained and the mathematical notation used for the problem is laid out.

Part II focuses on the extension on lateral level for which several solution methods are developed. A mixed integer programming formulation is described in Chapter 4. Subsequently, Chapter 5 describes small adjustments to the MIP of Chapter 4 and two different ways of formulating the MIP which could reduce the computation time. Valid inequalities are introduced and added to the MIP in Chapter 6 as the LP-relaxation does not provide a good lower bound on the optimal solution. The size of the problem is decreased significantly by the hierarchical solution method described in Chapter 7, which splits the total problem into two phases. This is supposed to decrease the computation time as the problems are simplified versions of the total problem. Column generation applied to this problem is explained in Chapter 8. The data provided by KLM is described in Chapter 9 and the evaluation criteria on which the different models and created schedules are evaluated are given in Chapter 10. The results of the methods described in this Part II are presented in Chapter 11.

Delaying lateral opening times in order to obtain feasible solutions has been made possible in Part III. Chapter 12 describes how and when the lateral opening times may be delayed. The methods used in this part of the research are chosen based on the outcomes of the results of Part II. Results for this part of the research are evaluated in Chapter 13.

Part IV introduces the driving tasks such that both the number of drivers and employees loading the baggage from the laterals into the tugs can be minimized. Chapter 14 describes how the driving tasks are included and which solution methods are chosen to be used based on the results of Part II and Part III. Chapter 15 summarizes the results of this part of the research.

Last, Part V summarizes the computational results of this research, provides the conclusions and discussion, which are found in Chapters 16 and 17, respectively.

2
Literature Review

Outbound baggage handling processes have been studied by various researchers. In this chapter, a distinction is made between studies that focus on different aspects of the problem and those that use different methods. Section 2.1 focuses on the variations of the problem and Section 2.2 describes the different solution approaches that have been employed throughout these variations. Section 2.3 describes what the contribution of this thesis is to the existing literature.

2.1. Variations of the Problem

Various papers have studied variations of the problem of baggage handling. Below, we discuss some of these studies, published between 2006 and 2018. The term baggage station is used to denote the conveyor belt on which the baggage is dropped, as most studies do not describe the type of conveyor belt.

Frey [13] handles outbound baggage handling by focusing on assigning departing flights to one handling facility. Their main goal is to obtain a balanced workload across all carousels, in order to avoid workload peaks. The workload is thereby defined as the number of bags at a time slot dropped on a conveyor belt. Other goals of the problem are to minimize the distance between the handling facility and the parking position of the departing airplane and to meet the preferences of ground handlers. Their goals are related to the ones of this thesis. However, this thesis also takes the shift capacity of working groups and the actual working shifts of employees into account. Hence, the number of working groups can be minimized instead of just the workload over the carousels. Another difference is that Frey [13] does not have the requirements for assigning flights to laterals, because they are using carousels.

A robust model for outgoing flights is created by Huang et al. [17] which assigns baggage to baggage stations. They take into account the possible flight changes due to mechanical problems and weather changes. The robust model consists of a two-stage program, in which an initial schedule is created one day in advance and unexpected changes can be dealt with on the spot while remaining as close to the original schedule as possible. Their objective is to minimize the expected number of unassigned flights and the changes between the planned and actual assignments. Unassigned flights are handled manually by the ground crew, whereas the other flights are handled by the BHS. In comparison to the problem studied in this thesis, the planning does not take into account employees. Huang et al. [17] do not mention how they ensure that enough ground crew is available to handle the aircraft.

Ascó et al. [5] handles the outbound baggage in the case where flights are already scheduled and allocated to gates. Remarkable is the main goal, which is to maximize the number of assigned flights, because there are not enough stations when it is too busy. Their approach includes a constraint which states that a flight must be assigned to at most one baggage station. Furthermore, the distance between the baggage station and the aircraft of the corresponding flight is minimized. Robustness is maximized in the third part of the objective by maximizing the gaps between two consecutive assignments. To avoid disturbances, they apply a reasonable minimum buffer time on a lateral in between two consecutive flights. Last, the preferred baggage halls are considered and matched as much as possible. This part of the objective is correlated to the distance minimization. Unassigned flights are not permitted in this thesis, so the main goal is different in this research. However, minimizing the distance between the baggage station and the aircraft matches one of the goals of this thesis. Buffer time between two consecutive flights is preferable, but at the moment not a priority for KLM and therefore it is not included in this thesis.

Another interesting objective criteria is used in the research of Abdelghany et al. [3]. They prefer baggage with the same destinations to be located on the same baggage station. When a bag has not made it to the flight in time, it can stay on the facility in case it is allowed to go on a next flight to the same destination. KLM currently stores left over baggage in the buffer until a lateral to the same destination is opened and the bag is allowed to go on this flight.

2.2. Solution Approaches

Ascó et al. [5] use constructive algorithms which start ordering flights followed by selecting a baggage station in order to solve the problem as described in Section 2.1. Two different flight ordering methods are considered: ordering by starting time and ordering by departure time. Baggage station sorting is done on a First In First Out (FIFO), Last In First Out (LIFO) or closest-to-gate basis. The FIFO method selects the earliest available baggage station. The LIFO method selects the available baggage station which is most recently used and this method results in a fair distribution among the systems, reduces the number of baggage sorting systems in use and minimizes the idle time between flights on the system. The closest-to-aircraft method chooses an available baggage station which is closest to the aircraft such that it meets the preference. In case the distance to the aircraft is the same, LIFO is used.

Frey et al. [15] created a model that minimizes the workload peaks over the entire system. A time-indexed mathematical programming formulation was made for the planning of the outbound baggage. In order to solve practical problem instances and to reduce the computation times, they proposed an innovative decomposition procedure in combination with a column generation scheme. The symmetry effect in the time-indexed formulation is significantly reduced by the decomposition procedure. Acceleration techniques for the primal and pricing problem are proposed for improving the column generation approach.

Another article by Frey et al. [14] modelled the outbound baggage process as a resource-constrained project scheduling problem (RCPSP). They also used a decomposition heuristic because of the high complexity. The objective in every aspect of the heuristic is to minimize the maximal workload. First, the flights are assigned to baggage stations, and subsequently, the starting times of baggage handling are scheduled for an assignment vector of working stations to flights. Next, the working stations are actually assigned to the flights, which can be formulated as a minimum cost flow problem. Cuts are introduced when a schedule violates the requirements.

A heuristic approach based on the Greedy Randomized Adaptive Search Procedure (GRASP) and a decomposition approach are created by Barth et al. [7]. In the GRASP formulation, flights are assigned to handling facilities every iteration in a randomized, greedy way. Afterwards, the solution is improved with a local search method. Within the decomposition approach, sub problems are created, which can be modelled as MIPs. Seven different objective criteria are introduced. The GRASP approach was shown to be much faster, however it was found suitable only for cases encompassing one or two of the objective criteria.

2.3. Contribution to Existing Literature

Previously mentioned related studies (see Section 2.1 and 2.2), either use carousels or do not mention the specifics of the handling facilities in their study. Therefore, the rules that apply for KLM related to the assignment of flights to specific laterals are not described in any of the reviewed papers. Furthermore, some of the mentioned studies have set balancing the workload as a goal, although they have not taken the real shift durations and starting times into account. Notably, in this thesis, peaks in workload could be a result of the optimal planning as it is covered by having overlap in the end and start time of consecutive shifts. These new rules of assigning flights to laterals and the objective which minimizes the number of employees working on a shift in order to divide the workload fairly across the sections are contributions to the existing literature.

3

Problem Description

This chapter elaborates upon the problem description as introduced in Chapter 1. The basic problem as was handled in the previous research done by ORTEC is described in Section 3.1. Subsequently, Section 3.2 formulates the currently existing Mixed Integer Program (MIP) developed by ORTEC for the basic problem. Shortcoming of and extensions for the current model are described in Section 3.3.

3.1. Basic Problem

The basic model determines to which baggage hall the baggage of a specific flight should be allocated. The objective when constructing such a planning is to minimize the amount of working hours, which is done by minimizing the total amount of groups working at each time slot in all the halls. The amount of groups that need to work at a certain time slot within the baggage hall is based on the estimated number of baggage units and how many baggage units an employee can handle per time slot. A group contains at least one employee and each group consists of an equal number of employees. These employees place the bags from the lateral onto the baggage tugs. The Baggage Handling System (BHS) itself does not need to be modelled, nor does the basic model assign baggage to a specific lateral. It only assigns baggage to a baggage hall.

Based on the scheduled departure time of a flight and at what time the laterals must open and close in specific halls, it can be determined during which time interval the laterals are open. The earliest lateral opening time within a hall indicates the opening of the hall, whereas the latest closing time of a lateral indicates the closing of the hall. The planning horizon of the model starts at the earliest opening time of a hall until the latest closing time across all halls. The planning horizon is divided into time slots of length δ, which are given by the set T.

The BHS runs through different baggage halls to which baggage and employees are assigned. The set of baggage halls is given by the set H. For each hall $h \in H$, a certain amount of laterals is available per time slot $t \in T$, which is indicated by the lateral capacity lc_{ht}.

The set of flights is given by the set F. Each day, the baggage of $|F|$ different flights needs to be assigned to exactly one baggage hall. The flight category and aircraft type are given for each flight. For each combination of aircraft type, flight type and baggage hall, a drop off profile is given for the baggage. A drop off profile indicates when the baggage starts ‘dropping’ into the BHS and what percentage of the total amount of bags will drop each time slot until fifteen minutes before the scheduled departure time. Using this drop off profile, the number of baggage units for flight $f \in F$ in hall $h \in H$ at time slot $t \in T$ can be estimated and is given by the parameter b_{fht}.

The parameter rq_{fht} indicates how many laterals are required for flight $f \in F$ if it is assigned to hall $h \in H$ at time slot $t \in T$. A flight only requires laterals at time slots that are between and including the opening and closing time of a lateral. At each time slot, the number of laterals used for all the flights assigned to a specific hall must be less than or equal to the lateral capacity of that hall.

The shift capacity at hall $h \in H$ is given by sc_h and is determined by the number of baggage units a group of employees can handle per time slot. It represents the productivity of the employees in a certain hall. Employees must be able to handle the total estimated amount of baggage at every time slot. The number of groups assigned to a hall at a specific time slot times the shift capacity of employees in that same hall indicates how many bags can be handled during that time slot. The total amount of baggage in a hall at a certain time slot is determined by the total baggage of all the flights assigned to that hall at that specific time slot.

As mentioned in Subsection 1.3.2, the number of full-time employees is calculated based on the planning. The duration of such a full-time working shift is given by sd. As mentioned in Subsection 1.3.1, by knowing how often a group is allowed to start a shift and the earliest possible opening time of laterals in a baggage hall, the possible starting times of the shifts in a hall can be determined. It must always be possible to fulfill the entire shift, so no shift can start later than the closing time minus the shift duration. Set $T_h \subseteq T$ contains the time slots at which the groups working in hall $h \in H$ may start their shift.

Before assigning baggage of flights to certain baggage halls, it needs to be checked whether there are any flights which are fixed to a baggage hall. Parameter a_{fh} is set to one if flight $f \in F$ is fixed to hall $h \in H$ and has value zero otherwise. The model must ensure that these flights are assigned to the correct hall.

3.2. MIP Formulation

The MIP model that was already created by ORTEC is formulated as follows:

Sets

F set of flights
H set of halls
T set of time slots of δ minutes each
T_h set of time slots at which groups may start their shift in baggage hall $h \in H$, $T_h \subseteq T$

Parameters

$$a_{fh} = \begin{cases} 1 & \text{if flight } f \in F \text{ is fixed to a specific hall } h \in H \\ 0 & \text{otherwise} \end{cases}$$

b_{fht} estimated number of baggage units from flight $f \in F$ in hall $h \in H$ during time slot $t \in T$
lc_{ht} lateral capacity in hall $h \in H$ at time slot $t \in T$
rq_{fht} number of required laterals at time slot $t \in T$ for flight $f \in F$ in case it is assigned to hall $h \in H$
sc_h shift capacity at hall $h \in H$, i.e. the number of baggage units that can be handled per time slot for one group
sd shift duration in number of time slots

Decision variables

U_{ht} number of groups assigned to start their shift in hall h at time slot $t \in T_h$

$$X_{fh} = \begin{cases} 1 & \text{if flight } f \in F \text{ is allocated to hall } h \in H \\ 0 & \text{otherwise} \end{cases}$$

Z_{ht} the number of groups working in hall $h \in H$ at time slot $t \in T$

Mixed integer program

$$\min \quad \sum_{h \in H} \sum_{t \in T} Z_{ht} \tag{3.1}$$

$$\text{s.t.} \quad \sum_{h \in H} X_{fh} = 1 \qquad \forall f \in F \tag{3.2}$$

$$\sum_{i=\max\{1,t-sd+1\}}^{t} U_{hi} = Z_{ht} \qquad \forall h \in H, \forall t \in T \tag{3.3}$$

$$\sum_{f \in F} b_{fht} \cdot X_{fh} \leq sc_h \cdot Z_{ht} \qquad \forall h \in H, \forall t \in T \tag{3.4}$$

$$\sum_{f \in F} rq_{fht} \cdot X_{fh} \leq lc_{ht} \qquad \forall h \in H, \forall t \in T \tag{3.5}$$

$$a_{fh} \leq X_{fh} \qquad \forall f \in F, \forall h \in H \tag{3.6}$$

$$X_{fh} \in \{0,1\} \qquad \forall f \in F, \forall h \in H \tag{3.7}$$

$$U_{ht} \in \mathbb{N}_{\geq 0} \qquad \forall h \in H, \forall t \in T_h \tag{3.8}$$

$$Z_{ht} \in \mathbb{N}_{\geq 0} \qquad \forall h \in H, \forall t \in T \tag{3.9}$$

Objective function (3.1) minimizes the total number of working hours in all halls. Assigning each flight to exactly one hall is done by Constraints (3.2). Given the number of groups starting a shift at a certain time slot in a specific hall, the number of groups working in a hall can be determined for each time slot using the shift duration, which is linked in Constraints (3.3).

Constraints (3.4) ensure that there are enough employees in every hall at each time slot to handle the baggage units assigned to that hall. Enough laterals must always be available for the baggage in each hall at each time slot, which is guaranteed by Constraint (3.5). The flights that are fixed to a specific hall must be assigned to that hall, which is fulfilled by Constraints (3.6). Constraints (3.7), (3.8) and (3.9) indicate the domain of the decision variables.

This basic model does not contain sufficient details in order to create a schedule which could be used in practice. For example, assigning flights to halls is not detailed enough. This is one of the reasons that this model needs to be extended. Other reasons and how to extend the model is described in the following Section 3.3.

3.3. Necessary Extensions

As described in the previous section, the basic model needs to be extended such that the schedule can be used in practice. When validating the outcomes of the basic model with the PlanControl tool of KLM, a decrease in the amount of workload in the baggage halls was observed, but the workload for transporting the luggage to the aircraft increased. The benefits of the decreased work load did not compensate the increase of the costs. Therefore, the basic model was not sufficient and some extensions are needed which are described in this section.

The first necessary extension is to extend the model to plan on lateral level, which means that baggage should be assigned to a specific lateral instead of assigning baggage just to a baggage hall. One of the reasons for this extension is that the BHS requires input information about to which lateral it needs to transport the baggage of a flight. There are different types of laterals and some laterals may only be used for specific flights. It can also happen that the lateral opening times of a certain flight differ per lateral, as the opening and closing times of laterals differ per baggage hall. Some flights require more than one lateral and in this case the baggage must be allocated to laterals that are located next to each other.

Along with the extension on the lateral level, baggage sections need to be included as well, because groups of employees are planned to work at a certain baggage section. When they work at a certain baggage section, they can work on all the laterals that are located in that baggage section. A baggage section is defined by a set of laterals close to each other. Each baggage section has a section lead which is responsible for all the employees working in this baggage section.

Another side effect of the basic model is that it just minimizes the workload in the baggage halls instead of the overall workload. As mentioned before, it resulted in a higher workload for driving the baggage to the aircraft. Therefore, it will be useful to extend the model with the driving tasks and to also minimize the amount of drivers in order to obtain an overall optimum.

The last extension derives from the fact that it is impossible to create a feasible schedule for all the flights, when following the determined opening and closing times for the laterals. Currently, in order to obtain a feasible schedule, the lateral opening times for certain flights must be delayed. Therefore, the model must be extended with the option to delay the lateral opening times. Only the opening time of a lateral can be delayed such that the baggage is stored in the storage a bit longer before it is send to the lateral. So it is not possible to postpone the closing time of a lateral, because otherwise the baggage that was supposed to drop on the lateral close to the scheduled time of departure of the flight can not be dropped and transported to the aircraft.

All mentioned extensions and their implementation are detailed in the following chapters. At first, only the extension for planning on lateral level, in which the baggage sections are added as well, and how to increase the efficiency of this model will be examined in Part II. Part III will discuss how the lateral opening times can be shortened in order to create a feasible schedule. Finally, the driving tasks will be added in Part IV.

II

Lateral Level

4

Problem Description and Formulation

As mentioned in Section 3.3, the basic model needs to be extended by including the lateral level. Along with the extensions of laterals, also baggage sections need to be included, which are a group of laterals to which groups of employees are assigned to work. This chapter describes the mixed integer programming formulation for this extension. Section 4.1 describes the problem on lateral level in more detail, whereas the mixed integer programming formulation is discussed in Section 4.2.

4.1. Lateral Level Problem

Compared to the basic problem described in Section 3.1, this problem assigns the baggage of flights to laterals instead of just to baggage halls. The set of laterals is given by L and the capacity of lateral $l \in L$ is given by ca_l, which indicates the number of sub laterals within that specific lateral. For example, a street that consists of two laterals on top of each other will have capacity two and carousels, will have capacity six.

The baggage sections are given by set S and the laterals that are located within this section are given by the set $L_s \subseteq L$. Set T_s contains the time slots at which groups of employees may start working their shift in section $s \in S$. The lateral capacity lc_{st} of a section $s \in S$ at time slot $t \in T$ can be determined by the total capacity of all the laterals that are in the section $s \in S$ (assuming that all laterals are available at every time slot).

Each flight $f \in F$ requires a certain number of laterals to handle all the baggage of the flight, which is given by r_f. Parameter rq_{fst} denotes the required laterals for flight $f \in F$ in section $s \in S$ at time slot $t \in T$ and is determined in the same way as parameter rq_{fht} of the basic problem. Flights $F_1 \subseteq F$ require one lateral and must be either planned on one single lateral or one lateral with capacity two. Flights that require two laterals are given by the set $F_2 \subseteq F$ and their baggage must either be placed on a lateral with capacity two or on two single laterals next to each other. Set $F_4 \subseteq F$ contains flights that require four laterals and the baggage of these flights must be placed on two laterals with capacity two, which are located next to each other. Parameter $n_{l_a l_b}$ is set to one if lateral l_a is next to lateral l_b and if the index of l_a is smaller than the index of l_b, avoid symmetric solutions and zero otherwise. Two laterals that are placed next to each other must also be located within the same baggage section for the parameter to be set to one.

The baggage of flights is not necessarily allowed to go on all the laterals, because it could have certain restrictions. Therefore, parameter a_{fl} is used which is set to one when the baggage of flight $f \in F$ is allowed to go on lateral $l \in L$ and zero otherwise. This parameter is also a replacement of the parameter a_{fh} of the basic problem which was set to one if a flight $f \in F$ was fixed to hall $h \in H$ and set to zero otherwise. If a flight is fixed to a specific hall, parameter a_{fl} will only have the value one for laterals that are located in that hall such that the baggage of the flight is forced to be allocated to this hall.

A maximum number of flights can be handled on a lateral at the same time, which is indicated by parameter ma_l. To ensure that this maximum is not exceeded, the parameter h_{flt} is used to check which flights are on a lateral at a certain time slot. The parameter is set to one when flight $f \in F$ would be on lateral $l \in L$ at time slot $t \in T$ in case it is assigned to that lateral $l \in L$ and set to zero otherwise. Parameters b_{fst}, sc_s, sd and are obtained in the same way as the parameters b_{fht}, sc_h and sd in the basic problem.

4.2. MIP Formulation

The notation of the sets, the parameters, the decision variables and the MIP are given in this section, with a description of the constraints.

Sets

F	set of flights
F_2	set of flights that require one lateral, $F_2 \subseteq F$
F_2	set of flights that require two laterals, $F_2 \subseteq F$
F_4	set of flights that require four laterals, $F_4 \subseteq F$
S	set of baggage sections
L	set of laterals
L_s	set of laterals that are positioned in section $s \in S$, where $L = \cup_{s\in S} L_s$ and $L_s \cap L_{s'} = \emptyset,\ \forall s,\ s' \in S$
L_1	set of laterals which have capacity one, i.e. a single lateral, $L_1 \subseteq L$
L_2	set of street laterals which have capacity two $L_2 \subseteq L$ and $L_1 \cup L_2 = L$ and $L_1 \cap L_2 = \emptyset$
T	set of time slots of δ minutes each
T_s	set of time slots at which groups of employees may start working their shift in section $s \in S$, $T_s \subseteq T$

Parameters

a_{fl}	$= \begin{cases} 1 & \text{if the baggage of flight } f \in F \text{ is allowed to go on lateral } l \in L \\ 0 & \text{otherwise} \end{cases}$
b_{fst}	estimated number of bags from flight $f \in F$ in section $s \in S$ during time slot $t \in T$
ca_l	capacity of lateral $l \in L$, i.e. the amount of (sub) laterals within that lateral
lc_{st}	lateral capacity in section $s \in S$ at time slot $t \in T$, assuming that all laterals are available at every time slot, determined by $lc_{st} = \sum_{l\in L_s} ca_l$
$n_{l_a l_b}$	$= \begin{cases} 1 & \text{if lateral } l_a \in L_s \text{ is located next to lateral } l_b \in L_s, \text{ where } l_a < l_b \\ 0 & \text{otherwise} \end{cases}$
h_{flt}	$= \begin{cases} 1 & \text{if flight } f \in F \text{ is on lateral } l \in L \text{ at time slot } t \in T \text{ in case it is assigned to lateral } l \in L \\ 0 & \text{otherwise} \end{cases}$
ma_l	maximum number of flights that can be on lateral $l \in L$ at the same time slot
r_f	number of required laterals for flight $f \in F$
rq_{fst}	number of required laterals at time slot $t \in T$ for flight $f \in F$ in case it is assigned to section $s \in S$, determined by $rq_{fst} = r_f \cdot h_{flt}$ for $l \in L_s$
sc_s	shift capacity at section $s \in S$, i.e. the amount of baggage that can be handled per time slot for one working group
sd	shift duration in number of time slots

Decision variables

U_{st}	number of groups assigned to start their shift at time slot $t \in T$ in section $s \in S$
$V_{fl_a l_b}$	$= \begin{cases} 1 & \text{if flight } f \in F \text{ is assigned to both lateral } l_a \in L \text{ and } l_b \in L \\ 0 & \text{otherwise} \end{cases}$
X_{fs}	$= \begin{cases} 1 & \text{if flight } f \in F \text{ is assigned to section } s \in S \\ 0 & \text{otherwise} \end{cases}$
Y_{fl}	$= \begin{cases} 1 & \text{if flight } f \in F \text{ is assigned to lateral } l \in L \\ 0 & \text{otherwise} \end{cases}$
Z_{st}	number of groups working in section $s \in S$ at time slot $t \in T$

Mixed integer program

$$
\begin{aligned}
\min \quad & \sum_{s\in S}\sum_{t\in T} Z_{st} & & & (4.1)\\
\text{s.t.} \quad & \sum_{s\in S} X_{fs} & = 1 & \quad \forall f \in F & (4.2)\\
& \sum_{f\in F} b_{fst}\cdot X_{fs} & \le sc_s\cdot Z_{st} & \quad \forall s\in S,\ \forall t\in T & (4.3)\\
& \sum_{i=\max\{1,t-sd+1\}}^{t} U_{si} & = Z_{st} & \quad \forall s\in S,\ \forall t\in T_s & (4.4)\\
& \sum_{f\in F} rq_{fst}\cdot X_{fs} & \le lc_{st} & \quad \forall s\in S,\ \forall t\in T & (4.5)\\
& \sum_{l\in L_1} Y_{fl_1} + \sum_{l\in L_2} Y_{fl_2} & = 1 & \quad \forall f\in F_1 & (4.6)\\
& \sum_{l\in L_s} ca_l\cdot Y_{fl} & = r_f\cdot X_{fs} & \quad \forall f\in F_2\cup F_4,\ \forall s\in S & (4.7)\\
& Y_{fl} & \le X_{fs} & \quad \forall f\in F,\ \forall s\in S,\ \forall l\in L_s & (4.8)\\
& \sum_{f\in F} h_{flt}\cdot Y_{fl} & \le ma_l & \quad \forall l\in L,\ \forall t\in T & (4.9)\\
& Y_{fl} & \le a_{fl} & \quad \forall f\in F,\ \forall l\in L & (4.10)\\
& V_{fl_al_b} & \le Y_{fl_a} & \quad \forall f\in F_2,\ \forall l_a,\ l_b\in L_1, \text{ with } n_{l_al_b}=1 & (4.11)\\
& V_{fl_al_b} & \le Y_{fl_b} & \quad \forall f\in F_2,\ \forall l_a,\ l_b\in L_1, \text{ with } n_{l_al_b}=1 & (4.12)\\
& \sum_{l_2\in L_2} Y_{fl_2} + \sum_{l_a,\ l_b\in L_1|n_{l_al_b}=1} V_{fl_al_b} & = 1 & \quad \forall f\in F_2 & (4.13)\\
& V_{fl_al_b} & \le Y_{fl_a} & \quad \forall f\in F_4,\ \forall l_a,\ l_b\in L_2, \text{ with } n_{l_al_b}=1 & (4.14)\\
& V_{fl_al_b} & \le Y_{fl_b} & \quad \forall f\in F_4,\ \forall l_a,\ l_b\in L_2, \text{ with } n_{l_al_b}=1 & (4.15)\\
& \sum_{l_a,\ l_b\in L_2|n_{l_al_b}=1} V_{fl_al_b} & = 1 & \quad \forall f\in F_4 & (4.16)\\
& X_{fs} & \in\{0,1\} & \quad \forall f\in F,\ \forall s\in S & (4.17)\\
& Y_{fl} & \in\{0,1\} & \quad \forall f\in F,\ \forall l\in L & (4.18)\\
& V_{fl_al_b} & \in\{0,1\} & \quad \forall f\in F,\ \forall\ l_a,\ l_b\in L, \text{ with } n_{l_al_b}=1 & (4.19)\\
& U_{st} & \in\mathbb{N}_{\ge 0} & \quad \forall s\in S,\ \forall t\in T_s & (4.20)\\
& Z_{st} & \in\mathbb{N}_{\ge 0} & \quad \forall s\in S,\ \forall t\in T & (4.21)
\end{aligned}
$$

Objective function (4.1) minimizes the total number of working hours of the groups in all the baggage sections. Each flight needs to be assigned to exactly one baggage section, which is ensured by Constraints (4.2).

Constraints (4.3) ensure that there are enough employees in each baggage section at every time slot to handle the baggage units assigned to that baggage section. Given the number of groups working at a certain time slot within a specific baggage section, it can be determined how many groups need to start their shift in that baggage section at which time slot, which is done by Constraints (4.4).

Enough laterals must be available for the baggage in each baggage section at every time slot, which is guaranteed by Constraint (4.5). Note that these constraints are based on Constraints (3.5) from the basic model. As mentioned before, the difference between parameters r_f and rq_{fst} is that the latter indicates how many laterals are needed at a certain time slot for a flight that is assigned to a certain baggage section, so it will have value zero outside the opening times of the laterals.

As already explained in Section 4.1, the baggage of a flight $f \in F_1$ must be assigned to either one single lateral or one lateral with capacity two, which is ensured by Constraints (4.6). Baggage of flights that require two or four laterals must be assigned to the exact amount of required laterals, which is ensured by Constraints (4.7).

Constraints (4.8) ensures that flights are only assigned to laterals that are located within the section to which the flight is assigned. Furthermore, the maximum amount of flights that can be on a lateral at the same time can not be exceeded, which is ensured by Constraints (4.9).

Flights can only be assigned to the laterals it is allowed to be assigned to, which is ensured by Constraints (4.10). Constraints (4.11) and (4.12) ensure that when a flight that requires two laterals is assigned to two single laterals, that the corresponding variable indeed indicates that this flight is assigned to these two laterals. A flight that requires two laterals can either be assigned to one lateral with capacity two or the baggage can be placed on two single laterals next to each other, which is guaranteed by Constraints (4.13).

Constraints (4.14) and (4.15) have the same function as Constraints (4.11) and (4.12), for flights that require four laterals instead of two and for street laterals instead of single laterals. If a flight requires four laterals, it can only be assigned to two street laterals which are next to each other, which is ensured by Constraints (4.16).

Constraints (4.17) up to and including (4.21) indicate the domain of the decision variables. A flight can only be assigned to two laterals next to each other if they are indeed located next to each other and a group may only start working if allowed by the parameter W_{st}.

5

Possible Speed-Ups

Speed-ups can be derived in order to possibly decrease the computation time needed to solve the model. Small adjustments to the MIP of Chapter 4 are described in Section 5.1 which could reduce the solving time. A binary program with symmetry breaking constraints is created in Section 5.2. In Section 5.3, the decision variable X_{fs} is removed from the problem which could increase the efficiency by decreasing the amount of variables.

5.1. Possible Speed-Ups for the MIP on Lateral Level

Small adjustments or extensions to the MIP can be made in order to possibly decrease the computation time. These possible speed-ups are described in this section.

5.1.1. Objective Function

Currently, Objective Function (4.1) minimizes the total number of working hours of the groups in all the baggage sections. However, minimizing the total number of starting groups will automatically minimize the number of groups working. Therefore, it is also possible to have the following objective function

$$\min \sum_{s \in S} \sum_{t \in T} U_{st} \tag{5.1}$$

Note that this can only be done because there is a fixed shift duration, namely the full time shift duration. In case various shift durations are used, the two objective functions do not result in the same optimal solution and the original objective formulation must be used.

First tests showed that using Objective Function 5.1 improves the efficiency of the model. Therefore, this objective function will be used in the following sections.

5.1.2. Lateral Capacity Constraints

Constraints (4.5) from the MIP formulation in Section 4.2 ensure that there are enough laterals in a section at each time slot for the amount of flights assigned to that baggage section. These constraints are related to Constraints (3.5) from the basic model. However, in the model on lateral level, it is already ensured by Constraints (4.6) and (4.7) that there are always enough laterals available by stating that at most ma_l flights are allowed on lateral $l \in L$ at every time slot. Therefore, Constraints (4.5) become redundant and deleting these constraints may reduce the computation time needed to solve the method.

5.1.3. Prioritizing Decision Variables

Decision variables can be prioritized with a non-negative integer value. The decision variables that have the lowest priority, will be considered last by the branch-and-cut process of the solver. The decision variable with the highest priority value will be considered first. Prioritizing the decision variables in a logical way could decrease the computation time.

5.2. Binary Programming and Symmetry Breaking

In practice, KLM might have a certain number of employees available to work, and therefore, a certain number of groups can be assigned to work. A different and perhaps more efficient way of modelling this problem is by assigning the work shifts to the predefined groups. In this research, it is assumed that all groups are identical. Suppose that a group is indexed by $g \in G$, where G is the set of groups. The variables U_{st} and Z_{st} need to be changed in binary variables, namely:

$$U_{gst} \quad \begin{cases} 1 & \text{if group } g \text{ starts their shift at time slot } t \text{ in section } s \\ 0 & \text{otherwise} \end{cases}$$

$$Z_{gst} \quad \begin{cases} 1 & \text{if group } g \text{ works in section } s \text{ at time slot } t \\ 0 & \text{otherwise} \end{cases}$$

By changing the variables in this way, the problem is changed into a binary program. This new set up makes it possible to add symmetry breaking constraints. Symmetry breaking constraints can decrease the computation time needed to solve the model, by reducing the search space without cutting off the optimal solution. However, they are not necessarily effective because more constraints need to be evaluated. Because it is assumed that all the groups are identical, interchanging groups from different baggage sections or time slots will result in a symmetric solution which can be broken. A complete overview of the MIP is given in Appendix B, but the overview which highlight the new constraints is given as follows:

$$\min \quad \sum_{g\in G}\sum_{s\in S}\sum_{t\in T} U_{gst} \tag{5.2}$$

$$\sum_{f\in F} b_{fst}\cdot X_{fs} \quad \leq sc_s \cdot \textstyle\sum_{g\in G} Z_{gst} \qquad \forall s\in S,\ \forall t\in T \tag{5.3}$$

$$\sum_{s\in S}\sum_{t\in T} U_{gst} \quad \leq 1 \qquad \forall g\in G \tag{5.4}$$

$$\sum_{s\in S}\sum_{t\in T} U_{gst} \quad \geq \sum_{s\in S}\sum_{t\in T} U_{g+1st} \qquad \forall g\in\{1,\dots,G-1\} \tag{5.5}$$

$$\sum_{t\in T} U_{g+1st} - \sum_{t\in T} U_{gst} - \sum_{t\in T} U_{g+2st} + 1 \quad \geq 0 \qquad \forall g\in\{1,\dots,G-2\},\ \forall s\in S \tag{5.6}$$

Constraints (4.2) and (4.4) - (4.19)

$$U_{gst} \quad \in \mathbb{N}_{\geq 0} \qquad \forall g\in G,\ \forall s\in S,\ \forall t\in T \text{ with } W_{st}=1 \tag{5.7}$$

$$Z_{gst} \quad \in \mathbb{N}_{\geq 0} \qquad \forall g\in G,\ \forall s\in S,\ \forall t\in T \tag{5.8}$$

Objective function (4.1) is changed into objective function (5.2). Constraints (4.3) are replaced by Constraints (5.3), because the indices of variable Z are changed. Constraints (5.4) ensure that a group may start a shift at most once a day.

One way to break symmetry is to assign groups to working shifts in an increasing order, saying that if only a number x of the in total $|G|$ groups need to work on a certain day, only the first x groups are assigned to work. This is formulated in Constraints (5.5), which probably will not break a lot of symmetry, because the groups that are working can still be interchanged. Constraints (5.6) ensure that in each baggage section, the groups that are working are in increasing order. For groups in one specific baggage section, the possible combinations are summarized in Table 5.1 below, which shows that the constraints are valid. The only invalid combination is given in red, where $g+1$ is assigned to another baggage section, and therefore, the groups are not assigned to baggage sections in an increasing order.

$\sum_{t\in T} U_{gst}$	$\sum_{t\in T} U_{g+1st}$	$\sum_{t\in T} U_{g+2st}$	$\sum_{t\in T} U_{g+1st} - \sum_{t\in T} U_{gst} - \sum_{t\in T} U_{g+2st} + 1$
0	0	0	1
0	0	1	0
0	1	0	2
0	1	1	1
1	0	0	0
1	0	1	-1
1	1	0	1
1	1	1	0

Table 5.1: Possible variable value combinations for Constraints (5.6)

The domains of the changed decision variables are indicated in Constraints (5.7) and (5.8).

5.3. Removing Decision Variable X_{fs}

When constructing a graphical overview of how the flights, laterals and baggage sections are related to each other in the model, the graph in Figure 5.1 is constructed.

From this graph, it is obtained that laterals are fixed to a certain baggage section, and that the main decision in this process is how to connect the flights to certain laterals. Furthermore, if this decision is made, it can be automatically be obtained to which baggage section the baggage of a flight is being transported. So the decision variable X_{fs} can be derived from the decision variable Y_{fl} and can be removed from the model. This might decrease the search space and could increase the efficiency.

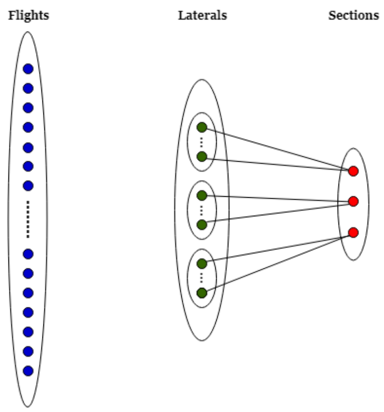


Figure 5.1: Graphical overview of the assignment of flights to laterals

When removing the decision variable X_{fs}, two constraints from the model described in Section 4.2 are adjusted, namely Constraints (4.3) are replaced by Constraints (5.9) and Constraints (4.6) are replaced by Constraints (5.10). For Constraints (5.9), parameter b_{fst} is changed into parameter b_{flt}. Furthermore, it needs to be ensured that all the laterals from a flight are located within the same section. Constraints (4.6) ensure that flights which require only one lateral are only assigned to either one single lateral or one lateral with capacity two, which automatically implies that it is not assigned to two laterals from different sections. Flights that require two or four laterals can not be assigned to laterals from different sections, as Constraints (4.13) and (4.16), respectively, ensure that if such a flight is assigned to two laterals, that the two laterals are located next to each other within the same section. Constraints (5.10) ensure that no more than the required amount of laterals are assigned to flights which require two or four laterals, such that it can not be assigned to extra laterals from different sections either. A complete overview of the MIP is given in Appendix B.

$$\min \sum_{s\in S}\sum_{t\in T} U_{st}$$

$$\sum_{f\in F}\sum_{l\in L} b_{flt}\cdot Y_{fl} \leq sc_s\cdot Z_{st} \qquad \forall s\in S,\ \forall t\in T \tag{5.9}$$

$$\sum_{l\in L} ca_l\cdot Y_{fl} = r_f \qquad \forall f\in F_2\cup F_4 \tag{5.10}$$

Constraints (4.2), (4.4), (4.6), (4.8) - (4.16) and Constraints (4.18) - (4.21) (5.11)

The speed-ups of this chapter will be empirically evaluated in Chapter 11.

6

Polyhedra and Valid Inequalities

First tests showed that the initial LP-relaxation of the model described in Section 4.2 provides a weak lower bound on the optimal solution. Valid inequalities, also called cutting planes, can be added in order to possibly tighten the LP-relaxation. Section 6.1 of this chapter introduces the definitions of valid inequalities, based on the scientific works of Cornuéjols [9], Van Essen [11] and Orlin [21] . Section 6.2 introduces the two valid inequalities used in this research.

6.1. Introduction

A polyhedron is the feasible region of a linear program, which an intersection of a collection of halfspaces.

Definition 6.1 (Halfspaces)**.** *A halfspace in $\mathbb{R}^n$ is the set of all points that satisfy a single inequality constraint, that is,* $\{x \in \mathbb{R}^n : a^T x \leq b\}$ *for some vector* $a \in \mathbb{R}^n$ *and* $b \in \mathbb{R}$ *.*

Definition 6.2 (Polyhedron)**.** *A polyhedron $P \subseteq \mathbb{R}^n$ is the intersection of finitely many halfspaces, that is,* $P := \{x \in \mathbb{R}^n | Ax \leq b\}$*, for a matrix* $A \in \mathbb{R}^{m \times n}$ *and a vector* $b \in \mathbb{R}^m$*.*

Here, it is assumed that the polyhedron is bounded.

Definition 6.3 (Polytope)**.** *A polytope is a bounded polyhedron.*

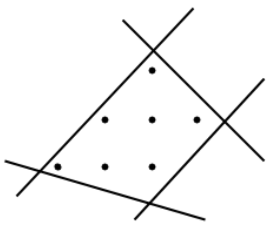


Figure 6.1: Polytope

The convex hull of a set S is the smallest closed convex set that contains S.

Definition 6.4 (Convex Hull)**.** *For $S \subseteq \mathbb{R}^n$, the convex hull of S is the set*

$$\text{Conv}(S) := \left\{ x \in \mathbb{R}^n \middle| x = \sum_{i=1}^{k} \lambda_i y_i,\ y_i \in S,\ ,\lambda_i \geq 0,\ \forall i \in \{1,\ldots,k\},\ \sum_{i=1}^{k} \lambda_i = 1 \right\}.$$

Denote by T, the set $T = P \cap \mathbb{Z}^n$ of integer solutions of the bounded LP-problem, then the integer hull P_I = Conv(T) is the smallest convex set that contains T. Conv(T) is the strongest possible formulation for the set T and it is also a polytope.

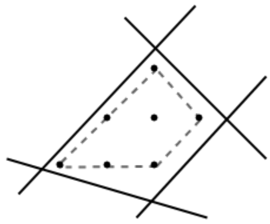


Figure 6.2: Convex hull of set S

Each extreme point of Conv(T) is integer, so when having a linear description of Conv(T), the LP-relaxation would give an integer solution. This would be an ideal situation, however in general, too many constraints are needed to describe Conv(T) and they are hard to find. An approach that is widely used in practice, is adding valid inequalities. Adding valid inequalities to the initial formulation will help finding a better approximation of Conv(T).

Definition 6.5 (Valid Inequality)**.** *An inequality $\alpha x \leq \beta$ is valid for T if $\alpha x \leq \beta$ holds for all $x \in T$.*

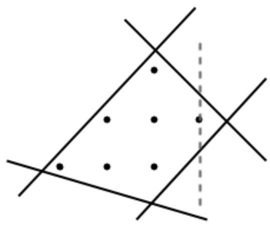


Figure 6.3: Valid Inequality

Valid inequalities are also called cutting planes and they eliminate a part of the LP feasible region without eliminating any feasible integer solution. The new formulation $P' = \{x \in \mathbb{R}^n | Ax \leq b, \alpha x \leq \beta\}$ is at least as strong as the formulation P, i.e. $P' \subseteq P$.

Theorem 6.1. *Given a set $S \subseteq \mathbb{R}^n$ with two formulations P_1 and P_2 and $P_1 \subset P_2$, then P_1 is a better formulation for S than P_2.*

A better formulation of P results in an LP-relaxation that provides a better bound.

Proposition 6.1. *Suppose P_1, P_2 are two formulations for the integer program* $\max\{cx : x \in S \subseteq \mathbb{Z}^n\}$ *and $P_1 \subset P_2$. If the values of the associated linear programming relaxation are denoted by $z_i^{LP} = \max\{cx : x \in P_i\}$ for $i = 1,2$, then $z_1^{LP} \geq z_2^{LP}$ for all c.*

6.2. Adding Valid Inequalities

Two different valid inequalities are used in this research which are introduced in this section. Subsection 6.2.1 introduces valid inequalities that give a lower bound on number of working groups in each time slot. It also illustrates the downside of the parameter that is used in these valid inequalities. Subsection 6.2.2 tries to improve this parameter and introduces the second type of valid inequalities.

6.2.1. Lower Bound on Number of Working Groups

A lower bound on the number of working groups can be derived for each flight $f \in F$ and time slots $t \in T$, where the minimum number of working groups is given by parameter mz_{ft}. This lower bound is derived by dividing the estimated number of baggage units b_{fst} of each flight $f \in F$ by the shift capacity sc_s for each section $s \in S$ at every time slot $t \in T$, i.e. $\frac{b_{fst}}{sc_s}$. By taking the minimum over all the baggage sections for each flight at every time slot, i.e. $\min_{s \in S} \dfrac{b_{fst}}{sc_s}$, the parameter mz_{ft} is obtained.
In order to tighten the LP-relaxation of the model described in Section 4.2, the following valid inequality can be added to the problem:

$$\sum_{s \in S} Z_{st} \geq \left\lceil \sum_{f \in F} mz_{ft} \right\rceil \quad \forall t \in T \tag{6.1}$$

Valid Inequalities (6.1) state that the number of groups working in a baggage section at a certain time slot must be greater or equal to the smallest integer that is greater than or equal to the sum of the minimum required number of working groups at a certain time slot for all flights.

Unfortunately, the values of the parameter mz_{ft} are expected to be low and not representative for the actual number of groups needed at a time slot. This is due to the different opening and closing times of laterals in different baggage sections. Table 6.1 denotes an example of how value mz is determined for a flight $f_1 \in F$ and time slots t_i, where $i \in \{1,\ldots,7\}$ assuming that there are three sections $s_1,\ s_2,\ s_3 \in S$. The presented values are obtained after dividing the parameter b_{fst} by sc_s for every section and time slot.

f_1	t_1	t_2	t_3	t_4	t_5	t_6	t_7
s_1	11.26	9.07	8.16	3.56	2.14	1.52	0.76
s_2	10.57	8.31	7.49	4.27	3.08	1.94	0.84
s_3	0	0	0	15.30	11.28	9.89	0
mz	**0**	**0**	**0**	**3.56**	**2.14**	**1.52**	**0**

Table 6.1: Example of how the parameter mz_{ft} is obtained from the parameter b_{fst} divided by sc_s.

The laterals in section $s_3 \in S$ open at time slot $t_4 \in T$ and close at the end of time slot $t_6 \in T$ for flight $f_1 \in F$. Therefore, the expected amount of baggage outside these opening times is zero for this baggage section and the value of mz_{ft} at these time slots will be zero, because it is the minimum over all the baggage sections. Because of the low value of mz_{ft} for most flights $f \in F$ and time slots $t \in T$, it is expected that Valid Inequalities (6.1) are not very effective in tightening the LP-relaxation.

6.2.2. Improving Parameter mz_{ft}

This section shows how parameter mz_{ft} can be improved. While determining the minimum number of working groups for every flight at each time slot as described above, it is not checked whether the flights can be actually assigned to the baggage sections for which the minimum is chosen. For example, suppose that 30 flights will have their minimum number of needed working groups at time slot t_1 for section s_3 as in the example of Table 6.1. Since, the baggage section only contains ten laterals, no more than 10 flights can be assigned to this baggage section at the same time slot. Therefore, the actual minimum number of needed working groups at that time slot will be greater than or equal to $\sum_{f \in F} mz_{ft_1}$.

Per time slot, the minimal number of working groups mw_t needed can be determined such that the problem mentioned before does not occur. This is done via an MIP which is compiled for every time slot $t \in T$ separately and which ensures that all flights can be assigned to baggage sections such that there are enough laterals and capacity available during that time slot. Flights for which baggage could drop in at least one of the baggage sections at time slot $t \in T$, flights for which it holds that $hs_{fst} = 1$, are stored in the set $F_b \subseteq F$. The minimal number of laterals that a flight $f \in F$ needs is stored in the parameter $ml_f = \left\lceil \frac{r_f}{\max_{l \in L} ca_l} \right\rceil$. The total number of laterals that are located in a section $s \in S$ is stored in the parameter tl_s and given by $\sum_{l \in L} ps_{ls}$. Parameters bt_{fs}, hst_{fs} and rqt_{fs} are equal to parameters b_{fst}, hs_{fst} and rq_{fst} within each of the MIPs for time slot $t \in T$. The notation of the new sets, parameters, decision variables and MIP for every time slot $t \in T$ is introduced as follows:

Sets

F_b set of flights for which baggage could drop in at least one of the baggage sections, $F_b \subseteq F$

Parameters

ml_f — minimum laterals that are required for flight $f \in F$, i.e. $ml_f = \left\lceil \frac{r_f}{\max_{l \in L} ca_l} \right\rceil$

tl_s — total number of laterals in section $s \in S$, i.e., $tl_s = \sum_{l \in L} ps_{ls}$

bt_{fs} — the estimated number of bags for flight $f \in F$ in section $s \in S$

hst_{fs} — $\begin{cases} 1 & \text{if lateral(s) for flight } f \in F \text{ would be open in section } s \in S \text{ in case it is assigned to section } s \in S \\ 0 & \text{otherwise} \end{cases}$

rqt_{fs} — number of required laterals for flight $f \in F$ in case it is assigned to section $s \in S$

Variables

MWS_s — minimum amount of working groups needed in section $s \in S$ such that all the flights can be assigned to a section

MIP Formulation - "Determining mw_t"

$$\min \sum_{s \in S} MWS_s \tag{6.2}$$

$$\text{s.t.} \quad \sum_{s \in S} X_{fs} = 1 \qquad \forall f \in F_b \tag{6.3}$$

$$\sum_{f \in F_b} bt_{fs} \cdot X_{fs} \leq sc_s \cdot MWS_s \qquad \forall s \in S \tag{6.4}$$

$$\sum_{f \in F_b} rqt_{fs} \cdot X_{fs} \leq lc_{st} \qquad \forall s \in S \tag{6.5}$$

$$\sum_{f \in F_b} hst_{fs} \cdot ml_f \cdot X_{fs} \leq tl_s \qquad \forall s \in S \tag{6.6}$$

$$X_{fs} \in \{0,1\} \qquad \forall f \in F, \ \forall s \in S \tag{6.7}$$

$$MWS_s \in \mathbb{N}_{\geq 0} \qquad \forall s \in S \tag{6.8}$$

Objective function (6.2) minimizes the total number of needed working groups. Constraints (6.3) ensure that all flights are assigned to exactly one baggage section. The minimum number of working groups required per section to handle all the baggage is determined by Constraints (6.4). Enough laterals must be available for the baggage in each section, which is guaranteed by Constraint (6.5). Constraints (6.6) ensure that the minimal required amount of laterals are available for all flights that are assigned to a section. The domains of the decision variables are indicated in Constraints (6.7) and (6.8).

In order to obtain the minimum total number of working groups mw_t at every time slot $t \in T$, Algorithm 3 is used.

Algorithm 1 Procedure for obtaining the parameter mw_t

procedure
  for $t \in T$ **do**
    Empty the set F_b
    for $f \in F$ **do**
      if $\sum_{s \in S} hst_{fs} = 1$ **then**
        Add flight f to the set F_b
      end if
    end for
    for $f \in F_b$ **do**
      $bt_{fs} \leftarrow b_{fst}$
      $hst_{fs} \leftarrow hs_{fst}$
      $rqt_{fs} \leftarrow rq_{fst}$
    end for
    Solve the model "Determining mw_t"
    $mw_t \leftarrow \sum_{s \in S} MWS_s$
  end for
end procedure

When the parameter mw_t is obtained, the Valid Inequalities (6.9) can be added to the MIP formulation of Section 4.2 and provide a lower bound on the number of working groups per time slot.

$$\sum_{s \in S} Z_{st} \quad \geq \quad mw_t \qquad \forall t \in T \tag{6.9}$$

The effect of both valid inequalities will be tested in Section 11.1. Note that the valid inequalities will not be used at the same time, because they are of the same sort with only a different used parameter.

7

Hierarchical Solution Method

First tests showed that adding the valid inequalities described in Chapter 6 did not completely resolve the issue concerning the weak LP-relaxation and these results also indicated that this weak LP-relaxation influences the computation time in a negative way. A hierarchical solution method which splits the total problem into two phases is described in this chapter, such that the size of the problem becomes smaller and the computation time is expected to decrease. This new method is exact as long as no time limit is used, but otherwise it is a heuristic.

In the first phase, flights are assigned to a baggage section and the total workload is minimized. This is introduced in Section 7.1. In the second phase, the flights are assigned to laterals given the assignment of flights to sections from the first phase. The sizes of both the first and second phase, are way smaller than the total problem, and because of this, the total computation time is expected to be less than for the total problem. However, just solving the first and second phase hierarchically will not automatically result in a feasible solution, which is shown in Section 7.3. In order to obtain a feasible optimal solution, an iterative method is introduced and described in Section 7.4.

7.1. First Phase

At first, flights will be assigned to baggage sections and the groups of employees will be assigned to start a working shift such that the number of working hours will be minimized. This problem does not take into account the assignment of flights to laterals. However, there must still be enough space in a baggage section for all the flights at every time slot. The MIP formulation and one adapted parameter are given below.

Parameters

hs_{fst} $\begin{cases} 1 & \text{if lateral(s) for flight } f \in F \text{ would be open in section } s \in S \text{ at time slot } t \in T \text{ in case it is} \\ & \text{assigned to section } s \in S,\ hs_{fst} = h_{flt} \text{ with } l \in L_s \\ 0 & \text{otherwise} \end{cases}$

MIP Formulation

$$
\begin{aligned}
\min \quad & \sum_{s \in S} \sum_{t \in T} U_{st} && && (7.1)\\
\text{s.t.} \quad & \sum_{f \in F} hs_{fst} \cdot ml_f \cdot X_{fs} && \leq tl_s && \forall s \in S,\ \forall t \in T \quad (7.2)\\
& \sum_{s \in S} X_{fs} && = 1 && \forall f \in F\\
& \sum_{f \in F} b_{fst} \cdot X_{fs} && \leq sc_s \cdot Z_{st} && \forall s \in S,\ \forall t \in T\\
& \sum_{i=\max\{1,t-sd+1\}}^{t} U_{si} && = Z_{st} && \forall s \in S,\ \forall t \in T_s\\
& \sum_{f \in F} rq_{fst} \cdot X_{fs} && \leq \sum_{l \in L_s} ca_l && \forall s \in S,\ \forall t \in T\\
& X_{fs} && \in \{0,1\} && \forall f \in F,\ \forall s \in S\\
& U_{st} && \in \mathbb{N}_{\geq 0} && \forall s \in S,\ \forall t \in T \text{ with } W_{st} = 1\\
& Z_{st} && \in \mathbb{N}_{\geq 0} && \forall s \in S,\ \forall t \in T
\end{aligned}
$$

The objective function (7.1) minimizes the number of working groups starting a shift. Constraints (7.2) ensure that the minimal required amount of laterals are available for all flights that are assigned to a baggage section. The other constraints are identical to Constraints (4.2) - (4.5), (4.17), (4.20) and (4.21).

7.2. Second Phase

After the flights are assigned to baggage sections in the first phase, the flights must be assigned to specific laterals while fulfilling all the constraints. This can be done for each section separately, by creating a second phase MIP that searches for a solution that satisfies Constraint (4.6) up to and including (4.16), Constraints (4.18) and (4.19) from the MIP in Section 4.2. Objective function (7.3) is a constant value ϵ, since the goal is to just find a feasible assignment of flights to laterals. Note that because this MIP is compiled for each of the sections $s \in S$ separately, that only the laterals $l \in L_s$ are used in the MIP. Furthermore, the variables X_{fs} that are stated in these original constraints, are stored into a parameter x_{fs} as the values are already obtained in the first phase.

MIP Formulation

$$
\begin{aligned}
\min \quad & \epsilon && && (7.3)\\
\text{s.t.} \quad & \sum_{l \in L_1 \cap L_s} Y_{fl_1} + \sum_{l \in L_2 \cap L_s} Y_{fl_2} && = 1 && \forall f \in F_1\\
& \sum_{l \in L_s} ca_l \cdot Y_{fl} && = r_f \cdot x_{fs} && \forall f \in F_2 \cup F_4,\ \forall s \in S\\
& Y_{fl} && \leq x_{fs} && \forall f \in F,\ \forall s \in S,\ \forall l \in L_s\\
& \sum_{f \in F} h_{flt} \cdot Y_{fl} && \leq ma_l && \forall l \in L_s,\ \forall t \in T\\
& Y_{fl} && \leq a_{fl} && \forall f \in F,\ \forall l \in L_s\\
& V_{fl_al_b} && \leq Y_{fl_a} && \forall f \in F_2,\ \forall l_a,\ l_b \in L_1 \cap L_s, \text{ with } n_{l_al_b} = 1\\
& V_{fl_al_b} && \leq Y_{fl_b} && \forall f \in F_2,\ \forall l_a,\ l_b \in L_1 \cap L_s, \text{ with } n_{l_al_b} = 1\\
& \sum_{l_2 \in L_2 \cap L_s} Y_{fl_2} + \sum_{l_a,\ l_b \in L_1 \cap L_s \mid n_{l_al_b} = 1} V_{fl_al_b} && = 1 && \forall f \in F_2\\
& V_{fl_al_b} && \leq Y_{fl_a} && \forall f \in F_4,\ \forall l_a,\ l_b \in L_2 \cap L_s, \text{ with } n_{l_al_b} = 1\\
& V_{fl_al_b} && \leq Y_{fl_b} && \forall f \in F_4,\ \forall l_a,\ l_b \in L_2 \cap L_s, \text{ with } n_{l_al_b} = 1\\
& \sum_{l_a,\ l_b \in L_2 \cap L_s \mid n_{l_al_b} = 1} V_{fl_al_b} && = 1 && \forall f \in F_4\\
& Y_{fl} && \in \{0,1\} && \forall f \in F,\ \forall l \in L_s\\
& V_{fl_al_b} && \in \{0,1\} && \forall f \in F,\ \forall\ l_a,\ l_b \in L_s, \text{ with } n_{l_al_b} = 1
\end{aligned}
$$

7.2.1. Complexity of Second Phase

Different types of problems are solved within the second phase and the complexity of some of them is desribed in this subsection. All the flights are already assigned an opening time in the first phase and the closing time of a lateral is already provided by the data, and therefore, the second phase is about scheduled jobs with fixed start and end times. In case only one type of lateral is located within a section and only flights with the same required amount of laterals are assigned to this section, the second phase is polynomial solvable according to Arkin et al. [2]. In case multiple types of laterals are located within a section and only flights with the same required amount of laterals are assigned to this section, the second phase becomes NP-Complete according to Arkin et al. [2]. However, Dondeti et al. [12] presented a polynomial algorithm in case there are only two different types of laterals within a section. Due to time restrictions, the complexity of the remaining second phase problems is not investigated.

7.3. Proof of Insufficiency

Unfortunately, the constraints included in the first phase are not sufficient to guarantee a feasible solution in the second phase, which will be proven in this section.

Theorem 7.1. *Solving the first phase of Section 7.1, does not guarantee a feasible solution for the sections in the second phase of Section 7.2 while using the output of the first phase.*

Proof. Assume, for the sake of contradiction that the assignment of flights to sections from the first phase, guarantees a feasible assignment of flights to laterals within each of the sections. Assume that flights f_1, f_2, f_3, f_4, f_5, f_6, f_7, $f_8 \in F$ are assigned to section $s_1 \in S$ in the first phase. Laterals l_1, l_2, l_3, $l_4 \in L$, with $ca_{l_1} = ca_{l_2} = 2$ and $ca_{l_3} = ca_{l_4} = 1$ are located in section $s_1 \in S$. Figure 7.1 illustrates the time slots in which baggage is dropped for a flight and the value r_f indicates the number of required laterals for a flight. Constraints (7.2) and (6.5) are satisfied for the assignment of these flights to the baggage section. However, no feasible solution exists in the second phase. As illustrated in Figure 7.2, it is not possible to assign flight $f_8 \in F$ to a lateral. Flight $f_5 \in F$ was forced to be scheduled on laterals l_3 and $l_4 \in L$ which forced flight $f_6 \in F$ to be assigned to lateral $l_2 \in L$.

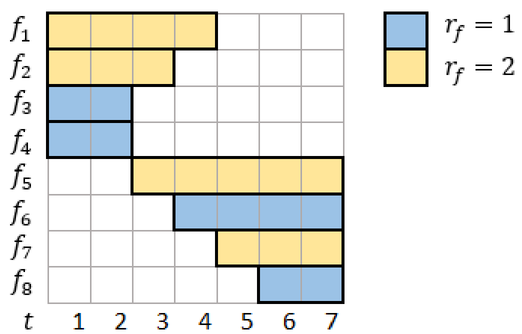


Figure 7.1: Overview of the opening times of the flights that are assigned to s_1 by the first phase, together with the r_f value.

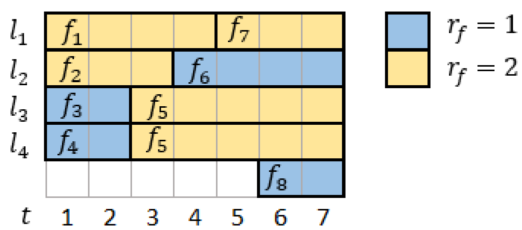


Figure 7.2: Planning on lateral level which shows that the outcomes of the first phase are infeasible, because flight $f_6 \in F$ can not be assigned to a lateral.

Since a contradiction is found, it is not guaranteed that the outcomes of the first phase result in feasible solutions for each of the sections in the second phase.

□

7.4. Iterative Method

As shown in the previous section, the resulting solution of the first phase is not always feasible for the problem in the second phase. Therefore, an iterative method is introduced in this section, which iterates between the first and second phase such that a feasible and optimal solution can be obtained. When the problem in the second phase is infeasible for one of the baggage sections, the flights that were assigned to this baggage section, can not be assigned to the laterals within that baggage section, while satisfying all constraints. In order to avoid the infeasible combination of set of flight to certain baggage sections, the first phase is slightly adjusted in Subsection 7.4.1. The complete iterative method which interacts between the first and second phase and stores the infeasible options is introduced and described in more detail in Subsection 7.4.2.

7.4.1. Adjusting First Phase

The first phase model introduced in Section 7.1 is adjusted such that the known infeasible assignments of certain sets of flights to certain baggage sections are avoided. When the second phase is infeasible for a certain section, the infeasible set of flights are denoted as one infeasible option, where I is the set of infeasible options. Parameter as_{ifs} is set to one if flight $f \in F$ is assigned to section $s \in S$ in infeasible option $i \in I$ and zero otherwise.

Sets

I infeasible options

Parameters

$$as_{ifs} = \begin{cases} 1 & \text{if flight } f \in F \text{ is assigned to section } s \in S \text{ in infeasible option } i \in I \\ 0 & \text{otherwise} \end{cases}$$

The first phase model will be adjusted such that infeasible options are not allowed anymore. Not all flights that are assigned to a baggage section in an infeasible option are allowed to be assigned to that same baggage section again, which is ensured by Constraints (7.4) which are added to the first phase model.

$$\sum_{f \in F} as_{ifs} \cdot X_{fs} \leq \sum_{f \in F} as_{ifs} - 1 \qquad \forall i \in I,\ \forall s \in S \text{ with } \sum_{f \in F} as_{ifs} > 0 \tag{7.4}$$

Already known infeasible assignments of flights to baggage sections are now avoided in the first phase method. The adjusted first and second phase models are used in the iterative method, from which the outline is described in the next subsection.

7.4.2. Outline of the Iterative Method

In order to obtain a feasible and optimal solution, an iterative method is used which is described in this subsection. The iterative method consists of the following steps:

1. Solve the first phase and interrupt in the following three cases: 1) when a better solution than the currently best solution is obtained, 2) when the time limit for solving the first phase is reached or 3) the program status was optimal. The reason for interrupting the first phase when an improved objective function value is reached, will be explained later in this subsection.

2. Solve second phase for all the baggage sections.

   (a) When none of the second phase MIP's results in infeasibility and the objective function value for the first phase is better than the currently best found objective function value, the feasible solution is stored.

   (b) When the second phase is infeasible for at least one of the baggage sections, the infeasible solutions are stored into the parameter as_{ifs}. Other solutions which will also be infeasible for a baggage section are also stored into parameter as_{ifs}. Such as the solution obtained by replacing a flight of the infeasible set of flights by another flight, which is not yet in the infeasible set and has the exact same characteristics. A flight has the exact same characteristics if the opening and closing time of the laterals in a section are the same and if the flight requires the same amount of laterals. Identical sections will have the same infeasible set of flights.

3. If a feasible solution is obtained, this solution is used as a starting solution for the first phase.

4. The iterative method terminates when no better objective function can be found or when the time limit is reached.

When solving the first phase, it is not known for sure whether the solution is feasible for the second phase. Using a time limit as stopping criteria for the first face is not efficient, because when the solution is infeasible, the first phase must start from scratch again later on which is not efficient. Therefore, when the first phase method is interrupted as soon as a better solution is found, it will be checked immediately if this solution is feasible and the solution is either saved as starting solution or the infeasible options are added to I. A maximum running time will still be used for the first phase in case the objective value does not improve and a maximal running time will be used for the entire iterative method. Therefore, the iterative method is a heuristic method.

The difference between the two stopping criteria is shown with a small example. Assume that the time limit for the first phase would be 1800 seconds. Table 7.1 shows the computation times for the runs with the different stopping criteria. The objective values 300, 200 and 100 in the first phase result in a feasible solution in the second phase, whereas the solution with objective value 90 does not result in a feasible solution in the second phase. Just before the time limit is reached in the first phase, an infeasible solution is obtained and a new run needs to be done for the first phase. The total computation time of the two runs with the time limit of 1800 seconds as a stopping criteria is 3600 seconds. When the stopping criteria is to interrupt when a new improved solution is found, the total computation time is only 1140 seconds. Here, it is assumed that it takes 20 seconds to prepare the warm start. This example shows that it is better to have the stopping criteria of interrupting as soon as a new improved objective value is found. If a correct time limit for the first phase is found, e.g. 1100 seconds for the first phase in this example, it results in a better total computation time. However, it is not likely that the correct time limit is chosen in advance.

		Best Objective Value Found After x Seconds				Total Computation
Stopping Criteria	**Run**	**300**	**200**	**100**	**90**	**Time**
Time limit on first phase	1	40 sec.	800 sec.	1100 sec.	1752 sec.	1800 seconds
	2	40 sec.	800 sec.	1100 sec.		**3600 seconds**
Interrupted when new improved objective is found	1	40 sec.				40 seconds
	2		780 sec.			820 seconds
	3			320 sec.		**1140 seconds**

Table 7.1: Example of the difference in total computation time of having a time limit on the first phase or interrupting when a new improved objective is found, where the computation times are presented in seconds

As explained in step 2a, if a feasible solution is found with a better objective function value than the previously stored best objective value, the solution will be stored into parameters. These parameters in which the decision variables belonging to this new feasible solution will be stored, are formulated below, along with some other parameters that are needed for the iterative method. The pseudocode of the iterative method is given in Algorithm 4 and the four steps described above are indicated within the pseudocode.

Parameters

q	$\begin{cases} 1 & \text{if a feasible solution is already obtained} \\ 0 & \text{otherwise} \end{cases}$
nf	$\begin{cases} 1 & \text{if at least one second phase MIP was infeasible for the solution of the first phase} \\ 0 & \text{otherwise} \end{cases}$
c_1	time limit stopping criteria for solving the model of the first phase
c	time limit stopping criteria for the entire iterative method
$\overline{ob}$	the currently best found objective function value

Algorithm 2 Outline of Iterative Method

```
procedure
    Empty $as_{ifs}$
    No feasible options yet, $q \leftarrow 0$
    $i \leftarrow 0$
    $\overline{ob} \leftarrow N$ where $N \in \mathbb{N}_{\geq 0}$ is big enough
    while time limit $c$ is not reached do                                          ▷ Step 4
        if $q = 1$ then                                                             ▷ Step 3
            the best obtained feasible solution so far as start solution
        end if
        Solve first phase and interrupt when a better objective function value is obtained than $\overline{ob}$ or
          when the time limit $c_1$ is reached                                      ▷ Step 1
        if no better solution is obtained, i.e., $\sum_{s \in S} \sum_{t \in T} Z_{st} \geq \overline{ob}$ then   ▷ Step 4
            Break
        end if
        $nf \leftarrow 0$
        for all sections $s^* \in S$ do
            Solve second phase                                                      ▷ Step 2
            if program status is infeasible then
                indicate that a feasible solution is found, $nf \leftarrow 1$
                $i \leftarrow i + 1$
                Store the infeasible solution, $as_{ifs^*} \leftarrow X_{fs^*}$
                for all sections $s' \in S$ that are identical to $s^*$ do             ▷ Step 2b
                    $i \leftarrow i + 1$
                    $as_{ifs'} \leftarrow X_{fs^*}$
                end for
                for all flights $f^* \in F$ that are assigned to the section $s^* \in S$, i.e., $X_{f^*s^*} = 1$ do
                    for all flights $f' \in F$ with the exact same characteristics as flight $f^*$ and which is not
                      already assigned to this section, i.e., $as_{if's^*} = 0$ do
                        The solution of replacing $f'$ and $f^*$ within this section is also infeasible
                        New infeasible solution will be stored, $i \leftarrow i + 1$
                        Store temporary $X^*_{fs^*} \leftarrow X_{fs^*}$, such that the flights $f'$ and $f^*$ can be swapped
                        The value of the original flight becomes zero $X^*_{f^*s^*} \leftarrow 0$
                        The identical flight in combination with the other flights is also infeasible $X^*_{f's^*} \leftarrow 1$
                        $as_{ifs^*} \leftarrow X^*_{f's^*}$
                        for all sections $s' \in S$ that are identical to $s^*$ do
                            $i \leftarrow i + 1$
                            $as_{ifs'} \leftarrow X^*_{fs^*}$
                        end for
                    end for
                end for
            end if
        end for
        if no infeasible status was given, $nf = 0$ and $\sum_{s \in S} \sum_{t \in T} Z_{st} < \overline{ob}$ then   ▷ Step 2a
            Feasible solution is obtained, so $q \leftarrow 1$
            Store all the variables of the best obtained feasible solution so far
        end if
    end while
end procedure
```

The results for the hierarchical solution method are described in Section 11.2.

8

Column Generation

An efficient method to solve large-scale linear programs is column generation. Column generation starts with only a small subset of the variables and adds only the variables which could improve the objective function. In order to prevent enumerating the large amount of variables in the problem of this research, column generation is applied. Section 8.1 introduces the concept of column generation, based on the scientific works of Descrosiers et al. [11] and Gilmore et al. [16]. Section 8.2 describes how column generation is applied on the problem of this research.

8.1. Introduction

Column generation is an interesting technique if there are too many variables in comparison to the number of constraints. It allows to solve certain LP-problems to optimality without writing down the complete constraint matrix explicitly. Only when the corresponding decision variable of a column of the constraint matrix improves the current basic solution, the column is generated into the problem formulation. Variables that are not considered are implicitly equal to zero. The theory behind column generation is explained further in this section.

Consider the following linear program, called the master problem (MP):

$$\begin{aligned} z^* := \min \quad & \sum_{j\in J} c_j \lambda_j & & \\ \text{s.t.} \quad & \sum_{j\in J} \boldsymbol{a}_j \lambda_j & \geq \boldsymbol{b} & \\ & \lambda_j & \geq 0, & \quad j \in J. \end{aligned}$$

This problem has n variables and m constraints and n is exponential in m for many applications. Therefore, explicitly stating all the variables of the master problem, may be computationally impossible [11]. Instead, a restricted master problem (RMP) is considered, with a reasonably small subset $J' \subseteq J$ of variables. By solving the RMP, an optimal primal $\boldsymbol{\lambda}^*$ and optimal dual solution $\boldsymbol{\pi}^*$ are obtained. This optimal solution $\boldsymbol{\lambda}^*$ of the RMP does not have to be optimal for the MP. Like in the simplex method, a nonbasic variable of negative reduced costs needs to be found in the pricing problem to enter the basis, because these negative reduce costs can be interpreted as a potential improvement in the objective function. The pricing problem (PP) or also called, sub problem is given by:

$$\bar{c}^* := \min\left\{c_j - \boldsymbol{\pi}^* \boldsymbol{a}_j \mid j \in J\right\}. \tag{8.1}$$

If the pricing problem yields an optimal solution with non-negative reduced cost, i.e., $\bar{c}^* \geq 0$, no further improving variable can be found, and in this case, $\boldsymbol{\lambda}^*$ is also an optimal solutions for the MP. When $\bar{c}^* < 0$, the variable λ_j and its coefficient column $(c_j, \boldsymbol{a}_j)$ are added to the RMP. The RMP will be optimized again and the process iterates until no further improving variable can be found.

The effectiveness of column generation depends, among other things, on whether it is possible to formulate a pricing problem and whether this pricing problem can be solved relatively fast.

The problem of this research is an MIP and its LP can be solved to optimality by column generation. If the solution of the LP happens to be integral, then it is a solution for the original MIP too. If the solution is fractional, it has to be converted to an integer solution [10]. Applying branch-and-bound to the RMP does not guarantee an optimal or even feasible solution, and therefore, column generation needs to be embedded into branch-and-bound in order to find an optimal solution. The hybrid of branch-and-bound and column generation is called branch-and-price [6].

8.2. Applying Column Generation

This section describes how column generation is applied in this research and the approach is inspired on the column generation approach for gate planning by Diepen et al. [10]. Column generation solves the LP-relaxation of the problem to optimality. Subsection 8.2.1 describes the master problem along with the restricted master problem. The pricing problem is given in Subsection 8.2.2. Subsection 8.2.3 describes how an integer solution can be obtained in case the optimal solution of the LP does not happen to be integral. The entire CG process is summarized in Subsection 8.2.4.

8.2.1. Master Problem and Restricted Master Problem

Before formulating the MP, the set P of section plans is introduced. A series of flights that are to be assigned to the same section is called a section plan $p \in P$. Parameter x_{fp} is set to one if flight $f \in F$ is in section plan $p \in P$ and zero otherwise and parameter e_{ps} is set to one if section plan $p \in P$ can be assigned to section $s \in S$ and zero otherwise. The cost of assigning section plan $p \in P$ to section $s \in S$ is given by c_{ps}. The objective function of the master problem is to minimize the total costs. Furthermore, each flight must be assigned to exactly one section and a section plan can be assigned to at most one section. In case a section plan is assigned to multiple sections, the first type of constraints is violated, because the flights within this section plan are also assigned to multiple sections. The master problem is formulated as follows:

Sets

P section plans

Parameters

$$x_{fp} = \begin{cases} 1 & \text{if flight } f \in F \text{ is in section plan } p \in P \\ 0 & \text{otherwise} \end{cases}$$

$$e_{ps} = \begin{cases} 1 & \text{if section plan } p \in P \text{ is allowed to be assigned to section } s \in S \\ 0 & \text{otherwise} \end{cases}$$

c_{ps} costs of assigning section plan $p \in P$ to section $s \in S$

Decision variables

$$\lambda_{ps} = \begin{cases} 1 & \text{if section plan } p \in P \text{ is assigned to section } s \in S \\ 0 & \text{otherwise} \end{cases}$$

Mixed integer program

$$\min \quad \sum_{p \in P} \sum_{s \in S} c_{ps} \cdot \lambda_{ps} \tag{8.2}$$

$$\text{s.t.} \quad \sum_{p \in P} \sum_{s \in S} x_{fp} \cdot \lambda_{ps} = 1 \qquad \forall f \in F \tag{8.3}$$

$$\sum_{p \in P} e_{ps} \cdot \lambda_{ps} \leq 1 \qquad \forall s \in S \tag{8.4}$$

$$\lambda_{ps} \in \{0,1\} \qquad \forall p \in P,\ \forall s \in S, \text{ with } e_{ps} = 1 \tag{8.5}$$

Objective Function (8.2) minimizes the total costs of assigning section plans to sections. Constraints (8.3) ensure that each flight is assigned to exactly one section and a section plan can not be assigned to more than one section because of Constraints (8.4). The domain of the variables is denoted in Constraints (8.5).

The restricted linear master (RLMP) is obtained by relaxing the integrality constraints (8.5) and by taking a small subset $P' \subseteq P$ of columns:

$$\min \quad \sum_{p\in P'}\sum_{s\in S} c_{ps}\cdot\lambda_{ps} \tag{8.6}$$

$$\text{s.t.} \quad \sum_{p\in P}\sum_{s\in S} x_{fp}\cdot\lambda_{ps} = 1 \qquad \forall f\in F \tag{8.7}$$

$$\sum_{p\in P'} e_{ps}\cdot\lambda_{ps} \leq 1 \qquad \forall s\in S \tag{8.8}$$

$$0\leq\lambda_{ps}\leq 1 \qquad \forall p\in P',\ \forall s\in S,\ \text{with } e_{ps}=1 \tag{8.9}$$

An initial solution is created in which all flights in a section plan $p* \in P' \subseteq P$ and are assigned to a certain section $s* \in S$. As this assignment is not feasible, the costs $c_{p's'}$ are set to M, where $M \in \mathbb{N}_{\geq 0}$ is big enough. Integrality Constraints (8.5) are relaxed and together with the initial solution, the resulting restricted linear master problem (RLMP) is obtained: After the RLMP is solved, the pricing problem must be compiled to see if negative reduced costs can be found.

8.2.2. Pricing Problem

When solving the RMP of Subsection 8.2.1, the dual multipliers π_f are found for Constraint (8.7) corresponding to flight $f \in F$ and dual multipliers μ_s for Constraints (8.8) corresponding to section $s \in S$. The pricing problem is solved separately for each baggage section $s \in S$ and is formulated as follows:

Decision variables

U_t number of groups assigned to start their shift at time slot $t \in T$

$$X_f = \begin{cases} 1 & \text{if flight } f\in F \text{ is assigned} \\ 0 & \text{otherwise} \end{cases}$$

Z_t number of groups working at time slot $t \in T$

Mixed integer program

$$\min \quad \sum_{t\in T} U_t - \sum_{f\in F}\pi_f\cdot X_f - \mu_s \tag{8.10}$$

$$\sum_{f\in F} b_{fst}\cdot X_f \leq sc_s\cdot Z_t \qquad \forall t\in T \tag{8.11}$$

$$\sum_{i=\max\{1,t-sd+1\}}^{t} U_i = Z_t \qquad \forall t\in T_s \tag{8.12}$$

$$\sum_{l\in L_1\cap L_s} Y_{fl_1} + \sum_{l\in L_2\cap L_s} Y_{fl_2} = X_f \qquad \forall f\in F_1 \tag{8.13}$$

$$\sum_{l\in L_s} ca_l\cdot Y_{fl} = r_f\cdot X_f \qquad \forall f\in F_2\cup F_4 \tag{8.14}$$

$$Y_{fl} \leq X_f \qquad \forall f\in F,\ \forall l\in L_s \tag{8.15}$$

$$\sum_{f\in F} h_{flt}\cdot Y_{fl} \leq ma_l \qquad \forall l\in L_s,\ \forall t\in T \tag{8.16}$$

$$Y_{fl} \leq a_{fl} \qquad \forall f\in F,\ \forall l\in L_s \tag{8.17}$$

$$V_{fl_al_b} \leq Y_{fl_a} \qquad \forall f\in F_2,\ \forall l_a,\ l_b\in L_1\cap L_s,\ \text{with } n_{l_al_b}=1 \tag{8.18}$$

$$V_{fl_al_b} \leq Y_{fl_b} \qquad \forall f\in F_2,\ \forall l_a,\ l_b\in L_1\cap L_s,\ \text{with } n_{l_al_b}=1 \tag{8.19}$$

$$\sum_{l_2\in L_2} Y_{fl_2} + \sum_{l_a,\ l_b\in L_1\cap L_s | n_{l_al_b}=1} V_{fl_al_b} = X_f \qquad \forall f\in F_2 \tag{8.20}$$

$$V_{fl_al_b} \leq Y_{fl_a} \qquad \forall f\in F_4,\ \forall l_a,\ l_b\in L_2\cap L_s,\ \text{with } n_{l_al_b}=1 \tag{8.21}$$

$$V_{fl_al_b} \leq Y_{fl_b} \quad \forall f \in F_4, \forall l_a, l_b \in L_2 \cap L_s, \text{ with } n_{l_al_b} = 1 \quad (8.22)$$

$$\sum_{l_a, l_b \in L_2 \cap L_s | n_{l_al_b}=1} V_{fl_al_b} = X_f \quad \forall f \in F_4 \quad (8.23)$$

$$X_f \in \{0,1\} \quad \forall f \in F \quad (8.24)$$

$$Y_{fl} \in \{0,1\} \quad \forall f \in F, \forall l \in L_s \quad (8.25)$$

$$V_{fl_al_b} \in \{0,1\} \quad \forall f \in F, \forall l_a, l_b \in L_s, \text{ with } n_{l_al_b} = 1 \quad (8.26)$$

$$U_t \in \mathbb{N}_{\geq 0} \quad \forall t \in T_s \quad (8.27)$$

$$Z_t \in \mathbb{N}_{\geq 0} \quad \forall t \in T \quad (8.28)$$

Objective Function (8.10) denote the reduced costs and the goal is to find the minimum reduced cost for all the baggage sections. If the reduced costs for all the sections are greater than or equal to zero, the LP-relaxation is solved to optimality. The total amount of groups starting their shift in a section $s \in S$, $\sum_{t \in T} U_t$, is stored into parameter c_{ps}, where $p \in P$ corresponds to the section plan generated by solving the pricing problem of the section $s \in S$. Constraints (8.11) up to and including (8.28) are based on or equal to Constraints (4.2) up to and including (4.21) except from the fact that it is solved per section and decision variables X_{fs}, U_{st} and Z_{st} are changed into X_f, U_t and Z_t. Furthermore, the right hand side of Constraints (8.13), (8.20) and (8.23) is changed from the value 1 into variable X_f to ensure that a flight is assigned to the correct lateral(s) within the section if and only if the flight is indeed assigned to the section.

After solving the pricing problems, the costs and outcomes need to be stored into the parameters for the restricted master problem. Because the pricing problem is solved separately for each of the sections, while using the same dual variables π_f, it might be possible that it results in the same solution. In case multiple pricing problems result in the same solution, a section plan is created from this solution which is allowed to be assigned to the baggage sections which pricing problem resulted in this specific solution. Costs for assigning the section plan to a certain section are stored for the corresponding baggage sections as well.

8.2.3. Retrieving Integer Solution

As stated before, column generation should be embedded into branch-and-bound in order to find an optimal solution, in case the solution of the LP did not happen to be integral. This subsection describes which branching strategy can be used and is based on the scientific work of Barnhart et al. [6]. A branching strategy is created by Ryan and Foster [22], which could be used for the problem in this research. Before the branching strategy is given, Theorem 8.1 states that if the solution to the master problem for a certain section $s \in S$ is fractional, then at least two fligths are assigned a fractional value for section $s \in S$.

Theorem 8.1. *Let X be a 0-1 matrix and let the basic solution to $X\lambda = \mathbb{1}$ be fractional, i.e., at least one of the components of λ is fractional. Assume that all columns of X are distinct and non-zero. Then, there exist two rows, f and f' of the master problem such that*

$$0 < \sum_{p: x_{fp}=1, x_{f'p}=1} \sum_{s \in S} \lambda_{ps} < 1.$$

Proof. Consider the fractional variable $\sum_{s \in S} \lambda_{ps}$ and any row f with $x_{fp} = 1$. Since $\sum_{p \in P} \sum_{s \in S} x_{fp} \lambda_{ps} = 1$ and the variable $\sum_{s \in S} \lambda_{ps}$ was considered to be fractional, there must exists some other basic column p' with $0 < \sum_{s \in S} \lambda_{p's} < 1$ and $x_{fp'} = 1$. Since all columns of X are distinct, there exists another row f' for which $x_{f'p} = 1$ or $x_{f'p'} = 1$, but not both, because otherwise it would be a duplicated column in the matrix X. This is illustrated in Table 8.1.

	p	p'
f	1	1
f'	0	1

Table 8.1: Illustration of the possible candidate branching pairs

This leads to the following sequence of relations:

$$\begin{aligned} 1 &= \sum_{p \in P} x_{fp} \sum_{s \in S} \lambda_{ps} \\ &= \sum_{p: x_{fp}=1} \sum_{s \in S} \lambda_{ps} \\ &> \sum_{p: x_{fp}=1, x_{f'p}=1} \sum_{s \in S} \lambda_{ps} \end{aligned}$$

where the strict inequality follows from the fact that either $\sum_{s \in S} \lambda_p s > 0$ or $\sum_{s \in S} \lambda_{p's} > 0$ is included in the last summation, but not both. □

Theorem 8.1 results in the branching constraints

$$\sum_{p: x_{fp}=1, x_{f'p}=1} \lambda_p = 1 \quad \text{and} \quad \sum_{p: x_{fp}=1, x_{f'p}=1} \lambda_k = 0$$

In the context of the (restricted) master problem of Subsection 8.2.1, two flights can be assigned to the same section in the first (left) branch and to different sections in the second (right) branch.

8.2.4. Summary of CG with Branch-and-Bound

Figure 8.1 illustrates the outline of the total branch-and-price algorithm. The results of using column generation to solve the problem of this part of the research, are evaluated in Section 11.5.

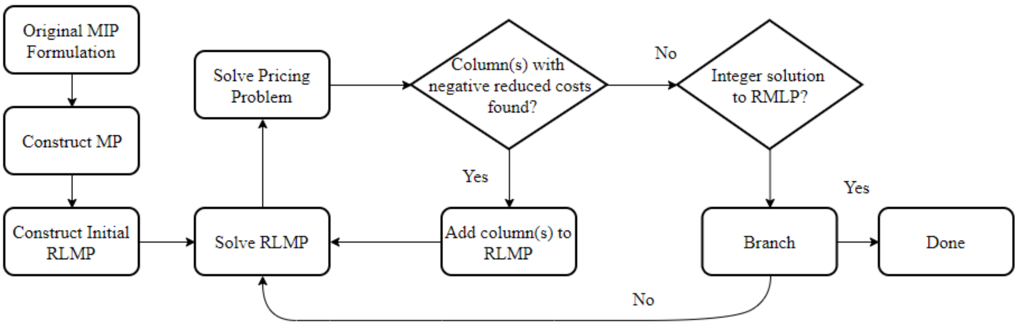


Figure 8.1: Outline of Branch-and-Price

9
Data Description

This chapter describes the data provided by KLM. KLM uses flight schedules in which the flight number, the expected number of baggage units, the scheduled departure gate and the scheduled time of departure are given. As described in Subsection 1.3.1, the flight characteristics for each flight number are given, which indicate the type of aircraft, type of flight, open and closing times of laterals for every hall and the amount of required laterals.

In order to evaluate the methods on a real data set, the flight schedule of week 28 of 2018 is used in this research, which starts on Monday July 9 and ends on Sunday July 15. It is chosen to pick a flight schedule from the summer period, as these schedules are busy compared to other seasons. Week 28 was chosen randomly from all the weeks in the summer period. In total, the baggage of 3306 flights was scheduled to be handled by KLM during this week, which is around 472 flights per day on average. Given these 472 flights on average per day, around 60,000 baggage units were expected to be handled every day. As explained in Subsection 1.3.1, drop off profiles indicate the percentage of baggage units belonging to a certain flight that is expected to 'drop' during a time slot. These time slots have a length of $\delta = 5$ minutes.

As mentioned in more detail in Subsection 1.3.1, KLM operates their outbound baggage process in three halls, namely the D, E and South hall. Table 9.1 provides the amount of laterals and the shift capacity of each of the halls. Figure 9.1 provides a detailed lay out of the baggage halls, along with the baggage sections and laterals. For each hall, it can be obtained which baggage sections are located within this hall and which laterals are located in each baggage section. Also the lateral capacity of all the laterals and the neighbors of each lateral can be obtained. Note that the laterals in the south hall can not be seen as neighbors of each other, meaning that only flights that require one lateral can be located in this baggage section. Furthermore, baggage section 2.4 contains three carousels which are no neighbours either. Three flights which require at most two laterals can be handled on one carousel at the same time. All the other laterals can only handle at most one flight at a time.

Hall	**Number of aterals**	**Baggage units per hour**
D	32	113
E	89	110
South	19	113

Table 9.1: Data of the baggage halls in which KLM operates their outbound baggage process

In this research, it is assumed that a working shift may start every fifteen minutes, so every quarter and that a group of employees consists of just one employee. The shift duration is 8.5 hours, which include half an hour of lunch break that does not need to be scheduled beforehand as described in Section 3.1.

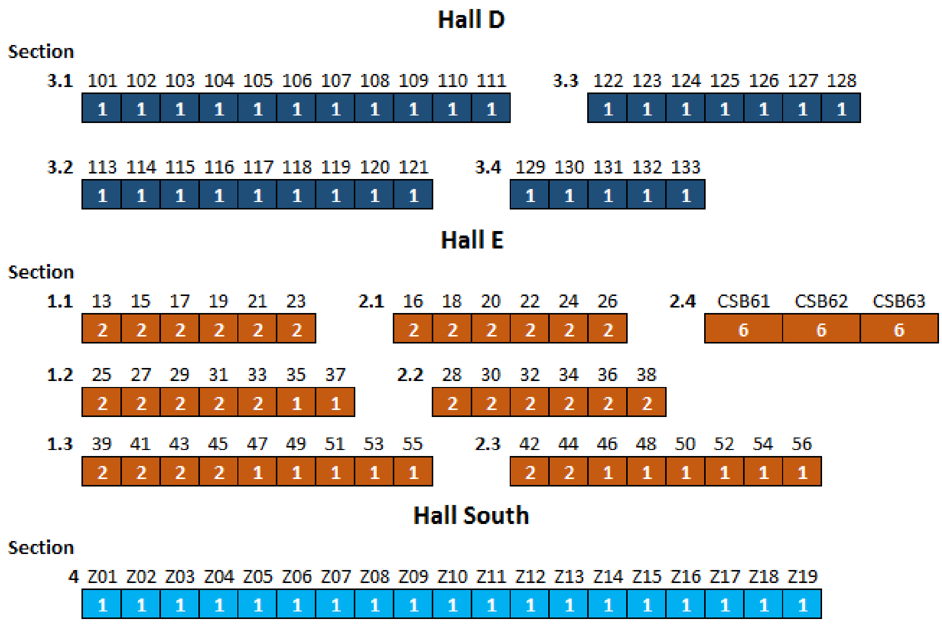


Figure 9.1: Lay out of the laterals and baggage sections within every baggage hall along with the capacity of each lateral

Three smaller data sets are created, to test the effect of the possible speed ups. Each data set represents one day of the given week flight schedule. A subset of flights is randomly chosen from a set of flights from a randomly chosen day, where the first and last scheduled flight are no more than six hours apart. The first small data set contains 28 flights, the second small data set 44 and the third small data set consist of 46 flights. The shift duration is decreased to two hours, because it is not possible to have 8.5 hour working shifts with a set of flights scheduled to departure within a 6 hour interval.

10 Evaluation Criteria

As already stated in Chapter 1, the aim of this research is to minimize the number of employees working on the outbound baggage process while accurately modeling reality. To evaluate the different models and created schedules, some Key Performance Indicators (KPIs) are defined in this chapter which are divided in two main criteria: solution quality and model characteristics.

10.1. Solution Quality

The method resulting in the best objective value, i.e., the least number of working groups is preferred. KLM uses regular computers and they prefer to obtain schedules within reasonable time and therefore the computation time of the methods is of great importance. A time limit could be used by KLM for all the methods, which will turn the methods of Chapters 4, 5 and 6 into heuristic methods instead of exact methods. In order to measure the solution quality of the schedules constructed by the different methods, the following indicators are used:

- *Number of working employees*: The main objective of the methods is to minimize the number of groups of employees working on the outbound baggage process. The number of employees working in the baggage sections should be minimized. Minimizing the number of employees results in reduced costs and a fair distribution of the workload, because the work is divided more equally among the employees.

- *Computation time*: In the extension for planning on lateral level, several possible speed-ups and other methods are introduced, which could improve the efficiency of the method. Therefore, it needs to be tested which combination of these speed-ups have a positive effect on the computation time. A low computation time is an advantage for KLM, because it can easily create different schedules in a shorter amount of time in order to evaluate the schedules.

10.2. Model Characteristics

Comparing the models based on other characteristics is also interesting, because it might provide insight into why certain methods do not solve to optimality while others do within a certain time limit. The following model characteristics indicators are used:

- *Problem size*: Formulations described in Chapters 4, 5, 6 and 7 differ in size. The size of a problem is reflected by the number or variables and constraints of the search three and could affect the computation time to solve a model.

- *Convergence speed of the best lower bound*: During the solving procedure of each model, different values of the best lower bound are evaluated. Eventually, these values should converge to the optimal value. The best lower bound of the methods will approach the optimal value from below. The converging behaviour is of interest since it indicates the proportion of time the solver spends on proving that the optimal solution has been found.

- *Convergence speed of the objective function value*: The objective value of the methods will approach the optimal value from above. If the solver spends a considerably large proportion of time on the proof of optimality, then the solution method can be interrupted at an earlier stage and still ending up with an optimal or near optimal solution, although there is no proof that it is in fact the optimal solution.

11

Computational Results

This chapter presents the results of the methods described in Chapters 4, 5, 6 and 7 evaluated on the criteria introduced in Chapter 10 on the data instances discussed in Chapter 9. Section 11.1 describes all the results on the different data sets of the methods described in Sections 4.2, 5.1 and 6.2. The results of the hierarchical solution method described in Chapter 7 are discussed in Section 11.2 and Section 11.3 summarizes the results for the binary program with the different symmetry breaking constraints which were described in Section 5.2. Subsequently, the results of removing the decision variables X_{fs} from the MIP as described in Section 5.3 are summarized in Section 11.4. Subsequently, Section 11.5 evaluates the results for the column generation approach. The conclusion of all the evaluations on the results presented in this chapter is given in Section 11.6.

Computations are done on a computer with an Intel ®Core (TM) i7-7600U Processor running on 2.9 GHz with 16 GB RAM memory. The models are implemented in AIMMS, using the CPLEX 12.8 solver. A maximum running time of half an hour, 1800 seconds, is used for the problems in this part of the research. It is indicated with a '*', if the program was interrupted because this stopping criteria was reached.

An example of a resulting schedule can be found in Figure 11.1 for the first small data set. Each flight type has a colored bar, where the green bars represent the Transavia flights, blue flights are European flights, pink ones are Commuter flights and yellow ones are the ICA flights. The x-axis represents the time in hours, so the first flight of this data set departs at 6 AM and the last flight is scheduled to depart at noon. The laterals that are used are shown on the y-axis.

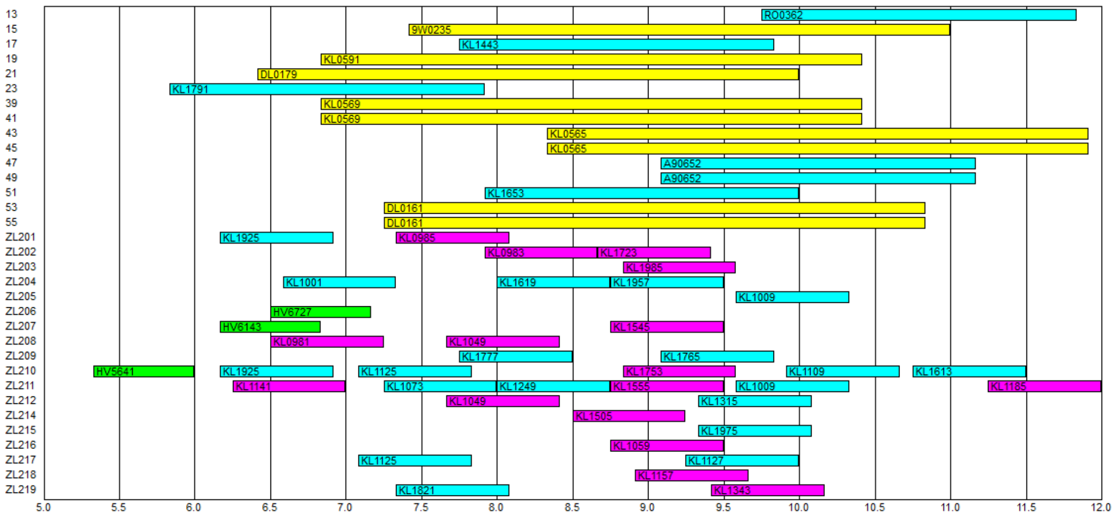


Figure 11.1: Optimal lateral planning for data set 1

11.1. MIP Formulation and Speed-Ups

This section presents the results of the MIP on lateral level as described in Section 4.2, along with the small adjustment to this MIP described in Section 5.1 and the valid inequalities of Section 6.2 that can be added to this MIP. Before the results are presented, the six different constraint settings on which the data sets are tested are introduced and described in Subsection 11.1.1 and Subsection 11.1.2 describes the four different priority settings.

The results of the data instances described in Chapter 9 are summarized for all these constraint and priority settings, but also for the two different Objective Functions (4.1) and (5.1) as was described in Section 5.1, so 144 results[1] are evaluated. The computation time, the time until the optimal objective value is reached and the percentage of time proving optimality are presented in Appendix D.1 for all the results. The objective function that results in the least computation time for the different settings on the data instances, is chosen in Subsection 11.1.3. Subsequently, in Subsection 11.1.4 the best combination of constraint setting and priority setting is chosen. Last, Subsection 11.1.5 will describe the model characteristics.

11.1.1. Constraint Settings

Different combinations of adding extra constraints to the MIP of Section 4.2 are evaluated. These extra constraints were introduced in Section 5.1 and 6.2, and are shortly recapped:

1. **Lateral Capacity** (4.5): These constraints became redundant as described in Subsection 5.1.2

$$\sum_{f \in F} rq_{fst} \cdot X_{fs} \quad \leq \quad lc_{st} \qquad \forall s \in S, \forall t \in T$$

2. **Baggage Valid Inequalities 1** (6.1): Provides a lower bound on the number of working groups per flight and time slot

$$\sum_{s \in S} Z_{st} \quad \geq \quad \left\lceil \sum_{f \in F} mz_{ft} \right\rceil \qquad \forall t \in T$$

3. **Baggage Valid Inequalities 2** (6.9): Provides a more advanced lower bound on the number of working groups per time slot

$$\sum_{s \in S} Z_{st} \quad \geq \quad mw_t \qquad \forall t \in T$$

The different combinations of these constraints are described as different constraints settings and these are given in Table 11.8. An "x" value indicates whether the constraints are used in solving the problem. Note that Constraints (6.1) and (6.9) are not used in the same setting, because Constraints (6.9) are an improved version of Constraints (6.1).

Constraint Settings	Lateral Capacity (4.5)	Baggage 1 (6.1)	Baggage 2 (6.9)
1			
2	x		
3		x	
4			x
5	x	x	
6	x		x

Table 11.1: Possible combinations of the different extra constraints for the MIP

11.1.2. Priority Settings

Different priority settings indicating on which decision variables to branch first are tested as described in Section 5.1. Table 11.2 presents the four different priority settings used in this research. In the first setting, no priorities are used. In the second and fourth setting , the variables X_{fs} are given the highest priority. One of

[1]6 combinations of constraint settings times 4 priority settings times 3 data sets times 2 objectives equals 144 results

the reasons for assigning the highest priority to these variables is that the decision variables U_{st} and Z_{st} are (in)directly linked to the decision variables X_{fs} via Constraints (4.3) and (4.4), meaning that the value of X_{fs} affects the objective function. Another reason for assigning the highest priority to these variables is based on the examples given in the AIMMS manual [8], from which is concluded that the decision variables X_{fs} must have a bigger priority than the decision variables $V_{fl_al_b}$ and Y_{fl}. This is because it must first be determined whether or not flights can be assigned to a section, before it is assigned to laterals within that section.

In the second priority setting, the decision variables U_{st} are given the second highest priority setting. These variables are minimized in the objective function and can only be assigned value one under certain conditions, and therefore, it is important that these variables are considered soon by the branch-and-cut process such that these conditions are certainly met.

Priority Settings	U_{st}	$V_{fl_al_b}$	X_{fs}	Y_{fl}	Z_{st}
1	0	0	0	0	0
2	1	0	2	0	0
3	1	0	1	0	0
4	0	0	1	0	0

Table 11.2: Priority values given to the decision variables in the different priority settings

In the third setting, the decision variables U_{st} and X_{fs} are assigned the same priority and the other decision variables have no priority. In the fourth setting, only decision variables X_{fs} have a priority, meaning that these variables must be branched on first if they take a fractional solution value at a given node.

11.1.3. Choosing the Best Objective Function

The effect of the two different Objective Functions (4.1), $\sum_{s\in S}\sum_{t\in T} U_{st}$ and (5.1), $\sum_{s\in S}\sum_{t\in T} Z_{st}$ is compared on the computation time and the time until the optimal objective value is reached for all the results. The computation time is unanimously less when using Objective Function (5.1), which minimizes the number of working groups starting a shift. This objective function is also more efficient when examining the time until the optimal objective value is reached. In 71% of the cases, it is faster than when the Objective Function (5.1) is used.

In order to give a possible explanation for the different computation times for the different objective functions, the branch-and-cut process of CPLEX is examined in more detail based on the information provided by IBM [19]. In order to solve all the continuous sub problems in the branch-and-cut algorithm, a tree is build by CPLEX in which each sub problem corresponds to a node. The root of the tree is the continuous relaxation of the original MIP problem. CPLEX uses a node selection parameter which indicates which unexplored node needs to be selected first. By default, this setting is set to the Best Bound Search [19], which states that the node with the best objective function for the associated LP-relaxation will be selected, generally near the top of the tree. It is left outside the scope of this research to check if this is indeed (one of) the reason(s) of the different computation times for the different objective functions.

The dual problem is different for the MIPs with the different objective functions and these dual problems are used in the solving. Therefore, the different dual problems for the different objective functions could also explain the different computation times. Further research needs to be done to investigate whether the different dual problems are the reasons of the different computation times.

All the results evaluated further on in this report are obtained while using Objective Function (5.1), as using this objective function resulted in better computation times.

11.1.4. Evaluating the Constraint and Priority Settings

The best combination of constraint and priority setting is chosen in this section based on the results obtained while using Objective Function (5.1). For each data set and combination of constraint and priority setting, the computation time and the time until an optimal solution is found is evaluated. This is done by taking the ratio of these values to the values of the first constraint setting and first priority setting for each data set and then taking the average over the data sets. These ratio scores are given for all three data sets in Appendix D.2. The average ratio scores over all three data sets is given in Table 11.3.

Average Over Data Sets	**Computation Time**				**Time Until Optimal Objective Value is Reached**			
Constraint Setting	**Prio 1**	**Prio 2**	**Prio 3**	**Prio 4**	**Prio 1**	**Prio 2**	**Prio 3**	**Prio 4**
1	1.00	1.23	1.36	0.99	1.00	1.36	1.54	1.09
2	1.07	0.83	2.22	1.33	0.97	0.87	2.16	1.51
3	1.46	**0.82**	2.36	1.40	1.46	**0.73**	2.62	1.56
4	2.54	1.13	3.17	1.44	2.30	1.00	2.99	1.50
5	1.22	1.00	1.64	1.32	1.00	0.89	1.58	1.36
6	2.01	1.29	2.83	1.34	2.35	1.09	3.06	1.12

Table 11.3: Average ratio scores over all three data sets where the ratio is determined for all Constraint and Priority settings compared to the values of the first constraint and first priority setting

Constraint setting 3 combined with priority setting 2 results in the best average ratio score for both the computation time and the time until the optimal objective value is reached. Contrary to what was expected, it is on average more efficient to add Valid Inequalities (6.1) than adding Valid Inequalities (6.9). Table 11.4 illustrates the lower bounds on $\sum_{s\in S}\sum_{t\in T} Z_{st}$ provided by both valid inequalities, along with the value of $\sum_{s\in S}\sum_{t\in T} Z_{st}$ itself. It also provides the optimal solution for Objective Function (5.1). The lower bound provided by Valid Inequalities (6.9) is indeed a bit stronger than for Valid Inequalities (6.1). However, it is still not representative for the actual minimal sum over Z_{st}. Solving the LP-relaxation while minimizing $\sum_{s\in S}\sum_{t\in T} Z_{st}$ results in the same objective value for either using none of these valid inequalities or one of them. It might be caused by the pre-solve of CPLEX that adding Valid Inequalities (6.1) is more efficient in most of the cases, however, further research needs to be done to investigate this.

Data Set	**Lower Bound by Valid Inequalities (6.1)**	**Lower Bound by Valid Inequalities (6.9)**	$\sum_{s\in S}\sum_{t\in T} \boldsymbol{Z_{st}}$	**Obj. Value** $\sum_{s\in S}\sum_{t\in T} \boldsymbol{Z_{st}}$ **for LP-relaxation**	$\sum_{s\in S}\sum_{t\in T} \boldsymbol{U_{st}}$
1	102	106	414	223.31	18
2	133	144	483	327.18	21
3	142	152	575	348.40	25

Table 11.4: Provides the lower bounds for and the value of $\sum_{s\in S}\sum_{t\in T} Z_{st}$ in the MIP and LP for all three data sets

When evaluating the different priority settings based on the ratios presented in Table 11.3, it can be concluded that priority setting 3 is the least effective. This indicates that decision variable X_{fs} must have a higher priority than U_{st}, in case at least one of the two variables is given a priority. As priority setting 2 is more efficient than setting 4 on the data sets, it is concluded that it is efficient to branch on variables U_{st} after branching on X_{fs} instead of some of the other variables. It can not be guaranteed that priority setting 2 is the best setting, as there are many different priority settings that are not tested. Further research needs to investigate which priority settings are most efficient. This could be done by investigating the search tree without using any priorities.

Priority setting 2 and constraint setting 3 are chosen as the best combination for solving the MIP on lateral level. When "MIP on lateral level" will be mentioned later in this chapter, it will refer to the MIP including these two settings.

11.1.5. Model Characteristics

Compared to the convergence speed of the objective function, the MIP provides a weak LP-relaxation. Figure 11.2 plots the best bound and objective value after a certain time. It can be obtained that the best bound converges slowly to the optimal value. Unfortunately, the valid inequalities added in this research did not speed up the LP-relaxation. For all the data instances, it is determined what percentage of the time the solver was busy proving that the optimal objective value found was indeed optimal. These percentages are shown in Appendix D.1. For all the data instances, the convergence speed of the objective function value was better than for the best lower bound and for only a few cases the speeds were equal. Hence, this suggests that the solution process can be interrupted at an earlier stage without ending up with a higher objective value. However, optimality has not been proven at that point.

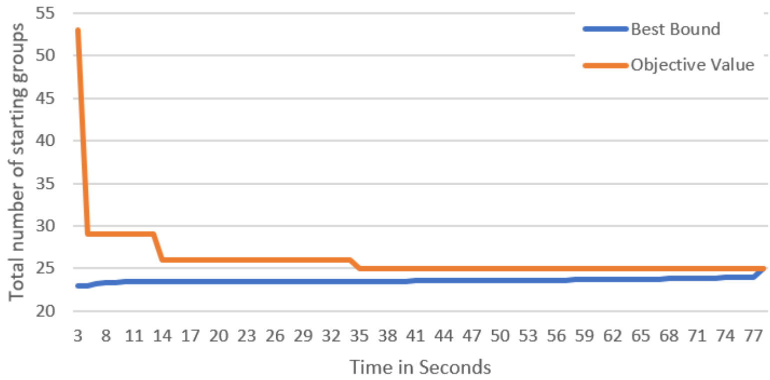


Figure 11.2: Plot of the objective value and best bound, for constraint setting 3, priority setting 2 and data set 3 for the MIP on lateral level

Table 11.5 denotes the number of constraints and variables of the model for the different data sets and the different constraint settings as described in Subsection 11.1.1. The extra constraints for setting 2 until 6 compared to setting 1 are inline with the expectations. Furthermore, there is only one non-integer variable as expected, namely the objective function. The number of variables that will be mentioned later in this thesis, all consist of just one non-integer variable. A high number of integer variables can lead to a large search tree, in case the LP-relaxation leads to many fractional variable values.

Data Set	**#Constraints**						**#Variables**	**#Integer Variables**
	Setting 1	**Setting 2**	**Setting 3**	**Setting 4**	**Setting 5**	**Setting 6**		
1	87,646	88,728	87,746	87,746	88,828	88,828	41,174	41,173
2	72,529	73,579	72,626	72,626	73,676	73,676	34,705	34,704
3	74,867	76,134	74,984	74,984	76,251	76,251	35,096	35,095

Table 11.5: Model sizes for the different constraint settings as described in Subsection 11.1.1 on the three data instances

11.2. Hierarchical Method

The hierarchical solution method as described in Chapter 7 is evaluated in this section. Table 11.6 is used to compare the speeds of the hierarchical method on the three data instances compared to the regular MIP on lateral level. The time for solving the total hierarchical method is presented, however the second phase solves within less than 0.05 seconds for all three data instances. Therefore, it mostly indicates the time of solving the first phase MIP and the iterative method procedure.

	Computation Time			**Time Until Optimal Objective Value is Reached**		
Data Set	**1**	**2**	**3**	**1**	**2**	**3**
MIP on Lateral Level	45	41	79	40	28	36
Total Hierarchical Method	19	16	12	14	16	11

Table 11.6: Computation time in seconds for the MIP problem on lateral level and the hierarchical method

Solving the hierarchical method is much faster than solving the MIP on lateral level. This can be partially explained by comparing the model sizes which are summarized in Table 11.7. The number of constraints and variables decreases significantly for the hierarchical solution method. The number of constraints and variables in the second phase of the hierarchical solution method, differ per section and depends on the outcomes of the first phase. These numbers are given in Appendix D.3 for the final optimal and feasible solution and the maximum, minimum and average amount of these constraints and variables are given in Table 11.7. In case that no flights are assigned to a section in the first phase, then the second phase consists of only one variable and one constraint, which is the objective function. The MIP on lateral level is simplified by the hierarchical method and inline with the expectations, the hierarchical method results in a decrease in computation time, constraints and variables.

	Data Set 1	Data Set 2	Data Set 3
# **Constraints MIP Lateral Level**	87,746	72,626	74,984
# Total Constraints Hierarchical Method	6,320	5,764	7,246
# Constraints First Phase	3,601	3,706	4,363
# Constraints in Total for Second Phase	2,719	2,058	2,883
Max # Constraints for Second Phase	1,112	1,433	1,695
Min # Constraints for Second Phase	1	1	1
Average # Constraints for Second Phase	226.6	171.5	240.3
# **Variables MIP Lateral Level**	41,174	34,705	35,096
# Total Variables Hierarchical Method	1,037	1,506	1,641
# Variables First Phase	606	781	859
# Variables in Total for Second Phase	431	725	782
Max # Variables for Second Phase	267	609	571
Min # Variables for Second Phase	1	1	1
Average # Variables for Second Phase	35.9	60.4	65.2

Table 11.7: Model sizes for the first and second phase of the hierarchical method compared to the model size of the MIP on lateral level

11.3. Binary Programming and Symmetry Breaking

Solving the problem while using the binary program with different combinations of symmetry breaking constraints as described in Section 5.2 is evaluated in this section. The two different symmetry breaking constraints (SBCs) are shortly recapped:

1. **Symmetry Working Groups** (5.5): assign groups to working shifts in an increasing order
2. **Symmetry Within Section** (5.6): ensure that in each baggage section, the groups that are working are assigned in an increasing order

The different combinations of these constraints are described as different constraints settings and given in Table 11.8. An "x" value indicates whether the constraints are used in solving the problem.

SBC Setting	Symmetry Working Groups (5.5)	Symmetry Within Section (5.6)
1		
2	x	
3		x
4	x	x

Table 11.8: Possible combinations of the different extra constraints for the MIP

When using the binary program, the number of working groups that are available must be set first, as described in Section 5.2. Two different number of working groups are tested, namely 30 and 50 groups, in order to examine the effect. These two number of working groups are chosen based on the outcomes of the previous methods, such that it is known that there are enough groups to fulfill the work. In practice, it is hard to give a good estimation of the number of working groups needed, so it is more likely to choose a big number of groups to ensure a feasible solution. The results for both number of workings groups and SBC settings are evaluated for all three data instances. This is done by taking the ratio of these values to the value of the first SBC setting and when setting the number of groups to 30. These ratio scores are given for all three data sets in Appendix D.5. The average ratio scores over all three data sets is given in Table 11.9, where the scores that are indicated by ‘**’ are not completely fair, because the program was interrupted by the time limit in at least one of the data sets for this combination of settings.

Average	Computation Time		Time Until Optimal Objective Value is Reached	
SBC Setting	30 Groups	50 Groups	30 Groups	50 Groups
1	**1,00**	2,03	**1,00**	1,82
3	2,00	3,59	2,40**	4,28**
2	2,66	3,07	3,03**	3,57**
4	3,80**	3,84**	4,69**	4,92**

Table 11.9: Average ratio scores over all data sets of all symmetry breaking constraints settings compared to the values of the first constraint and first priority setting

It can easily be obtained that setting 1 along with 30 groups is the most efficient setting for the binary program. The computation times and time until the optimal objective value is reached for the three data sets using this setting and number of groups are given in Table 11.10, compared to the times of the previous evaluated methods. Binary programming as described in Section 5.2 is not speeding up the computation time compared to the other methods.

	Computation Time			Time Until Optimal Objective Value is Reached		
Data Set	**1**	**2**	**3**	**1**	**2**	**3**
MIP on lateral level	45	41	79	40	28	36
Total Hierarchical Method	19	16	12	14	16	11
Binary Programming	606	299	714	400	269	505

Table 11.10: Computation time in seconds for the already evaluated methods

When using more groups, the model becomes less efficient. Inline with the expectations, the number of constraints and variables increases rapidly when the number of groups increases, as can be obtained by using Table 11.11. As mentioned before, it is hard to find a good estimation of the number of groups needed, so either a big enough number of groups should be taken to ensure feasibility or the chosen number of groups should be increased when it results in an infeasible solution.

Data Set	# Groups	# Constraints				#Variables
		Setting 1	Setting 2	Setting 3	Setting 4	
1	30	47.334	47.664	47.642	47.671	45.516
1	50	80.364	81.024	81.002	81.061	87.426
2	30	47.333	47.663	47.641	47.670	45.780
2	50	79.373	80.033	80.011	80.070	86.340
3	30	56.071	56.401	56.379	56.408	54.494
3	50	94.711	95.371	95.349	95.408	103.664

Table 11.11: Model sizes for the different SBC settings and group sizes on the three data instances

Figure 11.3 illustrates the plot of the best bound and objective value for the best setting and number of groups for the binary programming method for data set 1. The first feasible solution is found after 44 seconds, while after this time the hierarchical method would have already been solved and one second later also the MIP on lateral level. It can be concluded that the binary program does not speed-up the computation time.

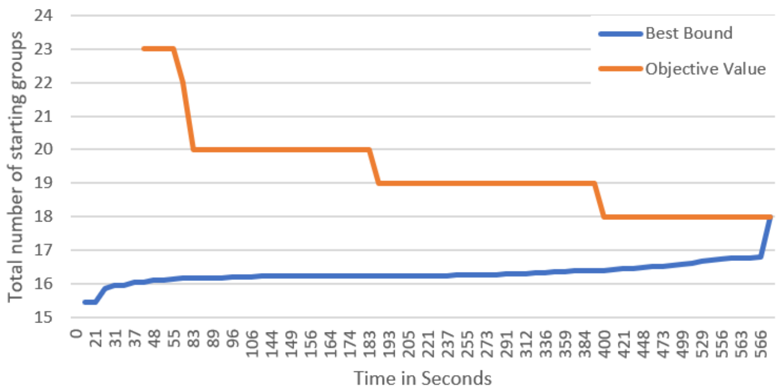


Figure 11.3: Plot of the objective value and best bound for SBC setting 1, 30 groups and data set 1 for the binary programming method

11.4. Removing Decision Variables X_{fs}

As described in Section 5.3, the decision variables X_{fs} can be removed from the MIP model, because its value can be directly derived from the decision variable Y_{fl}. By removing these variables, the amount of variables is expected to decrease. However, since there are more laterals compared to sections, it is also expected that the number of constraints will increase as the domain of some constraints change. Table 11.12 shows that these expectations are met.

Data Set	**1**	**2**	**3**
Computation Time	1800*	1800*	1800*
Time Until Optimal Objective Value Is Reached	109	272	56
# Constraints	12,582	11,963	14,207
# Variables	4,695	6,077	6,457

Table 11.12: Computation time in seconds

Furthermore, it is obtained that the model did not solve within the time limit for all the data instances. The time until the optimal objective value is reached is also worse than when using the MIP on lateral level or the hierarchical method, as can be obtained in Table 11.13. Therefore, it can be concluded that this potential speed-up has not resulted in an actual speed-up.

	Computation Time			**Time Until Optimal Objective Value is Reached**		
Data Set	**1**	**2**	**3**	**1**	**2**	**3**
MIP on lateral level	45	41	79	40	28	36
Total Hierarchical Method	19	16	12	14	16	11
Binary Programming	606	299	714	400	269	505
Removing Decision Variables X_{fs}	1800*	1800*	1800*	109	272	56

Table 11.13: Computation time in seconds for the already evaluated methods

Figure 11.4 plots the convergence speed of the best bound and objective value of this method for the third data set. Again, the best lower bound has a worse convergence speed than the objective function.

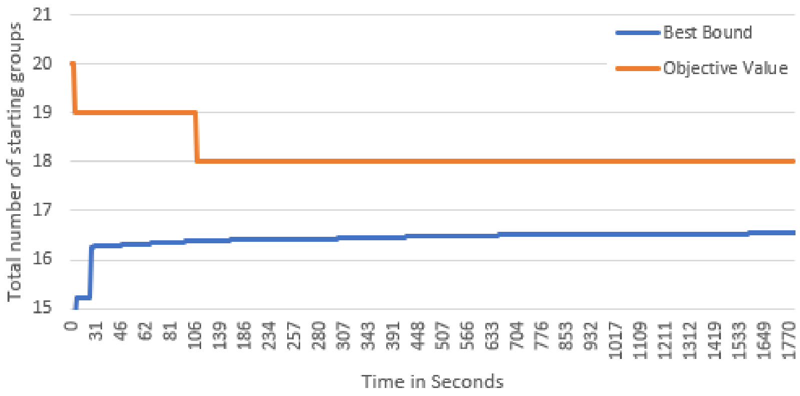


Figure 11.4: Plot of the objective value and best bound for data set 3 with the method where the decision variables X_{fs} are removed

11.5. Column Generation

The results for solving the problem using column generation as described in Chapter 8 are evaluated in this section. Table 11.14 summarizes the computation time for solving the LP-relaxation to optimality, the objective value and whether or not an integer solution can be obtained with the created columns. The computation times for solving the LP-relaxations are already worse than for the MIP on lateral level and the total hierarchical solution method. The created columns for the RMLP did not result in an integer feasible solution for the master problems. The branching strategies of Subsection 8.2.3 could be applied in order to obtain a integer solution, however this is left outside of the scope of this research due to time restrictions.

	Data Set 1	**Data Set 2**	**Data Set 3**
Computation Time for Solving LP-Relaxation	222	393	517
$\sum_{s\in S}\sum_{t\in T} U_{st}$ for LP-relaxation	5.91	6.52	7.75
Integer Feasible Solution Using Created Columns?	no	no	no

Table 11.14: Outcomes of using column generation

11.6. Conclusion

This section shortly summarizes the results described in this chapter and concludes which methods will be used in the remaining parts of this research. Table 11.15 summarizes the outcomes in terms of computation time and time until the optimal objective value is reached for all the methods of Part II on all three small data instances, except for column generation as using this method did not result in an integer solution. Using the entire flight schedule does not result in a feasible solution for the models in this part of the research as was described in Section 1.4. It can be obtained that the MIP on lateral level and the hierarchical solution method are the two most efficient methods, and therefore, these two models are extended for delaying the lateral opening times in Part III and for the driving tasks in Part IV. Priority setting 2 and constraint setting 3 are chosen for the MIP on lateral level.

	Computation Time			**Time Until Optimal Objective Value is Reached**		
Data Set	**1**	**2**	**3**	**1**	**2**	**3**
MIP on Lateral Level	45	41	79	40	28	36
Total Hierarchical Method	19	16	12	14	16	11
Binary Programming	606	299	714	400	269	505
Removing Decision Variables X_{fs}	1800*	1800*	1800*	514	1800*	319

Table 11.15: Computation time in seconds for the already evaluated methods

All the methods showed that the LP-relaxation is weak. The plots which illustrate the convergence speed of the best lower bound and objective value suggest that the solution process can be interrupted at an earlier stage without ending up with a higher objective value. However, the downside of interrupting at an earlier stage is that optimality can not be guaranteed.

III

Delaying Lateral Opening Times

12

Problem Description and Formulation

In order to fit the baggage of all the flights in the schedule, it is necessary to delay the lateral opening times of some of the flights as mentioned in Section 3.3. It is impossible to find a feasible schedule while maintaining all the predefined opening and closing times of the laterals. The lateral opening times can be delayed under certain conditions, whereas the predefined lateral closing times need to be maintained. When delaying the lateral opening time, the baggage that was scheduled to drop before the new opening time will be stored in the buffer.

A detailed description of this extension is given in Section 12.1, whereas the mixed integer programming formulation is given in Section 12.2. The hierarchical solution method from Chapter 7 needs to be adjusted such that it can be used for this part of the research, which is explained in Section 12.3.

12.1. Problem Description

All the possible time slots at which at least one of the flights is allowed to open in one of the baggage sections are defined by the set of opening times $O \subseteq T$. Parameter ao_{fos} is set to one when a flight $f \in F$ is allowed to open on one or multiple laterals in a baggage section $s \in S$ at opening time $o \in O$ and zero otherwise. The set of opening times is determined along with the parameter ao and the conditions for the parameter ao having value one for a flight in a certain baggage section at a specific opening time are as follows:

- The new, possibly delayed opening time $o \in O$ should be between the original lateral opening and closing time for the flight $f \in F$ in the baggage section $s \in S$.
- It must be possible to unload the total amount of baggage within the time that the lateral is opened. This means that the number of baggage per time slot can not be more than the shift capacity.

Only one flight at a time can be on a lateral, and therefore, it must be known when a flight would be on a lateral in case it is assigned to that lateral. Parameter ho_{folt} is set to one when a flight $f \in F$ is on a lateral $l \in L$ at a time slot $t \in T$ in case it is assigned to the lateral $l \in L$ and when the lateral(s) of the flight are opened at time slot $o \in O$, and zero otherwise.

As mentioned before, the baggage which was scheduled before the possibly later opening time is stored in the buffer. However, this baggage must be dropped at a later time slot at which the lateral is open. Therefore, the amount of estimated baggage per time slot will differ per opening time. The estimated number of bags of flight $f \in F$ in baggage section $s \in S$ during time slot $t \in T$, in case the lateral(s) of the flight are opened at opening time $o \in O$, are given by the parameter bo_{fost}. The baggage that is stored in the buffer must be spread out over the time slots at which the lateral(s) are open. As described in the second condition of delaying the lateral opening time, the total amount of baggage per time slot must always be less or equal to the shift capacity sc_s.

Figure 12.1 illustrates an example of determining bo_{fost}. The expected number of baggage units is illustrated by grey blocks, in which the dark grey blocks are the baggage units that would be stored in the buffer for three time slots when delaying the opening time from $o = 1$ to $o = 4$. These dark grey baggage units should be spread over the time slots at which the lateral is open. The stored baggage should preferably be dropped as soon as possible, because the capacity of the buffer is limited. Furthermore, the total number of baggage units per time slot can not exceed the shift capacity, given by the red dotted line. The figure shows where the dark grey bags will be scheduled when opening the lateral at $o = 4$.

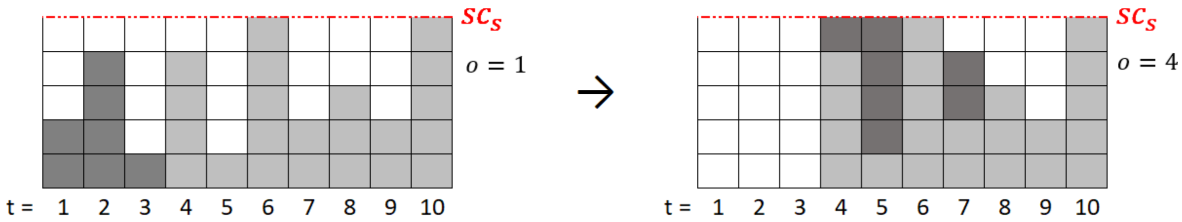


Figure 12.1: Example of how the parameter b_{fost} is determined

The difference in time slots between the original lateral opening time slot and the chosen opening time slot $o \in O$ for a flight $f \in F$ in baggage section $s \in S$ is given by parameter d_{fos}. A penalty γ will be given in the objective function on how many flights in total are delayed and by how many time slots. The later the opening time gets, the more time slots will have a number of estimated baggage units equal to the shift capacity. In practice, this is not robust as there will be not much slack. For this reason, it would be better, for example, to delay two flights by two time slots instead of one flight by four time slots. Therefore, d_{fos}^2 will be used in the objective function. Furthermore, the total number of groups of employees starting a shift will be minimized while having penalty α.

12.2. MIP Formulation

This section contains the description of the new or changed sets, parameters and decision variables as well as the MIP for this extension. A complete overview of this MIP is given in Appendix B.

Sets

O set of possible opening time slots for laterals, $O \subseteq T$

Parameters

$$ao_{fos} = \begin{cases} 1 & \text{if the lateral(s) of flight } f \in F \text{ is (are) allowed to open at time slot } o \in O \text{ in section } s \in S \\ 0 & \text{otherwise} \end{cases}$$

bo_{fost} estimated number of baggage units from flight $f \in F$ in section $s \in S$ during time slot $t \in T$ when opened at opening time $o \in O$

d_{fos} difference in time slots between the original lateral opening time slot and the opening time slot $o \in O$ for flight $f \in F$ in section $s \in S$

$$ho_{folt} = \begin{cases} 1 & \text{if flight } f \in F \text{ is on lateral } l \in L \text{ at time slot } t \in T \text{ in case it is assigned to lateral } l \in L \\ & \text{and in case it is opened at time slot } o \in O \\ 0 & \text{otherwise} \end{cases}$$

α, γ weights used in the objective function

Decision variables

$$XO_{fos} = \begin{cases} 1 & \text{if the lateral(s) of flight } f \in F \text{ open at time slot } o \in O \text{ in section } s \in S \\ 0 & \text{otherwise} \end{cases}$$

$$YO_{fol} = \begin{cases} 1 & \text{if flight } f \in F \text{ is assigned to lateral } l \in L \text{ and the lateral opens at time slot } o \in O \\ 0 & \text{otherwise} \end{cases}$$

Mixed integer program

$$\min \quad \underbrace{\alpha \sum_{s\in S} \sum_{t\in T} U_{st}}_{\text{Obj. Part I}} + \underbrace{\gamma \sum_{f\in F} \sum_{o\in O} \sum_{s\in S} d^2_{fos} \cdot XO_{fos}}_{\text{Obj. Part II}} \tag{12.1}$$

$$\text{s.t.} \quad \sum_{s\in S} \sum_{o\in O} XO_{fos} = 1 \qquad \forall f \in F \tag{12.2}$$

$$XO_{fos} \leq ao_{fos} \qquad \forall f \in F,\ \forall o \in O,\ \forall s \in S \tag{12.3}$$

$$YO_{fol} \leq XO_{fos} \qquad \forall f \in F,\ \forall o \in O,\ \forall s \in S,\ \forall l \in L_s \tag{12.4}$$

$$\sum_{f\in F} \sum_{o\in O} bo_{fost} \cdot XO_{fos} \leq sc_s \cdot Z_{st} \qquad \forall s \in S,\ \forall t \in T \tag{12.5}$$

$$\sum_{f\in F} \sum_{o\in O} ho_{folt} \cdot YO_{fol} \leq ma_l \qquad \forall l \in L,\ \forall t \in T \tag{12.6}$$

$$\sum_{o\in O} XO_{f,o,s} = X_{f,s} \qquad \forall f \in F,\ \forall s \in S \tag{12.7}$$

$$\sum_{o\in O} YO_{fol} = Y_{fl} \qquad \forall f \in F,\ \forall l \in L \tag{12.8}$$

Constraints (4.4), (4.6), (4.7) and (4.10) - (4.16)

$$X_{fs},\ Y_{fl},\ V_{fl_al_b} \in \{0,1\} \qquad \forall f \in F,\ \forall l,\ l_a,\ l_b \in L,\ \forall s \in S \tag{12.9}$$

$$XO_{fos},\ YO_{fol} \in \{0,1\} \qquad \forall f \in F,\ \forall o \in O,\ \forall l \in L,\ \forall s \in S \tag{12.10}$$

$$U_{st} \in \mathbb{N}_{\geq 0} \qquad \forall s \in S,\ \forall t \in T_s \tag{12.11}$$

$$Z_{st} \in \mathbb{N}_{\geq 0} \qquad \forall s \in S,\ \forall t \in T \tag{12.12}$$

The Objective function (12.1) consists of two parts: minimizing the number of groups starting a shift and the amount of delayed lateral opening times. The latter is done by minimizing the squared difference in time slots between the original and new opening time. The most important part is to minimize the number of employees, but it is also important to not shorten too many lateral openings. A downside of decreasing lateral opening times is that the number of baggage units per time slot will reach the shift capacity more often. This will result in not enough slack as mentioned before. Different values of the weights α and γ can be used in order to reach a good balance between the two goals.

Each flight must have exactly one opening time for its laterals in exactly one baggage section, which is ensured by Constraints (12.2). The laterals of a flight may only be opened in a baggage section if it is allowed, which is ensured by Constraints (12.3). Constraints (12.4) ensure that the laterals of a flight may only open at a certain time slot if the flight opens in the baggage section in which the laterals are located at that specific time slot. Constraints (12.4) are a replacement for Constraints (4.8), used in the MIP on lateral level, which stated that a flight could only be assigned to a lateral if it was assigned to the section in which the lateral is located. It needed to be replaced, because the opening time slot on a lateral must be the same as the opening time slot for the section in which the lateral is located. If the original Constraints (4.8) would be used, the model could just pick a random opening time for the laterals.

Constraints (12.5) and (12.6) have the same function as Constraints (4.3) and (4.9), respectively. However, they are updated to the new parameters including the opening times.

Variables $XO_{f,o,s}$ and $X_{f,s}$ are linked by Constraints (12.7) and variables $YO_{f,o,l}$ and $Y_{f,l}$ are linked by Constraints (12.8). Constraints (12.9) up to and including (12.12) indicate the domain of the decision variables, where a group may only start a shift if it is allowed.

12.3. Hierarchical Solution Method

In order to use the hierarchical solution method which was introduced in Chapter 7, it needs to be adjusted for this extension where laterals opening times may be delayed. These adjustments are described in this section, starting with the first phase MIP. One parameter needs to be adjusted before the new MIP can be formulated. Note that all the complete MIP's are given in Appendix B.

Parameters

$$hos_{fst} \begin{cases} 1 & \text{if lateral(s) for flight } f \in F \text{ would be open in section } s \in S \text{ at time slot } t \in T \text{ in case it is} \\ & \text{assigned to section } s \in S \text{ and opened at time slot } o \in O,\ hos_{fst} = ho_{flt} \text{ with } l \in L_s \\ 0 & \text{otherwise} \end{cases}$$

$$aos_{ifos} \begin{cases} 1 & \text{if flight } f \in F \text{ is supposed to open at time slot } o \in O \text{ in section } s \in S \text{ in infeasible option } i \in I \\ 0 & \text{otherwise} \end{cases}$$

MIP Formulation - "First Phase for Delaying Lateral Opening Times"

$$\min \quad \alpha \sum_{s\in S} \sum_{t\in T} U_{st} + \gamma \sum_{f\in F} \sum_{o\in O} \sum_{s\in S} d^2_{fos} \cdot XO_{fos} \tag{12.13}$$

$$\text{s.t.} \quad \sum_{f\in F} \sum_{o\in O} hos_{fost} \cdot ml_f \cdot XO_{fos} \leq tl_s \qquad \forall s \in S,\ \forall t \in T \tag{12.14}$$

$$\sum_{f\in F} \sum_{o\in O} r_f \cdot hos_{fost} \cdot XO_{fos} \leq lc_{st} \qquad \forall s \in S,\ \forall t \in T \tag{12.15}$$

$$\sum_{f\in F} \sum_{o\in O} aos_{ifos} \cdot XO_{fos} \leq \sum_{f\in F} \sum_{o\in O} aos_{ifos} - 1 \qquad \forall i \in I,\ \forall s \in S, \text{ with } \sum_{f\in F} as_{ifs} > 0 \tag{12.16}$$

Constraints (4.4), (4.20), (4.21), (12.2) - (12.5)

$$XO_{fos} \in \{0,1\} \qquad \forall f \in F,\ \forall o \in O,\ \forall s \in S \tag{12.17}$$

The Objective Function (12.13) is identical to Objective Function (12.1). Constraints (12.14) ensure that the minimum required number of laterals are available for all flights that are assigned to a baggage section and Constraints (12.15) ensure that the lateral capacity is sufficient within a section at every time slot. Constraints (12.16) are added such that infeasible options are not allowed anymore. The adjusted parameter aso_{ifos} stores the opening times for each of the flights in the infeasible option, because other opening times might still be possible for a feasible solution in the second phase. The second phase MIP formulation for this part of the research is given below. The variables X_{fs} and XO_{fos} that are stated in these original constraints, are parameters in this MIP as the values are obtained in the first phase.

MIP Formulation - "Second Phase for Delaying Lateral Opening Times"

$$\min \quad \epsilon \tag{12.18}$$

$$\text{s.t.} \quad \sum_{f\in F} ho_{folt} \cdot YO_{fl} \leq ma_l + P_{lt} \qquad \forall l \in L,\ \forall t \in T \tag{12.19}$$

Constraints (4.6), (4.7), (4.10) up to and including (4.16)

Constraints (12.4), (12.6), (12.8) and (12.10))

(12.20)

The iterative method is only slightly changed compared to the iterative method denoted in Subsection 7.4.2. A neighbor solution can be obtained by replacing a flight of the infeasible set of flights by another flight with the exact same characteristics, meaning that also the opening time should be the same. The full outline of the iterative method is given in Appendix B.

13
Computational Results

The two methods described in Chapter 12 are tested on three data instances, the Monday, Tuesday and Wednesday of the flight week schedule described in Chapter 9. In this chapter, the results are presented and analyzed based on the evaluation criteria described in Chapter 10. A maximum running time of 1.5 hours is used for the problems in this part of the research. If the maximum running time was reached before the model under consideration was solved to optimality, the best objective value up to that point is given and a '*' is added to the computation time.

The main goal of this research is to investigate if the number of working groups can be minimized for the outbound baggage handling process. Therefore, only one setting for the objective weights is tested, namely $\alpha = 1$ and $\beta = 0.001$. KLM is interested whether the number of working groups required can be minimized compared to the current planning, despite the amount of delayed lateral opening times.

Section 13.1 presents the results on the MIP described in Section 12.2 and the results on the hierarchical method, described in Section 12.3 are given in Section 13.2.

13.1. Results for the General MIP

The results for the MIP described in Section 12.2 are presented in Table 13.1. Only one feasible solution and one bound were given within the time limit and especially for the Monday data set, the number of working groups is not realistic and too high. More than half a million constraints and integer variables are needed in the MIP which results in a slow convergence speed of both the best bound and the objective value.

Data Set	**Monday**	**Tuesday**	**Wednesday**
Objective Value	285,072.112	185.796	181.443
LP Bound	72.35	70.93	71.14
Computation Time	5,400*	5,400*	5,400*
Time Until Optimal Objective Value Is Reached	5,400*	5,400*	5,400*
# Constraints	536,450	533,498	537,744
# Variables	566,378	563,652	569,095

Table 13.1: Computation time in seconds

13.2. Results for the Hierarchical Method

Results of the hierarchical method for the Monday, Tuesday and Wednesday data set are given in Tables 13.2, 13.3 and 13.4, respectively. Different iterations of the program are presented, with the corresponding total computation time, the corresponding objective value of the first phase, whether the outcome of the first phase was feasible in the second phase, the LP-Bound and the number of constraints. When an infeasible solution occurs in the second phase, the number of constraints increases for the next iteration as infeasible options are stored. The time of solving the the first phase is almost equal to the difference in total computation time, because each of the second phase problems solve within one second.

Monday

Iteration	Total Comp. Time	Feasible?	Obj. Value	LP Bound	#Constraints
1	43	Yes	31005.440	72.35	9,444
2	213	No	133.458	72.39	9,444
3	773	No	125.312	72.41	9,652
4	1,310	No	127.449	72.41	9,739
5	1,590	No	127.980	72.39	9,792
6	1,628	Yes	125.344	72.35	9,890
7	2,537	No	106.472	72.42	9,890
8	3,532	No	108.216	72.43	9,961
9	4,596	Yes	117.391	72.42	10,003
10	5,303	Yes	**112.048**	72.39	10,003

Table 13.2: Results for the Monday data set for the hierarchical method with delaying the lateral times

The objective value could increase and the LP bound could decrease over the iterations if an infeasible solution was obtained. However, it is not possible to find a worse objective value than objective values of previous iterations which belonged to a feasible solution. The same holds for a worse LP bound.

Tuesday

Iteration	Total Comp. Time	Feasible?	Obj. Value	LP Bound	#Constraints
1	122	No	56,533.200	72.35	15,204
2	248	No	69,975.070	72.35	15,241
3	377	Yes	56,535.340	72.35	15,289
4	511	No	125.432	72.35	15,289
5	575	Yes	49,812.530	72.35	15,342
6	641	Yes	126.843	72.35	15,342
7	1,578	No	107.049	72.42	15,342
8	1,643	No	122.468	72.35	15,381
9	2,562	Yes	114.431	72.43	15,452
10	4,732	Yes	**113.395**	72.45	15,452

Table 13.3: Results for the Tuesday data set for the hierarchical method with delaying the lateral times

Wednesday

Iteration	Total Comp. Time	Feasible?	Obj. Value	LP Bound	#Constraints
1	47	No	1,1684.797	71.14	14,931
2	94	No	164.917	71.14	15,004
3	141	No	1,344.704	71.14	15,062
4	190	No	163.650	71.14	15,148
5	233	No	31,129.120	71.14	15,234
6	275	Yes	31,129.136	71.14	15,375
7	934	No	128.807	71.19	15,375
8	1,782	No	130.546	71.19	15,694
9	2,394	No	126.750	71.19	15,908
10	3,474	Yes	**108.329**	71.21	16,001
11	4,419	No	103.765	71.21	16,001

Table 13.4: Results for the Wednesday data set for the hierarchical method with delaying the lateral times

It is not guaranteed that a feasible solution is found before the total time limit is reached. Therefore, another stopping criteria could be used to ensure that a ‘reasonable’ objective value is obtained. This stopping criteria could interrupt the program when either, the time limit was reached and a feasible solution with a ’reasonable’ objective value was obtained or if this is not the case, wait until a feasible solution with a reasonable objective value is indeed obtained. This ’reasonable’ objective value can be set by KLM, but the value must be chosen in such a way that a feasible solution exists.

Table 15.6 provides some insight on the objective value split into the number of working groups and the number of delayed time slots. It can be obtained that opening times of some of the flights are delayed drastically. For the Tuesday data set, only four flights have delayed lateral opening times, but it is interesting to obtain that the opening time of one flight is delayed for one hour and a half. All three data sets consists of at least one flight that is delayed for over one hour, so either the expected amount of baggage units happens to be low, or it might be a good idea for KLM to reexamine the standard lateral opening times for these flights.

	Monday	**Tuesday**	**Wednesday**
Number of working groups	116	113	106
Maximum number of delayed time slots of all flights	14	18	15
Number of flights which opening times are delayed	42	4	49
Total number of delayed time slots	183	24	279

Table 13.5: Information about the number of working groups and delayed time slots for the different flight schedules

Table 13.6 compares the number of variables, the number of constraints, the objective value and the LP bound for the two methods used in this part of the research. The number of constraints and variables in the second phase of the hierarchical solution method, differ per section and depends on the outcomes of the first phase. These numbers are given in Appendix E for the last iteration and the maximum, minimum and average amount of these constraints and variables are given in Table 13.6. Compared to the general MIP, the number of constraints and variables decreases significantly for the hierarchical solution method. By considering the minimum number of variables in the second phase, it can be concluded that none of the sections are empty.

	Monday	**Tuesday**	**Wednesday**
# **Constraints General MIP**	536,450	563,652	569,095
# Total Constraints Hierarchical Method	35,544	40,536	42,085
# Constraints First Phase	10,003	15,452	16,001
# Constraints in Total for Second Phase	25,541	25,084	26,084
Max # Constraints for Second Phase	700	667	436
Min # Constraints for Second Phase	9,461	9,561	9,147
Average # Constraints for Second Phase	2,128	2,090	2,174
# **Variables General MIP**	566,378	563,652	569,095
# Total Variables Hierarchical Method	77,446	88,942	83,420
# Variables First Phase	63,740	75,248	69,848
# Variables in Total for Second Phase	13,706	13,694	13,572
Max # Variables for Second Phase	10,033	10,223	9,653
Min # Variables for Second Phase	93	107	79
Average # Variables for Second Phase	1,142	1,141	1,131
Objective Value General MIP	285,072.112	185.796	181.443
Objective Value Hierarchical Method	112.048	113.395	108.329
LP Bound General MIP	72.35	70.93	71.14
LP Bound Hierarchical Method	72.39	72.45	71.21

Table 13.6: Model sizes for the first and second phase of the hierarchical method compared to the model size of the General MIP

It can easily be obtained using Table 13.6 that using the hierarchical method reaches a better objective value within the time limit than using the MIP presented in Section 13.1. Hence, it is still not proven that this objective value is optimal, as the gap between the objective value and LP-relaxation is still positive. It is tested to run the program of the Wednesday data set over 13 hours, but after 2 hours, the LP bound and the objective value did not improve any further. The objective value obtained after 2 hours was 105.418 and the LP bound had the value 71.22. The lateral planning for the Tuesday data set is illustrated in Appendix E.

13.3. Conclusion

The hierarchical solution method described in Section 12.3 outperforms the general MIP method described in Section 13.1, using the time limit. The number of constraints and variables decreases significantly for the hierarchical method compared to the general MIP. The second phase resulted to be solved within one second for all the sections, and therefore, further research could focus on speeding up the first phase.

It would be interesting to compare the results with the lateral planning that KLM actually used. However, the latter did not meet all the constraints described in this research. For example, lateral closing times are adjusted such that the laterals closes more early, flights are assigned to laterals they are not allowed to and overlap of flights is allowed in the planning.

IV

Driving Tasks

14

Problem Description and Formulation

In order to minimize the number of groups of employees working on the outbound baggage process, it is necessary to also minimize the number of drivers that transports the baggage to the aircraft. This extension was already mentioned in Section 3.3. A detailed description of this extension is given in Section 14.1, whereas the mixed integer programming formulation is given in Section 14.2. Subsequently, Section 14.3 describes how the hierarchical solution method is adjusted for the extension of adding driving tasks.

14.1. Problem Description

After the baggage is loaded from the lateral into the tugs, the tugs need to be transported to the aircraft. This is done by drivers who assigned to work at a certain baggage section. The possible shift starting times of the drivers are equal to the possible starting times of the employees that are loading the baggage from the laterals into the tugs in that same baggage section. Also, the shift duration of both types of employees are identical.

Per baggage hall and flight type, it is known how often a driving task needs to be performed and at what time before the scheduled departure the driver needs to depart from the lateral. Furthermore, the total driving times from each baggage hall to each gate are known. The total driving time indicates the driving time towards the aircraft plus the unloading time plus the driving time back to the baggage hall. The scheduled departure time and gate are given in the flight schedule. When combining all the provided information, the parameter dr_{fst} can be created, which indicates how many drivers are needed at time slot $t \in T$ to transport the baggage of flight $f \in F$ to the aircraft if it is assigned to baggage section $s \in S$. The second driving task for a flight can start before the first driving task is finished, and therefore, multiple drivers at the same time slot could be needed.

The latest allowed opening time is always before the start of the first driving task of a flight within each baggage section. The objective of the model for this section is to minimize all the employees and the amount of shortened lateral opening times. In order to minimize the overall amount of employees, the drivers that will perform the driving tasks will also be minimized and will have a weight β.

14.2. MIP Formulation

This section contains the description of the new or changed sets, parameters and decision variables as well as the complete MIP for this extension.

Parameters

dr_{fst}	The number of drivers needed at time slot $t \in T$ to transport the baggage of flight $f \in F$ to the aircraft, in case it is assigned to section $s \in S$

Decision variables

UD_{st}	number of driver groups assigned to start their shift at time slot $t \in T$ in section $s \in S$
ZD_{st}	the number of driver groups working in section $s \in S$ at time slot $t \in T$

Mixed integer program

$$\min \quad \overbrace{\alpha \sum_{s\in S}\sum_{t\in T} U_{st}}^{\text{Obj. Part I}} + \overbrace{\beta \sum_{s\in S}\sum_{t\in T} UD_{st}}^{\text{Obj. Part II}} + \overbrace{\gamma \sum_{f\in F}\sum_{o\in O}\sum_{s\in S} d^2_{fos}\cdot XO_{fos}}^{\text{Obj. Part III}} \tag{14.1}$$

$$\text{s.t.} \quad \sum_{f\in F} dr_{fst}\cdot X_{fs} \le ZD_{st} \qquad \forall s\in S,\ \forall t\in T \tag{14.2}$$

$$\sum_{i=\max\{1,t-sd+1\}}^{t} UD_{si} = ZD_{st} \qquad \forall s\in S,\ \forall t\in T \tag{14.3}$$

Constraints (4.4), (4.8) and (4.10) - (4.16)

Constraints (12.2) - (12.12)

$$UD_{st} \in \mathbb{N}_{\ge 0} \qquad \forall s\in S,\ \forall t\in T_s \tag{14.4}$$

$$ZD_{st} \in \mathbb{N}_{\ge 0} \qquad \forall s\in S,\ \forall t\in T \tag{14.5}$$

The Objective function (14.1) consists of a new third part, namely Part II, which minimizes the number of starting groups that perform the driving tasks. Part I and III are the same as in Objective function (12.1) which was explained in Section 12.2. Weights α and β will have the same value, because the main goal is to minimize the total number of employees.

Constraints (14.2) ensure that there are enough drivers in each baggage section at every time slot to fulfill all the driving tasks for the flights assigned to that baggage section. Given the number of driver groups working at a certain time slot within a specific baggage section, it can be determined how many driver groups need to start their shift in that baggage section at a certain time slot, which is done by Constraints (14.3).

Constraints (14.4) and (14.5) indicate the domain of the decision variables, where the loading and driving groups may only start a shift if it is allowed.

14.3. Hierarchical Solution Method

The hierarchical solution method described in Chapter 12.3 is already an extended version of the method from Chapter 7 and needs to be adjusted in order to include the driving tasks. Only the first phase is extended by adding Constraints (14.2) and (14.3). The second phase and iterative method remain the same. Appendix B, denotes the entire first phase MIP for the hierarchical solution method for the MIP with driving tasks.

15

Computational Results Driving Tasks

In this chapter, the results for the methods described in Chapter 14 are evaluated based on the KPI's described in Chapter 10. The evaluation criteria *Number of Employees*, must be extended to also include the amount of drivers and not just the number of employees working in the baggage sections. Before evaluating the results, the extra data required for the driving tasks is described in Section 15.1. Subsequently, Section 15.2 elaborates upon the results for the hierarchical solution method.

15.1. Data Description

In order to create the correct parameter needed for the extension with driving tasks, some extra data is needed that will be described in this section. The tugs with baggage units must be driven to the aircraft multiple times, where each time represents one driving task. KLM uses a predefined number of driving tasks with corresponding starting times for each combination of baggage halls and aircraft type, while creating a schedule. In reality, the tugs could be driven earlier or later, depending on the actual amount of baggage units that is dropped. These starting times are given in Table 15.1. Note that this data has been simplified in agreement with KLM, so that it could be implemented more easily. Rules of combining driving tasks under certain conditions are left out, since they do not occur very often in practice and inclusion can never lead to a worse result than the outcome of the current program.

Baggage Hall	**E**	**E**	**E**	**D**	**D**	**D**	**South**	**South**	**South**
Aircraft Type	**WIBO**	**NABO**	**Com**	**WIBO**	**NABO**	**Com**	**WIBO**	**NABO**	**Com**
Task 1	-105	-60	-45	-105	-65	-65	-60	-30	-30
Task 2	-60	-35	-25	-60	-40	-20	-50	-20	-15
Task 3	-35	-20	0	-35	-20	0	-30	-15	0
Task 4	-15	0	0	-15	0	0	-15	0	0

Table 15.1: For every combination of baggage hall and aircraft type, the starting time of each driving task is denoted in the number of minutes before the scheduled time of departure

The driving times in minutes from each baggage hall to every pier at Schiphol Airport is given in Table 15.2. Schiphol Airport is build using a pier design and every pier is just a long narrow building with gates on both sides. So the driving time to pier A for example, denotes the driving times to all gates that start with the letter A and so on.

From Baggage	**To Pier**							
Hall	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
D	9	7.02	5	2.72	3.94	6.33	7.9	9
E	10	7.93	5.71	4.98	2.41	5.33	6.9	8
South	5	3.43	2.36	4.88	4.76	5.33	7.6	9

Table 15.2: Driving times from every baggage hall to every pier in minutes

On top of the driving time, some extra time for the unloading must be added for every driving task. For WIBO aircraft, an additional 7.5 minutes are needed and for NABO and Commuter aircraft an additional 9.4 minutes will be added. The reason that the bigger WIBO aircraft takes less time is because the baggage is stored in big containers that are directly shipped into the aircraft, whereas the baggage of NABO and Commuter aircraft is loaded into the aircraft individually.

15.2. Results on Hierarchical Solution Method

Based on the conclusions of Section 13.2, only the hierarchical solution method including the driving tasks is used for the Monday, Tuesday and Wednesday data set. Tables 15.3, 15.4 and 15.5 presents for the Monday, Tuesday and Wednesday data set, respectively. Within the time limit of one hour and a half, multiple feasible solution are obtained. It is again not proven that the solutions are optimal, as there exists a gap between the objective value and the LP bound. It is tested to run the Wednesday data set for more than 8 hours, however the objective value did not change. The LP bound increased slightly, from 183.50 to 138.53 after 6.5 hours.

			Monday		
Iteration	**Total Comp. Time**	**Feasible?**	**Obj. Value**	**LP Bound**	**#Constraints**
1	56	No	331.582	184.28	17,969
2	111	Yes	335.799	184.28	18,139
3	3,983	Yes	243.192	184.45	18,139
4	5,304	Yes	**243.192**	184.40	18,139

Table 15.3: Results for the Monday data set for the hierarchical method including driving tasks

			Tuesday		
Iteration	**Total Comp. Time**	**Feasible?**	**Obj. Value**	**LP Bound**	**#Constraints**
1	57	No	334.917	184.40	17,969
2	110	No	327.264	184.40	18,064
3	165	Yes	341.179	184.40	18,159
4	210	Yes	282.779	184.40	18,159
5	4,704	Yes	245.805	184.40	18,159
6	5,305	Yes	**243.794**	184.40	18,159

Table 15.4: Results for the Tuesday data set for the hierarchical method including driving tasks

			Wednesday		
Iteration	**Total Comp. Time**	**Feasible?**	**Obj. Value**	**LP Bound**	**#Constraints**
1	58	No	333.231	183.33	17,661
2	116	No	339.784	183.33	17,736
3	168	No	334.716	183.33	17,802
4	223	No	331.981	183.33	17,840
5	276	Yes	322.758	183.33	17,893
6	322	Yes	288.197	183.33	17,893
7	3,520	Yes	**246.461**	183.50	17,893

Table 15.5: Results for the Wednesday data set for the hierarchical method including driving tasks

Table 15.6 provides some insight on the objective value and the number of delayed time slots. More drivers are needed than employees working in the baggage halls. The outcomes can not be compared to the lateral planning of KLM, because different constraints are used within the planning, as was already mentioned in Section 13.3. However, the average number of full time employees working on the outbound baggage handling process is 334. This suggests that the amount of workload can be decreased significantly using the hierarchical solution method to construct the lateral planning automatically.

	Monday	Tuesday	Wednesday
Number of employees in the baggage halls	111	113	116
Number of drivers	130	129	128
Total number of employees	241	242	244
Maximum number of delayed time slots of all flights	12	14	13
Number of flights which opening times are delayed	62	49	50
Total number of delayed time slots	292	239	285

Table 15.6: Information about the delayed time slots and the number of employees for the different flight schedules

Table 15.7 compares the number of variables, the number of constraints, the objective value and the LP bound for the two methods used in this part of the research. The number of constraints and variables in the second phase of the hierarchical solution method, differ per section and depends on the outcomes of the first phase. These spefic numbers per section are given in Appendix F for the last iteration and the maximum, minimum and average amount of these constraints and variables are given in Table 15.7. The second phase solves within one second for each of the sections.

	Monday	Tuesday	Wednesday
# Total Constraints Hierarchical Method	18,236	18,159	17,893
# Constraints First Phase	18,139	18,159	17,893
# Constraints in Total for Second Phase	26,960	27,009	25,826
Max # Constraints for Second Phase	8,821	8,807	8,547
Min # Constraints for Second Phase	406	269	410
Average # Constraints for Second Phase	2,247	2,251	2,152
# Total Variables Hierarchical Method	70,052	69,748	70,388
# Variables First Phase	70,052	69,748	70,388
# Variables in Total for Second Phase	13,380	13,482	12,956
Max # Variables for Second Phase	8,817	9,007	8,513
Min # Variables for Second Phase	115	31	103
Average # Variables for Second Phase	1,115	1,124	1,080

Table 15.7: Model sizes for the first and second phase of the hierarchical method compared

V

Conclusions and Discussion

16

Conclusions

In this research, several methods to model the outbound baggage handling process were examined. Specifically, this thesis focused on the outbound baggage handling process at Schiphol International Airport, performed by KLM. The main question researched in this study is as follows:

"How can reality be accurately modelled, to minimize the number of employees working on the outbound baggage handling process?"

In this chapter, the different methods employed in this thesis are presented along with a summary of their results based on the key performance indicators. All results are based upon data on the outbound baggage handling process at KLM in 2018. Furthermore, some conclusions are drawn based on the results as obtained for KLM.

This thesis encompasses three extensions on the basic model created by ORTEC, which assigned flights to baggage halls. First, the model is extended to a more detailed version, since schedules obtained with the basic model did not lead to the expected improvements and did not model reality very well. The extended version assigns flights to specific laterals and baggage sections, while meeting all the demand and flight allocation constraints. Second, in order to obtain a feasible schedule, the model is extended, such that delaying lateral opening times for flights is made possible under certain circumstances. Last, driving tasks and the shifts of drivers are added in order to minimize all the employees working on the outbound baggage process. The constraints of the models are based on the daily operation. All methods are tested on a variety of flight schedules and the results of the methods described in this thesis are evaluated and summarized.

Several methods are created for the first extension, on lateral level, of this thesis: an MIP, a hierarchical solution method and column generation approach. For the mixed integer programming formulation, several speed-ups and valid inequalities are suggested. These methods are tested on the data provided by KLM along with three smaller flight schedules, because the full flight schedule result in infeasibility for this extension. Three main conclusions are drawn for this extension: 1) results showed that the MIP formulation and the hierarchical solution method are the two most efficient methods, and therefore, these two methods are chosen for the other extensions; 2) The LP-relaxations resulted to be weak for all the methods; 3) The convergence speed of the LP-bound compared to the convergence speed of the objective value, suggest that the solution process can be interrupted at an earlier stage without ending up with a higher objective value. However, optimality can not be guaranteed when interrupting at an earlier stage.

For the second extension, lateral opening times to be delayed, in order to obtain feasible solutions when using the real flight schedules. The MIP and hierarchical solution method are tested in this extension and the hierarchical solution method outperformed MIP. The hierarchical solution method did not proof optimality within the used time limit, and therefore, it can not be concluded whether the optimal solution was obtained.

In the last extension, in which driving tasks are added, only the hierarchical solution methods is used. Feasible solutions are obtained within the time limit. The lateral plannings constructed in this research are meeting all the constraints and created automatically instead of manually, within a reasonable time frame. Unfortunately, the outcomes can not be compared to the actual lateral planning of KLM, as this planning violates multiple constraints set in this research. However, looking at the average number of full-time working employees currently working at KLM, the results obtained in this thesis suggest that a significant decrease in the workload can be achieved using the methods discussed in this research.

17

Discussion & Recommendations

In this chapter, the limitations and possible extensions of this study are discussed. The limitations due to time restrictions or data provided by KLM are elaborated in Section 17.1. Possible extensions are proposed in Section 17.2.

17.1. Limitations of the Research

In this research, some assumptions have been made to simplify the process. Without those assumptions, the conclusions from this research could be different. Therefore, it might be interesting to investigate the effect of relaxing the following assumptions, sorted on descending order based on the importance:

- The driving tasks are simplified in this research. In practice, certain driving tasks can be combined under certain circumstances. This might results in a a lower objective value, because the number of drivers could decrease. Further research could implement these specific rules for the driving tasks.
- The effect of adding two additional valid inequalities as a speed-up to MIP are examined, however it is not likely that these are the only possible cuts. As these valid inequalities did not appear to be beneficial, further research could investigate the existence of additional cuts. One way to achieve better cuts, is to examine the CPLEX log file in more detail to obtain the cuts CPLEX is adding itself.
- Due to time restrictions, the branching strategies in combination with column generation are not implemented. However, this might be interesting to implement for models that run on the complete data sets, as these models contain a huge number of variables.
- Different settings are only evaluated on the results obtained while using Objective Function 5.1 as it is assumed that in the near future only full time shifts are planned by the program. However, in the future it might be preferable to extend the program for using multiple different shift durations. In this case, Objective Function (4.1) must be used in order to minimize the workload. Using Objective (5.1) and minimizing the number of starting groups does not result in the same solution, when different shift durations are used.
- For the simplicity of this research, it is assumed that baggage can be stored in the buffer when the lateral opening times are delayed. In practice, however, the number of baggage units that can be stored in the buffer are limited. Furthermore, it is assumed that these baggage units that were stored in the buffer, can immediately be dropped when the lateral is opened. However, this is not realistic as the baggage units must transfer from the buffer to the laterals first.
- Groups of employees are assigned to baggage sections, based on the estimated workload per section. Theoretically, the groups of employees assigned to work in a baggage section can handle the workload of all the flights assigned to that same baggage section. Meaning that a group could handle the flights spread over all the different laterals within that baggage section. However, in practice, this is not very realistic as a group can not work at multiple laterals at the same time. IT is assumed that it will work out in practice, because the same planning process is used as the one KLM is using right know, which is described in Section 1.3.

- Further research could investigate why the different Objective Functions(4.1) and (5.1) result in different computation times, as discussed in Subsection 11.1.3.
- As the main goal is to minimize the number of working groups, the models of Part III and IV are only tested with the focus on this part of the objective functions. In the future, several weight configurations could be tested more extensively.
- In this research, every group consists of just one employee. In further research, the effect of different group sizes could be explored. It is expected that choosing bigger groups sizes, results in less flexibility during the solve, and therefore, it is expected that more employees are needed to start a shift.
- For the simplicity of this research, it is assumed that working shifts may start every quarter. However, different starting intervals can also be used or even fixed starting times could be chosen in reality.

Furthermore, there are some limitations due to the data provided by KLM. These are as follows:

- The data provided by KLM is not very accurate, e.g. the drop off profile and the driving times are not updated for at least five years. However, the outcomes of the models presented in this research are highly influenced by the parameters created from this data. It is important that this input is updated and it is recommended to create a more detailed drop off profile, for example, per flight number and day of the week.

17.2. Extensions of the Research

The following extensions to the research discussed in this report are suggested sorted on descending order based on the importance:

- As mentioned before, the created plannings are not very robust. The planning could be made more robust by, for example, adding more buffer time in between two flights on a lateral.
- The shift duration used by KLM and in this research, is equal to a full time shift plus a 30 minute break. However, this break is not included when creating the lateral planning. A good extension of this research would be to directly plan a break of half an hour for each of the working groups. This break could be placed within a certain defined time frame, somewhere in the middle of the shift.
- Infeasible options that are found in the hierarchical method that was described in Chapter 7, are stored into a parameter. Currently, the set of flights that was assigned to a section for which the second phase was infeasible, was stored. However, the combinations of the three main characteristics of the set of flights can also be stored instead, i.e., the starting and closing time of the laterals of the flights within the section and the required amount of laterals per flight. If these infeasible combination of characteristics along with the corresponding sections are stored every time the hierarchical method is used, the parameter becomes more advanced. In the long term, this could save some computation time, because many infeasible assignments of combinations of flight characteristics to certain sections can be avoided in advance.
- Employees are assigned to sections in this research and are only allowed to load baggage from the laterals into the tugs that are located within this baggage section. In practice, it could be possible that an employee helps loading for a short amount of time, in a baggage section next to the section it is assigned to, in case it is really necessary. A small amount of slack could be added which allows an employee of a certain baggage section to help a short period of time at a lateral close to its own baggage section.
- Currently, the baggage tugs are driven from the baggage hall to the aircraft where the baggage is loaded into the aircraft and the tugs are driven back empty to the hall. It might be interesting to combine the inbound and outbound baggage transportation, especially for flights that have a short turn around time. This means that the aircraft is standing at the gate for a short amount of time until it departs again. The drivers that bring the outbound baggage to the aircraft could then also bring the inbound baggage to the unloading docks, where the baggage must be entered into the BHS. An even more advanced extension would be to allow baggage tugs to return baggage from other aircraft that are located nearby. This extension is possible in theory, but it is not considered by KLM in the near future.

- After the lateral planning is created, the actual personnel roster is made by another planning department. However, these personnel rosters must also meet the collective employment agreement regulations and based on these regulations, some shifts can be more expensive than others. For example, a night shift is more expensive than a day shift. In order to minimize the amount of working groups working the expensive shifts, a cost could be assigned to certain starting times for shifts.

Bibliography

[1] Aarts, J. (2018). Collifactor-L5Y-Summary. Baggage Logistics KLM.

[2] Arkin, E.M. & e.B. Silverberg (1985). Scheduling Jobs with Fixed Start and End Times. Discrete Applied Mathematics, 18, 1-8

[3] Abdelghany, A., K. Abdelghany & R. Narasimhan (2006). Scheduling Baggage-handling facilities in congested airports. Journal of Air Transport Management, 12, 76-81.

[4] Air transport, passengers carried (2017). *The World Bank*. Retrieved from `https://data.worldbank.org/indicator/IS.AIR.PSGR`.

[5] Ascó, A., J.A.D. Afkin & E.K. Burke (2014). An Analysis of constructive algorithms for the airport baggage sorting station assignment problem. Springer Science + Business Media New York.

[6] Barnhart, C., E.L. Johnson, L. Nemhauser, M.W.P. Savelsbergh & P.H. Vance (2014). Branch-and-Price: Column Generation for Solving Huge Integer Programs. Operations Research 46 (3), 316-329.

[7] Barth, T. & D. Pisinger (2011). Scheduling of outbound luggage handling at airports. Operations Research Proceedings, Springer.

[8] Bisschop, J. (2019). AIMMS Language Reference - Variable and Constraints Declaration. AIMMS B.V.

[9] Cornuéjols, G. (2008). Valid Inequalities for Mixed Integer Linear Programs. Math Program, 112, 3–44

[10] Diepen, G., J.A. Hoogeveen, J.M. van den Akker & J.W. Smeltink (2007). Using column generation for gate planning at Amsterdam Airport Schiphol. Department of Information and Computing Sciences, Utrecht University.

[11] Descrosiers, J. & M.E. Lübbecke (2005). A Primer in Column Generation, in Column Gneration, G. Desaulniers, J. Desrosiers, and M.M. Solomon (Eds.) Springer US, 2005, 1-23.

[12] Dondeti, V.R. & H. Emmons (1991). Fixed Job Scheduling with two Types of Processors. Operations Research 40 (1-supplement-1), 76-85.

[13] Frey, M. (2014). Models and Methods for Optimizing Baggage Handling at Airports. Doctoral dissertation.

[14] Frey, M., C. Artigues, R. Kolisch & P. Lopez (2010). Scheduling and Planning the Outbound Baggage Process at International Airports. IEEE International Conference on Industrial Engineering and Engineering Management, 101-105.

[15] Frey, M., R. Kolisch & C. Artigues (2017). Column Generation for Outbound Baggage Handling at Airports. Transportation Science, INFORMS.

[16] Gilmore, P.C. & R.E. Gomory (1961). A linear programming approach to the cutting stock problem. Operations Research 11, 849–859.

[17] Huang, E., I. Liu & J.T. Lin (2018). Robust Model for the Assignment of Outgoing Flights on Airport Baggage Unloading Areas. Transportation Research Part E 115, 110-125.

[18] Huang, E., P. Mital, M. Goetschalckx & K. Wu (2016). Optimal assignment of airport baggage unloading zones to outgoing flights. Transportation Research Part E 94, 110-122.

[19] IBM Knowledge Center - Selecting Nodes (n.d.). Retrieved from `https://www.ibm.com/support/knowledgecenter/SSSA5P_12.6.3/ilog.odms.cplex.help/CPLEX/UsrMan/topics/discr_optim/mip/performance/20_node_seln.html`

[20] Madani, K., L. Read & L. Shalikarian (2014). Voting Under Uncertainty: A Stochastic Framework for Analyzing Group Decision Making Problems. Water Resource Manage, 28, 1839 - 1856.

[21] Orlin, J. (2013). Lecture 13: 15.0583/8 Cutting planes: Better bounds through better IP formulations [Lecture Nodes]. Retrieved from MIT, Massachusetts, https://ocw.mit.edu/courses/sloan-school-of-management/15-053-optimization-methods-in-management-science-spring-2013/lecture-notes/MIT15_053S13_lec13.pdf.

[22] Ryan, D.M. & B.A. Foster (1981). An Integer Programming Approach to Scheduling. Computer Scheduling of Public Transport Urban Passenger Vehicle and Crew Scheduling, A. Wren (ed.), 269-280.

[23] Weer werkonderbrekingen bagagepersoneel Schiphol (2018, May 3). *NOS*. Retrieved from http://nos.nl.

A

List of Abbreviations

ASP	Activity Selection Problem
BFCs	Baggage Flow Controllers
BHS	Baggage Handling System
CG	Column Generation
GRASP	Greedy Randomized Adaptive Search Procedure
ICA Flight	Intercontinental Aviation
MIP	Mixed Integer Program
MP	Master Problem
NABO	Narrow Body Aircraft
RMP	Restricted Master Problem
RLMP	Restricted Linear Master Problem
SBCs	Symmetry Breaking Constraints
WIBO	Wide Body Aircraft

B

Overview Models

Sets

F	set of flights
F_2	set of flights that require one lateral, $F_2 \subseteq F$
F_2	set of flights that require two laterals, $F_2 \subseteq F$
F_4	set of flights that require four laterals, $F_4 \subseteq F$
F_b	set of flights for which baggage could drop in at least one of the baggage sections, because $hst_{fs} = 1$, it holds that $F_b \subseteq F$
I	infeasible options
L	set of laterals
L_s	set of laterals that are positioned in section $s \in S$, where $L = \cap_{s \in S} L_s$
L_1	set of laterals which have capacity one, a single lateral, $L_1 \subseteq L$
L_2	set of street laterals which have capacity two $L_2 \subseteq L$ and $L_1 \cap L_2 = L$
O	set of possible opening time slots for laterals, $O \subseteq T$
P	section plans
H	set of halls
S	set of baggage sections
T	set of time slots of δ minutes each
T_h	set of time slots at which groups of employees may start working their shift in hall $h \in H$, $T_h \subseteq T$
T_s	set of time slots at which groups of employees may start working their shift in section $s \in S$, $T_s \subseteq T$

Parameters

α	weight for the objective function for MIP with delaying lateral opening times in Section 12.2
a_{fh}	$= \begin{cases} 1 & \text{if flight } f \in F \text{ is fixed to a specific hall } h \in H \\ 0 & \text{otherwise} \end{cases}$
a_{fl}	$= \begin{cases} 1 & \text{if the baggage of flight } f \in F \text{ is allowed to go on lateral } l \in L \\ 0 & \text{otherwise} \end{cases}$
ao_{fos}	$= \begin{cases} 1 & \text{if the lateral(s) of flight } f \in F \text{ is (are) allowed to open at time slot } o \in O \text{ in section } s \in S \\ 0 & \text{otherwise} \end{cases}$
aos_{ifos}	$\begin{cases} 1 & \text{if flight } f \in F \text{ is supposed to open at time slot } o \in O \text{ in section } s \in S \text{ in infeasible option } i \in I \\ 0 & \text{otherwise} \end{cases}$
as_{ifs}	$\begin{cases} 1 & \text{if flight } f \in F \text{ is assigned to section } s \in S \text{ in infeasible option } i \in I \\ 0 & \text{otherwise} \end{cases}$
b_{fht}	estimated number of bags from flight $f \in F$ in hall $h \in H$ during time slot $t \in T$
b_{flt}	estimated number of bags from flight $f \in F$ on lateral $l \in L$ during time slot $t \in T$

b_{fst}	estimated number of bags from flight $f \in F$ in section $s \in S$ during time slot $t \in T$
bo_{fost}	estimated number of bags from flight $f \in F$ in section $s \in S$ during time slot $t \in T$, in case the lateral(s) of the flight is (are) opened at opening time $o \in O$
bt_{fs}	the estimated number of bags for flight $f \in F$ in section $s \in S$
c	time limit stopping criteria for the entire iterative method
c_1	time limit stopping criteria for solving the model of the first phase
ca_l	capacity of lateral $l \in L$, so the amount of (sub) laterals within that lateral
c_{ps}	costs of assigning section plan $p \in P$ to section $s \in S$
d_{fos}	difference in time slots between the original lateral opening time slot and the opening time slot $o \in O$ for flight $f \in F$ in section $s \in S$
dr_{fst}	The amount of drivers needed at time slot $t \in T$ to transport the baggage of flight $f \in F$ to the aircraft, in case it is assigned to section $s \in S$
ϵ	constant value used for the second phase objective function, Section 7.2
e_{ps}	$= \begin{cases} 1 & \text{if section plan } p \in P \text{ is allowed to be assigned to section } s \in S \\ 0 & \text{otherwise} \end{cases}$
γ	weight for the objective function for MIP with delaying lateral opening times in Section 12.2
h_{flt}	$= \begin{cases} 1 & \text{if flight } f \in F \text{ is on lateral } l \in L \text{ at time slot } t \in T \text{ in case it is assigned to lateral } l \in L \\ 0 & \text{otherwise} \end{cases}$
ho_{folt}	$= \begin{cases} 1 & \text{if flight } f \in F \text{ is on lateral } l \in L \text{ at time slot } t \in T \text{ in case it is assigned to lateral } l \in L \\ & \text{and in case it is opened at time slot } o \in O \\ 0 & \text{otherwise} \end{cases}$
hos_{fst}	$\begin{cases} 1 & \text{if lateral(s) for flight } f \in F \text{ would be open in section } s \in S \text{ at time slot } t \in T \text{ in case it is} \\ & \text{assigned to section } s \in S \text{ and opened at time slot } o \in O \\ 0 & \text{otherwise} \end{cases}$
hst_{fs}	$\begin{cases} 1 & \text{if lateral(s) for flight } f \in F \text{ would be open in section } s \in S \text{ in case it is assigned to section } s \in S \\ 0 & \text{otherwise} \end{cases}$
hs_{fst}	$\begin{cases} 1 & \text{if lateral(s) for flight } f \in F \text{ would be open in section } s \in S \text{ at time slot } t \in T \text{ in case it is} \\ & \text{assigned to section } s \in S \\ 0 & \text{otherwise} \end{cases}$
lc_{ht}	lateral capacity in hall $h \in H$ at time slot $t \in T$
lc_{st}	lateral capacity in section $s \in S$ at time slot $t \in T$, assuming that all laterals are available at every time slot, determined by $lc_{st} = \sum_{l \in L_s} ca_l$
ma_l	maximum amount of flights that can be on lateral $l \in L$ at the same time slot
ml_f	minimal laterals that are required for flight $f \in F$, $ml_f = \left\lceil \frac{r_f}{\max_{l \in L} ca_l} \right\rceil$
mw_t	the minimal number of working groups required at time slot $t \in T$
mz_{ft}	minimum number of working groups required for flight $f \in F$ at time slot $t \in T$
$n_{l_a l_b}$	$= \begin{cases} 1 & \text{if lateral } l_a \in L \text{ is located next to lateral } l_b \in L \text{, where } l_a < l_b \\ 0 & \text{otherwise} \end{cases}$
nf	$\begin{cases} 1 & \text{if at least one second phase MIP was infeasible for the solution of the first phase} \\ 0 & \text{otherwise} \end{cases}$
$\overline{ob}$	the currently best found objective function value
q	$\begin{cases} 1 & \text{if a feasible solution is already obtained} \\ 0 & \text{otherwise} \end{cases}$
r_f	number of required laterals for flight $f \in F$
rq_{fht}	number of required laterals at time slot $t \in T$ for flight $f \in F$ in case it is assigned to hall $h \in H$
rq_{fst}	number of required laterals at time slot $t \in T$ for flight $f \in F$ in case it is assigned to section $s \in S$,
rqt_{fs}	number of required laterals for flight $f \in F$ in case it is assigned to section $s \in S$ determined by $rq_{fst} = r_f \cdot h_{flt}$ for $l \in L_s$
sc_h	shift capacity at hall $h \in H$, the amount of baggage that can be handled per time slot for one group
sc_s	shift capacity at section $s \in S$, i.e. the amount of baggage that can be handled per time slot for one working group
sd	shift duration in amount of time slots
tl_s	total amount of laterals in section $s \in S$, $tl_s = \sum_{l \in L} ps_{ls}$

$$w_{ht} = \begin{cases} 1 & \text{if a group may start working at hall } h \in H \text{ at time slot } t \in T \\ 0 & \text{otherwise} \end{cases}$$

$$w_{st} = \begin{cases} 1 & \text{if a group may start working at section } s \in S \text{ at time slot } t \in T \\ 0 & \text{otherwise} \end{cases}$$

$$x_{fp} = \begin{cases} 1 & \text{if flight } f \in F \text{ is in section plan } p \in P \\ 0 & \text{otherwise} \end{cases}$$

$$x_{fs} = \begin{cases} 1 & \text{if flight } f \in F \text{ is assigned to section } s \in S \\ 0 & \text{otherwise} \end{cases}$$

Decision variables

$$\lambda_{ps} = \begin{cases} 1 & \text{if section plan } p \in P \text{ is assigned to section } s \in S \\ 0 & \text{otherwise} \end{cases}$$

MWS_s minimum amount of working groups needed in section $s \in S$, such that all the flights can be assigned to a section while fulfilling Constraints (6.6) and (6.5)

U_{ht} number of groups assigned to start their shift at time slot t in hall h

U_{st} number of loading groups assigned to start their shift at time slot $t \in T$ in section $s \in S$

$$U_{gst} \quad \begin{cases} 1 & \text{if group } g \text{ starts their shift at time slot } t \text{ in section } s \\ 0 & \text{otherwise} \end{cases}$$

U_t number of groups assigned to start their shift at time slot $t \in T$

UD_{st} number of driver groups assigned to start their shift at time slot $t \in T$ in section $s \in S$

$$V_{fl_al_b} = \begin{cases} 1 & \text{if flight } f \in F \text{ is assigned to both lateral } l_a \in L \text{ and } l_b \in L \\ 0 & \text{otherwise} \end{cases}$$

$$X_{fh} = \begin{cases} 1 & \text{if flight } f \text{ is allocated to hall } h \\ 0 & \text{otherwise} \end{cases}$$

$$X_{fs} = \begin{cases} 1 & \text{if flight } f \in F \text{ is assigned to section } s \in S \\ 0 & \text{otherwise} \end{cases}$$

$$X_f = \begin{cases} 1 & \text{if flight } f \in F \text{ is assigned} \\ 0 & \text{otherwise} \end{cases}$$

$$XO_{fos} = \begin{cases} 1 & \text{if the lateral(s) of flight } f \in F \text{ open at time slot } o \in O \text{ in section } s \in S \\ 0 & \text{otherwise} \end{cases}$$

$$Y_{fl} = \begin{cases} 1 & \text{if flight } f \in F \text{ is assigned to lateral } l \in L \\ 0 & \text{otherwise} \end{cases}$$

$$YO_{fol} = \begin{cases} 1 & \text{if flight } f \in F \text{ is assigned to lateral } l \in L \text{ and the lateral opens at time slot } o \in O \\ 0 & \text{otherwise} \end{cases}$$

Z_{ht} the amount of groups working in hall h at time slot t

U_{st} number of groups assigned to start their shift at time slot $t \in T$ in section $s \in S$

Z_{st} the amount of groups working in section $s \in S$ at time slot $t \in T$

$$Z_{gst} \quad \begin{cases} 1 & \text{if group } g \text{ works in section } s \text{ at time slot } t \\ 0 & \text{otherwise} \end{cases}$$

Z_t number of groups working at time slot $t \in T$

ZD_{st} the amount of driver groups working in section $s \in S$ at time slot $t \in T$

Original Problem, Section 3.2

$$
\begin{aligned}
\min \quad & \sum_{h\in H}\sum_{t\in T} Z_{ht} & & \\
\text{s.t.} \quad & \sum_{h\in H} X_{fh} = 1 & & \forall f\in F \\
& \sum_{i=\max\{1,t-sd+1\}}^{t} U_{hi} = Z_{ht} & & \forall h\in H,\ \forall t\in T_h \\
& \sum_{f\in F} b_{fht}\cdot X_{fh} \le sc_h\cdot Z_{ht} & & \forall h\in H,\ \forall t\in T \\
& \sum_{f\in F} rq_{fht}\cdot X_{fh} \le lc_{ht} & & \forall h\in H,\ \forall t\in T \\
& a_{fh} \le X_{fh} & & \forall f\in F,\ \forall h\in H \\
& X_{fh} \in\{0,1\} & & \forall f\in F,\ \forall h\in H \\
& U_{ht} \in\mathbb{N}_{\ge 0} & & \forall h\in H,\ \forall t\in T_h \\
& Z_{ht} \in\mathbb{N}_{\ge 0} & & \forall h\in H,\ \forall t\in T
\end{aligned}
$$

MIP on Lateral Level, Section 4.1

$$
\begin{aligned}
\min \quad & \sum_{s\in S}\sum_{t\in T} Z_{st} & & \\
\text{s.t.} \quad & \sum_{s\in S} X_{fs} = 1 & & \forall f\in F \\
& \sum_{f\in F} b_{fst}\cdot X_{fs} \le sc_s\cdot Z_{st} & & \forall s\in S,\ \forall t\in T \\
& \sum_{i=\max\{1,t-sd+1\}}^{t} U_{si} = Z_{st} & & \forall s\in S,\ \forall t\in T_s \\
& \sum_{f\in F} rq_{fst}\cdot X_{fs} \le lc_{st} & & \forall s\in S,\ \forall t\in T \\
& \sum_{l\in L_1} Y_{fl_1} + \sum_{l\in L_2} Y_{fl_2} = 1 & & \forall f\in F_1 \\
& \sum_{l\in L_s} ca_l\cdot Y_{fl} = r_f\cdot X_{fs} & & \forall f\in F_2\cup F_4,\ \forall s\in S \\
& Y_{fl} \le X_{fs} & & \forall f\in F,\ \forall s\in S,\ \forall l\in L_s \\
& \sum_{f\in F} h_{flt}\cdot Y_{fl} \le ma_l & & \forall l\in L,\ \forall t\in T \\
& Y_{fl} \le a_{fl} & & \forall f\in F,\ \forall l\in L \\
& V_{fl_al_b} \le Y_{fl_a} & & \forall f\in F_2,\ \forall l_a,\ l_b\in L_1,\ \text{with } n_{l_al_b}=1 \\
& V_{fl_al_b} \le Y_{fl_b} & & \forall f\in F_2,\ \forall l_a,\ l_b\in L_1,\ \text{with } n_{l_al_b}=1 \\
& \sum_{l_2\in L_2} Y_{fl_2} + \sum_{l_a,\ l_b\in L_1 | n_{l_al_b}=1} V_{fl_al_b} = 1 & & \forall f\in F_2 \\
& V_{fl_al_b} \le Y_{fl_a} & & \forall f\in F_4,\ \forall l_a,\ l_b\in L_2,\ \text{with } n_{l_al_b}=1 \\
& V_{fl_al_b} \le Y_{fl_b} & & \forall f\in F_4,\ \forall l_a,\ l_b\in L_2,\ \text{with } n_{l_al_b}=1 \\
& \sum_{l_a,\ l_b\in L_2 | n_{l_al_b}=1} V_{fl_al_b} = 1 & & \forall f\in F_4 \\
& X_{fs} \in\{0,1\} & & \forall f\in F,\ \forall s\in S \\
& Y_{fl} \in\{0,1\} & & \forall f\in F,\ \forall l\in L \\
& V_{fl_al_b} \in\{0,1\} & & \forall f\in F,\ \forall\ l_a,\ l_b\in L,\ \text{with } n_{l_al_b}=1 \\
& U_{st} \in\mathbb{N}_{\ge 0} & & \forall s\in S,\ \forall t\in T_s \\
& Z_{st} \in\mathbb{N}_{\ge 0} & & \forall s\in S,\ \forall t\in T
\end{aligned}
$$

Binary Programming and Symmetry Breaking, Section 5.2

$$
\begin{array}{lll}
\min & \sum_{g\in G}\sum_{s\in S}\sum_{t\in T} U_{gst} & \\
\sum_{f\in F} b_{fst}\cdot X_{fs} & \le sc_s\cdot \sum_{g\in G} Z_{gst} & \forall s\in S,\ \forall t\in T\\
\sum_{s\in S}\sum_{t\in T} U_{gst} & \le 1 & \forall g\in G\\
\sum_{s\in S}\sum_{t\in T} U_{gst} & \ge \sum_{s\in S}\sum_{t\in T} U_{g+1st} & \forall g\in\{1,\dots,G-1\}\\
\sum_{t\in T} U_{g+1st}-\sum_{t\in T} U_{gst}-\sum_{t\in T} U_{g+2st}+1 & \ge 0 & \forall g\in\{1,\dots,G-2\},\ \forall s\in S\\
\sum_{s\in S} X_{fs} & =1 & \forall f\in F\\
\sum_{i=\max\{1,t-sd+1\}}^{t} U_{si} & = Z_{st} & \forall s\in S,\ \forall t\in T_s\\
\sum_{f\in F} rq_{fst}\cdot X_{fs} & \le lc_{st} & \forall s\in S,\ \forall t\in T\\
\sum_{l\in L_1} Y_{fl_1}+\sum_{l\in L_2} Y_{fl_2} & =1 & \forall f\in F_1\\
\sum_{l\in L_s} ca_l\cdot Y_{fl} & = r_f\cdot X_{fs} & \forall f\in F_2\cup F_4,\ \forall s\in S\\
Y_{fl} & \le X_{fs} & \forall f\in F,\ \forall s\in S,\ \forall l\in L_s\\
\sum_{f\in F} h_{flt}\cdot Y_{fl} & \le ma_l & \forall l\in L,\ \forall t\in T\\
Y_{fl} & \le a_{fl} & \forall f\in F,\ \forall l\in L\\
V_{fl_al_b} & \le Y_{fl_a} & \forall f\in F_2,\ \forall l_a,\ l_b\in L_1,\ \text{with } n_{l_al_b}=1\\
V_{fl_al_b} & \le Y_{fl_b} & \forall f\in F_2,\ \forall l_a,\ l_b\in L_1,\ \text{with } n_{l_al_b}=1\\
\sum_{l_2\in L_2} Y_{fl_2}+\sum_{l_a,\ l_b\in L_1\mid n_{l_al_b}=1} V_{fl_al_b} & =1 & \forall f\in F_2\\
V_{fl_al_b} & \le Y_{fl_a} & \forall f\in F_4,\ \forall l_a,\ l_b\in L_2,\ \text{with } n_{l_al_b}=1\\
V_{fl_al_b} & \le Y_{fl_b} & \forall f\in F_4,\ \forall l_a,\ l_b\in L_2,\ \text{with } n_{l_al_b}=1\\
\sum_{l_a,\ l_b\in L_2\mid n_{l_al_b}=1} V_{fl_al_b} & =1 & \forall f\in F_4\\
X_{fs} & \in\{0,1\} & \forall f\in F,\ \forall s\in S\\
Y_{fl} & \in\{0,1\} & \forall f\in F,\ \forall l\in L\\
V_{fl_al_b} & \in\{0,1\} & \forall f\in F,\ \forall\ l_a,\ l_b\in L,\ \text{with } n_{l_al_b}=1\\
U_{gst} & \in\mathbb{N}_{\ge 0} & \forall g\in G,\ \forall s\in S,\ \forall t\in T_s\\
Z_{gst} & \in\mathbb{N}_{\ge 0} & \forall g\in G,\ \forall s\in S,\ \forall t\in T
\end{array}
$$

Removing Decision Variable X_{fs}, Section 5.3

$$
\begin{array}{llll}
\min & \sum_{s\in S}\sum_{t\in T} U_{st} & & \\
 & \sum_{f\in F}\sum_{l\in L} b_{flt}\cdot Y_{fl} & \le sc_s\cdot Z_{st} & \forall s\in S,\ \forall t\in T \\
 & \sum_{s\in S} X_{fs} & = 1 & \forall f\in F \\
 & \sum_{l\in L} ca_l\cdot Y_{fl} & = r_f & \forall f\in F_2\cup F_4 \\
 & \sum_{i=\max\{1,t-sd+1\}}^{t} U_{si} & = Z_{st} & \forall s\in S,\ \forall t\in T_s \\
 & \sum_{l\in L_1} Y_{fl_1} + \sum_{l\in L_2} Y_{fl_2} & = 1 & \forall f\in F_1 \\
 & Y_{fl} & \le X_{fs} & \forall f\in F,\ \forall s\in S,\ \forall l\in L_s \\
 & \sum_{f\in F} h_{flt}\cdot Y_{fl} & \le ma_l & \forall l\in L,\ \forall t\in T \\
 & Y_{fl} & \le a_{fl} & \forall f\in F,\ \forall l\in L \\
 & V_{fl_al_b} & \le Y_{fl_a} & \forall f\in F_2,\ \forall l_a,\ l_b\in L_1,\ \text{with } n_{l_al_b}=1 \\
 & V_{fl_al_b} & \le Y_{fl_b} & \forall f\in F_2,\ \forall l_a,\ l_b\in L_1,\ \text{with } n_{l_al_b}=1 \\
 & \sum_{l_2\in L_2} Y_{fl_2} + \sum_{l_a,\ l_b\in L_1|n_{l_al_b}=1} V_{fl_al_b} & = 1 & \forall f\in F_2 \\
 & V_{fl_al_b} & \le Y_{fl_a} & \forall f\in F_4,\ \forall l_a,\ l_b\in L_2,\ \text{with } n_{l_al_b}=1 \\
 & V_{fl_al_b} & \le Y_{fl_b} & \forall f\in F_4,\ \forall l_a,\ l_b\in L_2,\ \text{with } n_{l_al_b}=1 \\
 & \sum_{l_a,\ l_b\in L_2|n_{l_al_b}=1} V_{fl_al_b} & = 1 & \forall f\in F_4 \\
 & Y_{fl} & \in \{0,1\} & \forall f\in F,\ \forall l\in L \\
 & V_{fl_al_b} & \in \{0,1\} & \forall f\in F,\ \forall\ l_a,\ l_b\in L,\ \text{with } n_{l_al_b}=1 \\
 & U_{st} & \in \mathbb{N}_{\ge 0} & \forall s\in S,\ \forall t\in T_s \\
 & Z_{st} & \in \mathbb{N}_{\ge 0} & \forall s\in S,\ \forall t\in T
\end{array}
$$

Determining mw_t, Section 6.2.2

$$
\begin{array}{llll}
\min & \sum_{s\in S} MWS_s & & \\
\text{s.t.} & \sum_{s\in S} X_{fs} & = 1 & \forall f\in F_b \\
 & \sum_{f\in F_b} bt_{fs}\cdot X_{fs} & \le sc_s\cdot MWS_s & \forall s\in S,\ \forall t\in T \\
 & \sum_{f\in F_b} hst_{fs}\cdot ml_f\cdot X_{fs} & \le tl_s & \forall s\in S,\ \forall t\in T \\
 & \sum_{f\in F_b} rqt_{fs}\cdot X_{fs} & \le lc_{st} & \forall s\in S,\ \forall t\in T \\
 & X_{fs} & \in \{0,1\} & \forall f\in F,\ \forall s\in S \\
 & MWS_s & \in \mathbb{N}_{\ge 0} & \forall s\in S
\end{array}
$$

Algorithm, Subsection 6.2.2

Algorithm 3 Procedure for obtaining the parameter mw_t

procedure
    for $t \in T$ **do**
        Empty the set F_b
        for $f \in F$ **do**
            if $\sum_{s \in S} hst_{fs} = 1$ **then**
                Add flight f to the set F_b
            end if
        end for
        for $f \in F_b$ **do**
            $bt_{fs} \leftarrow b_{fst}$
            $hst_{fs} \leftarrow hs_{fst}$
            $rqt_{fs} \leftarrow rq_{fst}$
        end for
        Solve the model "Determining mw_t"
        $mw_t \leftarrow \sum_{s \in S} MW_s$
    end for
end procedure

First phase, Section 7.1

$$
\begin{array}{llll}
\min & \displaystyle\sum_{s \in S} \sum_{t \in T} U_{st} & & \\
\text{s.t.} & \displaystyle\sum_{f \in F} hs_{fst} \cdot ml_f \cdot X_{fs} & \leq tl_s & \forall s \in S,\ \forall t \in T \\
& \displaystyle\sum_{s \in S} X_{fs} & = 1 & \forall f \in F \\
& \displaystyle\sum_{f \in F} b_{fst} \cdot X_{fs} & \leq sc_s \cdot Z_{st} & \forall s \in S,\ \forall t \in T \\
& \displaystyle\sum_{i=\max\{1, t-sd+1\}}^{t} U_{si} & = Z_{st} & \forall s \in S,\ \forall t \in T_s \\
& \displaystyle\sum_{f \in F} rq_{fst} \cdot X_{fs} & \leq lc_{st} & \forall s \in S,\ \forall t \in T \\
& X_{fs} & \in \{0, 1\} & \forall f \in F,\ \forall s \in S \\
& U_{st} & \in \mathbb{N}_{\geq 0} & \forall s \in S,\ \forall t \in T_s \\
& Z_{st} & \in \mathbb{N}_{\geq 0} & \forall s \in S,\ \forall t \in T
\end{array}
$$

Second phase, Section 7.2

$$
\begin{array}{llll}
\min & \epsilon & & \\
\text{s.t.} & \sum_{l\in L_1\cap L_s} Y_{fl_1} + \sum_{l\in L_2\cap L_s} Y_{fl_2} & = 1 & \forall f\in F_1 \\
& \sum_{l\in L_s} ca_l \cdot Y_{fl} & = r_f \cdot x_{fs} & \forall f\in F_2\cup F_4,\ \forall s\in S \\
& Y_{fl} & \le x_{fs} & \forall f\in F,\ \forall s\in S,\ \forall l\in L_s \\
& \sum_{f\in F} h_{flt}\cdot Y_{fl} & \le ma_l & \forall l\in L_s,\ \forall t\in T \\
& Y_{fl} & \le a_{fl} & \forall f\in F,\ \forall l\in L_s \\
& V_{fl_al_b} & \le Y_{fl_a} & \forall f\in F_2,\ \forall l_a,\ l_b\in L_1\cap L_s,\ \text{with } n_{l_al_b}=1 \\
& V_{fl_al_b} & \le Y_{fl_b} & \forall f\in F_2,\ \forall l_a,\ l_b\in L_1\cap L_s,\ \text{with } n_{l_al_b}=1 \\
& \sum_{l_2\in L_2\cap L_s} Y_{fl_2} + \sum_{l_a,\ l_b\in L_1\cap L_s | n_{l_al_b}=1} V_{fl_al_b} & = 1 & \forall f\in F_2 \\
& V_{fl_al_b} & \le Y_{fl_a} & \forall f\in F_4,\ \forall l_a,\ l_b\in L_2\cap L_s,\ \text{with } n_{l_al_b}=1 \\
& V_{fl_al_b} & \le Y_{fl_b} & \forall f\in F_4,\ \forall l_a,\ l_b\in L_2\cap L_s,\ \text{with } n_{l_al_b}=1 \\
& \sum_{l_a,\ l_b\in L_2\cap L_s | n_{l_al_b}=1} V_{fl_al_b} & = 1 & \forall f\in F_4 \\
& Y_{fl} & \in\{0,1\} & \forall f\in F,\ \forall l\in L_s \\
& V_{fl_al_b} & \in\{0,1\} & \forall f\in F,\ \forall\ l_a,\ l_b\in L_s,\ \text{with } n_{l_al_b}=1
\end{array}
$$

Adjusted First Phase, Subsection 7.4.1

$$
\begin{array}{llll}
\min & \sum_{s\in S}\sum_{t\in T} U_{st} & & \\
\text{s.t.} & \sum_{f\in F} hs_{fst}\cdot ml_f \cdot X_{fs} & \le tl_s & \forall s\in S,\ \forall t\in T \\
& \sum_{s\in S} X_{fs} & = 1 & \forall f\in F \\
& \sum_{f\in F} b_{fst}\cdot X_{fs} & \le sc_s\cdot Z_{st} & \forall s\in S,\ \forall t\in T \\
& \sum_{i=\max\{1,t-sd+1\}}^{t} U_{si} & = Z_{st} & \forall s\in S,\ \forall t\in T_s \\
& \sum_{f\in F} rq_{fst}\cdot X_{fs} & \le lc_{st} & \forall s\in S,\ \forall t\in T \\
& \sum_{f\in F} as_{ifs}\cdot X_{fs} & \le \sum_{f\in F} as_{ifs} - 1 & \forall i\in I,\ \forall s\in S,\ \text{with } \sum_{f\in F} as_{ifs} > 0 \\
& X_{fs} & \in\{0,1\} & \forall f\in F,\ \forall s\in S \\
& U_{st} & \in\mathbb{N}_{\ge 0} & \forall s\in S,\ \forall t\in T_s \\
& Z_{st} & \in\mathbb{N}_{\ge 0} & \forall s\in S,\ \forall t\in T
\end{array}
$$

Iterative method, Subsection 7.4.2

Algorithm 4 Outline of Iterative Method

procedure
  Empty as_{ifs}
  No feasible options yet, $q \leftarrow 0$
  $i \leftarrow 0$
  $\overline{ob} \leftarrow N$ where $N \in \mathbb{N}_{\geq 0}$ is big enough
  while time limit c is not reached **do** ▷ Step 4
    if $q = 1$ **then** ▷ Step 3
      the best obtained feasible solution so far as start solution
    end if
    Solve first phase and interrupt when a better objective function value is obtained than $\overline{ob}$ or when the time limit c_1 is reached ▷ Step 1
    if no better solution is obtained, i.e., $\sum_{s \in S} \sum_{t \in T} Z_{st} \geq \overline{ob}$ **then** ▷ Step 4
      Break
    end if
    $nf \leftarrow 0$
    for all sections $s^* \in S$ **do**
      Solve second phase ▷ Step 2
      if program status is infeasible **then**
        indicate that a feasible solution is found, $nf \leftarrow 1$
        $i \leftarrow i + 1$
        Store the infeasible solution, $as_{ifs^*} \leftarrow X_{fs^*}$
        for all sections $s' \in S$ that are identical to s^* **do** ▷ Step 2b
          $i \leftarrow i + 1$
          $as_{ifs'} \leftarrow X_{fs^*}$
        end for
        for all flights $f^* \in F$ that are assigned to the section $s^* \in S$, i.e., $X_{f^*s^*} = 1$ **do**
          for all flights $f' \in F$ with the exact same characteristics as flight f^* and which is not already assigned to this section, i.e., $as_{if's^*} = 0$ **do**
            The solution of replacing f' and f^* within this section is also infeasible
            New infeasible solution will be stored, $i \leftarrow i + 1$
            Store temporary $X^*_{fs^*} \leftarrow X_{fs^*}$, such that the flights f' and f^* can be swapped
            The value of the original flight becomes zero $X^*_{f^*s^*} \leftarrow 0$
            The identical flight in combination with the other flights is also infeasible $X^*_{f's^*} \leftarrow 1$
            $as_{ifs^*} \leftarrow X^*_{f's^*}$
            for all sections $s' \in S$ that are identical to s^* **do**
              $i \leftarrow i + 1$
              $as_{ifs'} \leftarrow X^*_{fs^*}$
            end for
          end for
        end for
      end if
    end for
    if no infeasible status was given, $nf = 0$ and $\sum_{s \in S} \sum_{t \in T} Z_{st} < \overline{ob}$ **then** ▷ Step 2a
      Feasible solution is obtained, so $q \leftarrow 1$
      Store all the variables of the best obtained feasible solution so far
    end if
  end while
end procedure

Master Problem for Column Generation, Subsection 8.2.1

$$
\begin{aligned}
\min \quad & \sum_{p\in P}\sum_{s\in S} c_{ps}\cdot\lambda_{ps} && && \\
\text{s.t.} \quad & \sum_{p\in P}\sum_{s\in S} x_{fp}\cdot\lambda_{ps} && = 1 && \forall f \in F\\
& \sum_{p\in P} e_{ps}\cdot\lambda_{ps} && \leq 1 && \forall s \in S\\
& \lambda_{ps} && \in \{0,1\} && \forall p\in P,\ \forall s\in S,\ \text{with } e_{ps}=1
\end{aligned}
$$

Restricted Linear Master Problem for Column Generation, Subsection 8.2.1

$$
\begin{aligned}
\min \quad & \sum_{p\in P}\sum_{s\in S} c_{ps}\cdot\lambda_{ps} && && \\
\text{s.t.} \quad & \sum_{p\in P}\sum_{s\in S} x_{fp}\cdot\lambda_{ps} && = 1 && \forall f \in F\\
& \sum_{p\in P} e_{ps}\cdot\lambda_{ps} && \leq 1 && \forall s \in S\\
& \lambda_{ps} && \in \{0,1\} && \forall p\in P,\ \forall s\in S,\ \text{with } e_{ps}=1
\end{aligned}
$$

Pricing Problem for Column Generation, Subsection 8.2.2

$$
\begin{aligned}
\min \quad & \sum_{t\in T} U_t - \sum_{f\in F}\pi_f\cdot X_f - \mu_s && && \\
& \sum_{f\in F} b_{fst}\cdot X_f && \leq sc_s\cdot Z_t && \forall t\in T\\
& \sum_{i=\max\{1,t-sd+1\}}^{t} U_i && = Z_t && \forall t\in T_s\\
& \sum_{l\in L_1\cap L_s} Y_{fl_1} + \sum_{l\in L_2\cap L_s} Y_{fl_2} && = X_f && \forall f\in F_1\\
& \sum_{l\in L_s} ca_l\cdot Y_{fl} && = r_f\cdot X_f && \forall f\in F_2\cup F_4\\
& Y_{fl} && \leq X_f && \forall f\in F,\ \forall l\in L_s\\
& \sum_{f\in F} h_{flt}\cdot Y_{fl} && \leq ma_l && \forall l\in L_s,\ \forall t\in T\\
& Y_{fl} && \leq a_{fl} && \forall f\in F,\ \forall l\in L_s\\
& V_{fl_al_b} && \leq Y_{fl_a} && \forall f\in F_2,\ \forall l_a,\ l_b\in L_1\cap L_s,\ \text{with } n_{l_al_b}=1\\
& V_{fl_al_b} && \leq Y_{fl_b} && \forall f\in F_2,\ \forall l_a,\ l_b\in L_1\cap L_s,\ \text{with } n_{l_al_b}=1\\
& \sum_{l_2\in L_2} Y_{fl_2} + \sum_{l_a,\ l_b\in L_1\cap L_s | n_{l_al_b}=1} V_{fl_al_b} && = X_f && \forall f\in F_2\\
& V_{fl_al_b} && \leq Y_{fl_a} && \forall f\in F_4,\ \forall l_a,\ l_b\in L_2\cap L_s,\ \text{with } n_{l_al_b}=1\\
& V_{fl_al_b} && \leq Y_{fl_b} && \forall f\in F_4,\ \forall l_a,\ l_b\in L_2\cap L_s,\ \text{with } n_{l_al_b}=1\\
& \sum_{l_a,\ l_b\in L_2\cap L_s | n_{l_al_b}=1} V_{fl_al_b} && = X_f && \forall f\in F_4\\
& X_f && \in\{0,1\} && \forall f\in F\\
& Y_{fl} && \in\{0,1\} && \forall f\in F,\ \forall l\in L_s\\
& V_{fl_al_b} && \in\{0,1\} && \forall f\in F,\ \forall\ l_a,\ l_b\in L_s,\ \text{with } n_{l_al_b}=1\\
& U_t && \in\mathbb{N}_{\geq 0} && \forall t\in T_s\\
& Z_t && \in\mathbb{N}_{\geq 0} && \forall t\in T
\end{aligned}
$$

MIP for delaying lateral opening times, Section 12.2

$$\min \quad \underbrace{\alpha \sum_{s\in S} \sum_{t\in T} U_{st}}_{\text{Obj. Part I}} + \underbrace{\gamma \sum_{f\in F} \sum_{o\in O} \sum_{s\in S} d^2_{fos} \cdot XO_{fos}}_{\text{Obj. Part II}}$$

$$
\begin{array}{llll}
\text{s.t.} & \sum_{s\in S} \sum_{o\in O} XO_{fos} & = 1 & \forall f \in F \\
& XO_{fos} & \le ao_{fos} & \forall f \in F,\ \forall o \in O,\ \forall s \in S \\
& YO_{fol} & \le XO_{fos} & \forall f \in F,\ \forall o \in O,\ \forall s \in S,\ \forall l \in L_s \\
& \sum_{f\in F} \sum_{o\in O} bo_{fost} \cdot XO_{fos} & \le sc_s \cdot Z_{st} & \forall s \in S,\ \forall t \in T \\
& \sum_{f\in F} \sum_{o\in O} ho_{folt} \cdot YO_{fol} & \le ma_l & \forall l \in L,\ \forall t \in T \\
& \sum_{o\in O} XO_{f,o,s} & = X_{f,s} & \forall f \in F,\ \forall s \in S \\
& \sum_{o\in O} YO_{fol} & = Y_{fl} & \forall f \in F,\ \forall l \in L \\
& \sum_{i=\max\{1,t-sd+1\}}^{t} U_{si} & = Z_{st} & \forall s \in S,\ \forall t \in T_s \\
& \sum_{l\in L_1} Y_{fl_1} + \sum_{l\in L_2} Y_{fl_2} & = 1 & \forall f \in F_1 \\
& \sum_{l\in L_s} ca_l \cdot Y_{fl} & = r_f \cdot X_{fs} & \forall f \in F_2 \cup F_4,\ \forall s \in S \\
& Y_{fl} & \le a_{fl} & \forall f \in F,\ \forall l \in L \\
& V_{fl_al_b} & \le Y_{fl_a} & \forall f \in F_2,\ \forall l_a,\ l_b \in L_1,\ \text{with } n_{l_al_b} = 1 \\
& V_{fl_al_b} & \le Y_{fl_b} & \forall f \in F_2,\ \forall l_a,\ l_b \in L_1,\ \text{with } n_{l_al_b} = 1 \\
& \sum_{l_2\in L_2} Y_{fl_2} + \sum_{l_a,\ l_b\in L_1 | n_{l_al_b}=1} V_{fl_al_b} & = 1 & \forall f \in F_2 \\
& V_{fl_al_b} & \le Y_{fl_a} & \forall f \in F_4,\ \forall l_a,\ l_b \in L_2,\ \text{with } n_{l_al_b} = 1 \\
& V_{fl_al_b} & \le Y_{fl_b} & \forall f \in F_4,\ \forall l_a,\ l_b \in L_2,\ \text{with } n_{l_al_b} = 1 \\
& \sum_{l_a,\ l_b\in L_2 | n_{l_al_b}=1} V_{fl_al_b} & = 1 & \forall f \in F_4 \\
& X_{fs},\ Y_{fl},\ V_{fl_al_b} & \in \{0,1\} & \forall f \in F,\ \forall l,\ l_a,\ l_b \in L,\ \forall s \in S \\
& XO_{fos},\ YO_{fol} & \in \{0,1\} & \forall f \in F,\ \forall o \in O,\ \forall l \in L,\ \forall s \in S \\
& U_{st} & \in \mathbb{N}_{\ge 0} & \forall s \in S,\ \forall t \in T_s \\
& Z_{st} & \in \mathbb{N}_{\ge 0} & \forall s \in S,\ \forall t \in T
\end{array}
$$

First Phase for Delaying Lateral Opening Times, Section 12.3

$$
\begin{aligned}
\min \quad & \alpha \sum_{s\in S}\sum_{t\in T} U_{st} + \gamma \sum_{f\in F}\sum_{o\in O}\sum_{s\in S} d_{fos}^2 \cdot XO_{fos} && \\
\text{s.t.} \quad & \sum_{f\in F}\sum_{o\in O} hos_{fost} \cdot ml_f \cdot XO_{fos} \le tl_s && \forall s \in S,\ \forall t \in T \\
& \sum_{f\in F}\sum_{o\in O} r_f \cdot hos_{fost} \cdot XO_{fos} \le lc_{st} && \forall s \in S,\ \forall t \in T \\
& \sum_{f\in F}\sum_{o\in O} aos_{ifos} \cdot XO_{fos} \le \sum_{f\in F}\sum_{o\in O} aos_{ifos} - 1 && \forall i \in I,\ \forall s \in S, \text{ with } \sum_{f\in F} as_{ifs} > 0 \\
& \sum_{i=\max\{1,t-sd+1\}}^{t} U_{si} = Z_{st} && \forall s \in S,\ \forall t \in T_s \\
& \sum_{s\in S}\sum_{o\in O} XO_{fos} = 1 && \forall f \in F \\
& XO_{fos} \le ao_{fos} && \forall f \in F,\ \forall o \in O,\ \forall s \in S \\
& \sum_{f\in F}\sum_{o\in O} bo_{fost} \cdot XO_{fos} \le sc_s \cdot Z_{st} && \forall s \in S,\ \forall t \in T \\
& XO_{fos} \in \{0,1\} && \forall f \in F,\ \forall o \in O,\ \forall s \in S \\
& U_{st} \in \mathbb{N}_{\ge 0} && \forall s \in S,\ \forall t \in T_s \\
& Z_{st} \in \mathbb{N}_{\ge 0} && \forall s \in S,\ \forall t \in T
\end{aligned}
$$

Second Phase for Delaying Lateral Opening Times

$$
\begin{aligned}
\min \quad & \epsilon && \\
\text{s.t.} \quad & \sum_{f\in F} ho_{folt} \cdot YO_{fl} \le ma_l + P_{lt} && \forall l \in L_s,\ \forall t \in T \\
& \sum_{l\in L_1\cap L_s} Y_{fl_1} + \sum_{l\in L_2\cap L_s} Y_{fl_2} = 1 && \forall f \in F_1 \\
& \sum_{l\in L_s} ca_l \cdot Y_{fl} = r_f \cdot X_{fs} && \forall f \in F_2 \cup F_4,\ \forall s \in S \\
& Y_{fl} \le a_{fl} && \forall f \in F,\ \forall l \in L_s \\
& V_{fl_al_b} \le Y_{fl_a} && \forall f \in F_2,\ \forall l_a,\ l_b \in L_1 \cap L_s, \text{ with } n_{l_al_b} = 1 \\
& V_{fl_al_b} \le Y_{fl_b} && \forall f \in F_2,\ \forall l_a,\ l_b \in L_1 \cap L_s, \text{ with } n_{l_al_b} = 1 \\
& \sum_{l_2\cap L_s\in L_2} Y_{fl_2} + \sum_{l_a,\ l_b\in L_1\cap L_s | n_{l_al_b}=1} V_{fl_al_b} = 1 && \forall f \in F_2 \\
& V_{fl_al_b} \le Y_{fl_a} && \forall f \in F_4,\ \forall l_a,\ l_b \in L_2 \cap L_s, \text{ with } n_{l_al_b} = 1 \\
& V_{fl_al_b} \le Y_{fl_b} && \forall f \in F_4,\ \forall l_a,\ l_b \in L_2 \cap L_s, \text{ with } n_{l_al_b} = 1 \\
& \sum_{l_a,\ l_b\in L_2 | n_{l_al_b}=1} V_{fl_al_b} = 1 && \forall f \in F_4 \\
& YO_{fol} \le XO_{fos} && \forall f \in F,\ \forall o \in O,\ \forall s \in S,\ \forall l \in L_s \\
& \sum_{f\in F}\sum_{o\in O} ho_{folt} \cdot YO_{fol} \le ma_l && \forall l \in L_s,\ \forall t \in T \\
& \sum_{o\in O} YO_{fol} = Y_{fl} && \forall f \in F,\ \forall l \in L_s \\
& Y_l,\ YO_{fol} \in \{0,1\} && \forall f \in F,\ \forall o \in O,\ \forall l \in L_s
\end{aligned}
$$

MIP for driving tasks, Section 14.1

$$
\begin{array}{llll}
\min & \overbrace{\alpha \sum_{s\in S}\sum_{t\in T} U_{st}}^{\text{Obj. Part I}} + \overbrace{\beta \sum_{s\in S}\sum_{t\in T} UD_{st}}^{\text{Obj. Part II}} + \overbrace{\gamma \sum_{f\in F}\sum_{o\in O}\sum_{s\in S} d_{fos}^2 \cdot XO_{fos}}^{\text{Obj. Part III}} & & \\
\text{s.t.} & \sum_{f\in F} dr_{fst}\cdot X_{fs} & \le ZD_{st} & \forall s\in S,\ \forall t\in T \\
 & \sum_{i=\max\{1,t-sd+1\}}^{t} UD_{si} & = ZD_{st} & \forall s\in S,\ \forall t\in T \\
 & Y_{fl} & \le X_{fs} & \forall f\in F,\ \forall s\in S,\ \forall l\in L_s \\
 & Y_{fl} & \le a_{fl} & \forall f\in F,\ \forall l\in L \\
 & V_{fl_al_b} & \le Y_{fl_a} & \forall f\in F_2,\ \forall l_a,\ l_b\in L_1,\ \text{with } n_{l_al_b}=1 \\
 & V_{fl_al_b} & \le Y_{fl_b} & \forall f\in F_2,\ \forall l_a,\ l_b\in L_1,\ \text{with } n_{l_al_b}=1 \\
 & \sum_{l_2\in L_2} Y_{fl_2} + \sum_{l_a,\ l_b\in L_1|n_{l_al_b}=1} V_{fl_al_b} & = 1 & \forall f\in F_2 \\
 & V_{fl_al_b} & \le Y_{fl_a} & \forall f\in F_4,\ \forall l_a,\ l_b\in L_2,\ \text{with } n_{l_al_b}=1 \\
 & V_{fl_al_b} & \le Y_{fl_b} & \forall f\in F_4,\ \forall l_a,\ l_b\in L_2,\ \text{with } n_{l_al_b}=1 \\
 & \sum_{l_a,\ l_b\in L_2|n_{l_al_b}=1} V_{fl_al_b} & = 1 & \forall f\in F_4 \\
 & \sum_{i=\max\{1,t-sd+1\}}^{t} U_{si} & = Z_{st} & \forall s\in S,\ \forall t\in T_s \\
 & \sum_{s\in S}\sum_{o\in O} XO_{fos} & = 1 & \forall f\in F \\
 & XO_{fos} & \le ao_{fos} & \forall f\in F,\ \forall o\in O,\ \forall s\in S \\
 & YO_{fol} & \le XO_{fos} & \forall f\in F,\ \forall o\in O,\ \forall s\in S,\ \forall l\in L_s \\
 & \sum_{f\in F}\sum_{o\in O} bo_{fost}\cdot XO_{fos} & \le sc_s\cdot Z_{st} & \forall s\in S,\ \forall t\in T \\
 & \sum_{f\in F}\sum_{o\in O} ho_{folt}\cdot YO_{fol} & \le ma_l & \forall l\in L,\ \forall t\in T \\
 & \sum_{o\in O} XO_{f,o,s} & = X_{f,s} & \forall f\in F,\ \forall s\in S \\
 & \sum_{o\in O} YO_{fol} & = Y_{fl} & \forall f\in F,\ \forall l\in L \\
 & X_{fs},\ Y_{fl},\ V_{fl_al_b} & \in\{0,1\} & \forall f\in F,\ \forall l,\ l_a,\ l_b\in L,\ \forall s\in S \\
 & XO_{fos},\ YO_{fol} & \in\{0,1\} & \forall f\in F,\ \forall o\in O,\ \forall l\in L,\ \forall s\in S \\
 & U_{st},\ UD_{st} & \in\mathbb{N}_{\ge 0} & \forall s\in S,\ \forall t\in T_s \\
 & Z_{st},\ ZD_{st} & \in\mathbb{N}_{\ge 0} & \forall s\in S,\ \forall t\in T
\end{array}
$$

First Phase for Adding Driving Tasks, Section 14.3

$$
\begin{array}{llll}
\min & \alpha \sum_{s\in S}\sum_{t\in T} U_{st} + \gamma \sum_{f\in F}\sum_{o\in O}\sum_{s\in S} d^2_{fos}\cdot XO_{fos} & & \\
\text{s.t.} & \sum_{f\in F} dr_{fst}\cdot X_{fs} & \le ZD_{st} & \forall s\in S,\ \forall t\in T \\
& \sum_{i=\max\{1,t-sd+1\}}^{t} UD_{si} & = ZD_{st} & \forall s\in S,\ \forall t\in T_s \\
& \sum_{f\in F}\sum_{o\in O} hos_{fost}\cdot ml_f\cdot XO_{fos} & \le tl_s & \forall s\in S,\ \forall t\in T \\
& \sum_{f\in F}\sum_{o\in O} r_f\cdot hos_{fost}\cdot XO_{fos} & \le lc_{st} & \forall s\in S,\ \forall t\in T \\
& \sum_{f\in F}\sum_{o\in O} aso_{ifos}\cdot XO_{fos} & \le \sum_{f\in F}\sum_{o\in O} aso_{ifos} - 1 & \forall i\in I,\ \forall s\in S,\ \text{with} \sum_{f\in F} as_{ifs} > 0 \\
& \sum_{s\in S}\sum_{o\in O} XO_{fos} & = 1 & \forall f\in F \\
& XO_{fos} & \le ao_{fos} & \forall f\in F,\ \forall o\in O,\ \forall s\in S \\
& YO_{fol} & \le XO_{fos} & \forall f\in F,\ \forall o\in O,\ \forall s\in S,\ \forall l\in L_s \\
& \sum_{f\in F}\sum_{o\in O} bo_{fost}\cdot XO_{fos} & \le sc_s\cdot Z_{st} & \forall s\in S,\ \forall t\in T \\
& XO_{fos} & \in \{0,1\} & \forall f\in F,\ \forall o\in O,\ \forall s\in S \\
& U_{st} & \in \mathbb{N}_{\ge 0} & \forall s\in S,\ \forall t\in T_s \\
& Z_{st} & \in \mathbb{N}_{\ge 0} & \forall s\in S,\ \forall t\in T
\end{array}
$$

Second Phase for Adding Driving Tasks, Section 14.3

$$
\begin{array}{llll}
\min & \epsilon & & \\
\text{s.t.} & \sum_{f\in F} ho_{folt}\cdot YO_{fl} & \le ma_l + P_{lt} & \forall l\in L_s,\ \forall t\in T \\
& \sum_{l\in L_1\cap L_s} Y_{fl_1} + \sum_{l\in L_2\cap L_s} Y_{fl_2} & = 1 & \forall f\in F_1 \\
& \sum_{l\in L_s} ca_l\cdot Y_{fl} & = r_f\cdot X_{fs} & \forall f\in F_2\cup F_4,\ \forall s\in S \\
& Y_{fl} & \le a_{fl} & \forall f\in F,\ \forall l\in L_s \\
& V_{fl_al_b} & \le Y_{fl_a} & \forall f\in F_2,\ \forall l_a,\ l_b\in L_1\cap L_s,\ \text{with } n_{l_al_b}=1 \\
& V_{fl_al_b} & \le Y_{fl_b} & \forall f\in F_2,\ \forall l_a,\ l_b\in L_1\cap L_s,\ \text{with } n_{l_al_b}=1 \\
& \sum_{l_2\cap L_s\in L_2} Y_{fl_2} + \sum_{l_a,\ l_b\in L_1\cap L_s | n_{l_al_b}=1} V_{fl_al_b} & = 1 & \forall f\in F_2 \\
& V_{fl_al_b} & \le Y_{fl_a} & \forall f\in F_4,\ \forall l_a,\ l_b\in L_2\cap L_s,\ \text{with } n_{l_al_b}=1 \\
& V_{fl_al_b} & \le Y_{fl_b} & \forall f\in F_4,\ \forall l_a,\ l_b\in L_2\cap L_s,\ \text{with } n_{l_al_b}=1 \\
& \sum_{l_a,\ l_b\in L_2 | n_{l_al_b}=1} V_{fl_al_b} & = 1 & \forall f\in F_4 \\
& YO_{fol} & \le XO_{fos} & \forall f\in F,\ \forall o\in O,\ \forall s\in S,\ \forall l\in L_s \\
& \sum_{f\in F}\sum_{o\in O} ho_{folt}\cdot YO_{fol} & \le ma_l & \forall l\in L_s,\ \forall t\in T \\
& \sum_{o\in O} YO_{fol} & = Y_{fl} & \forall f\in F,\ \forall l\in L_s \\
& Y_l,\ YO_{fol} & \in \{0,1\} & \forall f\in F,\ \forall o\in O,\ \forall l\in L_s
\end{array}
$$

C

Data

In this Appendix, the map that is discussed in Section 9 is shown in more detail for each hall. Hall D is illustrated in Figure C.1 in which can be seen that there are red, purple and blue single laterals. There is also one purple carousel, but this one is not used.

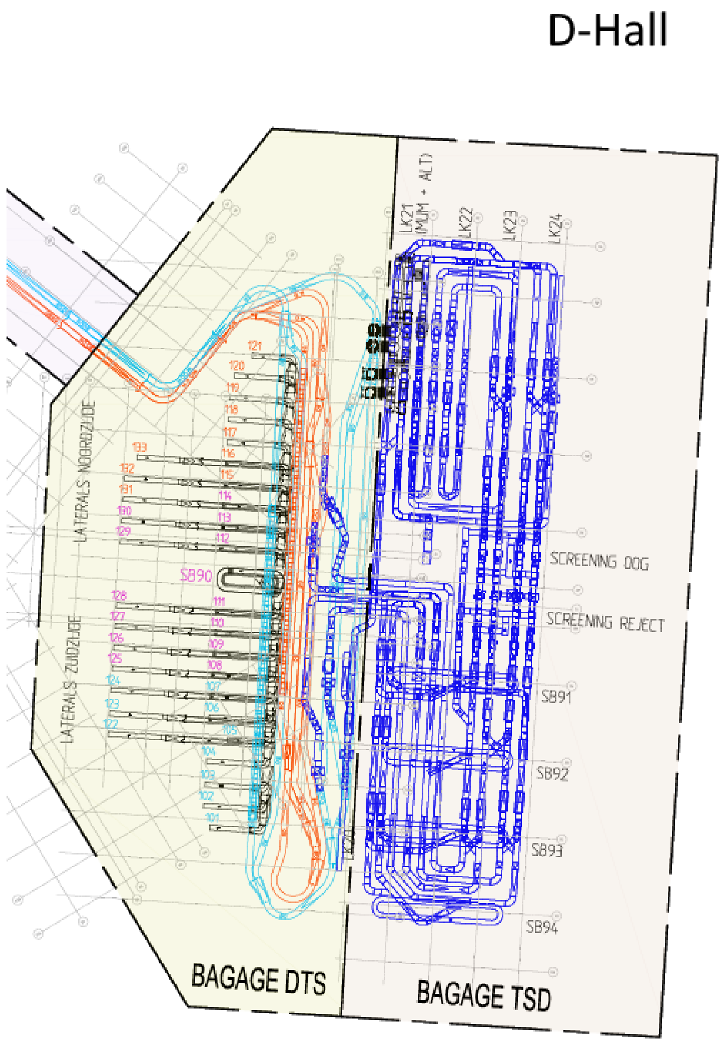


Figure C.1: Map of hall D

Figure C.2 shows the map of hall E. In this hall there are more double laterals, which are called 'streets'. These are used for WIBO aircraft as explained in the report. There is also one robot which is illustrated in pink.

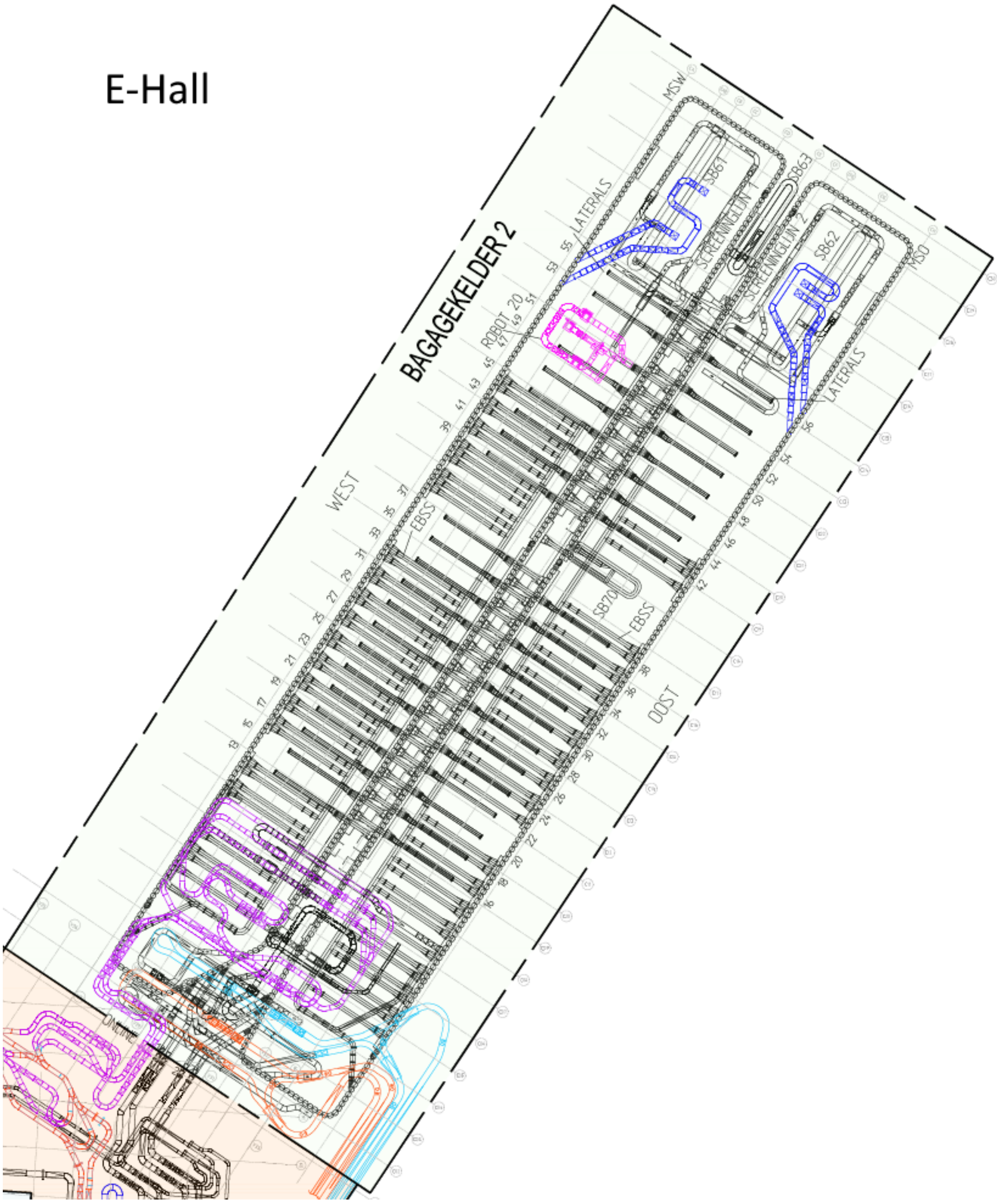


Figure C.2: Map of hall E

The South hall is illustrated in Figure C.3 and the pink laterals are the six robots in this hall. The laterals are hard to see, because the map exists of several layers, but below the pink robots, there are several black single laterals.

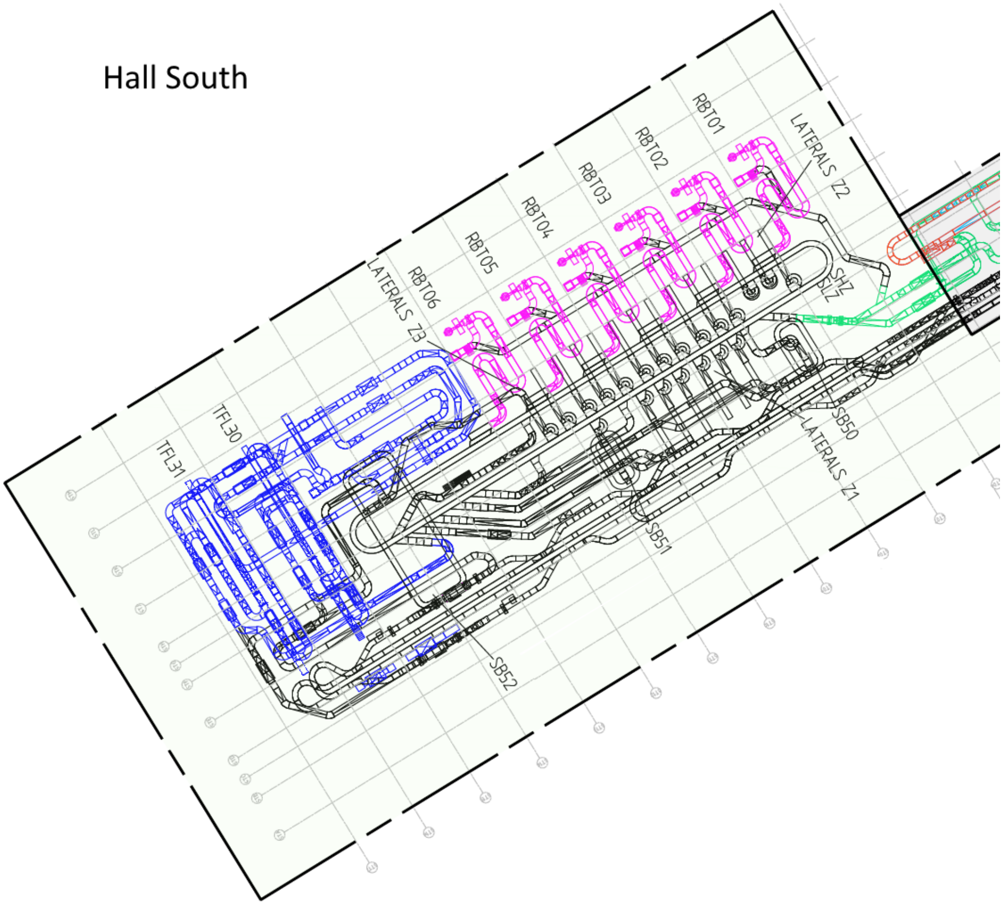


Figure C.3: Map of hall South

A street, which is actually just two laterals on top of each other, is illustrated in Figure C.4.

Figure C.4: Street lateral

Figure C.5 illustrates the robot that is used to load baggage into the cars which will be driven to the aircraft.

Figure C.5: Robot packing the cars

Figure C.6 illustrates the carousels.

Figure C.6: Carousels

D

Complete Results Part II

This chapter contains the results of all the methods of Part II. A maximum running time of half an hour is used for the problems in this part of the research. It is indicated with a '*', if the program was interrupted because this stopping criteria was reached. Ratio scores that are indicated by '**' are not completely fair, because the program was interrupted by the time limit in at least one of the data sets for this combination of settings.

D.1. Result for all combinations of constraint and priority settings

Data Set 1 - Prio 1	$\min \sum_{s\in S} \sum_{t\in T} U_{s,t}$			$\min \sum_{s\in S} \sum_{t\in T} Z_{s,t}$		
Constraint Setting	**Comp. Time**	**Time Until Opt. Obj. is Reached**	**Perc. Of Time Proving Optimal**	**Comp. Time**	**Time Until Opt. Obj. is Reached**	**Perc. Of Time Proving Optimal**
1	48	43	10%	1,800*	703	61%
2	66	56	15%	910	910	0%
3	42	14	67%	1,800*	119	93%
4	205	204	0%	1,253	29	98%
5	51	49	4%	1,102	112	90%
6	52	48	8%	1,455	90	94%

Table D.1: Computation time and time until the optimal objective value is reached in seconds for both Objective Functions (4.1) and (5.1), along with the percentage of time that the program was proving that the best objective value found was indeed optimal. Results for the different constraints settings, data set 1 and priority setting 1

Data Set 1 - Prio 2	$\min \sum_{s\in S} \sum_{t\in T} U_{s,t}$			$\min \sum_{s\in S} \sum_{t\in T} Z_{s,t}$		
Constraint Setting	**Comp. Time**	**Time Until Opt. Obj. is Reached**	**Perc. Of Time Proving Optimal**	**Comp. Time**	**Time Until Opt. Obj. is Reached**	**Perc. Of Time Proving Optimal**
1	47	45	4%	1,800*	837	54%
2	50	47	6%	1,352	182	87%
3	45	40	11%	741	58	92%
4	40	20	50%	1,800*	29	98%
5	38	33	13%	854	101	88%
6	76	70	8%	1,542	64	96%

Table D.2: Computation time and time until the optimal objective value is reached in seconds for both Objective Functions (4.1) and (5.1), along with the percentage of time that the program was proving that the best objective value found was indeed optimal. Results for the different constraints settings, data set 1 and priority setting 2

Data Set 1 - Prio 3	$\min \sum_{s\in S} \sum_{t\in T} U_{s,t}$			$\min \sum_{s\in S} \sum_{t\in T} Z_{s,t}$		
Constraint Setting	**Comp. Time**	**Time Until Opt. Obj. is Reached**	**Perc. Of Time Proving Optimal**	**Comp. Time**	**Time Until Opt. Obj. is Reached**	**Perc. Of Time Proving Optimal**
1	49	47	4%	1,800	353	80%
2	75	69	8%	1,800	23	99%
3	80	61	24%	1,800	359	80%
4	45	24	47%	1,800	179	90%
5	79	79	0%	1,449	24	98%
6	68	33	51%	1,800	197	89%

Table D.3: Computation time and time until the optimal objective value is reached in seconds for both Objective Functions (4.1) and (5.1), along with the percentage of time that the program was proving that the best objective value found was indeed optimal. Results for the different constraints settings, data set 1 and priority setting 3

Data Set 1 - Prio 4	$\min \sum_{s\in S} \sum_{t\in T} U_{s,t}$			$\min \sum_{s\in S} \sum_{t\in T} Z_{s,t}$		
Constraint Setting	**Comp. Time**	**Time Until Opt. Obj. is Reached**	**Perc. Of Time Proving Optimal**	**Comp. Time**	**Time Until Opt. Obj. is Reached**	**Perc. Of Time Proving Optimal**
1	24	18	25%	1,800*	1476	18%
2	115	115	0%	1,800*	23	99%
3	41	30	27%	1,800*	280	84%
4	60	52	13%	1,800*	177	90%
5	74	73	1%	1,800*	24	99%
6	77	71	8%	1,800*	199	89%

Table D.4: Computation time and time until the optimal objective value is reached in seconds for both Objective Functions (4.1) and (5.1), along with the percentage of time that the program was proving that the best objective value found was indeed optimal. Results for the different constraints settings, data set 1 and priority setting 4

Data Set 2 - Prio 1	$\min \sum_{s\in S} \sum_{t\in T} U_{s,t}$			$\min \sum_{s\in S} \sum_{t\in T} Z_{s,t}$		
Constraint Setting	**Comp. Time**	**Time Until Opt. Obj. is Reached**	**Perc. Of Time Proving Optimal**	**Comp. Time**	**Time Until Opt. Obj. is Reached**	**Perc. Of Time Proving Optimal**
1	39	31	21%	1,310	523	60%
2	53	36	32%	1,088	132	88%
3	115	114	1%	1,120	197	82%
4	55	45	18%	1,800*	140	92%
5	69	43	38%	1,718	142	92%
6	173	171	1%	1,800*	576	68%

Table D.5: Computation time and time until the optimal objective value is reached in seconds for both Objective Functions (4.1) and (5.1), along with the percentage of time that the program was proving that the best objective value found was indeed optimal. Results for the different constraints settings, data set 2 and priority setting 1

Data Set 2 - Prio 2	min $\sum_{s\in S}\sum_{t\in T} U_{s,t}$			min $\sum_{s\in S}\sum_{t\in T} Z_{s,t}$		
Constraint Setting	**Comp. Time**	**Time Until Opt. Obj. is Reached**	**Perc. Of Time Proving Optimal**	**Comp. Time**	**Time Until Opt. Obj. is Reached**	**Perc. Of Time Proving Optimal**
1	69	67	3%	1,800*	1230	32%
2	29	14	52%	1,606	358	78%
3	41	28	32%	1,629	75	95%
4	61	56	8%	1,800*	213	88%
5	62	50	19%	971	79	92%
6	67	40	40%	1,315	54	96%

Table D.6: Computation time and time until the optimal objective value is reached in seconds for both Objective Functions (4.1) and (5.1), along with the percentage of time that the program was proving that the best objective value found was indeed optimal. Results for the different constraints settings, data set 2 and priority setting 2

Data Set 2 - Prio 3	min $\sum_{s\in S}\sum_{t\in T} U_{s,t}$			min $\sum_{s\in S}\sum_{t\in T} Z_{s,t}$		
Constraint Setting	**Comp. Time**	**Time Until Opt. Obj. is Reached**	**Perc. Of Time Proving Optimal**	**Comp. Time**	**Time Until Opt. Obj. is Reached**	**Perc. Of Time Proving Optimal**
1	59	49	17%	1,800	23	99%
2	117	113	3%	1,800	76	96%
3	179	177	1%	1,800	35	98%
4	234	232	1%	1,800	194	89%
5	95	65	32%	1,800	115	94%
6	242	237	2%	1,800	42	98%

Table D.7: Computation time and time until the optimal objective value is reached in seconds for both Objective Functions (4.1) and (5.1), along with the percentage of time that the program was proving that the best objective value found was indeed optimal. Results for the different constraints settings, data set 2 and priority setting 3

Data Set 2 - Prio 4	min $\sum_{s\in S}\sum_{t\in T} U_{s,t}$			min $\sum_{s\in S}\sum_{t\in T} Z_{s,t}$		
Constraint Setting	**Comp. Time**	**Time Until Opt. Obj. is Reached**	**Perc. Of Time Proving Optimal**	**Comp. Time**	**Time Until Opt. Obj. is Reached**	**Perc. Of Time Proving Optimal**
1	56	52	7%	1,800*	1001	44%
2	27	13	52%	1,800*	77	96%
3	112	112	0%	1,800*	369	80%
4	81	79	2%	1,800*	196	89%
5	71	65	8%	1,800*	116	94%
6	67	43	36%	1,800*	36	98%

Table D.8: Computation time and time until the optimal objective value is reached in seconds for both Objective Functions (4.1) and (5.1), along with the percentage of time that the program was proving that the best objective value found was indeed optimal. Results for the different constraints settings, data set 2 and priority setting 4

Data Set 3 - Prio 1	$\min \sum_{s\in S} \sum_{t\in T} U_{s,t}$			$\min \sum_{s\in S} \sum_{t\in T} Z_{s,t}$		
Constraint Setting	**Comp. Time**	**Time Until Opt. Obj. is Reached**	**Perc. Of Time Proving Optimal**	**Comp. Time**	**Time Until Opt. Obj. is Reached**	**Perc. Of Time Proving Optimal**
1	171	101	41%	1,800*	48	97%
2	79	44	44%	1,800*	115	94%
3	96	37	61%	1,800*	99	95%
4	329	70	79%	1,800*	59	97%
5	144	48	67%	1,800*	104	94%
6	88	43	51%	1,800*	101	94%

Table D.9: Computation time and time until the optimal objective value is reached in seconds for both Objective Functions (4.1) and (5.1), along with the percentage of time that the program was proving that the best objective value found was indeed optimal. Results for the different constraints settings, data set 3 and priority setting 1

Data Set 3 - Prio 2	$\min \sum_{s\in S} \sum_{t\in T} U_{s,t}$			$\min \sum_{s\in S} \sum_{t\in T} Z_{s,t}$		
Constraint Setting	**Comp. Time**	**Time Until Opt. Obj. is Reached**	**Perc. Of Time Proving Optimal**	**Comp. Time**	**Time Until Opt. Obj. is Reached**	**Perc. Of Time Proving Optimal**
1	159	87	45%	1,800*	436	76%
2	122	109	11%	1,800*	89	95%
3	79	36	54%	1,800*	104	94%
4	171	75	56%	1,800*	73	96%
5	105	30	71%	1,800*	77	96%
6	97	36	63%	1,800*	41	98%

Table D.10: Computation time and time until the optimal objective value is reached in seconds for both Objective Functions (4.1) and (5.1), along with the percentage of time that the program was proving that the best objective value found was indeed optimal. Results for the different constraints settings, data set 3 and priority setting 2

Data Set 3 - Prio 3	$\min \sum_{s\in S} \sum_{t\in T} U_{s,t}$			$\min \sum_{s\in S} \sum_{t\in T} Z_{s,t}$		
Constraint Setting	**Comp. Time**	**Time Until Opt. Obj. is Reached**	**Perc. Of Time Proving Optimal**	**Comp. Time**	**Time Until Opt. Obj. is Reached**	**Perc. Of Time Proving Optimal**
1	264	198	25%	1,800*	60	97%
2	361	123	66%	1,800*	99	95%
3	140	74	47%	1,800*	115	94%
4	441	95	78%	1,800*	56	97%
5	141	82	42%	1,800*	222	88%
6	150	77	49%	1,800*	70	96%

Table D.11: Computation time and time until the optimal objective value is reached in seconds for both Objective Functions (4.1) and (5.1), along with the percentage of time that the program was proving that the best objective value found was indeed optimal. Results for the different constraints settings, data set 3 and priority setting 3

Data Set 3 - Prio 4	$\min \sum_{s \in S} \sum_{t \in T} U_{s,t}$			$\min \sum_{s \in S} \sum_{t \in T} Z_{s,t}$		
Constraint Setting	Comp. Time	Time Until Opt. Obj. is Reached	Perc. Of Time Proving Optimal	Comp. Time	Time Until Opt. Obj. is Reached	Perc. Of Time Proving Optimal
1	176	119	32%	1,800*	438	76%
2	154	146	5%	1,800*	99	95%
3	79	36	54%	1,800*	111	94%
4	170	75	56%	1,800*	47	97%
5	104	29	72%	1,800*	217	88%
6	118	31	74%	1,800*	67	96%

Table D.12: Computation time and time until the optimal objective value is reached in seconds for both Objective Functions (4.1) and (5.1), along with the percentage of time that the program was proving that the best objective value found was indeed optimal. Results for the different constraints settings, data set 3 and priority setting 4

D.2. Ratio scores for constraint and priority settings

Data Set 1	Computation Time				Time Until Optimal Objective Value is Reached			
Constraint Setting	Prio 1	Prio 2	Prio 3	Prio 4	Prio 1	Prio 2	Prio 3	Prio 4
1	1.00	0.98	1.02	0.50	1.00	1.05	1.09	0.42
2	1.38	1.04	1.56	2.40	1.30	1.09	1.60	2.67
3	0.88	0.94	1.67	0.85	0.33	0.93	1.42	0.70
4	4.27	0.83	0.94	1.25	4.74	0.47	0.56	1.21
5	1.06	0.79	1.65	1.54	1.14	0.77	1.84	1.70
6	1.08	1.58	1.42	1.60	1.12	1.63	0.77	1.65

Table D.13: Ratio scores of all Constraint and Priority settings compared to the values of the first constraint and first priority setting for Data Set 1

Data Set 2	Computation Time				Time Until Optimal Objective Value is Reached			
Constraint Setting	Prio 1	Prio 2	Prio 3	Prio 4	Prio 1	Prio 2	Prio 3	Prio 4
1	1.00	1.77	1.51	1.44	1.00	2.16	1.58	1.68
2	1.36	0.74	3.00	0.69	1.16	0.45	3.65	0.42
3	2.95	1.05	4.59	2.87	3.68	0.90	5.71	3.61
4	1.41	1.56	6.00	2.08	1.45	1.81	7.48	2.55
5	1.77	1.59	2.44	1.82	1.39	1.61	2.10	2.10
6	4.44	1.72	6.21	1.72	5.52	1.29	7.65	1.39

Table D.14: Ratio scores of all Constraint and Priority settings compared to the values of the first constraint and first priority setting for Data Set 2

Data Set 3	Computation Time				Time Until Optimal Objective Value is Reached			
Constraint Setting	Prio 1	Prio 2	Prio 3	Prio 4	Prio 1	Prio 2	Prio 3	Prio 4
1	1.00	0.93	1.54	1.03	1.00	0.86	1.96	1.18
2	0.46	0.71	2.11	0.90	0.44	1.08	1.22	1.45
3	0.56	0.46	0.82	0.46	0.37	0.36	0.73	0.36
4	1.92	1.00	2.58	0.99	0.69	0.74	0.94	0.74
5	0.84	0.61	0.82	0.61	0.48	0.30	0.81	0.29
6	0.51	0.57	0.88	0.69	0.43	0.36	0.76	0.31

Table D.15: Ratio scores of all Constraint and Priority settings compared to the values of the first constraint and first priority setting for Data Set 3

D.3. Results for the Hierarchical Solution Method

Monday	Sections												Total
	1.1	1.2	1.3	2.1	2.2	2.3	2.4	3.1	3.2	3.3	3.4	4	
# Constraints	627	345	1	1	627	1	1	1	1	1	1	1112	2719
# Variables	61	34	1	1	61	1	1	1	1	1	1	267	431

Table D.16: Number of constraints and variables per section in the second phase of the hierarchical solution method, compiled for the optimal and feasible solution of the first phase for Data Set 1

Tuesday	Sections												Total
	1.1	1.2	1.3	2.1	2.2	2.3	2.4	3.1	3.2	3.3	3.4	4	
# Constraints	1	615	1	1	1	1	1	1	1	1	1	1,433	2,058
# Variables	1	106	1	1	1	1	1	1	1	1	1	609	725

Table D.17: Number of constraints and variables per section in the second phase of the hierarchical solution method, compiled for the optimal and feasible solution of the first phase for Data Set 2

Wednesday	Sections												Total
	1.1	1.2	1.3	2.1	2.2	2.3	2.4	3.1	3.2	3.3	3.4	4	
# Constraints	1	1	1178	1	1	1	1	1	1	1	1	1,695	2,883
# Variables	1	1	201	1	1	1	1	1	1	1	1	571	782

Table D.18: Number of constraints and variables per section in the second phase of the hierarchical solution method, compiled for the optimal and feasible solution of the first phase for Data Set 3

D.4. Results for all Combinations of Symmetry Breaking Constraints

Data Set 1	30 Groups			50 Groups		
SBC Setting	Comp. Time	Time Until Opt. Obj. is Reached	Perc. Of Time Proving Optimal	Comp. Time	Time Until Opt. Obj. is Reached	Perc. Of Time Proving Optimal
1	606	400	34%	726	163	78%
2	1,127	1,125	0%	1,800*	1,800*	0%
3	750	748	0%	1,010	977	3%
4	1,737	1,718	1%	1,800*	1,800*	0%

Table D.19: Computation time and time until the optimal objective value is reached in seconds for different number of groups, along with the percentage of time that the program was proving that the best objective value found was indeed optimal. Results for the different SBC settings for data set 1

Data Set 2	30 Groups			50 Groups		
SBC Setting	Comp. Time	Time Until Opt. Obj. is Reached	Perc. Of Time Proving Optimal	Comp. Time	Time Until Opt. Obj. is Reached	Perc. Of Time Proving Optimal
1	299	269	10%	1,058	1,043	1%
2	859	737	14%	1,800*	1,800*	0%
3	1,756	1,752	0%	1,800*	1,800*	0%
4	1,800*	1,800*	0%	1,800*	1,800*	0%

Table D.20: Computation time and time until the optimal objective value is reached in seconds for different number of groups, along with the percentage of time that the program was proving that the best objective value found was indeed optimal. Results for the different SBC settings for data set 2

Data Set 3	30 Groups			50 Groups		
SBC Setting	Comp. Time	Time Until Opt. Obj. is Reached	Perc. Of Time Proving Optimal	Comp. Time	Time Until Opt. Obj. is Reached	Perc. Of Time Proving Optimal
1	714	505	29%	958	587	39%
2	914	833	9%	1,271	833	34%
3	628	352	44%	1,087	791	27%
4	1,800*	1,560	13%	1,800*	1,800*	0%

Table D.21: Computation time and time until the optimal objective value is reached in seconds for different number of groups, along with the percentage of time that the program was proving that the best objective value found was indeed optimal. Results for the different SBC settings for data set 3

D.5. Ratio scores for symmetry breaking constraints settings

Data set 1	Computation Time		Time Until Optimal Objective Value is Reached	
SBC Setting	30 Groups	50 Groups	30 Groups	50 Groups
1	1.00	1.20	1.00	0.41
3	1.86	2.97**	2.81	4.50**
2	1.24	1.67	1.87	2.44
4	2.87	2.97**	4.30	4.50**

Table D.22: Ratio scores of all symmetry breaking constraints settings compared to the values of the first constraint and first priority setting for Data Set 1

Data set 2	Computation Time		Time Until Optimal Objective Value is Reached	
SBC Setting	30 Groups	50 Groups	30 Groups	50 Groups
1	1.00	3.54	1.00	3.88
3	2.87	6.02**	2.74	6.69**
2	5.87	6.02**	6.51	6.69**
4	6.02**	6.02**	6.69**	6.69**

Table D.23: Ratio scores of all symmetry breaking constraints settings compared to the values of the first constraint and first priority setting for Data Set 2

Data set 3	Computation Time		Time Until Optimal Objective Value is Reached	
SBC Setting	30 Groups	50 Groups	30 Groups	50 Groups
1	1.00	1.34	1.00	1.16
3	1.28	1.78	1.65	1.65
2	0.88	1.52	0.70	1.57
4	2.53**	2.52 **	3.09	3.56**

Table D.24: Ratio scores of all symmetry breaking constraints settings compared to the values of the first constraint and first priority setting for Data Set 3

E

Complete Results for Part III

Monday	Sections												Total
	1.1	**1.2**	**1.3**	**2.1**	**2.2**	**2.3**	**2.4**	**3.1**	**3.2**	**3.3**	**3.4**	**4**	
# Constraints	1358	1528	1881	1151	1381	2027	753	2053	2415	833	700	9,461	25,541
# Variables	311	383	449	151	251	535	203	423	715	159	93	10,033	13,706

Table E.1: Number of constraints and variables per section in the second phase of the hierarchical solution method, compiled for the optimal and feasible solution of the first phase for the Monday data set

Tuesday	Sections												Total
	1.1	**1.2**	**1.3**	**2.1**	**2.2**	**2.3**	**2.4**	**3.1**	**3.2**	**3.3**	**3.4**	**4**	
# Constraints	1,368	1,369	1,434	667	1,199	1,378	814	2,561	2,065	1,539	1,129	9,561	25,084
# Variables	215	197	223	107	161	283	307	725	593	439	221	10,223	13,694

Table E.2: Number of constraints and variables per section in the second phase of the hierarchical solution method, compiled for the optimal and feasible solution of the first phase for the Tuesday data set

Wednesday	Sections												Total
	1.1	**1.2**	**1.3**	**2.1**	**2.2**	**2.3**	**2.4**	**3.1**	**3.2**	**3.3**	**3.4**	**4**	
# Constraints	1,278	1,746	2,293	1,361	1,357	2,068	436	2,537	2,053	990	818	9,147	26,084
# Variables	249	449	679	237	263	559	79	665	469	147	123	9,653	13,572

Table E.3: Number of constraints and variables per section in the second phase of the hierarchical solution method, compiled for the optimal and feasible solution of the first phase for the Wednesday set

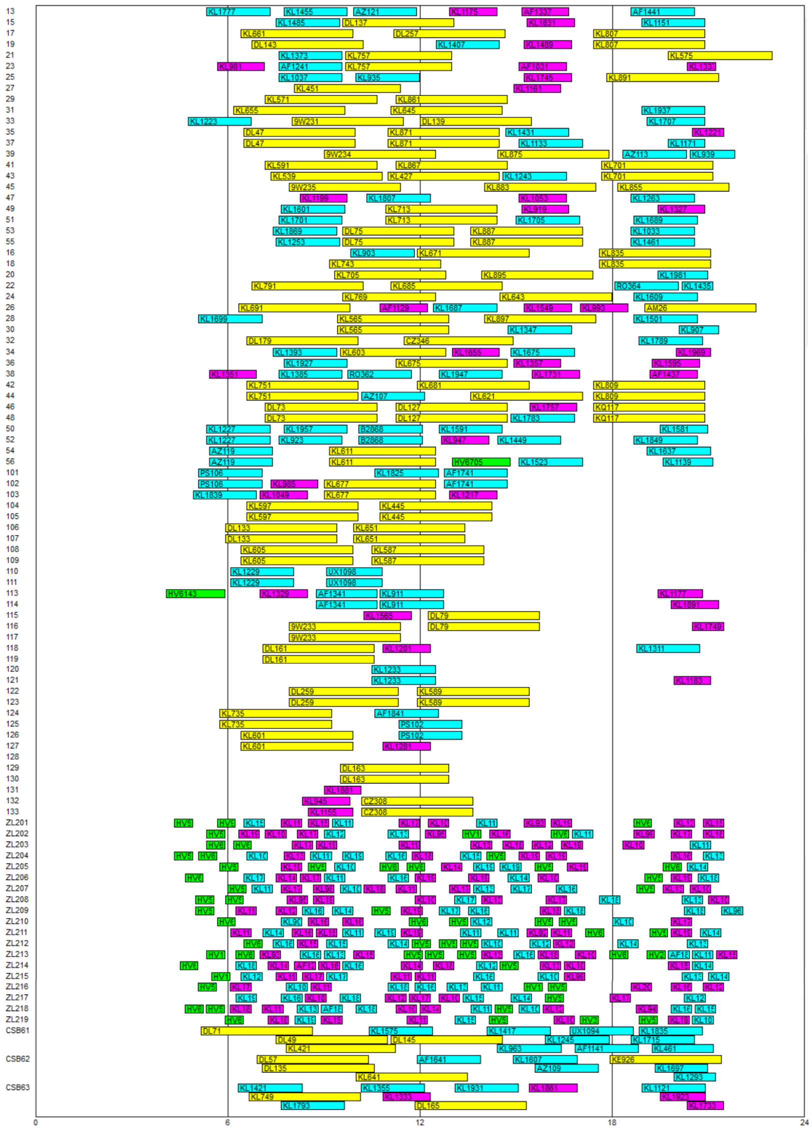


Figure E.1: Lateral planning for the hierarchical method for the Tuesday set

F

Complete Results for Part IV

Monday	Sections												Total
	1.1	**1.2**	**1.3**	**2.1**	**2.2**	**2.3**	**2.4**	**3.1**	**3.2**	**3.3**	**3.4**	**4**	
# Constraints	1,459	1,547	1,467	1,330	1,357	1,768	406	3,527	2,504	1,773	1,001	8,821	26,960
# Variables	323	389	431	235	247	359	115	1,205	627	491	141	8,817	13,380

Table F.1: Number of constraints and variables per section in the second phase of the hierarchical solution method, compiled for the optimal and feasible solution of the first phase for the Monday set

Tuesday	Sections												Total
	1.1	**1.2**	**1.3**	**2.1**	**2.2**	**2.3**	**2.4**	**3.1**	**3.2**	**3.3**	**3.4**	**4**	
# Constraints	1,382	1,628	2,057	1,341	1,486	1,819	269	3,078	2,526	1,630	986	8,807	27,009
# Variables	239	339	521	217	235	411	31	1,123	769	449	141	9,007	13,482

Table F.2: Number of constraints and variables per section in the second phase of the hierarchical solution method, compiled for the optimal and feasible solution of the first phase for the Tuesday set

Wednesday	Sections												Total
	1.1	**1.2**	**1.3**	**2.1**	**2.2**	**2.3**	**2.4**	**3.1**	**3.2**	**3.3**	**3.4**	**4**	
# Constraints	1,357	1,639	1,546	1,365	1,416	1,039	410	3,127	2,391	1,848	1,141	8,547	25,826
# Variables	273	395	395	251	247	193	103	1,233	721	429	203	8,513	12,956

Table F.3: Number of constraints and variables per section in the second phase of the hierarchical solution method, compiled for the optimal and feasible solution of the first phase for the Wednesday set