

Optimizing and Analyzing Staff Scheduling Under Disruptions in Airport Baggage Handling

J. M. Capel

Technische Universiteit Delft



Optimizing and Analyzing Staff Scheduling Under Disruptions in Airport Baggage Handling

by

J. M. Capel

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday October 31, 2025 at 13:00.

Student number: 4905180
Project duration: January 27, 2025 – October 31, 2025
Thesis committee: Prof. dr. D. Gijswijt, TU Delft, supervisor
Dr. G. F. Nane, TU Delft
Dr. T. Pippia, KLM

Cover: Aron Marinelli / Unsplash

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Preface

This thesis presents my graduation research for completing the masters in Applied Mathematics at Delft University of Technology. With this thesis, I also conclude my years as a student. Looking back, I am grateful for the friendships, experiences and opportunities that have shaped my journey in Delft. I would like to thank my supervisor at TU Delft, Dion Gijswijt, for all the nice meetings, his guidance, and constructive feedback throughout this project. Your support has been very helpful in shaping both the direction and quality of my research.

I am also very grateful to Tomas at KLM. I have greatly benefited from our weekly meetings, where we discussed my progress, challenges I encountered, and new ideas that helped move the project forward. Thank you as well for sharing your expertise and providing insides into the operational context of my research. Furthermore I want to thank everyone at KLM in the department for making me feel so welcome, I greatly enjoyed our lunches, coffee breaks and the many walks after lunch. You all made the work environment a really enjoyable place.

Finally, I want to thank my family and friends for their continuous support. Thank you for listening to my doubts, for the encouragement when things got challenging, and for always believing in me. And especially sorry for all the times I needed to share my worries, but your support has meant so much to me.

I hope you enjoy reading this thesis.

*J. M. Capel
Delft, October 2025*

Abstract

Efficient staff scheduling is crucial for smooth baggage handling at Schiphol Airport. Each day, KLM must allocate hundreds of employees to hundreds of time-sensitive loading and unloading tasks. Even small disruptions, such as flight delays or staff shortages, can quickly lead to infeasible schedules and unhandled flights. This thesis develops an optimization framework to provide more insight into the robustness of KLM's day-ahead baggage handling schedule.

The problem is modeled as a Vehicle Routing Problem with Time Windows (VRPTW), extended mode selection, travel times, and mandatory breaks. To handle the model's complexity, a Fix-and-Optimize approach is applied, iteratively optimizing subsets of the problem while keeping others fixed. Two extensions are introduced to improve task coverage: allowing staff to assist across work areas (cross-zone assignments) and slightly delaying flights (postponements) when necessary.

Scenario analyses evaluate how disruptions, flight delays and staff shortages, affect task coverage. Results based on real KLM data show that the Base Model already increases the number of planned tasks compared to the current planning tool when comparing optimization for a full shift. Allowing postponements further improves coverage, often enabling full task completion. Under disruptions, the model identifies critical time periods and areas where additional flexibility or actions are required.

This research combines mathematical modeling and analysis to better understand the structure and resilience of KLM's daily planning. The proposed approach highlights where schedules are vulnerable and demonstrates how optimization can support planners in preparing for operational uncertainty.

Contents

Preface	iii
Abstract	iv
1 Introduction	1
1.1 Background and motivation	1
1.2 Research questions	3
1.3 Structure of the thesis	3
2 Operational Context and Planning Problem	5
2.1 Operational context.	5
2.2 Planning problem description	7
2.3 Relation to general scheduling problems	7
3 Current Method	8
3.1 Operational planning with GroundOpt.	8
3.2 System infrastructure.	8
3.2.1 Preprocessing	8
3.2.2 Optimizer	9
3.2.3 Postprocessing	9
3.3 The model	10
4 Literature Review	11
4.1 Workforce allocation in aviation	11
4.1.1 General challenges and approaches	11
4.1.2 Operational staff allocation.	12
4.2 The vehicle routing problem	12
4.2.1 Mapping to staff allocation	14
4.3 Positioning of this research	14
5 Mathematical Model	16
5.1 Choice of model	16
5.2 Base Model	16
5.2.1 Model input	17
5.2.2 Sets and variables	18
5.2.3 Objective function	19
5.2.4 Constraints	19
5.3 Model Extensions for increased task coverage	21
5.3.1 Cross-Zone Assignment	21
5.3.2 Postpone Model	22
5.4 Modeling disruption scenarios	24
6 Solution Method	26
6.1 Motivation for an iterative approach	26
6.2 Fix-and-Optimize base model	27
6.3 Fix-And-Optimize with Cross-Area assignments	29
6.4 Fix-and-Optimize with Postponements	29
6.5 Infrastructure integration	30
6.5.1 Base model	31
6.5.2 Cross-zone	31
6.5.3 Postponement	31
6.5.4 Disruptions	32

7	Results	33
7.1	Experimental setup	33
7.2	Detailed results of <i>Instance 1</i>	35
7.2.1	Overall comparison of Base and Postpone Model	36
7.2.2	Impact of flight delay disruption	39
7.2.3	Impact of Staff Shortage disruption	40
7.2.4	Flights postponed.	41
7.3	Results of the other instances	44
8	Conclusion and Discussion	48
8.1	Conclusion	48
8.2	Discussion	49
A	Preliminary results cross zone model	53
B	Results Base and Postpone Models	54
C	Overview postponed flights	64

Introduction

1.1. Background and motivation

When travelers go on holiday and check in their suitcases at the airport, they rarely give it another thought. They board the aircraft expecting their luggage to travel with them and arrive safely at their destination. Behind the scenes, however, fulfilling this seemingly simple expectation requires hundreds of staff members to handle thousands of bags each day, often within a very tight time window. Take for example a flight arriving from Berlin in the morning: within less than two hours this aircraft must be on its way to Bordeaux again. In this small time window a lot of tasks in and around the aircraft need to be done, by multiple different teams. A single delayed flight or an unexpected staff absence can trigger a domino effect: bags remain behind, teams cannot handle the flights within their working hours, and passengers may not receive their luggage.

This example illustrates the operational challenges at an airport, a complex environment where efficient task allocation is crucial for smooth daily operations. Many different activities must be completed around each aircraft within strict time windows, ranging from catering and cleaning to baggage handling and push-back. A single disruption in one of these processes can have big consequences on the rest of the operation. Figure 1.1 illustrates this turnaround process, showing how tasks such as disembarking, refueling, catering, and baggage handling must be carried out in parallel or sequence between the scheduled arrival time (SIBT in the figure) and departure time (SOBT in the figure). The figure shows the inter dependencies between tasks, where a delay in one step can directly affect subsequent activities.

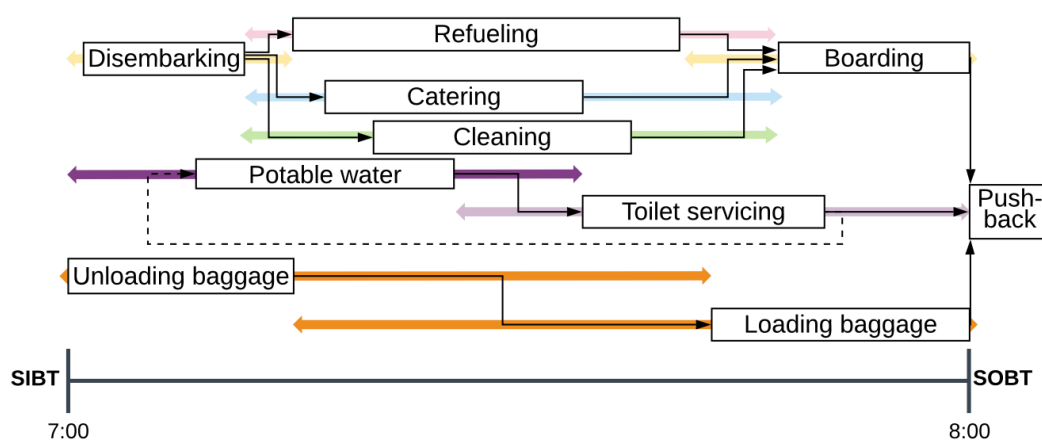


Figure 1.1: Illustration of an aircraft turnaround operation and its task dependencies [1].

In this thesis, we focus on baggage handling, specifically the loading and unloading of baggage into the airplanes. At Schiphol Airport, KLM Royal Dutch Airlines handles thousands of bags each day, requiring hundreds of staff to be assigned to geographically-defined and time-sensitive tasks. Figure 1.2 shows part of an operational roster which serves as the visual interface used by planners to construct the rosters and assign the tasks. In this representation, each horizontal line corresponds to an individual worker, while the colored blocks placed along these lines denote the assigned tasks or breaks. Specifically, the blue blocks represent the tasks of loading and unloading activities, and the light yellow blocks indicate the scheduled breaks. It is important to note that it might seem like some tasks appear multiple times in the figure, but this is because the plane handling is a team effort. The loading and unloading of an aircraft cannot be done by a single worker therefore multiple workers must be assigned to the “same” task. The figure also shows some workers who appear to have overlapping tasks. Normally, overlapping tasks are not allowed, but these correspond to a turnaround, where an arriving aircraft is unloaded and then loaded again for its next departure. Although these two tasks may overlap in time, they can legitimately be performed by the same worker, as they concern different flights but the same aircraft. As it can be seen there are also some gray blocks, which represent assignments outside the current optimization window, meaning they are fixed in the schedule and can not be moved. Furthermore, the purple blocks on the bottom of the figure represent the unplanned tasks, for which no available worker is found yet. This scenario demonstrates the complexity of a planner’s job, which is essentially a challenging puzzle. The core difficulty lies in attempting to place these remaining purple blocks into the roster while simultaneously ensuring no operational constraints are violated. The figure thus highlights that finding a solution is not immediately obvious, underscoring the potential usefulness of planning and optimization techniques to create complete and efficient rosters.

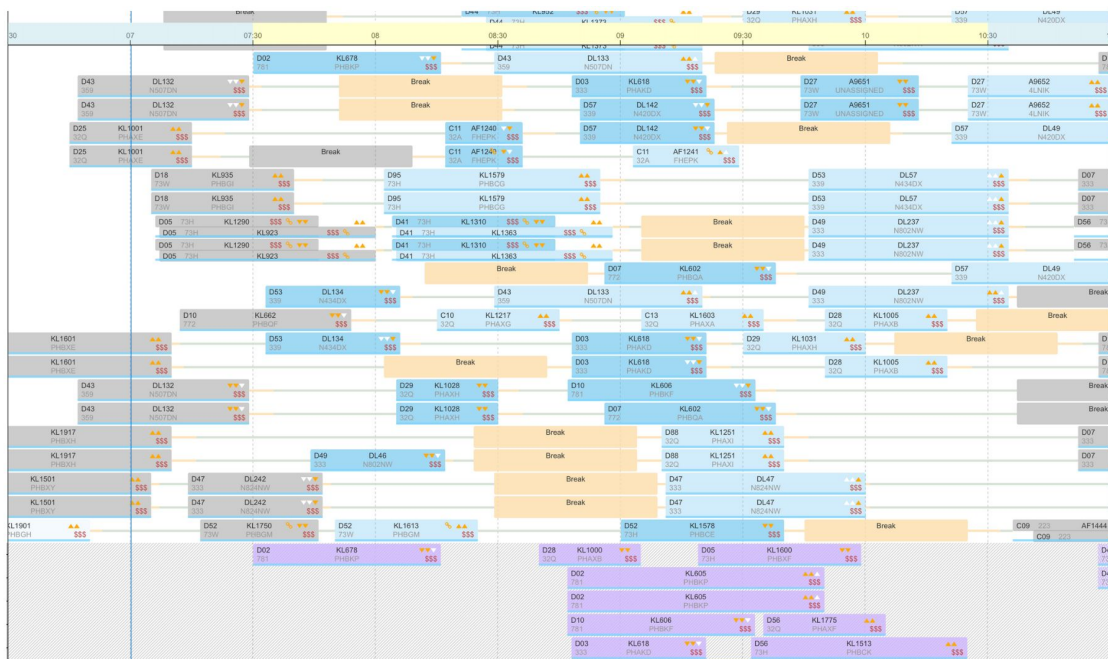


Figure 1.2: Snapshot of a schedule for the assignment of the baggage handling tasks to shift members

To solve this scheduling puzzle, KLM relies on planners who manually assign staff to tasks, supported by an optimization tool referred to in this thesis as GroundOpt. This tool is able to quickly generate feasible schedules for a short time horizon, taking into account constraints such as staff qualifications, task start times, workload balance, and availability. While GroundOpt is effective at producing short-term task assignments, it does not provide planners with insight into how the schedule behaves when disruptions occur. For example, a delayed flight or unexpected staff absence may make some tasks impossible to complete on time. Planners would benefit from greater insight into the structure of the schedule, such as where weak points or potential bottlenecks exist. Understanding these issues would allow them to anticipate problems before they arise. In addition, having knowledge of possible correc-

tive actions, e.g., delaying certain tasks or reassigning staff would help planners increase the number of tasks that can be successfully executed.

This need for insight is not only necessary in the operations of KLM. Robust scheduling, where the feasibility of a plan under disruptions is evaluated, is essential in many industries where resources are limited and time-sensitive operations must be maintained. By evaluating how robust a schedule is, organizations can anticipate bottlenecks and reduce risks for both operations and staff well-being. At airports, this is particularly important, as high work pressure and personnel shortages can lead to stress, missed breaks, and even safety risks [2, 3, 4].

In this thesis, we develop a baggage handling scheduling model and use scenario analysis to evaluate the impact of disruptions on the schedule. The model aims to schedule as many tasks as possible, while considering actions that can increase the number of tasks assigned, such as delaying certain flights or reallocating staff. In addition, the goal is to simulate disruptions, such as flight delays or staff unavailability, and evaluate their effects on the schedule. Although the focus is on KLM's baggage handling operations, the methodology contributes more broadly to the study of operational planning under uncertainty.

1.2. Research questions

To address the challenges outlined above, this thesis aims to provide more insights into the planning and robustness of the next day's schedule for staff responsible for loading and unloading baggage. To achieve this goal, at the beginning of the thesis project the following main research question have been formulated.

Main Research Question: How can more insights be gained into the next day's schedule for staff responsible for loading and unloading baggage, considering potential disruptions?

To guide the research towards answering this main question, the following sub-questions will be investigated:

1. How can we accurately model the scheduling process for the staff responsible for loading and unloading baggage at KLM?
2. How can we analyze if all tasks can be completed within each work-area without considering disruptions?
3. Which actions can we take to maximize the number of tasks planned and how can we incorporate this into the model?
4. How can we change/reformulate the model to integrate:
 - A) flight delays?
 - B) staff (un)availability?
5. How can we measure the impact of the disruptions?

By addressing these sub-questions, this thesis aims to create feasible schedules and to provide actionable insights for planners. In particular, a key focus is on identifying strategies and actions that maximize the number of tasks that can be successfully scheduled, while also anticipating bottlenecks and measuring the impact of potential disruptions. This approach enables planners to improve schedule robustness, maintain safe and efficient operations, and make informed decisions under uncertainty.

1.3. Structure of the thesis

The remainder of this thesis is organized as follows. Each chapter builds upon the previous one, gradually moving from the operational context to the development, implementation, and evaluation of the proposed model.

- **Chapter 2** introduces the operational context of KLM's baggage handling at Schiphol Airport. It provides a detailed description of the assignment problem that arises in day-to-day operations and highlights the challenges that motivate this research.
- **Chapter 3** outlines the current planning approach used by KLM to assign staff members to loading and unloading tasks. The GroundOpt optimization tool, together with the mathematical model on which it is based, is discussed. Understanding this method provides the foundation for the new approaches developed in this thesis.
- **Chapter 4** provides a review of the relevant literature on workforce allocation and related optimization problems. It discusses general staff scheduling challenges, operational allocation methods, and exact and heuristic solution approaches. The chapter also presents the Vehicle Routing Problem (VRP) and illustrates how the staff allocation problem at KLM can be mapped to this class of problems. Finally, it positions this research within the literature, highlighting the specific combination of constraints and operational challenges considered in this thesis.
- **Chapter 5** presents the mathematical model developed to address the baggage handling scheduling problem. First, the motivation for the modeling choices is discussed. Then, the different models are introduced: the base model, which captures the assignment problem without corrective actions, and two model extensions. The first extension incorporates the possibility of reassigning tasks to another area, while the second allows for postponing flights (and their associated tasks) to increase the number of feasible assignments. For each model, the objective function, decision variables, and constraints are described in detail.
- **Chapter 6** presents the solution method used to solve the staff assignment problem. Because the model is large and complex, solving it directly with a solver can take a long time. To handle this, the chapter explains an iterative Fix-and-Optimize approach, where only a subset of teams or shifts is optimized at a time while keeping the rest fixed. The chapter also describes how this method is extended to handle flight delays, allowing tasks to be postponed in order to increase the number of planned tasks.
- **Chapter 7** presents the results of the optimization models. It includes a comparison between the Base Model and the Postpone Model, analyzes the impact of flight delays and staff shortages, and discusses insights on task coverage, delayed flights, and model performance under different disruption scenarios.
- Finally, **Chapter 8** summarizes the main findings of the thesis and reflects on their implications. It answers the research question and subquestions, discusses the strengths and limitations of the developed models, and provides suggestions for future research to improve robustness, fairness, and practical applicability.

2

Operational Context and Planning Problem

This chapter describes the operational planning problem addressed in this thesis. The aim is to clearly define the context of the problem.

2.1. Operational context

Luggage handling at Schiphol Airport is complex. After passengers check in, their luggage enters the baggage handling system, operated by Schiphol Airport Group. This system transfers the bags from the check-in counters to the airside area, the part of the airport accessible only after passing through security. At this point, baggage carts collect the luggage and transport it to the aircraft. Once the luggage arrives at the airside, KLM takes over and is responsible for loading it onto the aircraft.

To efficiently assign ground handling staff to loading and unloading tasks, KLM has divided Schiphol Airport into three distinct work areas. Each area covers a different section of the airport and corresponds to specific types of aircraft:

- **K1** handles KLM Cityhopper flights using smaller narrow-body (NABO) aircraft.
- **K4** manages medium-haul flights, also using NABO aircraft, typically larger than those in K1.
- **K5/6** covers intercontinental flights operated with wide-body (WIBO) aircraft.

Different aircraft types require different baggage handling procedures. For NABO aircraft, baggage is manually loaded using a conveyor belt and stored in two holds: forward (FWD) and aft (AFT) (Figure 2.1a). In contrast, WIBO aircraft use containerized luggage, which is loaded into the FWD and AFT holds using specialized equipment (Figure 2.1b). A small portion of irregular baggage may still be loaded separately in the bulk (BLK) compartment.



(a) Narrow-body [5]



(b) Wide-body [6]

Figure 2.1: Two images of the loading of baggage in two different types of aircraft

Shifts and teams

For the problem considered in this thesis, the available shifts are known. Each shift corresponds with a specific staff member, with a start and end time. Furthermore, each staff member is already assigned to one of the three areas and each member has specific qualifications, indicating which tasks the person is allowed to perform. To reduce complexity and increase employee satisfaction, workers are pre-assigned into teams of two. These teams remain unchanged during the day and are treated mostly as one unit in the planning process.

Aircraft, holds, and task modes

As explained earlier, different aircraft types have distinct baggage handling requirements. To support the assignment process, the planning system works with predefined task modes. A mode specifies both the number of staff members and their role composition required to perform the loading and unloading of a given aircraft type and its holds.

For narrow-body (NABO) aircraft, both the smaller ones in K1 and the slightly bigger ones, there are two modes. In the simplest case, two staff members handle both the forward (FWD) and the aft (AFT) holds. For the second configuration, we have two staff members handle the FWD hold and two staff members handling the AFT hold, resulting in a total of 4 staff members handling the flight. In addition, for some specific flights a dedicated mode is used that always involves four staff members in total. Wide-body (WIBO) aircraft require larger teams due to containerized baggage handling. Here, two standard modes exist: one with four staff members and one with five staff members. The different modes are summarized in Table 2.1.

Table 2.1: Overview of task modes for baggage handling.

Mode	Aircraft type	Number of staff members
1	K1 NABO	2
2	K1 NABO	4
3	Larger NABO	2
4	Larger NABO	4
5	Specific flights	4
6	WIBO	4
7	WIBO	5

The durations and start times of baggage handling tasks depend on the selected mode. In general, assigning more staff to a flight reduces the duration of the tasks.

As an illustration, consider a Warsaw–Amsterdam flight operated with a larger NABO aircraft. This aircraft can be handled in either mode 3 or mode 4. In mode 3, two staff members are assigned, resulting in two tasks that cover both the FWD and AFT holds. In mode 4, four staff members are assigned, creating two tasks for the FWD hold and two for the AFT hold. The corresponding task times are summarized in Table 2.2.

Table 2.2: Task durations for a Warsaw–Amsterdam flight (larger NABO aircraft)

Mode	Hold(s)	Time window	Duration
3	FWD + AFT	05:56–06:30	34 min
4	FWD	05:56–06:23	27 min
4	AFT	05:56–06:09	13 min

As you can see in the Table 2.2 the start times for those arrival tasks are the same. In general, this is the case for arrival tasks. For departure tasks, in general, the tasks end at the same time but the start times differ.

2.2. Planning problem description

The goal of the planning process is to generate a feasible schedule that assigns as many loading and unloading tasks as possible to the available teams. Each task represents the baggage handling for a specific aircraft, hold, and movement (either loading or unloading), and is associated with a time window, duration, and required mode.

The assignment of these tasks must respect various constraints, including:

- Only qualified personnel can be assigned to a task.
- A shift is only able to perform one task at a time, meaning tasks assigned to the same shift are not allowed to overlap.
- Travel times between consecutive tasks must be taken into account, ensuring that tasks do not overlap and that sufficient time is reserved for transitions.
- Each full-time shift includes a mandatory 40-minute break, which must be scheduled within a predefined time window. The break must start no earlier than 2 hours after the shift begins and the break needs to end no later than two hours before the end of the shift.
- For each flight, a mode must be selected, which determines both the number of tasks and their timing.

In the current process, the planning tool GroundOpt generates a schedule that incorporates many operational constraints, including a balanced workload among employees. The goal of this thesis is not to improve the GroundOpt-generated plan, but rather to build a model that provides insights into the overall structure and feasibility of the schedule hours before the operation.

2.3. Relation to general scheduling problems

Scheduling problems with strict time windows and geographical constraints arise in many domains such as logistics, healthcare, and maintenance scheduling [7]. Although the problem studied in this thesis originates from airline operations, its structure is representative of this broader class of problems. In essence, the problem consists of a set of resources and a set of tasks, with the objective to assign as many tasks as possible to the available resources. Several elements are particularly important here: the strict time windows, travel times, mandatory breaks and the presence of different modes. Both the resources and the tasks are subject to strict time windows: start and end times are fixed (and therefore also the duration) and cannot be altered. Moreover, tasks are tied to specific locations, which implies that the travel times between tasks must be taken into account, in contrast to many classical scheduling problems where tasks can be planned consecutively without additional delay. Another important characteristic is that some resources are required to take a break, which can be modeled as additional tasks. Unlike the regular tasks, however, these breaks must be assigned to specific resources and offer more flexibility in terms of timing. Finally, each flight can be executed in different modes where the selected mode determines the number of staff needed and the corresponding timing of the tasks. The mode choice therefore directly influences the feasibility of the schedule. This adds an additional layer of complexity compared to standard resource–task assignment problems.

The problem can thus be viewed as a variant of resource-constrained scheduling problems with time windows, closely related to vehicle routing and workforce scheduling.

Consequently, the problem addressed in this thesis should not be regarded as unique to airline operations, but rather as a concrete instance of a class of resource constrained scheduling problems. For the sake of clarity and readability, this thesis will nevertheless adopt the terminology specific to the airline context in which the problem arises.

3

Current Method

This chapter explains how KLM currently assigns staff to baggage handling tasks using the GroundOpt optimizer. Although this thesis does not aim to replace GroundOpt, it builds upon its principles and structure, which serve as a foundation for the approach developed for this research. Understanding the current method provides important context and highlights aspects of staff assignment considered in the model presented in Chapter 5.

3.1. Operational planning with GroundOpt

KLM uses the optimizer GroundOpt to support manual planners in assigning staff to tasks on the day of operation. GroundOpt incorporates the operational context as described in Chapter 2. The optimizer is used in real-time and continuously updates the schedule, ensuring the schedule becomes more accurate as new information becomes available.

The main optimization window for which GroundOpt assigns tasks spans approximately 3 hours. To ensure a fast runtime, this window is divided into smaller iteration windows, each covering a shorter time interval. The number and duration of these iteration windows depend on the number of tasks within the main window. By solving smaller subproblems sequentially, the optimizer can quickly generate partial schedules. This fast runtime is essential, as GroundOpt operates in real-time and must continuously adapt to new information during the day.

3.2. System infrastructure

To understand how GroundOpt transforms data into feasible staff schedules, it is important to examine its software architecture. The optimizer is implemented in Python and structured into three main components: preprocessing, optimization, and postprocessing. An overview of these stages can be found in Figure 3.1. Together, these components transform the data into a feasible schedule that can be applied by planners.

3.2.1. Preprocessing

In the preprocessing stage, input data from the operational environment is provided in a structured JSON format. The main goal of this stage is to convert the input into components required by the optimizer and to determine the iteration structure for the optimization part. As described in Section 3.1, the full optimization window in GroundOpt is divided into smaller iteration windows. The preprocessing

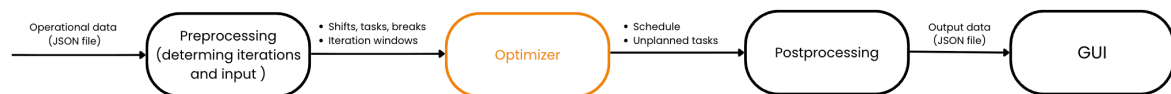


Figure 3.1: Overview of the infrastructure of the GroundOpt optimizer

step establishes these iteration windows and generates the following components, which serve as input for the optimization step:

- *Optimization window*: defined by a start and end time for the current iteration.
- *Current schedule*: a dictionary mapping each active shift to a list of tasks that have already been assigned to that shift.
- *Tasks*: a list of all tasks within the current iteration window that can be scheduled, including tasks from different modes.
- *Tasks in the full optimization window*: tasks that must be scheduled within the complete optimization window.
- *Shifts*: a list of all available shifts within the full optimization window that can be assigned tasks.
- *Breaks in the iteration window*: a dictionary indicating, for each shift, whether its break should be scheduled in the current iteration window. This is important because GroundOpt schedules breaks iteratively.
- *Run parameters*: various parameters defining the duration, type, and iteration index for the current run.

3.2.2. Optimizer

During the optimization stage, the preprocessed input is used to construct and solve the staff assignment problem. GroundOpt employs a Mixed-Integer Programming (MIP) approach to generate feasible assignments of tasks to shifts within the optimization window, as described in more detail in the following section, Section 3.3. This stage represents the core of the system. It is important to note that this optimization step is executed for each iteration window separately, meaning that the full optimization process is repeated multiple times.

The optimization stage itself consists of several sub-steps:

1. *Data preparation*: The input is further processed to generate sets and structures required for the model. For example, for each shift, the set of tasks it can potentially perform is determined based on its available time. Additionally, relevant information is extracted from the tasks, such as which flights need to be handled within the current iteration window.
2. *Model construction*: Using these sets, the MIP model is built with the corresponding variables and constraints.
3. *Model solving*: The constructed model is solved using the chosen solver.
4. *Output processing*: The results from the solver are transformed into the final outputs of the optimization stage. This consists of the generated schedule, which maps each shift to its assigned tasks, and the unplanned tasks, which lists tasks that could not be scheduled within the current iteration window.

In this thesis, the newly developed model will replace this optimization stage. Outside of this stage, in the rest of the system, including preprocessing and postprocessing, only a few changes need to be done.

3.2.3. Postprocessing

In the final postprocessing stage, the outputs from each iteration window are combined and reformatted into a JSON file again. These results are then presented through the graphical user interface (GUI), which supports KLM's planners in their daily operations. An example of the GUI output is shown in Figure 1.2.

3.3. The model

The optimization stage of GroundOpt, described in the previous section, is implemented using a mathematical model solved as a Mixed-Integer Program (MIP) with Gurobi [8]. This section provides a description of that model, which is responsible for generating feasible assignments of tasks to shifts within each iteration window.

GroundOpt's planning model uses an activity slot representation for each shift. In this model, a shift consists of a fixed number of consecutive activity slots. Each slot can either remain empty, be assigned to a task, or be filled with a break. The slots are ordered linearly, and a slot can only be used if all preceding slots in the shift are already filled. This enforces a logical and sequential assignment of activities within a shift.

Objectives

The optimization balances three main objectives:

1. Minimizing operational costs and penalties.
This includes minimizing the cost of unhandled flights and the sum of various penalties. Penalties apply to undesired task–shift combinations, modifications to the current schedule, and splitting teams across multiple tasks.
2. Scheduling breaks effectively.
Most shifts must contain a break within a specified break window. Each shift also has a target break time, and deviations from this target are penalized.
3. Balancing workload.
The third objective promotes a fair distribution of workload across all available teams and shifts.

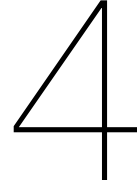
The model defines a set of variables to assign tasks to shifts and uses several constraints to ensure a feasible schedule. We do not list all variables and constraints explicitly, but describe their general structure below.

Variables:

- Assignment variables: Indicate whether a task is assigned to a specific activity slot within a shift.
- Mode variables: Indicate the selected operation mode for each flight.
- Penalty variables: Track violations such as undesired task–shift combinations or team splits.
- Break variables: Indicate whether and when a break is assigned to a shift.

Constraints:

- Each task must be assigned to at most one shift.
- Tasks assigned to the same shift must not overlap in time.
- Breaks must be scheduled within the allowed time window.
- For each flight, exactly one mode must be selected, and all corresponding tasks must be assigned accordingly or remain unplanned.



Literature Review

The efficient allocation of ground handling staff is a critical operational challenge for airlines and airport operators like KLM. Ground handling operations, including baggage loading and unloading, are time-sensitive, resource-intensive, and subject to significant variability due to flight delays, weather conditions, and other operational uncertainties. This chapter reviews the relevant literature to provide context for the staff allocation problem addressed in this thesis. We will explore workforce allocation strategies, particularly within the aviation context and delve into the structural related Vehicle Routing Problem with Time Windows (VRPTW). This review aims to identify relevant methodologies applicable to the KLM staff allocation challenge.

4.1. Workforce allocation in aviation

4.1.1. General challenges and approaches

Workforce allocation in general, also known as staff scheduling, or manpower planning, is a well studied subject in the field of operational research. The general goal is to assign employees to shifts or tasks over a specific planning horizon, to meet demand while satisfying various constraints (e.g. labor regulations, skill requirements, employee preferences) and optimizing objectives (e.g. minimizing costs, maximizing service level) [9] [10].

Applying these concepts to the specific context of airport ground handling introduces distinct environmental, time, and spatial challenges. The environment of an airport is characterized by high dynamism, where flight schedules frequently change due to delays, cancellation, or gate reassignments, requiring responsive adjustments to staff assignments. Operations are extremely time-sensitive, as ground handling tasks must be completed within strict time windows dictated by the aircraft turnaround process to ensure punctuality. Resource constraints are common, with limited availability of qualified staff and specialized equipment requiring careful and efficient allocation. Spatial constraints also play a significant role, as staff often need to traverse between gates or other work locations within the airport. Lastly, tasks often have dependencies and teaming requirements, where specific skills are essential, or multiple staff members must collaborate to complete a task.

Research in the workforce planning spans different planning horizons. We distinguish the following different planning stages as mentioned by Claussen [11].

- **Strategical planning:** This is the planning the furthest away from the day operations. In this stage long term decisions regarding workforce size and skill mix are made.
- **Tactical planning:** Medium-term decisions, primarily focusing on shift design and personnel rostering to meet anticipated demand over weeks or months. This area has received considerable attention for both flight crew and ground staff [12], [13], [11].
- **Operational planning:** This is the stage where we will be focusing on in this thesis. This stage concerns the assignment of specific tasks to the available staff.

- Real-time planning: This is the final phase, reacting to events during the day of operation.

Cappanera et al. [14] presented a method to combine the task scheduling and personnel rostering for airport ground staff. Using mathematical programming and heuristics, weekly rosters for check-in services are generated, highlighting the complexities of integrating the different decision levels.

4.1.2. Operational staff allocation

To address the specific operational task assignment for ground staff, several studies employ Integer Linear Programming (ILP) to model the problem structure accurately. Due to the complexity this way of modeling is combined with advanced solution techniques. Tian et al. [15] formulates the assignment of ground service tasks with specific qualification and time windows as an ILP. By applying Branch-and-Price, based on Dantzig-Wolfe decomposition, the goal is to achieve optimal solutions by generating feasible schedules for staff members. This approach is powerful for capturing complex constraints and finding optimal solutions, but it can be computationally demanding. Similarly, the work by Dohn et al. [16], on the Manpower Allocation Problem with Time Windows and Job Teaming Constraints (MAPTWTC) also utilizes an ILP formulation solved via Branch-and-Price.

The Branch-and-Price method, as applied by Dohn and colleagues, is designed to handle ILPs where the number of variables (representing feasible team work schedules) is too large to enumerate explicitly [16]. It works by decomposing the problem into two parts. A Restricted Master Problem optimizes the selection from a limited set of currently known schedules. Simultaneously, a Pricing Subproblem uses dual information from the master problem to search for new, potentially better schedules from the entire space of possibilities. This is often done by solving a Resource-Constrained Shortest Path Problem. This iterative process, solving the master problem, solving the pricing subproblem, and adding promising columns to the master problem, continues until no further improving columns are found. Since an integer solution is required, the column generation procedure is embedded within a branch-and-bound framework, where branching on fractional variables ensures an optimal integer assignment of teams to schedules [16].

The contribution of Dohn et al. is the explicit modeling of job-teaming constraints with the branch and price framework, which distinguishes it from simpler assignment problems. Both studies by Tian et al. and Dohn et al., focus on maximizing an objective related to task completion. The former study focuses on maximizing the total benefit and the latter on maximizing the number of completed tasks. With this they implicitly assume that resource might be insufficient to cover all tasks.

In contrast to the exact methods, other research prioritizes computational speed, often using heuristics. While Anreatta et al. [17] start with an ILP formulation, a fast constructive heuristic approach is proposed for a combined staff and ground service equipment allocation problem. The iterative approach assigns tasks based on start times, making quick decision that rapidly finds acceptable solutions, without the guarantee of finding an optimal solution. This highlights a common trade-off: the guarantee of optimality of exact methods [16], [15] versus speed of heuristics [17].

Furthermore, the objective is different in the discussed researches. While Dohn et al. and Tian et al. both assume that there might need to be unplanned tasks, Andreatta et al. assumes constraints to plan all the tasks. The manpower allocation problem, discussed by Dohn et al., is also discussed in the studies by Lim et al. [18] and Li et al [19]. Here the objective focuses on minimizing the total number of workers required to cover a given set of tasks. They employ metaheuristic approaches, tabu search embedded simulated annealing [18] and construction heuristics embedded simulated annealing [19], suited for this minimization goal.

4.2. The vehicle routing problem

The staff allocation problem studied in this thesis shares significant structural similarities with the well-studied Vehicle Routing Problem (VRP), and especially with the VRPTW. The VRP is a classical combinatorial optimization problem introduced by Dantzig and Ramser [20], where the goal is to determine optimal routes for a fleet of vehicles to serve a set of customers, starting and ending at a central depot,

while minimizing total travel cost or distance.

In the basic VRP, a fleet of vehicles must serve customers, each with a certain demand, while respecting vehicle capacities. The VRP can be represented as a graph $G = (V, A)$, where V denotes the set of nodes (customers and depot) and A the set of arcs connecting them, each associated with a cost c_{ij} . A schematic overview of a simple VRP instance is shown in Figure 4.1.

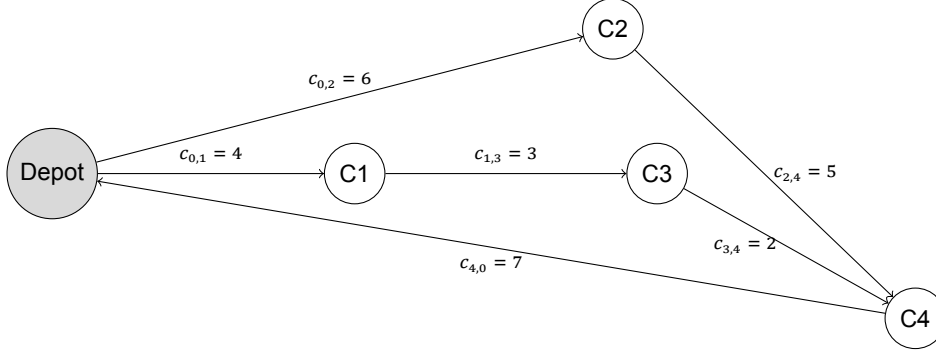


Figure 4.1: Schematic representation of a VRP instance as a graph $G = (V, A)$, with one depot and four customers. Each arc (i, j) is associated with a travels cost c_{ij} .

A common way to model the basic VRP mathematically is through a vehicle flow formulation [21]. Let V be the set of nodes (the customers and the depot 0) and A the set of arcs connecting them. The model is formulated as follows:

$$\min \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \quad (4.1)$$

$$\text{s.t.} \quad \sum_{i \in V} x_{ij} = 1, \quad \forall j \in V \setminus \{0\} \quad (4.2)$$

$$\sum_{j \in V} x_{ij} = 1, \quad \forall i \in V \setminus \{0\} \quad (4.3)$$

$$\sum_{i \in V \setminus \{0\}} x_{i0} = K, \quad (4.4)$$

$$\sum_{j \in V \setminus \{0\}} x_{0j} = K, \quad (4.5)$$

$$\sum_{i \notin S} \sum_{j \in S} x_{ij} \geq r(S), \quad \forall S \subseteq V \setminus \{0\}, S \neq \emptyset \quad (4.6)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in V \quad (4.7)$$

In this formulation x_{ij} is a binary variable equal to 1 if a vehicle travels directly from node i to node j , c_{ij} denotes the cost from node i to node j , K is the number of vehicles and $r(S)$ is the minimum number of vehicles needed to serve the set S .

Constraints 4.2 and 4.3 ensure that each customer is visited exactly once. Constraint 4.4 and 4.5 are the depot flow constraints which ensure that the number of vehicles leaving the depot is equal to the number of vehicles returning. Constraints 4.6 are the capacity cuts constraints. These constraints ensure that for any subset S , there are enough arcs entering S from the outside to guarantee that the customers in S can be reached and served. In particular, the number of such arcs must be at least $r(S)$, which represents the minimum number of vehicles required to serve the customers in S . This prevents any group of customers from being disconnected from the depot or from receiving fewer vehicles than

necessary. Lastly constraints 4.7 ensure all arc decisions are binary.

From this model the model can be extended to include the time windows. Additional constraints are imposed in the form of time windows specifying when service at each customer must start and finish [21]. An overview of heuristic approaches for the vehicle routing problem and the vehicle routing problem with time windows can be found in [22].

4.2.1. Mapping to staff allocation

The staff allocation problem in this thesis can be compared to the VRPTW in the following ways:

- Vehicles can represent the staff members or teams.
- Customers can represent the tasks.
- The depot can be seen as the start location of the shift, e.g. the break room.
- Travel time corresponds to the time it takes a staff member to go from one aircraft to another aircraft.
- The time windows map to the required start and end time of the tasks.
- Vehicle capacity could potentially map to limitations like the number of tasks a staff member is allowed to perform (for example because of workload).
- Unlike the classical VRP, where the objective is to minimize total travel cost, the main objective in the staff allocation problem in this thesis, is to maximize the number of planned tasks.

Additionally, specific constraints for KLM's staff allocation are like those in specialized versions of the VRPTW. An important constraint in our problem is the limited number of available staff. This mirrors the VRPTW with a limited number of vehicles, sometimes referred to as m-VRPTW. Lau et al. formulated this problem and proposed a Tabu search heuristic approach to solve the problem [23]. Other heuristic approaches for the m-VRPTW have also been explored [24] [25]. Exact methods are also applicable to these types of problems. Boussier et al. developed a branch-and-price algorithm specifically for a VRP variant considering both a limited number of vehicles and teaming constraints [26].

This teaming requirement, when certain tasks must be performed simultaneously by multiple staff members, is another important aspect of the problem studied in this thesis. In the VRP literature, this is often addressed as the VRP with synchronization constraints or VRP with teaming. As mentioned, the work by Boussier et al. is relevant here as well [26]. Problems involving time windows, synchronization, and multiple objectives have also been tackled by other researchers using various approaches, including MIP-based heuristics [27], Simulated Annealing [28], and specialized genetic algorithms like Non-Dominates Sorting Genetic Algorithms [29], and Memetic Algorithms [30].

A distinctive element of the staff allocation problem studied in this thesis is the presence of different modes for each flight. Unlike in standard VRPTW formulations, where the task structure is fixed, here each flight can be executed in alternative configurations. The selected mode determines the team size, the division of tasks across the aircraft holds, and the task durations. As a result, the task structure itself becomes a decision variable, adding a level of complexity not typically present in classical VRPTW or workforce scheduling problems.

4.3. Positioning of this research

The planning problem studied in this thesis is closely related to well-know optimization problems in the literature, in particular workforce scheduling and the Vehicle Routing Problem with Time Windows (VRPTW). Similar to workforce scheduling, the problem involves assigning a limited number of qualified staff to tasks under time and resource constraints. At the same time, the structure of the problem mirrors the VRPTW, where routes correspond to task sequences for staff members, and additional constraint arise from travel times, strict task windows, and synchronization requirements.

However, most existing studies look at these problems separately. Workforce scheduling research often focuses on longer-term planning and does not usually include detailed aspects such as travel times or teaming. On the other hand, VRP-based models typically assume fixed service requirements for each customer, without considering regulations, breaks, or the need for multiple staff members. What makes this thesis different is that it brings these elements together in one operational staff assignment problem.

Specifically, the problem incorporates:

- Strict task time windows dictated by the flight schedule.
- Travel times for the resources that need to travel between the geographically distributed location.
- Teaming constraints, where tasks must be executed by multiple staff members simultaneously.
- Break requirements for individual shifts.
- Limited staff availability, leading to potential under coverage.
- Multiple modes per flight, where the selected mode determines the number of staff needed and timing of tasks.

The last element, the inclusion of modes, represents a particularly distinctive contribution. Unlike most workforce scheduling or VRP formulations, where staffing requirements per task are fixed in advance, this thesis treats the team size and task structure as decision variables. To the best of our knowledge, very limited literature exists that integrates mode selection into operational workforce assignment problems.

Moreover, a lot of literature of workforce allocation or VRP focuses on generating feasible or cost-efficient schedules under nominal conditions. The contribution of this thesis also lies in analyzing how schedules behave under operational disruptions. By analyzing what-if scenarios such as flight delays or staff shortages, the model provides insights into their effect on task coverage. In addition, it explores actions that can improve coverage, for example by allowing flight delays within the model to create additional scheduling opportunities. This highlights weak points in the planning process and explores possible strategies to address them.

In this way, the research bridges the gap between classical optimization formulations and the practical needs of airline ground handling operations, where operational feasibility and the ability to handle disruptions is equally important.

5

Mathematical Model

This chapter presents the mathematical model designed to solve the problem described in the previous chapters. The goal of the model is to assign as many tasks as possible to the available resources, while respecting the relevant constraints.

The formulation is based on a variant of the Vehicle Routing Problem with Time Windows (VRPTW), which provides greater flexibility than the slot allocation model currently used in practice. The chapter first presents the base model, detailing its input parameters, decision variables, objective function, and constraints. It then introduces two extensions designed to improve task coverage: allowing cross-zone assignments and enabling task postponements. Finally, the chapter describes how disruption scenarios, such as flight delays and staff shortages, are incorporated to assess the model's performance under different conditions.

5.1. Choice of model

As described in Chapter 3, a slot allocation model is currently used in GroundOpt to support the planning on the day of operation. Initially for this thesis, an attempt was made to adopt this existing model. However, this approach was ultimately set aside to focus on a different formulation based on a version of the Vehicle Routing Problem with Time Windows (VRPTW), following Dohn [16]. In this formulation, resources are viewed as vehicles, and tasks of the flights are seen as customers that need to be visited. The ordered sequence of tasks assigned to each resource forms its route, and the objective is to assign as many tasks as possible to the available resources.

The primary motivation to use a different formulation is the intended extension of the model to allow for postponement in task execution. In the activity slot formulation, the set of feasible task assignments is determined in advance by specifying, for each shift and task combination, which tasks can be scheduled immediately before or after. These relationships are based on fixed timing assumptions. When we would allow for postponements in this model, many of these predefined combinations may no longer be valid, which adds complexity to the planning process. The VRPTW-based formulation, on the other hand, avoids this issue by not relying on predefined feasibly consecutive tasks, offering greater flexibility.

5.2. Base Model

To introduce the proposed modeling approach, we first present a version of the model that does not include any actions to see how many tasks we are able to plan in a perfect world scenario. This base model focuses on planning within a single area, assuming that all task start times and durations are known in advance. It serves as a foundation for later extensions, such as incorporating postponements or cross-zone interactions. In the following sections, we describe the input structure, the sets and decision variables, the objective function, and the constraints of the developed optimization model.

5.2.1. Model input

The model requires the following elements as input to accurately represent the planning problem in a specific area:

Planning horizon

- $[t_{min}, t_{max}]$: Time window of the problem.

Flights and tasks

- F : Set of flights.
- M_f : Set of feasible execution modes for each flight $f \in F$.
- $T_{f,m}$: Tasks associated with flight f and mode m .
- $H_{f,m}$: Holds (two tasks) corresponding to flight f and mode m .
- $\mathcal{O}_F \subseteq F \times F$: Pair of flights $(f1, f2)$ from which the corresponding tasks are allowed to overlap in time(e.g., turnaround flights).
- T : the set of all tasks with for each task $t \in T$:
 - a_t : start time of task t .
 - b_t : end time of task t .
 - f_t : associated flight.
 - h_t : associated hold.
 - loc_t : location of task t .
- H : Set of all holds, with for each $h \in H$:
 - a_h : start time (earliest of its tasks).
 - b_h : end time (latest of its tasks).
 - f_h : associated flight.
 - loc_h : location of hold h .

Resources

- S : Set of individual shifts. For each shift $s \in S$:
 - e_s : Shift start time.
 - g_s : Shift end time.
- P : Set of resource pairs (teams). For each $p \in P$:
 - e_p : Team start time.
 - g_p : Team end time.

For each resource $r \in S \cup P$:

- $last_r$: The last task or hold assigned to r before the planning window (if there is such a task or hold).

Break requirements

- $SB \subseteq S$: Individual resources that require a break in the planning horizon.
- $PB \subseteq P$: Teams that require a break in the planning horizon.
- $[k_s, l_s]$: window in which the break for shift $s \in SB$ can start.
- $[k_p, l_p]$: window in which the break for pair $p \in PB$ can start.

Travel times

- $tt_{i,j}$: Time required to travel between the two locations i and j , defined for any pair of task or break location.

5.2.2. Sets and variables

Sets

To decrease the complexity of the model, multiple precomputed sets are created which are listed in Table 5.1.

Table 5.1: Overview of the created sets to decrease the complexity of the model.

Symbol	Description
$T_{f,m} \subseteq T$	Tasks for flight f in mode m
$H_{f,m} \subseteq H$	Holds for flight f in mode m
$T_{f,m,h} \subseteq T$	Tasks for flight f in mode m , that belong to hold h
$T_s \subseteq T$	Tasks that can be performed by shift $s \in S$, based on qualifications and availability.
$H_p \subseteq H$	Holds that can be performed by team $p \in P$, based on qualifications and timing.
$S_t \subseteq S$	Shifts that can perform task $t \in T$
$P_h \subseteq P$	Pairs that can perform hold $h \in H$
$O_T \subseteq T \times T$	Set of task pairs (t_1, t_2) that may overlap (e.g., turnaround flights) but are allowed to be performed by the same resource
$O_H \subseteq H \times H$	Set of task pairs (h_1, h_2) that may overlap (e.g., turnaround flights) but are allowed to be performed by the same resource

Virtual tasks

To model the planning problem as a routing problem, each resource's task sequence is structured using artificial tasks. These include a start task, an end task, and, if applicable, a break task. They enable the formulation of complete routes for each resource within the optimization model.

Start and End Tasks

- For each resource $r \in S \cup P$, we define:
 - **Start task** start_task_r :
 - ◊ If the resource was assigned a task prior to the planning window, the start location and time of start_task_r correspond to the location and end time of that last task.
 - ◊ Otherwise, the task is located at the depot (e.g., break room) and starts at e_r , the shift start time.
 - ◊ The end time of the start task is equal to its start time.
 - **End task** end_task_r :
 - ◊ This task is always located at the depot, with both its start and end times equal to g_r , the shift end time.

Break Tasks

- For each resource requiring a break ($s \in SB$ or $p \in PB$), a dedicated break task is added:
 - b_s and b_p : tasks located at the depot (break room), with variable start and end times.
 - The start time of the break must lie within the allowed window $[k_s, l_s]$ or $[k_p, l_p]$.
 - The duration of the break is fixed and defined by $\text{break_duration}_{b_s}$ or $\text{break_duration}_{b_p}$.

Variables

- $x_{ijs} \in \{0, 1\}$ 1 if shift s performs task j after it has performed tasks i .
- $z_{ijp} \in \{0, 1\}$ 1 if pair p performs hold j after it has performed hold i .
- $y_{fm} \in \{0, 1\}$ 1 if flight f is performed in mode m .
- $bt_s \in [k_s, l_s]$ this variable determines the start time of the break of shift $s \in SB$.
- $bt_p \in [k_p, l_p]$ this variable determines the start time of the break of shift $p \in PB$.
- $unplanned_tasks_f \in \mathbb{Z}_{\geq 0}$ for each flight $f \in F$ we have a variable to determine the number of tasks of the flight which are not assigned to a shift or a pair.

5.2.3. Objective function

The objective function of the model is as follows, we want to minimize the number of unplanned tasks:

$$\min \sum_{f \in F} unplanned_tasks_f \quad (5.1)$$

5.2.4. Constraints

Start and end constraints

The route of each resource must start at the created start task.

$$\sum_{j \in end_task_s \cup T_s} x_{start_task_s j s} + x_{start_task_s b_s s} = 1 \quad \forall s \in S \quad (5.2)$$

$$\sum_{j \in end_task_p \cup H_p} z_{start_task_p j p} + z_{start_task_p b_p p} = 1 \quad \forall p \in P \quad (5.3)$$

Break constraints

For each resource that needs a break, this break must be visited in its route.

$$\sum_{t \in T_s \cup start_task_s} x_{t b_s s} = 1 \quad \forall s \in SB \quad (5.4)$$

$$\sum_{h \in H_p \cup start_task_p} z_{h b_p p} = 1 \quad \forall p \in PB \quad (5.5)$$

Flow conservation

If a task is visited in a resource's route, it must also be left.

$$\sum_{i \in start_task_s \cup T_s \cup b_s} x_{i t s} - \sum_{j \in end_task_s \cup T_s \cup b_s} x_{t j s} = 0 \quad \forall s \in S, \forall t \in T_s \cup b_s \quad (5.6)$$

$$\sum_{i \in start_task_p \cup H_p \cup b_p} z_{i h p} - \sum_{j \in end_task_p \cup H_p \cup b_p} z_{h j p} = 0 \quad \forall p \in P, \forall h \in H_p \cup b_p \quad (5.7)$$

Time constraints

The first task a resource visits must start later than the start time of the resource.

$$e_s + tt_{\text{start_task}_s j} - (e_s + tt_{\text{start_task}_s j} - a_j)(1 - x_{\text{start_task}_s j s}) \leq a_j \quad \forall s \in S, \forall j \in T_s \quad (5.8)$$

$$e_p + tt_{\text{start_task}_p j} - (e_p + tt_{\text{start_task}_p j} - a_j)(1 - z_{\text{start_task}_p j p}) \leq a_j \quad \forall p \in P, \forall j \in H_p \quad (5.9)$$

Similarly the last task a resource visits must end no later than the end time of the resource.

$$b_i + tt_{\text{end_task}_s} - (b_i + tt_{\text{end_task}_s} - g_s)(1 - x_{\text{end_task}_s i}) \leq g_s \quad \forall s \in S, \forall i \in T_s \quad (5.10)$$

$$b_i + tt_{\text{end_task}_p} - (b_i + tt_{\text{end_task}_p} - g_p)(1 - z_{\text{end_task}_p i}) \leq g_p \quad \forall p \in P, \forall i \in H_p \quad (5.11)$$

Furthermore, if two tasks are visited by a resource consecutively, the end time of the first task plus the travel time from one task to the other must be earlier than the start time of this second task.

$$b_i + tt_{ij} - (b_i + tt_{ij} - a_j)(1 - x_{ijs}) \leq a_j \quad \forall s \in S, \forall (i, j) \in (T_s \times T_s) \setminus \mathcal{O}_T \quad (5.12)$$

$$b_i + tt_{ij} - (b_i + tt_{ij} - a_j)(1 - z_{ijp}) \leq a_j \quad \forall p \in P, \forall (i, j) \in (H_p \times H_p) \setminus \mathcal{O}_H \quad (5.13)$$

Break timing

The start time of the break for a shift must be greater or equal then the time of the task that is visited before the break and the travel time between the task and the break room.

$$b_i + tt_{ib_s} - (b_i + tt_{ib_s} - bt_s)(1 - x_{ib_s s}) \leq bt_s \quad \forall s \in SB, \forall i \in T_s \cup \text{start_task}_s \quad (5.14)$$

$$b_i + tt_{ib_p} - (b_i + tt_{ib_p} - bt_p)(1 - z_{ib_p p}) \leq bt_p \quad \forall p \in PB, \forall i \in H_p \cup \text{start_task}_p \quad (5.15)$$

$$bt_s + \text{break_duration}_s + tt_{b_s j} - (bt_s + \text{break_duration}_s + tt_{b_s j} - a_j)(1 - x_{jb_s s}) \leq a_j \quad \forall s \in SB, \forall j \in T_s \cup \text{end_task}_s \quad (5.16)$$

$$bt_p + \text{break_duration}_p + tt_{b_p j} - (bt_p + \text{break_duration}_p + tt_{b_p j} - a_j)(1 - z_{jb_p p}) \leq a_j \quad \forall p \in PB, \forall j \in H_p \cup \text{end_task}_p \quad (5.17)$$

Finally the start time of the break must be within the allowed time window.

$$k_s \leq bt_s \leq l_s \quad \forall s \in SB \quad (5.18)$$

$$k_p \leq bt_p \leq l_p \quad \forall p \in PB \quad (5.19)$$

Mode and task assigned

The number of unplanned tasks needs to be determined and at exactly one mode must be chosen for each flight.

$$\sum_{t \in T_{f,m}} \left(y_{fm} - \sum_{\substack{s \in S_t, \\ i \in T_s \cup \text{start_task}_s \cup b_s}} x_{its} - \sum_{\substack{p \in P_{h_t}, \\ j \in H_p \cup \text{start_task}_p \cup b_p}} z_{jhp} \right) \leq \text{unplanned_tasks}_f \quad \forall f \in F, \forall m \in M_f \quad (5.20)$$

$$\sum_{m \in M_f} y_{fm} = 1 \quad \forall f \in F \quad (5.21)$$

If a mode is chosen only those tasks can be planned.

$$\sum_{t \in T_{f,m,h}} \left(\sum_{\substack{s \in S_t, \\ i \in T_s \cup \text{start_task}_s \cup b_s}} x_{its} + 2 \cdot \sum_{\substack{p \in P_{h_t}, \\ j \in H_p \cup \text{start_task}_p \cup b_p}} z_{jhp} \right) \leq |T_{f,m,h}| y_{fm} \quad \forall h \in H_{f,m} \quad (5.22)$$

Unique assignment

Each task can be planned at most once

$$\sum_{\substack{s \in S_t, \\ j \in \text{start_task}_s \cup T_s \cup b_s}} x_{its} + \sum_{\substack{p \in P_{h_t}, \\ j \in \text{start_task}_p \cup H_p \cup b_p}} z_{jhp} \leq 1 \quad \forall t \in T \quad (5.23)$$

$$x_{ijs} \in \{0, 1\} \quad \forall s \in S, \forall (i, j) \in (\text{start_task}_s \cup T_s \cup b_s) \times (\text{end_task}_s \cup T_s \cup b_s) \quad (5.24)$$

$$z_{ijp} \in \{0, 1\} \quad \forall p \in P, \forall (i, j) \in (\text{start_task}_p \cup H_p \cup b_p) \times (\text{end_task}_p \cup H_p \cup b_p), \quad (5.25)$$

$$y_{fm} \in \{0, 1\} \quad \forall f \in F, \forall m \in M_f \quad (5.26)$$

5.3. Model Extensions for increased task coverage

The solution obtained from the base model may include tasks that cannot be scheduled within the given constraints. This thesis explores strategies to increase the number of scheduled tasks. Two approaches are proposed to achieve this: cross-zone assignment, which allows resources to be assigned to tasks from different areas, and postponing tasks, which investigates whether shifting the timing of the tasks can create additional planning opportunities.

5.3.1. Cross-Zone Assignment

One way to make the schedule more flexible, and potentially fit more tasks, is to allow shifts or teams to handle tasks from other areas. From a modeling perspective, the formulation remains almost identical to the base model: the same decision variables and constraints apply. The only difference is that instead of considering a single area, the model is extended to cover multiple areas simultaneously. As a result, the number of tasks and feasible assignments increases, which enlarges the model but does not change its structure.

Because the model becomes larger, it can be harder to find good solutions. To guide the optimizer, we add a penalty for cross-area assignments in the objective function, so that they are only chosen when necessary to improve the number of tasks planned. To achieve this, the following extensions to the model are introduced.

Additional sets

Two new subsets are introduced that explicitly capture cross-area assignments. These are shown in Table 5.2.

Table 5.2: Overview of the two sets added for cross-area model.

Symbol	Description
$T_s^{\text{diff}} \subseteq T_s$	Tasks that can be performed by shift $s \in S$ but are located in a different area than the one to which s is assigned
$H_p^{\text{diff}} \subseteq H_p$	Holds that can be performed by team $p \in P$ but are located in a different area than the one to which p is assigned

New variable

To explicitly track the number of cross area assignment the following integer variable is introduced:

$$\text{number_switched_area} \in \mathbb{Z}_{\geq 0}$$

This variable will be used in the objective function to minimize cross-zone assignments.

Constraint for Cross-Zone Assignment

To ensure that the variable accurately reflects the number of cross-area assignments made in the schedule, the following constraint is added to the model:

$$\begin{aligned}
 & \sum_{\substack{s \in S, \\ i \in T_s, \\ j \in T_s^{\text{diff}}}} x_{ijs} + \sum_{\substack{s \in S, \\ j \in T_s^{\text{diff}}}} x_{\text{start_task}_s js} + \sum_{\substack{s \in S, \\ j \in T_s^{\text{diff}}}} x_{b_s js} \\
 & + \sum_{\substack{p \in P, \\ i \in H_p, \\ j \in H_p^{\text{diff}}}} z_{ijp} + \sum_{\substack{p \in P, \\ j \in H_p^{\text{diff}}}} z_{\text{start_task}_p jt} + \sum_{\substack{p \in P, \\ j \in H_p^{\text{diff}}}} z_{b_p jt} \leq \text{number_switched_area}
 \end{aligned} \tag{5.27}$$

Adjustment to the objective Function

Lastly also the objective has to be changed, because we want to penalize if shifts are assigned to tasks from a different area. The objective function is as follows:

$$\begin{aligned}
 & \min 10 \sum_{f \in F} \text{unplanned_tasks}_f \\
 & \quad + \text{number_switched_area}
 \end{aligned} \tag{5.28}$$

The first term ensures that as many tasks as possible are planned, while the second introduces a small penalty for assigning resources to tasks outside their own area. This encourages within-area assignments but still allows cross-zone planning when it increases overall task coverage.

5.3.2. Postpone Model

Another way to increase the number of scheduled tasks is by allowing certain flights to be postponed, the corresponding tasks are shifted to a time window in which they can be assigned. To make this possible, several modifications to the base model were required, which will be discussed in this section. For this extended model we can solve each area separately. The modifications made for this postponement are described below.

New variables for the postpone model

To model postponements explicitly, two variables were added:

- $\text{delayed_minutes}_f \in [0, 60]$: for each flight $f \in F$, this variable captures the number of minutes the flight is postponed.
- $\text{delayed_flight}_f \in \{0, 1\}$: a binary variable that is 1 if flight f is postponed, and 0 otherwise.

Updated objective function

The objective function was then updated to reflect the trade-off between scheduling more tasks and the cost of introducing postponements:

$$\begin{aligned}
 \min & 100000 \sum_{f \in F} \text{unplanned_tasks}_f \\
 & + 1000 \sum_{f \in F} \text{delayed_flight}_f \\
 & + \sum_{f \in F} \text{delayed_minutes}_f
 \end{aligned} \tag{5.29}$$

The objective balances three priorities: maximizing task coverage, minimizing the number of postponed flights, and limiting the total delay time. The weights are chosen to ensure that scheduling tasks remains the primary goal, while postponements are only applied when they meaningfully improve task feasibility. The penalty on delayed minutes ensures that, if a flight is postponed, the delay remains as short as possible, preventing unnecessary time shifts

Constraint adjustments

To ensure feasibility, several existing constraints had to be updated. In particular, the delayed_minutes_f variable was added to all constraints involving the start and end times of tasks. This resulted in adjustments to constraints 5.8 through 5.17.

$$\begin{aligned}
 e_s + \text{tt}_{\text{start_task}_s j} - (e_s + \text{tt}_{\text{start_task}_s j} - a_j - \text{delayed_minutes}_{f_j})(1 - x_{\text{start_task}_s j s}) \\
 \leq a_j + \text{delayed_minutes}_{f_j} \quad \forall s \in S, \forall j \in T_s
 \end{aligned} \tag{5.30}$$

$$\begin{aligned}
 e_p + \text{tt}_{\text{start_task}_p j} - (e_p + \text{tt}_{\text{start_task}_p j} - a_j - \text{delayed_minutes}_{f_j})(1 - z_{\text{start_task}_p j p}) \\
 \leq a_j + \text{delayed_minutes}_{f_j} \quad \forall p \in P, \forall j \in H_p
 \end{aligned} \tag{5.31}$$

$$\begin{aligned}
 b_i + \text{delayed_minutes}_{f_i} + \text{tt}_{\text{end_task}_s} \\
 - (b_i + \text{delayed_minutes}_{f_i} + \text{tt}_{\text{end_task}_s} - g_s)(1 - x_{\text{end_task}_s s}) \\
 \leq g_s \quad \forall s \in S, \forall i \in T_s
 \end{aligned} \tag{5.32}$$

$$\begin{aligned}
 b_i + \text{delayed_minutes}_{f_i} + \text{tt}_{\text{end_task}_p} \\
 - (b_i + \text{delayed_minutes}_{f_i} + \text{tt}_{\text{end_task}_p} - g_p)(1 - z_{\text{end_task}_p p}) \\
 \leq g_p \quad \forall p \in P, \forall i \in H_p
 \end{aligned} \tag{5.33}$$

$$\begin{aligned}
 b_i + \text{delayed_minutes}_{f_i} + \text{tt}_{ij} \\
 - (b_i + \text{delayed_minutes}_{f_i} + \text{tt}_{ij} - (a_j + \text{delayed_minutes}_{f_j}))(1 - x_{ij s}) \\
 \leq a_j + \text{delayed_minutes}_{f_j} \quad \forall s \in S, \forall (i, j) \in (T_s \times T_s) \setminus \mathcal{O}_T
 \end{aligned} \tag{5.34}$$

$$\begin{aligned}
 b_i + \text{delayed_minutes}_{f_i} + \text{tt}_{ij} \\
 - (b_i + \text{delayed_minutes}_{f_i} + \text{tt}_{ij} - (a_j + \text{delayed_minutes}_{f_j}))(1 - z_{ij p}) \\
 \leq a_j + \text{delayed_minutes}_{f_j} \quad \forall p \in P, \forall (i, j) \in (H_p \times H_p) \setminus \mathcal{O}_H
 \end{aligned} \tag{5.35}$$

$$\begin{aligned}
& b_i + \text{delayed_minutes}_{f_i} + \text{tt}_{ib_s} - \\
& (b_i + \text{delayed_minutes}_{f_i} + \text{tt}_{ib_s} - \text{bt}_s)(1 - x_{ib_s s}) \\
& \leq \text{bt}_s \quad \forall s \in SB, \forall i \in T_s \cup \text{start_task}_s
\end{aligned} \tag{5.36}$$

$$\begin{aligned}
& b_i + \text{delayed_minutes}_{f_i} + \text{tt}_{ib_s} - \\
& (b_i + \text{delayed_minutes}_{f_i} + \text{tt}_{ib_s} - \text{bt}_p)(1 - z_{ib_s p}) \\
& \leq \text{bt}_p \quad \forall p \in PB, \forall i \in H_p \cup \text{start_task}_p
\end{aligned} \tag{5.37}$$

$$\begin{aligned}
& \text{bt}_s + \text{break_duration}_s + \text{tt}_{b_s j} \\
& (\text{bt}_s + \text{break_duration}_s + \text{tt}_{b_s j} - a_j + \text{delayed_minutes}_{f_j})(1 - x_{jb_s s}) \\
& \leq a_j + \text{delayed_minutes}_{f_j} \quad \forall s \in SB, \forall j \in T_s \cup \text{end_task}_s
\end{aligned} \tag{5.38}$$

$$\begin{aligned}
& \text{bt}_p + \text{break_duration}_p + \text{tt}_{b_p j} \\
& - (\text{bt}_p + \text{break_duration}_p + \text{tt}_{b_p j} - a_j + \text{delayed_minutes}_{f_j})(1 - z_{jb_p p}) \\
& \leq a_j + \text{delayed_minutes}_{f_j} \quad \forall p \in PB, \forall j \in H_p \cup \text{end_task}_p
\end{aligned} \tag{5.39}$$

In addition, several new constraints were introduced to ensure the model behaves as intended under the postponed flight scenario.

The first constraint enforces a logical relationship between the two delay variables: if the number of postponed minutes for a flight is greater than zero, the corresponding binary variable indicating whether the flight is postponed must also be set to one. This is captured by the following constraint:

$$\text{delayed_minutes}_f \leq 60 \cdot \text{delayed_flight}_f \quad \forall f \in F \tag{5.40}$$

Moreover, special care must be taken in the case of turnaround flights, i.e. when two consecutive flights share the same aircraft, such that one is an arrival and the next a departure. If the first flight in a turnaround pair is postponed, the second flight may also need to be postponed. To handle this dependency, the remaining buffer time between the end of the first flight and the start of the second is precomputed. This value, denoted by $\text{rest_time}_{(f1, f2)}$, represents the number of minutes that flight $f1$ can be postponed without requiring a delay for flight $f2$.

This leads to the following constraint:

$$\text{delayed_minutes}_{f2} \geq \text{delayed_minutes}_{f1} - \text{rest_time}_{(f1, f2)} \quad \forall (f1, f2) \in \mathcal{O}_F \tag{5.41}$$

This ensures that if flight $f1$ is postponed beyond the available buffer time, the postponement is propagated to flight $f2$ as well.

5.4. Modeling disruption scenarios

An important objective of this thesis is to analyze how disruptions effect the outcomes of the model. To enable this analysis disruption scenarios need to be incorporated into the model. This is done in the stage where the instance is generated, before the sets, variables and constraints of the model, described earlier in this chapter, are constructed. Two types of disruptions are considered in this thesis: flight delays and staff shortages. These are chosen because they are impactful sources of irregularity in daily baggage handling operations.

Flight delays are modeled by modifying the scheduled start times of tasks associated with delayed flights. For each instance a given percentage of flights within the planning horizon are randomly selected to experience a delay. The length of the delay is drawn from a uniform distribution ranging from

one to sixty minutes. Once a flight was chosen to get delayed, all tasks associated with this flight are updated accordingly: their start and end times are shifted by the same delay amount. This ensures that the disruption is consistently propagated to all tasks related to the delayed flight, thereby capturing the operational consequences in the flight schedule. It should be noted that, for this thesis, delays are applied to individual flights, rather than to connected turnaround pairs. Consequently, a delay in an arriving flight does not automatically affect its corresponding departure. This simplification allows us to study the effects of independent delays and to generate flexible disruption scenarios. Furthermore, all flights with scheduled start times within the considered time window are eligible for delay. As a result, some flights may be delayed to the extent that their tasks are shifted partially or entirely outside the planning horizon. Within the Postpone Model, tasks of delayed flights are also allowed to be postponed further, as the model treats these tasks in the same way as any other task.

Staff shortages are modeled by reducing the set of shifts available for task assignment. A given percentage of shifts in the planning horizon is randomly selected to be unavailable, representing, for example, staff calling in sick or other unforeseen absences. These shifts are removed from the set of available resources before optimization, which means they cannot be assigned any tasks. The removal directly reduces the pool of staff that can be deployed and thus simulates the operational impact of reduced workforce capacity.

By incorporating these adjustments at the instance level, the model can be tested for not only "perfect world" circumstances with no disruptions, but also on scenarios where these disruptions occur. This provides a way to evaluate the effect these disruptions have on the scheduling of the tasks.

Solution Method

The mathematical models described in Chapter 5 results in a large mixed integer programming formulation. Solving this model directly proved computationally demanding. To address this challenge, this chapter discusses the alternative solution approach that was developed. In addition, the chapter explains how both the model and the solution method are integrated into KLM's existing infrastructure, requiring several adaptations to ensure that the approach can be applied in practice.

6.1. Motivation for an iterative approach

To solve the model described in Chapter 5, the Gurobi Optimizer is used [8]. Gurobi is a solver for mathematical optimization problems, offering efficient algorithms for solving large-scale Mixed Integer Programming (MIP) models. It effectively captures the structural features of complex optimization problems and aims to find high-quality solutions within reasonable computation time. Moreover, the operational system currently used at KLM also relies on Gurobi, which makes it practical to use the same solver for testing and comparison within a familiar environment.

As discussed in the literature review (Chapter 4), operational staff allocation problems can be approached using a variety of methods. Heuristic or metaheuristic approaches could provide faster computation, but applying them to this problem would require significant adaptation and extensive validation due to the specific operational constraints. Given the limited time available for this thesis, Gurobi was therefore chosen as the most practical approach, as it already supports the exact modeling of the problem's constraints.

Despite these advantages, the planning problems considered in this thesis remain large and computationally demanding. A KLM baggage planning instance for an 8-hour horizon typically involves around 540 tasks and 120 resources, resulting in more than 2.8 million constraints and 3.8 million variables. In addition, the model contains significant symmetry and a wide range of feasible options, which complicates the branch-and-bound tree and slows convergence toward optimal or even feasible solutions. Solving the complete model in one run using Gurobi often leads to large solving times.

Therefore, an alternative strategy is adopted in this thesis: an iterative approach. As discussed in Chapter 3, the current operational system at KLM (GroundOpt) tackles complexity by dividing the 3 hour planning horizon into small iteration windows. These smaller subproblems are solved separately and then combined to form a complete plan. To better understand the impact of this iterative setup, additional tests were performed where GroundOpt was executed without its iterative structure. One run found a feasible solution after approximately five hours, while two other runs did not find any feasible solution within ten hours before the solver terminated. These results confirm that the iterative setup used by GroundOpt is essential to achieve practical runtimes, and further motivate the adoption of an iterative approach in this thesis. However, this same approach is not suitable for our model because we allow flight delays to improve the number of tasks that can be scheduled. This introduces a temporal dependency between tasks: postponing one task may affect the feasibility of others outside the local

time window. Hence, decomposing the model by time is not feasible.

Instead, we propose a different iterative strategy: the set of resources (shifts and teams) is divided into smaller groups. In each iteration, only a subset of these resources is optimized, while the assignments of the remaining resources are held fixed. This approach is known as a Fix-and-Optimize strategy [31].

6.2. Fix-and-Optimize base model

At first, the Fix-and-Optimize method is applied to the base model. Recall that in this problem two types of resources are considered: individual shifts and teams. A team consists of two staff members who always work together and therefore receive joint task assignments. A shift represents a single staff member working individually. Each person is either part of a fixed team or treated as an individual shift. The iterative solution approach consists of three phases described below. Phase 1-3 correspond directly to the steps summarized in Algorithm 1.

Phase 1: Initialization with fixed skeleton routes

To ensure that a feasible solution exists before optimizing task assignments, all decision variables are initially fixed. Specifically, the upper bounds of all assignment variables x_{ijs} and z_{ijp} are set to zero. The only exceptions are the variables that represent transitions between the artificial start, break, and end nodes. These remain active, allowing each resource (shift or team) to follow a trivial but feasible path:

- For shifts without a required break: start $\rightarrow$ end.
- For shifts or teams with a break: start $\rightarrow$ break $\rightarrow$ end.

This step creates a minimal but feasible solution that provides a starting point for the optimization procedure.

Phase 2: Iterative optimization of teams and shifts

Next, the resources are treated into the two separate categories: teams and individual shifts. The Fix-and-Optimize strategy is applied in two stages: first to the teams, and then to the shifts.

1. Team optimization:

- Teams are divided into disjoint groups of at most 20 teams.
- In each iteration, one group is selected for optimization.
- For the selected teams, the upper and lower bounds of their variables are unfixed (Upper Bound = 1, Lower Bound = 0).
- For all other teams and shifts the assignment variables are fixed to the values from the previous solution.
- The previous solution is used as initial solution.

2. Shift optimization:

Once all team groups have been optimized the same process is applied to shifts.

- Shifts are divided into disjoint groups of at most 8 shifts.
- During each iteration, the selected shifts are optimized while all team assignments are fixed to their latest values.

This strategy of solving the teams and shifts separately ensures that computational load is manageable, while still allowing improvement to the global solution.

Phase 3: Refinement Iterations

After all individual teams and shifts have been optimized once and there are still unplanned tasks, a final refinement phase is performed to further improve the solution. In this phase, three additional round

of optimization are performed. In each round, the set of teams and the set of shifts are randomly partitioned into new groups, and each group is optimized in turn. In the first round, the number of groups is computed based on the number of groups used in Phase 2: half the number of team groups, half the number of shift groups, or two, whichever is largest. In the other two rounds, the number of groups is halved again, but never below two, allowing larger groups of shifts and teams to be optimized together.

In each of these rounds:

- The groups of teams and shifts are reshuffled randomly, and divided into groups. As mentioned, the size of these groups increases in the rounds.
- Within each round, all groups are optimized sequentially: for each group of teams and each group of shifts, the model is solved while the rest of the solution is kept fixed.
- The previous solution is used as a starting point, and the Gurobi solver is given a time limit per iteration.

This approach helps to escape local optima and enables interaction between shifts and teams that were not optimized together earlier. Increasing the group sizes in each round provides the opportunity to discover further improvements.

Algorithm 1: Fix-and-Optimize for the Base Model

Input: Base model M

Output: Optimized task assignment S

Phase 1: Initialize skeleton routes

forall assignment variables x_{ijs}, z_{ijp} **do**

 set upper bound = 0;

 Activate assignment variables (x_{ijs}, z_{ijs}) if $i, j \in \text{start_task}_s \cup \text{end_task}_s \cup \text{break_task}_s$
 $(x, z \in [0, 1])$;

 Create minimal feasible solution S ;

Phase 2: Iterative Optimization of Teams and Shifts

 Divide teams into groups of size ≤ 20 ;

foreach team group G **do**

 Activate variables for G ($z \in [0, 1]$);

 Fix variables for all other teams and shifts;

 Solve (*model M using S as initial solution*);

 UpdateSolution(S);

 Divide shifts into groups of size ≤ 8 ;

foreach shift group H **do**

 Activate variables for H ($x \in [0, 1]$);

 Fix all team variables;

 Solve (*model M using S as initial solution*);

 UpdateSolution(S);

Phase 3: Refinement Iterations

 Initialize number_of_groups = max(#team groups from Phase 2 / 2, #shift groups from Phase 2 / 2, 2);

for round = 1 to 3 **do**

 Randomly reshuffle teams and shifts into new groups according to number_of_groups ;

foreach group **do**

 Activate variables for the group;

 Fix all other variables;

 Solve (*model M using S as initial solution*);

 UpdateSolution(S);

 Update number_of_groups = max(number_of_groups/2, 2)

return S

6.3. Fix-And-Optimize with Cross-Area assignments

The first expansion of our model allows for cross-area assignments, where shifts can perform tasks in different area's then their assigned work area. Since solving all areas simultaneously results in a very large model, a Fix-and-Optimize heuristic has also been developed to manage this complexity. The procedure is summarized in Algorithm 2.

To simplify the problem, we first solve the initial problem by allowing only assignments of shifts to tasks within their own area, using the same Fix-and-Optimize strategy as the base model. To achieve this, it is crucial that all variables representing a shift performing a task in a different area are fixed to zero. This ensures that the initial solution only considers assignments within a shift's primary area, providing a strong starting point.

After this initial solution is found, we introduce the possibility of cross-area assignments. This is done through a new iterative solving strategy. The procedure is as follows:

- All shifts and teams are randomly shuffled.
- They are partitioned into disjoint groups of at most 15 shifts and 15 teams.
- In each iteration, one group is selected for optimization.
- The assignment variables for the selected resources are unfixed, meaning they can be either 0 or 1, while all others are fixed to their value from the previous solution.
- The previous solution is provided as a starting point to the optimizer (Gurobi) and the model is optimized again.

This process is repeated over three rounds, such that every team and shift has been optimized multiple times in different group combinations.

Algorithm 2: Fix-and-Optimize with Cross-Area Assignments

Input: Cross-Area model M

Output: Optimized task assignment S

Phase 1: *Initial solution within primary areas*

```

forall assignment variables  $x_{ijs}, z_{ijp}$  do
  if area of task  $i$  or  $j \neq$  area of shift  $s$  or team  $p$  then
    set upper bound = 0;

```

Apply base-model Fix-and-Optimize procedure from Algorithm 1 restricted to primary-area assignments;

Store initial feasible solution S ;

Phase 2: *Cross-area assignment optimization*

```

for round = 1 to 3 do
  Randomly reshuffle all shifts and teams;
  Partition shifts and teams into disjoint groups of size  $\leq 15$ ;
  foreach group  $G$  do
    Activate variables for group  $G$  ( $x, z \in [0, 1]$ );
    Fix all other variables to their values in  $S$ ;
    Solve (model  $M$  using  $S$  as initial solution);
    UpdateSolution( $S$ );

```

return S

6.4. Fix-and-Optimize with Postponements

In the second implementation, the Fix-and-Optimize method is extended to the model that allows flight to be postponed. Including postponements increases the solution space and introduces additional complexity. Each task can now be scheduled with a certain delay, which results in more variables and

new temporal dependencies between tasks.

To simplify the problem, we first solve the model without allowing postponements using the Fix-and-Optimize strategy described in Section 6.2. This provides a good and feasible initial solution. After this, if there are unplanned tasks, postponements are introduced into the model and a new iterative solving strategy is applied that takes into account the increased flexibility. The full procedure is summarized in Algorithm 3.

In the base model, the variable representing the number of postponed minutes per flight has an upper bound of zero. As a first step in this phase, the upper bound of all $delayed_minutes_f$ variables is increased to 60, which means the model can assign up to 60 minutes of delay per flight.

Next, five postponement optimization rounds are performed. Each round consists of several iterations, where in each iteration a subset of the teams and shifts is optimized. The procedure is as follows:

- All teams and shifts are randomly shuffled.
- They are partitioned into disjoint groups of at most 15 teams and 15 shifts.
- In each iteration, one group of teams and one group of shifts is selected for optimization.
- The assignment variables (x_{ijs} and z_{ijp}) for the selected resources are relaxed (lower bound = 0, upper bound = 1), while all others are fixed to their values from the previous solution.
- The previous solution is provided as a starting point to Gurobi, improving solution speed and quality.
- A time limit of 500 seconds is set for each optimization run, so for each group.

This process is repeated for five full rounds, such that every team and shift has been optimized multiple times in different group combinations. By reshuffling the groups in every round, interaction between tasks assigned to different teams and shifts is encouraged, which helps to escape local optima.

Algorithm 3: Fix-and-Optimize with Postponements

Input: Postponement model M

Output: Optimized task assignment S

Phase 1: *Initial solution without postponements*

```

  Fix all postponement variables  $delayed\_minutes_f = 0$ ;
  Apply base-model Fix-and-Optimize procedure (Algorithm 1);
  Store initial feasible solution  $S$ ;

```

Phase 2: *Allow postponements*

```

  forall postponement variables  $delayed\_minutes_f$  do
    set upper bound = 60;

```

Phase 3: *Iterative postponement optimization*

```

  for round = 1 to 5 do
    Randomly reshuffle all shifts and teams;
    Partition shifts and teams into disjoint groups of size  $\leq 15$ ;
    foreach group  $G$  do
      Activate assignment variables for group  $G$  ( $x, z \in [0, 1]$ );
      Fix all other variables to their values in  $S$ ;
      Solve (model  $M$  using  $S$  as initial solution, time limit = 500 sec);
      UpdateSolution( $S$ );

```

6.5. Infrastructure integration

While the mathematical model described in Chapter 5 captures the assignment problem in a formal way, implementing the model within the existing infrastructure, explained in Section 3.2, required sub-

stantial adjustments. The infrastructure of GroundOpt, illustrated in Figure 3.1, provides the foundation for these adaptations. These adjustments ensure that the new models and solution methods can be applied in practice. In this section we will explain where and how adjustments are made in the existing infrastructure.

6.5.1. Base model

In Chapter 5, the developed base model is introduced, and in Section 6.2 the solution method is explained. To embed this model within the existing infrastructure, several adjustments were required.

First, small modifications were made in the preprocessing step. As described in Section 3.2, the preprocessing stage in GroundOpt determines the iteration structure for the activity slot model. Since the models developed in this thesis do not rely on the same iteration framework, the system was adapted to work with a single optimization run rather than multiple iterations. As a result, the input to the optimization stage also changed slightly. The optimization window, which in the original system was a subinterval of the full horizon, is now the entire planning horizon. Consequently, the sets *tasks* and *tasks in full optimization window* are identical, and several run parameters became redundant, as there was no need to keep track of iteration indices.

The second component of the infrastructure is the optimizer, which required the most extensive changes. The optimizer consists of four stages: data preparation, model construction, model solving, and output processing. In the data preparation stage, additional processing was required to construct the specific input structures of the base model, such as the feasible sets of tasks per shift and holds per team. Many of these sets were already generated in the current method and could therefore be reused with minor adjustments. The model construction stage was entirely reimplemented, since a fundamentally different formulation was developed. At this point the constraints, variables, and objective function of the VRPTW-based model were created. In the model solving stage, the Fix-and-Optimize strategy introduced in Section 6.2 was implemented. Whereas GroundOpt determined the number of optimization runs outside the optimizer part, the iterative procedure in this thesis is embedded directly in the solving component of the optimizer. Finally, in the output processing stage, the results of the solver were reformatted so that they matched the expected input for the subsequent stages. Several adjustments were required to ensure that the new model output could be correctly transformed into the JSON structure used by the system.

No further modifications were required in the later stages of the infrastructure.

6.5.2. Cross-zone

The first step in investigating whether more tasks could be scheduled than the base model allows was the implementation of the cross-area extension. This extension builds on the base model, but additional modifications were required to ensure that it functions correctly. First, there were minor modifications in the model itself, which required changes in both the model preparation and model construction in the optimizer section, as explained in Section 5.3.1. Furthermore, some adjustments were necessary in preprocessing. In preprocessing, it is determined which people are qualified for a task. In GroundOpt and in our base model, it is also checked whether the person works in the same area as the task is located. Consequently, in the optimizer section, only shifts from the “correct” area are known to be qualified for a task. In the cross-area model, it is essential that all qualified shifts are available in the optimizer section. Therefore, modifications to preprocessing were required.

6.5.3. Postponement

The second action we investigated is postponing flights. Again, the starting point is the Base Model, although several adjustments were made. In addition to the modifications of the model itself, as discussed in Section 5.3.2, we had to adjust some important sets in the data preparation within the optimizer infrastructure. One of the sets we create determines who can perform which task and considers the timing of tasks. This is done to avoid creating unnecessary variables. In the base model, we only check whether the task time falls within the working hours of a shift. In this extension, the goal is to schedule more tasks by postponing flights, because a task occurs later, someone who could not perform it before might now be able to. Therefore, this set was adjusted accordingly. Additionally, a small

change was made in postprocessing to display in the GUI which flights are postponed in our model.

6.5.4. Disruptions

Finally there were changes made in the infrastructure to be able to simulate the disruptions including flight delays and staff shortage.

Due to the infrastructure, the delaying of flights is implemented in preprocessing. First, it is determined for how many flights tasks need to be scheduled. Then, a percentage of these flights is selected to receive a delay. After that, the number of delay minutes is chosen, and for each task associated with the delayed flights, the task is replaced by an identical task whose start and end times are shifted by the delay duration. Since these tasks are also stored in multiple places for the GUI later on, it is important that the updated end time is correctly reflected in every location where the task is stored.

The second type of disruption is staff shortage. This disruption is implemented at the beginning of the optimizer section. First, the total number of staff is determined, and a percentage of them is randomly selected as “unavailable.” These unavailable staff members are then excluded from the sets that determine which tasks each person can perform. Additionally, some adjustments were necessary to ensure that if a member of a team is absent, the remaining team member is treated as a separate shift rather than as part of a team, so that they can still be assigned to tasks appropriately.

Results

In this chapter, we present the results of the experiments conducted with the scheduling models developed in this thesis. The analysis evaluates model performance under normal operations as well as in scenarios with flight delays and staff shortages. We first provide the experimental setup and evaluation metrics, followed by a detailed analysis of *Instance 1* across all work areas. Finally, results for the other instances (*Instances 2–5*) are summarized to highlight general patterns and insights.

7.1. Experimental setup

The aim of the experiments is to evaluate the performance of three model variants under various operational conditions, including scenarios with and without disruptions. The following model variants are being considered:

- **Base Model:** The deterministic planning within a single work area, assuming all task start times and durations are fixed and known in advance. The model aims to maximize task coverage without introducing any changes to the scheduled start times of flights.
- **Cross-Zone Model:** A model extension allowing resources to be assigned to tasks in different areas. Preliminary experiments were conducted for both a 3-hour and an 8-hour planning horizon. For the 3-hour horizon, the complete fix-and-optimize procedure could be executed, although the total runtime increased to approximately 4 hours. The improvement in task coverage was limited: out of 14 initially unassigned tasks, 3 additional tasks could be scheduled by relocating 2 tasks across areas. For the 8-hour horizon, however, the computation time became excessive. After 10 hours the solver was terminated before the full fix-and-optimize strategy had completed. This means that a feasible solution was obtained, but likely not the best one achievable within the heuristic framework. Given the excessive runtime and the minor improvement in task coverage, the model does not scale to practical horizons and is therefore excluded from the experiments conducted for this thesis. More detailed results of the two preliminary experiments can be found in Table A.1 in Appendix A.
- **Postpone Model:** An extension of the base model in which the optimizer is allowed to postpone flights up to 60 minutes in order to increase the number of tasks that can be assigned. Each postponement is penalized in the objective function to balance the trade-off between task coverage and on-time execution.

Two types of disruptions are considered to analyze the impact on the planning model, as described in Section 5.4. The first type involves flight delays, where a certain percentage of flights experience external delays. These delays are randomly assigned with a duration uniformly distributed between one and sixty minutes. The second type concerns staff shortage disruptions, which simulate scenarios in which a percentage of shifts become unavailable due to factors such as sickness or other unforeseen circumstances.

The analysis includes a baseline scenario with no disruptions, serving as a reference point. In addition, for both types of disruptions, flight delays and staff shortages, several disruption scenarios are considered. For the flight delay disruption we consider scenarios where 5%, 10%, 15% or 20% of the flights is delayed. For the staff shortage disruptions we only look at scenarios where 5% or 10% of the staff is unavailable. These percentages are chosen based on input of KLM experts, where a 5% staff shortage represents a common shortage, while 10% reflects more exceptional but still realistic circumstances. Higher absence levels, such as those caused by large scale train disruptions or strikes, are possible but beyond the current scope. For flight delays, equal step sizes were used to enable comparison between scenarios, although a 20% delay rate is rare, including it allows for testing the model's performance under severe disruption conditions.

In addition to disruption scenarios, five different days are selected for the experiments in this thesis, each referred to as *Instance 1* through *Instance 5*. This choice ensures that the evaluation captures model performance across multiple operational instances rather than a single day. The corresponding day and calendar dates are summarized in Table 7.1. For each scenario, an 8-hour planning horizon is considered, covering tasks between 07:00 and 15:00. This time window was chosen because it covers a full shift of ground handling staff. The days included in the experiments were selected to represent typical operational conditions, without extreme events such as storms or mass flight cancellations. This ensures that the analysis focuses on “normal” days, where the system operates close to its standard capacity and performance, thereby providing a baseline for evaluating the model's results.

Table 7.1: Overview of days used in the experiments.

Instance	Date
Instance 1	03-07-2025
Instance 2	04-07-2025
Instance 3	12-09-2025
Instance 4	08-09-2025
Instance 5	11-09-2025

For *Instance 1*, the model is evaluated under several disruption levels: a baseline scenario without disruptions, four levels of flight delays (5%, 10%, 15%, and 20%), and two levels of staff shortages (5% and 10%). The two types of disruptions are considered separately; combinations of flight delays and staff shortages are not included. To account for randomness in the model—such as the random selection of delayed flights and removed shifts, ten independent runs are performed for each disruption level. Each disruption level is evaluated for the three work areas (K1, K4, and K5/6), resulting in a total of 210 runs for *Instance 1*.

For the instances 2 through 5, only one work area (K1) is considered. For each instances a subset of the disruption scenarios are evaluated, the baseline scenario (no disruptions), flight delay levels of 10% and 20%, and staff shortage levels of 5% and 10%. For these instances each scenario was tested through five independent runs, resulting in a total of 25 runs per instance.

This experimental design ensures a detailed evaluation of model performance under a wide range of disruption conditions for *Instance 1*, where all disruption levels are tested across all three work areas with ten independent runs per setting, while still providing representative results for the other instances (Instances 2–5), where a subset of disruption levels is tested for one work area per instance with five runs per configuration. This approach captures variability across different operational days while keeping computational effort manageable.

For each run the following metrics are noted:

- Task coverage, percentage of tasks that are successfully assigned.
- Flights postponed (Postpone Model only), percentage of flights to be postponed.
- Solve time, total computation time to reach the solution.

These metrics are used both to compare the Base Model and Postpone Model for the same disruption scenario, and to analyze how performance changes across disruption levels for each model and area.

flights, and computation time.

7.2.1. Overall comparison of Base and Postpone Model

First the results of the Base and Postpone Models in scenarios without disruptions are analyzed and compared with the activity slot model currently used by KLM. To enable a fair comparison, minor adjustments were made to the KLM model: its objective function was modified to match that of our models, focusing solely on minimizing the number of unplanned tasks. Additionally, certain specific business constraints were removed to align the models, as these constraints are not included in our formulations.

Tables 7.2 to 7.7 present the results for *Instance 1* without disruptions across all work areas. Tables 7.2, 7.4, and 7.6 show the results of the original GroundOpt model, while Tables 7.3, 7.5, and 7.7 present the results of the model developed in this thesis. Within each table, results for the Base Model are presented first, followed by the Postpone Model, which allows flight to be postponed to schedule additional tasks.

Table 7.2: Overview of the results from the GroundOpt model of *Instance 1*, area K1

Run	Number of tasks to be planned	Number task unplanned	Percentage of tasks planned	Total run time
1	624	18.0	97.12	00:01:17
2	622	23.0	96.30	00:01:10
3	622	18.0	97.11	00:01:11
4	622	23.0	96.30	00:01:23
5	624	20.0	96.79	00:01:39
6	624	22.0	96.47	00:01:21
7	624	18.0	97.12	00:01:23
8	622	20.0	96.78	00:01:20
9	622	21.0	96.62	00:01:35
10	624	24.0	96.15	00:01:26
Average	623	20.7	96.68	00:01:23

Table 7.3: Overview of the results of *Instance 1*, area K1. The number of flights for which the tasks that need to be planned is 216.

Run	Deterministic Model				Postpone model				
	Number of tasks to be planned	Number task unplanned	Percentage of tasks planned	Total run time	Percentage task planned	Number of flights postponed	Percentage of flights postponed	Total flight postpone-ments(min)	Total run time
1	624	9	98.6	00:26:15	100	5	2.31	52	01:01:20
2	618	9	98.5	00:31:46	100	5	2.31	54	01:23:43
3	620	9	98.6	00:26:37	100	5	2.31	87	01:33:35
4	616	9	98.5	00:23:08	100	6	2.78	54	01:35:48
5	620	9	98.6	00:27:15	100	5	2.31	51	01:40:10
6	618	9	98.5	00:23:31	100	5	2.31	33	01:28:48
7	622	9	98.6	00:31:45	100	6	2.78	50	01:22:12
8	622	9	98.6	00:33:09	100	6	2.78	53	01:39:08
9	620	9	98.6	00:31:26	100	6	2.78	57	01:48:42
10	622	9	98.6	00:28:00	100	5	2.31	43	01:31:29
	620.2	9	98.55	00:28:17	100	5.4	2.50	53.4	01:30:30

Comparing the base model with the GroundOpt model reveals that more tasks can be scheduled in all areas using the base model. This is likely due to the GroundOpt model dividing time into discrete slots. While the number of unscheduled tasks is minimized within each slot, this approach does not take into account the subsequent slots, resulting in more tasks remaining unscheduled overall. However, the developed base model requires significantly more time.

Another observation is that the base model consistently finds the same number of scheduled tasks across runs suggesting the fix-and-optimize approach performs well and may even reach the global

Table 7.4: GroundOpt model K4 *Instance 1*

Run	Number of tasks to be planned	Number task unplanned	Percentage of tasks planned	Total run time
1	347	26	92.51	00:01:45
2	345	25	92.75	00:01:46
3	338	26	92.31	00:02:27
4	338	25	92.60	00:01:32
5	347	26	92.51	00:01:39
6	341	24	92.86	00:02:04
7	336	26	92.26	00:02:16
8	341	22	93.55	00:01:42
9	353	29	91.78	00:01:22
10	346	25	92.77	00:01:32
Average	343.2	25.4	92.59	00:01:49

Table 7.5: Overview of the results of *Instance 1*, area K4. The number of flights for which the tasks that need to be planned is 113.

Run	Deterministic Model				Postpone model				
	Number of tasks to be planned	Number task unplanned	Percentage of tasks planned	Total run time	Percentage task planned	Number of flights postponed	Percentage of flights postponed	Total flight postponed (min)	Total run time
1	351	1	99.72	00:41:52	100	1	0.88	3	02:27:11
2	341	1	99.71	01:08:17	100	1	0.88	11	02:29:09
3	345	1	99.71	00:56:44	100	1	0.88	11	02:36:23
4	355	1	99.72	01:12:03	100	1	0.88	22	02:47:17
5	351	1	99.72	01:18:19	100	1	0.88	3	02:50:27
6	343	1	99.71	00:34:28	100	1	0.88	3	02:32:38
7	353	1	99.72	00:47:36	100	1	0.88	17	02:34:49
8	351	1	99.72	01:05:11	100	1	0.88	7	02:47:35
9	347	1	99.71	00:52:09	100	1	0.88	3	02:50:04
10	345	1	99.71	01:16:55	100	2	1.77	19	03:02:18
	345	1	99.72	00:59:15	100	1.1	0.97	9.9	02:41:47

Table 7.6: GroundOpt model K5 *Instance 1*

Run	Number of tasks to be planned	Number task unplanned	Percentage of tasks planned	Total run time
1	364	63	82.69	00:00:28
2	361	60	83.38	00:00:29
3	359	59	83.57	00:00:23
4	363	61	83.20	00:00:34
5	363	61	83.20	00:00:21
6	361	58	83.93	00:00:23
7	360	60	83.33	00:00:20
8	361	59	83.66	00:00:22
9	361	60	83.38	00:00:20
10	359	59	83.57	00:00:26
	361.2	60	83.39	00:00:25

optimum. In contrast, the objective values of the Postpone Model vary across runs, indicating the fix and optimize doesn't reach the global optimum. Nevertheless, allowing flights to get postponed enables all tasks to be scheduled across all areas.

When comparing the total run times across areas, an interesting observation can be made: despite involving fewer tasks, area K4 required substantially longer computation times than K1 and K5/6. This can be explained by the behavior of the Fix-and-Optimize procedure. For K1 and K5/6 new best solutions were frequently found in subsequent iterations. In contrast, for K4, the final solution was already obtained after the first few iterations, but all remaining iterations were still executed while no further improvements were achieved. As a result, the runtime was dominated by redundant iterations.

Table 7.7: Overview of the results of *Instance 1*, area K5/6. The number of flights for which the tasks that need to be planned is 75.

Run	Deterministic Model				Postpone model				
	Number of tasks to be planned	Number task unplanned	Percentage of tasks planned	Total run time	Percentage task planned	Number of flights postponed	Percentage of flights postponed	Total flight postponed (min)	Total run time
1	354	14	96.05	00:50:12	100	13	17.33	224	02:02:11
2	354	14	96.05	00:32:57	100	10	13.33	217	01:45:44
3	354	14	96.05	00:44:27	100	9	12	221	01:49:14
4	354	14	96.05	00:46:04	100	14	18.67	329	01:54:24
5	354	14	96.05	00:45:59	100	11	14.67	297	01:45:12
6	354	14	96.05	00:37:45	100	13	17.33	388	02:00:02
7	354	14	96.05	00:44:29	100	7	9.33	161	01:44:28
8	354	14	96.05	00:49:06	100	15	20	360	01:49:53
9	354	14	96.05	00:34:17	100	9	12	200	02:08:56
10	354	14	96.05	00:43:12	100	10	13.33	325	01:27:45
	345	14	96.05	00:42:51	100	11.1	14.8	272.2	01:50:47

Analysis of postponed flights in *Instance 1*

Beyond the total number of flights that need to be postponed, it is interesting to investigate whether the same flights, or flights occurring at similar times, are consistently postponed across different runs. Table 7.8 provides an overview of all postponed flights, including the original start time of one of their tasks and the number of runs in which each flight was postponed. For confidentiality reasons, flight numbers are not displayed.

In area K1 (Table 7.8a), three flights are postponed in every run. Other flights are postponed only in selected runs. Most postponements are concentrated within a narrow time window between 08:00 and 09:30, with the earliest postponed flight at 08:09 and the latest at 09:28.

For area K4 (Table 7.8b), nine out of ten runs have a single postponed flight, whereas one run involves two postponements. In seven of the runs, the same flight is postponed, while in two runs a different flight is affected. The run with two postponed flights involves flights scheduled at different times. Overall, postponed flights in K4 occur close together in time (around 09:40), except for one flight.

In area K5/6 (Table 7.8c), postponements are more spread across both flights and time. Across the 10 runs, a total of 39 different flights experience postponements. The most frequently postponed flight, occurs in 7 runs. While the timing of postponements is more varied, the majority occur between 09:00 and 10:00, totaling 40 occurrences across 10 different flights. This suggests that, in K5/6, the Postpone model distributes flight postponements more evenly throughout the planning horizon.

Table 7.8: Overview of flights that get postponed with corresponding time and count, for all runs of *Instance 1*.

(a) K1.

Time	Count
13:58	10
14:08	10
14:01	10
08:31	4
09:01	4
08:21	4
08:15	3
08:21	1
08:16	1
09:28	1
08:00	1
08:26	1
08:31	1
08:31	1
08:33	1
08:31	1

(b) K4.

Flight	Time	Count
09:44	8	
09:36	1	
09:42	1	
10:21	1	

(c) K5/6.

Flight	Time	Count
09:24	7	
12:29	6	
09:27	6	
12:22	6	
09:10	5	
09:02	4	
12:16	4	
11:39	4	
12:36	4	
09:44	4	
08:52	4	
09:24	4	
13:08	3	
11:59	3	
09:07	3	
13:04	3	
09:04	3	
09:02	3	
08:47	3	
11:57	2	
08:47	2	
11:35	2	
10:02	2	
12:51	2	
12:46	2	
12:02	2	
10:18	1	
08:10	1	
11:54	1	
13:51	1	
11:54	1	
11:52	1	
09:04	1	
08:38	1	
11:45	1	
11:26	1	

7.2.2. Impact of flight delay disruption

This subsection analyses the effect of the delay disruptions on the base and postpone models, for the three areas tested for *Instance 1*.

In Appendix B, an overview of the results of all runs for the different disruption scenarios and areas is presented. Table 7.9 summarizes the average outcomes per area and delay scenario across all runs. For each scenario, the table reports the average number of tasks left unplanned, the average percentage of tasks successfully planned, the average number of postponed flights, and the average percentage of flights postponed. An important observation from this table is that, across all runs and thus for every disruption scenario and area, the Postpone Model is always able to construct a schedule in which all tasks are planned by postponing flights. Figure 7.2a visualizes these average results. Across all areas, a similar overall trend can be observed: as the level of delay increases, both the number of unplanned tasks in the Base Model and the number of postponed flights in the Postpone Model initially decrease before increasing again at the highest disruption level.

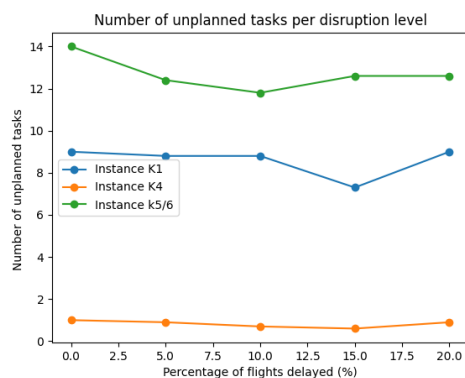
In area K1, the number of unplanned tasks slightly decreases up to 15% delay level, after which it returns to the same level as in the baseline scenario. A similar pattern can be seen in the number of postponed flights, which decreases from around 5.5 to 4.0 and then increases again to just above 5 at the 20% level.

In area K4, the same tendency appears on smaller scale: the number of tasks slightly drops from 1.0 to 0.6 and then rises again to 0.9.

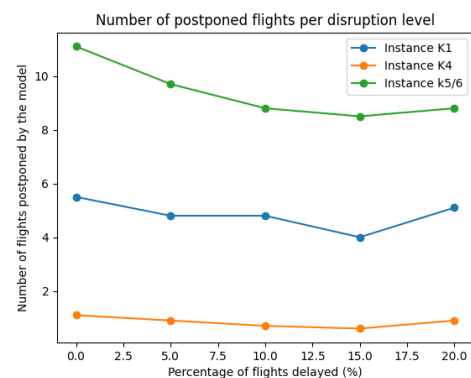
For K5/6, a similar pattern can be observed, although the increase in unplanned tasks already starts at the 15% delay level. At this point, the number of unplanned tasks rises slightly again after the initial decrease. In the Postpone Model, however, the number of flights that need to be delayed remains somewhat below the level observed in the baseline scenario, indicating that fewer postponements are still sufficient to achieve full task coverage.

Table 7.9: Overview of the average results of the ten runs per delay disruption level for all areas for *Instance 1*

Area	Base Model			Postpone Model			
	Disruption level	Number of un-planned tasks	Percentage of tasks planned	Number of un-planned tasks	Percentage of tasks planned	Number of flights postponed	Percentage of flights postponed
K1	0%	9	98.55	0	100	5.5	2.55
K1	5%	8.8	98.58	0	100	4.8	2.22
K1	10%	8.8	98.59	0	100	4.8	2.22
K1	15%	7.3	98.82	0	100	4.0	1.85
K1	20%	9	98.55	0	100	5.1	2.36
K4	0%	1	99.72	0	100	1.1	0.97
K4	5%	0.9	99.74	0	100	0.9	0.79
K4	10%	0.7	99.8	0	100	0.7	0.61
K4	15%	0.6	99.83	0	100	0.6	0.53
K4	20%	0.9	99.74	0	100	0.9	0.79
K5/6	0%	14	96.05	0	100	11.1	14.8
K5/6	5%	12.4	96.5	0	100	9.7	12.93
K5/6	10%	11.8	96.66	0	100	8.8	11.76
K5/6	15%	12.6	96.44	0	100	8.5	11.33
K5/6	20%	12.6	96.44	0	100	8.8	11.73



(a) The number of unplanned tasks per disruption level in the Base Model.



(b) The number of flights postponed by the Postpone model per disruption level.

Figure 7.2: Overview results for the different flight delay disruptions in the different areas for *Instance 1*.

7.2.3. Impact of Staff Shortage disruption

This section analyses the impact of staff shortages on the base and postpone models.

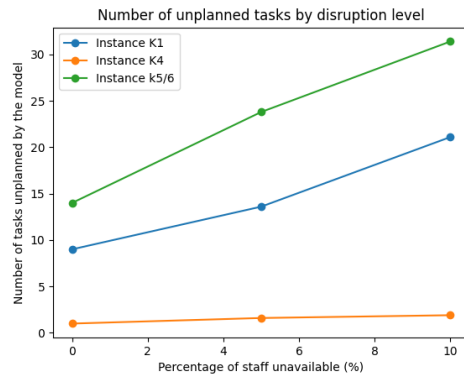
In Appendix B, a complete overview of the results of all runs of the disruption scenarios for staff shortage per area is presented. The average outcomes are summarized in Table 7.10.

The table clearly shows that the level of disruption, measured as the percentage of absent staff, has a significant impact on the feasibility of the planning. In the Base Model, the number of unplanned tasks consistently increases as the disruption level rises across all areas.

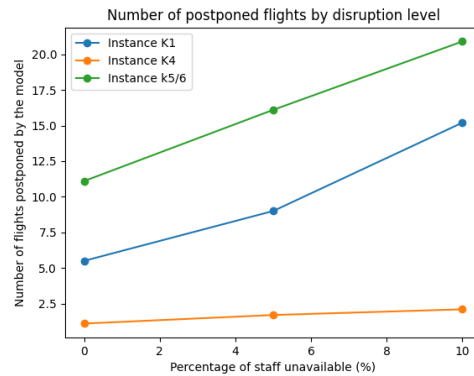
When examining the Postpone Model, where flight postponement is allowed, we observe that the number of flights that need to be postponed is increasing with higher staff absence, although the magnitude of this increase varies by area. Furthermore, even in this model, there are runs where not all tasks can be planned anymore. Specifically, in areas K1 and K4 at the 10% staff absence level, tasks remained unplanned in 3 and 4 runs, respectively (indicated by the non-zero 'Number of unplanned tasks' average).

Table 7.10: Overview of the average results of the ten runs per staff shortage disruption level for all areas for *Instance 1*

Area	Base Model			Postpone model			
	Disruption level	Number of unplanned tasks	Percentage of tasks planned	Number of unplanned tasks	Percentage of tasks planned	Number of flights postponed	Percentage of flights postponed
K1	0%	9	98.55	0	100	5.5	2.55
K1	5%	13.6	97.81	0	100	9	4.1
K1	10%	21.1	96.61	0.4	99.94	15.2	7.04
K4	0%	1	99.72	0	100	1.1	0.97
K4	5%	1.6	99.54	0	100	1.7	1.50
K4	10%	1.9	99.46	0.3	99.91	2.1	1.86
K5/6	0%	14	96.05	0	100	11.1	14.8
K5/6	5%	23.8	93.28	0	100	16.1	21.47
K5/6	10%	31.4	91.13	0	100	20.9	27.87



(a) The number of unplanned tasks per disruption level in the Base Model.



(b) The number of flights postponed by the Postpone model per disruption level.

Figure 7.3: Overview results for the different staff shortage disruptions in the different areas for *Instance 1*.

This general trend of increased postponement is most pronounced in area K5/6. Although the average number of unplanned tasks remains zero in this area, the percentage of postponed flights is notably high and rises sharply. At the 10% staff shortage level, the percentage of flights that must be postponed reaches nearly 28 percent.

7.2.4. Flights postponed

In this subsection, we analyze the flights postponed by the model across all areas, for all different runs, considering both types of disruptions and the varying disruption levels. Before analyzing the postponed flights, it is important to recall that, for each work area, a total of 70 runs are considered: 10 runs for each of the seven disruption scenarios (baseline, four flight delay levels, and two staff shortage levels). All results presented here refer to these 70 runs per area, providing a comprehensive overview of which flights are most frequently postponed under various conditions.

The results for areas K1, K4 and K5/6 are presented in Figure 7.4, Figure 7.5 and 7.6, respectively. Each scatter point represents a flight that was postponed in at least one of the runs. The x-axis indicates the number of minutes the flight was postponed, while the y-axis shows the scheduled start time of a task of the flight ranging from 6:30 to 15:50, which corresponds to the scheduling window between 07:00 and 15:00. Darker points do not necessarily represent the same flight; they may also correspond to different flights that were postponed by a similar number of minutes around the same scheduled start time. This visualization therefore highlights time periods in which postponements occur

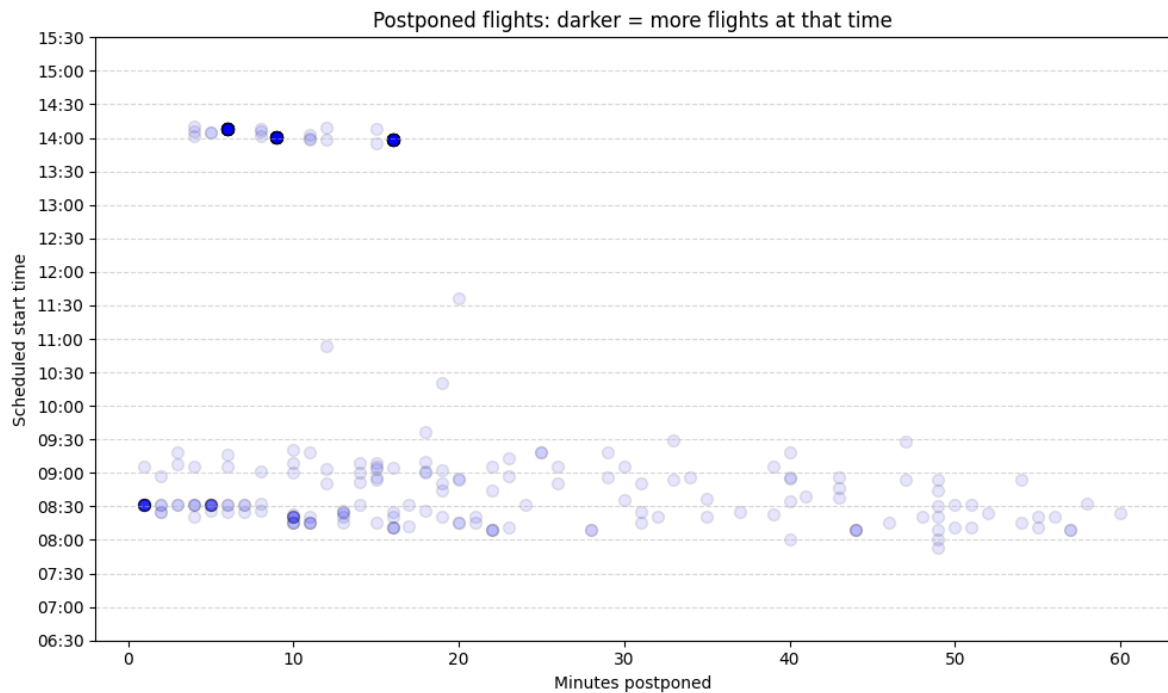


Figure 7.4: Scatterplot giving an overview of the flights postponed by the model and the minutes they get postponed for all different runs of *Instance 1*, area K1.

more frequently. By observing these darker regions, we can identify potential bottlenecks in the schedule, moments during the day where the available workforce or capacity is insufficient, causing multiple flights to be postponed across different runs and disruption levels. Across all three areas, two distinct clusters can be observed in the scatterplots. These clusters align with the main departure and arrival peaks in KLM's daily operation. The concentration of postponements in these time windows reflects the underlying flight schedule, during which most tasks are scheduled and operational pressure is highest.

For area K1, two main clusters of postponed flights are observed: one around 08:30 and another around 14:00. The points around 08:30 are more scattered, with the required delay minutes distributed across the entire 60-minute interval. However, the majority of postponements occur with shorter delay times. Figure 7.7a identifies the flights that are postponed most frequently. It is evident that three flights stand out as being the most affected. The most frequently postponed flights is delayed 68 times and the other two are delayed 67 times. Considering a total of 70 different runs analyzed, this means these three specific flights require postponements in nearly all runs and scenarios.

For area K4, postponements mainly occur between 09:30 and 10:00, with some points observed around 14:00. The flights most frequently postponed in this area are visualized in Figure 7.7b. Notably, for this instance and area, there are runs where no flights are postponed, specifically, 55 out of the 70 runs necessitated flight postponements. Furthermore, one flight is clearly postponed significantly more often than the others: it is delayed 37 times while the next most frequent flight is delayed 15 times.

For area K5/6, the postponed points are much more widely distributed, spanning from 08:30 until short after 14:00. The highest density of points is observed between 09:00 and 10:00, and again between 11:30 and 13:30. Examining the frequency of postponements, as shown in Figure 7.7a, we see that there is no single dominant flight. The most frequently postponed flight appears in slightly more than half of the runs, but the other nine flights are postponed only marginally less often indicating a broader distribution of vulnerable flights across the scenarios.

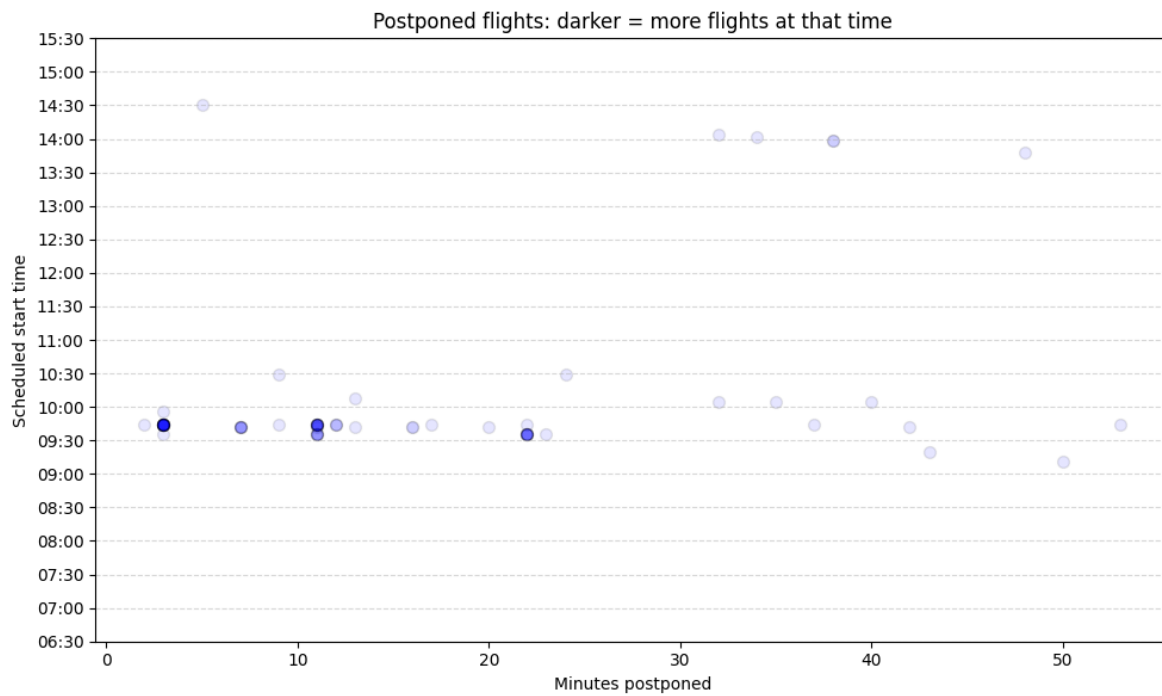


Figure 7.5: Scatterplot giving an overview of the flights postponed by the model and the minutes they get postponed for all different runs of *Instance 1*, area K4.

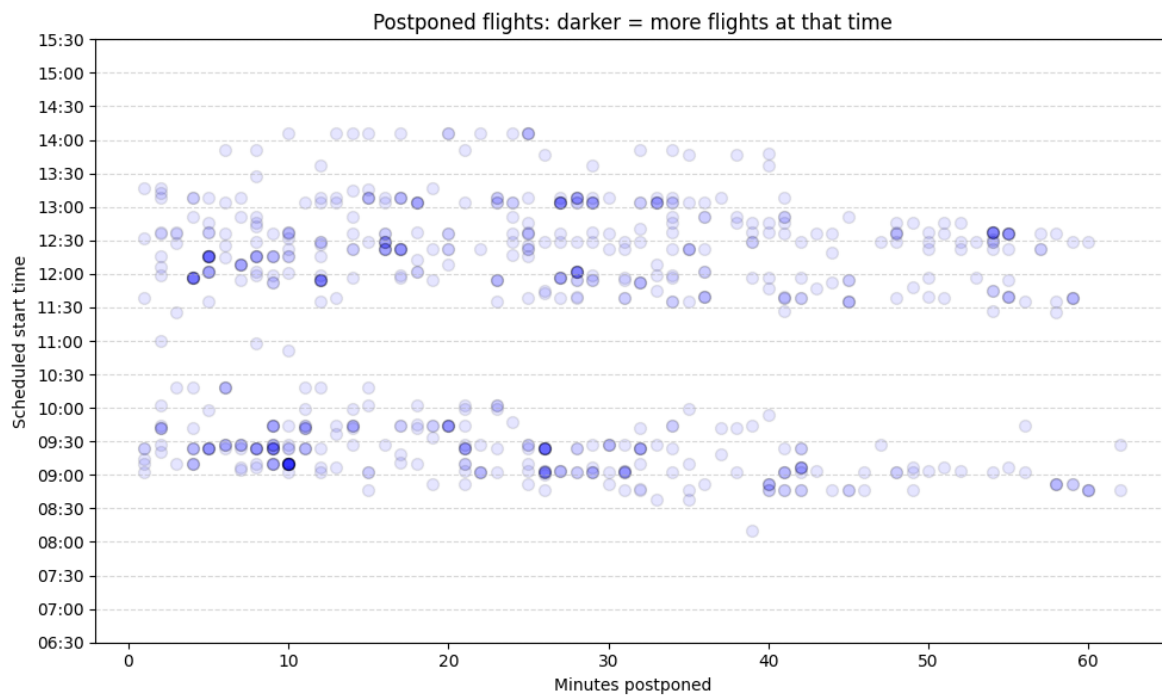


Figure 7.6: Scatterplot giving an overview of the flights postponed by the model and the minutes they get postponed for all different runs of *Instance 1*, area K5/6.

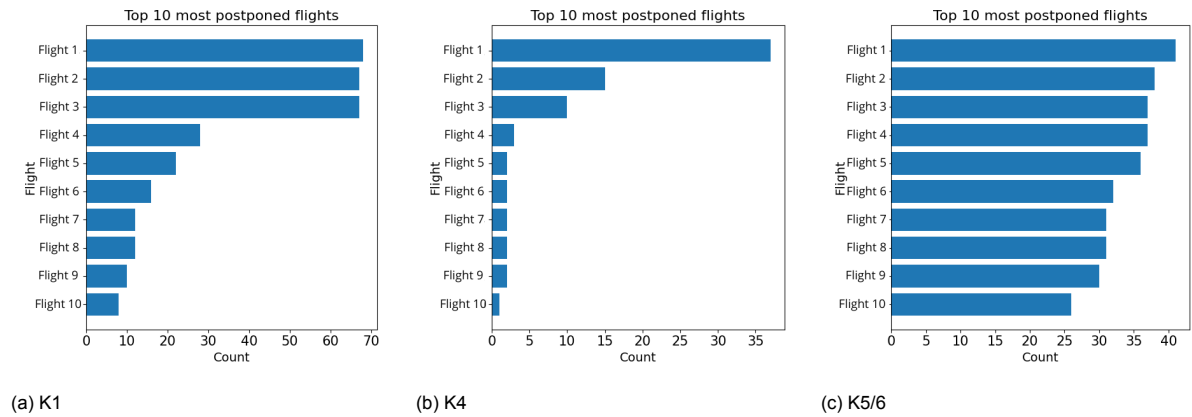


Figure 7.7: Top 10 most frequently postponed flights across all runs for *Instance 1*

7.3. Results of the other instances

In addition to the detailed analysis of *Instance 1*, the performance of the models was also evaluated for four additional instances. Table 7.11 provides an overview of the average number of unplanned tasks and the percentage of tasks successfully planned, comparing the GroundOpt model with the Base model developed in this thesis. More detailed results for each individual run can be found in Appendix B. The results show a clear improvement across all instances: in every case, the Base model is able to schedule a higher share of tasks than the GroundOpt model. The improvement is especially visible in *Instance 5*, where the GroundOpt model leaves about 90 tasks unplanned (about 62%), while the developed Base model, reduces this model to on average less than 20 tasks, reaching almost 97% task coverage.

It is important to note that the instances differ in terms of workload and staff availability. Table 7.11 gives an overview of the number of flights and available staff per instance. As shown, *Instance 5* has the highest number of flights while at the same time only the second lowest number of staff. This imbalance makes it unsurprising, that this instance results in more unplanned tasks in both the GroundOpt and Base model. Figure 7.8 visualizes the consistent improvement of the base model compared to the GroundOpt model across all instances.

Table 7.11: Average number of unplanned tasks and percentage of tasks planned for five instances, comparing the GroundOpt and Base models.

	Number of flights	Number of available staff	GroundOpt		Base model	
			Number of tasks unplanned	Percentage of tasks planned	Number of tasks unplanned	Percentage of tasks planned
<i>Instance 1</i>	216	136	20.7	96.68	9	98.55
<i>Instance 2</i>	213	125	35.8	94.33	10	98.37
<i>Instance 3</i>	224	132	20.4	96.82	10	98.43
<i>Instance 4</i>	224	133	14.6	96.76	8	98.22
<i>Instance 5</i>	236	126	90.2	61.78	18.4	96.98

As mentioned before, the goal of this thesis is not to replace the GroundOpt model, but to use optimization to provide more insight. A key part of this is to understand how different types of disruptions influence the results and which corrective actions, such as postponing flights, can help.

Tables 7.12 and 7.13 show the results for the scenarios with flight delays and staff shortages. Figure 7.9 and Figure 7.10b give a visual overview for the different instances.

In the delay scenarios (Table 7.12), the Base model already reaches a high level of task coverage, with only a few tasks left unplanned. The Postpone model manages to schedule all tasks by shifting a small number of flights.

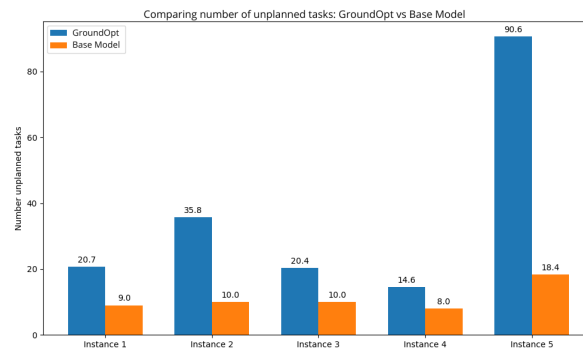


Figure 7.8: Overview of the number of unplanned tasks in the GroundOpt and Base Model

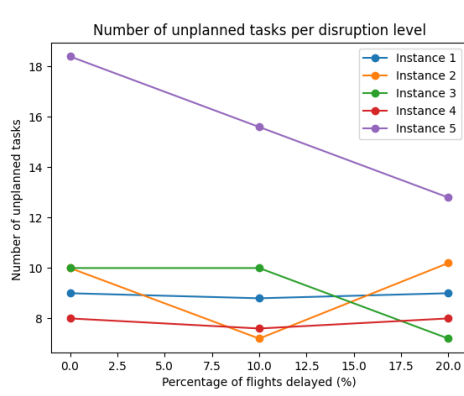
Figure 7.9 visualizes these results. For instances 1,2 and 4, the number of unplanned tasks in the Base Model slightly decreases when 10% of flights are delayed and remains relatively stable or slightly increases again at 20% disruption. A similar pattern is visible for the postponed flights in the Postpone model for these instances, which tends to be lowest at 10% delay. *Instance 5*, however, shows a distinct pattern: both the number of unplanned tasks and the number of postponed flights decrease as the percentage of delayed flights increase, suggesting delays in this instance help smooth the workload over time. *Instance 3* behaves differently, the number of unplanned tasks and postponed flights remain almost constant between 0% and 10% delay, but both decrease noticeably at 20%, indicating that higher delays may incidentally relieve temporal workload peaks in this case. Overall, these results suggest that a moderate level of delays can improve task distribution by reducing workload peaks, although the effect varies per instance.

Table 7.12: Overview results of delay disruptions for the different instances

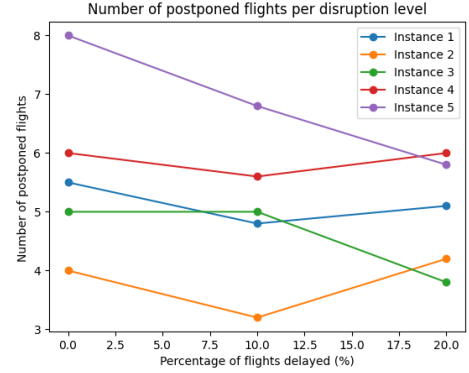
	Disruption level (delays)	Base model		Postpone model			
		Number of tasks unplanned	Percentage of tasks planned	Number of tasks unplanned	Percentage of tasks planned	Number of flights postponed	Percentage of flights postponed
Instance 1	0%	9.0	98.55	0	100	5.5	2.55
	10%	8.8	98.59	0	100	4.8	2.22
	20%	9.0	98.55	0	100	5.1	1.85
Instance 2	0%	10.0	98.37	0	100	4.0	1.88
	10%	7.2	98.83	0	100	3.2	1.50
	20%	10.2	98.33	0	100	4.2	1.98
Instance 3	0%	10.0	98.43	0	100	5.0	2.23
	10%	10.0	98.43	0	100	5.0	2.23
	20%	7.2	98.88	0	100	3.8	1.70
Instance 4	0%	8.0	98.22	0	100	6.0	2.67
	10%	7.6	98.31	0	100	5.6	2.50
	20%	8.0	98.21	0	100	6.0	2.69
Instance 5	0%	18.4	96.98	0	100	8.0	3.39
	10%	15.6	97.45	0	100	6.8	2.89
	20%	12.8	97.91	0	100	5.8	2.48

The staff shortage scenarios (Table 7.13) show a stronger negative effect. As expected, when more staff are absent, the Base model leaves more tasks unplanned. The Postpone model can reduce this problem by postponing flights, but this also creates more disruption in the schedule. In some runs, even the Postpone model cannot schedule all tasks, meaning a few tasks remain unplanned. This problem is again most visible in *Instance 5*, which already combines many flights with relatively few staff. Here, a 10% staff shortage leads to more than 36 unplanned tasks in the Base model and requires postponing almost 9% of flights in the Postpone model to cover nearly all tasks. Instances 2–4, on the other hand, are less sensitive to staff shortages, showing only small changes between the disruption levels.

In addition to summarizing average results, Figure 7.4 for *Instance 1* and Figures C.1-C.4 in Appendix C for the other instances provide a more detailed view of where the model decides to postpone flights.



(a) The number of unplanned tasks per disruption level

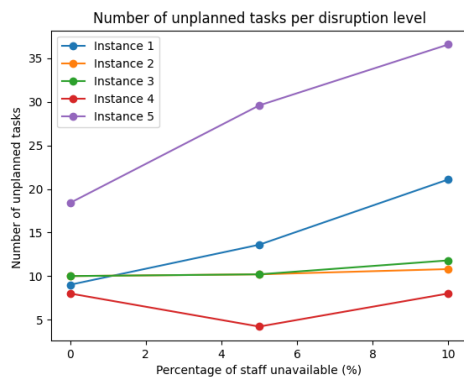


(b) The number of postponed flights in the postpone model per disruption level

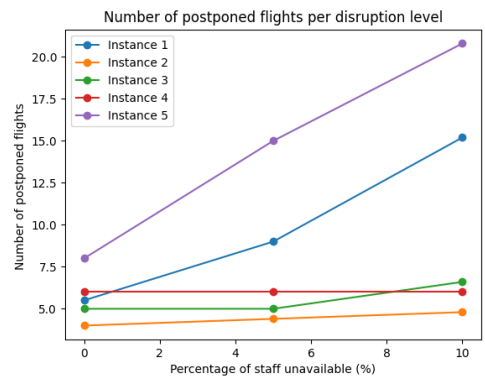
Figure 7.9: Results of our model with delay disruptions for the five instances.

Table 7.13: Overview results of staff shortage disruptions for the different instances

	Disruption level (staff shortage)	Base model		Postpone model			
		Number of tasks unplanned	Percentage of tasks planned	Number of tasks unplanned	Percentage of tasks planned	Number of flights postponed	Percentage of flights postponed
Instance 1	0%	9.0	98.55	0	100	5.5	2.55
	5%	13.6	97.81	0	100	9.0	4.10
	10%	21.1	96.61	0.4	99.94	15.2	7.04
Instance 2	0%	10.0	98.37	0	100	4.0	1.88
	5%	10.2	98.34	0	100	4.4	2.07
	10%	10.8	98.24	0.2	99.97	4.8	2.26
Instance 3	0%	10.0	98.43	0	100	5.0	2.23
	5%	10.2	98.39	0.2	99.97	5.0	2.23
	10%	11.8	98.15	0.2	99.97	6.6	2.95
Instance 4	0%	8.0	98.22	0	100	6.0	2.67
	5%	4.2	98.13	0.4	99.91	6.0	2.67
	10%	8.0	98.22	0	100	6.0	2.67
Instance 5	0%	18.4	96.98	0	100	8.0	3.39
	5%	29.6	95.21	0.4	99.94	15.0	6.37
	10%	36.6	94.04	0.6	99.90	20.8	8.81



(a) The number of unplanned tasks per disruption level



(b) The number of flights postponed by the model per disruption level.

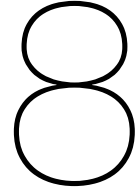
Figure 7.10: Overview results of staff shortage disruptions for the different instances

Each figure shows, for all runs of a given instance, the flights that are delayed by the optimizer and the corresponding delay duration. These visualizations allow us to identify specific time periods in which postponements are required to achieve a feasible and high-coverage schedule.

Across instances, the figures reveal that postponements are not uniformly distributed throughout the day but tend to cluster around certain peak periods. These peaks correspond to moments when many loading and unloading tasks overlap and the available workforce is temporarily insufficient to handle all flights on time. The fact that the same time windows consistently appear across runs and instances suggests that the model successfully detects structural bottlenecks in the planning process.

Moreover, the figures provide insight into the amount of minutes that flights need to be postponed in order to obtain a feasible schedule. The majority of postponements are relatively short, with most points lying below ten minutes, while only a few flights are delayed by more than forty minutes. This indicates that the optimizer typically resolves scheduling conflicts through minor adjustments, and only in exceptional cases larger postponements are introduced. This behavior aligns with the model objective, in which both the number of postponed flights and the total number of postponed minutes are penalized. As a result, the optimizer aims to minimize not only how many flights are delayed, but also the total delay introduced across the schedule, ensuring that postponements remain as short as possible.

These visualizations thus serve as a valuable diagnostic tool: they make it possible to pinpoint when postponements are needed and around which time flights are most affected. This confirms that the postponement logic functions correctly and that the model helps identify the flights and time windows where adjustments are most effective to achieve feasible schedules where as much tasks as possible can be planned.



Conclusion and Discussion

This chapter presents the overall conclusions of this thesis and provides a critical discussion of the findings. The first section summarizes the main results and their implications, structured around the subquestions formulated at the start of the research. The second section reflects on limitations, methodological considerations, and directions for future research.

8.1. Conclusion

The central aim of this thesis was to answer the main research question: *How can more insights be gained into the next day's schedule for staff responsible for loading and unloading baggage, considering potential disruptions?* To address this, five subquestions guided the modeling, implementation, and analysis phases of the research.

The first subquestion focused on accurately modeling the day-ahead scheduling process. By formulating the assignment of tasks as a variant of the Vehicle Routing Problem with Time Windows (VRPTW), the model captured key operational characteristics of KLM's baggage-handling process, including travel times between tasks, mandatory breaks, and multiple task modes. Using this formulation instead of the activity-slot model currently used, facilitated later extensions with corrective actions and disruption scenarios. However, the resulting model was large and could not be solved directly within reasonable time, leading to the use of an iterative Fix-and-Optimize approach, which maintained both solvability and flexibility.

The second subquestion examined whether the model could assess the feasibility of a full schedule. Even in disruption-free conditions, not all tasks could be scheduled due to overlapping task windows and limited staff availability, indicating that the current schedules already do not have enough staff to cover all tasks. Integration with KLM's infrastructure allowed visualization in the planner's GUI, highlighting unassigned tasks and bottlenecks.

The third subquestion investigated corrective strategies to increase task coverage. Two approaches were explored: cross-zone staff assignment and postponements of flight times. Postponements proved most effective, as small shifts in task timing reduced conflicts and increased coverage. Cross-zone assignments were less beneficial.

The fourth subquestion considered disruptions such as flight delays and staff shortages. Moderate delays sometimes increased scheduling flexibility, while staff shortages consistently reduced task coverage. In many scenarios, the Postpone Model was able to schedule all tasks, whereas the Base Model could not.

Finally, the fifth subquestion assessed how disruption impacts could be measured. Comparing unplanned tasks and postponed flights across scenarios provided quantitative insights into schedule robustness, identifying sensitive flights and periods. Although based on a limited set of instances, this

analysis demonstrated the model's potential as a diagnostic tool to support planners before operations begin.

Overall, this research showed that combining an optimization model with scenario-based analysis provides valuable insights into the structure and feasibility of day-ahead schedules. Small delays can increase task coverage, whereas staff shortages reduce it. Academically, this thesis presents a novel modeling framework for large scale, team-based scheduling, which incorporates travel times, breaks, teaming constraints and multiple task modes. A key innovation lies in including mode selection as a decision variable, allowing the model to determine how many staff members handle each flight, which affects task durations and timing. While similar elements have been studied individually, this particular combination of features has received limited attention in existing approaches, offering a more realistic and adaptable representation of baggage handling operations.

8.2. Discussion

The results provide a foundation for insight into day-ahead planning under uncertainty, but several limitations and areas for improvement remain.

A first limitation concerns workload balance. The current model focuses on task feasibility rather than equitable distribution of work. Some staff may be assigned disproportionately many tasks while others remain idle. Incorporating workload fairness constraints or objectives in future research would help generate schedules that are both feasible and implementable.

Another consideration is the computational approach. The Fix-and-Optimize method enabled solving large problem instances iteratively, but the strategy was applied uniformly without adapting to problem characteristics. As observed in tests for area K4 of *Instance 1*, the iterative process continued even when no further solution improvement was found, resulting in unnecessary computational effort. Future work could explore adaptive strategies that adjust groupings dynamically, terminate iterations when no further improvements are observed, or integrate heuristic methods to find high-quality solutions more quickly.

The handling of corrective actions also has limitations. Postponing flights proved effective, but is not always practical, as delays can be costly or operationally infeasible. Future studies could include realistic constraints on which flights may be postponed.

Robustness was only indirectly addressed through scenario analysis. While this provided insights into the impact of delays and staff shortages, the schedules themselves were not explicitly designed to be robust. The disruption scenarios could be improved in several ways. For flight delays, the current model only considered delays for individual flights, without accounting for dependent turnaround flights that might also be delayed. Additionally, delays were simulated using a random number of minutes, whereas future work could analyze historical data to model more realistic delay distributions. For staff disruptions, the current model assumed unavailability at the start of the day. Future research could explore scenarios where staff become unavailable mid-shift due to illness, or larger-scale disruptions such as public transport failures preventing staff from arriving at work. Incorporating these more realistic and complex scenarios, possibly combined with robustness or stochastic optimization techniques, could generate schedules that perform better under a wider range of disruptions while minimizing deviations from the planned schedule.

Integrating the model into KLM's existing GroundOpt system presented both strengths and challenges. Getting the model to work within the infrastructure required significant time, which limited experimentation with heuristics, additional instances, or robustness-focused strategies. On the positive side, integration ensured compatibility with KLM operations and enabled testing different instances and direct visualization in the schedule. A more modular setup, allowing the model to run independently, could provide more flexibility and free time for additional analyses.

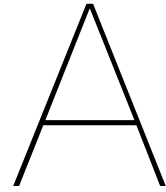
In summary, while this thesis demonstrates that optimization can enhance insight into day-ahead

baggage-handling planning, improvements in workload balancing, computational efficiency, practical applicability of corrective actions, and explicit robustness design are necessary before direct operational implementation. These limitations highlight promising directions for future research that could make schedules not only feasible but also fair, robust, and operationally realistic.

Bibliography

- [1] Yagmur S. Gok et al. "A Simheuristic Approach for Robust Scheduling of Airport Turnaround Teams". In: *2018 Winter Simulation Conference (WSC)* (Dec. 2020), pp. 1336–1347. DOI: 10.1109/wsc48552.2020.9383947. URL: <https://doi.org/10.1109/wsc48552.2020.9383947>.
- [2] R. Müller et al. *Zorgen over veiligheid Schiphol door personeelstekort bagageafhandelaren*. Nov. 2023. URL: <https://nos.nl/artikel/2499417-zorgen-over-veiligheid-schiphol-door-personeelstekort-bagageafhandelaren>.
- [3] E. Hendriks. *Angstcultuur bij medewerkers bagageafhandelaar: 'Uitgeknepen als citroen'*. July 2025. URL: <https://www.omroepbrabant.nl/nieuws/4717035/angstcultuur-bij-medewerkers-bagageafhandelaar-uitgeknepen-als-citroen>.
- [4] J. Rooseboom de Vries. *Nieuw onderzoek: Schiphol-personeel voelt hoge werkdruk en heeft weinig te zeggen*. June 2025. URL: <https://www.nhnieuws.nl/nieuws/349282/nieuw-onderzoek-schiphol-personeel-voelt-hoge-werkdruk-en-heeft-weinig-te-zeggen>.
- [5] KLM [@KLM]. *Ground handling perfection: Boeing 737-800 getting ready for departure*. <https://x.com/KLM/status/1909488747260899595> [Accessed: 05-08-2025]. Apr. 2025.
- [6] KLM [@KLM]. *First day at the office for our @Airbus A321neo*. <https://x.com/KLM/status/1835651097395875857> [Accessed: 05-08-2025]. Sept. 2024.
- [7] M.L. Pinedo. "Scheduling: Theory, Algorithms, and Systems". In: *IIE Transactions* 28.8 (Aug. 1996), pp. 695–697. DOI: 10.1080/15458830.1996.11770714.
- [8] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. <https://www.gurobi.com> [Accessed: 15-07-2025]. 2024.
- [9] A.T. Ernst et al. "Staff scheduling and rostering: A review of applications, methods and models". In: *European Journal of Operational Research* 153.1 (Apr. 2003), pp. 3–27. DOI: 10.1016/s0377-2217(03)00095-x.
- [10] J. Van Den Bergh et al. "Personnel scheduling: A literature review". In: *European Journal of Operational Research* 226.3 (Nov. 2012), pp. 367–385. DOI: 10.1016/j.ejor.2012.11.029.
- [11] T. Clausen. "Airport Ground Staff Scheduling". English. PhD thesis. Technical University of Denmark, Nov. 2010. ISBN: 978-87-90855-95-6.
- [12] C. Barnhart, P. Belobaba, and A.R. Odoni. "Applications of operations research in the air transport industry". In: *Transportation science* 37.4 (2003), pp. 368–391.
- [13] B. Gopalakrishnan and E.L. Johnson. "Airline crew scheduling: State-of-the-art". In: *Annals of Operations Research* 140 (2005), pp. 305–337.
- [14] P. Cappanera et al. "Integrated task scheduling and personnel rostering of airports ground staff: A case study". In: *Expert Systems with Applications* 238 (Oct. 2023), p. 121953. DOI: 10.1016/j.eswa.2023.121953.
- [15] Q. Tian et al. "Solving an airport ground service task assignment problem with an exact algorithm". In: *PLoS ONE* 17.12 (Dec. 2022), e0279131. DOI: 10.1371/journal.pone.0279131.
- [16] A. Dohn, E. Kolind, and J. Clausen. "The manpower allocation problem with time windows and job-teaming constraints: A branch-and-price approach". In: *Computers & Operations Research* 36.4 (2009), pp. 1145–1157. ISSN: 0305-0548. DOI: <https://doi.org/10.1016/j.cor.2007.12.011>.
- [17] G. Andreatta, L. De Giovanni, and M. Monaci. "A fast heuristic for airport Ground-Service Equipment- and-Staff allocation". In: *Procedia - Social and Behavioral Sciences* 108 (Jan. 2014), pp. 26–36. DOI: 10.1016/j.sbspro.2013.12.817.

- [18] A Lim, B. Rodrigues, and L. Song. "Manpower allocation with time windows". In: *Journal of the Operational Research Society* 55.11 (July 2004), pp. 1178–1186. DOI: 10.1057/palgrave.jors.2601782.
- [19] Y. Li, A. Lim, and B. Rodrigues. "Manpower allocation with time windows and job teaming constraints". In: *Naval Research Logistics (NRL)* 52.4 (Mar. 2005), pp. 302–311. DOI: 10.1002/nav.20075.
- [20] G. B. Dantzig and J. H. Ramser. "The truck dispatching problem". In: *Management Science* 6.1 (Oct. 1959), pp. 80–91. DOI: 10.1287/mnsc.6.1.80.
- [21] P. Toth and D. Vigo. *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics, 2002. DOI: 10.1137/1.9780898718515. URL: <https://epubs.siam.org/doi/abs/10.1137/1.9780898718515>.
- [22] J.F. Cordeau et al. "A guide to vehicle routing heuristics". In: *Journal of the Operational Research society* 53.5 (2002), pp. 512–522.
- [23] H.C. Lau, M. Sim, and K.M. Teo. "Vehicle routing problem with time windows and a limited number of vehicles". In: *European journal of operational research* 148.3 (2003), pp. 559–569.
- [24] A. Lim and X. Zhang. "A two-stage heuristic for the vehicle routing problem with time windows and a limited number of vehicles". In: *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*. IEEE. 2005, pp. 82c–82c.
- [25] Z. Li et al. "Improved GRASP with tabu search for vehicle routing with both time window and limited number of vehicles". In: *Innovations in Applied Artificial Intelligence: 17th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE 2004, Ottawa, Canada, May 17-20, 2004. Proceedings 17*. Springer. 2004, pp. 552–561.
- [26] S. Boussier, D. Feillet, and M. Gendreau. "An exact algorithm for team orienteering problems". In: *4or* 5 (2007), pp. 211–230.
- [27] D. Bredström and M. Rönnqvist. "Combined vehicle routing and scheduling with temporal precedence and synchronization constraints". In: *European Journal of Operational Research* 191.1 (Dec. 2007), pp. 19–31. DOI: 10.1016/j.ejor.2007.07.033.
- [28] S. Afifi, D. Dang, and A. Moukrim. "Heuristic solutions for the vehicle routing problem with time windows and synchronized visits". In: *Optimization Letters* 10.3 (Mar. 2015), pp. 511–525. DOI: 10.1007/s11590-015-0878-3.
- [29] S.R.A. Haddadene, N. Labadie, and C. Prodhon. "Bicriteria Vehicle Routing Problem with Preferences and Timing Constraints in Home Health Care Services". In: *Algorithms* 12.8 (July 2019), p. 152. DOI: 10.3390/a12080152.
- [30] J. Decerle et al. "A memetic algorithm for a home health care routing and scheduling problem". In: *Operations Research for Health Care* 16 (Mar. 2018), pp. 59–71. DOI: 10.1016/j.orhc.2018.01.004.
- [31] M. Lindahl, M. Sørensen, and T.R. Stidsen. "A fix-and-optimize matheuristic for university timetabling". In: *Journal of Heuristics* 24.4 (Apr. 2018), pp. 645–665. DOI: 10.1007/s10732-018-9371-3.



Preliminary results cross zone model

Table A.1: Preliminary results for the Cross-Zone Model

Planning horizon	Runtime	Unassigned tasks before allowing Cross-Zone assignments	Unassigned tasks with allowing Cross-Zone assignments	Tasks Cross zone assigned	Procedure completed	Remarks
3 hours (7:30-10:30)	03h 50 min	14	11	2	Yes	3 extra tasks planned by re-locating 2 tasks across areas
8 hours (7:00-15:00)	> 10 hours	24	22	2	No	Solver stopped after 10 h. Feasible solution found but likely suboptimal

B

Results Base and Postpone Models

Table B.1: Overview of the results of Instance 1 with flight delay disruptions, area K1.

Run	Number of flights	Deterministic Model					Delay model				
		Number of tasks to be planned	Number task un-planned	Ratio tasks planned	Run time	Ratio task planned	Objective	Number flights delayed	Ratio flights delayed	Total flight delay (min)	Total run time
5% delayed											
1	216	620	11	98.23	00:31:01	100	6049	6	2.78	49	01:35:56
2	217	622	11	98.23	00:36:59	100	6064	6	2.76	64	01:37:05
3	216	618	11	98.22	00:34:46	100	6082	6	2.78	82	01:51:50
4	216	624	9	98.56	00:35:27	100	5068	5	2.31	68	01:56:08
5	216	620	7	98.87	00:23:39	100	4032	4	1.85	32	01:03:36
6	216	624	4	99.36	00:34:47	100	2025	2	0.93	25	00:59:03
7	216	622	8	98.71	00:28:31	100	4089	4	1.85	89	01:02:29
8	217	618	7	98.87	00:26:50	100	4044	4	1.84	44	01:20:03
9	216	626	11	98.24	00:34:49	100	7060	7	3.24	60	01:25:50
10	216	616	9	98.54	00:29:12	100	4059	4	1.85	59	01:30:53
	216.2	621	8.8	98.58	00:31:36	100	4857.2	4.8	2.22	57.2	01:26:17
10% delayed											
1	217	616	4	99.35	00:21:41	100	2015	2	0.92	15	00:47:01
2	216	620	6	99.03	00:23:15	100	3031	3	1.39	31	00:54:25
3	216	632	9	98.58	00:40:02	100	5054	5	2.31	54	01:19:27
4	216	628	11	98.25	00:23:38	100	6085	6	2.78	85	01:21:04
5	216	622	11	98.23	00:31:04	100	6048	6	2.78	48	01:33:13
6	216	614	6	99.02	00:21:18	100	3031	3	1.39	31	00:45:23
7	216	618	6	99.03	00:22:12	100	3031	3	1.39	31	00:52:22
8	216	624	13	97.92	00:30:32	100	6084	6	2.78	84	01:31:03
9	216	624	13	97.92	00:26:32	100	7084	7	3.24	84	01:33:04
10	216	622	9	98.55	00:33:20	100	7074	7	3.24	74	01:41:36
	216.1	622	8.8	98.59	00:27:21	100	4853.7	4.8	2.22	53.7	01:13:52
15% delayed											
1	216	618	6	99.03	00:31:20	100	3031	3	1.39	31	00:56:51
2	216	616	10	98.38	00:26:41	100	5046	5	2.31	46	00:56:58
3	216	616	8	98.7	00:24:50	100	4047	4	1.85	47	01:04:43
4	216	616	4	99.35	00:25:40	100	2025	2	0.93	25	00:46:23
5	216	612	7	98.86	00:33:44	100	4088	4	1.85	88	01:02:40
6	216	618	6	99.03	00:27:15	100	3031	3	1.39	31	00:51:54
7	217	616	7	98.86	00:29:20	100	4043	4	1.84	43	00:52:19
8	216	624	6	99.04	00:29:05	100	3031	3	1.39	31	00:54:24
9	216	620	10	98.39	00:43:06	100	6081	6	2.78	81	01:46:40
10	217	622	9	98.55	00:28:04	100	6042	6	2.76	42	01:54:30
	216.2	617.8	7.3	98.82	00:29:55	100	4046.5	4	1.85	46.5	01:06:44
20% delayed											
1	216	618	6	99.03	00:26:15	100	3029	3	1.39	29	00:57:09
2	216	618	7	98.87	00:36:12	100	4071	4	1.85	71	01:13:46
3	215	630	13	97.94	00:54:01	100	6099	6	2.79	99	01:53:12
4	217	618	9	98.54	00:28:17	100	9203	9	4.15	203	01:53:27
5	215	614	4	99.35	00:40:37	100	2015	2	0.93	15	01:56:19
6	216	628	9	98.54	00:30:49	100	5086	5	2.31	86	02:09:06
7	216	624	13	97.92	00:31:46	100	7075	7	3.24	75	01:27:01
8	216	626	15	97.6	00:31:39	100	8139	8	3.7	139	02:02:07
9	216	622	4	99.36	00:23:37	100	2022	2	0.93	22	00:51:24
10	216	610	10	98.36	00:28:50	100	5055	5	2.31	55	00:52:00
	215.9	620.8	9	98.55	00:33:12	100	5179.4	5.1	2.36	79.4	01:31:33

Table B.2: Overview of the results of Instance 1 with flight delay disruptions, area K4.

Base Model						Delay model					
Run	number of flights	Number of tasks to be planned	Number task un-planned	Ratio tasks planned	Run time	Ratio task planned	Objective	Number flights delayed	Ratio flights delayed	Total flight delay (min)	Total run time
5% delayed											
1	113	349	0	100	00:26:43	100	0	0	0	0	00:26:55
2	113	343	1	99.71	01:16:32	100	1011	1	0.88	11	02:42:10
3	113	349	1	99.71	01:13:48	100	1003	1	0.88	3	02:57:22
4	113	353	1	99.72	01:14:43	100	1003	1	0.88	3	02:32:58
5	113	353	0	100	00:45:57	100	0	0	0	0	00:46:08
6	113	339	1	99.71	01:15:42	100	1011	1	0.88	11	02:36:09
7	113	343	2	99.42	01:13:49	100	2035	2	1.77	35	02:40:29
8	113	347	2	99.42	01:10:48	100	2022	2	1.77	22	02:45:59
9	113	343	0	100	00:39:10	100	0	0	0	0	00:39:22
10	113	341	1	99.71	01:11:10	100	1012	1	0.88	12	02:22:41
			0.9	99.74	01:02:50	100	909.7	0.9	0.794	9.7	02:03:01
10% delayed											
1	113	347	0	100	00:44:24	100	0	0	0	0	00:44:35
2	113	345	0	100	00:59:31	100	0	0	0	0	00:59:43
3	113	355	0	100	00:40:59	100	0	0	0	0	00:41:10
4	113	357	1	99.72	01:19:26	100	1003	1	0.88	3	03:00:40
5	113	345	1	99.71	01:11:28	100	1003	1	0.88	3	02:46:04
6	113	343	1	99.71	01:07:12	100	1003	1	0.88	3	02:39:33
7	113	345	1	99.71	01:15:28	100	1016	1	0.88	16	02:53:08
8	113	359	2	99.44	01:14:31	100	2055	2	1.77	55	02:35:09
9	112	343	1	99.71	01:18:31	100	1037	1	0.89	37	02:51:19
10	113	349	0	100	00:32:39	100	0	0	0	0	00:32:51
			0.7	99.80	01:02:25	100	711.7	0.7	0.618	11.7	01:58:25
15% delayed											
1	113	349	1	99.71	01:05:43	100	1038	1	0.88	38	02:17:08
2	113	351	2	99.43	01:16:31	100	2015	2	1.77	15	02:55:34
3	113	341	0	100	00:39:06	100	0	0	0	0	00:39:21
4	113	355	0	100	00:47:45	100	0	0	0	0	00:47:59
5	113	347	1	99.71	01:02:39	100	1038	1	0.88	38	02:13:51
6	112	353	1	99.72	04:07:10	100	1007	1	0.89	7	05:39:31
7	113	359	1	99.72	01:14:37	100	1022	1	0.88	22	02:49:09
8	113	345	0	100	00:42:47	100	0	0	0	0	00:42:58
9	113	347	0	100	00:37:20	100	0	0	0	0	00:37:31
10	112	343	0	100	00:32:12	100	0	0	0	0	00:33:22
			0.6	99.83	01:12:35	100	612	0.6	0.53	12	01:55:38
20% delayed											
1	113	339	0	100	00:45:39	100	0	0	0	0	00:45:53
2	113	349	1	99.71	01:08:11	100	1011	1	0.88	11	02:22:13
3	113	351	2	99.43	01:09:09	100	2028	2	1.77	28	02:33:30
4	113	337	0	100	00:37:26	100	0	0	0	0	00:37:41
5	113	337	1	99.7	01:08:50	100	1011	1	0.88	11	02:26:58
6	113	349	1	99.71	01:15:37	100	1022	1	0.88	22	02:44:00
7	113	351	2	99.43	01:10:05	100	2025	2	1.77	25	02:48:07
8	113	341	0	100	00:26:57	100	0	0	0	0	00:27:10
9	113	341	1	99.71	00:57:19	100	1005	1	0.88	5	02:05:23
10	113	335	1	99.7	00:57:54	100	1034	1	0.88	34	02:06:25
			0.9	99.74	00:57:43	100	913.6	0.9	0.794	13.6	01:53:44

Table B.3: Overview of the results of Instance 1 with flight delay disruptions, area K5/K6.

Run	number of flights	Base Model				Delay model					
		Number of tasks to be planned	Number task un-planned	Ratio tasks planned	Run time	Ratio task planned	Objective	Number flights delayed	Ratio flights delayed	Total flight delay (min)	Total run time
5% delayed											
1	75	354	9	97.46	00:44:29	100	15311	15	20	311	01:51:25
2	75	354	10	97.18	00:45:59	100	6163	6	8	163	02:07:21
3	75	354	14	96.05	00:54:43	100	8182	8	10.67	182	01:45:07
4	75	354	18	94.92	00:44:56	100	13299	13	17.33	299	01:57:28
5	75	354	10	97.18	00:45:04	100	6113	6	8	113	01:40:01
6	75	354	10	97.18	00:46:35	100	9128	9	12	128	02:16:56
7	75	354	12	96.61	00:40:44	100	6173	6	8	173	01:35:49
8	75	354	18	94.92	00:44:49	100	15362	15	20	362	01:36:38
9	75	354	14	96.05	00:44:29	100	10253	10	13.33	253	01:49:54
10	75	354	9	97.46	00:49:06	100	9166	9	12	166	01:47:53
		354	12.4	96.50	00:46:05	100	9915	9.7	12.933	215	01:50:51
10% delayed											
1	74	349	20	94.27	00:39:09	100	14343	14	18.92	343	01:27:42
2	75	354	10	97.18	00:40:18	100	8172	8	10.67	172	01:34:37
3	75	354	9	97.46	00:46:43	100	8194	8	10.67	194	01:34:09
4	75	354	15	95.76	00:38:45	100	10279	10	13.33	279	01:38:34
5	75	354	14	96.05	00:52:56	100	9212	9	12	212	01:50:48
6	75	354	8	97.74	00:53:34	100	5166	5	6.67	166	01:48:45
7	75	354	18	94.92	00:46:34	100	12328	12	16	328	01:33:38
8	75	354	5	98.59	00:44:09	100	6142	6	8	142	01:35:15
9	75	354	8	97.74	00:50:18	100	9297	9	12	297	01:53:45
10	75	354	11	96.89	00:42:10	100	7121	7	9.33	121	01:38:41
	74.9	353.5	11.8	96.66	00:45:28	100	9025.4	8.8	11.759	225.4	01:39:35
15% delayed											
1	75	354	22	93.79	00:41:19	100	9286	9	12	286	01:43:27
2	75	354	10	97.18	00:49:48	100	9209	9	12	209	01:46:33
3	75	354	8	97.74	00:48:29	100	9162	9	12	162	01:32:19
4	75	354	10	97.18	00:41:27	100	5112	5	6.67	112	01:37:20
5	75	354	10	97.18	00:43:51	100	5115	5	6.67	115	01:42:04
6	75	354	15	95.76	00:42:58	100	13256	13	17.33	256	01:48:11
7	75	354	14	96.05	00:49:21	100	12280	12	16	280	01:41:50
8	75	354	9	97.46	00:53:30	100	7175	7	9.33	175	01:45:04
9	75	354	10	97.18	00:44:33	100	7130	7	9.33	130	01:50:08
10	75	354	18	94.92	00:44:45	100	9284	9	12	284	01:48:41
	75	354	12.6	96.44	00:46:00	100	8700.9	8.5	11.33	200.9	01:43:34
20% delayed											
1	75	354	14	96.05	00:40:16	100	7193	7	9.33	193	01:45:20
2	75	354	8	97.74	00:39:00	100	10251	10	13.33	251	01:22:46
3	75	354	14	96.05	00:40:23	100	7189	7	9.33	189	01:40:33
4	75	354	23	93.5	00:42:26	100	16336	16	21.33	336	02:01:34
5	75	354	5	98.59	00:41:30	100	2107	2	2.67	107	01:20:36
6	75	354	5	98.59	00:41:08	100	3055	3	4	55	01:24:11
7	75	354	17	95.2	00:41:24	100	9152	9	12	152	01:45:23
8	75	354	11	96.89	00:43:24	100	10153	10	13.33	153	01:44:19
9	75	354	19	94.63	00:42:27	100	15257	15	20	257	01:45:26
10	75	354	10	97.18	00:50:53	100	9164	9	12	164	01:47:00
	75	354	12.6	96.44	00:42:17	100	8985.7	8.8	11.732	185.7	01:39:43

Table B.4: Overview of the results of *Instance 1* with staff shortage disruptions, area K1.

Run	Number of flights	Deterministic Model				Delay model						
		Number of tasks to be planned	Number task un-planned	Percentage tasks planned	Run time	Number of tasks un-planned	Percentage task planned	Objective	Number flights post-poned	Percentage flights Post-poned	Total flight post-ponement (min)	Total run time
5% unavailable												
1	216	612	14	97.71	00:27:48	0.00	100.00	8111	8	3.7	111	02:00:27
2	216	616	14	97.73	00:36:53	0.00	100.00	8159	8	3.7	159	02:13:02
3	216	622	12	98.07	00:34:00	0.00	100.00	6093	6	2.78	93	02:21:09
4	216	618	14	97.73	00:42:40	0.00	100.00	9216	9	4.17	216	02:27:10
5	216	622	12	98.07	00:37:05	0.00	100.00	6091	6	2.78	91	01:44:38
6	216	624	14	97.76	00:46:17	0.00	100.00	7142	7	3.24	142	02:15:10
7	216	626	13	97.92	00:38:48	0.00	100.00	11175	11	5.09	175	02:19:27
8	216	626	14	97.76	00:40:08	0.00	100.00	9172	9	4.17	172	02:27:59
9	216	620	15	97.58	00:59:38	0.00	100.00	12183	12	5.56	183	02:49:18
10	216	628	14	97.77	00:38:39	0.00	100.00	14269	14	6.48	269	02:45:41
	216	621.4	13.6	97.81	00:40:12	0	100	9161.1	9	4.17	161.1	02:20:24
10% unavailable												
1	216	620	23	96.29	00:54:56	1.00	99.84	121421	21	9.72	421	03:13:30
2	216	628	28	95.54	00:49:07	0.00	100.00	20556	20	9.26	556	03:13:49
3	216	618	20	96.76	00:57:06	0.00	100.00	15275	15	6.94	275	03:20:00
4	216	620	25	95.97	00:52:54	1.00	99.84	117569	17	7.87	569	03:22:26
5	216	618	18	97.09	00:58:56	0.00	100.00	12265	12	5.56	265	03:25:46
6	216	624	16	97.44	00:52:04	0.00	100.00	10157	10	4.63	157	02:39:06
7	216	622	21	96.62	00:54:57	1.00	99.84	114323	14	6.48	323	03:02:09
8	216	628	18	97.13	00:51:50	0.00	100.00	11204	11	5.09	204	03:10:01
9	216	624	21	96.63	00:53:40	0.00	100.00	17335	17	7.87	335	03:24:30
10	216	622	21	96.62	00:56:36	1.00	99.84	115360	15	6.94	360	03:18:44
	216	622.4	21.1	96.609	00:54:13	0.4	99.94	55546.5	15.2	7.04	346.5	03:13:00

Table B.5: Overview of the results of *Instance 1* with staff shortage disruptions, area K4.

		Deterministic Model				Delay model						
Run	Number of flights	Number of tasks to be planned	Number task un-planned	Percentage tasks planned	Run time	Number of tasks un-planned	Percentage task planned	Objective	Number flights post-poned	Percentage flights Post-poned	Total flight post-ponement (min)	Total run time
5% unavailable												
1	113	341	1	99.71	01:03:34	0.00	100.00	1003	1	0.88	3	02:22:18
2	113	339	1	99.71	01:21:57	0.00	100.00	1011	1	0.88	11	02:36:52
3	113	351	2	99.43	01:03:18	0.00	100.00	2025	2	1.77	25	02:53:21
4	113	349	4	98.85	01:09:05	0.00	100.00	4173	4	3.54	173	02:57:04
5	113	345	1	99.71	01:08:20	0.00	100.00	1011	1	0.88	11	02:21:44
6	113	347	2	99.42	01:05:40	0.00	100.00	2022	2	1.77	22	02:31:01
7	113	345	1	99.71	01:11:46	0.00	100.00	1011	1	0.88	11	02:32:53
8	113	351	1	99.72	01:16:47	0.00	100.00	1003	1	0.88	3	02:36:35
9	113	343	1	99.71	01:12:26	0.00	100.00	1003	1	0.88	3	02:37:46
10	113	359	2	99.44	01:01:24	0.00	100.00	3061	3	2.65	61	02:51:11
	113	347	1.6	99.54	01:09:26	0	100	1732.3	1.7	1.50	32.3	02:38:05
10% unavailable												
1	113	355	2	99.44	00:59:18	0.00	100.00	2022	2	1.77	22	02:14:28
2	113	347	2	99.42	00:56:00	0.00	100.00	4074	4	3.54	74	02:27:22
3	113	351	2	99.43	01:05:21	0.00	100.00	3038	3	2.65	38	02:40:24
4	113	355	2	99.44	01:01:46	0.00	100.00	3069	3	2.65	69	02:20:46
5	113	355	1	99.72	01:06:30	0.00	100.00	1003	1	0.88	3	02:22:15
6	113	351	1	99.72	01:08:10	0.00	100.00	1007	1	0.88	7	02:27:45
7	113	353	3	99.15	01:04:06	1.00	99.72	103082	3	2.65	82	02:29:43
8	113	341	2	99.41	01:19:20	0.00	100.00	2025	2	1.77	25	03:09:52
9	113	347	2	99.42	01:09:50	1.00	99.71	101011	1	0.88	11	02:21:12
10	113	351	2	99.43	01:13:06	1.00	99.72	101003	1	0.88	3	02:41:18
	113	350.6	1.9	99.46	01:06:21	0.3	99.91	32133.4	2.1	1.86	33.4	02:31:30

Table B.6: Overview of the results of Instance 1 with staff shortage disruptions, area K5/K6.

Run	Number of flights	Deterministic Model				Delay model						
		Number of tasks to be planned	Number task un-planned	Percentage tasks planned	Run time	Number of tasks un-planned	Percentage task planned	Objective	Number flights post-poned	Percentage flights Post-poned	Total flight post-ponement (min)	Total run time
5% unavailable												
1	75	354	20	94.35	00:39:53	0	100	19505	19	25.33	505	01:40:13
2	75	354	24	93.22	00:37:17	0	100	14426	14	18.67	426	01:43:00
3	75	354	24	93.22	00:46:37	0	100	13387	13	17.33	387	01:47:22
4	75	354	22	93.79	00:47:28	0	100	16417	16	21.33	417	01:56:06
5	75	354	26	92.66	00:53:06	0	100	16516	16	21.33	516	01:58:53
6	75	354	22	93.79	00:40:47	0	100	15502	15	20	502	01:31:32
7	75	354	24	93.22	00:37:26	0	100	18414	18	24	414	01:32:22
8	75	354	27	92.37	00:34:46	0	100	20378	20	26.67	378	01:36:09
9	75	354	27	92.37	00:41:48	0	100	15482	15	20	482	01:36:44
10	75	354	22	93.79	00:43:17	0	100	15414	15	20	414	02:10:24
	75	354	23.8	93.28	00:42:15	0	100	16544.1	16.1	21.47	444.1	01:45:17
10% unavailable												
1	75	354	32	90.96	00:43:43	0	100	18546	18	24	546	02:00:20
2	75	354	30	91.53	00:34:17	0	100	26649	26	34.67	649	01:29:33
3	75	354	31	91.24	00:47:57	0	100	25737	25	33.33	737	01:38:54
4	75	354	24	93.22	00:42:28	0	100	17444	17	22.67	444	01:39:36
5	75	354	33	90.68	00:37:42	0	100	22536	22	29.33	536	01:44:09
6	75	354	28	92.09	00:35:33	0	100	17509	17	22.67	509	01:27:04
7	75	354	36	89.83	00:43:39	0	100	22556	22	29.33	556	01:43:32
8	75	354	40	88.7	00:43:33	0	100	23597	23	30.67	597	01:48:38
9	75	354	32	90.96	00:49:40	0	100	21508	21	28	508	01:49:18
10	75	354	28	92.09	00:48:31	0	100	18425	18	24		01:59:20
	75	354	31.4	91.13	00:42:42	0	100	21450.7	20.9	27.87	508.2	01:44:02

Table B.7: Overview of the results for the different disruption scenarios for *Instance 2*.

Run	number of flights	Deterministic Model				Delay model						
		Number of tasks to be planned	Number task un-planned	Ratio tasks planned	Run time	number un-planned tasks	Ratio task planned	Objective	Number flights de-layed	Ratio flights delayed	Total flight delay (min)	Total run time
No disruptions												
1	213	614	10	98.37	00:15:59	0.00	100	4029	4	1.88	29	00:35:50
2	213	612	10	98.37	00:15:26	0.00	100	4029	4	1.88	29	00:39:29
3	213	614	10	98.37	00:18:48	0.00	100	4027	4	1.88	27	00:47:40
4	213	614	10	98.37	00:32:03	0.00	100	4029	4	1.88	29	00:54:18
5	213	612	10	98.37	00:26:14	0.00	100	4027	4	1.88	27	00:54:15
	213	613.2	10	98.37	00:21:42	0.00	100	4028.2	4	1.88	28.2	00:46:18
10% flights delayed												
1	213	614	9	98.53	00:17:09	0.00	100	4027	4	1.88	27	00:41:21
2	213	616	6	99.03	00:23:29	0.00	100	3019	3	1.41	19	00:41:31
3	214	614	6	99.02	00:16:04	0.00	100	3023	3	1.4	23	00:45:02
4	213	614	10	98.37	00:17:37	0.00	100	4032	4	1.88	32	00:46:23
5	213	614	5	99.19	00:18:16	0.00	100	2020	2	1.5	20	00:35:00
	213.2	614.4	7.2	98.83	00:18:31	0.00	100	3224.2	3.2	1.5	24.2	00:41:51
20% flights delayed												
1	211	608	6	99.01	00:21:17	0.00	100	2020	2	0.95	20	00:47:47
2	212	620	10	98.39	00:16:46	0.00	100	4036	4	1.89	36	00:38:42
3	212	610	6	99.02	00:18:46	0.00	100	3019	3	1.42	19	00:40:35
4	211	608	8	98.68	00:19:21	0.00	100	3023	3	1.42	23	00:51:17
5	213	614	21	96.58	00:26:56	0.00	100	9136	9	4.23	136	01:31:16
	211.8	612	10.2	98.33	00:20:37	0.00	100	4246.8	4.2	1.98	46.8	00:53:55
5% staff shortage												
1	213	616	10	98.38	00:21:00	0.00	100.00	4032	4	1.88	32	00:56:51
2	213	614	10	98.37	00:32:37	0.00	100.00	4032	4	1.88	32	01:06:14
3	213	616	10	98.38	00:20:25	0.00	100.00	5041	5	2.35	41	01:10:21
4	213	614	10	98.37	00:25:38	0.00	100.00	4029	4	1.88	29	01:15:37
5	213	616	11	98.21	01:01:34	0.00	100.00	5046	5	2.35	46	02:01:18
	213	615.2	10.2	98.34	00:32:15	0	100	4436	4.4	2.07	36	01:18:04
10% staff shortage												
1	213	612	10	98.37	00:24:56	0.00	100.00	4027	4	1.88	27	01:02:09
2	213	614	11	98.21	00:22:29	1.00	99.84	104032	4	1.88	32	01:12:41
3	213	612	10	98.37	00:41:31	0.00	100.00	5041	5	2.35	41	01:53:10
4	213	612	12	98.04	00:39:38	0.00	100.00	6065	6	2.82	65	01:53:56
5	213	612	11	98.2	00:57:42	0.00	100.00	5038	5	2.35	38	02:10:16
	213	612.4	10.8	98.24	00:37:15	0.2	99.97	24840.6	4.8	2.27	40.6	01:38:26

Table B.8: Overview of the results for the different disruption scenarios for *Instance 3*.

Run	number of flights	Deterministic Model				Delay model						
		Number of tasks to be planned	Number task un-planned	Ratio tasks planned	Run time	number un-planned tasks	Ratio task planned	Objective	Number flights delayed	Ratio flights delayed	Total flight delay (min)	Total run time
No disruptions												
1	224	640	10	98.44	00:27:07	0	100	5041	5	2.23	41	00:57:15
2	224	642	10	98.44	00:32:29	0	100	5041	5	2.23	41	01:12:40
3	224	632	10	98.42	00:29:48	0	100	5041	5	2.23	41	01:19:09
4	224	636	10	98.43	00:59:39	0	100	5041	5	2.23	41	01:33:39
5	224	638	10	98.43	00:28:24	0	100	5041	5	2.23	41	00:57:14
	224	637.6	10	98.43	00:35:29	0	100	5041	5	2.23	41	01:11:59
10% flights delayed												
1	224	640	10	98.44	00:24:07	0	100	5046	5	2.23	46	00:58:58
2	225	634	10	98.42	00:26:27	0	100	5041	5	2.22	41	01:03:57
3	224	638	12	98.12	00:31:49	0	100	6047	6	2.68	47	01:11:04
4	223	632	8	98.73	00:40:29	0	100	4031	4	1.79	31	01:14:44
5	224	642	10	98.44	00:24:17	0	100	5041	5	2.23	41	01:10:40
	224	637.2	10	98.43	00:29:26	0	100	5041.2	5	2.23	41.2	01:07:53
20% flights delayed												
1	224	640	6	99.06	00:23:03	0	100	4018	4	1.79	18	00:52:49
2	224	636	8	98.74	00:29:17	0	100	4035	4	1.79	35	01:04:52
3	225	640	8	98.75	00:26:00	0	100	4036	4	1.78	36	01:04:01
4	225	642	8	98.75	00:35:55	0	100	4036	4	1.78	36	01:17:52
5	224	644	6	99.07	00:21:12	0	100	3021	3	1.34	21	00:47:40
	224.4	640.4	7.2	98.88	00:27:05	0	100	3829.2	3.8	1.70	29.2	01:01:27
5% staff shortage												
1	224	636	10	98.43	00:48:48	0.00	100.00	5041	5	2.23	41	01:43:39
2	224	630	11	98.25	00:36:17	1.00	99.84	105041	5	2.23	41	01:28:36
3	224	636	10	98.43	00:37:03	0.00	100.00	5041	5	2.23	41	01:33:32
4	224	632	10	98.42	00:46:17	0.00	100.00	5041	5	2.23	41	01:34:29
5	224	632	10	98.42	00:51:37	0.00	100.00	5041	5	2.23	41	02:00:05
	224	633.2	10.2	98.39	00:44:00	0.2	99.97	25041	5	2.23	41	01:40:04
10% staff shortage												
1	224	632	12	98.1	00:44:25	1.00	99.84	107110	7	3.13	110	02:23:41
2	224	638	13	97.96	00:37:33	0.00	100.00	9177	9	4.02	177	02:25:24
3	224	644	12	98.14	00:59:51	0.00	100.00	6068	6	2.68	68	03:05:43
4	224	638	10	98.43	00:29:33	0.00	100.00	5041	5	2.23	41	01:41:59
5	224	636	12	98.11	00:39:28	0.00	100.00	6082	6	2.68	82	01:54:52
	224	637.6	11.8	98.148	00:42:10	0.2	99.97	26695.6	6.6	2.948	95.6	02:18:20

Table B.9: Overview of the results for the different disruption scenarios for *Instance 4*.

Run	number of flights	Deterministic Model				Delay model						
		Number of tasks to be planned	Number task un-planned	Ratio tasks planned	Run time	number un-planned tasks	Ratio task planned	Objective	Number flights de-layed	Ratio flights delayed	Total flight delay (min)	Total run time
No disruptions												
1	225	450	8	98.22	00:07:57	0	100	6051	6	2.67	51	00:18:23
2	225	450	8	98.22	00:08:21	0	100	6051	6	2.67	51	00:19:50
3	225	450	8	98.22	00:09:59	0	100	6051	6	2.67	51	00:22:35
4	225	450	8	98.22	00:08:06	0	100	6051	6	2.67	51	00:22:29
5	225	450	8	98.22	00:08:12	0	100	6051	6	2.67	51	00:22:57
	225	450	8	98.22	00:08:31	0	100	6051	6	2.67	51	00:21:15
10% flights delayed												
1	224	448	8	98.21	00:09:08	0	100	6051	6	2.68	51	00:19:57
2	225	450	6	98.67	00:08:27	0	100	4037	4	1.78	37	00:20:43
3	225	450	8	98.22	00:06:51	0	100	6051	6	2.67	51	00:26:13
4	223	446	10	97.76	00:18:05	0	100	7060	7	3.14	60	00:27:11
5	225	450	6	98.67	00:14:24	0	100	5044	5	2.22	44	00:31:01
	224.4	448.8	7.6	98.31	00:11:23	0	100	5648.6	5.6	2.50	48.6	00:25:01
20% flights delayed												
1	223	446	8	98.21	00:06:29	0	100	6048	6	2.69	48	00:14:29
2	223	446	6	98.65	00:08:06	0	100	5044	5	2.24	44	00:16:33
3	224	448	6	98.66	00:06:43	0	100	5032	5	2.23	32	00:17:29
4	225	450	8	98.22	00:07:09	0	100	6044	6	2.67	44	00:18:29
5	222	444	12	97.3	00:07:25	0	100	8064	8	3.6	64	00:20:19
	223.4	446.8	8	98.21	00:07:10	0	100	6046.4	6	2.69	46.4	00:17:28
5% staff shortage												
1	225	450	8	98.22	00:08:46	0.00	100.00	6051	6	2.67	51	00:20:41
2	225	450	10	97.78	00:07:52	2.00	99.56	206051	6	2.67	51	00:21:03
3	225	450	8	98.22	00:09:50	0.00	100.00	6051	6	2.67	51	00:36:37
4	225	450	8	98.22	00:08:43	0.00	100.00	6051	6	2.67	51	00:28:05
5	225	450	8	98.22	00:08:53	0.00	100.00	6051	6	2.67	51	00:35:15
	225	450	8.4	98.13	00:08:49	0.4	99.91	46051	6	2.67	51	00:28:20
10% staff shortage												
1	225	450	8	98.22	00:08:23	0.00	100.00	6051	6	2.67	51	00:29:54
2	225	450	8	98.22	00:08:32	0.00	100.00	6051	6	2.67	51	00:46:29
3	225	450	8	98.22	00:10:04	0.00	100.00	6051	6	2.67	51	00:49:26
4	225	450	8	98.22	00:08:16	0.00	100.00	6051	6	2.67	51	00:26:47
5	225	450	8	98.22	00:11:36	0.00	100.00	6051	6	2.67	51	00:50:40
	225	450	8	98.22	00:09:22	0	100.00	6051	6	2.67	51	00:40:39

Table B.10: Overview of the results for the different disruption scenarios for *Instance 5*.

Run	number of flights	Deterministic Model				Delay model						
		Number of tasks to be planned	Number task un-planned	Ratio tasks planned	Run time	number un-planned tasks	Ratio task planned	Objective	Number flights de-layed	Ratio flights delayed	Total flight delay (min)	Total run time
No disruptions												
1	236	608	18	97.04	00:41:31	0	100	8056	8	3.39	56	01:49:54
2	236	606	19	96.86	00:36:38	0	100	8078	8	3.39	78	02:05:05
3	236	612	18	97.06	00:36:17	0	100	8078	8	3.39	78	02:15:05
4	236	608	17	97.2	00:40:21	0	100	8057	8	3.39	57	02:21:55
5	236	614	20	96.74	00:45:01	0	100	8156	8	3.39	156	02:56:51
	236	609.6	18.4	96.98	00:39:58	0	100	8085	8	3.39	85	02:17:46
10% flights delayed												
1	234	608	12	98.03	00:26:47	0	100	5031	5	2.14	31	01:07:39
2	235	610	14	97.7	00:37:32	0	100	6044	6	2.55	44	01:56:27
3	236	610	16	97.38	00:27:05	0	100	7092	7	2.97	92	01:23:28
4	236	618	18	97.09	00:49:14	0	100	8100	8	3.39	100	02:15:08
5	234	610	18	97.05	00:41:51	0	100	8089	8	3.42	89	02:42:32
	235	611.2	15.6	97.45	00:36:30	0	100	6871.2	6.8	2.894	71.2	01:53:03
20% flights delayed												
1	235	616	10	98.38	00:57:24	0	100	5082	5	2.13	82	02:03:58
2	236	622	8	98.71	00:29:01	0	100	3059	3	1.27	59	01:27:24
3	234	612	16	97.39	00:35:02	0	100	8118	8	3.42	118	01:42:17
4	233	606	14	97.69	00:48:59	0	100	6041	6	2.58	41	01:52:22
5	235	612	16	97.39	00:32:41	0	100	7078	7	2.98	78	01:52:31
	234.6	613.6	12.8	97.91	00:40:37	0	100	5875.6	5.8	2.48	75.6	01:47:42
5% staff shortage												
1	236	618	30	95.15	00:40:17	0.00	100.00	15374	15	6.36	374	02:28:10
2	236	612	29	95.26	00:44:40	1.00	99.84	115380	15	6.36	380	02:49:07
3	236	620	32	94.84	00:44:41	1.00	99.84	116341	16	6.78	341	02:56:58
4	236	620	26	95.81	00:47:07	0.00	100.00	15273	15	6.36	273	03:11:35
5	236	616	31	94.97	00:50:45	0.00	100.00	14365	14	5.93	365	03:19:04
	236	617.2	29.6	95.21	00:45:30	0.4	99.94	55346.6	15	6.36	346.6	02:56:59
10% staff shortage												
1	236	620	34	94.52	00:43:43	0.00	100.00	24479	24	10.17	479	02:28:10
2	236	618	36	94.17	00:43:58	0.00	100.00	18498	18	7.63	498	02:49:07
3	236	614	31	94.95	00:56:10	0.00	100.00	16433	16	6.78	433	02:56:58
4	236	606	44	92.74	00:47:05	0.00	100.00	28630	28	11.86	630	03:11:35
5	236	614	38	93.81	00:59:20	3.00	99.51	318469	18	7.63	469	03:19:04
	236	614.4	36.6	94.038	00:50:03	0.6	99.90	81301.8	20.8	8.81	501.8	02:56:59

C

Overview postponed flights

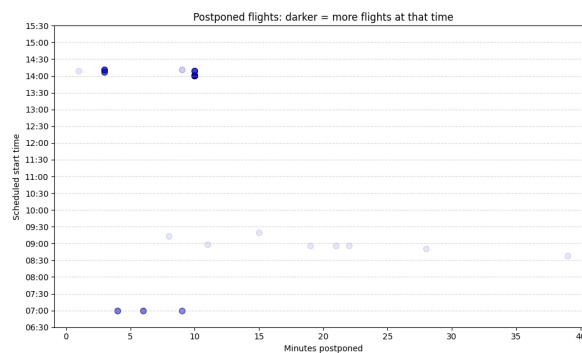


Figure C.1: Overview of all flights postponed by the Postpone Model for *Instance 2*.

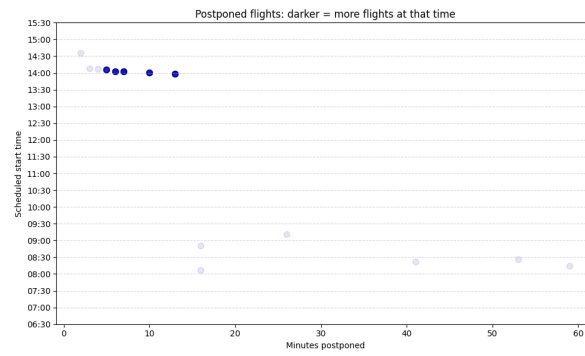


Figure C.2: Overview of all flights postponed by the Postpone Model for *Instance 3*.

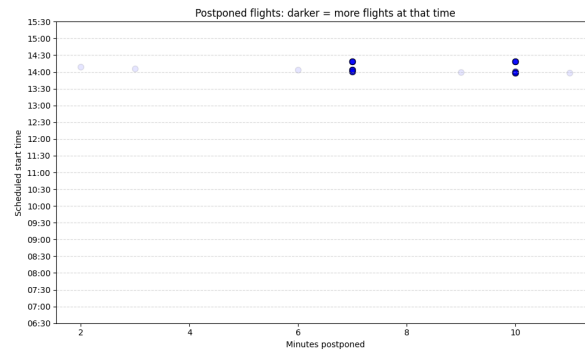


Figure C.3: Overview of all flights postponed by the Postpone Model for *Instance 4*.

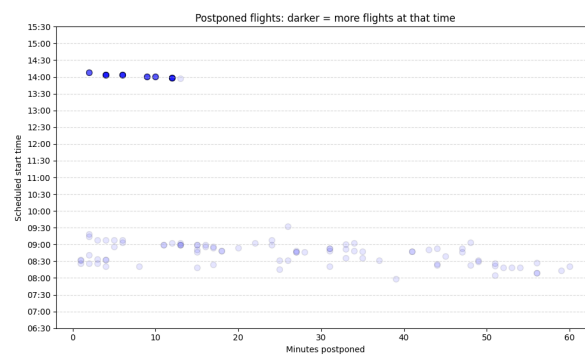


Figure C.4: Overview of all flights postponed by the Postpone Model for *Instance 5*.